



SunOS リファレンスマニュアル 1M: システム管理コマンド

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 817-5082-10
2004 年 4 月

Copyright 2004 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

本製品およびそれに関連する文書は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。フォント技術を含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions.

本製品に含まれる HG-MinchoL、HG-MinchoL-Sun、HG-PMinchoL-Sun、HG-GothicB、HG-GothicB-Sun、および HG-PGothicB-Sun は、株式会社リコーがリコービマジクス株式会社からライセンス供与されたタイプフェースマスタをもとに作成されたものです。HeiseiMin-W3H は、株式会社リコーが財団法人日本規格協会からライセンス供与されたタイプフェースマスタをもとに作成されたものです。フォントとして無断複製することは禁止されています。

Sun、Sun Microsystems、docs.sun.com、AnswerBook、AnswerBook2、JumpStart、Solaris Web Start、Power Management、Sun ONE Application Server、Solaris Flash、Solaris Live Upgrade は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

サンのロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

OPENLOOK、OpenBoot、JLE は、サン・マイクロシステムズ株式会社の登録商標です。

Wnn は、京都大学、株式会社アステック、オムロン株式会社で共同開発されたソフトウェアです。

Wnn6 は、オムロン株式会社、オムロンソフトウェア株式会社で共同開発されたソフトウェアです。© Copyright OMRON Co., Ltd. 1995-2000. All Rights Reserved. © Copyright OMRON SOFTWARE Co., Ltd. 1995-2002 All Rights Reserved.

「ATOK」は、株式会社ジャストシステムの登録商標です。

「ATOK Server/ATOK12」は、株式会社ジャストシステムの著作物であり、「ATOK Server/ATOK12」にかかる著作権その他の権利は、株式会社ジャストシステムおよび各権利者に帰属します。

本製品に含まれる郵便番号辞書 (7 桁/5 桁) は郵政事業庁が公開したデータを元に制作された物です (一部データの加工を行なっています)。

本製品に含まれるフェイスマーク辞書は、株式会社ビレッジセンターの許諾のもと、同社が発行する『インターネット・パソコン通信フェイスマークガイド '98』に添付のものを使用しています。© 1997 ビレッジセンター

Unicode は、Unicode, Inc. の商標です。

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

DtComboBox ウィジェットと DtSpinBox ウィジェットのプログラムおよびドキュメントは、Interleaf, Inc. から提供されたものです。(© 1993 Interleaf, Inc.)

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されず、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

本製品が、外国為替および外国貿易管理法 (外為法) に定められる戦略物資等 (貨物または役務) に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典: *man pages section 1M : System Administration Commands*

Part No: 817-0708-10

Revision A



040304@7940



目次

はじめに 7

SunOS リファレンスマニュアル **1M**: システム管理コマンド **11**

Intro(1M) 12
accept(1M) 17
acct(1M) 19
acctdisk(1M) 22
acctdusg(1M) 25
accton(1M) 28
acctwtmp(1M) 31
add_install_client(1M) 34
add_to_install_server(1M) 43
afbconfig(1M) 52
automount(1M) 61
boot(1M) 69
catman(1M) 91
cfgadm(1M) 95
cfgadm_ac(1M) 106
cfgadm_sysctrl(1M) 110
check(1M) 115
closewtmp(1M) 124
cvcd(1M) 127
dd(1M) 128
df(1M) 134
fdisk(1M) 138
ffbconfig(1M) 144

format(1M) 153
fsck(1M) 158
fuser(1M) 162
GFXconfig(1M) 165
halt(1M) 170
init(1M) 171
installer(1M) 177
install_scripts(1M) 178
kdmconfig(1M) 187
lpmove(1M) 190
lpsched(1M) 192
lpshut(1M) 194
lu(1M) 195
luactivate(1M) 199
lucancel(1M) 201
lucompare(1M) 202
lucreate(1M) 205
lucurr(1M) 221
ludelete(1M) 223
lufslst(1M) 225
lumake(1M) 227
lumount(1M) 229
lurename(1M) 232
lustatus(1M) 234
luumount(1M) 236
luupgrade(1M) 239
luxadm(1M) 250
m64config(1M) 265
mkfile(1M) 270
mkfs(1M) 271
modify_install_server(1M) 273
modinfo(1M) 282
modload(1M) 283
modunload(1M) 285
mount(1M) 286
mountall(1M) 290
newfs(1M) 292
patchadd(1M) 298

patchrm(1M)	309
pgxconfig(1M)	315
pkgadd(1M)	320
pkgrm(1M)	326
pmconfig(1M)	329
powerd(1M)	331
poweroff(1M)	332
prodreg(1M)	333
prtconf(1M)	351
prtdiag(1M)	353
raidctl(1M)	354
reboot(1M)	357
reject(1M)	359
restricted_shell(1M)	361
rm_install_client(1M)	363
rsh(1M)	372
scadm(1M)	374
setup_install_server(1M)	381
share(1M)	390
shareall(1M)	392
showmount(1M)	393
shutdown(1M)	394
su(1M)	396
SUNWafb_config(1M)	400
SUNWffb_config(1M)	409
SUNWm64_config(1M)	418
sys-unconfig(1M)	423
telinit(1M)	425
TSIgfxp_config(1M)	431
ttymon(1M)	436
umount(1M)	440
umountall(1M)	444
unshare(1M)	446
unshareall(1M)	447
utmp2wtmp(1M)	448
wall(1M)	451

はじめに

概要

SunOS リファレンスマニュアルは、初めて SunOS を使用するユーザーやすでにある程度の知識を持っているユーザーのどちらでも対応できるように解説されています。このマニュアルを構成するマニュアルページは一般に参照マニュアルとして作られており、チュートリアルな要素は含んでいません。それぞれのコマンドを実行すると、どのような結果が得られるかについて、詳しく説明されています。なお、各マニュアルページの内容はオンラインでも参照することができます。

このマニュアルは、マニュアルページの内容によっていくつかのセクションに分かれています。各セクションについて以下に簡単に説明します。

- セクション 1 は、オペレーティングシステムで使えるコマンドを説明します。
- セクション 1M は、システム保守や管理用として主に使われるコマンドを説明します。
- セクション 2 は、すべてのシステムコールについて説明します。ほとんどのシステムコールに 1 つまたは複数のエラーがあります。エラーの場合、通常ありえない戻り値が返されます。
- セクション 3 は、さまざまなライブラリ中の関数について説明します。ただし、UNIX システムプリミティブを直接呼び出す関数については、セクション 2 で説明しています。
- セクション 4 は、各種ファイルの形式について説明します。また、ファイル形式を宣言する C 構造体を適用できる場合には随時説明しています。
- セクション 5 は、文字セットテーブルなど他のセクションには該当しないものについて説明します。
- セクション 7 は、特殊なハードウェア周辺装置またはデバイスドライバに関するさまざまな特殊ファイルについて説明します。STREAMS ソフトウェアドライバ、モジュール、またはシステムコールの STREAMS 汎用セットについても説明します。

- セクション 9 は、カーネル環境でデバイスドライバを記述するのに必要な参照情報を提供します。2つのデバイスドライバインタフェース仕様について説明します。デバイスドライバインタフェース (DDI) とドライバ/カーネルインタフェース (DKI) です。
- セクション 9E は、開発者がデバイスドライバに含めることのできる DDI/DKI、DDI のみ、および DKI のみのエントリポイントルーチンについて説明します。
- セクション 9F は、デバイスドライバが使用できるカーネル関数について説明します。
- セクション 9S は、ドライバとカーネル間で情報を共有するためにドライバが使用するデータ構造について説明します。

以下に、このマニュアルの項目を表記されている順に説明します。ほとんどのマニュアルページが下記の項目からなる共通の書式で書かれていますが、必要でない項目については省略されています。たとえば、記述すべきバグがコマンドにない場合などは、「使用上の留意点」という項目はありません。各マニュアルページの詳細は各セクションの intro を、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	コマンドや関数の名称と概略が示されています。
形式	コマンドや関数の構文が示されています。標準パスにコマンドやファイルが存在しない場合は、フルパス名が示されます。字体は、コマンド、オプションなどの定数にはボールド体 (bold) を、引数、パラメータ、置換文字などの変数にはイタリック体 (<i>Italic</i>) または <日本語訳> を使用しています。オプションと引数の順番は、アルファベット順です。特別な指定が必要な場合を除いて、1文字の引数、引数のついたオプションの順に書かれています。
	以下の文字がそれぞれの項目で使われています。
[]	このかっこに囲まれたオプションや引数は省略できません。このかっこが付いていない場合には、引数を必ず指定する必要があります。
...	省略符号。前の引数に変数を付けたり、引数を複数指定したりできることを意味します (例: 'filename.. .')。
	区切り文字 (セパレータ)。この文字で分割されている引数のうち 1 つだけを指定できます。
{ }	この大かっこに囲まれた複数のオプションや引数は省略できます。かっこ内を 1 組として扱います。
プロトコル	この項が使われているのは、プロトコルが記述されているファイルを示すサブセクション 3R だけです。パス名は常にボールド体 (bold) で示されています。
機能説明	コマンドの機能とその動作について説明します。実行時の詳細を説明していますが、オプションの説明や使用例はここでは示され

	<p>ていません。対話形式のコマンド、サブコマンド、リクエスト、マクロ、関数などに関しては「使用法」で説明します。</p>
IOCTL	<p>セクション7だけに使用される項です。ioctl(2) システムコールへのパラメータは ioctl と呼ばれ、適切なパラメータを持つデバイスクラスのマニュアルページだけに記載されています。特定のデバイスに関する ioctl 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。デバイスの特定のクラスに関する ioctl は、mtio(7I) のように io で終わる名前が付いているデバイスクラスのマニュアルページに記載されています。</p>
オプション	<p>各オプションがどのように実行されるかを説明しています。「形式」で示されている順に記述されています。オプションの引数はこの項目で説明され、必要な場合はデフォルト値を示します。</p>
オペランド	<p>コマンドのオペランドを一覧表示し、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ています。</p>
出力	<p>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) を説明しています。</p>
戻り値	<p>値を返す関数の場合、その値を示し、値が返される時の条件を説明しています。関数が 0 や -1 のような一定の値だけを返す場合は、値と説明の形で示され、その他の場合は各関数の戻り値について簡単に説明しています。void として宣言された関数はこの項では扱いません。</p>
エラー	<p>失敗の場合、ほとんどの関数はその理由を示すエラーコードを errno 変数の中に設定します。この項ではエラーコードをアルファベット順に記述し、各エラーの原因となる条件について説明します。同じエラーの原因となる条件が複数ある場合は、エラーコードの下にそれぞれの条件を別々のパラグラフで説明しています。</p>
使用法	<p>この項では、使用する際の手がかりとなる説明が示されています。特定の決まりや機能、詳しい説明の必要なコマンドなどが示されています。組み込み機能については、以下の小項目で説明しています。</p>
	<p>コマンド 修飾子 変数 式 入力文法</p>
使用例	<p>コマンドや関数の使用例または使用方法を説明しています。できるだけ実際に入力するコマンド行とスクリーンに表示される内容を例にしています。例の中には必ず example% のプロンプトが出てきます。スーパーユーザーの場合は example# のプロンプトになります。例では、その説明、変数置換の方法、戻り値が示さ</p>

	れ、それらのほとんどが「形式」、「機能説明」、「オプション」、「使用法」の項からの実例となっています。
環境	コマンドや関数が影響を与える環境変数を記述し、その影響について簡単に説明しています。
終了ステータス	コマンドが呼び出しプログラムまたはシェルに返す値と、その状態を説明しています。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	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を示し、各ファイルについて簡単に説明しています。
属性	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、およびデバイスドライバの特性を一覧しています。詳細は <code>attributes(5)</code> を参照してください。
関連項目	関連するマニュアルページ、当社のマニュアル、および一般の出版物が示されています。
診断	エラーの発生状況と診断メッセージが示されています。メッセージはボールド体 (bold) で、変数はイタリック体 (<i>Italic</i>) または <日本語訳> で示されており、Cロケール時の表示形式です。
警告	作業に支障を与えるような現象について説明しています。診断メッセージではありません。
注意事項	それぞれの項に該当しない追加情報が示されています。マニュアルページの内容とは直接関係のない事柄も参照用に扱っています。ここでは重要な情報については説明していません。
使用上の留意点	すでに発見されているバグについて説明しています。可能な場合は対処法も示しています。

SunOS リファレンスマニュアル 1M: システム管理コマンド

Intro(1M)

名前	Intro, intro – 管理コマンドおよびアプリケーションプログラムの序章
機能説明	<p>本セクションでは、主にシステムの保守や管理に使用するコマンドを、アルファベット順に説明します。</p> <p>コマンドが仮想ファイルシステムのアーキテクチャに合わせて再構成されているため、同じ名前で始まる複数のマニュアルページが存在します。たとえば、mount の名前に関しては、mount(1M)、mount_cachefs(1M)、mount_hfsfs(1M)、mount_nfs(1M)、mount_tmpfs(1M)、mount_ufs(1M) のように 6 つのマニュアルページが存在します。このような場合、最初のマニュアルページにだけ、その総称コマンドの構文およびオプションが説明されています。つまり、これらのオプションは、すべてのファイルシステムのタイプに適用されるということです。以降のマニュアルページには、そのコマンドの機能のうちファイルシステムタイプに特有な部分が説明されています。このようなマニュアルページ名には、下線()とそのコマンドに関係するファイルシステムのタイプ名が伴われています。管理者は、このようなファイルシステムに特有な部分を直接呼び出してはなりません。総称コマンドは、すべてのファイルシステムに共通なインタフェースを提供します。ですから、ファイルシステムタイプに特有なマニュアルページは、個々のコマンドを説明していると考えべきではなく、コマンドのファイルシステムに特有な面を詳細に述べたものだと考えるべきです。</p>
コマンドの構文	<p>特に説明しないかぎり、本セクションで説明するコマンドは、以下の構文に従って、オプションやその他の引数を受け付けます。</p> <pre>name [option(s)] [cmdarg(s)]</pre> <p><i>name</i> 実行可能ファイルの名前です。</p> <p><i>option</i> -noargletter(s) または -argletter<>optarg。</p> <p> <>は、空白 (オプション)。</p> <p><i>noargletter</i> 引数が必要でないオプション 1 文字を表します。</p> <p><i>argletter</i> 引数が必要なオプション 1 文字を表します。</p> <p><i>optarg</i> <i>argletter</i> に必要な引数 (文字列) です。</p> <p><i>cmdarg</i> パス名 (または他のコマンドの引数)。 - だけを指定すると標準入力を表します。</p>
属性	このセクションにリストされた属性については attributes(5) のマニュアルページを参照してください。
関連項目	getopt(1), getopt(3C), attributes(5)
診断	終了時、すべてのコマンドは、正常に終了すると 0 を返します。ゼロでない値を返した場合、間違った引数を指定した、不良で受け入れることができないデータを指定した、その他、現在はうまく処理できないなどの障害を示します。このような値は、「終了コード」、「終了ステータス」、「リターンコード」などさまざまな呼ばれ方をします。そして、特別な使い方がある場合にかぎって説明されます。

注意事項 コマンド一覧	名前	説明
	Intro(1M)	すべてのコマンドにおいて標準の構文に準拠しているわけではありません。 管理コマンドおよびアプリケーションプログラムの序章
	accept(1M)	印刷要求の受付または拒否
	acct(1M)	アカウントリングおよびいろいろなアカウントコマンドの概要
	acctdisk(1m)	acct(1M) を参照
	acctdusg(1m)	acct(1M) を参照
	accton(1m)	acct(1M) を参照
	acctwtmp(1m)	acct(1M) を参照
	add_install_client(1m)	install_scripts(1M) を参照
	add_to_install_server(1m)	install_scripts(1M) を参照
	afbconfig(1M)	AFB グラフィックスアクセラレータの設定
	automount(1M)	自動マウントポイントのインストール
	boot(1M)	システムカーネルまたはスタンドアロンプログラムの起動
	catman(1M)	参照マニュアル用のフォーマット整形したファイルの作成
	cfgadm(1M)	構成の管理
	cfgadm_ac(1M)	EXX00 メモリーシステムの管理
	cfgadm_sysctrl(1M)	EXX00 システムボードの管理
	check(1m)	install_scripts(1M) を参照
	closewtmp(1m)	acct(1M) を参照
	cvcd(1M)	仮想コンソールデーモン
	dd(1M)	ファイルの変換とコピー
	df(1M)	使用可能なディスクブロックおよびファイル数の表示
	fdisk(1M)	固定ディスクパーティションテーブルの作成または変更
	ffbconfig(1M)	FFB グラフィックスアクセラレータの設定
	format(1M)	ディスクのパーティション分割および保守のためのユーティリティ
	fsck(1M)	ファイルシステムの検査および修復

Intro(1M)

fuser(1M)	ファイルまたはファイル構造を使用しているプロセスの特定
GFXconfig(1m)	pgxconfig(1M) を参照
halt(1M)	プロセッサの停止
init(1M)	プロセス制御の初期化
installer(1M)	Solaris Web Start インストールユーティリティ
install_scripts(1M)	Solaris ソフトウェアをインストールするためのスクリプト
kdmconfig(1M)	OpenWindows および国際化対応のキーボード、ディスプレイ、およびマウスオプションの設定または設定解除
lpmove(1M)	印刷要求の移動
lpsched(1M)	LP 印刷サービスの起動
lpshut(1M)	LP 印刷サービスの停止
lu(1M)	Live Upgrade 機能を使用するための FMLI ベースのインタフェース
luactivate(1M)	ブート環境のアクティブ化
lucancel(1M)	スケジュールされた Live Upgrade 機能でのコピー/作成操作の取り消し
lucompare(1M)	2つのブート環境の比較
lucrate(1M)	新しいブート環境の作成
lucurr(1M)	アクティブなブート環境名の表示
ludelete(1M)	ブート環境の削除
lufslis(1M)	ブート環境の構成の一覧表示
lumake(1M)	ブート環境の生成
lumount(1M)	ブート環境のすべてのファイルシステムのマウントまたはマウント解除
lurename(1M)	ブート環境名の変更
lustatus(1M)	ブート環境のステータスの表示
luumount(1m)	lumount(1M) を参照
luupgrade(1M)	ブート環境上のソフトウェアのインストール、アップグレード、およびその他の機能の実行

luxadm(1M)	Sun StorEdge A5000、RSM、SPARCstorage Array (SSA)サブシステムおよび各 Fiber Channel Arbitrated Loop (FC_AL) デバイスの管理プログラム
m64config(1M)	M64 グラフィックスアクセラレータの設定
mkfile(1M)	ファイルの作成
mkfs(1M)	ファイルシステムの構築
modify_install_server(1M)	install_scripts(1M) を参照
modinfo(1M)	ロードされているカーネルモジュールについての情報の表示
modload(1M)	カーネルモジュールのロード
modunload(1M)	モジュールのアンロード
mount(1M)	ファイルシステムとリモート資源のマウントおよびマウント解除
mountall(1M)	複数のファイルシステムのマウントおよびマウント解除
newfs(1M)	新しい UFS ファイルシステムの構築
patchadd(1M)	Solaris オペレーティング環境を実現するシステムへのパッチ適用
patchrm(1M)	システムからのパッチ削除とパッチ適用前のファイル復元
pgxconfig(1M)	PGX32 (Raptor GFX) グラフィックスアクセラレータの設定
pkgadd(1M)	システムへのソフトウェアパッケージの転送
pkgrm(1M)	システムからのパッケージの削除
pmconfig(1M)	電源管理システムの設定
powerd(1M)	電源管理デーモン
poweroff(1M)	halt(1M) を参照
prodreg(1M)	Solaris Product Registry の管理
prtconf(1M)	システム構成の出力
prtdiag(1M)	システムの診断情報の出力
raidctl(1M)	RAID ハードウェアユーティリティー
reboot(1M)	オペレーティングシステムの再起動
reject(1M)	accept(1M) を参照
restricted_shell(1M)	rsh(1M) を参照

Intro(1M)

<code>rm_install_client(1m)</code>	<code>install_scripts(1M)</code> を参照
<code>rsh(1M)</code>	制限付きシェルのコマンドインタプリタ
<code>scadm(1M)</code>	システムコントローラ (SC) の管理
<code>setup_install_server(1m)</code>	<code>install_scripts(1M)</code> を参照
<code>share(1M)</code>	ローカル資源をリモートシステムからマウントできるようにする
<code>shareall(1M)</code>	複数リソースの共有または共有解除
<code>showmount(1M)</code>	すべてのリモートマウントの表示
<code>shutdown(1M)</code>	システムの停止とシステム状態の変更
<code>su(1M)</code>	スーパーユーザーまたは別のユーザーに変更
<code>SUNWafb_config(1m)</code>	<code>afbconfig(1M)</code> を参照
<code>SUNWffb_config(1m)</code>	<code>ffbconfig(1M)</code> を参照
<code>SUNWm64_config(1m)</code>	<code>m64config(1M)</code> を参照
<code>sys-unconfig(1M)</code>	システム構成の解除
<code>telinit(1m)</code>	<code>init(1M)</code> を参照
<code>TSIgfxp_config(1m)</code>	<code>pgxconfig(1M)</code> を参照
<code>ttymon(1M)</code>	端末ポートのポートモニター
<code>umount(1m)</code>	<code>mount(1M)</code> を参照
<code>umountall(1m)</code>	<code>mountall(1M)</code> を参照
<code>unshare(1M)</code>	ローカルシステムをリモートシステムからマウント不可能にする設定
<code>unshareall(1m)</code>	<code>shareall(1M)</code> を参照
<code>utmp2wtmp(1m)</code>	<code>acct(1M)</code> を参照
<code>wall(1M)</code>	すべてのユーザーへの通知

名前	accept, reject – 印刷要求の受付または拒否
形式	accept <i>destination</i> ... reject [-r <i>reason</i>] <i>destination</i> ...
機能説明	<p>accept コマンドは、指定した宛先への印刷要求を待ち行列に加えることを許可します。</p> <p>reject コマンドは、指定した宛先への印刷要求を待ち行列に加えることができないようにします。</p> <p>宛先が印刷要求を受け付けたか、または拒否したかを確認するには、lpstat -a を実行してください。</p> <p>accept と request は印刷サーバー上で実行する必要があります。クライアントシステム上では意味をもちません。</p>
オプション	<p>reject コマンドには、次のオプションを指定できます。</p> <p>-r <i>reason</i> <i>destination</i> への印刷要求を抑止する理由を文字列として記述します。<i>reason</i> に空白が含まれる場合は引用符で囲ってください。<i>reason</i> は、プリンタの状況を調べる lpstat -a コマンドの出力中に表示されます。<i>reason</i> のデフォルト値は、既存の宛先に関しては unknown reason、システムに追加されたばかりでまだ印刷要求を受け付けたことのない宛先に関しては new printer となります。</p>
オペランド	<p>次のオペランドを指定できます。</p> <p><i>destination</i> 印刷要求を受け付ける、または拒否する宛先名。宛先にはプリンタ名やプリンタクラスを指定します (lpadmin(1M) 参照)。<i>destination</i> は名前を使用して指定します。名前の命名規約については printers.conf(4) のマニュアルページを参照してください。</p>
終了ステータス	<p>以下の終了ステータスが返されます。</p> <p>0 正常終了</p> <p>0 以外 エラーが発生した</p>
ファイル	/var/spool/lp/* LP 印刷待ち行列
属性	次の属性については attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWpcu
CSI	対応済み(「注意事項」参照)

accept(1M)

関連項目	enable(1), lp(1), lpstat(1), lpadmin(1M), lpsched(1M), printers.conf(4), attributes(5)
注意事項	<p>accept と reject は印刷サーバーのスプール用システムの待ち行列に対してのみ有効です。したがって、クライアントシステムから出された要求については、印刷サーバーのスプール用システムが取り消したり受け取ったりするまでは、クライアントシステムの印刷待ち行列に入れられたままの状態になります。</p> <p>accept は <i>destinations</i> 名を除いて CSI 対応が可能です。</p>

名前	acct, acctdisk, acctdusg, accton, acctwtmp, closewtmp, utmp2wtmp – アカウンティングおよびいろいろなアカウントコマンドの概要
形式	<pre> /usr/lib/acct/acctdisk /usr/lib/acct/acctdusg [-u filename] [-p filename] /usr/lib/acct/accton [filename] /usr/lib/acct/acctwtmp reason filename /usr/lib/acct/closewtmp /usr/lib/acct/utmp2wtmp </pre>
機能説明	<p>アカウンティングソフトウェアは、アカウンティングシステムを構築するためのツール群です (C 言語のプログラムおよびシェルプロシージャから構成される)。acctsh(1M) では、C 言語のプログラムの一番上に組むシェルプロシージャ群を説明します。</p> <p>接続時間のアカウンティングは記録を /var/adm/wtmpx (utmpx(4) を参照) に書き込むプログラム群により管理されます。acctcon(1M) では、このファイルをセッションおよび課金記録に変換するプログラムを説明します。また、acctmerg(1M) では、この課金記録について集約します。</p> <p>プロセスアカウンティングは、システムカーネルによって行われます。プロセスを終了すると、1つのプロセスにつき1つのレコードが、あるファイル (通常は、/var/adm/pacct) に書き込まれます。acctprc(1M) では、課金のためにこのデータを集約します。コマンドの使用状況を集約するには、acctcms(1M) を使用します。acctcom(1) を用いて、現在のプロセスデータを調査することができます。</p> <p>acctmerg (acct(3HEAD) の tacct フォーマットを参照) を使うと、プロセスのアカウンティングレコードおよび接続時間のアカウンティングレコード (あるいは acct(3HEAD) で説明する tacct フォーマットのアカウンティングレコードのいずれか) をアカウンティングレコードの合計にマージしたり、集約したりすることができます。どのアカウンティングレコードにも、あるいはすべてのアカウンティングレコードにでも prtacct (acctsh(1M) を参照) をフォーマットに使用します。</p> <p>acctdisk は、ユーザー ID、ログイン名およびディスクブロック数がある行を読み取り、他のアカウンティングレコードとマージすることができるアカウンティングレコードの合計に、それらの情報を変換します。入力ファイルが壊れているか、または正しくフォーマットされていない場合は、acctdisk はエラーを返します。</p> <p>acctdusg は、その標準入力を (通常は、find / -print から) 読み取り、ログインによるディスクの資源消費 (間接ブロックを含む) を計算します。</p> <p>accton では、引数がない場合、プロセスアカウンティングをオフにします。filename を指定する場合は、そのファイルは、カーネルがプロセスのアカウンティングレコード (acct(2) および acct(3HEAD) を参照) を追加するための、既存のファイル名である必要があります。</p>

acct(1M)

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

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

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

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

オプション 以下のオプションを指定できます。

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

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

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

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

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

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

属性 次の属性については attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWaccu

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

『Solaris のシステム管理 (基本編)』

acctdisk(1M)

名前	acct, acctdisk, acctdusg, accton, acctwtmp, closewtmp, utmp2wtmp – アカウンティングおよびいろいろなアカウントコマンドの概要
形式	<pre>/usr/lib/acct/acctdisk /usr/lib/acct/acctdusg [-u filename] [-p filename] /usr/lib/acct/accton [filename] /usr/lib/acct/acctwtmp reason filename /usr/lib/acct/closewtmp /usr/lib/acct/utmp2wtmp</pre>
機能説明	<p>アカウンティングソフトウェアは、アカウンティングシステムを構築するためのツール群です (C 言語のプログラムおよびシェルプロシージャから構成される)。acctsh(1M) では、C 言語のプログラムの一番上に組むシェルプロシージャ群を説明します。</p> <p>接続時間のアカウンティングは記録を /var/adm/wtmpx (utmpx(4) を参照) に書き込むプログラム群により管理されます。acctcon(1M) では、このファイルをセッションおよび課金記録に変換するプログラムを説明します。また、acctmerg(1M) では、この課金記録について集約します。</p> <p>プロセスアカウンティングは、システムカーネルによって行われます。プロセスを終了すると、1つのプロセスにつき1つのレコードが、あるファイル (通常は、/var/adm/pacct) に書き込まれます。acctprc(1M) では、課金のためにこのデータを集約します。コマンドの使用状況を集約するには、acctcms(1M) を使用します。acctcom(1) を用いて、現在のプロセスデータを調査することができます。</p> <p>acctmerg (acct(3HEAD) の tacct フォーマットを参照) を使うと、プロセスのアカウンティングレコードおよび接続時間のアカウンティングレコード (あるいは acct(3HEAD) で説明する tacct フォーマットのアカウンティングレコードのいずれか) をアカウンティングレコードの合計にマージしたり、集約したりすることができます。どのアカウンティングレコードにも、あるいはすべてのアカウンティングレコードにでも prtacct (acctsh(1M) を参照) をフォーマットに使用します。</p> <p>acctdisk は、ユーザー ID、ログイン名およびディスクブロック数がある行を読み取り、他のアカウンティングレコードとマージすることができるアカウンティングレコードの合計に、それらの情報を変換します。入力ファイルが壊れているか、または正しくフォーマットされていない場合は、acctdisk はエラーを返します。</p> <p>acctdusg は、その標準入力を (通常は、find / -print から) 読み取り、ログインによるディスクの資源消費 (間接ブロックを含む) を計算します。</p> <p>accton では、引数がない場合、プロセスアカウンティングをオフにします。filename を指定する場合は、そのファイルは、カーネルがプロセスのアカウンティングレコード (acct(2) および acct(3HEAD) を参照) を追加するための、既存のファイル名である必要があります。</p>

acctdisk(1M)

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

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

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

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

オプション 以下のオプションを指定できます。

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

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

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

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

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

acctdisk(1M)

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

属性 次の属性については attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWaccu

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

『Solaris のシステム管理 (基本編)』

名前	acct, acctdisk, acctdusg, accton, acctwtmp, closewtmp, utmp2wtmp – アカウンティングおよびいろいろなアカウントコマンドの概要
形式	<pre> /usr/lib/acct/acctdisk /usr/lib/acct/acctdusg [-u filename] [-p filename] /usr/lib/acct/accton [filename] /usr/lib/acct/acctwtmp reason filename /usr/lib/acct/closewtmp /usr/lib/acct/utmp2wtmp </pre>
機能説明	<p>アカウンティングソフトウェアは、アカウンティングシステムを構築するためのツール群です (C 言語のプログラムおよびシェルプロシージャから構成される)。acctsh(1M) では、C 言語のプログラムの一番上に組むシェルプロシージャ群を説明します。</p> <p>接続時間のアカウンティングは記録を /var/adm/wtmpx (utmpx(4) を参照) に書き込むプログラム群により管理されます。acctcon(1M) では、このファイルセッションおよび課金記録に変換するプログラムを説明します。また、acctmerg(1M) では、この課金記録について集約します。</p> <p>プロセスアカウンティングは、システムカーネルによって行われます。プロセスを終了すると、1つのプロセスにつき1つのレコードが、あるファイル (通常は、/var/adm/pacct) に書き込まれます。acctprc(1M) では、課金のためにこのデータを集約します。コマンドの使用状況を集約するには、acctcms(1M) を使用します。acctcom(1) を用いて、現在のプロセスデータを調査することができます。</p> <p>acctmerg (acct(3HEAD) の tacct フォーマットを参照) を使うと、プロセスのアカウンティングレコードおよび接続時間のアカウンティングレコード (あるいは acct(3HEAD) で説明する tacct フォーマットのアカウンティングレコードのいずれか) をアカウンティングレコードの合計にマージしたり、集約したりすることができます。どのアカウンティングレコードにも、あるいはすべてのアカウンティングレコードにでも prtacct (acctsh(1M) を参照) をフォーマットに使用します。</p> <p>acctdisk は、ユーザー ID、ログイン名およびディスクブロック数がある行を読み取り、他のアカウンティングレコードとマージすることができるアカウンティングレコードの合計に、それらの情報を変換します。入力ファイルが壊れているか、または正しくフォーマットされていない場合は、acctdisk はエラーを返します。</p> <p>acctdusg は、その標準入力を (通常は、find / -print から) 読み取り、ログインによるディスクの資源消費 (間接ブロックを含む) を計算します。</p> <p>accton では、引数がない場合、プロセスアカウンティングをオフにします。filename を指定する場合は、そのファイルは、カーネルがプロセスのアカウンティングレコード (acct(2) および acct(3HEAD) を参照) を追加するための、既存のファイル名である必要があります。</p>

acctdusg(1M)

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

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

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

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

オプション 以下のオプションを指定できます。

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

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

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

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

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

acctdusg(1M)

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

属性 次の属性については attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWaccu

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

『Solaris のシステム管理 (基本編)』

accton(1M)

名前	acct, acctdisk, acctdusg, accton, acctwtmp, closewtmp, utmp2wtmp – アカウンティングおよびいろいろなアカウントコマンドの概要
形式	<pre>/usr/lib/acct/acctdisk /usr/lib/acct/acctdusg [-u filename] [-p filename] /usr/lib/acct/accton [filename] /usr/lib/acct/acctwtmp reason filename /usr/lib/acct/closewtmp /usr/lib/acct/utmp2wtmp</pre>
機能説明	<p>アカウンティングソフトウェアは、アカウンティングシステムを構築するためのツール群です (C 言語のプログラムおよびシェルプロシージャから構成される)。acctsh(1M) では、C 言語のプログラムの一番上に組むシェルプロシージャ群を説明します。</p> <p>接続時間のアカウンティングは記録を /var/adm/wtmpx (utmpx(4) を参照) に書き込むプログラム群により管理されます。acctcon(1M) では、このファイルをセッションおよび課金記録に変換するプログラムを説明します。また、acctmerg(1M) では、この課金記録について集約します。</p> <p>プロセスアカウンティングは、システムカーネルによって行われます。プロセスを終了すると、1つのプロセスにつき1つのレコードが、あるファイル (通常は、/var/adm/pacct) に書き込まれます。acctprc(1M) では、課金のためにこのデータを集約します。コマンドの使用状況を集約するには、acctcms(1M) を使用します。acctcom(1) を用いて、現在のプロセスデータを調査することができます。</p> <p>acctmerg (acct(3HEAD) の tacct フォーマットを参照) を使うと、プロセスのアカウンティングレコードおよび接続時間のアカウンティングレコード (あるいは acct(3HEAD) で説明する tacct フォーマットのアカウンティングレコードのいずれか) をアカウンティングレコードの合計にマージしたり、集約したりすることができます。どのアカウンティングレコードにも、あるいはすべてのアカウンティングレコードにでも prtacct (acctsh(1M) を参照) をフォーマットに使用します。</p> <p>acctdisk は、ユーザー ID、ログイン名およびディスクブロック数がある行を読み取り、他のアカウンティングレコードとマージすることができるアカウンティングレコードの合計に、それらの情報を変換します。入力ファイルが壊れているか、または正しくフォーマットされていない場合は、acctdisk はエラーを返します。</p> <p>acctdusg は、その標準入力を (通常は、find / -print から) 読み取り、ログインによるディスクの資源消費 (間接ブロックを含む) を計算します。</p> <p>accton では、引数がない場合、プロセスアカウンティングをオフにします。filename を指定する場合は、そのファイルは、カーネルがプロセスのアカウンティングレコード (acct(2) および acct(3HEAD) を参照) を追加するための、既存のファイル名である必要があります。</p>

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

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

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

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

オプション 以下のオプションを指定できます。

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

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

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

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

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

accton(1M)

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

属性 次の属性については attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWaccu

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

『Solaris のシステム管理 (基本編)』

名前	acct, acctdisk, acctdusg, accton, acctwtmp, closewtmp, utmp2wtmp – アカウンティングおよびいろいろなアカウントコマンドの概要
形式	<pre> /usr/lib/acct/acctdisk /usr/lib/acct/acctdusg [-u filename] [-p filename] /usr/lib/acct/accton [filename] /usr/lib/acct/acctwtmp reason filename /usr/lib/acct/closewtmp /usr/lib/acct/utmp2wtmp </pre>
機能説明	<p>アカウンティングソフトウェアは、アカウンティングシステムを構築するためのツール群です (C 言語のプログラムおよびシェルプロシージャから構成される)。acctsh(1M) では、C 言語のプログラムの一番上に組むシェルプロシージャ群を説明します。</p> <p>接続時間のアカウンティングは記録を /var/adm/wtmpx (utmpx(4) を参照) に書き込むプログラム群により管理されます。acctcon(1M) では、このファイルセッションおよび課金記録に変換するプログラムを説明します。また、acctmerg(1M) では、この課金記録について集約します。</p> <p>プロセスアカウンティングは、システムカーネルによって行われます。プロセスを終了すると、1つのプロセスにつき1つのレコードが、あるファイル (通常は、/var/adm/pacct) に書き込まれます。acctprc(1M) では、課金のためにこのデータを集約します。コマンドの使用状況を集約するには、acctcms(1M) を使用します。acctcom(1) を用いて、現在のプロセスデータを調査することができます。</p> <p>acctmerg (acct(3HEAD) の tacct フォーマットを参照) を使うと、プロセスのアカウンティングレコードおよび接続時間のアカウンティングレコード (あるいは acct(3HEAD) で説明する tacct フォーマットのアカウンティングレコードのいずれか) をアカウンティングレコードの合計にマージしたり、集約したりすることができます。どのアカウンティングレコードにも、あるいはすべてのアカウンティングレコードにでも prtacct (acctsh(1M) を参照) をフォーマットに使用します。</p> <p>acctdisk は、ユーザー ID、ログイン名およびディスクブロック数がある行を読み取り、他のアカウンティングレコードとマージすることができるアカウンティングレコードの合計に、それらの情報を変換します。入力ファイルが壊れているか、または正しくフォーマットされていない場合は、acctdisk はエラーを返します。</p> <p>acctdusg は、その標準入力を (通常は、find / -print から) 読み取り、ログインによるディスクの資源消費 (間接ブロックを含む) を計算します。</p> <p>accton では、引数がない場合、プロセスアカウンティングをオフにします。filename を指定する場合は、そのファイルは、カーネルがプロセスのアカウンティングレコード (acct(2) および acct(3HEAD) を参照) を追加するための、既存のファイル名である必要があります。</p>

acctwtmp(1M)

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

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

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

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

オプション 以下のオプションを指定できます。

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

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

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

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

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

acctwtmp(1M)

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

属性 次の属性については attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWaccu

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

『Solaris のシステム管理 (基本編)』

add_install_client(1M)

名前	install_scripts, add_install_client, add_to_install_server, modify_install_server, rm_install_client, setup_install_server, check – Solaris ソフトウェアをインストールするためのスクリプト
形式	<pre>cdrom-mnt-pt/Solaris_XX/Tools/add_install_client [-i IP_address] [-e Ethernet_address] [-s server_name : path] [-c server_name : path] [-n [server] : name_service [(netmask)]] [-p server_name : path] [-f boot_file_name] host_name platform_group cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server_name:path] [-c server_name:path] [-p server_name:path] [-t install_boot_image_path] [-f boot_file_name] platform_name platform_group cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server_name:path] [-c server_name:path] [-p server_name:path] [-t install_boot_image_path] [-f boot_file_name] -e Ethernet_address [-b property=value] platform_group cdrom-mnt-pt/Solaris_XX/Tools/add_to_install_server [-s] [-p product_image_path] install_server_path cdrom-mnt-pt/Solaris_XX/Tools/jumpstart_sample/check [-p install_dir_path] [-r rulesfile] cdrom-mnt-pt/modify_install_server [-p] install_dir_path installer_miniroot_path cdrom-mnt-pt/Solaris_XX/Tools/rm_install_client host_name cdrom-mnt-pt/Solaris_XX/Tools/setup_install_server [-b] [-t install_boot_image_path] [-w wanboot_image_path] install_dir_path</pre>
機能説明	<p>これらのコマンドは、Solaris Software and Solaris Installer CD のスライス 0 に置かれています。Solaris CD をすでにローカルディスクにコピーしている場合、<code>cdrom_mnt_pt</code> はコピーされている Solaris CD へのパスになります。これらのコマンドは、さまざまなインストール作業に使用できます。</p> <p>Solaris_XX の XX は、使用している Solaris リリースのバージョン番号です。</p> <p><code>add_install_client</code> コマンドには 3 種類の形式があります。「形式」の項を参照してください。</p> <p>ネットワークを使用してインストールするクライアントを追加する場合は、次の形式の <code>add_install_client</code> を使用します (これらのコマンドは <code>bootparams(4)</code> ファイルを更新します)。<code>add_install_client</code> コマンドは、インストールサーバーの Solaris インストールイメージ (マウントした CD またはディスクにコピーした Solaris CD) あるいはブートサーバーの起動ディレクトリ (ブートサーバーが必要な場合) から実行する必要があります。Solaris インストールイメージまたはブート専用ディレクトリの Solaris リリースは、クライアントにインストールする予定の Solaris リリースと同じでなければなりません。</p> <pre>cdrom-mnt-pt/Solaris_XX/Tools/add_install_client [-i IP_address] [-e Ethernet_address] [-s server_name : path] [-c server_name : path]</pre>

add_install_client(1M)

```
[ -n [server] ] : name_service [ ( netmask) ] [ -p server_name : path ]
host_name platform_group
```

プラットフォームグループ内のあるプラットフォームのインスタンスをインストールサーバーに追加する場合は、次の形式の `add_install_client` コマンドを使用します。このグループは DHCP を使用して起動および構成されます。スクリプトは、必要な構成作業をサーバー上で実行し、ユーザーがそのグループのために DHCP サーバーに追加しなければならないデータを出力します。

```
cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server:path]
[-c server:path] [-p server:path] [-t install boot image path]
[-f boot file name] platform_name platform_group
```

インストールサーバーに単一のクライアントを追加する場合は、次の形式の `add_install_client` コマンドを使用します。このクライアントは DHCP を使用して起動および構成されます。スクリプトは、必要な構成作業をサーバー上で実行し、ユーザーがそのクライアントのために DHCP サーバーに追加しなければならないデータを出力します。すでに使用しているものにも、上記のように `-f` フラグを追加する必要があります。`-f` を使用すると、ユーザーは所定のクライアントに使用する起動ファイルの名前を指定できます。

```
cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d
[-s server_name:path] [-c server_name:path] [-p server_name:path]
[-t install_boot_image_path] [-f boot_file_name] -e Ethernet_address
platform_group
```

注 - x86 アーキテクチャの Pre-boot eXecution Environment (PXE) クライアントを登録する場合は、必ず `-d` オプションを使用してください。x86 PXE クライアントは構成に DHCP を使用します。

他の Solaris CD とネットワークインストールサーバー上の既存のイメージをマージする場合は、`add_to_install_server` を使用します。マージ可能な各 CD (現時点では OS CD 2 および Language CD) には専用の `add_to_install_server` スクリプトが含まれています。配布された CD に含まれているもの以外の `add_to_install_server` スクリプトを使用しないでください。

`rules` ファイル (カスタム JumpStart インストールを使用する場合に限り必要) のルールを検証する場合は、`check` を使用します。

既存のネットワークインストールサーバーのミニルートを Solaris Installation CD のミニルートに置き換える場合は、`modify_install_server` を使用します。ネットワークインストールサーバーのインストール時のユーザーインターフェースが、Solaris Installation CD の Web Start ユーザーインターフェースに変更されます。

`modify_install_server` コマンドを使用する場合は、既存のインストールイメージ (`setup_install_server` を使用して作成) があらかじめ存在していなければなりません。

add_install_client(1M)

ネットワークインストール用のクライアントを削除する場合は、`rm_install_client` を使用します (このコマンドは `bootparams(4)` ファイルを更新します)。

Solaris CD を (インストールサーバーを設定するために) ディスクにコピーする、(WANboot インストールサーバーを設定するために) WANboot ミニルートイメージを構築する、または Solaris CD のブートソフトウェアだけを (ブートサーバーを設定するために) ディスクにコピーする場合は、`setup_install_server` を使用します。ネットワーク経由でクライアントをインストールするには、インストールサーバーが必要です。インストールサーバーとインストールするクライアントが異なるサブネットに属している場合、ネットワークインストールを行うには、ブートサーバーも必要です (ブートサーバーはクライアントのサブネットに配置する必要があります)。

オプション `add_install_client` には、次のオプションを指定できます。

`-b property=value`

ブートサーバーの TFTP ディレクトリ (デフォルトは `/tftpboot`) にあるクライアント特有の `bootenv.rc` ファイルにプロパティ値を設定します。クライアントに特有のブートプロパティを設定する場合に、このオプションを使用します。`bootpath` プロパティを使用すると、デバイス構成補助を使用して手動で選択しなくても、ブートデバイスを自動的に選択できます。`boot-args` プロパティを使用すると、JumpStart インストールを自動的に選択できます。`eeeprom(1M)` を参照してください。

このオプションは x86 クライアントにのみ使用できます。このオプションを使用するときには、必ず `-d` オプションと `-e` を同時に指定してください。

`-c server_name:path`

このオプションが必要なのは、カスタム JumpStart インストール用の JumpStart ディレクトリを指定する場合だけです。`server_name` は、JumpStart ディレクトリが置かれているサーバーのホスト名です。`path` は JumpStart ディレクトリの絶対パスです。

`-d`

DHCP クライアントを指定します。

`-e Ethernet_address`

インストールするシステムの Ethernet アドレスを指定します。

`-f`

インストールするクライアントの `boot_file_name` を指定します。

`-i IP_address`

インストールするクライアントの IP アドレスを指定します。

`-n [server]: name_service[(netmask)]`

このオプションでは、システム構成時に使用するネームサービスを指定します。`bootparams(4)` ファイル内の `ns` キーワードを設定します。

name_service

有効なエントリは `nis`、`nisplus`、および `none` です。

netmask

数字を4つずつピリオドで区切って並べたもので、IPアドレスのどの部分がネットワーク部分で、どれがホスト部分かを指定します。

server

サーバーの名前または指定したネームサービスのIPアドレス。指定したサーバーが異なるサブネット上にある場合は、クライアントからサーバーにアクセスするために、*netmask*が必要なことがあります。

-p *server_name*: *path*

このオプションは、システム情報またはネットワーク情報を前もって設定するために必要な、ユーザーが定義した *sysidcfg* ファイルの位置です。*server_name* は有効なホスト名またはIPアドレスです。*path* はJumpStart ディレクトリの絶対パスです。

-s *server_name*:*path*

このオプションが必要なのは、ブートサーバーから *add_install_client* を使用する場合だけです。サーバー名およびこのインストールで使用する Solaris インストールイメージの絶対パスを指定します。*path* はマウントされている Solaris CD のパスまたは Solaris CD のコピーが置かれているディレクトリのパスです。

add_to_install_server コマンドには、次のオプションを指定できます。

-p

コピーする CD (補助的な製品が収められている) の位置を指定します。

-s

ユーザーが、インストールする必要がある製品だけをリストから選択できるようにします。

check コマンドには、次のオプションを指定できます。

-p *install_dir_path*

使用中のシステムの *check* スクリプトではなく、指定された Solaris インストールイメージの *check* スクリプトを使用することによって、ルールファイルを検査します。*install_dir_path* は、ローカルディスク上またはマウントした Solaris CD 上の Solaris インストールイメージのパスです。

旧バージョンの Solaris が稼働しているシステムでは、このオプションを使用して、最新バージョンの *check* を実行できます。

-r *rulesfile*

rules 以外の名前のルールファイルを指定します。このオプションを使用すると、ルールの妥当性を検証してから、ルールファイルに組み込むことができます。*check* はルールが有効であるかどうかを報告するだけで、カスタム JumpStart インストールに必要な *rules.ok* ファイルは作成しません。

modify_install_server コマンドには、次のオプションを指定できます。

-p

このオプションは、*install_dir_path/Solaris_XX/Tools/Boot.orig* にすでに存在しているイメージのミニルートを保持します。

add_install_client(1M)

setup_install_server コマンドには、次のオプションを指定できます。

- b
サーバーの設定を、ブートサーバー専用にします。
- t
代替ミニルート指定できるようにします。
- w
WANboot ミニルートイメージを構築します。

オペランド add_install_client コマンドには、次のオペランドを指定できます。

host_name

インストールするクライアントの名前です。

platform_group

特定のソフトウェアを配布するために、ベンダーが定義したハードウェアプラットフォームグループ。以下に有効なプラットフォームグループの例を示します。

システム	プラットフォームグループ
x86	i86pc
Sun Fire 4800	sun4c

システムのプラットフォームグループを調べるには、(-m オプションを指定して) `uname(1)` コマンドを使用します。

platform_name

システムのプラットフォーム名を調べるには、-i オプションを指定して `uname(1)` コマンドを使用します。

次に、`uname command` コマンドを使用して Ultra 1 のシステムプラットフォーム名を調べる例を示します。

```
uname -i
システムは、次のように応答します。
```

```
SUNW,Ultra-1
システムのプラットフォーム名は SUNW,Ultra1 です。
```

次のコマンドは、Ultra 1 用の `add_install_client` を呼び出します。

```
add_install_client -d SUNW,Ultra-1 sun4u
```

IA32 プラットフォームでは、プラットフォーム名はつねに `SUNW.i86pc` です。

次のコマンドは、IA32 プラットフォーム用の `add_install_client` を呼び出します。

```
add_install_client -d SUNW.i86pc i86pc
```

`rm_install_client` コマンドには、次のオペランドを指定できます。

host_name

削除するクライアントの名前です。

setup_install_server コマンドには、次のオペランドを指定できます。

install_dir_path

Solaris ソフトウェアのコピー先となるディレクトリの絶対パスです。このディレクトリは空でなければなりません。

wanboot_image_path

WANboot ミニルートイメージを含むファイルが作成されるディレクトリの絶対パスです。

使用例 例 1 add_install_client の使用

次の add_install_client コマンドは、マウント済みの Solaris CD からネットワークインストール用のクライアントを Solaris 8 のインストールサーバーに追加します。

```
example# cd /cdrom/cdrom0/s0/Solaris_9/Tools
example# ./add_install_client system_2/sun4u
```

例 2 add_install_client の使用

次の add_install_client コマンドは、ネットワークインストール用のクライアントをインストールサーバーに追加します。カスタム JumpStart インストールを実行するためのルールファイルおよびプロファイルファイルが置かれている JumpStart のサーバーとディレクトリのパスを -c オプションで指定します。Solaris CD は Solaris 8 上の /export/install ディレクトリにコピーされています。

```
example# cd /export/install/Solaris_8/Tools
example# /add_install_client -c install_server:/jumpstart system_1 i86pc
example# ./add_install_client -c install_server:/jumpstart system_2 i86pc
```

例 3 add_install_client の使用

次の add_install_client コマンドは、起動ファイル sun4u.solaris8 を使用する特定の sun4u プラットフォームマシン (8:0:20:99:88:77) に対するサポートを追加します。

```
example# add_install_client -d -f sun4u.solaris8 -e 8:0:20:99:88:77 sun4u
```

例 4 add_install_client の使用

次の add_install_client コマンドは、PXE 規格を使用してネットワークから起動する x86 クライアントを追加します。

```
example# add_install_client -d -s svrname:/mnt/export/root SUNW.i86pc i86p
```

例 5 add_to_install_server の使用

次の add_to_install_server コマンドは、CD 上のすべての製品ディレクトリにあるパッケージを Solaris 8 上の既存のインストールサーバーにコピーします。

add_install_client(1M)

例 5 add_to_install_server の使用 (続き)

```
example# cd /cdrom/cdrom0/s0
example# ./add_to_install_server /export/Solaris_8
```

例 6 check の使用

次の check コマンドは、カスタム JumpStart インストールに使用するルールファイルの構文を検査します。

```
example# cd jumpstart_dir_path
example# ./check -p /cdrom/cdrom0/s0
```

例 7 modify_install_server の使用

次の modify_install_server コマンドは、上記の setup_install_server を使用して作成されたミニルートを Boot.orig に移し、Solaris Installer CD 上のミニルートに置き換えます。

```
example# cd /cdrom/cdrom0/s0
example# ./modify_install_server -p /export/install /cdrom/cdrom0/s1
```

例 8 modify_install_server の使用

次の modify_install_server コマンドは、上記の setup_install_server を使用して作成されたミニルートを Solaris Installer CD 上のミニルートに置き換えます。

```
example# cd /cdrom/cdrom0/s0
example# ./modify_install_server /export/install /cdrom/cdrom0/s1
```

例 9 rm_install_client の使用

次の rm_install_client コマンドは、Solaris 8 上のネットワークインストール用クライアントを削除します。

```
example# cd /export/install/Solaris_8/Tools
example# ./rm_install_client holmes
example# ./rm_install_client watson
```

例 10 setup_install_server の使用

次の setup_install_server コマンドは、マウント済みの Solaris CD を Solaris 8 のローカルディスク上にある /export/install というディレクトリにコピーします。

```
example# cd /cdrom/cdrom0/s0/Solaris_8/Tools
example# ./setup_install_server /export/install
```

例 11 setup_install_server の使用

次の setup_install_server コマンドは、マウント済みの Solaris CD のブートソフトウェアを、Solaris 8 上のサブネット用ブートサーバーになるシステムの /boot_dir というディレクトリにコピーします。

例 11 setup_install_server の使用 (続き)

```
example# cd /cdrom/cdrom0/s0/Solaris_8/Tools
example# ./setup_install_server -b /boot_dir
```

例 12 setup_install_server の使用

setup_install_server はデフォルトで、マウント済みの Solaris 配布ディスク上の Solaris ../Tools/Boot にあるインストールブートディレクトリを検索します。

以前に ./setup_install_server -b /boot_dir コマンドを使用して、ネットワークブートサーバー上に作成したディレクトリと別のディレクトリが必要な場合は、-t オプションを使用できます。

```
example# cd /cdrom/cdrom0/s0/Solairs_9/Tools
example# ./setup_install_server -t /boot_dir /export/install
```

例 13 setup_install_server と WANboot オプションの使用

次の setup_install_server コマンドは WANboot ミニルートファイルシステムのイメージを作成し、それを /wanboot_dir/miniroot に保存します。

```
example# cd /cdrom/cdrom0/s0/Solairs_9/Tools
example# ./setup_install_server -w /wanboot_dir /export/install
```

例 14 x86: ネットワークインストール時に使用するシリアルコンソールの指定 (CD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するシリアルコンソールを指定する方法を示します。この例では、インストールクライアントを次の方法で設定します。

- -d オプションは、クライアントが DHCP を使用してインストールパラメータを設定することを示します。
- -e オプションは、Ethernet アドレスが 00:07:e9:04:4a:bf であるクライアントでのみインストールが行われることを示します。
- 1 つめと 2 つめの -b オプションは、インストールプログラムが入力デバイスおよび出力デバイスとしてシリアルポート ttya を使用することを示します。

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
i86pc
```

-b オプションで使用できるブートプロパティの変数とその値については、eeprom(1M) を参照してください。

例 15 ネットワークインストール時に使用するブートデバイスの指定 (CD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するブートデバイスを指定する方法を示します。インストールクライアントを設定するときにブートデバイスを指定する場合は、インストール時にデバイス構成補助によって出されるこの情報は表示されません。

add_install_client(1M)

例 15 ネットワークインストール時に使用するブートデバイスの指定 (CD) (続き)

この例では、次の方法でインストールクライアントを設定します。

- -d オプションは、クライアントが DHCP を使用してインストールパラメータを設定することを示します。
- -e オプションは、Ethernet アドレスが 00:07:e9:04:4a:bf であるクライアントでのみインストールが行われることを示します。
- 1 つめと 2 つめの -b オプションは、インストールプログラムが入力デバイスおよび出力デバイスとしてシリアルポート ttya を使用することを示します。
- 3 つめの -b オプションは、インストールプログラムがインストール時に特定のブートデバイスを使用することを示します。
- ブートデバイスのパスは、使用しているハードウェアによって異なります。
- i86pc というプラットフォーム名は、クライアントが x86 ベースのシステムであることを示します。

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
-b "bootpath=/pci@0,0/pci108e,16a8@8" i86pc
```

-b オプションで使用できるブートプロパティの変数とその値について詳細は、eeprom(1M) を参照してください。

終了ステータス 次の終了ステータスが返されます。

```
0      正常終了
1      エラーが発生した
```

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	Solaris CD

関連項目 uname(1), bootparams(4), attributes(5)

『Solaris 9 インストールガイド』

add_to_install_server(1M)

名前 | install_scripts, add_install_client, add_to_install_server, modify_install_server, rm_install_client, setup_install_server, check – Solaris ソフトウェアをインストールするためのスクリプト

形式 | `cdrom-mnt-pt/Solaris_XX/Tools/add_install_client [-i IP_address] [-e Ethernet_address] [-s server_name : path] [-c server_name : path] [-n [server] : name_service [(netmask)]] [-p server_name : path] [-f boot_file_name] host_name platform_group`
`cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server_name:path] [-c server_name:path] [-p server_name:path] [-t install_boot_image_path] [-f boot_file_name] platform_name platform_group`
`cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server_name:path] [-c server_name:path] [-p server_name:path] [-t install_boot_image_path] [-f boot_file_name] -e Ethernet_address [-b property=value] platform_group`
`cdrom-mnt-pt/Solaris_XX/Tools/add_to_install_server [-s] [-p product_image_path] install_server_path`
`cdrom-mnt-pt/Solaris_XX/Tools/jumpstart_sample/check [-p install_dir_path] [-r rulesfile]`
`cdrom-mnt-pt/modify_install_server [-p] install_dir_path installer_miniroot_path`
`cdrom-mnt-pt/Solaris_XX/Tools/rm_install_client host_name`
`cdrom-mnt-pt/Solaris_XX/Tools/setup_install_server [-b] [-t install_boot_image_path] [-w wanboot_image_path] install_dir_path`

機能説明 | これらのコマンドは、Solaris Software and Solaris Installer CD のスライス 0 に置かれています。Solaris CD をすでにローカルディスクにコピーしている場合、`cdrom_mnt_pt` はコピーされている Solaris CD へのパスになります。これらのコマンドは、さまざまなインストール作業に使用できます。

Solaris_XX の XX は、使用している Solaris リリースのバージョン番号です。

`add_install_client` コマンドには 3 種類の形式があります。「形式」の項を参照してください。

ネットワークを使用してインストールするクライアントを追加する場合は、次の形式の `add_install_client` を使用します (これらのコマンドは `bootparams(4)` ファイルを更新します)。`add_install_client` コマンドは、インストールサーバーの Solaris インストールイメージ (マウントした CD またはディスクにコピーした Solaris CD) あるいはブートサーバーの起動ディレクトリ (ブートサーバーが必要な場合) から実行する必要があります。Solaris インストールイメージまたはブート専用ディレクトリの Solaris リリースは、クライアントにインストールする予定の Solaris リリースと同じでなければなりません。

`cdrom-mnt-pt/Solaris_XX/Tools/add_install_client [-i IP_address] [-e Ethernet_address] [-s server_name : path] [-c server_name : path]`

add_to_install_server(1M)

```
[ -n [server] ] : name_service [ ( netmask) ] [ -p server_name : path ]  
host_name platform_group
```

プラットフォームグループ内のあるプラットフォームのインスタンスをインストールサーバーに追加する場合は、次の形式の `add_install_client` コマンドを使用します。このグループは DHCP を使用して起動および構成されます。スクリプトは、必要な構成作業をサーバー上で実行し、ユーザーがそのグループのために DHCP サーバーに追加しなければならないデータを出力します。

```
cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server:path]  
[-c server:path] [-p server:path] [-t install boot image path]  
[-f boot file name] platform_name platform_group
```

インストールサーバーに単一のクライアントを追加する場合は、次の形式の `add_install_client` コマンドを使用します。このクライアントは DHCP を使用して起動および構成されます。スクリプトは、必要な構成作業をサーバー上で実行し、ユーザーがそのクライアントのために DHCP サーバーに追加しなければならないデータを出力します。すでに使用しているものにも、上記のように `-f` フラグを追加する必要があります。`-f` を使用すると、ユーザーは所定のクライアントに使用する起動ファイルの名前を指定できます。

```
cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d  
[-s server_name:path] [-c server_name:path] [-p server_name:path]  
[-t install_boot_image_path] [-f boot_file_name] -e Ethernet_address  
platform_group
```

注 - x86 アーキテクチャの Pre-boot eXecution Environment (PXE) クライアントを登録する場合は、必ず `-d` オプションを使用してください。x86 PXE クライアントは構成に DHCP を使用します。

他の Solaris CD とネットワークインストールサーバー上の既存のイメージをマージする場合は、`add_to_install_server` を使用します。マージ可能な各 CD (現時点では OS CD 2 および Language CD) には専用の `add_to_install_server` スクリプトが含まれています。配布された CD に含まれているもの以外の `add_to_install_server` スクリプトを使用しないでください。

`rules` ファイル (カスタム JumpStart インストールを使用する場合に限り必要) のルールを検証する場合は、`check` を使用します。

既存のネットワークインストールサーバーのミニルートを Solaris Installation CD のミニルートに置き換える場合は、`modify_install_server` を使用します。ネットワークインストールサーバーのインストール時のユーザーインターフェースが、Solaris Installation CD の Web Start ユーザーインターフェースに変更されません。

`modify_install_server` コマンドを使用する場合は、既存のインストールイメージ (`setup_install_server` を使用して作成) があらかじめ存在していなければなりません。

add_to_install_server(1M)

ネットワークインストール用のクライアントを削除する場合は、`rm_install_client` を使用します (このコマンドは `bootparams(4)` ファイルを更新します)。

Solaris CD を (インストールサーバーを設定するために) ディスクにコピーする、(WANboot インストールサーバーを設定するために) WANboot ミニルートイメージを構築する、または Solaris CD のブートソフトウェアだけを (ブートサーバーを設定するために) ディスクにコピーする場合は、`setup_install_server` を使用します。ネットワーク経由でクライアントをインストールするには、インストールサーバーが必要です。インストールサーバーとインストールするクライアントが異なるサブネットに属している場合、ネットワークインストールを行うには、ブートサーバーも必要です (ブートサーバーはクライアントのサブネットに配置する必要があります)。

オプション `add_install_client` には、次のオプションを指定できます。

-b *property=value*

ブートサーバーの TFTP ディレクトリ (デフォルトは `/tftpboot`) にあるクライアント特有の `bootenv.rc` ファイルにプロパティ値を設定します。クライアントに特有のブートプロパティを設定する場合には、このオプションを使用します。`bootpath` プロパティを使用すると、デバイス構成補助を使用して手動で選択しなくても、ブートデバイスを自動的に選択できます。`boot-args` プロパティを使用すると、JumpStart インストールを自動的に選択できます。`eeeprom(1M)` を参照してください。

このオプションは x86 クライアントにのみ使用できます。このオプションを使用するときには、必ず `-d` オプションと `-e` を同時に指定してください。

-c *server_name:path*

このオプションが必要なのは、カスタム JumpStart インストール用の JumpStart ディレクトリを指定する場合だけです。`server_name` は、JumpStart ディレクトリが置かれているサーバーのホスト名です。`path` は JumpStart ディレクトリの絶対パスです。

-d

DHCP クライアントを指定します。

-e *Ethernet_address*

インストールするシステムの Ethernet アドレスを指定します。

-f

インストールするクライアントの `boot_file_name` を指定します。

-i *IP_address*

インストールするクライアントの IP アドレスを指定します。

-n [*server*]: *name_service*[(*netmask*)]

このオプションでは、システム構成時に使用するネームサービスを指定します。`bootparams(4)` ファイル内の `ns` キーワードを設定します。

name_service

有効なエントリは `nis`、`nisplus`、および `none` です。

add_to_install_server(1M)

netmask

数字を4つずつピリオドで区切って並べたもので、IPアドレスのどの部分がネットワーク部分で、どれがホスト部分かを指定します。

server

サーバーの名前または指定したネームサービスのIPアドレス。指定したサーバーが異なるサブネット上にある場合は、クライアントからサーバーにアクセスするために、*netmask*が必要なことがあります。

-p *server_name*: *path*

このオプションは、システム情報またはネットワーク情報を前もって設定するために必要な、ユーザーが定義した *sysidcfg* ファイルの位置です。*server_name* は有効なホスト名またはIPアドレスです。*path* はJumpStart ディレクトリの絶対パスです。

-s *server_name*:*path*

このオプションが必要なのは、ブートサーバーから *add_install_client* を使用する場合だけです。サーバー名およびこのインストールで使用する Solaris インストールイメージの絶対パスを指定します。*path* はマウントされている Solaris CD のパスまたは Solaris CD のコピーが置かれているディレクトリのパスです。

add_to_install_server コマンドには、次のオプションを指定できます。

-p

コピーする CD (補助的な製品が収められている) の位置を指定します。

-s

ユーザーが、インストールする必要がある製品だけをリストから選択できるようにします。

check コマンドには、次のオプションを指定できます。

-p *install_dir_path*

使用中のシステムの *check* スクリプトではなく、指定された Solaris インストールイメージの *check* スクリプトを使用することによって、ルールファイルを検査します。*install_dir_path* は、ローカルディスク上またはマウントした Solaris CD 上の Solaris インストールイメージのパスです。

旧バージョンの Solaris が稼働しているシステムでは、このオプションを使用して、最新バージョンの *check* を実行できます。

-r *rulesfile*

rules 以外の名前のルールファイルを指定します。このオプションを使用すると、ルールの妥当性を検証してから、ルールファイルに組み込むことができます。*check* はルールが有効であるかどうかを報告するだけで、カスタム JumpStart インストールに必要な *rules.ok* ファイルは作成しません。

modify_install_server コマンドには、次のオプションを指定できます。

-p

このオプションは、*install_dir_path/Solaris_XX/Tools/Boot.orig* にすでに存在しているイメージのミニルートを保持します。

add_to_install_server(1M)

setup_install_server コマンドには、次のオプションを指定できます。

- b
サーバーの設定を、ブートサーバー専用にします。
- t
代替ミニルート指定できるようにします。
- w
WANboot ミニルートイメージを構築します。

オペランド add_install_client コマンドには、次のオペランドを指定できます。

host_name

インストールするクライアントの名前です。

platform_group

特定のソフトウェアを配布するために、ベンダーが定義したハードウェアプラットフォームグループ。以下に有効なプラットフォームグループの例を示します。

システム	プラットフォームグループ
x86	i86pc
Sun Fire 4800	sun4c

システムのプラットフォームグループを調べるには、(-m オプションを指定して) `uname(1)` コマンドを使用します。

platform_name

システムのプラットフォーム名を調べるには、-i オプションを指定して `uname(1)` コマンドを使用します。

次に、`uname command` コマンドを使用して Ultra 1 のシステムプラットフォーム名を調べる例を示します。

```
uname -i  
システムは、次のように応答します。
```

```
SUNW,Ultra-1  
システムのプラットフォーム名は SUNW,Ultra1 です。
```

次のコマンドは、Ultra 1 用の `add_install_client` を呼び出します。

```
add_install_client -d SUNW,Ultra-1 sun4u
```

IA32 プラットフォームでは、プラットフォーム名はつねに `SUNW.i86pc` です。

次のコマンドは、IA32 プラットフォーム用の `add_install_client` を呼び出します。

```
add_install_client -d SUNW.i86pc i86pc
```

`rm_install_client` コマンドには、次のオペランドを指定できます。

add_to_install_server(1M)

host_name

削除するクライアントの名前です。

setup_install_server コマンドには、次のオペランドを指定できます。

install_dir_path

Solaris ソフトウェアのコピー先となるディレクトリの絶対パスです。このディレクトリは空でなければなりません。

wanboot_image_path

WANboot ミニルートイメージを含むファイルが作成されるディレクトリの絶対パスです。

使用例 例 1 add_install_client の使用

次の add_install_client コマンドは、マウント済みの Solaris CD からネットワークインストール用のクライアントを Solaris 8 のインストールサーバーに追加します。

```
example# cd /cdrom/cdrom0/s0/Solaris_9/Tools
example# ./add_install_client system_2/sun4u
```

例 2 add_install_client の使用

次の add_install_client コマンドは、ネットワークインストール用のクライアントをインストールサーバーに追加します。カスタム JumpStart インストールを実行するためのルールファイルおよびプロファイルファイルが置かれている JumpStart のサーバーとディレクトリのパスを -c オプションで指定します。Solaris CD は Solaris 8 上の /export/install ディレクトリにコピーされています。

```
example# cd /export/install/Solaris_8/Tools
example# /add_install_client -c install_server:/jumpstart system_1 i86pc
example# ./add_install_client -c install_server:/jumpstart system_2 i86pc
```

例 3 add_install_client の使用

次の add_install_client コマンドは、起動ファイル sun4u.solaris8 を使用する特定の sun4u プラットフォームマシン (8:0:20:99:88:77) に対するサポートを追加します。

```
example# add_install_client -d -f sun4u.solaris8 -e 8:0:20:99:88:77 sun4u
```

例 4 add_install_client の使用

次の add_install_client コマンドは、PXE 規格を使用してネットワークから起動する x86 クライアントを追加します。

```
example# add_install_client -d -s svrname:/mnt/export/root SUNW.i86pc i86p
```

例 5 add_to_install_server の使用

次の add_to_install_server コマンドは、CD 上のすべての製品ディレクトリにあるパッケージを Solaris 8 上の既存のインストールサーバーにコピーします。

例 5 add_to_install_server の使用 (続き)

```
example# cd /cdrom/cdrom0/s0
example# ./add_to_install_server /export/Solaris_8
```

例 6 check の使用

次の check コマンドは、カスタム JumpStart インストールに使用するルールファイルの構文を検査します。

```
example# cd jumpstart_dir_path
example# ./check -p /cdrom/cdrom0/s0
```

例 7 modify_install_server の使用

次の modify_install_server コマンドは、上記の setup_install_server を使用して作成されたミニルートを Boot.orig に移し、Solaris Installer CD 上のミニルートに置き換えます。

```
example# cd /cdrom/cdrom0/s0
example# ./modify_install_server -p /export/install /cdrom/cdrom0/s1
```

例 8 modify_install_server の使用

次の modify_install_server コマンドは、上記の setup_install_server を使用して作成されたミニルートを Solaris Installer CD 上のミニルートに置き換えます。

```
example# cd /cdrom/cdrom0/s0
example# ./modify_install_server /export/install /cdrom/cdrom0/s1
```

例 9 rm_install_client の使用

次の rm_install_client コマンドは、Solaris 8 上のネットワークインストール用クライアントを削除します。

```
example# cd /export/install/Solaris_8/Tools
example# ./rm_install_client holmes
example# ./rm_install_client watson
```

例 10 setup_install_server の使用

次の setup_install_server コマンドは、マウント済みの Solaris CD を Solaris 8 のローカルディスク上にある /export/install というディレクトリにコピーします。

```
example# cd /cdrom/cdrom0/s0/Solaris_8/Tools
example# ./setup_install_server /export/install
```

例 11 setup_install_server の使用

次の setup_install_server コマンドは、マウント済みの Solaris CD のブートソフトウェアを、Solaris 8 上のサブネット用ブートサーバーになるシステムの /boot_dir というディレクトリにコピーします。

add_to_install_server(1M)

例 11 setup_install_server の使用 (続き)

```
example# cd /cdrom/cdrom0/s0/Solaris_8/Tools
example# ./setup_install_server -b /boot_dir
```

例 12 setup_install_server の使用

setup_install_server はデフォルトで、マウント済みの Solaris 配布ディスク上の Solaris ../Tools/Boot にあるインストールブートディレクトリを検索します。

以前に ./setup_install_server -b /boot_dir コマンドを使用して、ネットワークブートサーバー上に作成したディレクトリと別のディレクトリが必要な場合は、-t オプションを使用できます。

```
example# cd /cdrom/cdrom0/s0/Solairs_9/Tools
example# ./setup_install_server -t /boot_dir /export/install
```

例 13 setup_install_server と WANboot オプションの使用

次の setup_install_server コマンドは WANboot ミニルートファイルシステムのイメージを作成し、それを /wanboot_dir/miniroot に保存します。

```
example# cd /cdrom/cdrom0/s0/Solairs_9/Tools
example# ./setup_install_server -w /wanboot_dir /export/install
```

例 14 x86: ネットワークインストール時に使用するシリアルコンソールの指定 (CD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するシリアルコンソールを指定する方法を示します。この例では、インストールクライアントを次の方法で設定します。

- -d オプションは、クライアントが DHCP を使用してインストールパラメータを設定することを示します。
- -e オプションは、Ethernet アドレスが 00:07:e9:04:4a:bf であるクライアントでのみインストールが行われることを示します。
- 1 つめと 2 つめの -b オプションは、インストールプログラムが入力デバイスおよび出力デバイスとしてシリアルポート ttya を使用することを示します。

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
i86pc
```

-b オプションで使用できるブートプロパティの変数とその値については、eprom(1M) を参照してください。

例 15 ネットワークインストール時に使用するブートデバイスの指定 (CD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するブートデバイスを指定する方法を示します。インストールクライアントを設定するときにブートデバイスを指定する場合は、インストール時にデバイス構成補助によって出されるこの情報は表示されません。

例 15 ネットワークインストール時に使用するブートデバイスの指定 (CD) (続き)

この例では、次の方法でインストールクライアントを設定します。

- -d オプションは、クライアントが DHCP を使用してインストールパラメータを設定することを示します。
- -e オプションは、Ethernet アドレスが 00:07:e9:04:4a:bf であるクライアントでのみインストールが行われることを示します。
- 1 つめと 2 つめの -b オプションは、インストールプログラムが入力デバイスおよび出力デバイスとしてシリアルポート ttya を使用することを示します。
- 3 つめの -b オプションは、インストールプログラムがインストール時に特定のブートデバイスを使用することを示します。
- ブートデバイスのパスは、使用しているハードウェアによって異なります。
- i86pc というプラットフォーム名は、クライアントが x86 ベースのシステムであることを示します。

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
-b "bootpath=/pci@0,0/pci108e,16a8@8" i86pc
```

-b オプションで使用できるブートプロパティの変数とその値について詳細は、eeprom(1M) を参照してください。

終了ステータス 次の終了ステータスが返されます。

```
0      正常終了
1      エラーが発生した
```

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	Solaris CD

関連項目 uname(1), bootparams(4), attributes(5)

『Solaris 9 インストールガイド』

afbconfig(1M)

名前	afbconfig, SUNWafb_config – AFB グラフィックスアクセラレータの設定
形式	<pre>/usr/sbin/afbconfig [-dev <i>device-filename</i>] [-res <i>video-mode</i> [now try] [noconfirm nocheck]] [-file machine system] [-deflinear true false] [-defoverlay true false] [-overlayorder first last] [-expvis enable disable] [-sov enable disable] [-maxwinds <i>n</i>] [-extovl enable disable] [-g <i>gamma-correction-value</i>] [-gfile <i>gamma-correction-file</i>] [-propt] [-prconf] [-defaults] /usr/sbin/afbconfig [-propt] [-prconf] /usr/sbin/afbconfig [-help] [-res ?]</pre>
機能説明	<p>afbconfig は、AFB グラフィックスアクセラレータおよび AFB 対応の X11 ウィンドウシステムのデフォルトの一部を設定します。</p> <p>afbconfig の次の形式では、指定したオプションを OWconfig ファイルに保存します。</p> <pre>/usr/sbin/afbconfig [-dev<i>device-filename</i>] [-res <i>video-mode</i> [now try] [noconfirm nocheck]] [-file machine system] [-deflinear true false] [-defoverlay true false] [-overlayorderfirst last] [-expvisenable disable] [-sov enable disable] [-maxwinds<i>n</i>] [-extovl enable disable] [-g<i>gamma-correction-value</i>] [-gfile<i>gamma-correction-file</i>] [-propt] [-prconf] [-defaults]</pre> <p>これらのオプションは、次にウィンドウシステムをそのデバイスで実行するときに AFB デバイスを初期化するために使用されます。OWconfig ファイル内のオプションの更新は、異なるウィンドウセッションや再起動したシステムでも有効となります。</p> <p>-prconf、-propt、-help、-res ? オプションだけを起動する次の形式では、OWconfig ファイルは更新されません。</p> <pre>/usr/sbin/afbconfig [-propt] [-prconf] /usr/sbin/afbconfig [-help] [-res ?]</pre> <p>また、次の形式では、その他のオプションはすべて無視されます。</p> <pre>/usr/sbin/afbconfig [-help] [-res ?]</pre> <p>オプションは、一度に 1 つの AFB デバイスに対してのみ指定することができます。複数の AFB デバイスに対してオプションを指定するには、afbconfig コマンドを複数回起動する必要があります。</p> <p>afbconfig で指定できるのは、AFB 固有のオプションだけです。デフォルトの表示色数、デフォルトのビジュアルクラスなどを指定する通常のウィンドウシステムのオプションは、openwin コマンド行のデバイス修飾子で指定してください。</p>

ユーザーは、更新する OWconfig ファイルを指定することもできます。デフォルトでは、/usr/openwin ディレクトリツリーにあるマシン固有のファイルが更新されます。別のファイルを指定するには、-file オプションを使用します。たとえば、/etc/openwin ディレクトリツリーにあるシステム共通の OWconfig ファイルを代わりに更新することができます。

これらの標準 OWconfig ファイルのどちらもスーパーユーザーのみが書き込みを行えます。したがって、スーパーユーザーが所有する afbconfig プログラムは、setuid による root の権限で実行されます。

オプションのデフォルト値

afbconfig コマンド行で指定されていないオプションについては、対応する OWconfig ファイル中のオプションは更新されず、ファイル内の値がそのまま使用されます。ウィンドウシステムを実行する際に、afbconfig による AFB オプションの指定がまったくなかった場合は、デフォルト値が使用されます。オプションのデフォルトは次のとおりです。

```
-dev          /dev/fbs/afb0
-file         machine
-res         none
-deflinear   false
-defoverlay  false
-linearorder last
-overlayorder last
-expvis      enabled
-sov        enabled
-maxwids     32
-extovl     enabled
-g          2.22
```

-res オプションのデフォルト値 none とは、ウィンドウシステムが実行された場合に、画面解像度がそのデバイスに現在プログラムされている表示モードになることを意味しています。

これによって、PROM によってデバイスの解像度を指定しているユーザーとの共用性が保てます。(Gx などの)一部のデバイスでは、PROM が表示モードを指定する唯一の手段です。これは、デフォルトの AFB 表示モードは、最終的に PROM によって決まることを意味しています。

オプション

次のオプションがサポートされています。

```
-defaults
  すべてのオプションの値をそれぞれのデフォルト値に戻します。
```

afbconfig(1M)

-deflinear true | false

AFB には、2 種類の画像表示形式があります。リニア画像と非リニア画像です。リニア画像はガンマ補正され、非リニア画像は補正されません。リニア画像版も非リニア画像版も、ともに持つ画像表示形式が 2 つあります。24 ビットトゥルーカラーと 8 ビットスタティックグレーです。

true のときは、デフォルトの画像表示形式として、デフォルトで選択されたオプション (特に、Xsun(1) の defdepth および defclass オプション。詳細は OpenWindows のマニュアルページを参照) を満たすリニア画像がセットされます。

false のとき、または他のデフォルトで選択されたオプションを満たすリニア画像がないときは、これらの他のオプションを満たす、非リニア画像がデフォルトとして選択されます。AFB にはリニアオーバーレイ画像表示形式がないため、-defoverlay オプションが存在するときには、このオプションを使用することはできません。

-defoverlay true | false

AFB が、残りの AFB 画像から切り離されたピクセルを持つ 8 ビット疑似カラー画像を提供します。これを、オーバーレイ画像といいます。この画像表示形式で作成されたウィンドウは、他の画像表示形式で作成されたウィンドウに影響を与えません。逆に、他の画像表示形式で作成されたウィンドウは、オーバーレイウィンドウに影響を与えます。

この画像を使用して作成されたウィンドウで使用することができる色の数は -extov1 オプションの設定に依存します。-extov1 オプションが有効になっている場合は、256 種類の不透明カラーの値による拡張オーバーレイを使用することができます(-extov1 を参照)。-extov1 が無効になっている場合は、拡張オーバーレイを使用することはできず、この画像には、不透明カラーの (256 -maxwids) の値が使用されます (-maxwids を参照)。

-defoverlay の値が true である場合には、オーバーレイ画像がデフォルト画像になります。-defoverlay の値が false の場合には、他のデフォルトで選択された def、depth および defclass オプションを満たすオーバーレイでない画像表示形式が、デフォルトの画像表示形式として選択されます。詳細は、OpenWindows のマニュアルページを参照してください。

-defoverlay true オプションが使用されるときは、常に openwin コマンド行で選択されたデフォルトの深さとクラスは 8 ビット疑似カラーである必要があります。それ以外の場合は、警告メッセージが出力され、-defoverlay オプションは false として扱われます。

-deflinear オプションが存在するときには、AFB にはリニアオーバーレイ画像表示形式がないため、このオプションは使用することができません。

-dev device-filename

AFB 特殊ファイルを指定します。デフォルトは /dev/fbs/afb0 です。

-expvis enable | disable

enable にすると、OpenGL Visual Expansion が起動されます。選択された画像表示形式グループ (8 ビット PseudoColor、24 ビット TrueColor など) が画像表示形式リストに見つかります。

-extovl enable | disable

enable にすると、拡張オーバーレイを使用することができます。このオーバーレイ画像には 256 種類の不透明カラーがあります。SOV 画像には 255 種類の不透明カラーと 1 種類の透明カラーがあります。

また、このオプションは、ハードウェアによる透明カラーを有効にするため、SOV 画像を使用するウィンドウで、より高い性能が得られます。

-file machine | system

更新する OWconfig ファイルを指定します。machine が指定された場合は、/etc/openwin ディレクトリツリーにあるマシン固有の OWconfig ファイルが使用されます。system が指定された場合は、/usr/openwin ディレクトリツリーにある共通の OWconfig ファイルが使用されます。指定されたファイルがない場合は、新たに生成されます。

-g gamma-correction-value

ガンマ補正の値を変更することができます。すべてのリニア画像ではガンマ補正を使用することができます。デフォルトでは、gamma-correction-value は 2.22 です。0 より小さい値は無効 (不正) です。ガンマ補正の値はリニア画像に適用され、リニア画像の有効ガンマ値は 1.0 になります。これは、XSolarisGetVisualGamma(3) によって返される値です。この機能については、XSolarisGetVisualGamma(3) を参照してください。

このオプションは、ウィンドウシステムが稼動しているときに使用することができます。ガンマ補正の値を変更すると、リニア画像を使用して表示されているすべてのウィンドウに影響を受けます。

-gfile gamma-correction-file

指定されたファイル (gamma-correction-file) からガンマ補正表を読み込みます。gamma-correction-file は、各行が R、G、B チャネルの値を持つように書式化されている必要があります。それらの値は、16 進数で指定し、値と値の間は 1 つ以上のスペースで区切ります。また、gamma-correction-file は、そのような 3 つの値の組が 256 種類定義されます。

gamma-correction-file の例を以下に示します。

```
0x00 0x00 0x00
0x01 0x01 0x01
0x02 0x02 0x02
...
...
0xff 0xff 0xff
```

このオプションを使用することによって、ウィンドウシステムが稼動しているときにガンマ補正表を読み込むことができます。新しいガンマ補正は、このリニア画像によって表示されているすべてのウィンドウに影響を与えます。ユーザーが指定した表によってガンマ補正を行う際は、ガンマ補正の値は定義されません。デフォルト

afbconfig(1M)

トでは、ウィンドウシステムはガンマ補正值として 2.22 を使用し、このガンマ補正值に対応してウィンドウシステムが作成したガンマ補正表を読み込みます。

-help

afbconfig コマンド行のオプションと機能の概要を一覧で表示します。

-linearorder first | last

first のときには、AFB スクリーン用の X11 スクリーン画像表示形式リスト上で、リニア画像が非リニア画像より前に表示されます。last のときには、非リニア画像は、リニア画像より前に表示されます。

-maxwidths *n*

ウィンドウ ID s (WIDs) として使用するために予約される最大数の AFB X チャンネルピクセル値を指定します。オーバーレイカラーマップのピクセル値の残りは、通常の X11 の未使用のカラーピクセルのために使用されます。確保された WIDs は、(XGL などの) 3 次元グラフィックスウィンドウ、MBX ウィンドウと、デフォルト以外の画像表示形式をもつウィンドウにより発生順に割り当てられます。X チャンネルコードの 0 から (255 - *n*) は、未使用のカラーピクセルです。(255 - *n* + 1) から 255 の X チャンネルコードは、WID として使用するために予約されます。適切な値は、1、2、4、8、16、32、64 です。

このオプションは -extov1 が無効になっている場合のみ使用することができます。

-overlayorder first | last

first のときには、AFB スクリーン用の X11 スクリーン画像表示形式リスト上で、8 ビット疑似カラーオーバーレイ画像が、非オーバーレイ画像より前に表示されます。last のときには、非オーバーレイ画像は、オーバーレイ画像より前に表示されます。

-propt

-file オプションで指定された OWconfig ファイルに書かれた AFB オプションの値のうち、-dev オプションで指定されたデバイスに対するものすべてを表示します。afbconfig の呼び出しが終了した後に、OWconfig ファイルに書き込まれるオプションの値を表示します。

次に表示例を示します。

```
--- OpenWindows Configuration for /dev/fbs/afb0 ---
OWconfig: machine
Video Mode: 1280x1024x76
Default Visual: Non-Linear Normal Visual
Visual Ordering: Linear Visuals are last
                  Overlay Visuals are last
OpenGL Visual Expansion: enabled
Server Overlay Visuals: enabled
Extended Overlay: enabled
Underlay WIDs: 64 (not configurable)
Overlay WIDs: 4 (not configurable)
Gamma Correction Value: 2.220
Gamma Correction Table: Available
```


-prconf

AFB ハードウェア構成を表示します。

次に表示例を示します。

```
--- Hardware Configuration for /dev/fbs/afb0 ---
Type: double-buffered AFB with Z-buffer
Board: rev 0 (Horizontal)
Number of Floats: 6
PROM Information: @(#)afb.fth x.xx xx/xx/xx
AFB ID: 0x101df06d
DAC: Brooktree 9070, version 1 (Pac2)
3DRAM: Mitsubishi 130a, version x
EDID Data: Available - EDID version 1 revision x
Monitor Sense ID: 4 (Sun 37x29cm RGB color monitor)
Monitor possible resolutions: 1024x768x77, 1024x800x84, 1
                             1152x900x76, 1280x1024x67, 1280x1024x76, 960x680xx108s
Current resolution setting: 1280x1024x76
```

-sov enable | disable

enable にすると、ルートウィンドウの `SERVER_OVERLAY_VISUALS` 属性が有効になります。SOV 画像が転送され、それらの透過タイプ、値、階層は、この属性によって参照することができます。**disable** にすると、`SERVER_OVERLAY_VISUALS` 属性は定義されません。SOV 画像は転送されません。

-res video-mode [now | try [noconfirm | nocheck]]

指定した AFB デバイスに接続されているモニターを制御する際に使われる表示モードを指定します。

組み込まれている表示モードの形式は次のとおりです。 `widthxheightxrate width` はピクセル単位のスクリーン幅、`height` はピクセル単位のスクリーンの高さ、`rate` は画面を垂直方向に再描画する周期です。

960x680x112s や 960x680x108s の `s` 接尾辞は、これらが立体表示モードであることを意味します。640x480x60i や 768x575x50i の `i` 接尾辞は、インタレース表示タイミングを有効にします。この接尾辞がない場合は、ノンインタレースタイミングが使用されます。

便宜上、`-res` にリフレッシュレートを指定する際、値の直前に `x` の代わりに `@` を使用できます。たとえば、1280x1024@76 のように指定できます。AFB が対応している一部の表示モードには、モニターが対応していない場合があります。また、AFB がサポートする表示モードにも、モニターがサポートしていないものがあります。AFB デバイスとモニターの両方がサポートしている表示モードのリストは、`-res ?` オプション付きの `afbconfig` (形式の項に記された 3 番目の形式) を実行することによって得ることができます。

AFB がサポートしている表示モードのリストを次に示します。

```
1024x768x60
1024x768x70
1024x768x75
```

afbconfig(1M)

1024x768x77
1024x800x84
1152x900x66
1152x900x76
1280x800x76
1280x1024x60
1280x1024x67
1280x1024x76
960x680x112s (立体表示)
960x680x108s (立体表示)
640x480x60
640x480x60i (インタレース)
768x575x50i (インタレース)

便宜上、AFB がサポートしている表示モードのいくつかには記号名が定義されています。*widthxheightxrate* の形式の代わりに、記号名を *-res* オプションの引数として指定することができます。記号名 *none* は、ウィンドウシステムを実行すると、画面の解像度は現在デバイスにプログラムされている表示モードになることを意味します。

AFB がサポートしている表示モードのリストを次に示します。

記号名 対応する表示モード

svga 1024x768x60
1152 1152x900x76
1280 1280x1024x76
stereo 960x680x112s
ntsc 640x480x60i
pal 768x575x50i
none (上記参照)

-res オプションには、表示モードの直後に次の追加引数を指定することができます。追加引数は、単独でも複数でも指定できます。

noconfirm *-res* オプションを指定した際に、システムが使用不可であっても、表示出力のない状態になる場合があります。このような状況は、特定のコードが読み込まれた際のモニターセンスコードにあいまいさがあった場合などに発生します。このような事態を避けるために *afbconfig* のデフォルトの動作では、この問題についての警告メッセージと、処理を継続するかどうかを確認するメッセージを表示します。*noconfirm* オプションを指定すると、*afbconfig* コマンドはこの確認をせずに、要求のあった表示モードにプログラムします。このオプションは、*afbconfig* がシェルスクリプトから実行されている場合に便利です。

nocheck このオプションを指定すると、モニターセンスコードに基づく通常のエラーチェックが行われません。ユーザーによって指定された表示モードは、現在接続されているモニターに適切かど

afbconfig(1M)

うかにかかわらず受け付けられます。このオプションは、AFB デバイスに異なるモニターを接続する場合に便利です。このオプションの指定は、noconfirm の指定も兼ねます。

now OWconfig ファイルの表示モードを更新するとともに、AFB デバイスが指定した表示モードにただちにプログラムされます。この機能は、ウィンドウシステムを開始する前に表示モードを変更する際に便利です。

対象となるデバイスが稼働している間 (たとえば、ウィンドウシステムの稼働中) に、この引数をafbconfig に指定することはお勧めしません。予期しない結果になることもあります。now 引数を指定してafbconfig コマンドを実行する場合は、最初にウィンドウシステムを終了してください。now 引数がウィンドウシステムのセッション中に使用された場合、表示モードはただちに更改されますが、画面の幅や高さはそのセッションが終了して次のセッションに入るまで更改されません。さらに、立体表示モードではシステムが更改を認識しないことがあります。したがって、ウィンドウシステムの稼働中には絶対に now オプションを指定しないでください。

try このオプションを指定すると、指定した表示モードが試験的にプログラムされます。ユーザーは、指定した表示モードを使用する場合は、メッセージが表示されてから 10 秒以内に「y」を入力します。表示されたモードを使用しない場合は、10 秒以内に任意の文字を入力します。「y」またはReturn キー以外の文字の入力は、すべて「使用しない」と判断され、以前の表示モードに戻されるため、OWconfig ファイル中の表示モードは書き換えられません (その他の指定されたオプションは有効となります)。Return キーの入力があった場合は、新しい表示モードを保持するかどうかを yes または no で確認するメッセージが表示されます。このオプションの指定は、now 引数の指定も兼ねます (now 引数の注意を参照)。

使用例 例 1 モニターの種類の変更

モニターの種類を、垂直周波数 76 Hz で解像度 1280x1024 に変更する例を以下に示します。

```
example% /usr/sbin/afbconfig -res 1280x1024x76
```

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	SUNWafbcf

afbconfig(1M)

関連項目 | mmap(2), attributes(5)

名前	automount – 自動マウントポイントのインストール
形式	<code>/usr/sbin/automount [-t <i>duration</i>] [-v]</code>
機能説明	<p>automount ユーティリティは、autofs マウントポイントをインストールし、automount マップと各マウントポイントを対応付けます。autofs ファイルシステムは、自身のファイルシステム内のディレクトリへのアクセスを監視し、automountd(1M) デーモンに通知します。このデーモンはマップを使用してファイルシステムを特定し、autofs ファイルシステム内の参照ポイントにそのファイルシステムをマウントします。マップは、<code>/etc/auto_master</code> マップのエントリを使用するか、または直接マップを使用することによって、autofs マウントに割り当てることができます。</p> <p>一定時間内(デフォルトでは 10 分間) にファイルシステムに対してアクセスがなければ、automountd デーモンはファイルシステムのマウントを解除します。</p> <p>ファイル <code>/etc/auto_master</code> により、すべての autofs マウントポイントの位置が決定されます。デフォルトでは、このファイルには 4 つのエントリが含まれます。</p> <pre># Master map for automounter # +auto_master /net -hosts -nosuid /home auto_home /xfn -xfn</pre> <p><code>+auto_master</code> エントリは、外部 NIS または NIS+ マスターマップへの参照です。マスターマップが存在する場合、<code>+auto_master</code> エントリの位置へ読み取られます。マスターファイルの残りのエントリは、autofs マウントが行われるディレクトリを指定し、さらにそのディレクトリに対応付けるオートマウントマップを指定します。各エントリの 3 番目のオプションフィールドには、任意のマウントオプションを指定できます。これらのオプションは、マウントオプションが明示的に指定されていない、マップ内のすべてのエントリに適用されます。automount コマンドは通常、引数を指定しないで実行します。automount コマンドは、<code>/etc/auto_master</code> 内のエントリと <code>/etc/mnttab</code> 内の autofs マウントのその時点のリストを比較し、autofs マウントを追加、削除、または変更することによって、<code>/etc/auto_master</code> の情報を <code>/etc/mnttab</code> ファイルに反映させます。automount コマンドはブート時に、マスターマップからすべての autofs マウントをインストールします。その後、マスターマップまたは直接マップの新規エントリを autofs マウントにインストールするか、削除されたエントリに関してのマウントを解除します。</p>
オプション	<p>次のオプションを指定できます。</p> <p><code>-t <i>duration</i></code> ファイルシステムが使用されていないときに、マウントしたままにしておく時間の間隔を秒数で指定します。デフォルトは 10 分です。</p> <p><code>-v</code> 詳細表示モード。autofs のマウント、マウント解除、またはその他の関連情報を通知します。</p>
使用法	

automount(1M)

マップエントリの形式

単純マップエントリ (マッピング) の形式は、次のとおりです。

```
key [ -mount-options ] location . . .
```

key は、直接マップで使用する場合はマウントするディレクトリのフルパス名であり、間接マップの場合はサブディレクトリからの相対名です。*mount-options* は、コンマで区切った *mount* オプションのリストです。*location* には、ファイルシステムがマウントされるディレクトリを指定します。単純 NFS マウントの場合は、*mount_nfs(1M)* で指定されているのと同じオプションが使用できます。*location* の形式は次のとおりです。

```
host: pathname
```

host は、マウントするファイルシステムが置かれているホスト名です。*pathname* は、マウントするディレクトリの絶対パス名です。

他のファイルシステム用のオプションについては、*mount_cachefs(1M)* など、他の *mount_** のマニュアルページを参照してください。

複製されたファイルシステム

複製された NFS ファイルシステムには複数の *location* フィールドを指定できます。その場合、*automount* とカーネルのそれぞれがその情報を使用して、可用性の向上を計ります。読み取り専用フラグがマップエントリに設定されている場合、*automount* はカーネルが使用する、いくつかの条件によってソートされている位置リストを使用してマウントします。サーバーが応答しなかった場合、カーネルは代替サーバーに切り替えます。*automount* のソート順序は、次のサーバーの選択方法を決定するのに使用されます。読み取り専用フラグが設定されていない場合、*automount* は同じソート順序によって選択された、最善の位置をマウントします。新しいサーバーが選択されるのは、マウント解除が可能だった場合に限られ、再マウントが実行されます。同じローカルサブネット上のサーバーに最も高い優先順位が与えられ、その次にローカルネット上のサーバーに高い優先順位が与えられます。等距離のサーバー間では、重み付け係数 (後述) が使用されていないかぎり、応答時間によって順序が決まります。

NFS Version 2 プロトコルと NFS Version 3 プロトコルの両方を使用するサーバーの位置がリストに含まれている場合、*automount* はリストに指定されたサーバー位置の一部だけを選択し、すべてのエントリが同じプロトコルになるようにします。NFS Version 3 プロトコルを使用するサーバーが選択されるのは、ローカルサブネット上にある、NFS Version 2 プロトコルを使用するサーバーが無視されない場合に限られます。詳細は『Solaris のシステム管理 (IP サービス)』を参照してください。

リストの各 *location* が同じ *pathname* を共有している場合、コンマで区切ったホスト名に対して、1 つの *location* を使用できます。

```
hostname,hostname . . . : pathname
```

automount(1M)

括弧で囲んだ整数を重み付け係数としてサーバー名のうしろに指定することにより、サーバー要求に重み付けを設定できます。重み付けが設定されていないサーバーは、ゼロの値 (選択される可能性が最も高い) が与えられているものとみなされます。値が大きくなるにしたがって、選択される可能性が小さくなります。次に例を示します。

```
man -ro alpha,bravo,charlie(1),delta(4) : /usr/man
```

ホスト alpha および bravo に最も高い優先順位が与えられており、ホスト delta には最も低い優先順位が与えられています。

選択プロセスではサーバーの近接度が優先されます。上記の例で、サーバー delta はクライアントと同じネットワークセグメントにあり、他のサーバーは異なるネットワークセグメントにある場合、重み付けの値は無視され、delta が選択されます。重み付けが有効なのは、同じネットワーク近接度を持つサーバー間で選択が行われる場合だけです。

サーバーごとにエクスポートポイントが異なる場合にも、重み付けを適用できます。以下に例を示します。

```
man -ro alpha:/usr/man bravo,charlie(1):/usr/share/man
      delta(3):/export/man
```

マッピングを複数の入力行にわたって記述する場合は、バックスラッシュ (\) で復帰改行をエスケープします。コメントは番号記号 (#) で始まり、その後の復帰改行で終了します。

マップキーの置換

アンパサンド (&) 記号は、使用されているエントリの key フィールドの値に展開されます。

```
jane sparcserver : /home/&
```

上記の場合、& は jane に展開されます。

ワイルドカード キー

アスタリスク (*) 記号は、key フィールドとして与えられた場合、どのようなキーにも当てはまるエントリとして認識されます。このようなエントリは、それまで一致しなかったあらゆるキーと一致します。たとえば、/configの間接マップに次のエントリが指定されている場合、

```
*          & : /export/config/&
```

位置を次のように指定できるリモートファイルシステムは、/config にすべて自動マウントされます。

```
hostname : /export/config/hostname
```

変数の置換

automount マップ内でクライアント固有の変数を使用できます。たとえば、マップに \$HOST が指定されている場合、automount は \$HOST をクライアントのホスト名に対応する現在値に展開します。次の変数を使用できます。

automount(1M)

ARCH	uname -m の出力に基づくアプリケーションアーキテクチャ	アーキテクチャ名。たとえば、sun4u マシンでは sun4
CPU	uname -p の出力	プロセッサタイプ たとえば、sparc
HOST	uname -n の出力	ホスト名 たとえば、biggles
OSNAME	uname -s の出力	OS 名 たとえば、SunOS
OSREL	uname -r の出力	OS リリース名 たとえば、5.3
OSVERS	uname -v の出力	OS バージョン たとえば、beta1.0
NATISA	isainfo -n の出力	システム固有の 命令セットアーキテクチャ たとえば、sparcv9

参照が接辞付き文字から保護される必要がある場合は、変数名を中括弧 ({ }) で囲みます。

多重マウント

多重マウントエントリの形式は、次のとおりです。

```
key [-mount-options] [ [mountpoint] [-mount-options] location. . . ] . . .
```

最初の `/[mountpoint]` は、最初のマウントでは省略可能ですが、以降のすべてのマウントでは必須です。省略可能な `mountpoint` は、`key` で指定されたディレクトリからの相対パス名とみなされます。最初の `mountpoint` を省略した場合、`/` (ルート) の `mountpoint` が暗黙に使用されます。

`/src` の直接マップに次のエントリがあるとします。

```
beta    -ro\  
/       svr1,svr2:/export/src/beta \  
/1.0    svr1,svr2:/export/src/beta/1.0 \  
/1.0/man svr1,svr2:/export/src/beta/1.0/man
```

すべてのオフセットは、`beta` の下のサーバー上になければなりません。automount は `svr1` または `svr2` のどちらかから、つまり最も近くにあり先に応答した方のホストから、`/src/beta`、`/src/beta/1.0`、および `/src/beta/1.0/man` を必要に応じて自動的にマウントします。

その他のファイル
システムタイプ

オートマウントは、デフォルトのファイルシステムタイプとして NFS マウントを想定しています。他のファイルシステムタイプを指定する場合は、`fstype` マウントオプションを使用します。`fstype` オプションと組み合わせることにより、そのファイルシステムタイプ固有の他のマウントオプションを使用できます。位置フィールドにはファイルシステムタイプ固有の情報を指定する必要があります。たとえば、CD ファイルシステムをマウントする場合のように、位置フィールドがスラッシュで始まる場合は、その前にコロンが必要です。

```
cdrom -fstype=hsfs,ro : /dev/sr0
```

`autofs` マウントを実行するには、次のように指定します。

```
src -fstype=autofs auto_src
```

注: この手順は、Volume Manager を使用しない場合に限って使用してください。

マップのデフォルトとしてマップ全体に適用する場合は、CacheFS によるマウントが最も便利です。マスターマップの次のエントリで、キャッシュファイルシステムのホームディレクトリのマウントを指定します。この場合、キャッシュディレクトリのデフォルト位置は `/cache` です。

```
/home auto_home -fstype=cachefs,backfstype=nfs
```

オプションの継承については、「注意事項」を参照してください。

間接マップ

間接マップを使用すると、コマンド行で指定した `directory` からマウントするサブディレクトリの、マッピングを指定できます。間接マップの各 `key` は、必要に応じてマウントする 1 つ以上のファイルシステムを参照する単純名です。

直接マップ

直接マップのエントリは、`autofs` マウントポイントと直接対応づけられます。各 `key` は `autofs` マウントポイントのフルパス名です。直接マップ全体が 1 つのディレクトリに対応付けられることはありません。

インクルード
マップ

次の形式のエントリを使用すると、マップ内に別のマップの内容を含めることができます。

```
+mapname
```

`mapname` がスラッシュで始まる場合、`mapname` はローカルファイルのパス名とみなされます。それ以外の場合、マップの位置は次に示すように、`/etc/nsswitch.conf` 内のオートマウント用のエントリに基づくネームサービススイッチのポリシーによって決まります。

```
automount: files nis
```

ネームサービスが `files` の場合、`/etc` 内のローカルファイルの名前が想定されます。検索対象のキーが取り込まれたマップにない場合、次のエントリで検索が続けられます。

automount(1M)

特殊マップ	<p>-hosts、-xfn、および -null という 3 種類の特種マップがあります。-hosts マップは /net ディレクトリと組み合わせて使用します。この場合、マップのキーは NFS サーバーのホスト名とみなされます。automountd デーモンは、エクスポートされたファイルシステムのサーバーリストに基づいて、マップエントリを動的に作成します。/net/hermes のディレクトリ参照では、hermes のルートからの相対位置にある、対応するディレクトリが参照されます。</p> <p>-xfn マップは、/xfn ディレクトリにあるフェデレーテッドネーミングサービス (FNS) 名前空間の初期コンテキストをマウントします。FNS の詳細は、fns(5)、fns_initial_context(5)、fns_policies(5) の各マニュアルページ、および『Federated Naming Service Guide』を参照してください。</p> <p>-null マップは、指定されたディレクトリに対してすでに設定されているマップを取り消します。これは /etc/auto_master において、+auto_master によって与えられるエントリから継承されるエントリを取り消す場合に便利です。ただし、取り込まれたマップのエントリの前に -null エントリを挿入しないと、効果はありません。</p>
実行可能マップ	<p>ファイルパーミッションの実行ビットが設定されているローカルマップは、オートマウンタによって実行され、引数として検索されるキーを与えられます。実行可能マップは、標準出力にオートマウンタのマップエントリの内容を返すようにします。エントリが特定できなかった場合、出力はありません。直接マップを実行可能にすることはできません。</p>
構成と auto_master マップ	<p>引数を指定しないで automount を起動した場合、automount はマスターマップで、autofs マウントポイントのリストおよび対応するマップを調べます。automount はまだマウントされていない autofs マウントがあればそれらをマウントし、マスターマップまたは直接マップから削除されている autofs マウントがあればそれらをマウント解除します。</p> <p>マスターマップは auto_master とみなされ、その位置はネームサービススイッチのポリシーによって決まります。通常、ローカルファイル /etc/auto_master がマスターマップとして最初に検索されます。</p>
ブラウズ機能	<p>Solaris 2.6 リリースでは、間接マップをブラウズできます。これにより、マウントされているかどうかに関係なく、マウント可能なマウントポイントをすべて表示できます。任意の間接 autofs マップに -nobrowse オプションを追加すると、ブラウズ機能を無効にできます。以下に例を示します。</p> <pre>/net -hosts -nosuid,nobrowse /home auto_home</pre> <p>この場合、/net で表示されるのはマウントされている <i>hostnames</i> ですが、/home ではマウント可能なすべてのマウントポイントが表示されます。autofs ファイルシステムのブラウズ機能は、-browse オプションによって有効になります。これはすべての間接マップにおけるデフォルトの設定です。</p>
終了ステータス	<p>次の終了ステータスが返されます。</p> <pre>0 正常終了</pre>

	1	エラーが発生した				
ファイル	/etc/auto_master	マスターオートマウントマップ				
	/etc/auto_home	オートマウントされるホームディレクトリ用のマップ				
	/etc/nsswitch.conf	ネームサービススイッチの構成ファイル				
属性	次の属性については attributes(5) のマニュアルページを参照してください。					
	<table border="1"> <thead> <tr> <th>属性タイプ</th> <th>属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWcsu</td> </tr> </tbody> </table>		属性タイプ	属性値	使用条件	SUNWcsu
属性タイプ	属性値					
使用条件	SUNWcsu					
関連項目	isainfo(1), ls(1), uname(1), automountd(1M), mount(1M), mount_cachefs(1M), mount_nfs(1M), attributes(5), fns(5), fns_initial_context(5), fns_policies(5), nfssec(5)					
	Solaris のシステム管理 (IP サービス)					
注意事項	<p>autofs マウントポイント間に階層関係が存在してはなりません。automount では、autofs マウント内に別の autofs マウントポイントを作成できません。</p> <p>直接マップエントリごとに新しい autofs マウントが作成されるので、直接マップはできるだけ短くしてください。</p> <p>直接マップと間接マップのエントリはどちらも、いつでも変更できます。新しい情報は、automountd が次回そのマップエントリを使用してマウントを実行するとき使用されます。</p> <p>マスターマップまたは直接マップに追加された新しいエントリは、automount コマンドを実行してそれらの新しいエントリを新しい autofs マウントポイントとしてインストールするまで使用できません。間接マップに追加された新しいエントリは、ただちに使用できます。</p> <p>Solaris 2.6 リリースでは、間接マップに対応づけられた autofs ディレクトリの表示 (ls(1) のマニュアルページを参照) には、潜在的にマウント可能なエントリがすべて含まれます。潜在的にマウント可能なエントリに対応づけられる属性は一時的なものです。実際のファイルシステム属性が表示されるのは、ファイルシステムのマウント後に限られます。</p> <p>マスターマップの 3 番目のオプションフィールドを指定すると、デフォルトのマウントオプションをマップ全体に割り当てることができます。これらのオプションが適用されるのは、マウントオプションが指定されていないマップエントリだけです。マップエントリにオプションが指定されていると、デフォルトのオプションは無効になります。現時点では、オプションの連結は行われません。</p> <p>オートマウントのデフォルトの NFS マウント操作の再試行回数は 0 です。つまり、マウント試行は 1 回だけで、再試行されません。これは、mount_nfs(1M) ユーティリティのデフォルト (10000 回) と大きく異なる点なので、注意してください。</p>					

automount(1M)

ネットワーク情報サービス (NIS) は従来、Sun イエローページ (YP) と呼ばれていました。これらの機能は同等です。

名前	boot – システムカーネルまたはスタンドアロンプログラムの起動
形式	
SPARC	boot [<i>OBP names</i>] [<i>file</i>] [-afV] [-D <i>default-file</i>] [<i>boot-flags</i>] [—] [<i>client-program-args</i>]
x86	b [<i>file</i>] [-f] [<i>boot-args</i>] i
機能説明	<p>ブートストラップとは、スタンドアロンプログラムを読み込んで実行する処理のことです。ここでのブートストラップとは、起動可能なオペレーティングシステムを読み込んで実行することを意味します。通常、スタンドアロンプログラムはオペレーティングシステムカーネル (kernel(1M) のマニュアルページを参照) ですが、代わりに任意のスタンドアロンプログラムを起動することもできます。SPARC ベースのシステムでは、オペレーティングシステム以外に起動できるスタンドアロンプログラムの代表的な例として、マシンの診断モニターがあります。</p> <p>スタンドアロンプログラムが動的にリンクされる実行可能プログラムとして識別された場合、boot は実行可能形式によって指定されたインタプリタ (リンカー / ローダー) を読み込み、そのインタプリタに制御を渡します。スタンドアロンプログラムが静的にリンクされている場合は、そのスタンドアロンプログラムに直接ジャンプします。</p> <p>カーネルは読み込まれると、UNIX システムを起動して、必要なファイルシステム (vfstab(4) のマニュアルページを参照) をマウントし、/sbin/init を実行して、システムを /etc/inittab で指定されている <i>initdefault</i> 状態にします。inittab(4) のマニュアルページを参照してください。</p> <p>SPARC ベースのシステムでは、ほとんどのマシンでのブートストラップ手続きは次の基本フェーズからなります。</p> <p>マシンの電源を投入すると、(PROM 内の) システムファームウェアが電源投入時自己診断テスト (POST) を実行します。この診断テストの形式と範囲は、システムに搭載されているファームウェアのバージョンによって異なります。</p> <p>診断テストが正常に完了した後、ファームウェアが使用する不揮発性記憶領域に適切なフラグが設定されていれば、ファームウェアは自動起動を試みます。ファームウェアは、読み込むファイルの名前とそのファイルを読み込むデバイスを選択することができます。</p> <p>これらのフラグと名前は、シェルから <i>eeprom(1M)</i> コマンドを使用するか、またはシステムの停止後に <i>ok</i> プロンプトで <i>PROM</i> コマンドを使用することにより、設定できます。</p> <p>第 2 レベルのプログラムは、<i>ufsboot</i> (ディスクから起動する場合) または <i>inetboot</i> あるいは <i>wanboot</i> (ネットワークから起動する場合) のどれかです。</p> <p>ネットワーク起動</p>
SPARC でのブート ストラップ手続き	

boot(1M)

ネットワーク起動は2段階で行われます。まず、クライアントが、二次起動プログラムの読み込みに必要な IP アドレスとその他のすべてのパラメータを取得します。続いて、二次起動プログラムが、UNIX カーネルを読み込みます。

IP アドレスは、PROM で使用可能な機能と PROM の構成に応じて、RARP、DHCP、手動構成のいずれかの方法を使って取得できます。sun4u カーネルアーキテクチャのマシンには、DHCP 対応の PROM が搭載されています。

次に、2種類のネットワーク起動方式を指定する boot コマンド構文を示します。

```
boot net:rarp
boot net:dhcp
```

次のコマンドのように、rarp または dhcp 指示子を指定しないと、net が別名になっているネットワークインタフェースを介して、デフォルトのネットワーク起動方式が呼び出されます。

```
boot net
```

以降では、RARP/bootparams を使用するネットワーク起動のイベントシーケンスについて説明します。そのあとで、DHCP を用いたシーケンスについて説明します。

RARP/bootparams を使ってネットワークから起動する場合、PROM はまず、応答を受信するまで逆 ARP 要求をブロードキャストします。応答を受信すると、TFTP 要求をブロードキャストして inetboot の最初のブロックを取得します。続いて、その最初のブロック要求に一番先に応答したサーバーに後続の要求が送信されます。読み込みの完了後、inetboot も同様に、逆 ARP を使って IP アドレスを取得した後、bootparams RPC 呼び出し (bootparams(4) を参照) をブロードキャストすることで、構成情報とルートファイルシステムを検索します。続いて、inetboot は NFS 経由でカーネルを読み込んだ後、そのカーネルに制御を移します。

DHCP を使用してネットワークから起動する場合、PROM はハードウェアアドレスおよびカーネルアーキテクチャをブロードキャストし、IP アドレス、起動パラメータ、およびネットワーク構成情報を要求します。(可能性のある複数のサーバーの中から) DHCP サーバーが応答し選択されると、そのサーバーがクライアントに IP アドレスおよびクライアントの起動に必要な他のすべての情報を送信します。この情報を受信したクライアントの PROM は、読み込むファイルの名前を調べ、そのファイル名が HTTP URL であるかどうかにより、2種類の動作のいずれかを実行します。ファイル名が HTTP URL でなかった場合、PROM は、inetboot をダウンロードし、それをメモリー内に読み込んで実行します。すると、inetboot はカーネルを呼び出し、さらにそのカーネルが必要なファイルを読み込んでから inetboot を解放します。その後、起動スクリプトが DHCP エージェント (dhcpagent(1M) を参照) を起動します。以降の DHCP アクティビティは、その DHCP エージェントによって実行されます。

読み込むファイルが HTTP URL の場合、PROM は、その参照ファイルを HTTP を使って読み込みます。クライアントが HMAC SHA-1 鍵を使って構成されていた場合、クライアントは読み込んだファイルの完全性を検査した後で、そのファイルを実行します。そのファイルは wanboot バイナリであるとみなされます。起動された wanboot は、処理を継続できるだけの情報が揃っているかどうかを判断します。必要な情報が不足していた場合は、適切なエラーを出力して処理を終了するか、コマンドインタプリタを起動して必要な構成情報の入力をユーザーに求めます。必要な情報を

取得した wanboot は、HTTP 経由で起動ファイルシステムをメモリー内に読み込みます。クライアント上に暗号鍵がインストール済みである場合、wanboot は、起動ファイルシステムイメージと関連ハッシュを復号化した後、そのハッシュを検証します (暗号鍵は存在するがハッシュ鍵が存在しない、という状況はエラーになります)。起動ファイルシステムには、wanboot が正しい時刻の設定とルートファイルシステムの取得を実行するのに必要な各種構成データが含まれています。

wanboot は、起動ファイルシステムを調べることで、HTTP、HTTPS のいずれを使用すべきかを判断します。HTTP を使用すべきであることが判明した場合でかつ、クライアントが HMAC SHA-1 鍵を使って構成されていた場合、wanboot はルートファイルシステムの完全性検査を実行します。ルートファイルシステムのメモリー内への読み込み (および必要に応じて完全性検査の実行) が完了すると、wanboot はそこから UNIX を読み込みおよび実行します。なお、wanboot.conf(4) ファイルで boot_logger URL が指定されていた場合、wanboot は処理状況に関するログを定期的に記録します。

すべての PROM が URL を処理できるわけではありません。クライアントにその機能が備わっているかどうかを判断するには、list-security-keys OBP コマンド (monitor(1M) を参照) を使用します。

現在のところ、x86 プラットフォーム上では WAN 起動は利用できません。

wanboot コマンド行

クライアントプログラムが wanboot である場合、クライアントプログラムは、次の形式の client-program-args を受け付けます。

```
boot ... -o opt1[,opt2[,...]]
```

ここで、各オプションに指定できるアクションは、次のとおりです。

dhcp

DHCP 経由で構成パラメタを取得するように wanboot に指示します。

prompt

コマンドインタプリタを起動するように wanboot に指示します。

<cmd>

下記のインタプリタコマンドのいずれか。

または、下記のインタプリタパラメタ名を使用した代入文。

wanboot コマンドインタプリタ

起動時に client-program-args として "-o prompt" を指定すると、wanboot コマンドインタプリタが起動されます。単一のコマンドまたは代入文、もしくは、コマンドで区切られた一連のコマンドまたは代入文が入力可能です。利用可能な構成パラメタは、次のとおりです。

host-ip

クライアントの IP アドレス (ドット区切り 10 進表記)。

boot(1M)

router-ip
デフォルトのルーターの IP アドレス (ドット区切り 10 進表記)。

subnet-mask
サブネットマスク (ドット区切り 10 進表記)。

client-id
DHCP クライアント識別子 (引用符付き ASCII 文字列または 16 進 ASCII)。

hostname
DHCP トランザクションで要求するホスト名 (ASCII)。

http-proxy
HTTP プロキシサーバー指定 (IPADDR[:PORT])。

利用可能な鍵名は、次のとおりです。

3des
トリプル DES 暗号鍵 (48 個の 16 進 ASCII 文字)。

aes
AES 暗号鍵 (32 個の 16 進 ASCII 文字)。

sha1
HMAC SHA-1 署名鍵 (40 個の 16 進 ASCII 文字)。

最後に、WAN 起動 CGI の URL を参照する方法を示します。

bootserver
WAN 起動 CGI の URL (OBP の `file` パラメタと同等)。

インタプリタが受け付けるコマンドは、次のとおりです。

help
利用可能なコマンドの簡単な説明文を出力します。

var=val
`val` を `var` に代入します。ここで、`var` は、構成パラメタ名、鍵名、`bootserver` のいずれかです。

var=
パラメタ `var` の設定を解除します。

list
すべてのパラメタとその設定値を一覧表示します (OBP 経由で取得された鍵の値は表示されません)。

prompt
未設定パラメタに対する値の入力を、ユーザーに求めます。各パラメタの名前と現在値 (もしあれば) が出力されます。ユーザーは、**Return** キーを押して現在値を受け入れることもできますし、新しい値を入力することもできます。

go
すべての値の入力を完了したら、インタプリタを終了し起動処理を継続します。

exit

起動インタプリタを終了し、OBP の ok プロンプトに戻ります。

上記の代入文やコマンドは、コマンド行の `-o` オプションの一部として、いくつでも指定できますが、boot コマンドの引数の長さには 128 バイトの OBP 制限があります。たとえば、「`-o list,go`」とすると、パラメタの現在値 (デフォルト値) が一覧表示された後、起動処理が継続します。

ディスクからの起動

ディスク (またはディスク同様のデバイス) から起動する場合、ブートストラップ処理は、一次起動および二次起動という概念的に異なる 2 種類のフェーズからなります。一次起動フェーズでは、PROM が、起動デバイスとして選択されたディスクパーティションのブロック 1 ~ 15 から主起動ブロックを読み込みます。

スタンドアロンプログラムのパス名が相対パス名の場合 (スラッシュで始まらない場合)、第 2 レベルの起動において、プラットフォームに依存する検索パスでスタンドアロンプログラムが検索されます。このパスには必ず、`/platform/platform-name` が含まれています。多くの SPARC プラットフォームでは、次にプラットフォーム固有のパスエントリ `/platform/hardware-class-name` が検索されます。filesystem(5) のマニュアルページを参照してください。絶対パス名の場合、boot は指定されたパスを使用します。その後、boot プログラムは該当するアドレスからスタンドアロンプログラムを読み込み、制御を渡します。

ファイル名がコマンド行で、または `boot-file` NVRAM 変数などで指定されていない場合、boot はシステムにインストールされているソフトウェア、ハードウェアおよびファームウェアの能力、さらにユーザーが設定可能なポリシーファイル (「ファイル」を参照) に基づいて、読み込むべきデフォルトのファイルを選択します。

OpenBoot PROM boot コマンドの動作

OpenBoot boot コマンドは、次の形式の引数をとります。

```
ok boot [device-specifier] [arguments]
```

デフォルトの boot コマンドに引数はありません。

```
ok boot
```

boot コマンド行で `device-specifier` を指定しない場合、OpenBoot は通常、`boot-device` 変数または `diag-device` NVRAM 変数を使用します。オプションの `arguments` をコマンド行で指定しないと、OpenBoot は通常、`boot-file` または `diag-file` NVRAM 変数をデフォルトの boot 引数として使用します (システムが診断モードの場合、`boot-device` および `boot-file` の代わりに、`diag-device` と `diag-file` が使用されます)。

`arguments` には複数の文字列を指定できます。すべての `argument` 文字列は二次起動プログラムに渡され、OpenBoot では解釈されません。

boot(1M)

boot コマンド行に *arguments* を指定した場合、*boot-file* または *diag-file* NVRAM 変数のどちらも使用されません。NVRAM 変数の内容がコマンド行の引数とマージされることもありません。次のコマンドでは、*boot-file* および *diag-file* の設定値は無視され、文字列 *-s* は *arguments* として解釈されます。この場合、boot は *boot-file* または *diag-file* の内容を使用しません。

```
ok boot -s
```

以前の PROM におけるコマンド例を示します。

```
ok boot net
```

このコマンドでは引数が指定されていないため、代わりに、*boot-file* または *diag-file* の設定値 (設定されている場合) が、boot に渡すデフォルトのファイル名および引数として使用されます。ほとんどの場合、システムタイプ、システムのハードウェアとファームウェア、およびルートファイルシステムにインストールされているソフトウェアに基づいて boot コマンドに適切なデフォルト値を選択させるのが、最善の方法です。boot コマンドのポリシーを追加するためにポリシーファイルを変更してもかまいませんが、*boot-file* または *diag-file* を変更すると、状況によっては予期せぬ結果を招くおそれがあります。

これは、大部分の OpenBoot 2.x および 3.x ベースのシステムにおける一般的な動作です。ただし、プラットフォームによっては動作が異なる可能性もあります。

別のコマンド例を示します。

```
ok boot cdrom
```

このコマンドも通常は、引数を指定しません。したがって、*boot-file* に 64 ビットカーネルのファイル名が設定されている環境で、「boot cdrom」と入力してインストール CD を起動しようとしても、インストール CD に 32 ビットカーネルしか含まれていなければ、その起動処理は失敗します。

使用される boot コマンドの形式によっては、*boot-file* または *diag-file* の内容が無視されるので、プロダクションシステムで *boot-file* に依存することは一般に推奨しません。OS のポリシーを変更するには、ポリシーファイルを変更します。プロダクションシステムに 32 ビットのパッケージと 64 ビットのパッケージの両方がインストールされている場合は例外ですが、プロダクションシステムでは 32 ビット OS を使用する必要があります。

ローカル (CD 上) に存在する wanboot コピーから WAN 起動を実行するには、次のコマンドを使用する必要があります。

```
ok boot cdrom -F wanboot - install
```

最近の PROM に含まれるネットワーク起動サポートパッケージは、次の構文をサポートするように強化されており、引数を処理できるようになっています。

```
[protocol,] [key=value,]*
```

すべての引数は省略可能であり、指定する順序にも制限はありません。ただし、リストの末尾でない限り、引数の後にはコンマが必要です。ここで指定した引数は、すべてのデフォルト値よりも優先されるほか、DHCP を使って起動する場合には、それらの引数に対応する DHCP サーバーから提供される構成情報よりも優先されます。

上記の *protocol* には、使用するアドレス検索プロトコルを指定します。

key=value 属性ペアには、以下の構成パラメタを指定します。

tftp-server

TFTP サーバーの IP アドレス。

file

TFTP を使ってダウンロードするファイルの名前、または WAN 起動の URL。

host-ip

クライアントの IP アドレス (ドット区切り 10 進表記)。

router-ip

デフォルトのルーターの IP アドレス。

subnet-mask

サブネットマスク (ドット区切り 10 進表記)。

client-id

DHCP クライアント識別子。

hostname

DHCP トランザクションで使用するホスト名。

http-proxy

HTTP プロキシサーバー指定 (IPADDR[:PORT])。

tftp-retries

TFTP の最大リトライ回数。

dhcp-retries

DHCP の最大リトライ回数。

ネットワーク起動サポートパッケージによって処理される一連の引数は、次のいずれかの方法で指定します。

- パッケージの *open* メソッドに渡される引数として
- NVRAM 変数 *network-boot-arguments* 内に指定する引数として

network-boot-arguments 内に指定された引数は、パッケージの *open* メソッドに引数が 1 つも渡されなかった場合にのみ処理されます。

引数の値

protocol には、使用するアドレス検索プロトコルを指定します。rarp、dhcp のいずれかを指定できます。

boot(1M)

このドキュメントで定められた新しい構文とスタイルを使って他の構成パラメタが指定されていた場合、*protocol* パラメタが指定されていない場合は、手動の構成を意味します。

他の構成パラメタが1つも指定されていない場合や、他の構成パラメタが現在サポートされている位置に基づくパラメタ構文を使って指定されていた場合、*protocol* パラメタを指定しなければ、ネットワーク起動サポートパッケージは、プラットフォーム固有のデフォルトのアドレス検索プロトコルを使用することになります。

手動構成を行う場合、IP アドレス、起動ファイル名、起動ファイルイメージの提供元サーバーのアドレスの各情報を、クライアントに提供する必要があります。また、ネットワークの構成によっては、*subnet-mask* と *router-ip* も指定する必要があります。

protocol 引数が指定されなかった場合、ネットワーク起動サポートパッケージは、プラットフォーム固有のデフォルトのアドレス検索プロトコルを使用します。

tftp-server は、TFTP を使用する場合、ダウンロードするファイルを提供する TFTP サーバーの IP アドレス (標準 IPv4 のドット区切り 10 進表記) です。

DHCP を使用する場合、その値は、DHCP 応答に指定された TFTP サーバーの値よりも優先されます。

サーバーが引数として指定された場合と DHCP 応答内に指定された場合、TFTP RRQ はサーバーにユニキャストされます。それ以外の場合、TFTP RRQ はブロードキャストされます。

file には、TFTP を使用する場合は TFTP サーバーから読み込むファイルを指定し、HTTP を使用する場合は URL を指定します。ファイル名が URL である場合、すなわち、ファイル名が *http:* で始まる場合 (大文字小文字の区別あり)、HTTP が使用されます。

RARP と TFTP を使用する場合、このドキュメントで前述したように、デフォルトのファイル名はクライアントの IP アドレスの ASCII 16 進表記になります。

DHCP を使用する場合、この引数は、DHCP 応答に指定された起動ファイル名よりも優先されます。

DHCP と TFTP を使用する場合、デフォルトのファイル名は、*root* ノードの *name* プロパティから自動生成されます。その際、コンマ (,) はピリオド (.) に置き換えられます。

ファイル名をコマンド行から指定する場合、ファイル名にスラッシュ (/) を含むことはできません。

URL の形式については、RFC 2396 で説明されています。HTTP サーバーは、IP アドレス (標準 IPv4 のドット区切り 10 進表記) として指定する必要があります。ポート番号 (省略可能) は、10 進数として指定します。ポートを指定しなかった場合、ポート 80 (10 進) が使用されます。

boot(1M)

指定された URL は「安全にエンコードされている」必要があります。というのも、パッケージは、指定された URL にエスケープエンコーディングを適用しないからです。コンマを含む URL は、引用符付きの文字列として指定する必要があります。それ以外の URL は、必ずしも引用符で囲む必要はありません。

host-ip には、クライアントすなわち起動対象システムの、IP アドレス (標準 IPv4 のドット区切り 10 進表記) を指定します。アドレス検索プロトコルとして RARP を使用している場合にこの引数を指定すると、RARP を使用する必要がなくなります。

DHCP を使用している場合に host-ip 引数を指定すると、クライアントは、RFC 2131 の「Externally Configured Network Address」で規定されている手順に従うようになります。

router-ip には、直接接続されたネットワーク上にあるルーターの IP アドレス (標準 IPv4 のドット区切り 10 進表記) を指定します。このルーターは、ネットワーク通信の最初の接続先として使用されます。この引数を指定した場合、そこで指定されたルーターは、DHCP 応答に指定された推奨ルーターよりも優先されます。

subnet-mask (標準 IPv4 のドット区切り 10 進表記で指定) は、クライアントが存在するネットワークのサブネットマスクです。サブネットマスクが (この引数または DHCP 応答を通じて) 指定されなかった場合、起動対象クライアントに割り当てられたアドレスに対するネットワーククラス (Class A、B、C のいずれか) に適するデフォルトマスクが使用されます。

client-id には、クライアントの一意に決まる識別子を指定します。DHCP クライアント識別子は、この値に基づいて生成されます。クライアント識別子の指定方法には、次の 2 つがあります。

- 識別子の ASCII 16 進表現
- 引用符付きの文字列

したがって、「client-id="openboot"」と「client-id=6f70656e626f6f74」はどちらも、DHCP クライアント識別子 006F70656E626F6F74 を表わします。

コマンド行から指定される識別子には、スラッシュ (/) や空白を含めることはできません。

DHCP クライアント識別子の最大長は、32 バイト (ASCII 16 進形式を使用する場合は 32 バイトを表す 64 文字) です。後者の形式を使用する場合、識別子を構成する文字数は偶数でなければなりません。有効な文字は、0 ~ 9、a ~ f、A ~ F です。

クライアントが正しく認識されるためには、クライアントが接続されているサブネット上で、クライアント識別子が一意に決まる必要があります。この要件を満足する識別子を選択するのは、システム管理者の役割です。

コマンド行で指定されたクライアント識別子は、任意の DHCP メカニズムを使って指定された識別子よりも優先されます。

boot(1M)

`hostname` (文字列として指定) には、DHCP トランザクション内で使用されるホスト名を指定します。この名前は、ローカルドメイン名で修飾してもしなくてもかまいません。ホスト名の最大長は、255 文字です。

注 - `hostname` パラメータは、クライアントが、希望するホスト名を DHCP サーバーに提供することが必要なサービス環境で使用できます。クライアントが、希望するホスト名を DHCP サーバーに提供すると、DHCP サーバーは、そのホスト名とクライアントに割り当てられた IP アドレスを、DNS を使って登録します。

`http-proxy` は、次のようなホストの標準記法を使って指定します。

```
host [":" port]
```

ここで、`host` には IP アドレス (標準 IPv4 のドット区切り 10 進表記) を指定し、`port` (省略可能) には 10 進数を指定します。`port` を指定しなかった場合、ポート 8080 (10 進) が使用されます。

`tftp-retries` は、TFTP 処理が失敗したとみなされるまでの、最大リトライ回数 (10 進で指定) です。デフォルトのリトライ回数は、無限回です。

`dhcp-retries` は、DHCP 処理が失敗したとみなされるまでの、最大リトライ回数 (10 進で指定) です。デフォルトのリトライ回数は、無限回です。

x86 でのブートストラップ手続き

x86 ベースのシステムでは、ブートストラップ処理は一次起動および二次起動という概念的に異なる 2 つのフェーズからなります。一次起動はシステムボード上の BIOS ROM および周辺ボード上の BIOS ROM 拡張機能で実装されています。x86 でのブートストラップ処理の特徴は、搭載されている周辺デバイスを制御するとソフトウェア割り込みによって入出力サービスを提供する機能に分けられます。起動処理はフロッピーディスク、ハードディスク、または CD-ROM から最初の物理セクターを読み込むことにより、開始されます。あるいは、システムまたはネットワークアダプタの BIOS がサポートしている場合には、ネットワーク起動サーバーからブートストラッププログラムを読み込むことで、開始されます。一次起動は、x86 リアルモードコードで実装されています。

二次起動は一次起動によって読み込まれます。二次起動は 32 ビットのページング保護モードで実装されています。二次起動はまた、x86 リアルモードで作成された周辺デバイス固有の BIOS 拡張機能を読み込んで使用します。二次起動は `boot.bin` と呼ばれ、ハードディスクまたは CD 上の UFS ファイルシステムから、または NFS プロトコルを使用して、LAN から読み込まれ、起動します。

二次起動は、システムに搭載されているデバイスを判別 (ユーザーの関与が必要) する Configuration Assistant プログラムを実行します。また、二次起動は、起動処理を制御する `/etc/bootrc` 内のスクリプトを読み込みます。このファイルには、`boot interpreter` コマンドが含まれています。これらのコマンドについては以降で説明しますが、これらのコマンドを修正することによって、デフォルト値を変更したり特定のマシンに合わせて調整したりできます。

標準の /etc/bootrc スクリプトは、オプションを指定して起動する場合は b、対話方式でインタプリタを呼び出す場合は i を入力するように、ユーザーに求めます。文字を入力せずに ENTER キーを押すと、デフォルトのカーネルが起動されます。それ以外の応答はすべてエラーとみなされ、スクリプトが再起動されます。

カーネルが読み込まれると、カーネルがオペレーティングシステムを起動し、必要なモジュールを読み込み、必要なファイルシステムをマウントして (vfstab(4) のマニュアルページを参照)、/sbin/init を実行し、システムを /etc/inittab で指定された initdefault 状態にします。inittab(4) のマニュアルページを参照してください。

オプション

SPARC

<i>OBP names</i>	OpenBoot PROM 指定を行います。たとえば、Desktop SPARC ベースのシステム上で /sbus/esp@0,800000/sd@3,0:a を指定した場合は、スロット 0 の esp ホストアダプタで、SCSI バス上の lun0、ターゲット 3 の SCSI ディスク (sd) を意味します。
<i>file</i>	起動するスタンドアロンプログラムの名前。boot コマンド行または <i>boot-file</i> NVRAM 変数でファイル名を明示的に指定しないと、boot は適切なデフォルトファイル名を選択します。ほとんどのシステムでは、デフォルトのファイル名は 32 ビットカーネルです。32 ビットカーネルと 64 ビットカーネルを両方ともサポートできるシステムでは、32 ビットカーネルより 64 ビットカーネルが優先的に選択されます。boot はシステムにインストールされているソフトウェア、ハードウェアおよびファームウェアの能力、さらにユーザーが設定可能なポリシーファイルに基づいて、boot の実行に適したデフォルトの <i>file</i> を選択します。
-a	boot プログラムはこのフラグを問い合わせと解釈し、スタンドアロンプログラムの名前を要求します。-a フラグはその後、スタンドアロンプログラムに渡されます。
-f	Autoclient システムを起動する場合、このフラグによって boot プログラムがクライアントのローカルキャッシュを迂回して、ネットワーク経由でクライアントのファイルサーバーからすべてのファイルを読み込むようにします。Autoclient 以外のシステムでは、このフラグは無視されます。-f フラグはその後、スタンドアロンプログラムに渡されます。
-v	詳細なデバッグ情報を表示します
-D <i>default-file</i>	<i>default-file</i> を明示的に指定します。一部のシステムでは、なにも指定されていない場合、boot は動的デ

boot(1M)

		<p>フォルトファイルを選択します。このオプションを使用すると、<i>default-file</i> を明示的に指定できます。<i>kadb(1M)</i> を起動する場合に便利です。これは、<i>kadb</i> がデフォルトで、<i>boot</i> プログラムによってエクスポートされたデフォルトファイルを読み込むためです。</p>
	<i>boot-flags</i>	<p><i>boot</i> プログラムはすべての <i>boot-flags</i> を <i>file</i> に渡しません。これらのフラグは、<i>boot</i> では解釈されません。デフォルトのスタンドアロンプログラムで使用できるオプションについては <i>kernel(1M)</i> および <i>kadb(1M)</i> のマニュアルページを参照してください。</p>
	<i>client-program-args</i>	<p><i>boot</i> プログラムはすべての <i>client-program-args</i> を <i>file</i> に渡します。これらの引数は、<i>boot</i> では解釈されません。</p>
x86	<i>file</i>	<p>起動するスタンドアロンプログラムの名前。デフォルトではルートパーティションから <i>/platform/platform-name/kernel/unix</i> を起動しますが、コマンド行に他のプログラムを指定することもできます。</p>
	<i>-f</i>	<p>Autoclient システムを起動する場合、このフラグによって <i>boot</i> プログラムがクライアントのローカルキャッシュを迂回して、ネットワーク経由でクライアントのファイルサーバーからすべてのファイルを読み込むようにします。Autoclient 以外のシステムでは、このフラグは無視されます。 <i>-f</i> フラグはその後、スタンドアロンプログラムに渡されます。</p>
	<i>boot-args</i>	<p><i>boot</i> プログラムはすべての <i>boot-args</i> を <i>file</i> に渡します。これらの引数は、<i>boot</i> では解釈されません。カーネルで使用できるオプションについては、<i>kernel(1M)</i> および <i>kadb(1M)</i> のマニュアルページを参照してください。</p>
	<i>-D default-file</i>	<p><i>default-file</i> を明示的に指定します。一部のシステムでは、なにも指定されていない場合、<i>boot</i> は動的デフォルトファイルを選択します。このオプションを使用すると、<i>default-file</i> を明示的に指定できます。<i>kadb(1M)</i> を起動する場合に便利です。これは、<i>kadb</i> がデフォルトで、<i>boot</i> プログラムによってエクスポートされたデフォルトファイルを読み込むためです。</p>
x86 での起動シーケンスの詳細		<p>PC 互換マシンの電源を投入すると、BIOS ROM のシステムファームウェアが電源投入時自己診断テスト (POST) を実行し、周辺ボード上の ROM の BIOS 拡張機能を実行し、ソフトウェア割り込み INT 19h のブートストラップを起動します。INT 19h のハンドラは通常、標準の PC 互換起動処理を実行します。その場合、1 番目のフロッピーディスクドライブから最初の物理セクターを読み取ります。読み取れない場合は、1 番目のハードディスクから読み取ります。次にプロセッサは、メモリー内でこのセクターイメージの先頭バイトにジャンプします。</p>

x86 での一次起動

フロッピーディスクの最初のセクターには、マスター起動ブロックがあり、この起動ブロックがブートローダー strap.com のイメージを読み込み、さらに strap.com が二次起動プログラム boot.bin を読み込みます。CD-ROM から起動する場合も、同様なシーケンスが実行されますが、マスター起動ブロックの位置および内容は、El Torito 仕様で規定されています。El Torito 準拠の起動でも、strap.com が読み込まれ、次に boot.bin が読み込まれます。

ハードディスクの最初のセクターには、マスター起動ブロックがあり、そのブロックにはマスター起動プログラムと FDISK テーブルが含まれており、PC プログラムがそのテーブルを管理しています。マスター起動プログラムは、FDISK テーブルでアクティブパーティションを見つけ、その最初のセクターを読み込み、メモリー内のセクターイメージの先頭バイトにジャンプします。標準の PC 互換ハードディスクでの起動処理はこれで完了します。

Solaris ソフトウェア用の x86 FDISK パーティションは、1 シリンダの起動スライスから始まり、最初のセクターにパーティション起動プログラム (pboot) が含まれています。第 2 および第 3 セクターにはそれぞれ、標準 Solaris ディスクラベルおよびボリューム構成テーブル (VTOC) があり、第 4 およびそれ以降のセクターには bootblk プログラムが含まれています。Solaris ソフトウェアの FDISK パーティションがアクティブパーティションの場合、マスター起動プログラム (mboot) は第 1 セクターのパーティション起動プログラムをメモリーに読み込んでそこにジャンプします。次に、bootblk プログラムをメモリーに読み込んでそこにジャンプします。ドライブに FDISK パーティションが複数ある場合、ユーザーはアクティブパーティションのタイプに関係なく、別のパーティションを再起動できます。

(アクティブパーティションのタイプに応じて) bootblk または strap.com は、Solaris ルートスライスのファイルシステムから boot.bin を読み込み、メモリー内でその先頭バイトにジャンプします。

ネットワークから起動する場合、起動フロッピーまたは Intel の Preboot eXecution Environment (PXE) 規格のどちらかを使用するか選択できます。起動フロッピーを使用してネットワークから起動する場合は、起動特性を編集し、net-config-strategy の設定値を変更することによって、必要なネットワーク構成ストラテジを選択できます。デフォルトでは、net-config-strategy は rarp に設定されています。設定できるのは rarp または dhcp のいずれかです。PXE を使用してネットワークから起動する場合、システムまたはネットワークアダプタの BIOS が DHCP を使用して、起動サーバー上のネットワークストラップストラッププログラム (NBP) を見つけ、Trivial File Transfer Protocol (TFTP) を使用してそのプログラムを読み込みます。BIOS はメモリー内の NBP イメージの先頭バイトにジャンプすることによって NBP を実行します。NBP は DHCP を使用して、起動サーバー上で第 2 ブートストラップを見つけて、TFTP を使用して読み込んで実行します。

x86 での二次起動

二次起動プログラムの boot.bin は、プロセッサを 32 ビットのページング保護モードに切り替え、限定的なマシン初期化処理を実行します。次に、auto-boot? 変数 (eeprom(1M) のマニュアルページを参照) の状態に応じて、システムを自動的に起動する Configuration Assistant プログラムを実行するか、または使用できる起動デバイスの一覧を表示します。

boot(1M)

	<p>ディスクターゲットデバイス (CDROM ドライブを含む) に UFS ファイルシステムが組み込まれていることが前提です。ネットワーク装置は、DHCP または逆アドレス解決プロトコル (RARP) のどちらか一方および bootparams RPC を使用して、マシンの IP アドレスおよびルートファイルシステムを提供するサーバーを検出するように構成できます。その後、ルートファイルシステムが NFS を使用してマウントされます。ルートのマウントが正常に完了すると、boot.bin がコマンドインタプリタを起動し、インタプリタが /etc/bootrc を解釈します。</p>
x86 用の二次起動プログラミング言語	<p>x86 ベースのシステムではさまざまなハードウェアをサポートしなければならないため、起動処理に最大限の柔軟性が求められます。この柔軟性は、二次起動プログラム可能にすることによって、ある程度達成されます。二次起動プログラムには、sh や csh のような単純なコマンド言語を受け付けるインタプリタが組み込まれています。主な違いはパイプライン、ループ、標準出力、出力の転送がサポートされていない点です。</p>
x86 の字句構造	<p>起動インタプリタは、入力行を空白文字およびタブで区切られたワード単位に分割します。メタキャラクタはドル記号 (\$)、単一引用符 (')、二重引用符 (")、番号記号 (#)、復帰改行、およびバックスラッシュ (\) です。前にバックスラッシュを指定すると、メタキャラクタが持つ特殊な意味を打ち消すことができます。復帰改行の前にバックスラッシュを指定すると、空白文字として扱われます。番号記号はコメント行の始まりを意味し、次に復帰改行が現れるまでコメントとして扱われます。</p> <p>単一引用符または二重引用符のペアで囲まれた文字列は、1 つのワードの全体または部分を形成します。引用符で囲まれた文字列の中の空白および復帰改行記号は、ワードの一部になります。引用符で囲まれた文字列の中で、文字に引用符をつける場合は、その前にバックスラッシュを使用します。単一引用符の前にバックスラッシュを指定することによって、単一引用符で囲まれた文字列の中で引用符を使用できます。バックスラッシュを 2 つ続けて指定すると、単一のバックスラッシュになります。また、復帰改行の前にバックスラッシュを指定すると、文字列の中で復帰改行を使用できます。</p>
x86 の変数	<p>boot には一連の変数があり、各変数には文字列値が割り当てられます。変数名の先頭文字は、英字でなければなりません。後続文字には英字、数字、または下線を使用できます。set コマンドを使用すると、変数の作成や、変数への値の割り当て、変数値の表示が行えます。変数を削除するには unset コマンドを使用します。</p> <p>インタプリタが先行するバックスラッシュのないドル記号を検出すると、変数置換が実行されます。ドル記号に続く変数名が変数の値に置き換えられ、値の先頭から構文解析が続けられます。変数置換は、二重引用符で囲まれた文字列では実行されますが、単一引用符で囲まれた文字列では実行されません。変数名を中括弧で囲むと、後続文字と分離できます。</p>
x86 のコマンド	<p>コマンドは、復帰改行文字で終了する一連のワードです。最初のワードはコマンド名で、以降のワードはコマンドに対する引数です。コマンドはすべて組み込みコマンドです。スタンドアロンプログラムは run コマンドで実行されます。</p>
x86 コマンドの条件付き実行	<p>コマンドを if、elseif、else、および endif コマンドで囲むことによって、条件付きの実行が可能になります。</p>

```

if expr1
. . .
elseif expr2
. . .
elseif expr3
. . .
else
. . .
endifif

```

endifif ブロックは他の if ブロックの中を含めることができます。

x86 の式

set、if、および elseif コマンドは、C プログラミング言語の構文および意味論を使用して算術式を評価します。||、&&、|、^、&、==、!=、<、>、<=、>=、>>、<<、+、-、*、/、%、~、および ! という演算子を (、) およびコンマと同様に使用できます。符号付き 32 ビット整数演算が実行されます。

式はコマンド行全体が形成された後で構文解析されます。式の中の各トークンは、別の引数ワードにし、空白文字でコマンド行上のすべてのトークンを区切る必要があります。

オペランドワードは、それに対する算術演算が実行される前に、文字列から符号付き 32 ビット整数値に変換されます。オプションの先行符号のうしろに先行する 0 があると 8 進数変換が行われ、先行する 0x または 0X がある場合は、16 進数変換が行われます。それ以外の場合は、10 進数変換が行われます。文字列が有効な整数にならない場合は、ゼロに変換されます。

文字列を操作できるように、いくつかの組み込み関数が用意されています。組み込み関数名はドットで始まります。これらの関数に対する文字列引数は、整数に変換されません。たとえば、演算子 - が文字列として扱われるようにするには、バックスラッシュを前につけ、そのバックスラッシュをもう 1 つのバックスラッシュでクォートする必要があります。また、ヌル文字列によって空白文字の引数が生成されるため、式で構文エラーが発生することがあるので注意してください。以下に例を示します。

```
if .strneq ( ${usrarg}x , \- , 1 )
```

これにより、変数 *usrarg* が - で始まるかどうかを、ヌルになる可能性がある場合でも問題なくテストできます。

x86 での入出力

起動インタプリタはシステムコンソールまたは 1 つ以上のファイルから入力を読み込みます。ソースコマンドは、インタプリタがファイルをメモリーに読み込み、ファイルの構文解析を開始するようにします。また、コンソールコマンドは、インタプリタがシステムコンソールからの入力を受け付けるようにします。EOF に到達すると、インタプリタはその前の入力ソースの構文解析を再開します。コンソール行の先頭に CTRL-D を入力すると、EOF として扱われます。

echo コマンドは引数をディスプレイに出力します。read コマンドはシステムコンソールを読み込み、引数変数にワード値を割り当てます。

x86 でのデバッグ

verbose コマンドは、詳細表示モードのオン / オフを切り替えます。詳細表示モードでは、インタプリタは現在のソースファイルの行を表示し、変数置換後に実際に実行されたコマンドを表示します。

boot(1M)

	<p>singlestep コマンドは、ステップ実行モードのオン / オフを切り替えます。ステップ実行モードでは、インタプリタは次のコマンドを処理する前に <code>step ?</code> を表示し、キーボード入力を待ちます。そのキーボード入力は破棄されます。ENTER を押すと、処理を進められるのでこれにより、詳細表示モードでは実行速度を遅くすることができます。</p>
x86 での初期化	<p>boot がインタプリタを最初に呼び出したとき、インタプリタはコンパイル時に設定された初期化文字列の実行を開始します。この文字列は通常、ルートファイルシステム内の起動スクリプトを実行する <code>source /etc/bootrc</code> からなります。</p>
x86 でのスタンドアロンプログラムとの通信	<p>boot は run コマンドに対する引数を通じて、スタンドアロンプログラムに情報を渡します。スタンドアロンプログラムは <code>var_ops()</code> 起動サービス関数を使用して、起動インタプリタ変数を設定することにより、boot に情報を返すことができます。また、<code>setprop()</code> 起動サービス関数を使用してカーネルに情報を渡すこともできます。whoami 属性はスタンドアロンプログラムの名前に設定します。</p>
x86 の組み込みコマンド	<p>console CTRL-D を検出するまでコンソールからの入力を解釈します。</p> <p>echo <i>arg1</i> ... 空白文字で区切られ、復帰改行で終わる引数を表示します。</p> <p>echo -n <i>arg1</i> ... 空白文字で区切られているが復帰改行で終わらない引数を表示します。</p> <p>getprop <i>propname varname</i> 属性 <i>propname</i> の値を変数 <i>varname</i> に割り当てます。長さゼロの属性値からはヌル文字列が生成されます。属性が存在しない場合、変数は設定されません。</p> <p>getproplen <i>propname varname</i> 変数 <i>varname</i> に属性 <i>propname</i> の値の長さを 16 進数で割り当てます。属性値の長さには終了を示すヌルが含まれます。属性が存在しない場合、変数は 0xFFFFFFFF (-1) に設定されます。</p> <p>if <i>expr</i> 式 <i>expr</i> が真の場合、次の <code>elseif</code>、<code>else</code>、または <code>endif</code> までの命令を実行します。<i>expr</i> が偽の場合は、命令を実行しません。</p> <p>elseif <i>expr</i> それまでの <code>if</code> コマンドおよび <code>elseif</code> コマンドがすべてエラーとなり、なおかつ <i>expr</i> が真の場合、次の <code>elseif</code>、<code>else</code>、または <code>endif</code> までの命令を実行します。それ以外の場合は、命令を実行しません。</p> <p>else それまでの <code>if</code> コマンドおよび <code>elseif</code> コマンドがすべてエラーになった場合、次の <code>elseif</code>、<code>else</code>、または <code>endif</code> までの命令を実行します。それ以外の場合は、命令を実行しません。</p> <p>endif 囲んでいるブロックの実行モードに戻ります。</p> <p>help 利用可能なすべての起動シェルコマンドの要約を示すヘルプ画面を表示します。</p>

`read name1 ...`
 コンソールから行を読み取り、ワードに分割し、それらを値として変数 *name1* (以下同様) に割り当てます。

`readt time ...`
`read` と同じですが、`time` で指定された秒数が経過したら、タイムアウトします。

`run name arg1 ...`
 スタンドアロンプログラム *name* を読み込んで制御を渡し、*arg1* とそれに続く引数を渡します。

`set`
 現在の変数およびその値をすべて表示します。

`set name`
 変数 *name* の値をヌル文字列に設定します。

`set name word`
 変数 *name* の値を *word* に設定します。

`set name expr`
 変数 *name* の値を *expr* の値に設定します。*expr* は複数のワードで構成されていなければなりません。値は符号なしの 16 進数で符号化されます。たとえば、-1 は 0xFFFFFFFF で表されます。

`setcolor`
 テキストモードの表示属性を設定します。使用できる色は *black*、*blue*、*green*、*cyan*、*red*、*magenta*、*brown*、*white*、*gray*、*lt_blue*、*lt_green*、*lt_cyan*、*lt_red*、*lt_magenta*、*yellow*、および *hi_white* です。

`setprop propname word`
 属性 *propname* の値を *word* に設定します。

`singlestep` または `singlestep on`
 ステップ実行モードを有効にします。この場合、インタプリタは各コマンドを処理する前に `step ?` を表示し、キーボード入力を待ちます。ENTER キーを押すと次のコマンドが実行されます。

`singlestep off`
 ステップ実行モードを無効にします。

`source name`
 ファイル *name* をメモリーに読み込み、解釈を開始します。EOF に到達したら、その前の入力ソースに戻ります。

`unset name`
 変数 *name* を削除します。

`verbose` または `verbose on`
 詳細表示モードを有効にします。ソースファイルの各行と、実行されるコマンドが表示されます。

boot(1M)

	<code>verbose off</code> 詳細表示モードを無効にします。
x86 の組み込み関数	次の組み込み関数を式の中で使用できます。 <code>.strcmp(string1, string2)</code> <i>string1</i> が字句構造上、 <i>string2</i> より小さい、等しい、または大きいことにより、ゼロより小さい、または大きい整数値を返します。 <code>.strncmp(string1, string2, n)</code> <i>string1</i> が字句構造上、 <i>string2</i> より小さい、等しい、または大きいことにより、ゼロより小さい、または大きい整数値を返します。最大 <i>n</i> 文字が比較されます。 <code>.streq (string1, string2)</code> <i>string1</i> が <i>string2</i> に等しい場合は真、それ以外の場合は偽を返します。 <code>.strneq (string1, string2, n)</code> <i>string1</i> が <i>string2</i> に等しい場合は真、それ以外の場合は偽を返します。最大 <i>n</i> 文字が比較されます。 <code>.strfind (string, addr, n)</code> <i>addr</i> から始めてメモリー内の位置を <i>n</i> 個走査し、 <i>string</i> の開始位置を探します。 <i>string</i> はヌルで終わっていてもかまいません。 <i>string</i> が見つかった場合は真、見つからなかった場合は偽を返します。 .strfind を使用すると、さまざまなマシンや周辺ボードを識別するために ROM BIOS および BIOS 拡張機能内の文字列を検索できます。
使用例	
SPARC	例 1 対話モードでシングルユーザーとしてデフォルトのカーネルを起動する 対話モードでシングルユーザーとしてデフォルトのカーネルを起動するには、ok プロンプトで次のいずれかを入力します。 <code>boot -as</code> <code>boot disk3 -as</code>
32 ビット SPARC	例 2 デフォルトファイルとして 32 ビットカーネルを指定して kadb を起動する デフォルトファイルとして 32 ビットカーネルを指定して kadb を起動するには、次のように入力します。 <code>boot kadb -D kernel/unix</code>

例 2 デフォルトファイルとして 32 ビットカーネルを指定して kadb を起動する (続き)

例 3 32 ビットカーネルを明示的に起動する

32 ビットカーネルを明示的に起動するには、カーネルファイル名を指定する必要があります。そのため、32 ビットカーネルを対話モードでシングルユーザーとして起動するには、ok プロンプトで次のいずれかを入力します。

```
boot kernel/unix -as
boot disk3 kernel/unix -as
```

64 ビット SPARC

例 4 64 ビットカーネルを明示的に起動する

64 ビットカーネルを明示的に起動するには、カーネルファイル名を指定する必要があります。そのため、64 ビットカーネルを対話モードでシングルユーザーとして起動するには、ok プロンプトで次のいずれかを入力します。

```
boot kernel/sparcv9/unix -as
boot disk3 kernel/sparcv9/unix -as
```

明示的なファイル名を使用して 64 ビットカーネルを起動する前に、「注意事項 UltraSPARC システムの起動」を参照してください。

例 5 WAN 起動に対応した PROM によるネットワーク起動

さまざまな boot コマンド行呼び出し間の微妙な違いを示すために、ここでは network-boot-arguments が設定されており、以下のコマンドのように net に対してデバイス別名が設定されているものとします。

次のコマンドでは、デバイス別名内のデバイス引数が、デバイスドライバによって処理されます。ネットワーク起動サポートパッケージは、network-boot-arguments 内の引数を処理します。

```
boot net
```

次のコマンドでは、デバイス引数なし、とみなされます。ネットワーク起動サポートパッケージは、network-boot-arguments 内の引数を処理します。

```
boot net:
```

次のコマンドでは、デバイス引数なし、とみなされます。rarp は唯一のネットワーク起動サポートパッケージ引数であり、network-boot-arguments は無視されません。

```
boot net:rarp
```

次のコマンドでは、指定されたデバイス引数が処理されます。ネットワーク起動サポートパッケージは、network-boot-arguments 内の引数を処理します。

```
boot net:speed=100,duplex=full
```

boot(1M)

	<p>例 6 以前の PROM における wanboot の使用</p> <p>次のコマンドでは、CD-ROM から wanboot バイナリが読み込まれます。読み込まれた wanboot は、DHCP の実行時にコマンドインタプリタを起動し、鍵やその他の必要な構成情報をユーザーが入力できるようにします。</p> <pre>boot cdrom -F wanboot -o dhcp,prompt</pre>
x86	<p>例 7 対話モードでシングルユーザーとしてデフォルトのカーネルを起動する</p> <p>シングルユーザー状態の対話モードでデフォルトのカーネルを起動するには、> プロンプトで次のいずれかを入力します。</p> <pre>b -as</pre> <pre>b kernel/unix -as</pre>
ファイル	<pre>/platform/platform-name/ufsboot</pre> <p>ディスクまたは CD から起動する 第 2 レベルのプログラム</p> <pre>/etc/inittab</pre> <p>initdefault 状態が指定されているテーブル</p> <pre>/sbin/init</pre> <p>システムを initdefault 状態に移行するプログラム</p> <pre>/platform/platform-name/boot.conf</pre> <pre>/platform/hardware-class-name/boot.conf</pre> <p>boot ポリシーファイルの主パス名および代替パス名。ポリシーファイルはすべてのプラットフォーム上で実装されているわけではありません。</p>
32 ビット SPARC および x86	<pre>/platform/platform-name/kernel/unix</pre> <p>システムを起動するデフォルトのプログラム</p>
64 ビット SPARC のみ	<pre>/platform/platform-name/kernel/sparcv9/unix</pre> <p>システムを起動するデフォルトのプログラム</p> <p>「注意事項UltraSPARC システムの起動」を参照してください。</p>
x86 のみ	<pre>/etc/bootrc</pre> <p>起動処理を制御するスクリプト</p> <pre>/platform/platform-name/boot/solaris/boot.bin</pre> <p>x86 システムで ufsboot の代わりに使用される第 2 レベルの起動プログラム</p> <pre>/platform/platform-name/boot</pre> <p>起動関連ファイルが置かれているディレクトリ</p>
関連項目	<p>uname(1), eeprom(1M), init(1M), installboot(1M), kadb(1M), kernel(1M), shutdown(1M), uadmin(2), bootparams(4), inittab(4), vfstab(4), wanboot.conf(4), filesystem(5)</p> <p>Solaris のシステム管理 (基本編)</p>

RFC 903, *A Reverse Address Resolution Protocol*,
<http://www.ietf.org/rfc/rfc903.txt>

RFC 2131, *Dynamic Host Configuration Protocol*,
<http://www.ietf.org/rfc/rfc2131.txt>

RFC 2132, *DHCP Options and BOOTP Vendor Extensions*,
<http://www.ietf.org/rfc/rfc2132.txt>

RFC 2396, *Uniform Resource Identifiers (URI): Generic Syntax*,
<http://www.ietf.org/rfc/rfc2396.txt>

Solaris のシステム管理 (基本編)

『*Sun Hardware Platform Guide*』

『*OpenBoot Command Reference Manual*』

警告 boot ユーティリティは、起動可能プログラムとして使用できるファイルかどうか判別できません。起動不可能なファイルを起動するように要求された場合、boot ユーティリティはそのファイルを読み込み、そのファイルに制御を渡します。その場合の結果は予測できません。

注意事項 *platform-name* を調べるには、`uname(1)` の `-i` オプションを使用します。*hardware-class-name* を調べるには、`uname(1)` の `-m` オプションを使用します。

64 ビット SPARC

UltraSPARC システムの起動

一部のプラットフォームでは、64 ビットカーネルを実行するために、ファームウェアのアップグレードが必要な場合があります。詳細は『*Sun Hardware Platform Guide*』を参照してください。64 ビットカーネルパッケージがインストールされており、boot が、そのプラットフォームで 64 ビットカーネルを実行するにはファームウェアのアップグレードが必要であることを検出した場合、コンソールにメッセージが表示され、デフォルトのファイルとして 32 ビットカーネルが選択されます。

200MHz 以下の UltraSPARC-1 プロセッサを搭載したシステムでは、ユーザーが、プロセッサを停止することが可能な 64 ビットプログラムを実行してしまうことがあります。32 ビットカーネルで 64 ビットプログラムを実行することはできないので、このようなシステムでは、32 ビットカーネルがデフォルトのファイルとして選択されます。

問題を引き起こすコードシーケンスはきわめてまれなもので、コンパイラによって生成される可能性はほとんどありません。この問題を再現するためには、特殊なアセンブラコードを作成する必要があります。手作業で作成された適切なアセンブラーチンがこのコードシーケンスを使用することはまずありません。

boot(1M)

ユーザーが偶然または故意に、プロセッサの停止を引き起こすプログラムを実行してしまうリスクを甘受するのであれば、boot ポリシーファイルを変更することによって 64 ビットカーネルを実行するという選択も可能です。/platform/platform-name/boot.conf を編集し、以下の例に示すように、ALLOW_64BIT_KERNEL_ON_UltraSPARC_1_CPU という変数の値を true に設定したコメント行を解除します。

```
ALLOW_64BIT_KERNEL_ON_UltraSPARC_1_CPU=true
```

詳細は『Sun Hardware Platform Guide』を参照してください。

x86 のみ

各国語キーボードから「-」キーがなくなったので、このようなキーボードを使用する x86 ベースのシステムでは、代わりのキーを使用して boot コマンドに引数を渡してください。数字キーパッドの - を使用してください。キーボードの対応言語別に、起動時に - の代わりに使用するキーを示します。

キーボード	代替キー
イタリア	'
スペイン	'
スウェーデン	+
フランス	?
ドイツ	?

たとえば、スウェーデン語対応のキーボードの場合、b -r は b+r と打鍵します。画面には b -r が表示されます。

名前	catman – 参照マニュアル用のフォーマット整形したファイルの作成
形式	<code>/usr/bin/catman [-c] [-n] [-p] [-t] [-w] [-M directory] [-T macro-package] [sections]</code>
機能説明	<p>catman は nroff(1) または sgml(5) の入力ファイルから、オンラインマニュアルページのプレフォーマット・バージョンを生成します。プレフォーマットされたマニュアルページのディレクトリは、自己包括的にかつフォーマットされていないエンタリから独立して作成されるので、一群のマシンの間でそれらのマニュアルページを (rdist(1) などによって) 容易に配布することができます。</p> <p>catman は、MANPATH または -M で指定されたディレクトリに windex のデータベースファイルも作成します。windex のデータベースファイルは、キーワード、そのキーワードが指す参照マニュアルページ、参照マニュアルページに書かれたユーティリティまたはインタフェースの目的を説明するテキスト行の 3 つのカラムで構成されているリストです。各キーワードは、「名前」(NAME) の行の ‘-’ (ダッシュ) の前にあるコンマで区切られた単語のリストから抽出します。キーワードが指す参照マニュアルページは、「名前」の行の最初の単語です。3 つ目のカラムの記述は、「名前」の行の - のあとに続くテキストから抽出します。「名前」の行は、.TH マクロによって作成されるページヘッダーのすぐあとに置く必要があります (必要とされる形式については、「注意事項」を参照)。</p> <p>各マニュアルページを検査し、それに対応するプレフォーマット・バージョンが存在しないものあるいは、現状のマニュアルページよりも古いものについて、プレフォーマット・バージョンを再生成します。変更箇所があれば、catman は windex データベースも再生成します。</p> <p>マニュアルページがシャドウページである場合、つまりその内容が書かれた別のマニュアルページをソースファイルにしている場合、対象となるプレフォーマットされたマニュアルページへのシンボリックリンクが catx または fmtx ディレクトリ内に作成されます。フォーマットされていない nroff のソースファイルにあるシャドウファイルは、最初の行に <code>.so manx/yyy.x</code> という形式の行があることで識別できます。</p> <p>SGML のソースファイルにあるシャドウファイルは、文字列 SHADOW_PAGE があることで識別できます。シャドウファイルで宣言されたファイルの実体は、ソースとなるファイルを示しています。</p>
オプション	<p>以下のオプションを指定できます。</p> <p>-c SGML のソースファイルから、フォーマットされていない nroff のソースファイルを適切な man サブディレクトリに作成します。このオプションは SGML ファイルと同じ名前で man ディレクトリにある既存のファイルをすべて上書きします。</p> <p>-n windex データベースの作成または再生成を行いません。このオプションを指定した場合は windex データベースが作成されないのので、apropos、whatis、man -f、man -k コマンドを実行しても失敗します。</p>

catman(1M)

	-p	実際の処理は実行せず、どのような処理が行われるかの表示だけを行います。
	-t	cat サブディレクトリへの nroff を行う代わりに、該当する fmt サブディレクトリ内に troff 処理後のエントリを作成します。
	-w	whatis(1) および man(1) の -f と -k オプションで使用する windex データベースの作成だけを行います。マニュアルページの再フォーマットは実行しません。
	-M <i>directory</i>	引数に指定したディレクトリ (デフォルトは /usr/share/man) 中にあるマニュアルページを更新します。コンマは、マニュアルセクションの番号を示すために使用されているので、-M オプションの引数として指定するディレクトリ名には ';' (コンマ) が含まれないようにしてください (man(1) 参照)。
	-T <i>macro-</i> <i>package</i>	標準のマニュアルページマクロの代わりに、 <i>macro-package</i> 引数で指定したマクロパッケージを使用します (デフォルトは man(5))。
オペランド		次のオペランドを使用できます。
	<i>sections</i>	先頭文字が '-' でないパラメタは、catman によって処理されるマニュアルセクションとみなします。各セクションは空白で区切ります。このオペランドが指定されると、指定するマニュアルセクションだけが処理されます。次に例を示します。 <pre>catman 1 2 3</pre>
環境	TROFF	-t オプションが指定された場合に用いるフォーマットの名前。この環境変数が設定されていない場合は、troff(1) が用いられます。
	MANPATH	catman と man(1) で処理されるディレクトリの、コロンで区切られたリスト。各ディレクトリは、コンマで区切られたセクションのリストの後に続いて指定できます。この環境変数に値を設定すると、デフォルトのディレクトリ検索パスである /usr/share/man と、デフォルトのセクション検索パスである man.cf を無効にします。また、-M フラグはデフォルトのディレクトリ検索パスを、-s フラグはデフォルトのセクション検索パスを、それぞれ無効にします。
ファイル	/usr/share/man	マニュアルページのデフォルトのディレクトリ
	/usr/share/man/man*/*.*	nroff 入力ファイル (未処理)
	/usr/share/man/sman*/*.*	SGML 入力ファイル (未処理)

catman(1M)

/usr/share/man/cat*/**.* プレフォーマットされた nroff 処理後のマニュアルページ

/usr/share/man/fmt*/**.* プレフォーマットされた troff 処理後のマニュアルページ

/usr/share/man/windex 目次とキーワードのデータベース

/usr/lib/makewhatis windex データベース生成用のコマンドスクリプト

/usr/share/lib/tmac/an デフォルトのマクロパッケージ

属性 次の属性については attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWdoc
CSI	対応済み

関連項目 apropos(1), man(1), nroff(1), rdist(1), rm(1), troff(1), whatis(1), attributes(5), man(5), sgml(5)

診断 man?/xxx.? (.so'ed from man?/yyy.?): No such file or directory
 メッセージの冒頭に示されたファイルは、カッコ内のファイルが参照しようとしたが、存在しませんでした。

target of .so in man?/xxx.? must be relative to /usr/man
 catman では、ディレクトリ /usr/man からの相対パスで示したファイル名のみを参照できます。

opendir:man?: No such file or directory
 catman が通常検索するディレクトリのうちの 1 つが見つかりません (軽度の警告メッセージ)。

.: No such file or directory
 catman によって、中身が空のディレクトリが見つかりました (軽度の警告メッセージ)。

警告 以前に catman を実行して cat* ディレクトリがすでにインストールされている場合に、オペレーティングシステムをアップグレードした場合は、catman を実行する前に cat* ディレクトリ構造全体を削除してください (rm(1) 参照)。

すべての man* ディレクトリがそろっていない場合には、whatis データベースを再構築するために catman を再度実行しないでください。catman は、man* ディレクトリにもとづいてこの windex ファイルを構築します。

注意事項 catman には、正しい windex のインデックスファイルを生成するための要件があります。catman は、個々のマニュアルページのファイル中に、特定の形式を持つ 2 つのマクロ行 (ページの先頭の .TH 行と .SH NAME の行) を必要とします。

catman(1M)

.TH マクロには、最低 3 つの引数 (ファイル名、セクション番号、日付) が必要です。
.TH 行は、.TH マクロで始まり、そのあとに、1 つの空白、マニュアルページの
ファイル名、1 つの空白、セクション番号、1 つの空白、日付が続きます。日付は、
"day month year" (日本語の場合は "年月日") として二重引用符で囲みます。このうち
の month には 3 文字の省略形 (Jan、Feb、Mar、など) を指定します。

「名前」 (NAME) の行としても知られる .SH NAME マクロは、.TH 行のすぐあとに
置く必要があります (これら 2 つの行の間には何も置かない)。「名前」の行では、
フォントの変更はできません。.SH NAME のすぐ次の行には、マニュアルページの
ファイル名、コンマで区切られたシャドウファイル名 (もしあれば)、ダッシュ、簡単
な概要を含む行が続きます。これらの要素は、すべて 1 つの行に収める必要がありま
す (改行を入れないでください)。

.TH 行と .SH NAME の行の正しいコーディング例を次に示します。

```
.TH nismatch 1M "10 Apr 1998"  
.SH NAME  
nismatch, nisgrep \- utilities for searching NIS+ tables
```

名前 | cfgadm – 構成の管理

形式 | `/usr/sbin/cfgadm [-f] [-y | -n] [-v] [-o hardware_options] -c function ap_id...`
`/usr/sbin/cfgadm [-f] [-y | -n] [-v] [-o hardware_options] -x hardware_function ap_id...`
`/usr/sbin/cfgadm [-v] [-a] [-s listing_options] [-o hardware_options] [-l [ap_id | ap_type]]`
`/usr/sbin/cfgadm [-v] [-o hardware_options] -t ap_id...`
`/usr/sbin/cfgadm [-v] [-o hardware_options] -h [ap_id | ap_type]`

機能説明

cfgadm コマンドを使用して、動的な再構成が可能なハードウェア資源に対して構成の管理を行うことができます。これらの操作には、状態 (state) の表示 (-l)、検査の開始 (-t)、構成状態の変更の開始 (-c)、ハードウェア固有の機能の実行 (-x)、および構成管理のヘルプ情報の表示 (-h)、が含まれます。構成管理は、接続点 (attachment point) で実行されます。接続点は、Solaris の動作中にハードウェア資源の動的再構成を行うことにシステムソフトウェアが対応している場所です。

構成の管理では、マシン上に実際にあるハードウェア資源と、構成済みで Solaris が認識できるハードウェア資源が区別されます。構成管理機能の特性はハードウェアに依存し、ハードウェア固有のライブラリを呼び出すことで実行されます。

構成管理は、接続点で実行されます。接続点に設置されているハードウェア資源には、システムの稼働中に物理的な交換ができるものとできないものがありますが、構成管理インタフェースによって、動的に再構成することはできません。

接続点は、接続点の向こう側に位置するハードウェア資源とは別の2つの固有の要素を定義します。接続点の2つの要素は、受容体 (receptacle) と占有装置 (occupant) です。ハードウェア資源の物理的な取り付け、取り外しは接続点で行われ、その結果、受容体に占有装置が追加されたり削除されたりします。構成管理は、接続点での構成管理機能だけでなく、この物理的な着脱操作にも対応しています。

接続点には、状態 (state) と条件 (condition) の情報が関連付けられています。構成管理インタフェースを使用して、接続点の状態の変化を制御することができます。受容体は、empty、disconnected、connected の3つの状態のいずれかになります。また、占有装置は、configured と unconfigured のいずれかの状態になります。

受容体は、接続点に占有装置がない場合に、必ず受容体の通常の状態である empty になります。この状態には、稼働中のシステムの一部を一時的に停止することができます。ハードウェア固有の機能が関係します。受容体が占有装置をシステムの通常の使用から切り離すことができる場合に、その受容体は disconnected 状態になることもできます。この状態は、占有装置のハードウェア資源をシステムが完全に利用できるようにする前にそのハードウェアに対する検査を実行する場合や、占有装置の物理的な取り外しや再構成のための準備の1つの段階として、主に使用されます。disconnected 状態の受容体は、ハードウェアの許容範囲内で占有装置をシステムから分離しますが、検査や設定が必要な場合は使用を許可する場合があります。受容体

は、占有装置に含まれるハードウェア資源の通常の使用が許可されている場合に、必ず `connected` 状態になります。`connected` 状態は、占有装置を含み、かつ構成管理操作が実行されていない受容体の通常の状態です。

`unconfigured` 状態の占有装置に含まれるハードウェア資源は、Solaris の通常のデータ構造では表現されないため、Solaris はそのハードウェアを使用できません。未構成の占有装置に対して実行できる操作は、構成管理操作に限られています。

`configured` 状態の占有装置に含まれるハードウェア資源は、Solaris の通常のデータ構造で表現されるため、Solaris は、一部またはすべてのハードウェア資源を使用することができます。占有装置は、必ず `configured` 状態か `unconfigured` 状態になります。

接続点は、`unknown`、`ok`、`failing`、`failed`、`unusable` の 5 つの条件のいずれかになります。接続点は、電源投入検査と不揮発性記録保存の結果によって、システムをどの条件にも置く可能性があります。

`configured` 状態の占有装置を持つ接続点は、`unknown`、`ok`、`failing`、`failed` の条件のいずれかになります。`failing` または `failed` 条件にない接続点は、ハードウェア固有の回復可能なエラーがしきい値を超えると、操作中に `failing` 状態になる場合があります。また、`failed` 条件にない接続点は、回復不可能なエラーによって、操作中に `failed` 条件に変わる場合があります。

`unconfigured` 状態にある占有装置を持つ接続点は今までに挙げた条件のどれになる可能性もあります。`unconfigured` 状態にある占有装置を持つ接続点の条件は、マシン固有の時間しきい値が経過した後に、`ok` から `unknown` になる場合があります。検査機能を開始した場合は、検査の結果によって接続点の条件が `ok`、`failing`、`failed` のいずれかに変わります。検査機能を持たない接続点は、接続点を `unknown` 条件のままにする場合があります。検査が中断された場合は、接続点の条件は、以前の条件、`unknown`、`failed` に設定することができます。`unknown`、`ok`、`failing`、`failed` のいずれかの条件にある接続点には、再検査を行うことができます。

接続点は、さまざまな理由によって `unusable` 条件になります。理由としては、受容体に対する不適切な電力投入や冷却、占有装置が認識できない、対応していない、不適切に構成されている、などが挙げられます。`unusable` 条件にある接続点は、システムで使用することができません。通常、この条件は、接続点に対して物理的な対処がなされない限り変わりません。

また、接続点は、状態の変更処理が進行中である場合や、条件が再評価されている場合に、それを示す使用状態情報を保持します。

接続点は、システムデバイス階層構造の中での接続点のタイプと位置に関連するハードウェア固有の識別子 (`ap_ids`) に対応しています。`ap_id` は単一の接続点を特定するために、一意になっている必要があります。`ap_id` の仕様には、物理タイプと論理タイプの 2 種類が用意されています。物理 `ap_id` には、完全なバス名を指定します。論理 `ap_id` には、簡略表記法を使用し、ユーザーにとってより簡単な方法で接続点を指定します。

たとえば、システムのバックプレーンスロット番号 7 にある接続点の物理 *ap_id* は `/devices/central/fhc/sysctrl:slot7` となり、論理 *ap_id* は `system:slot7` になります。また、システムの第 2 PCI 入出力バス上にある 3 番目の受容体の論理 *ap_id* は `pci2:plug3` になります。

接続点も動的に作成されます。動的接続点には、そのシステムに設定されている基本接続点を基にして名前が付けられます。動的接続点の *ap_ids* は、2 つのコロン、基本構成要素、および動的構成要素で構成されます。基本構成要素は、基本接続点 *ap_id* です。動的構成要素は、ハードウェア固有で、対応するハードウェア固有のライブラリによって生成されます。

たとえば、SCSI HBA を表現し、物理 *ap_id* が `/devices/sbus@1f,0/SUNW,fas@e,8800000:scsi` で、論理 *ap_id* が `c0` である基本接続点を想定します。この SCSI HBA に接続されているディスクは、論理 *ap_id* が `c0::dsk/c0t0d0` である動的接続点によって表現されます。ここで、`c0` は基本構成要素で、`dsk/c0t0d0` はハードウェア固有の動的構成要素です。同様に、この動的接続点の物理 *ap_id* は `/devices/sbus@1f,0/SUNW,fas@e,8800000:scsi::dsk/c0t0d0` になります。

ap_type は *ap_id* の一部で、それ自身だけでは一意にならず、単一の接続点を特定することができません。*ap_type* は、論理 *ap_id* の一部を含み、コロン (:) 区切り記号を含まない部分文字列です。たとえば、`pci` の *ap_type* は、論理 *ap_id* が `pci` で始まる接続点をすべて出力します。

ap_types は、できるだけ使用しないでください。-s オプションの新しい選択サブオプションを使用すれば、より汎用的な方法で柔軟に接続点を選択することができます。【オプション】を参照してください。

cfgadm コマンドは、ハードウェア固有ライブラリに含まれるハードウェア固有の機能と主に対話するため、cfgadm コマンドの動作はハードウェアに依存します。

それぞれの構成管理操作では、サービスの中断が必要になる場合があります。要求された機能を完了するために、対話式で操作中のユーザーにとって目に見えるサービスの中断が必要になる場合は、機能の開始前に標準エラー出力に確認メッセージが表示され、標準入力による確認を促します。すべての質問に対する `yes` を意味する -y オプション、または `no` を意味する -n オプションを指定することによって、確認を省略することもできます。検査レベルなどのハードウェア固有のオプションは -o オプションを使用して、サブオプションとして指定することができます。

システム構成の状態を変更する操作は、システムログデーモンである syslogd(1M) によって監視されます。

このコマンドの引数は、getopt(3C) および getsubopt(3C) の構文規約に従います。

オプション 以下のオプションを使用することができます。

-a
-1 オプションによって、動的接続点のリストも出力されるように指定します。

`-c function`

`ap_id` で指定された接続点の状態を `function` で指定された状態に変更します。

`function` には、`insert`、`remove`、`disconnect`、`connect`、`configure`、`unconfigure` のいずれかを指定することができます。これらの関数は、ハードウェア固有のライブラリルーチン呼び出しで接続点の状態を変更します。これらの関数の定義を以下に示します。

<code>insert</code>	占有装置を手動で追加する操作を実行したり、物理的な追加を実行するハードウェア機能を起動します。 <code>insert</code> には、システムの一部を一時的に停止するハードウェア固有の副作用が伴う場合があります。このような場合、ハードウェア固有のライブラリは、対応する警告メッセージを生成し、ユーザーに対して、そのハードウェア固有の問題と手順を提供します。ハードウェアに起因するさまざまなエラーによってこの関数が失敗し、受容体の条件が <code>unusable</code> になる場合があります。
<code>remove</code>	占有装置を手動で削除する操作を実行したり、物理的な削除を実行するハードウェア機能を起動します。 <code>remove</code> には、システムの一部を一時的に停止するハードウェア固有の副作用が伴う場合があります。このような場合、ハードウェア固有のライブラリは、対応する警告メッセージを生成し、ユーザーに対して、そのハードウェア固有の注意事項と手順を提供します。ハードウェアに起因するさまざまなエラーによってこの関数が失敗し、受容体の条件が <code>unusable</code> になる場合があります。
<code>disconnect</code>	ハードウェア固有の操作を実行して、受容体を <code>disconnected</code> 状態にします。 <code>disconnected</code> 状態にすることによって、占有装置に対して、受容体を介した通常の方法による操作が行えなくなります。
<code>connect</code>	ハードウェア固有の操作を実行して、受容体を <code>connected</code> 状態にします。受容体が <code>connect</code> 状態にすることによって、占有装置に対して、受容体を介した通常の方法による操作が行えるようになります。
<code>configure</code>	ハードウェア固有の操作を実行して、占有装置のハードウェア資源を Solaris が使用できるようにします。構成された占有装置はシステム構成の一部になり、 <code>psradm(1M)</code> 、 <code>mount(1M)</code> 、 <code>ifconfig(1M)</code> などの Solaris デバイス操作メンテナンスコマンドによる操作の対象となります。
<code>unconfigure</code>	ハードウェア固有の操作を実行して、占有装置のハードウェア資源をシステムから論理的に削除します。この関数を使用するには、占有装置が現在構成されていて、占有装置のハードウェアが Solaris によって使用されていない必要があります。

状態変更関数は、接続点の条件や、その他のハードウェア固有の問題によって失敗する場合があります。資源を追加するための状態変更関数 (`insert`、`connect`、`configure`) は、接続点が `ok` または `unknown` 条件にある場合に、ハードウェア固有のライブラリに渡されます。接続点がそれ以外の条件にある場合は、強制オプション (`-f`) を使用した場合に限って、資源を追加するための関数がハードウェア固有のライブラリに渡されます。システムからハードウェア資源を削除するための関数 (`remove`、`disconnect`、`unconfigure`) によるハードウェア固有のライブラリの呼び出しは、接続点の条件によって妨げられることはありません。接続点が `unknown` 条件にある場合に、関数は、ハードウェア固有のライブラリによって拒否される場合があります。

接続点の条件は、状態変更関数によって変更されないこともありますが、状態変更操作中のエラーによって接続点の条件が変わる場合があります。条件の書き換えと状態の強制的な変更は、強制オプション (`-f`) を指定した場合だけ実行することができます。これらの処理は、強制オプションを指定しないと失敗します。強制オプションは、ハードウェア固有の安全性検査および完全性検査によって無効になる場合があります。

`-f`

指定された処理を強制的に実行します。これは主に、ハードウェア固有の安全機能を無効にするために使用します。状態の変更操作を強制することによって、ハードウェア固有の安全検査によって `ok` や `unknown` 状態にない占有装置のハードウェア資源を使用することができる場合があります。

`-h [ap_id | ap_type ...]`

ヘルプメッセージテキストを出力します。`ap_id` または `ap_type` を指定すると、この引数によって指定された接続点に関するハードウェア固有のライブラリのヘルプルーチンが呼び出されます。

`-l [ap_id | ap_type ...]`

指定された接続点の状態や条件を一覧表示します。接続点を抽出するには、`-s` オプションと `select` サブオプションを使用します。いずれかの処理オプションを使用せずに `cfgadm` コマンドを起動するのは、引数を使用せずに `-l` を使用することと同じです。表示画面の書式は `-v` および `-s` オプションによって制御されます。`-a` オプションが指定されているときは、接続点が動的に展開されます。

`-n`

対話型の確認を行わず、応答が `no` であるとみなします。`-n` と `-y` のいずれも指定しないと、標準エラー出力と標準入力によって対話型の確認が行われます。これらの標準的なチャンネルのいずれも端末 (`isatty(3C)` によって判定されている) に対応していない場合は、`-n` オプションが想定されます。

`-o hardware_options`

コマンドの主オプションに対してハードウェア固有のオプションを指定します。`hardware_options` の文字列の書式と内容は完全にハードウェア固有のものです。オプション文字列の `hardware_options` は `getsubopt(3C)` の構文規約に従います。

`-s listing_options`

一覧表示 (`-l`) コマンドに対して一覧表示オプションを指定します。`listing_options` は `getsubopt(3C)` の構文規約に従います。サブオプションを使用して、接続点の選択条件 (`select=select_string`)、適切な照合タイプ (`match=match_type`)、一覧表

cfgadm(1M)

示する順序 (*sort=field_spec*)、表示するデータ (*cols=field_spec* と *cols2=field_spec*)、列の区切り記号 (*delim=string*)、列の見出し行の抑制 (*noheadings*) を指定します。

select サブオプションを指定すると、指定された条件と一致する接続点だけが一覧表示されます。 *select* サブオプションの構文は次のとおりです。

```
cfgadm -s select=attr1(value1):attr2(value2)...
```

attr は、 *ap_id*、 *class*、 *type* のいずれかです。 *ap_id* は論理 *ap_id* フィールド、 *class* は接続点のクラス、 *type* はタイプフィールドです。 *value1*、 *value2* などは、照合する値です。照合タイプを指定するには、次のように *match* サブオプションを使用します。

```
cfgadm -s match=match_type,select=attr1(value1)...
```

match_type は、 *exact* または *partial* から選択します。デフォルト値は *exact* です。

select サブオプションの引数は、シェルから保護するために引用符で囲みます。

field_spec には、1つの *data-field* または *data-field:data-field:data-field* のようにコロンの(:)で区切った複数の *data-field* を指定します。 *data-field* は、 *ap_id*、 *physid*、 *r_state*、 *o_state*、 *condition*、 *type*、 *busy*、 *status_time*、 *status_time_p*、 *class*、 *info* のいずれかです。 *ap_id* フィールドの出力は接続点の論理名で、 *physid* フィールドは物理名です。 *r_state* フィールドは、 *empty*、 *disconnected*、 *connected* のいずれかになります。 *o_state* フィールドは、 *configured* と *unconfigured* のいずれかになります。 *busy* フィールドは、接続点が使用中の場合に *y* になり、使用中でない場合に *n* となります。 *type* フィールドと *info* フィールドはハードウェア固有のフィールドです。 *status_time* フィールドには、 *r_state*、 *o_state*、または接続点の条件が最後に変更された時刻が表示されます。 *status_time_p* フィールドは、構文解析が可能な *status_time* フィールドです。接続点にクラスが関連付けられている場合は、 *class* フィールドにクラス名が表示されます。接続点にクラスが関連付けられていない場合は、 *class* フィールドに *none* が表示されます。

field_spec 内のフィールドの順序は重要です。 *sort* サブオプションでは、最初に与えられたフィールドが主ソートキーになります。 *cols* および *cols2* サブオプションでは、指定した順序でフィールドが出力されます。 *cols* 内のフィールドは常に出力されます。 *cols2* 内のフィールドは、指定された場合に限って2行目に出力されます。 *data-field* に対するソートの順序は、 *sort* サブオプションに対する *field_spec* 内の *data-field* 名の前にマイナス(-)を付けることによって逆になります。 *sort* のデフォルトの値は *ap_id* です。 *cols* および *cols2* のデフォルトの値は *-v* オプションが指定されているかどうかによって異なります。 *-v* が指定されていない場合は、 *cols* は *ap_id:r_state:o_state:condition* になり、 *cols2* は設定されません。 *-v* が指定されている場合は、 *colss* は *ap_id:r_state:o_state:condition:info* になり、 *cols2* は *status_time:type:activity:physid:* になります。 *delim* のデフォルトの値は、単一の空白文字です。 *delim* の値には、任意の長さの文字列を指定することもできます。区切り記号にはコンマ(,)を含めることはできません (*getsubopt(3C)* を参照)。これらの一覧表示オプションは、構文解析が可能な出力を生成するために使用することができます。「注意事項」を参照してください。

-t

1 つまたは複数の接続点の検査を実行します。この検査機能は、接続点の条件を再評価するために使用します。*hardware_options* の中で検査レベル指示子を指定しないと、重度の障害を検出する最も早い検査方法が使用されます。

より包括的な検査は個々のハードウェアに依存するため、*hardware_options* を使用して選択します。

検査の結果は、指定された占有装置の条件を >ok (障害が発見されなかった場合)、*failing* (回復可能な障害が発見された場合)、*failed* (回復不可能な障害が発見された場合) のいずれかに更新するために使用されます。

検査が中断された場合は、接続点の条件は、以前の値に戻るか、ok (障害が発見されなかった場合)、*failing* (回復可能な障害が発見された場合)、*failed* (回復不可能な障害が発見された場合) のいずれかに設定されます。接続点は、エラーがなく、検査が正常に完了した場合のみ ok に設定されます。

-v

詳細表示モードで実行します。-c、-t、-x オプションを指定した場合に、各試行操作の結果を表示するメッセージを出力します。-h オプションを指定した場合は、詳細なヘルプ情報が表示されます。-l オプションを指定した場合は、各接続点に関するすべての情報を出力します。

-x*hardware_function*

ハードウェア固有の機能を実行します。受容体や占有装置の状態は、専用ハードウェア固有の機能を使用して変更します。接続点の状態は、ハードウェア固有の機能の動作中に検出されたエラーの結果として変化する場合があります。

hardware_function 文字列の書式と内容は完全にハードウェア固有のものです。オプション文字列の*hardware_function* は `getsubopt(3C)` の構文規約に従います。

-y

対話型の確認を行わず、応答が *yes* であるとみなします。

使用方法

このコマンドを使用するために必要な特権は、ハードウェアに依存します。一般的に、デフォルトのシステム設定では、一覧表示オプション以外のすべての機能はスーパーユーザーのみが使用することができます。

使用例

例 1 デバイスツリー内の接続点の一覧表示

以下の例は、動的接続点以外のすべての接続点を一覧表示します。

example# `cfgadm`

Ap_Id	Type	Receptacle	Occupant	Cond
system:slot0	cpu/mem	connected	configured	ok
system:slot1	sbus-upa	connected	configured	ok
system:slot2	cpu/mem	connected	configured	ok
system:slot3	unknown	connected	unconfigured	unknown
system:slot4	dual-sbus	connected	configured	failing
system:slot5	cpu/mem	connected	configured	ok
system:slot6	unknown	disconnected	unconfigured	unusable
system:slot7	unknown	empty	unconfigured	ok
c0	scsi-bus	connected	configured	unknown

例 1 デバイスツリー内の接続点の一覧表示 (続き)

```
c1                scsi-bus        connected        configured        unknown
```

例 2 構成することができるすべてのハードウェアの情報の一覧表示

以下の例は、現在構成することができるすべてのハードウェアの情報を一覧表示します。動的接続点で表現されるハードウェアも表示されます。

```
example# cfgadm -al
```

Ap_Id	Type	Receptacle	Occupant	Cond
system:slot0	cpu/mem	connected	configured	ok
system:slot1	sbus-upa	connected	configured	ok
system:slot2	cpu/mem	connected	configured	ok
system:slot3	unknown	connected	unconfigured	unknown
system:slot4	dual-sbus	connected	configured	failing
system:slot5	cpu/mem	connected	configured	ok
system:slot6	unknown	disconnected	unconfigured	unusable
system:slot7	unknown	empty	unconfigured	ok
c0	scsi-bus	connected	configured	unknown
c0::dsk/c0t14d0	disk	connected	configured	unknown
c0::dsk/c0t11d0	disk	connected	configured	unknown
c0::dsk/c0t8d0	disk	connected	configured	unknown
c0::rmt/0	tape	connected	configured	unknown
c1	scsi-bus	connected	configured	unknown

例 3 接続点の属性に基づいて選択された情報の一覧表示

以下の例は、接続点のうち、scsi で始まるクラス、c で始まる ap_id、および scsi で始まる type フィールドを持つものをすべて一覧表示します。-s オプションの引数は、シェルから保護するために引用符で囲みます。

```
example# cfgadm -s "match=partial,select=class(scsi):ap_id(c):type(scsi)"
```

Ap_Id	Type	Receptacle	Occupant	Cond
c0	scsi-bus	connected	configured	unknown
c1	scsi-bus	connected	configured	unknown

例 4 現在構成することができるハードウェアの情報の詳細表示

以下の例は、ap-type system の現在構成することができるハードウェアの情報を詳細表示モードで示します。

```
example# cfgadm -v -l system
```

Ap_Id	Type	Receptacle	Occupant	Condition	Information
When	Type	Busy	Phys_Id		
system:slot1			connected	configured	ok
Apr 4 23:50	sbus-upa	n	/devices/central/fhc/sysctrl:slot1		
system:slot3			connected	configured	ok non-detachable
Apr 17 11:20	cpu/mem	n	/devices/central/fhc/sysctrl:slot3		
system:slot5			connected	configured	ok
Apr 4 23:50	cpu/mem	n	/devices/central/fhc/sysctrl:slot5		
system:slot7			connected	configured	ok

例 4 現在構成することができるハードウェアの情報の詳細表示 (続き)

```
Apr  4 23:50 dual-sbus n          /devices/central/fhc/sysctrl:slot7
```

When 列は status_time フィールドの内容を表示します。

例 5 ハードウェア固有の拡張検査

以下の例は、ハードウェア固有の拡張検査を使用して 2 つの占有装置を検査します。

```
example# cfgadm -v -o extended -t system:slot3 system:slot5
Testing attachment point system:slot3 ... ok
Testing attachment point system:slot5 ... ok
```

例 6 強制オプション

以下の例は、強制オプションを使用して、failing 状態の占有装置をシステムに構成します。

```
example# cfgadm -f -c configure system:slot3
```

例 7 システムから占有装置を構成解除する

以下の例は、システムから占有装置を構成解除します。

```
example# cfgadm -c unconfigure system:slot4
```

例 8 接続点の占有装置を構成する

以下の例は、占有装置を構成します。

```
example# cfgadm -c configure c0::dsk/c0t0d0
```

環境 cfgadm の実行に影響を及ぼす環境変数の LC_TIME、LC_MESSAGES、NLSPATH、TZ については、environ(5) を参照してください。

LC_MESSAGES cfgadm が見出し行とエラーメッセージを表示する方法を特定しません。出力されるデータは、この環境変数の影響を受けません。

LC_TIME cfgadm が状態の変更時間 (status_time) を表示する方法 (人による判読が可能な形式) を特定します。

TZ 状態の変更時間を変換する際に使用される時間帯を特定します。これは人による判読が可能な (status_time) と構文解析が可能な (status_time_p) 形式の両方に当てはまります。

終了ステータス 以下の終了値が返されます。

- 0 正常に完了しました。
- 1 エラーが発生しました。
- 2 指定された宛先が構成管理に対応していません。

cfgadm(1M)

3 使用方法上のエラー。

属性 以下の属性については、attributes(5)を参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 `cfgadm_pci(1M)`, `cfgadm_sbd(1M)`, `cfgadm_scsi(1M)`, `ifconfig(1M)`, `mount(1M)`, `psradm(1M)`, `prtdiag(1M)`, `syslogd(1M)`, `config_admin(3CFGADM)`, `getopt(3C)`, `getsubopt(3C)`, `isatty(3C)`, `attributes(5)`, `environ(5)`

診断 診断メッセージは標準エラー出力に表示されます。オプションや使用方法のエラー以外に、次の診断メッセージがこのユーティリティによって表示されます。

`cfgadm: Configuration administration not supported on ap_id`

`cfgadm: No library found for ap_id`

`cfgadm: ap_id is ambiguous`

`cfgadm: operation: Insufficient privileges`

`cfgadm: Attachment point is busy, try again`

`cfgadm: No attachment points with specified attributes found`

`cfgadm: System is busy, try again`

`cfgadm: operation: Operation requires a service interruption`

`cfgadm: operation: Data error: error_text`

`cfgadm: operation: Hardware specific failure: error_text`

エラーメッセージの詳細については `config_admin(3CFGADM)`を参照してください。

注意事項 ハードウェア資源は、ハードウェア固有の方法で未構成プールに入ります。これは、システムの初期設定や構成解除操作の結果としてなど、さまざまな状況で発生します。unconfigured 状態にある占有装置は、システムによる特定の介入が発生するまで、システムが使用することはできません。このような干渉は、オペレーターが起動したコマンドや、自動構成機構によって発生します。

`cfgadm` コマンドの一覧表示オプションは、シェルスクリプトの中などで、別のコマンドに対する構文解析可能な入力として使用することができます。構文解析可能な出力の場合は、必要なフィールドを選択するときに、`-s` オプションを使用する必要があります。`-s` オプションは、列の見出しを抑制するときにも使用できます。`ap_id`、`physid`、`r_state`、`o_state`、`condition`、`busy`、`status_time_p`、`class`、`type` フィールドは、常に構文解析が可能な出力を生成します。

以下はシェルスクリプトの一部分で、タイプが CPU の正常な未構成の占有装置で最初のものを検出します。


```
found=
cfgadm -l -s "noheadings,cols=ap_id:r_state:condition:type" | \
while read ap_id r_state cond type
do
    if [ "$r_state" = unconfigured -a "$cond" = ok -a "$type" = CPU ]
    then
        if [ -z "$found" ]
        then
            found=$ap_id
        fi
    fi
done
if [ -n "$found" ]
then
    echo "Found CPU $found"
fi
```

構文解析が可能な時間フィールド (`status_time_p`) の形式は、`YYYYMMDDhhmmss` で、文字列の比較を行うのに都合がよい書式で、年、月、日、時間、分、秒を表示します。

システム構成管理で使用できるものの詳細については、ハードウェア固有のマニュアルを参照してください。

cfgadm_ac(1M)

名前	cfgadm_ac - EXX00 メモリーシステムの管理
形式	<pre>/usr/sbin/cfgadm [-c configure] [-f] [-o disable-at-boot enable-at-boot] ac#:bank# ... /usr/sbin/cfgadm [-c unconfigure] [-o disable-at-bootp enable-at-boot] ac#:bank# ... /usr/sbin/cfgadm [-v] [-o quick normal extended, [max_errors=#]] -t ac#:bank#... /usr/sbin/cfgadm -x relocate-test ac#:bank# ... /usr/sbin/cfgadm [-l] -o disable-at-boot enable-at-boot ac#:bank# ...</pre>
機能説明	<p>ac ハードウェア固有ライブラリ</p> <p>/usr/platform/sun4u/lib/cfgadm/cfgadm_ac.so.1 は、 cfgadm_sysctrl(1M) を使用する CPUU/メモリーボードの Dynamic Reconfiguration (DR: 動的再構成) の一部として、E6X00、E5X00、E4X00、E3X00 システムのメモリーバンクの構成と構成解除に関連する機能を提供します。</p> <p>メモリーバンクは、デバイスツリー上の接続点として表されます。CPU/メモリー ボードそれぞれに2つの接続点が用意されます。1つがボード上の各バンク (バンク 0 とバンク 1) 用になります。バンクが空いている場合は、受容体の状態は empty (空 き) になります。バンクが使用されている場合は、受容体の状態は connected(接続) になります。受容体の状態が disconnected (接続解除) になることはありません。接続 されているメモリーバンクの占有状態は、構成することも構成解除することもできま す。占有状態が構成されている場合は、メモリーは Solaris によって使用されていま す。構成解除されている場合は、使用されていません。</p>
オプション	<p>コマンドオプションの詳細については、cfgadm(1M) を参照してください。</p> <p>以下のオプションがサポートされています。</p> <p>-c configure unconfigure 占有状態を変更します。引数が configure の場合は、メモリーが初期化され、 Solaris のメモリープールに追加します。引数が unconfigure の場合は、Solaris によって使用されているメモリーを切り離します。CPU/メモリーボードをシステ ムから取り外すには、2つのメモリーバンクを構成解除する必要があります。</p> <p>ボード上のメモリーが disabled-at-boot (infoフィールド参照) に指定されて いて、-f (強制) オプションやブート許可フラグ (-o enable-at-boot) がいずれ も指定されていない場合は、cfgadm コマンドは、構成操作を許可しません。構成 操作には、初期化する必要のあるメモリーのサイズに応じて、少し時間がかかります。</p> <p>システムに十分な使用可能メモリーがない (VM viability エラー)、または構成 解除されるべきバンクに切り離せないメモリーがある (non-relocatable pages エラー) 場合、cfgadm コマンドは、構成解除操作を許可しません。再配置不可の ページの現状は、info フィールドの一覧にある permanent によって表されま す。Solaris が使用しているメモリーの取り外しを行うと、システム負荷や補助記憶</p>

装置のページングの大きさによって、かなり時間がかかります。構成解除操作は、いつでも中止することができ、シグナルによるコマンドの中断によって、メモリーは完全に構成されている状態に戻ります。もし、タイムアウトまでの期間内に削除できるメモリーがない場合は、構成解除操作は、自動的に取り消されます。デフォルトのタイムアウト期間は 60 秒ですが、`-o timeout=#` オプションで変更することもできます。この数値を 0 にすると、タイムアウトは不許可になります。

`-f`

強制オプション。このオプションは、非揮発性の変数 `disabled-memory-list` で、`disabled at boot` (ブート不可) に指定された、構成されているメモリーバンクのブロックを無効にしたい場合に使用してください。詳細は、『特記事項: Sun Enterprise 6x00/5x00/4x00/3x00 システム』を参照してください。

`-l`

リストオプション。このオプションについての説明は、`cfgadm(1M)` のマニュアルページを参照してください。

`type` フィールドは、常に `memory` です。

`info` フィールドには、以下に示すような空のメモリーバンクについての情報が保持されます。

`slot# empty`

`slot#` は、CPU/メモリーボードが挿入されているシステムスロットを示しています。たとえば、これがスロット 11 である場合、`cfgadm` が関連付けられたボードを操作するために使用する接続点は、`sysctrl0:slot11` になります。`info` フィールドには、以下に示すような接続されているバンクについての情報が保持されます。

`slot# sizeMB|sizeGB [(sizeMB|sizeGB を使用)] base 0x###
[interleaved #-way] [disabled at boot] [permanent]`

バンクのサイズは、MB か GB か適切な単位で与えられます。メモリーがまだ使い切られていない場合、使用サイズが表示されます。物理ベースアドレスは、16 進数で与えられます。メモリーバンクが他のバンクによってインタリーブされている場合は、インタリーブファクター(因子)が出力されます。ボード上のメモリーが、非揮発性の変数 `disabled-memory-list` を使ってブート不可になっている場合は、そのことが表示されます。バンクに切り離せないメモリーがあると、固定 (permanent) として出力されます。

`-o disable-at-boot | enable-at-boot`

このオプションは、非揮発性の `disabled-memory-list` 変数を変更することを許可します。これらのオプションは、コマンド要求がない場合は、`-c` オプションの発行や、明白または暗示的なリスティングオプション `-l` と結合して、使用することができます。使用不可メモリーの一覧にあるボード上に構成されているメモリーブロックを無効にするには、`configure` コマンドを `-o enable-at-boot` オプションで使用してください。

`-o extended | normal | quick`

テストレベルを指定するには、`-t` オプションを使用してください。

テストレベル `normal` では、各メモリーセルに 0 と 1 のいずれも格納できることを確認して、さらにすべてのセルが個別にアドレス可能かどうか検査します。テストレベル `quick` では、すべてのメモリーに 0 と 1 を書き込むテストだけ

cfgadm_ac(1M)

を行うため、アドレス線の障害を発見することはできません。extended テストでは、近接したセル同士の干渉の問題をテストするパターンを使用します。デフォルトのテストレベルは、normal です。-t オプションを参照してください。

-o max_errors=#

-t オプションと共に使用して、許可するエラーの最大数を指定します。このオプションを指定しなかった場合は、デフォルト値の 32 が適用されます。

-o timeout=#

構成解除コマンドと共に使用して、自動取り消しによるタイムアウトを設定します。デフォルト値は 60 で、単位は秒です。数値が 0 の場合は、タイムアウトしません。

-t

構成されていないメモリーのバンクをテストします。-o quick | normal | extended オプションでテストレベルを指定してください。

メモリーバンクテストが実行できた場合は、cfgadm コマンドはステータス 0 (成功) で終了します。テストの結果は、接続点のためという条件で利用可能です。

-v

詳細表示オプションです。-t オプションと組み合わせて使用することにより、テストの進行状況と結果を詳細に表示できます。

-x relocate-test

指定されたメモリーバンクで使用されているメモリーの全ページに対し、構成解除コマンドで行われるように、再配置を試みます。この操作の成功は、バンクが構成解除できるかどうかを保証しません。この操作の失敗は、構成解除できなかったことを意味します。このオプションの用途は、テストのみに限定されます。

オペラント 以下のオペラントがサポートされています。

ac#:bank#

メモリーバンクの接続点は、アドレスコントローラー (ac#) ドライバ (ac) のインスタンスによって作成されます。ac ドライバの 1 つのインスタンスは、各システムボードによって作成されますが、CPU/メモリーボードに関連付けられたインスタンスだけは、バンク 0 とバンク 1 の 2 つの接続点を作成します。

このフォームは、cfgadm(1M) によって与えられた論理 ap_id 指定と一致します。これに対応する物理 ap_id のリストは、ファイルの項目にあります。

ac ドライバインスタンスの番号付けは、対応するボードのロット番号とは関係はありません。完全な物理接続点の識別子には、fhc@ に続く、ロット番号の 2 倍を 16 進数で表したロット番号が入ります。

ファイル /devices/fhc@*,f8800000/ac@0,1000000:bank?
接続点

/usr/platform/sun4u/lib/cfgadm/cfgadm_ac.so.1
 ハードウェア固有ライブラリファイル

属性 以下の属性についての説明は、attributes(5)を参照してください。

属性タイプ	属性値
使用条件	SUNWkvm.u

関連項目 cfgadm(1M), cfgadm_sysctrl(1M), config_admin(3CFGADM), attributes(5)

『日本語Solaris7 Sun Enterprise 6X00, 5X00, 4X00, 3X00 システム Dynamic Reconfiguration ユーザーマニュアル』

『特記事項: Sun Enterprise 6x00/5x00/4x00/3x00 システム』

注意事項 EXX00 システムの CPU/メモリーボードの Dynamic Reconfiguration に関する詳細は、『日本語Solaris7 Sun Enterprise 6x00, 5x00, 4x00, 3x00 システム Dynamic Reconfiguration ユーザーマニュアル』を参照してください。

cfgadm_sysctrl(1M)

名前	cfgadm_sysctrl - EXX00 システムボードの管理
形式	<pre>/usr/sbin/cfgadm -c function [-f] [-o disable-at-boot enable-at-boot] [-n -y] sysctrl0:slot# ... /usr/sbin/cfgadm -x quiesce-test sysctrl0:slot# /usr/sbin/cfgadm -x insert-test remove-test sysctrl0:slot# ... /usr/sbin/cfgadm -x set-condition-test=# sysctrl0:slot# ... /usr/sbin/cfgadm [-l] -o disable-at-boot enable-at-boot sysctrl0:slot# ...</pre>
機能説明	<p>sysctrl1 ハードウェア固有ライブラリ、 /usr/platform/sun4u/lib/cfgadm/sysctrl1.so.1 により、E6X00、E5X00、E4X00、E3X00 システム上で、動的再構成を用いてシステムボードの構成や構成解除ができるようになります。これによりシステムが稼働中であっても再起動せずに、I/O ボード、および CPU ボードを、Solaris 用に設定されたシステムのスロットに挿入できるようになります。また同様に、いずれのタイプのボードも稼働中のシステムから切断および取り外しが可能になり、再起動の必要はなくなります。</p> <p>システムスロットはデバイスツリー上で「接続点 (attachment point)」として表現され、システムシャーシ内の各スロットには、この接続点が 1 つずつ設定されます。スロットにボードが設置されていない場合は、受容体の状態は empty と認識されます。ボードの電源が切断されシステムから取り外せる状態の場合は、受容体の状態は disconnected (接続解除) と認識されます。ボードに電源が投入されシステムバスに接続されている状態であれば、受容体の状態は connected (接続) と認識されます。</p> <p>受容体の状態が empty の場合、その受容体の占有装置は unconfigured の状態にあると認識されます。受容体の状態が connected の場合は、その受容体の占有装置は configured または unconfigured のいずれかの状態を取ります。</p> <p>Solaris でボード上のデバイスを利用できるのは、この状態が configured になっている場合です。unconfigured になっている場合、ボード上のデバイスは利用できません。</p> <p>ボードを挿入すると、受容体の状態は empty から disconnected に変わります。ボードを取り外すと、受容体の状態は disconnected から empty に変わります。connected になっているボードを取り外してしまうと、オペレーティングシステムのクラッシュを起こし、システムに修復不可能な障害を与える場合があります。</p>
オプション	<p>オプションについての詳細は、マニュアルページの <code>cfgadm(1M)</code> の項目を参照してください。</p> <p>サポートしているオプションは、以下の通りです。</p> <pre>-c function</pre> <p>状態の変更に使用します。function の部分に connect、disconnect、configure、unconfigure のいずれかを指定します。</p>

configure

占有装置の状態を `configure` に変更します。

受容体の状態が `disconnected` になっている場合は、`configure` 機能はまず受容体への接続を試みます。`connect` 機能の一部として作成される OBP デバイスツリーを調べて Solaris デバイスノードを作成し、要求のあったデバイスを接続します。CPU/メモリーボードの場合は、電源が切断状態にある CPU のリストに、この CPU を加えます。この情報は `psrinfo(1M)` または `psradm(1M)` コマンドを用いて見ることができます。CPU/メモリーボードにはメモリー接続点が 2 つ作成されます。新規に設置したボード上の I/O デバイスを使えるように設定するには、`mount(1M)` および `ifconfig(1M)` コマンドを利用します。新たなプロセッサをオンラインに設定するには、`psradm -n` コマンドを利用します。メモリーバンクのテストおよび設定には `cfgadm_ac(1M)` を利用します。

connect

受容体の状態を `connected` に変更します。

受容体の状態の変更を要求すると、バス信号の接続中はシステムバスを凍結し、その間にボードをテストします。休止処理によってバスは凍結します。この休止処理の間は、すべてのプロセス活動は停止し、すべてのドライバの動作が中断します。この休止処理およびそれに後続する復元処理にはかなりの時間がかかります。またすべてのドライバがこの機能をサポートしているわけではありません。このため、`-x quiesce-test` というオプションを用意しており、これを使用してシステムバスを凍結させている間に、接続されているボードをファームウェアからテストできます。I/O ボードの場合はこの処理には短時間しか費やしません。CPU/メモリーボードの場合は CPU 外部キャッシュのテストが原因で非常に長時間かかります。ここではメモリーのテストはしません。休止処理を開始する前には、実行の確認を促してきます。オプションに `-y` または `-n` を使用すると、この確認作業を省略できます。`disabled-at-boot` に指定されているボードは、接続処理を拒否します。ただし、指定された処理を強制実行する `-f` フラグを利用、または `-o enable-at-boot` を使って指定を無効化した場合は、接続処理が実行されます。`-1` の解説を参照してください。

disconnect

受容体の状態を `disconnected` に変更します。

占有装置の状態が `configure` になっている場合は、まず占有装置を構成解除します。この `disconnect` では休止処理を実行しないので、短時間で終了します。ボードの電源は切断され、取り外しが可能な状態になります。

unconfigure

占有装置の状態を `unconfigured` に変更します。

この処理の間、ボード上のデバイスは Solaris からは見えない状態になります。I/O ボード上の I/O デバイスは、Solaris のデバイスツリーから削除されます。デバイスが使用中の場合、この `unconfigure` の処理は停止し、使用中であることを報告してきます。この場合はデバイスの使用を停止して、その後で `unconfigure` 処理を再試行する必要があります。CPU/メモリーボードの場合は、ボードの `unconfigure` 処理を実行する前に、メモリーの状態を `unconfigure` に設定しておかなければいけません。ボード上の CPU をオフラインにしてか

cfgadm_sysctrl(1M)

ら、電源を切断して Solaris CPU リストから削除します。接続中のプロセスが存在する CPU をオフラインにすることはできません。CPU のオフライン化についての詳細は、マニュアルページの `psradm(1M)`、`psrinfo(1M)`、`pbind(1M)`、`p_online(2)` の各項目を参照してください。

-f

強制的に、ボードに接続されているブロックの非揮発性変数 `disabled-board-list` を、`disabled-at-boot` に指定します。「特記事項: Sun Enterprise 6x00/5x00/4x00/3x00 システム」のマニュアルも参照してください。

-l

リストを表示させるオプションです。これはマニュアルページの `cfgadm(1M)` で解説しているものと同様の機能です。

`type` フィールドには `cpu/mem`、`mem`、`dual-sbus`、`sbus-upa`、`dual-pci`、`soc+sbus`、`soc+upa`、`disk`、`unknown` のいずれかを指定できます。

ハードウェア固有情報のフィールドは、`[disabled at boot]` `[non-detachable]` `[100 MHz capable]` のように設定されています。

ボードのタイプが `sbus-upa` または `soc+upa` の場合は、まず最初に次のような追加情報が表示されます。

```
[single buffered ffb|double buffered ffb|no ffb installed]
```

ボードのタイプが `disk` の場合は、次のような追加情報が最初に表示されます。

```
{target: # | no disk} {target: # | no disk}
```

-o disable-at-boot | enable-at-boot

非揮発性変数 `disabled-board-list` の変更に使います。この `-o` オプションを使う時は、`-c function` または `-l` オプションと併用します。

`disabled-at-boot` に設定されているボードに関連するブロックを起動時に有効となるようにするには、`-o enable-at-boot` と `-c connect` を併用します。

-x insert-test | remove-test

テストを実行します。

ここで `remove-test` を使うと、ボードを物理的に取り外さなくても、テストシーケンスを自動処理している間、指定したスロットのドライバ状態は `disconnected` から `empty` に変更されます。

また、`insert-test` の方を使った場合は、`remove-test` コマンドで `empty` に指定したスロットのドライバ状態が `isconnected` に変更され、ボードはスロットに挿入されているものとして認識されます。

-x quiesce-test sysctrl0:slot1

テストを実行します。

cfgadm_sysctrl(1M)

このテストの実行により、ボードの接続処理に必要なとなるquiesce処理が実行できる状態になり、現在のソフトウェアおよびハードウェアの構成を保ったままで、システムを確実に休止できるようになります。休止処理できないデバイスやプロセスが存在した場合は、エラーメッセージでその名前を表示します。このコマンドは有効なボード接続点に対してであればどれにでも使用可能ですが、いかなるシステムも必ず slot1 を 1 つ所有しているのので、上に示した形式での使用を推奨しておきます。

`-x set-condition-test=#`
テストを実行します。

状態変更コマンドのポリシーロジックをテストできるように、システムボードの接続点の条件を設定します。新しい設定の内容は、1 から 4 の数字で表現されます。各数字の意味は、以下のようになります。

- 0 unknown (現在の条件が不明)
- 1 ok (障害なし)
- 2 failing (回復可能な障害がある)
- 3 failed (回復不可能な障害がある)
- 4 unusable (接続点が empty)

オペランド 以下のオペランドをサポートしています。

`sysctrl0:slot#` `sysctrl1` ドライバのインスタンス 0 (`sysctrl10`) が、EXX00 システム上のボードに接続点を発行します。接続点の名前には、`slot0` から `slot15` までの番号を割り当てます。# の部分には 0 から 15 までの数字を入れます。この値がスロット番号を示すことになります。この形式は、`cfgadm(1M)` を用いた論理的な `ap_id` の指定と一致します。これに対応する物理的な `ap_id` は、次項の【ファイル】に記載してあります。

ファイル `/usr/platform/sun4u/lib/cfgadm/sysctrl.so.1`
ハードウェア固有ライブラリ

`/devices/central@1f,0/fhc@0,f8800000/clock-board@0,900000:slot*`
接続点

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	SUNWkvm.u

関連項目 `cfgadm(1M)`, `cfgadm_ac(1M)`, `ifconfig(1M)`, `mount(1M)`, `pbind(1M)`, `psradm(1M)`, `psrinfo(1M)`, `config_admin(3CFGADM)`, `attributes(5)`

Sun Enterprise 6x00, 5x00, 4x00 and 3x00 Systems DynamicReconfiguration ユーザーマニュアル,

特記事項:*Sun Enterprise 6x00/5x00/4x00/3x00 Systems*

cfgadm_sysctrl(1M)

注意事項 EXX00 システムの CPU/メモリーボードの動的再構成について、更に詳しい情報が必要な場合は、*Sun Enterprise 6x00, 5x00, 4x00 and 3x00 Systems Dynamic Reconfiguration* ユーザーマニュアルを参照してください。

名前 | install_scripts, add_install_client, add_to_install_server, modify_install_server, rm_install_client, setup_install_server, check – Solaris ソフトウェアをインストールするためのスクリプト

形式 | `cdrom-mnt-pt/Solaris_XX/Tools/add_install_client [-i IP_address] [-e Ethernet_address] [-s server_name : path] [-c server_name : path] [-n [server] : name_service [(netmask)] [-p server_name : path] [-f boot_file_name] host_name platform_group`

`cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server_name:path] [-c server_name:path] [-p server_name:path] [-t install_boot_image_path] [-f boot_file_name] platform_name platform_group`

`cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server_name:path] [-c server_name:path] [-p server_name:path] [-t install_boot_image_path] [-f boot_file_name] -e Ethernet_address [-b property=value] platform_group`

`cdrom-mnt-pt/Solaris_XX/Tools/add_to_install_server [-s] [-p product_image_path] install_server_path`

`cdrom-mnt-pt/Solaris_XX/Tools/jumpstart_sample/check [-p install_dir_path] [-r rulesfile]`

`cdrom-mnt-pt/modify_install_server [-p] install_dir_path installer_miniroot_path`

`cdrom-mnt-pt/Solaris_XX/Tools/rm_install_client host_name`

`cdrom-mnt-pt/Solaris_XX/Tools/setup_install_server [-b] [-t install_boot_image_path] [-w wanboot_image_path] install_dir_path`

機能説明 | これらのコマンドは、Solaris Software and Solaris Installer CD のスライス 0 に置かれています。Solaris CD をすでにローカルディスクにコピーしている場合、`cdrom_mnt_pt` はコピーされている Solaris CD へのパスになります。これらのコマンドは、さまざまなインストール作業に使用できます。

Solaris_XX の XX は、使用している Solaris リリースのバージョン番号です。

`add_install_client` コマンドには 3 種類の形式があります。「形式」の項を参照してください。

ネットワークを使用してインストールするクライアントを追加する場合は、次の形式の `add_install_client` を使用します (これらのコマンドは `bootparams(4)` ファイルを更新します)。`add_install_client` コマンドは、インストールサーバーの Solaris インストールイメージ (マウントした CD またはディスクにコピーした Solaris CD) あるいはブートサーバーの起動ディレクトリ (ブートサーバーが必要な場合) から実行する必要があります。Solaris インストールイメージまたはブート専用ディレクトリの Solaris リリースは、クライアントにインストールする予定の Solaris リリースと同じでなければなりません。

`cdrom-mnt-pt/Solaris_XX/Tools/add_install_client [-i IP_address] [-e Ethernet_address] [-s server_name : path] [-c server_name : path]`

check(1M)

```
[ -n [server] ] : name_service [ ( netmask) ] [ -p server_name : path ]  
host_name platform_group
```

プラットフォームグループ内のあるプラットフォームのインスタンスをインストールサーバーに追加する場合は、次の形式の `add_install_client` コマンドを使用します。このグループは DHCP を使用して起動および構成されます。スクリプトは、必要な構成作業をサーバー上で実行し、ユーザーがそのグループのために DHCP サーバーに追加しなければならないデータを出力します。

```
cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server:path]  
[-c server:path] [-p server:path] [-t install boot image path]  
[-f boot file name] platform_name platform_group
```

インストールサーバーに単一のクライアントを追加する場合は、次の形式の `add_install_client` コマンドを使用します。このクライアントは DHCP を使用して起動および構成されます。スクリプトは、必要な構成作業をサーバー上で実行し、ユーザーがそのクライアントのために DHCP サーバーに追加しなければならないデータを出力します。すでに使用しているものにも、上記のように `-f` フラグを追加する必要があります。`-f` を使用すると、ユーザーは所定のクライアントに使用する起動ファイルの名前を指定できます。

```
cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d  
[-s server_name:path] [-c server_name:path] [-p server_name:path]  
[-t install_boot_image_path] [-f boot_file_name] -e Ethernet_address  
platform_group
```

注 - x86 アーキテクチャの Pre-boot eXecution Environment (PXE) クライアントを登録する場合は、必ず `-d` オプションを使用してください。x86 PXE クライアントは構成に DHCP を使用します。

他の Solaris CD とネットワークインストールサーバー上の既存のイメージをマージする場合は、`add_to_install_server` を使用します。マージ可能な各 CD (現時点では OS CD 2 および Language CD) には専用の `add_to_install_server` スクリプトが含まれています。配布された CD に含まれているもの以外の `add_to_install_server` スクリプトを使用しないでください。

`rules` ファイル (カスタム JumpStart インストールを使用する場合に限り必要) のルールを検証する場合は、`check` を使用します。

既存のネットワークインストールサーバーのミニルートを Solaris Installation CD のミニルートに置き換える場合は、`modify_install_server` を使用します。ネットワークインストールサーバーのインストール時のユーザーインターフェースが、Solaris Installation CD の Web Start ユーザーインターフェースに変更されません。

`modify_install_server` コマンドを使用する場合は、既存のインストールイメージ (`setup_install_server` を使用して作成) があらかじめ存在していなければなりません。

ネットワークインストール用のクライアントを削除する場合は、`rm_install_client` を使用します (このコマンドは `bootparams(4)` ファイルを更新します)。

Solaris CD を (インストールサーバーを設定するために) ディスクにコピーする、(WANboot インストールサーバーを設定するために) WANboot ミニルートイメージを構築する、または Solaris CD のブートソフトウェアだけを (ブートサーバーを設定するために) ディスクにコピーする場合は、`setup_install_server` を使用します。ネットワーク経由でクライアントをインストールするには、インストールサーバーが必要です。インストールサーバーとインストールするクライアントが異なるサブネットに属している場合、ネットワークインストールを行うには、ブートサーバーも必要です (ブートサーバーはクライアントのサブネットに配置する必要があります)。

オプション `add_install_client` には、次のオプションを指定できます。

-b *property=value*

ブートサーバーの TFTP ディレクトリ (デフォルトは `/tftpboot`) にあるクライアント特有の `bootenv.rc` ファイルにプロパティ値を設定します。クライアントに特有のブートプロパティを設定する場合には、このオプションを使用します。`bootpath` プロパティを使用すると、デバイス構成補助を使用して手動で選択しなくても、ブートデバイスを自動的に選択できます。`boot-args` プロパティを使用すると、JumpStart インストールを自動的に選択できます。`eeeprom(1M)` を参照してください。

このオプションは x86 クライアントにのみ使用できます。このオプションを使用するときには、必ず `-d` オプションと `-e` を同時に指定してください。

-c *server_name:path*

このオプションが必要なのは、カスタム JumpStart インストール用の JumpStart ディレクトリを指定する場合だけです。`server_name` は、JumpStart ディレクトリが置かれているサーバーのホスト名です。`path` は JumpStart ディレクトリの絶対パスです。

-d

DHCP クライアントを指定します。

-e *Ethernet_address*

インストールするシステムの Ethernet アドレスを指定します。

-f

インストールするクライアントの `boot_file_name` を指定します。

-i *IP_address*

インストールするクライアントの IP アドレスを指定します。

-n [*server*]: *name_service*[(*netmask*)]

このオプションでは、システム構成時に使用するネームサービスを指定します。`bootparams(4)` ファイル内の `ns` キーワードを設定します。

name_service

有効なエントリは `nis`、`nisplus`、および `none` です。

check(1M)

netmask

数字を4つずつピリオドで区切って並べたもので、IPアドレスのどの部分がネットワーク部分で、どれがホスト部分かを指定します。

server

サーバーの名前または指定したネームサービスのIPアドレス。指定したサーバーが異なるサブネット上にある場合は、クライアントからサーバーにアクセスするために、*netmask*が必要なことがあります。

-p *server_name*: *path*

このオプションは、システム情報またはネットワーク情報を前もって設定するために必要な、ユーザーが定義した *sysidcfg* ファイルの位置です。*server_name* は有効なホスト名またはIPアドレスです。*path* はJumpStart ディレクトリの絶対パスです。

-s *server_name*:*path*

このオプションが必要なのは、ブートサーバーから *add_install_client* を使用する場合だけです。サーバー名およびこのインストールで使用される Solaris インストールイメージの絶対パスを指定します。*path* はマウントされている Solaris CD のパスまたは Solaris CD のコピーが置かれているディレクトリのパスです。

add_to_install_server コマンドには、次のオプションを指定できます。

-p

コピーする CD (補助的な製品が収められている) の位置を指定します。

-s

ユーザーが、インストールする必要がある製品だけをリストから選択できるようにします。

check コマンドには、次のオプションを指定できます。

-p *install_dir_path*

使用中のシステムの *check* スクリプトではなく、指定された Solaris インストールイメージの *check* スクリプトを使用することによって、ルールファイルを検査します。*install_dir_path* は、ローカルディスク上またはマウントした Solaris CD 上の Solaris インストールイメージのパスです。

旧バージョンの Solaris が稼働しているシステムでは、このオプションを使用して、最新バージョンの *check* を実行できます。

-r *rulesfile*

rules 以外の名前のルールファイルを指定します。このオプションを使用すると、ルールの妥当性を検証してから、ルールファイルに組み込むことができます。*check* はルールが有効であるかどうかを報告するだけで、カスタム JumpStart インストールに必要な *rules.ok* ファイルは作成しません。

modify_install_server コマンドには、次のオプションを指定できます。

-p

このオプションは、*install_dir_path/Solaris_XX/Tools/Boot.orig* にすでに存在しているイメージのミニルートを保持します。

setup_install_server コマンドには、次のオプションを指定できます。

- b
サーバーの設定を、ブートサーバー専用にします。
- t
代替ミニルート指定できるようにします。
- w
WANboot ミニルートイメージを構築します。

オペランド add_install_client コマンドには、次のオペランドを指定できます。

host_name

インストールするクライアントの名前です。

platform_group

特定のソフトウェアを配布するために、ベンダーが定義したハードウェアプラットフォームグループ。以下に有効なプラットフォームグループの例を示します。

システム	プラットフォームグループ
x86	i86pc
Sun Fire 4800	sun4c

システムのプラットフォームグループを調べるには、(-m オプションを指定して) uname(1) コマンドを使用します。

platform_name

システムのプラットフォーム名を調べるには、-i オプションを指定して uname(1) コマンドを使用します。

次に、uname command コマンドを使用して Ultra 1 のシステムプラットフォーム名を調べる例を示します。

```
uname -i
システムは、次のように応答します。
```

```
SUNW,Ultra-1
システムのプラットフォーム名は SUNW,Ultra1 です。
```

次のコマンドは、Ultra 1 用の add_install_client を呼び出します。

```
add_install_client -d SUNW,Ultra-1 sun4u
```

IA32 プラットフォームでは、プラットフォーム名はつねに SUNW.i86pc です。

次のコマンドは、IA32 プラットフォーム用の add_install_client を呼び出します。

```
add_install_client -d SUNW.i86pc i86pc
```

rm_install_client コマンドには、次のオペランドを指定できます。

check(1M)

host_name

削除するクライアントの名前です。

setup_install_server コマンドには、次のオペランドを指定できます。

install_dir_path

Solaris ソフトウェアのコピー先となるディレクトリの絶対パスです。このディレクトリは空でなければなりません。

wanboot_image_path

WANboot ミニルートイメージを含むファイルが作成されるディレクトリの絶対パスです。

使用例 例 1 add_install_client の使用

次の add_install_client コマンドは、マウント済みの Solaris CD からネットワークインストール用のクライアントを Solaris 8 のインストールサーバーに追加します。

```
example# cd /cdrom/cdrom0/s0/Solaris_9/Tools
example# ./add_install_client system_2/sun4u
```

例 2 add_install_client の使用

次の add_install_client コマンドは、ネットワークインストール用のクライアントをインストールサーバーに追加します。カスタム JumpStart インストールを実行するためのルールファイルおよびプロファイルファイルが置かれている JumpStart のサーバーとディレクトリのパスを -c オプションで指定します。Solaris CD は Solaris 8 上の /export/install ディレクトリにコピーされています。

```
example# cd /export/install/Solaris_8/Tools
example# ./add_install_client -c install_server:/jumpstart system_1 i86pc
example# ./add_install_client -c install_server:/jumpstart system_2 i86pc
```

例 3 add_install_client の使用

次の add_install_client コマンドは、起動ファイル sun4u.solaris8 を使用する特定の sun4u プラットフォームマシン (8:0:20:99:88:77) に対するサポートを追加します。

```
example# add_install_client -d -f sun4u.solaris8 -e 8:0:20:99:88:77 sun4u
```

例 4 add_install_client の使用

次の add_install_client コマンドは、PXE 規格を使用してネットワークから起動する x86 クライアントを追加します。

```
example# add_install_client -d -s svrname:/mnt/export/root SUNW.i86pc i86p
```

例 5 add_to_install_server の使用

次の add_to_install_server コマンドは、CD 上のすべての製品ディレクトリにあるパッケージを Solaris 8 上の既存のインストールサーバーにコピーします。

例 5 add_to_install_server の使用 (続き)

```
example# cd /cdrom/cdrom0/s0
example# ./add_to_install_server /export/Solaris_8
```

例 6 check の使用

次の check コマンドは、カスタム JumpStart インストールに使用するルールファイルの構文を検査します。

```
example# cd jumpstart_dir_path
example# ./check -p /cdrom/cdrom0/s0
```

例 7 modify_install_server の使用

次の modify_install_server コマンドは、上記の setup_install_server を使用して作成されたミニルートを Boot.orig に移し、Solaris Installer CD 上のミニルートに置き換えます。

```
example# cd /cdrom/cdrom0/s0
example# ./modify_install_server -p /export/install /cdrom/cdrom0/s1
```

例 8 modify_install_server の使用

次の modify_install_server コマンドは、上記の setup_install_server を使用して作成されたミニルートを Solaris Installer CD 上のミニルートに置き換えます。

```
example# cd /cdrom/cdrom0/s0
example# ./modify_install_server /export/install /cdrom/cdrom0/s1
```

例 9 rm_install_client の使用

次の rm_install_client コマンドは、Solaris 8 上のネットワークインストール用クライアントを削除します。

```
example# cd /export/install/Solaris_8/Tools
example# ./rm_install_client holmes
example# ./rm_install_client watson
```

例 10 setup_install_server の使用

次の setup_install_server コマンドは、マウント済みの Solaris CD を Solaris 8 のローカルディスク上にある /export/install というディレクトリにコピーします。

```
example# cd /cdrom/cdrom0/s0/Solaris_8/Tools
example# ./setup_install_server /export/install
```

例 11 setup_install_server の使用

次の setup_install_server コマンドは、マウント済みの Solaris CD のブートソフトウェアを、Solaris 8 上のサブネット用ブートサーバーになるシステムの /boot_dir というディレクトリにコピーします。

例 11 setup_install_server の使用 (続き)

```
example# cd /cdrom/cdrom0/s0/Solaris_8/Tools
example# ./setup_install_server -b /boot_dir
```

例 12 setup_install_server の使用

setup_install_server はデフォルトで、マウント済みの Solaris 配布ディスク上の Solaris ../Tools/Boot にあるインストールブートディレクトリを検索します。

以前に ./setup_install_server -b /boot_dir コマンドを使用して、ネットワークブートサーバー上に作成したディレクトリと別のディレクトリが必要な場合は、-t オプションを使用できます。

```
example# cd /cdrom/cdrom0/s0/Solairs_9/Tools
example# ./setup_install_server -t /boot_dir /export/install
```

例 13 setup_install_server と WANboot オプションの使用

次の setup_install_server コマンドは WANboot ミニルートファイルシステムのイメージを作成し、それを /wanboot_dir/miniroot に保存します。

```
example# cd /cdrom/cdrom0/s0/Solairs_9/Tools
example# ./setup_install_server -w /wanboot_dir /export/install
```

例 14 x86: ネットワークインストール時に使用するシリアルコンソールの指定 (CD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するシリアルコンソールを指定する方法を示します。この例では、インストールクライアントを次の方法で設定します。

- -d オプションは、クライアントが DHCP を使用してインストールパラメータを設定することを示します。
- -e オプションは、Ethernet アドレスが 00:07:e9:04:4a:bf であるクライアントでのみインストールが行われることを示します。
- 1 つめと 2 つめの -b オプションは、インストールプログラムが入力デバイスおよび出力デバイスとしてシリアルポート ttya を使用することを示します。

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
i86pc
```

-b オプションで使用できるブートプロパティの変数とその値については、eprom(1M) を参照してください。

例 15 ネットワークインストール時に使用するブートデバイスの指定 (CD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するブートデバイスを指定する方法を示します。インストールクライアントを設定するときにブートデバイスを指定する場合は、インストール時にデバイス構成補助によって出されるこの情報は表示されません。

例 15 ネットワークインストール時に使用するブートデバイスの指定 (CD) (続き)

この例では、次の方法でインストールクライアントを設定します。

- -d オプションは、クライアントが DHCP を使用してインストールパラメータを設定することを示します。
- -e オプションは、Ethernet アドレスが 00:07:e9:04:4a:bf であるクライアントでのみインストールが行われることを示します。
- 1 つめと 2 つめの -b オプションは、インストールプログラムが入力デバイスおよび出力デバイスとしてシリアルポート ttya を使用することを示します。
- 3 つめの -b オプションは、インストールプログラムがインストール時に特定のブートデバイスを使用することを示します。
- ブートデバイスのパスは、使用しているハードウェアによって異なります。
- i86pc というプラットフォーム名は、クライアントが x86 ベースのシステムであることを示します。

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
-b "bootpath=/pci@0,0/pci108e,16a8@8" i86pc
```

-b オプションで使用できるブートプロパティの変数とその値について詳細は、eeprom(1M) を参照してください。

終了ステータス 次の終了ステータスが返されます。

```
0      正常終了
1      エラーが発生した
```

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	Solaris CD

関連項目 uname(1), bootparams(4), attributes(5)

『Solaris 9 インストールガイド』

closewtmp(1M)

名前	acct, acctdisk, acctdusg, accton, acctwtmp, closewtmp, utmp2wtmp – アカウンティングおよびいろいろなアカウントコマンドの概要
形式	<pre>/usr/lib/acct/acctdisk /usr/lib/acct/acctdusg [-u filename] [-p filename] /usr/lib/acct/accton [filename] /usr/lib/acct/acctwtmp reason filename /usr/lib/acct/closewtmp /usr/lib/acct/utmp2wtmp</pre>
機能説明	<p>アカウンティングソフトウェアは、アカウンティングシステムを構築するためのツール群です (C 言語のプログラムおよびシェルプロシージャから構成される)。acctsh(1M) では、C 言語のプログラムの一番上に組むシェルプロシージャ群を説明します。</p> <p>接続時間のアカウンティングは記録を /var/adm/wtmpx (utmpx(4) を参照) に書き込むプログラム群により管理されます。acctcon(1M) では、このファイルをセッションおよび課金記録に変換するプログラムを説明します。また、acctmerg(1M) では、この課金記録について集約します。</p> <p>プロセスアカウンティングは、システムカーネルによって行われます。プロセスを終了すると、1つのプロセスにつき1つのレコードが、あるファイル (通常は、/var/adm/pacct) に書き込まれます。acctprc(1M) では、課金のためにこのデータを集約します。コマンドの使用状況を集約するには、acctcms(1M) を使用します。acctcom(1) を用いて、現在のプロセスデータを調査することができます。</p> <p>acctmerg (acct(3HEAD) の tacct フォーマットを参照) を使うと、プロセスのアカウンティングレコードおよび接続時間のアカウンティングレコード (あるいは acct(3HEAD) で説明する tacct フォーマットのアカウンティングレコードのいずれか) をアカウンティングレコードの合計にマージしたり、集約したりすることができます。どのアカウンティングレコードにも、あるいはすべてのアカウンティングレコードにでも prtacct (acctsh(1M) を参照) をフォーマットに使用します。</p> <p>acctdisk は、ユーザー ID、ログイン名およびディスクブロック数がある行を読み取り、他のアカウンティングレコードとマージすることができるアカウンティングレコードの合計に、それらの情報を変換します。入力ファイルが壊れているか、または正しくフォーマットされていない場合は、acctdisk はエラーを返します。</p> <p>acctdusg は、その標準入力を (通常は、find / -print から) 読み取り、ログインによるディスクの資源消費 (間接ブロックを含む) を計算します。</p> <p>accton では、引数がない場合、プロセスアカウンティングをオフにします。filename を指定する場合は、そのファイルは、カーネルがプロセスのアカウンティングレコード (acct(2) および acct(3HEAD) を参照) を追加するための、既存のファイル名である必要があります。</p>

closewtmp(1M)

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

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

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

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

オプション 以下のオプションを指定できます。

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

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

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

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

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

closewtmp(1M)

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

属性 次の属性については attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWaccu

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

『Solaris のシステム管理 (基本編)』

名前	cvcd – 仮想コンソールデーモン						
形式	/platform/platform_name/cvcd						
機能説明	<p>仮想コンソールデーモン cvcd は、一部のプラットフォームで提供されているネットワークコンソールをサポートするサーバープロセスです。cvcd デーモンは、遠隔ホストからのネットワークコンソールの接続を受け付けます (一回につき 1 つのホストのみ)。コンソールの入力はこの接続から読み取られ、cvcredir(7D) を経由して cvc(7D) へ転送されます。</p> <p>同様に、コンソールの出力は cvcredir(7D) から読み取られ、ネットワークコンソールの接続を介して転送されます。cvcd が終了した場合は、コンソールトラフィックは自動的に内部ハードウェアインタフェースによってルート変更されます。</p> <p>通常、cvcd デーモンはシステム起動時に起動されます。各ドメインで同時に実行することができる cvcd プロセスは、1 つだけです。</p> <p>注意: Sun Enterprise 10000 ドメインでは、cvcd は構成ファイル (/etc/ssphostname) を使用して、ネットワークコンソールの接続が許可されるホストの名前を決定します。遠隔コンソールホストの名前が変更された場合は、構成ファイルを編集し、その変更点を反映させる必要があります。</p>						
オペランド	<p>以下のオペランドがサポートされています。</p> <p><i>platform_name</i> パッケージおよびコードで使用されている、Sun の正式なプラットフォーム名。たとえば、Sun Fire 15000 サーバーの場合は <i>platform_name</i> は SUNW, Sun-Fire-15000 になります。</p>						
属性	以下の属性については、attributes(5) を参照してください。						
	<table border="1"> <thead> <tr> <th>属性タイプ</th> <th>属性値</th> </tr> </thead> <tbody> <tr> <td>アーキテクチャ</td> <td>Sun Enterprise 10000 サーバー、Sun Fire High-End Systems</td> </tr> <tr> <td>使用条件</td> <td>SUNWcvc.u</td> </tr> </tbody> </table>	属性タイプ	属性値	アーキテクチャ	Sun Enterprise 10000 サーバー、Sun Fire High-End Systems	使用条件	SUNWcvc.u
属性タイプ	属性値						
アーキテクチャ	Sun Enterprise 10000 サーバー、Sun Fire High-End Systems						
使用条件	SUNWcvc.u						
関連項目	<p>services(4), attributes(5), cvc(7D), cvcredir(7D)</p> <p>『Sun Enterprise 10000 SSP リファレンスマニュアル』</p> <p>System Management Services (SMS) リファレンスマニュアル</p>						

dd(1M)

名前	dd - ファイルの変換とコピー
形式	<code>/usr/bin/dd [operand=value...]</code>
機能説明	<p>dd は、指定した入力ファイルに可能な変換を行なって、指定した出力へコピーします。デフォルトでは、標準入力および標準出力が使用されます。raw 入出力装置の特性を利用するために入出力のブロックサイズで指定することが可能です。サイズは、バイト単位で指定し、数字の後に k、b、または w を付加することができ、それぞれ 1024、512、または 2 の倍数として指定します。また、数字を x で区切ることで乗算を表すことができます。</p> <p>dd は、指定された入力ブロックサイズを用いて、入力データを 1 ブロックずつ読み込みます。その後、実際に渡されたデータブロックを処理します。そのサイズは、指定されたブロックサイズより小さい場合があります。dd は指定された変換処理をブロックに対して行い、結果のデータを、指定された出力ブロックサイズに従ってブロック単位で書き出します。</p> <p>cbs は、ascii、asciib、unblock、ebcdic、ebcdicb、ibm、ibmb、または block 変換が指定されている場合にかぎり使用されます。最初の 2 つの指定では、cbs 文字は変換バッファにコピーされ、任意の指定文字のマッピングが行われます。また後続の空白文字は切り捨てられ、行を送信する前に復帰改行が追加されます。残りの 3 つの指定では、復帰改行までの文字が変換バッファに読み込まれ、サイズ cbs の出力レコードを構成するために空白文字が追加されます。ASCII ファイルは復帰改行文字を含むものとします。cbs が指定されていないかまたは 0 であると、ascii、asciib、ebcdic、ebcdicb、ibm、ibmb の各オプションは入力ファイルのブロック構造を変更せずに文字セットを変換します。unblock および block の各オプションは単純なファイルコピーを行います。</p> <p>終了後に、dd は全体および部分的な入出力ブロック数を報告します。</p>
オペランド	<p>以下のオペランドが指定できます。</p> <p><code>if=file</code> 入力パス名。デフォルトは標準入力です。</p> <p><code>of=file</code> 出力パス名。デフォルトは標準出力です。seek=expr 変換が指定されていないとき、conv=notrunc も指定されていない場合は、コピー処理の実行前に出力ファイルは切り捨てられます。seek=expr が指定され、conv=notrunc が指定されていない場合は、コピー処理の結果としては dd がシークする出力ファイル中のブロックは保持されますが、出力ファイルのその他の部分は保持されません。シークするサイズと入力ファイルのサイズの合計が出力ファイルの元のサイズより小さい場合、コピー処理により出力ファイルは小さくなります。</p> <p><code>ibs=n</code> 入力ブロックサイズを n バイトとします。デフォルト値は 512 です。</p> <p><code>obs=n</code> 出力ブロックサイズを n バイトとします。デフォルト値は 512 です。</p>

bs=*n*

入力ブロックサイズと出力ブロックサイズをともに *n* バイトとします。この指定は **ibs=** と **obs=** 指定よりも優先されます。**sync**、**noerror**、**notrunc** 以外の変換が 1 つも指定されない場合、各入力ブロックは複数の短いブロックを 1 つにまとめる処理は行われず、それぞれ単独のブロックとして出力側にコピーされます。

cbs=*n*

block と **unblock** 用の変換ブロックサイズを *n* バイトに指定します。デフォルト値は 0 です。**cbs=** を指定しないかまたは 0 を指定した場合、**block** や **unblock** を使うと結果は予測できません。

このオプションは ASCII または EBCDIC 変換を指定した場合にだけ有効です。**ascii** と **asciib** オペランドを指定した場合、後続の空白文字を消去する前に文字が ASCII に変換されるという点を除き、入力処理は **unblock** オペランド指定時と同じです。**ebcdic**、**ebcdicb**、**ibm**、または **ibmb** オペランドを指定した場合、後方に空白文字を追加した後で文字が EBCDIC または IBM EBCDIC に変換されるという点を除き、入力処理は **block** オペランド指定時と同じです。

files=*n*

終了する前に、*n* 個の入力ファイルをコピーして連結します (入力が磁気テープまたは同様な装置の場合だけ有効です)。

skip=*n*

コピーを開始する前に、指定された入力ブロックサイズを用いて *n* 個の入力ブロックを読み飛ばします。シーク可能なファイルに対しては、システムはそれらのブロックを読み込むか、あるいはシークを行います。シーク不可能なファイルに対しては、ブロックを読み込んで、そのデータを捨てます。

iseek=*n*

コピーを行う前に、入力ファイルの先頭から *n* 個のブロックをシークします (**skip** の動作が遅いディスクファイルに適しています)。

oseek=*n*

コピーを行う前に、出力ファイルの先頭から *n* 個のブロックをシークします。

seek=*n*

コピーを行う前に、出力ファイルの先頭から *n* 個のブロックをスキップします (指定された出力ブロックサイズを使用)。シーク不可能なファイルに対しては、既存のブロックを読み込み、現在のファイルの終わり位置から指定されたオフセット位置までの間に空白があれば、その空白を NULL バイトで埋めます。シーク可能なファイルに対しては、指定されたオフセット位置までをシークするか、またはシーク不可能なファイルの場合と同様にブロックを読み込みます。

count=*n*

n 個の入力ブロックだけをコピーします。

conv=*value*[, *value*...]

1 つ以上の *value* をコンマで区切って記述します。各 *value* は以下のいずれかです。

ascii	EBCDIC を ASCII に変換します。
asciib	BSD 互換の文字変換を使用して、EBCDIC を ASCII に変換します。

dd(1M)

<code>ebcdic</code>	ASCII を EBCDIC に変換します。復帰改行のない固定長の ASCII レコードを変換する場合は、前もって <code>dd conv=unblock</code> でパイプラインを整えてください。
<code>ebcdicb</code>	BSD 互換の文字変換を使用して、ASCII を EBCDIC に変換します。復帰改行のない固定長の ASCII レコードを変換する場合は、前もって <code>dd conv=unblock</code> でパイプラインを整えてください。
<code>ibm</code>	ASCII から EBCDIC への変換とわずかに異なるマップを使用します。復帰改行のない固定長の ASCII レコードを変換する場合は、前もって <code>dd conv=unblock</code> でパイプラインを整えてください。
<code>ibmb</code>	BSD 互換の文字変換を使用して ASCII から EBCDIC への変換とわずかに異なるマップを使用します。復帰改行のない固定長の ASCII レコードを変換する場合は、前もって <code>dd conv=unblock</code> でパイプラインを整えてください。
<code>ascii</code> (または <code>asciib</code>)、 <code>ebcdic</code> (または <code>ebcdicb</code>)、 <code>ibm</code> (または <code>ibmb</code>) のうち 2 つ以上を同時に指定することはできません。	
<code>block</code>	入力データを、入力ブロック境界に関係なく、復帰改行文字もしくは EOF で終わる可変長レコードの集まりと見なします。各レコードは、変換ブロックサイズで指定した長さの固定長レコードに変換されます。入力行に復帰改行文字があれば削除されます。変換ブロックサイズより短い行に関しては、ブロックを埋めるために空白文字が付加されます。変換ブロックサイズより長い行に関しては、ブロックサイズを満たす最大長になるように文字が捨てられます。切り捨てられた行の総数が報告されます。
<code>unblock</code>	固定長のレコードを可変長に変換します。変換ブロックサイズと等しい長さ分のバイト (それより短いデータしか残っていないければそのすべて) を読み込み、後続の空白文字を削除して、復帰改行文字を付加します。
<code>block</code> と <code>unblock</code> を同時に指定することはできません。	
<code>lcase</code>	LC_CTYPE カテゴリ中のキーワードの <code>tolower</code> で指定された大文字を、対応する小文字にマップします。マッピングを指定されなかった文字は、この変換によって変更されることはありません。
<code>ucase</code>	LC_CTYPE カテゴリ中のキーワードの <code>toupper</code> で指定された小文字を、対応する大文字にマップします。マッピングを指定されなかった文字は、この変換によって変更されることはありません。
<code>lcase</code> と <code>ucase</code> を同時に指定することはできません。	

swab	入力データを2バイトずつの対になっているものとし、各々の対についてバイトの値を交換します。入力レコードの長さが奇数バイトの場合には、最終バイトは無視されます。
noerror	入力エラーが発生しても処理を停止しません。入力エラーが起これば、標準エラー出力に診断メッセージと、入力および出力ブロック数が出力されます。ブロック数の形式は、正常に終了したときに出力されるものと同じです。sync 変換が指定されたときは、入力データのうち得られなかったバイトは NULL バイトに置き換えられて通常どおりに処理されます。sync が指定されなければ、入力ブロックは出力上には現れません。
notrunc	出力ファイルを切り捨てません。今回の dd 呼び出しで明示的に書き出されなかったブロックも出力ファイル中に保持します(前述の of=file オペランドの説明を参照)。
sync	入力ブロックに NULL バイトを付加して、サイズが ibs= で指定した値に等しくなるようにします。なお block または unblock も指定されている場合には、NULL バイトの代わりに空白文字を付加します。

conv= 以外のオペランドが複数回指定されたときは、最後に記述された operand=value が有効となります。

bs=、cbs=、ibs=、obs= に関しては、バイト単位でサイズを指定する式をアプリケーション側で提供しなければなりません。式 *expr* は以下のいずれかです。

1. 正の 10 進数。
2. 正の 10 進数の後に 1024 倍を示す k を付加したもの。
3. 正の 10 進数の後に 512 倍を示す b を付加したもの。
4. 2 つ以上の正の 10 進数 (k や b を付加してもしなくてもよい) を文字 x で区切ったもの。その 2 つの値を乗算した結果を表す。

オペランドはすべて、入力データを読み込む前に処理されます。

使用法 ファイルが 2G バイト (2³¹ バイト) 以上ある場合の dd の動作については、largefile(5) を参照してください。

使用例 例 1 テープドライブ 0 から 1 へのコピー

次のコマンドは、標準的なデバイス命名規約を用いて、テープドライブ 0 から 1 へコピーします。

```
example% dd if=/dev/rmt/0h of=/dev/rmt/1h
```

例 2 標準入力の先頭の 10 バイトの削除

次のコマンドは、標準入力の先頭の 10 バイトを削除します。

```
example% dd ibs=10 skip=1
```

dd(1M)

例 2 標準入力の前頭の 10 バイトの削除 (続き)

例 3 テープを ASCII ファイルに読み込む

次のコマンドは、テープブロックごとに 10 個の 80 バイト EBCDIC カードイメージにブロック化した EBCDIC テープを ASCII ファイル `x` に読み込みます。

```
example% dd if=/dev/tape of=x ibs=800 cbs=80 conv=ascii,lcase
```

例 4 テープの書き込みに `conv=sync` を使用

次のコマンドは、テープに書き込む場合に `conv=sync` を使用する例です。

```
example% tar cvf - . | compress | dd obs=1024k of=/dev/rmt/0 conv=sync
```

環境 dd の実行に影響を与える環境変数 `LC_CTYPE`、`LC_MESSAGES`、`NLSPATH` についての詳細は、`environ(5)` を参照してください。

終了ステータス 以下の終了ステータスが返されます。

0 ファイルは正常にコピーされた

>0 エラーが発生した

`noerror` 変換が指定されていないときに入力エラーが発生すると、部分的に生成された出力ブロックがあればそれを出力ファイルに書き出し、診断メッセージを出力し、コピー処理を中止します。その他のエラーを検出した場合には、診断メッセージを出力し、コピー処理を中止します。

属性 次の属性については `attributes(5)` のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 `cp(1)`、`sed(1)`、`tr(1)`、`attributes(5)`、`environ(5)`、`largefile(5)`

診断 `f+p records in(out)` 読み取られた (書き込まれた) 完全 (f) および部分 (p) ブロック数

注意事項 ブロックサイズが異なるファイルシステム間でファイルのコピーを行う場合には、`dd` を使用しないでください。

ファイルをコピーするのにブロック型デバイスを使用すると、最終ブロックをブロック境界にあわせるために余分な NULL バイトがファイルに追加されます。

`dd` が `ibs=X` および `obs=Y` オペランドを用いてパイプから読み取ると、その出力はつねにサイズ Y の固まりでブロック化されます。`bs=Z` が使用されると、出力ブロックはその時点でパイプから読み取ることができる大きさになります。

dd(1M)

dd を使用してテープデバイスにファイルをコピーする場合、ファイルのサイズはデバイスのセクタサイズ (たとえば 512K) の倍数でなければなりません。任意のサイズのファイルをテープデバイスにコピーする場合は、tar(1) または cpio(1) を使用してください。

SIGINT が発生した場合、dd はステータス情報を標準エラー出力に書き出して処理を終了します。その他のシグナル発生時には、標準的な動作を行います。

df(1M)

名前	df - 使用可能なディスクブロックおよびファイル数の表示
形式	<pre> /usr/bin/df [-F FSType] [-abeghklntV] [-o FSType-specific_options] [block_device directory file resource ...] /usr/xpg4/bin/df [-F FSType] [-abeghklntPtV] [-o FSType-specific_options] [block_device directory file resource ...] </pre>
機能説明	<p>df コマンドは、マウントされているもしくはマウントされていないファイルシステムが使用しているディスク容量を表示します。さらに、使用中の容量、使用可能な容量、ファイルシステムの全容量のうちどの程度が使用されたかを表示します。対象のファイルシステムとしては、デバイス、または特定のファイルシステム上のファイルあるいはディレクトリを指定します。</p> <p>オペランドもオプションも指定せずに df を実行すると、全ファイルシステムに関する情報が出力されます。</p> <p>df はすべての <i>FSTypes</i> (ファイルシステムタイプ) をサポートしません。</p>
オプション	<p>以下のオプションは、<code>/usr/bin/df</code> と <code>/usr/xpg4/bin/df</code> で指定できます。</p> <p>-a <code>/etc/mnttab</code> のエントリ (<code>mnttab(4)</code> を参照) に <code>ignore</code> オプションセットを含む ファイルシステムについても情報を出力しません。</p> <p>-b 使用可能な容量の合計を、K バイト単位で報告します。</p> <p>-e 使用可能なファイル数だけを報告します。</p> <p>-F <i>FSType</i> df コマンドが動作する <i>FSType</i> を指定します。-F オプションはマウントされていないファイルシステムを使用するためのオプションです。<i>FSType</i> は、このオプションで指定するか、あるいは <code>/etc/vfstab</code> 内の記述 (<code>vfstab(4)</code> を参照) から特定できるようにしておく必要があります。後者の場合、具体的には <i>directory</i>、<i>block_device</i>、または <i>resource</i> 引数の指定値とテーブル中のエントリの値を一致させるか、あるいは <code>/etc/default/fs</code> を参照して決定されます。詳細は <code>default_fs(4)</code> を参照してください。</p> <p>-g <code>statvfs(2)</code> 構造体全体を報告します。このオプションはマウントされているファイルシステムに対してのみ有効であり、-o オプションと同時に指定することはできません。このオプションは、-b、-e、-k、-n、-P、-t の各オプション (指定されている場合) よりも優先されます。</p> <p>-h -k と似ていますが、サイズ情報がより読みやすい形式で表示されます。1つのファイルシステムにつき1行の情報が出力されます。情報の内容は、ファイルシステム名、そのファイルシステムに割り当てられている容量の合計、既存のファイルに割り当てられている容量の合計、特権を持たないユーザーが新たなファイルを生成する場合に使用できる容量の合計、そのファイルシステム上の全ファイルに現在割り当てられている通常使用可能な容量の割合 (パーセント単位) です。すべてのサイズは縮小され、読みや</p>

df(1M)

すい形式で表示されます。たとえば、14K、234M、2.7G、3.0T などのようになります。縮尺は、1024 を除数として行われます。

このオプションは `-b`、`-e`、`-g`、`-k`、`-n`、`-t`、および `-V` オプションを無効にします。このオプションはマウントされているファイルシステム上でのみ有効であり、`-o` オプションと一緒に使用できません。

- `-k` 割り当てられているディスク容量を K バイト単位で出力します。1 つのファイルシステムにつき 1 行の情報が出力されます。情報の内容は、ファイルシステム名、そのファイルシステムに割り当てられている容量の合計、既存のファイルに割り当てられている容量の合計、特権を持たないユーザーが新たなファイルを生成する場合に使用できる容量の合計、そのファイルシステム上の全ファイルに現在割り当てられている通常使用可能な容量の割合 (パーセント単位) です。このオプションは、`-b`、`-e`、`-n`、`-t` の各オプション (指定されている場合) よりも優先されます。
- `-l` ローカルファイルシステムについての情報だけを報告します。このオプションは マウントされているファイルシステムに対してのみ有効であり、`-o` オプションと同時に指定することはできません。
- `-n` *FSType* の名前だけを報告します。オペランド指定を省略すると、このオプションはマウントされているファイルシステムタイプの一覧を出力します。このオプションは マウントされているファイルシステムに対してのみ有効であり、`-o` オプションと同時に指定することはできません。
- `-o FSType-specific_options` *FSType* 固有のオプションを指定します。オプションとオプションとの間は コマンドだけで区切り、空白は入れないでください。詳細については *FSType* コマンド用のマニュアルページを参照してください。
- `-t` 合計値を含む完全なリストを出力します。このオプションは、`-b`、`-e`、`-n` の各オプション (指定されている場合) よりも優先されます。
- `-V` 指定されたコマンド行の全内容のエコーだけを行い、コマンド自体は実行しません。コマンド行の内容としては、ユーザーが指定したオプションやオペランドに加え、`/etc/mnttab`、`/etc/vfstab`、`/etc/default/fs` の各ファイルから得られた情報が付加されます。このオプションは、コマンド行の記述内容の正当性を検査する目的で使用できます。

`/usr/xpg4/bin/df` 以下のオプションは、`/usr/xpg4/bin/df` でのみ指定できます。

- `-P` `-k` オプションと同じですが、出力の単位は 512 バイトとなります。

df(1M)

オペランド	<p>df は、<i>block_device</i>、<i>directory</i>、<i>file</i> の優先度に従ってオペランドを解釈します。以下のオペランドを指定できます。</p> <p><i>block_device</i> ブロック型特殊デバイス (たとえば /dev/dsk/c1d0s7) を指定します。対応するファイルシステムはマウントされている必要はありません。</p> <p><i>directory</i> 既存のディレクトリ名を指定します。</p> <p><i>file</i> 有効なファイル名を指定します。df は <i>file</i> があるファイルシステムについて報告します。</p> <p><i>resource</i> NFS リソース名を指定します。</p>				
使用法	<p>ファイルが 2 ギガバイト (2³¹ バイト) 以上ある場合の df の動作については、largefile(5) を参照してください。</p>				
使用例	<p>例 1 ファイルシステム /usr に関する情報を出力する</p> <p>次の例は、ファイルシステム /usr に関する情報を出力します。</p> <pre>example% /usr/xpg4/bin/df -P /usr</pre> <p>例 2 ファイルシステム /usr に関する情報を出力する (/usr/src が /usr ファイルシステムの一部である場合)</p> <p>/usr/src が /usr ファイルシステムの一部である場合、次の例も上記の例と同じ結果を出力します。</p> <pre>example% /usr/xpg4/bin/df -P /usr/src</pre> <p>例 3 すべての ufs ファイルシステムの i ノード使用率を表示する</p> <p>次の例は、すべての ufs ファイルシステムにおける i ノードの使用率を表示します。</p> <pre>example% /usr/bin/df -F ufs -o i</pre>				
環境	<p>SYSV3 この環境変数はデフォルトの df の動作を無効にして、INTERACTIVE UNIX システムと SCO UNIX のインストールスクリプトとの互換性を提供するために使用します。SYSV3 は互換性だけを目的とした環境変数なので、新しいスクリプトでは使用しないでください。</p> <p>設定した場合、通常、「ファイル」を表示するヘッダーはすべて、「ノード」を表示するようになります。df の実行に影響を与える環境変数 LC_CTYPE、LC_MESSAGES、NLSPATH についての詳細は、environ(5) を参照してください。</p>				
終了ステータス	<p>以下の終了ステータスが返されます。</p> <table><tr><td>0</td><td>正常終了</td></tr><tr><td>>0</td><td>エラーが発生した</td></tr></table>	0	正常終了	>0	エラーが発生した
0	正常終了				
>0	エラーが発生した				

ファイル /dev/dsk/* ディスクデバイス

/etc/default/fs ローカルファイルシステムタイプのデフォルト値。デフォルト値は /etc/default/fs の以下のフラグに対して設定することができる

例: LOCAL=ufs

LOCAL: *FSType* が指定されなければ、コマンドのデフォルトパーティションになる

/etc/mnttab マウントテーブル

/etc/vfstab 各ファイルシステム用のデフォルトパラメタ

属性 次の属性については `attributes(5)` のマニュアルページを参照してください。

/usr/bin/df

属性タイプ	属性値
使用条件	SUNWcsu

/usr/xpg4/bin/df

属性タイプ	属性値
使用条件	SUNWxcu4

関連項目 `find(1)`, `df_ufs(1M)`, `mount(1M)`, `statvfs(2)`, `default_fs(4)`, `mnttab(4)`, `vfstab(4)`, `attributes(5)`, `environ(5)`, `largefile(5)`, `XPG4(5)`

注意事項 ファイルシステムで UFS ログが有効になっている場合、ログで使用されるディスク容量は `df` の報告に反映されます。ログは、ファイルシステムの空きブロックから割り当てられ、ファイルシステム 1G バイト当たり 約 1M バイトから最大約 64M バイトのサイズになります。

fdisk(1M)

名前	fdisk – 固定ディスクパーティションテーブルの作成または変更
形式	fdisk [-o <i>offset</i>] [-s <i>size</i>] [-P <i>fill_patt</i>] [-S <i>geom_file</i>] [-w r d n I B t T g G R] [-F <i>fdisk_file</i>] [[-v] -w { <i>fdisk_file</i> -}] [-h] [-b <i>masterboot</i>] [-A <i>id : act : bhead : bsect : bcyl : ehead : esect : ecyl : rsect : numsect</i>] [-D <i>id : act : bhead: bsect : bcyl : ehead: esect : ecyl : rsect : numsect</i>] <i>rdevice</i>
機能説明	<p>このコマンドは、次のように使用します。</p> <ul style="list-style-type: none"> ■ x86 システム上の fdisk パーティションテーブルを作成および変更する ■ SPARC または x86 システム上のリムーバブルメディアで fdisk パーティションテーブルを作成および変更する ■ x86 システムのみ、固定ディスクの最初のセクターに格納されるマスターブートレコードをインストールする <p>このテーブルは、異なるオペレーティングシステムごとに予約されているディスク領域を識別するために、または第 2 段階のブートストラップのあるパーティション (アクティブな Solaris パーティション) を識別するために、第 1 段階のブートストラップ (またはファームウェア) により使用されます。 <i>rdevice</i> 引数は、固定ディスクに対応する raw デバイスを指定する場合に使用しなくてはなりません。たとえば、<code>/dev/rdsk/c0t0d0p0</code> のように指定します。</p> <p>このプログラムは、3 種類の動作モードで実行します。最初の動作モードは対話モードです。対話モードでは、ディスク上に存在するパーティションテーブルが表示され、さらにユーザーがテーブルを変更できるように、メニューが表示されます。メニュー、問い合わせ、警告、およびエラーメッセージは、特に説明がなくても理解できるものです。</p> <p>対話モードでは、ディスク上にパーティションテーブルがない場合、ユーザーはデフォルトのパーティション分割を行うか、または初期テーブル値を指定するかの、どちらかを選択できます。デフォルトのパーティション分割では、ディスク全体が Solaris システムに割り当てられ、Solaris システムパーティションがアクティブになります。いずれの場合も、初期テーブルが作成された時点で、fdisk はパーティションテーブルとともに、第 1 段階のブートストラップ (x86 のみ) コードも書き込みます。</p> <p>2 番目の動作モードは、エントリの追加、削除、または fdisk テーブル全体の置き換えを自動的に行う場合に使用します。このモードでは、コマンド行に指定したエントリを追加または削除できます。また、ファイルから fdisk テーブル全体を読み込んで、元のテーブルと置き換えることもできます。fdisk はこのファイルを作成する目的でも使用できます。コマンド行オプションを使用すると、任意の fdisk テーブルが、ディスク全体を Solaris システムに割り当てるデフォルトの fdisk テーブルに置き換えられます。</p> <p>3 番目の動作モードは、ディスクの診断時に使用します。このモードでは、ディスクのあるセクションをユーザーが指定したパターンで埋めることができます。また、ディスクのモードセクションを読み書きすることもできます。</p>

メニューオプション

fdisk プログラムの対話モードで使用できるメニューオプションは、次のとおりです。

Create a partition

このオプションを使用すると、新しいパーティションを作成できます。パーティションの最大数は4つです。パーティションのタイプ (SOLARIS、MS-DOS、UNIX、またはその他) が尋ねられます。次に、パーティションのサイズをディスクの割合で指定するように求められます。この時点で `c` と入力すると、開始シリンダ番号およびシリンダ内のパーティションのサイズが尋ねられます。`c` を入力しなかった場合は、パーティションが収まるように、fdisk プログラムが開始シリンダ番号を決定します。いずれの場合でも、パーティションが既存のパーティションとオーバーラップする場合、または収まりきらない場合は、メッセージが表示されて最初のメニューに戻ります。

Change Active (Boot from) partition

このオプションを使用すると、第1段階のブートストラップが第2段階のブートストラップを検索するパーティション (アクティブパーティションと呼ばれる) を指定できます。

Delete a partition

このオプションを使用すると、作成済みのパーティションを削除できます。そのパーティション内のすべてのデータが破壊されるので注意してください。

この時点で、次に示すオプションを使用してパーティションテーブルの設定を変更することも、あるいは、テーブルを変更せずにセッションを中止することもできます。

Exit

このオプションを使用すると、fdisk によってこのセッションで作成された新しいテーブルが固定ディスクに書き込まれ、fdisk は終了します。

Cancel

パーティションテーブルを変更せずに終了します。

オプション

fdisk で使用できるオプションは、次のとおりです。

-A *id:act:bhead:bsect:bcyl:ehead:esect:ecyl:rsect:numsect*

引数で指定されたパーティションを追加します (形式については以下の `-F` オプションを参照)。fdisk テーブルが変更された場合、このオプションを使用すると、Solaris パーティション上の VTOC がゼロで埋められます。

-b *master_boot*

マスターブートプログラムとしてファイル *master_boot* を指定します。デフォルトのマスターブートプログラムは `/usr/lib/fs/ufs/mboot` です。

-B

デフォルトでディスク全体を1つの Solaris パーティションに割り当てます。

-d

詳細デバッグモードを有効にします。fdisk を使用するたびに、その状態を標準エラー出力に送ります。このオプションの出力を `-F` オプションで使用しないでください。。

fdisk(1M)

-D *id:act:bhead:bsect:bcyl:ehhead:esect:ecyl:rsect:numsect*

引数で指定されたパーティションを削除します (形式については以下の -F オプションを参照)。引数が完全に一致しないと、エントリが削除されないことに注意してください。fdisk テーブルが変更された場合、このオプションを使用すると、Solaris パーティションの VTOC がゼロで埋められます。

-F *fdisk_file*

ファイル *fdisk_file* を使用してテーブルを初期化します。このオプションを使用して fdisk テーブルを変更すると、Solaris パーティションの VTOC がゼロで埋められます。

fdisk_file には指定行を 4 行まで含めることができます。各行は復帰改行文字 (\n) で区切ります。行の先頭文字がアスタリスク (*) の場合、その行はコメント行とみなされます。各行は位置に依存するエントリからなり、エントリは空白またはコロンで区切ります。形式は次のとおりです。

id act bhead bsect bcyl ehhead esect ecyl rsect numsect

エントリの値は次のとおりです。

<i>id</i>	パーティションのタイプ。指定可能な数値は、fdisk.h に記述されています。
<i>act</i>	アクティブパーティションプラグ。0 は非アクティブ、128 はアクティブを意味します。
<i>bhead</i>	パーティションが開始するヘッド。0 に設定されている場合、fdisk は他の情報に基づいて適切な値を設定します。
<i>bsect</i>	パーティションが開始するセクター。0 に設定されている場合、fdisk は他の情報に基づいて適切な値を設定します。
<i>bcyl</i>	パーティションが開始するシリンダ。0 に設定されている場合、fdisk は他の情報に基づいて適切な値を設定します。
<i>ehhead</i>	パーティションが終了するヘッド。0 に設定されている場合、fdisk は他の情報に基づいて適切な値を設定します。
<i>esect</i>	パーティションが終了するセクター。0 に設定されている場合、fdisk は他の情報に基づいて適切な値を設定します。
<i>ecyl</i>	パーティションが終了するシリンダ。0 に設定されている場合、fdisk は他の情報に基づいて適切な値を設定します。
<i>rsect</i>	パーティションが開始するディスクの先頭からの相対的な位置にあるセクター。この指定は必須です。この値は、fdisk が他のフィールドを設定するのに使用できます。
<i>numsect</i>	セクター数で表したこのディスクパーティションのサイズ。この指定は必須です。この値は、fdisk が他のフィールドを設定するのに使用できます。

- g
ディスクのラベルジオメトリを取得して、標準出力に出力します (形式については -s オプションを参照)。
- G
ディスクの物理ジオメトリを取得して、標準出力に出力します (形式については -s オプションを参照)。
- h
詳細メッセージを表示します。メッセージには全オプションのリストとともに、各オプションの説明が示されます。
- I
デバイスチェックを省略します。デバイスを使用せずに、ディスクに送られる内容のファイルイメージを生成する場合に使用します。このオプションは -s と組み合わせて使用する必要があります (上記を参照)。
- n
他のオプションで明示的に指定されている場合を除き、fdisk テーブルを更新しません。他のオプションを指定しない場合、-n オプションは、マスターブートレコードだけをディスクに書き込みます。また、-n オプションを指定した場合、fdisk は対話モードで起動しないことにも注意してください。
- o *offset*
ディスクの先頭からのブロックオフセット。このオプションは -p、-r、および -w とともに使用します。このオプションを指定しない場合、ゼロが想定されます。
- p *fill_patt*
パターン *fill_patt* でディスクを埋めます。 *fill_patt* は 10 進数または 16 進数にすることができ、一定のロングパターンを表す数として使用します。 *fill_patt* が # の場合、パターンは各ブロックで # になります。パターンはロングワードとして各ブロックに格納されて各ブロックを埋めます (-o および -s を参照)。
- r
ディスクから読み込んで標準出力に書き込みます。操作の開始ポイントおよびサイズを指定するオプション -o と -s を参照してください。
- R
読み取り専用ディスクとして扱います。このオプションはテスト用です。
- s *size*
操作を実行するブロック数 (-o を参照)
- S *geom_file*
geom_file の内容をラベルジオメトリとして設定します。 *geom_file* には、行ごとに 1 つの指定を記述します。各行は復帰改行文字 (\n) で区切ります。行の先頭の文字がアスタリスク (*) の場合、その行はコメントとみなされます。各行は位置に依存するエントリからなり、空白またはコロンで区切ります。形式は次のとおりです。

pcyl ncyl acyl bcyl nheads nsectors sectsiz

エントリの値は次のとおりです。

fdisk(1M)

pcyl ドライブの物理シリンダ数
ncyl ドライブの使用可能シリンダ数
acyl ドライブの代替シリンダ数
bcyl ドライブのオフセットシリンダ数 (ゼロにすべき)
nheads このドライブのヘッド数
nsectors トラックあたりのセクター数
sectsiz セクターサイズ (バイト単位)

-t
 パーティションテーブル境界を越えないように、無効なスライステーブルエントリを調整します。

-T
 パーティションテーブル境界にまたがる無効なスライステーブルエントリを削除します。

-v
 HBA (仮想) ジオメトリディメンションを出力します。このオプションは -w フラグと組み合わせて使用する必要があります。このオプションは、仮想ジオメトリをサポートするプラットフォーム (x86 のみ) で使用できます。

-w
 ディスクに書き込み、標準入力から読み込みます。操作の開始ポイントおよびサイズを指定する -o と -s オプションを参照してください。

-W-
 ディスクテーブルを標準出力に書き込みます。

-w *fdisk_file*
 ディスクテーブルから fdisk ファイル *fdisk_file* を作成します。このオプションは、-F オプションと組み合わせて使用できます。

ファイル /dev/rdsk/c0t0d0p0 固定ディスクに対応づけられる raw デバイス
 /usr/lib/fs/ufs/mboot デフォルトのマスターブートプログラム

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
アーキテクチャ	x86
使用条件	SUNWcsu

関連項目 uname(1), fmthard(1M), prtvtoc(1M) attributes(5)

診断 ほとんどのメッセージは、読めば理解できるものです。プログラムの開始直後に次のメッセージが表示される場合があります。

Fdisk: cannot open <device>

このメッセージは、デバイス名引数が無効であることを示します。

Fdisk: unable to get device parameters for device <device>

このメッセージは、固定ディスクの構成に問題があるか、または固定ディスクドライバでエラーが発生したことを示します。

Fdisk: error reading partition table

このメッセージは、固定ディスクから最初の読み込み時にエラーが発生したことを示します。固定ディスクのコントローラまたはドライバに問題があるか、固定ディスクの構成に問題がある可能性があります。

Fdisk: error writing boot record

このメッセージは、固定ディスクに新しいパーティションテーブルを書き込むときに、エラーが発生したことを示します。固定ディスクコントローラ、ディスク自体、ドライバ、または固定ディスクの構成に問題がある可能性があります。

ffbconfig(1M)

名前	ffbconfig, SUNWffb_config – FFB グラフィックスアクセラレータの設定
形式	<pre>/usr/sbin/ffbconfig [-dev <i>device-filename</i>] [-res <i>video-mode</i> [now try] [noconfirm nocheck]] [-file machine system] [-deflinear true false] [-defoverlay true false] [-linearorder first last] [-overlayorder first last] [-expvis enable disable] [-sov enable disable] [-maxwids <i>n</i>] [-extovl enable disable] [-g <i>gamma-correction-value</i>] [-gfile <i>gamma-correction-file</i>] [-propt] [-prconf] [-defaults] /usr/sbin/ffbconfig [-propt] [-prconf] /usr/sbin/ffbconfig [-help] [-res ?]</pre>
機能説明	<p>ffbconfigは、FFB グラフィックスアクセラレータおよび FFB 対応の X11 ウィンドウシステムのデフォルトの一部を設定します。</p> <p>ffbconfigの1番目の形式では、指定したオプションを OWconfigファイルに保存します。これらのオプションは、次にウィンドウシステムをそのデバイスで実行するときに FFB デバイスを初期化するために使用されます。OWconfig ファイル内のオプションの更新は、異なるウィンドウセッションや再起動したシステムでも有効となります。</p> <p>-prconf、-propt、-help、-res? オプションだけを起動する2番目と3番目の形式では、OWconfig ファイルは更新されません。また、3番目の形式では、その他のオプションはすべて無視されます。</p> <p>オプションは、一度に1つの FFB デバイスに対してのみ指定することができます。複数の FFB デバイスに対してオプションを指定するには、ffbconfig を複数回起動する必要があります。</p> <p>ffbconfig で指定できるのは、FFB 固有のオプションだけです。デフォルトの表示色数、デフォルトの画像表示形式クラスなどを指定する通常のウィンドウシステムのオプションは、openwin コマンド行のデバイス修飾子で指定してください。詳細については、OpenWindows デスクトップ・リファレンスマニュアルを参照してください。</p> <p>ユーザーは、更新する OWconfig ファイルを指定することもできます。デフォルトでは、/etc/openwin ディレクトリツリーにあるマシン固有のファイルが更新されます。別のファイルを指定するには、-file オプションを使用します。たとえば、/usr/openwin ディレクトリツリーにあるシステム共通の OWconfig ファイルを代わりに更新することができます。</p> <p>これらの標準 OWconfig ファイルのどちらもスーパーユーザーのみが書き込みを行います。したがって、スーパーユーザーが所有する ffbconfig プログラムは、setuid による root の権限で実行されます。</p>
オプション	<pre>-dev <i>device-filename</i> FFB 特殊ファイルを指定します。デフォルトは/dev/fbs/ffb0 です。 -file machine system 更新する OWconfig ファイルを指定します。machine を指定すると、 /etc/openwin ディレクトリツリーにあるマシン固有の OWconfig ファイルが使</pre>

ffbconfig(1M)

用されます。system を指定すると、/usr/openwin ディレクトリツリーにある共通のOWconfig ファイルが使用されます。ファイルがない場合は、新たに生成されます。

`-res video-mode [now | try [noconfirm | nocheck]]`

指定した FFB デバイスに接続されているモニターを制御する際に使われる表示モードを指定します。

表示モードの形式は `widthxheightxrate` で、`width` はピクセル単位の画面幅、`height` はピクセル単位の画面の高さ、`rate` は画面を垂直方向に再描画する周期です。

960x680x112s や 960x680x108s の s 接尾辞は、これらが立体表示モードであることを意味します。640x480x60i や 768x575x50i の i 接尾辞は、インタレース表示タイミングを有効にします。この接尾辞がない場合は、ノンインタレースタイミングが使用されます。

`-res` (「形式」に記されている 3 番目の形式) にリフレッシュレートを指定する際は、値の直前に `x` の代わりに `@` を使用することができます。たとえば、1280x1024@76 のように指定することができます。

一部の表示モードは、FFB の一部のバージョンのみが対応しています。また、FFB が対応している表示モードには、モニターが対応していないものもあります。FFB デバイスとモニターの両方が対応している表示モードのリストは、`-res ? オプション付きの ffbconfig` を実行することによって得ることができます。

FFB が対応している表示モードのリストを以下に示します。

表示モード	説明
1024x768x60	
1024x768x70	
1024x768x75	
1024x768x77	
1024x800x84	
1152x900x66	
1152x900x76	
1280x800x76	
1280x1024x60	
1280x1024x67	
1280x1024x76	
960x680x112s	(立体表示)

fbconfig(1M)

表示モード	説明
960x680x108s	(立体表示)
640x480x60	
640x480x60i	(インタレース)
768x575x50i	(インタレース)
1440x900x76	(高解像度)
1600x1000x66	(高解像度)
1600x1000x76i	(高解像度)
1600x1280x76	(高解像度)
1920x1080x72	(高解像度)
1920x1200x70	(高解像度)

記号名

便宜上、表示モードのいくつかには記号名が定義されています。 *widthxheightxrate* の形式の代わりに、記号名を *-res* の引数として指定することができます。記号名 *none* は、ウィンドウシステムを実行すると、画面の解像度は現在デバイスにプログラムされている表示モードになることを意味します。

記号名	対応する表示モード
svga	1024x768x60
1152	1152x900x76
1280	1280x1024x76
stereo	960x680x112s
ntsc	640x480x60i
pal	768x575x50i
none	(デバイスでプログラムされている表示モード)

-res オプションには、表示モードの直後に次の追加引数を指定することができます。追加引数は、単独でも複数でも指定することができます。

now

OWconfig ファイルの表示モードを更新するとともに、FFB デバイスが指定した表示モードにただちにプログラムされます (この機能は、ウィンドウシステムを開始する前に表示モードを変更する際に便利です)。

ffbconfig(1M)

対象となるデバイスが稼働している間 (たとえば、ウィンドウシステムの稼働中) に、この追加オプションを `ffbconfig` に指定することはお勧めしません。予期しない結果になることもあります。 `now` オプションを指定して `ffbconfig` コマンドを実行する場合は、最初にウィンドウシステムを終了してください。 `now` オプションがウィンドウシステムのセッション中に使用された場合、表示モードはただちに変更されますが、画面の幅や高さはそのセッションが終了して次のセッションに入るまで変更されません。さらに、立体表示モードではシステムが変更を認識しないことがあります。したがって、ウィンドウシステムの稼働中には絶対に `now` オプションを指定しないでください。

`noconfirm`

確認と警告メッセージを省略し、要求された表示モードにプログラムします。

`-res` オプションを指定した際に、システムが使用できない状態になり、表示出力がなくなる場合があります。このような状況は、特定のコードが読み込まれた際のモニターセンスコードにあいまいさがあった場合などに発生します。このような事態を避けるために `ffbconfig` のデフォルトの動作では、この問題についての警告メッセージと処理を継続するかどうかを確認するメッセージを表示します。このオプションは、`ffbconfig` がシェルスクリプトから実行されている場合に便利です。

`nocheck`

モニターセンスコードに基づく通常のエラーチェックが行われません。ユーザーによって指定された表示モードは、現在接続されているモニターに適切かどうかにかかわらず受け付けられます。このオプションは、FFB デバイスに異なるモニターを接続する場合に便利です。このオプションを指定すると、`noconfirm` も指定されます。

`try`

指定した表示モードに試験的にプログラムされます。ユーザーは、指定した表示モードを使用する場合は、メッセージが表示されてから 10 秒以内に `y` と入力します。表示されたモードを使用しない場合は、10 秒以内に任意の文字を入力します。 `y` または `Return` キー以外の文字の入力は、すべて「使用しない」とみなされ、以前の表示モードに戻され、`OWconfig` ファイル中の表示モードは書き換えられません。その他の指定されたオプションは有効となります。 `Return` キーの入力があつた場合は、新しい表示モードを保持するかどうかを `yes` または `no` で確認するメッセージが表示されます。このオプションを指定すると、`now` サブオプションも指定されます。(`now` サブオプションの注意を参照)。

`-deflinear true | false`

FFB には、2 種類の画像表示形式があります。リニア画像と非リニア画像です。リニア画像はガンマ補正され、非リニア画像は補正されません。リニア画像版も非リニア画像版も、ともに持つ画像表示形式が 2 つあります。24 ビット `TrueColor` と 8 ビット `StaticGray` です。

`-deflinear true` を指定すると、デフォルトの画像表示形式として、デフォルトで選択されたオプション (特に、Xsun(1)の `defdepth` および `defclass` オプション。詳細は *OpenWindows* デスクトップ・リファレンスマニュアルのマニュアルページを参照) を満たすリニア画像を設定します。具体的には、デフォルトの画像表示形式の選択オプションは、Xsun(1) の `defdepth` および `defclass` オプションによって設定されたものです。

`-deflinear false` を指定すると、または他のデフォルトで選択されたオプションを満たすリニア画像がない場合は、これらの他のオプションを満たす非リニア画像がデフォルトとして選択されます。

FFB にはリニアオーバーレイ画像表示形式がないため、オプションが存在する場合は、このオプションを使用することはできません。

`-defoverlay true | false`

FFB が、残りの FFB 画像から切り離されたピクセルを持つ 8 ビット疑似カラー画像を提供します。これを、オーバーレイ画像といいます。この画像表示形式で作成されたウィンドウは、他の画像表示形式で作成されたウィンドウに影響を与えません。逆に、他の画像表示形式で作成されたウィンドウは、オーバーレイウィンドウに影響を与えます。この画像表示形式では、256 種類の不透明カラーの値による拡張オーバーレイを使用することができます。 `-maxwids` を参照してください。

`-defoverlay` に `true` を指定すると、オーバーレイ画像がデフォルト画像になります。 `-defoverlay` に `false` を指定すると、他のデフォルトで選択された `defdepth` および `defclass` オプションを満たすオーバーレイでない画像表示形式が、デフォルトの画像表示形式として選択されます。詳細は、*OpenWindows* デスクトップ・リファレンスマニュアルのマニュアルページを参照してください。

`-defoverlay true` を使用する場合は、`openwin` コマンド行で選択されたデフォルトの深さとクラスは常に 8 ビット疑似カラーである必要があります。それ以外の場合は、警告メッセージが出力され、`-defoverlay` オプションは `false` として扱われます。 `-deflinear` オプションが存在する場合、FFB にはリニアオーバーレイ画像表示形式がないため、このオプションは使用することができません。

`-linearorder first | last`

`first` を指定すると、FFB 画面用の X11 画面画像表示形式リスト上で、リニア画像が非リニア画像より前に表示されます。 `last` を指定すると、非リニア画像は、リニア画像より前に表示されます。

`-overlayorder first | last`

`first` を指定すると、FFB 画面用の X11 画面画像表示形式リスト上で、8 ビット疑似カラーオーバーレイ画像が、非オーバーレイ画像より前に表示されます。 `last` を指定すると、非オーバーレイ画像は、オーバーレイ画像より前に表示されます。

`-expvis enable | disable`

`enable` を指定すると、OpenGL Visual Expansion が起動されます。選択された画像表示形式グループ (8 ビット `PseudoColor`、24 ビット `TrueColor` など) は、画面画像表示形式リストで見つけることができます。

-sov enable | disable

enable を指定すると、ルートウィンドウの `SERVER_OVERLAY_VISUALS` 属性が有効になります。SOV 画像が転送され、それらの透過タイプ、値、階層は、この属性によって参照することができます。disable を指定すると、`SERVER_OVERLAY_VISUALS` 属性は定義されません。SOV 画像は転送されません。

-maxwids *n*

ウィンドウ ID (WID) として使用するために予約される最大数の FFB チャンネルピクセル値を指定します。オーバーレイカラーマップのピクセル値の残りは、通常の X11 の未使用のカラーピクセルのために使用されます。確保された WID は、(XGL などの) 3次元グラフィックスウィンドウ、MBX ウィンドウと、デフォルト以外の画像表示形式をもつウィンドウにより発生順に割り当てられます。X チャンネルコードの 0 から (255-*n*) は、未使用のカラーピクセルです。(255-*n*+1) から 255 の X チャンネルコードは、WID として使用するために予約されます。FFB と FFB2 の有効な値は、1、2、4、8、16、32 です。FFB2+ の有効な値は、1、2、4、8、16、32、64 です。

-extovl enable | disable

このオプションは FFB2+ のみで使用することができます。enable を指定すると、拡張オーバーレイを使用することができます。このオーバーレイ画像には 256 種類の不透明カラーがあります。SOV 画像には 255 種類の不透明カラーと 1 種類の透明カラーがあります。このオプションは、ハードウェアによる透明カラーを有効にするため、SOV 画像を使用するウィンドウで、より高い性能が得られます。

-g *gamma-correction value*

このオプションは FFB2+ のみで使用することができます。このオプションによって、ガンマ補正の値を変更することができます。すべてのリニア画像ではガンマ補正を使用することができます。デフォルトでは、ガンマ補正の値は 2.22 です。0 より小さい値は無効 (不正) です。ガンマ補正の値はリニア画像に適用され、リニア画像の有効ガンマ値は 1.0 になります。これは、`XSolarisGetVisualGamma(3)` によって返される値です。この機能については、`XSolarisGetVisualGamma(3)` を参照してください。

このオプションは、ウィンドウシステムが稼動しているときに使用することができます。ガンマ補正の値を変更すると、リニア画像を使用して表示されているすべてのウィンドウが影響を受けます。

-gfile *gamma-correction file*

このオプションは FFB2+ のみで使用することができます。このオプションは、指定されたファイルからガンマ補正表を読み込みます。このファイルは、各行が R、G、B チャンネルの値を持つように書式化されている必要があります。それらの値は、16 進数で指定し、値と値の間は 1 つ以上の空白文字で区切ります。このファイルでは、3 つの値の組が 256 種類定義されます。このファイルの例を以下に示します。

```
0x00 0x00 0x00
0x01 0x01 0x01
0x02 0x02 0x02
...
...
```

ffbconfig(1M)

```
0xff 0xff 0xff
```

このオプションを使用することによって、ウィンドウシステムが稼動しているときにガンマ補正表を読み込むことができます。新しいガンマ補正は、このリニア画像によって表示されているすべてのウィンドウに影響を与えます。ユーザーが指定した表によってガンマ補正を行う際は、ガンマ補正の値は定義されません。デフォルトでは、ウィンドウシステムはガンマ補正值として2.22を使用し、このガンマ補正值に対応してウィンドウシステムが作成したガンマ補正表を読み込みます。

-defaults

すべてのオプションの値をそれぞれのデフォルト値に戻します。

-propt

-file オプションで指定された OWconfig ファイルに書かれた FFB オプションの値のうち、-dev オプションで指定されたデバイスに対するものすべてを表示します。ffbconfig の呼び出しが終了した後に、OWconfig ファイルに書き込まれるオプションの値を表示します。次に表示例を示します。

```
--- OpenWindows Configuration for /dev/fbs/ffb0 ---
OWconfig: machine
Video Mode: NONE
Default Visual: Non-Linear Normal Visual
Visual Ordering: Linear Visuals are last
                  Overlay Visuals are last
OpenGL Visuals: disabled
SOV: disabled
Allocated WIDs: 32
```

-prconf

FFB のハードウェア構成を表示します。次に表示例を示します。

```
--- Hardware Configuration for /dev/fbs/ffb0 ---
Type: double-buffered FFB2 with Z-buffer
Board: rev x
PROM Information: @(#)ffb2.fth x.x xx/xx/xx
FBC: version x
DAC: Brooktree 9068, version x
3DRAM: Mitsubishi 1309, version x
EDID Data: Available - EDID version 1 revision x
Monitor Sense ID: 4 (Sun 37x29cm RGB color monitor)
Monitor possible resolutions: 1024x768x60, 1024x768x70,
                              1024x768x75, 1152x900x66, 1152x900x76,
                              1280x1024x67, 1280x1024x76, 960x680x112s,
                              640x480x60
Current resolution setting: 1280x1024x76
```

-help

ffbconfig コマンド行のオプションと機能の概要を一覧で表示します。

デフォルト設定

ffbconfig コマンド行で指定されていないオプションについては、対応する OWconfig ファイル中のオプションは更新されず、ファイル内の値がそのまま使用されます。

ffbconfig(1M)

ウィンドウシステムを実行する際に、ffbconfig による FFB オプションの指定がまったくなかった場合は、デフォルト値が使用されます。オプションのデフォルト値を以下に示します。

オプション	デフォルト値
-dev	/dev/fbs/ffb0
-file	machine
-res	none
-deflinear	false
-defoverlay	false
-linearorder	last
-overlayorder	last
-expvis	enabled
-sov	enabled
-maxwids	32

-res オプションのデフォルト値 none とは、ウィンドウシステムが実行された場合に、画面解像度がそのデバイスに現在プログラムされている表示モードになることを意味しています。

注：これによって、PROM によってデバイスの解像度を指定しているユーザーとの共用性が保てます。(gx などの)一部のデバイスでは、PROM が表示モードを指定する唯一の手段です。これは、デフォルトの FFB 表示モードは、最終的に PROM によって決まることを意味しています。

使用例 例 1 モニターの種類の変更

モニターの種類を、垂直周波数 76 Hz で解像度 1280 × 1024 に変更する例を以下に示します。

```
example% /usr/sbin/ffbconfig -res 1280x1024x76
```

ファイル /dev/fbs/ffb0 デバイス特殊ファイル

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	SUNWffbcf

関連項目 mmap(2), attributes(5), fbio(7I), ffb(7D)

fbconfig(1M)

OpenWindows デスクトップ・リファレンスマニュアルのマニュアルページ

名前	format - ディスクのパーティション分割および保守のためのユーティリティ
形式	format [-f <i>command-file</i>] [-l <i>log-file</i>] [-x <i>data-file</i>] [-d <i>disk-name</i>] [-t <i>disk-type</i>] [-p <i>partition-name</i>] [-s] [-m] [-M] [-e] [<i>disk-list</i>]
機能説明	<p>format を使用すると、システムディスクのフォーマット、ラベル付け、修復、および分析が行えます。従来のディスク保守用プログラムとは異なり、format は SunOS 環境で実行します。システムの稼働中はシステムディスクへの操作が制限されるので、format はメモリー常駐型のシステム環境でも使用することができます。しかし、大部分のアプリケーションでは、SunOS 環境で format を実行する方が簡単です。</p> <p>-x オプションが指定されると、format は最初に <i>data-file</i> で定義されているディスクリストを使用します。次に format は、FORMAT_PATH 環境変数、コロンで区切られたファイル名またはディレクトリ (あるいはその両方) のリストを調べます。ディレクトリの場合、format はそのディレクトリで <i>format.dat</i> というファイルを検索します。ファイル名は絶対パス名でなければならず、そのまま使用されます。format は指定された各ファイルのディスクおよびパーティション定義をすべて作業用セットに追加します。複数の同じ定義は無視され、そのことが通知されることはありません。FORMAT_PATH が設定されていない場合のデフォルトのパスは /etc/format.dat です。</p> <p><i>disk-list</i> は、c?t?d? または /dev/rdisk/c?t?d?s? の形式のディスクリストです。後者の形式では、シェルのワイルドカードを指定できます。たとえば、/dev/rdisk/c2* を指定すると、format はコントローラ c2 だけに接続されている全ドライブを操作対象とします。<i>disk-list</i> が指定されていない場合、format は、操作対象となり得る、システム上に存在するすべてのディスクを一覧表示します。</p> <p>着脱式媒体装置がリストに含まれるのは、ユーザーがエキスパートモード (-e オプション) で format を実行した場合だけです。この機能は、下位互換性を確保するために用意されています。書き替え可能な着脱式媒体装置には rmformat(1) を使用します。</p>
オプション	<p>次のオプションを指定できます。</p> <p>-d <i>disk-name</i> format プログラム開始時にカレントにするディスクを指定します。ディスクは論理名 (-d c0t1d0 など) で指定します。ディスクリストにディスクを 1 つだけ指定しても、同じ結果が得られます。</p> <p>-e SCSI エキスパートメニューを有効にします。ただし、このオプションは、不用意に使用しないでください。</p> <p>-f <i>command-file</i> 標準入力ではなく、<i>command-file</i> からコマンド入力を受け付けます。ファイルにはキーボードから入力した場合とまったく同じコマンドが指定されていない限りありません。このオプションが指定されている場合、format は continue? プロンプトを出力しません。<i>command-file</i> 内に y(es) または n(o) の応答を指定する必要はありません。非対話モードの場合、format はディスク選択番号の入力を求めることはありません。</p>

format(1M)

	<p>format を起動するときに、<code>-d disk-name</code> オプションを使用して現在の作業用ディスクを指定するか、または <code>command-file</code> に <code>disk</code> およびディスク選択番号を指定する必要があります。</p>
<code>-l log-file</code>	<p>指定された <code>log-file</code> に format セッションのトランスクリプトを記録します。これには標準入力、標準出力、および標準エラー出力が含まれます。</p>
<code>-m</code>	<p>拡張メッセージを有効にします。エラー発生時に詳細情報が得られます。</p>
<code>-M</code>	<p>拡張メッセージおよび診断メッセージを有効にします。フォーマット中に、SCSI デバイスのモードページの状態についてさまざまな情報が得られます。</p>
<code>-p partition-name</code>	<p>プログラムの開始時にカレントにするディスクのパーティションテーブルを指定します。このテーブルは、データファイルで定義されている名前を使用して指定します。このオプションを使用できるのは、ディスクがカレントになっており、さらにディスクタイプが指定されているか、ディスクタイプがディスクラベルから取得できる場合に限られます。</p>
<code>-s</code>	<p>サイレント。あらゆる標準出力を抑制します。エラーメッセージは引き続き表示されます。このオプションは通常、<code>-f</code> オプションと組み合わせて使用します。</p>
<code>-t disk-type</code>	<p>プログラムの開始時にカレントにするディスクのタイプを指定します。ディスクタイプは、データファイルで定義されている名前を使用して指定します。このオプションを使用できるのは、上記と同様、ディスクがカレントになっている場合に限られます。</p>
<code>-x data-file</code>	<p><code>data-file</code> に指定されているディスクのリストを使用します。</p>
使用法	<p>オプションを指定しないで format を起動した場合、あるいは、<code>-e</code>、<code>-l</code>、<code>-m</code>、<code>-M</code>、または <code>-s</code> オプションを指定して format を起動した場合、使用できるディスクを示した番号付きリストが表示され、リスト内の番号でディスクを指定するように求められます。マシンに 10 台以上のディスクが搭載されている場合は、スペースキーを押して、次の 1 画面分のディスクを表示します。</p> <p>ディスクが現在の画面に表示されていない場合でも、リスト内の番号でディスクを指定できます。たとえば、画面にディスク 11 ~ 12 が表示されている場合に 25 を入力すると、リスト内の 25 番目のディスクを指定できます。現在の画面に表示されていないディスクの番号を入力した場合は、選択の確認を求めるプロンプトが表示されます。表示リストに含まれている番号を入力した場合は、そのまま選択が受け付けられ、プロンプトは出力されません。</p>

ディスクを指定すると、メインメニューが表示されます。このメニューでは次の作業を実行できます。

analyze	読み取り、書き込み、および比較テストを実行します。
backup	バックアップラベルを検索します。
cache	書き込みキャッシュおよび読み取りキャッシュを有効または無効にします。またはその状態を照会します。このメニュー項目が表示されるのは、 <code>-e</code> オプションを指定して <code>format</code> を起動した場合だけです。このオプションを使用できるのは、SCSI デバイスに限られます。
current	デバイス名、ディスクジオメトリ、およびディスク装置のパス名を表示します。
defect	欠陥領域リストを検索して出力します。このオプションを使用できるのは、SCSI デバイスに限られます。IDE ディスクの場合は、自動欠陥領域管理機能が実行されます。 <code>defect</code> オプションを IDE ディスクで使用すると、次のメッセージが出力されます。 <code>Controller does not support defect management or disk supports automatic defect management</code>
disk	以降の操作で使用するディスク (カレントディスク) を選択します。
fdisk	<code>fdisk(1M)</code> プログラムを実行し、Solaris ソフトウェア (x86 ベースのシステムのみ) 用の <code>fdisk</code> パーティションを作成します。
format	カレントディスクをフォーマットして検証します。このオプションを使用できる、SCSI デバイスに限られます。IDE ディスクは製造元により、あらかじめフォーマットされています。IDE ディスクに対して <code>format</code> オプションを使用すると、次のメッセージが出力されます。 <code>Cannot format this drive. Please use your manufacturer-supplied formatting utility.</code>
inquiry	カレントドライブの製造元、製品名、およびリビジョンレベルを表示します。
label	カレントディスクに新しいラベルを書き込みます。
partition	スライスを作成または変更します。
quit	フォーマットメニューを終了します。
repair	ディスク上の特定のブロックを修復します。
save	新しいディスクおよびスライス情報を保存します。
type	ディスクタイプを選択 (定義) します。

format(1M)

verify	ラベルを読み取って表示します。シリンダ数、代替シリンダ数、ヘッド数、セクター数、パーティションテーブルなどの情報を出力します。				
volname	8文字の新しいボリューム名を用いて、ディスクにラベルを付けます。				
環境	FORMAT_PATH ディスクとパーティション定義用のコロンで区切られたファイル名またはディレクトリ (あるいはその両方)。ディレクトリを指定した場合、formatはそのディレクトリでファイル format.dat を検索します。				
ファイル	/etc/format.dat デフォルトのデータファイル				
属性	次の属性については、attributes(5) を参照してください。				
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">属性タイプ</th> <th style="text-align: center;">属性値</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">使用条件</td> <td style="text-align: center;">SUNWcsu</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWcsu
属性タイプ	属性値				
使用条件	SUNWcsu				
関連項目	fmthard(1M), prtvtoc(1M), rmformat(1), format.dat(4), attributes(5), sd(7D) 『Solaris のシステム管理 (基本編)』のディスク管理に関する章を参照してください。				
x86のみ	fdisk(1M)				
警告	format 機能を選択して Maxtor 207MB ディスクをフォーマットすると、次のメッセージが出力されます。 Mode sense page(4) reports rpm value as 0, adjusting it to 3600 これはドライバのバグであり、他社の旧式のドライブでも発生する可能性があります。上記メッセージはエラーではありません。ドライブは正常に動作します。 シリンダ 0 にはパーティションテーブル (ディスクラベル) がありますが、raw ディスクパーティションで使用すると、他社のソフトウェアによって上書きされることがあります。 format は容量が 1 TB を超えるディスクまたは LUN をサポートするために、EFI に準拠したディスクラベルの記載をサポートします。しかし、ファイルシステムやボリュームマネージャなど、多くのソフトウェアコンポーネントは容量がまだ 1 TB もしくはそれ以下に制限されているので、注意が必要です。詳細については、『Solaris のシステム管理 (基本編)』を参照してください。				
注意事項	format にはヘルプ機能が用意されており、入力を求められたときにいつでも使用できます。要求されている情報についてヘルプが必要な場合は、単に疑問符 (?) を入力します。求められている情報についての簡単な説明が出力されます。メニュープロンプトに ? を入力すると、使用できるコマンドの一覧が表示されます。				

format(1M)

SCSI ディスクの場合、Primary および Grown 両方の欠陥領域リストでフォーマットがデフォルトで実行されます。フォーマットを実行する前に、欠陥領域メニューで Primary リストだけを抽出しておくこと、Primary リストに対するフォーマットだけが実行されます。

キャッシュの状態を変更できるのは、SCSI デバイスだけです。また、すべての SCSI デバイスがキャッシュ状態の変更または保存をサポートしているわけではありません。

fsck(1M)

名前	fsck – ファイルシステムの検査および修復
形式	<pre>fsck [-F <i>FSType</i>] [-m] [-V] [<i>special...</i>]</pre> <pre>fsck [-F <i>FSType</i>] [-n N y Y] [-V] [-o <i>FSType-specific-options</i>]</pre> <pre> [<i>special...</i>]</pre>
機能説明	<p>fsck はファイルシステムの状態を検査し、不整合を対話形式で修復します。ファイルシステムに不整合がある場合、デフォルトでは、修復処理を行う前に、ユーザーからの yes または no の応答を待機します。ユーザーに書き込み権が与えられていない場合、fsck はデフォルトで no の応答に対する動作をします。修復処理によっては、データが失われることがあります。データが失われる量と重要度は、診断出力から判断できます。</p> <p><i>FSType-specific-options</i> は、(空白を入れずに) コンマで区切ったオプションのリストまたはキーワード / 属性のペアのリストとして指定します。これらは、<i>FSType</i>- 固有のコマンドモジュールによって解釈されます。</p> <p><i>special</i> は、ファイルシステムが配置されている文字型特殊デバイスを指定します。/dev/rdsk/c1t0d0s7 などがその例です。ブロック型特殊デバイスではなく、文字型特殊デバイスを使用することに注意してください。fsck は、ブロック型特殊デバイスがマウントされている場合には、動作しません。</p> <p><i>special</i> デバイスを指定しなかった場合、fsck は /etc/vfstab に指定されているファイルシステムを検査します。/etc/vfstab の中で、fsckdev フィールドに文字型特殊デバイスエントリがあり、fsckpass フィールドにゼロ以外の数値エントリのあるエントリが検査されます。-F <i>FSType</i> を指定すると、指示されたタイプのファイルシステムタイプだけが検査されます。</p> <p><i>special</i> が指定されていても、-F が指定されていない場合は、/etc/vfstab の対応するエントリを探すことによって、ファイルシステムタイプが判別されます。対応するエントリがない場合は、/etc/default/fs に指定されているデフォルトのローカルファイルシステムタイプが使用されます。</p> <p>ファイルシステムタイプが並列検査に対応している場合 (たとえば ufs)、検査可能ないくつかのファイルシステムを並列に検査できます。詳細は、ファイルシステム固有のマニュアルページ (fsck_ufs(1M) など) を参照してください。</p>
オプション	<p>次の汎用オプションを指定できます。</p> <p>-F <i>FSType</i> 操作するファイルシステムのタイプを指定します。</p> <p>-m 検査だけで修復は行いません。このオプションを指定すると、ファイルシステムがマウントできる状態かどうかを検査され、該当する終了ステータスが返されます。ファイルシステムがマウントできる状態であれば、fsck は次のメッセージを表示します。</p> <pre>ufs fsck: sanity check: /dev/rdsk/c0t3d0s1 okay</pre>

fsck(1M)

-n -N	fsck からのすべての問い合わせに対して no の応答をするものとみなします。ファイルシステムを書き込みでオープンしません。
-v	コマンド行を展開して表示しますが、コマンドは実行しません。このオプションは、コマンド行の検査および検証するのに使用します。
-y Y	fsck からのすべての問い合わせに対して yes の応答をするものとみなします。
-o <i>specific-options</i>	<p><i>specific-options</i> には、次のサブオプションの組み合わせを (空白は入れずに) コンマで区切って指定します。</p> <p>b=<i>n</i> ファイルシステムのスーパーブロックとしてブロック <i>n</i> を使用します。ブロック 32 はつねに、代替スーパーブロックの 1 つです。-N<i>v</i> オプションを指定して newfs(1M) を実行すると、他のスーパーブロックの位置を調べることができます。</p> <p>c 古い (静的テーブル) 形式のファイルシステムを新しい (動的テーブル) 形式に変換します。ファイルシステムが新しい形式のときは、ファイルシステム構成が古い形式をサポートできる場合、古い形式に変換します。対話モードの場合、変換の向きが表示され、変換を実行するかどうか尋ねられます。否定応答を返すと、そのファイルシステムに対してそれ以上の操作は実行されません。非対話モードの場合は、変換の向きが表示され、可能であれば、ユーザーとの対話なしで実行されます。すべてのファイルシステムを一括変換する場合は、非対話モードでの変換が便利です。ファイルシステムのタイプは fstyp(1M) の出力の先頭行からわかります。c オプションを使用することはほとんどありません。4.1 より前のリリースとの互換性を確保する場合には、限って指定してください。今後のリリースにこのオプションが含まれるかどうかは保証されません。</p> <p>f スーパーブロックのクリーンフラグの状態に関係なく、ファイルシステムを強制的に検査します。</p> <p>p 非対話 (preen) モードでファイルシステムを検査して修復します。ユーザーとの対話を必要とする問題が検出された場合には、ただちに終了します。並列ファイルシステム検査を有効にする場合、このオプションは必須です。</p>

fsck(1M)

		w	書き込み可能なファイルシステムだけを検査します。				
終了ステータス	0	ファイルシステムに問題はなく検査不要					
	1	誤ったパラメタが指定された					
	32	ファイルシステムがマウント解除されており、検査が必要 (fsck -m オプションの場合のみ)					
	33	ファイルシステムはマウント済み					
	34	デバイスの状態が取得できない					
	36	修正不可能なエラーが検出された - 通常どおりに終了					
	37	処理中にシグナルが捕捉された					
	39	修正不可能なエラーが検出された - 即時終了					
	40	ルートの場合 0 と同じ					
使用法	2G バイト (2 ³¹ バイト) 以上のファイルに対する fsck の動作については、largefile(5) のマニュアルページを参照してください。						
ファイル	/etc/default/fs	デフォルトのローカルファイルシステムタイプ。デフォルト値は /etc/default/fs 内の次のフラグに設定できる。例: LOCAL=ufs.					
		LOCAL FSType が指定されていない場合はコマンドのデフォルトファイルタイプ					
	/etc/vfstab	各ファイルシステム用のデフォルトのパラメタリスト					
属性	次の属性については、attributes(5) のマニュアルページを参照してください。						
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">属性タイプ</th> <th style="text-align: center;">属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWcsu</td> </tr> </tbody> </table>			属性タイプ	属性値	使用条件	SUNWcsu
属性タイプ	属性値						
使用条件	SUNWcsu						
関連項目	clri(1M), fsck_cacheufs(1M), fsck_ufs(1M), fsdb_ufs(1M), fsirand(1M), fstyp(1M), mkfs(1M), mkfs_ufs(1M), mountall(1M), newfs(1M), reboot(1M), vfstab(4), attributes(5), largefile(5), fs_ufs(7FS)						
警告	オペレーティングシステムはファイルシステムデータをバッファリングします。マウントされているファイルシステムに fsck を実行すると、オペレーティングシステムのバッファ内容がディスク上のものよりも古くなります。そのため、fsck を使用する場合は、ファイルシステムをマウント解除してください。それができない場合は、fsck の実行後に、システムを停止させ、ただちに再起動してください。それでも不十分な場合がしばしばあります。ファイルシステム上で fsck を実行した結果、ファイルシステムが変更された場合には、パニックが発生することがあります。						

注意事項 このコマンドは、すべての *FSTypes* で使用できるわけではありません。

raw (文字型特殊) デバイス (`fsck /dev/rdisk/c?t?d?s?`) ではなく、ブロック型デバイス (`fsck /dev/dsk/c?t?d?s?`) を使用するように選択した場合、2G バイトを超えるファイルシステム上で `fsck` を実行すると、`fsck` は失敗します。

Solaris 9 以降、`fsck` はディスク上の拡張属性データを管理します (拡張ファイル属性については、`fsattr(5)` のマニュアルページを参照)。拡張属性が設定されたファイルシステムを、属性を認識しないバージョンの Solaris (Solaris 9 より前のバージョン) にマウントすることはできますが、属性にアクセスすることはできません。`fsck` はファイルからそれらの属性を取り除き、`lost+found` に移します。属性が取り除かれたファイルシステムは、属性を認識するバージョンの Solaris 上でも完全に安定動作しますが、破壊しているファイルシステムとみなされてしまいます。この場合、属性を認識するバージョンの `fsck` を実行し、ファイルシステムを安定させてから、属性を認識する環境でそのファイルシステムを使用してください。

fuser(1M)

名前	fuser – ファイルまたはファイル構造を使用しているプロセスの特定
形式	<code>/usr/sbin/fuser [-c -f] [-nu] [-k -s sig] files [[-] [-c -f] [-nu] [-k -s sig] files] ...</code>
機能説明	<p>fuser は、引数として指定された <i>files</i> を使用しているプロセスのプロセス ID を表示します。</p> <p>各プロセス ID には文字修飾詞が続きます。文字修飾詞は、プロセスがファイルをどのように使用しているかを示すもので、以下があります。</p> <p>c カレントディレクトリとしてファイルを使用している</p> <p>m mmap(2) によってマップされたファイルを使用している。詳細は mmap(2) のマニュアルページを参照</p> <p>n そのファイルに対して、非ブロッキング強制ロックを保持している</p> <p>o オープンファイルとしてファイルを使用している</p> <p>r ルートディレクトリとしてファイルを使用している</p> <p>t テキストファイルとしてファイルを使用している</p> <p>y 制御端末としてファイルを使用している</p> <p>ファイルシステムがマウントされているブロック型特殊デバイスについては、そのデバイス上のファイルを使用しているすべてのプロセスが表示されます。その他のタイプのファイル(テキストファイル、実行可能ファイル、ディレクトリ、デバイスなど)については、そのファイルを使用しているプロセスだけが報告されます。</p> <p>複数のファイルグループを指定する場合は、ファイルグループごとにオプションを指定できます。現在使用中のオプションを取り消す場合は、単独のダッシュを使用します。</p> <p>プロセス ID は標準出力に単一行として出力されます。プロセス ID は空白で区切られ、1つの復帰改行 (NEWLINE) で終了します。その他の出力はすべて、標準エラー出力に書き込まれます。</p> <p>fuser はだれでも実行できますが、他のユーザーのプロセスを終了させることができるのは、スーパーユーザーだけです。</p>
オプション	<p>次のオプションを指定できます。</p> <p>-c ファイルシステムのマウントポイントであるファイル、およびマウントされているファイルシステム内のすべてのファイルについて報告します。</p> <p>-f マウントされているファイルシステム内のファイルではなく、指定したファイルについて報告します。</p> <p>-k 各プロセスに SIGKILL シグナルを送ります。このオプションは各プロセスに kill コマンドを発行するので、強制終了メッセージがすぐに表示されないことがあります (kill(2) のマニュアルページを参照)。</p>

- n ファイルに対して非ブロッキング強制ロックを保持しているプロセスだけを表示します。
- s sig 各プロセスにシグナルを送ります。sig オプション引数には、<signal.h> ヘッダーで定義されている記号名の 1 つを指定するか、または 10 進整数のシグナル番号を指定します。sig が記号名で、SIG 接頭辞を付けずに、大文字小文字の区別なしで認識されます。-k オプションは -s KILL または -s 9 と同じです。
- u プロセス ID の後に、括弧で囲んだユーザーのログイン名を表示します。

使用例 例1 マウントポイントおよびファイルについて報告する

次の例は、マウントポイントおよびマウントされているファイルシステム内のファイルについて報告します。

```
fuser -c /export/foo
```

例2 マウントポイントおよびファイルについて報告するときの出力を制限する

次の例は、マウントポイントおよびマウントされているファイルシステム内のファイルについて報告し、その出力を非ブロッキング強制ロックを保持しているプロセスに限定します。

```
fuser -cn /export/foo
```

例3 非ブロッキング強制ロックを保持しているプロセスへ SIGTERM を送信する

次のコマンドは、ファイル /export/foo/my_file に対して非ブロッキング強制ロックを保持しているすべてのプロセスに、SIGTERM を送ります。

```
fuser -fn -s term /export/foo/my_file
```

環境 fuser の実行に影響を与える環境変数、LANG、LC_ALL、LC_CTYPE、LC_MESSAGES、および NLSPATH については、environ(5) 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) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 ps(1), mount(1M), kill(2), mmap(2), signal(3C), attributes(5), environ(5)

fuser(1M)

注意事項 fuser はシステムイメージのスナップショットで動作するため、fuser の実行中にファイルを使用し始めたプロセスを見逃してしまうことがあります。また、ファイルを使用していると報告されたプロセスが、fuser の実行中にファイルの使用を中止していることもあります。このような理由から、-k オプションの使用は勧められません。

名前	pgxconfig, GFXconfig, TSIfgxp_config – PGX32 (Raptor GFX) グラフィックスアクセラレータの設定
形式	<pre> /usr/sbin/pgxconfig [-dev device-filename] [-res video-mode [try noconfirm nocheck]] [-file machine system] [-depth 8 24] [-24only true false] [-cachedpixmap true false] [-defaults] /usr/sbin/pgxconfig [-propt] [-prconf] /usr/sbin/pgxconfig [-help] [-res ?] /usr/sbin/pgxconfig [-i] </pre>
機能説明	<p>pgxconfig ユーティリティーは PGX32 (Raptor GFX) グラフィックスアクセラレータの設定用コマンドで、これには X11 ウィンドウシステムの一部を PGX32 (Raptor GFX) 用にデフォルト設定する機能も含まれます。以前のバージョンでは、このユーティリティーは GFXconfig という名称でした。</p> <p>形式の項に記された pgxconfig の第 1 の形式では、指定したオプションを OWconfig ファイルに保存します。次回の PGX32 (Raptor GFX) デバイス上でのウィンドウシステム起動時に、ここで指定したオプションに従って PGX32 (Raptor GFX) デバイスを初期化します。OWconfig ファイルに保存されたオプションの更新内容は、異なる複数のウィンドウシステムセッションや再起動後のシステムでも有効となります。</p> <p>その他の形式 (2、3、4 番目の形式) を使用した場合はオプション <code>-prconf</code>、<code>-propt</code>、<code>-help</code>、<code>-res ?</code> を呼び出すだけで、OWconfig ファイルに保存されているオプションを更新することはありません。更に、3 番目の形式を使用した場合はこれ以外のオプションはすべて無視されます。</p> <p><code>-i</code> オプションを使用すると、pgxconfig は、対話型モードで起動します。</p> <p>一度にオプションを指定できる PGX32 (Raptor GFX) デバイスの数は、1 つだけです。</p> <p>pgxconfig コマンドで指定できるのは、PGX32 (Raptor GFX) 固有のオプションだけです。デフォルトの表示色数、デフォルトのビジュアルクラスなどを指定する通常のウィンドウシステムのオプションは、openwin コマンド行のデバイス修飾子で指定してください。SUNWxwman パッケージに含まれている Xsun(1) マニュアルページを参照してください。</p> <p>ユーザーは、更新する OWconfig ファイルを指定することもできます。デフォルトでは /usr/openwin ディレクトリツリーにあるマシン固有のファイルが更新されます。別のファイルを指定するには、<code>-file</code> オプションを使用します。たとえば、/etc/openwin ディレクトリツリーにあるシステム共通の OWconfig ファイルを代わりに更新することができます。</p> <p>いずれの標準 OWconfig ファイルでも、書き込み権限があるのはスーパーユーザーだけです。</p>
オプション	次のオプションを指定できます。

GFXconfig(1M)

```
-cachedpixmap true | false
false に設定した場合、PGX32 (Raptor GFX) デバイスは、OpenWindows を実行
するときだけに 24 ビットを使用します。デフォルト値は true です。

アプリケーションの中には、ディスプレイデバイスに書き込むときにキャッシュさ
れたピクスマップを使用するものがあります。このような手法を使用すると、出
力が歪曲され、X サーバーがクラッシュする可能性があります。このような問題が
発生した場合は、-cachedpixmap オプションを false に設定してください。

-defaults
すべてのオプションの値をそれぞれのデフォルト値に戻します。

-depth 8 | 24
スクリーンデプスをピクセル当り 8 または 24 ビットに設定します。24 ビット/ピ
クセルに設定すると、ウィンドウシステムで TrueColor グラフィックスを使用でき
ます。

-dev device-filename
PGX32 (Raptor GFX) 特殊ファイルを指定します。デフォルトは
/dev/fbs/gfxp0、または使用可能な場合であれば /dev/fbs/raptor0 です。

-file machine | system
更新する OWconfig ファイルを指定します。ここで machine が指定された場合
は、/etc/openwin ディレクトリツリーにあるマシン固有の OWconfig ファイル
が使用されます。system が指定された場合は、/usr/openwin ディレクトリツ
リーにある共通の OWconfig ファイルが使用されます。ファイルがない場合は、新
たに生成されます。

-help
pgxconfig コマンド行のオプションと機能の概要を一覧で表示します。

-i
pgxconfig コマンドを対話型モードで起動します。

-prconf
PGX32 (Raptor GFX) のハードウェア構成を表示します。以下に表示例を示しま
す。

--- Hardware Configuration for /dev/fbs/gfxp0 ---
DAC: version 0x0
Type:
Board:
PROM: version 0x0
PROM Information:
RAM:
EDID Data:
Monitor Sense ID:
Card possible resolutions: 640x480x60, 800x600x75, 1024x768x60
1024x768x70, 1024x768x75, 1280x1024x75, 1280x1024x76
1280x1024x60, 1152x900x66, 1152x900x76, 1280x1024x67
960x680x112S, 960x680x108S, 640x480x60i, 768x575x50i,
1280x800x76, 1440x900x76, 1600x1000x66, 1600x1000x76,
vga, svga, 1152, 1280, stereo, ntsc, pal
Monitor possible resolutions: 720x400x70, 720x400x88, 640x480x60
640x480x67, 640x480x72, 640x480x75, 800x600x56,
```

GFXconfig(1M)

```
800x600x60, 800x600x72, 800x600x75, 832x624x75,  
1024x768x87, 1024x768x60, 1024x768x70, 1024x768x75,  
1280x1024x75, 1280x1024x76, 1152x900x66, 1152x900x76,  
1280x1024x67, 960x680x112S, vga, svga, 1152, 1280  
stereo
```

```
Current resolution setting: 1280x1024x76
```

```
Possible depths:
```

```
Current depth: 8
```

-propt

-file オプションで指定された OWconfig ファイルに書かれた PGX32 (Raptor GFX) オプションの値のうち、-dev オプションで指定されたデバイスに対するものすべてを表示します。pgxconfig の呼び出しが終了した後に、OWconfig ファイルに書き込まれるオプションの値を表示します。以下に表示例を示します。

```
--- OpenWindows Configuration for /dev/fbs/gfxp0 ---
```

```
OWconfig: machine
```

```
Video Mode: not set
```

```
Depth: 8+24
```

-res video-mode [try | noconfirm | nocheck]

PGX32 (Raptor GFX) デバイスに接続されているモニターの制御に使用する組み込みのビデオモードを指定します。

video-mode には、以下のいずれかの書式で指定できます。

widthxheightxrate

width はピクセル単位の画面の幅、*height* はピクセル単位の画面の高さ、*rate* は垂直方向の画面再描画周期です。-res では、再描画周期 *rate* の前の *x* は、@ でも代用できます。つまり周期の指定に限っては、たとえば 1280x1024@76 のような形式にも対応しています。オプションに -res ? を付けて pgxconfig を実行すると (コマンド形式の項に記された 3 番目の形式)、ビデオモードの一覧が表示されます。ビデオボードとモニターの両方が、すべての解像度をサポートしているわけではありません。noconfirm または nocheck オプションを指定しないで、ボードのサポートしていない解像度を入力した場合には、pgxconfig はその要求を許可しません。また、nocheck オプションを指定しないで、モニターのサポートしていない解像度を指定した場合には、その値を適用する前に確認を求めてきます。

Symbolic names

便宜上、一部のビデオモードには記号名が定義されています。*widthxheightxrate* の形式の代わりに、記号名を -res の引数として指定することができます。記号名 none は、ウィンドウシステムを実行すると、画面の解像度が現在デバイスにプログラムされているビデオモードになることを意味します。

```
svga      1024x768x60
```

GFXconfig(1M)

1152	1152x900x76
1280	1280x1024x76
vga	640x480x60
none	デフォルトのコンソールの解像度

-res オプションには、ビデオモードの直後に次の追加引数を指定することができます。追加引数は、単独でも複数でも指定できます。

noconfirm	-res オプションを指定した際に、システムが使用可能であっても、表示出力のない状態になる場合があります。このような状況は、特定のコードが読み込まれた際のモニターセンスコードにあいまいさがあった場合などに発生します。このような事態を避けるために pgxconfig のデフォルトの動作では、この問題についての警告メッセージと、処理を継続するかどうかを確認するメッセージを表示します。noconfirm オプションを指定すると、pgxconfig コマンドはこの確認をせずに、要求のあったビデオモードにプログラムします。このオプションは、pgxconfig がシェルスクリプトから実行されている場合に便利です。
nocheck	このオプションを指定すると、モニターセンスコードに基づく通常のエラーチェックが行われません。ユーザーによって指定されたビデオモードは、現在接続されているモニターに適切かどうかにかかわらず受け付けられます。このオプションは、PGX32 (Raptor GFX) デバイスに異なるモニターを接続する場合に便利です。このオプションの指定は、noconfirm の指定も兼ねます。
try	このオプションを指定すると、指定したビデオモードを適用する前にテストすることができます。まず、指定したモードに基づいたテストパターンが表示されます。テストパターンが正常に表示された場合は、「y」(次いでキャリッジリターン)を入力します。「y」以外の文字を入力(次いでキャリッジリターンを入力)した場合は、「no」となります。

-res ?

PGX32 およびモニターがサポートする解像度の一覧を表示します。

-24only

Openwindows の実行時に、PGX32 (Raptor GFX) デバイスが 24 ビットカラーのみを使用するようにします。

デフォルト

pgxconfig コマンド行で指定されていないオプションについては、対応する OWconfig ファイル中のオプションは更新されず、ファイル内の値がそのまま使用されます。ただし、-depth と -24only については、その限りではありません。

ウィンドウシステムを実行する際に、pgxconfig による PGX32 (Raptor GFX) のオプションの指定がまったくなかった場合は、デフォルト値が使用されます。オプションのデフォルト値は次のとおりです。


```
-dev          /dev/fbs/gfxp0
-file         system
-res         none
```

-res オプションのデフォルト値 **none** とは、ウィンドウシステムが実行された場合に、画面解像度がそのデバイスに現在プログラムされているビデオモードになることを意味しています。

使用例 例 1 モニターの種類を変更する。

モニターの種類を、垂直周波数 76 Hz で解像度 1280 x 1024 に変更する例を以下に示します。

```
example# /usr/sbin/pgxconfig -res 1280x1024x76
```

ファイル /dev/fbs/gfxp0
デバイス特殊ファイル

/usr/openwin/server/etc/OWconfig
システム構成ファイル

/etc/openwin/server/etc/OWconfig
マシン構成ファイル

関連項目 PGX32 PCI グラフィックカード インストールマニュアル

halt(1M)

名前	halt, poweroff – プロセッサの停止				
形式	<code>/usr/sbin/halt [-dlnqy]</code> <code>/usr/sbin/poweroff [-dlnqy]</code>				
機能説明	<p>halt および poweroff ユーティリティは、保留されている情報をディスクに書き出してから、プロセッサを停止させます。poweroff ユーティリティは、可能であれば、マシンの電源を自動的に切断します。</p> <p>halt および poweroff ユーティリティは通常、システム停止をシステムログデーモン syslogd(1M) に送信し、ログインアカウントファイル /var/adm/wtmpx に停止情報を記録します。ただし、-n または -q オプションが指定されている場合、これらのアクションは禁止されます。</p>				
オプション	<p>次のオプションを指定できます。</p> <ul style="list-style-type: none"> -d 再起動の前にシステムクラッシュダンプを強制的に実行します。システムクラッシュダンプの設定については、dumpadm(1M) のマニュアルページを参照してください。 -l halt を実行したユーザーに関するメッセージを、システムログデーモン syslogd(1M) に送信しないようにします。 -n 停止前の sync(1M) を行いません。 -q 即時停止。通常の停止手続きを実行しません。 -y ダイアルアップ端末からでも、システムを停止します。 				
ファイル	/var/adm/wtmpx ユーザーアクセスと管理情報の履歴				
属性	<p>次の属性については、attributes(5) のマニュアルページを参照してください。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">属性タイプ</th> <th style="text-align: center;">属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWcsu</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWcsu
属性タイプ	属性値				
使用条件	SUNWcsu				
関連項目	dumpadm(1M), init(1M), reboot(1M), shutdown(1M), sync(1M), syslogd(1M), inittab(4), attributes(5)				
注意事項	halt ユーティリティおよび poweroff ユーティリティは、/etc/rc<num>.d のスクリプトを実行しません。また、inittab(4) の停止アクションも実行しません。システムサービスを完全に停止させるためには、shutdown(1M) を使用するか、または init(1M) を使用して Solaris システムを再起動してください。				

名前	init, telinit – プロセス制御の初期化	
形式	/sbin/init [0123456abcQqSs] /etc/telinit [0123456abcQqSs]	
機能説明	init は汎用のプロセス生成コマンドです。その主な役割は、ファイル /etc/inittab に格納されている情報に基づいてプロセスを生成することです。	
定義されている実行レベル	任意の時点で、システムは 8 つの実行レベルのいずれか 1 つにあります。実行レベルはソフトウェア構成で、ここでは選択されたプロセスグループだけが存在します。各実行レベルに対する、init によって生成されるプロセスは、/etc/inittab で定義されています。init は 8 つの実行レベル、つまり 0-6 および s または S (S と s は同じ) のいずれか 1 つを取ることができます。実行レベルは、特権ユーザーが /sbin/init を実行すると変更されます。/sbin/init は、起動時にオペレーティングシステムによって生成された元の init に適切なシグナルを送り、移行する実行レベルを通知します。	
init とシステムの起動	システムが起動されると、init が呼び出され、次のイベントが発生します。最初に init は、/etc/default/init を読み込み、環境変数を設定します。通常はここで、TZ (タイムゾーン) およびロケール関連の環境、すなわち LANG、LC_CTYPE など が設定されます (このページの最後にある「ファイル」の項を参照)。次に init は、/etc/inittab を調べ、initdefault エントリを探します (inittab(4) のマニュアルページを参照)。initdefault のエントリ	
	存在する場合	init は通常、このエントリで指定された実行レベルを、開始時の最初の実行レベルとして使用します。
	存在しない場合	init はユーザーに、システムコンソールから実行レベルを入力するように要求します。
	s または S	init はシングルユーザー状態に移行します。この状態では、システムコンソールデバイス (/dev/console) が読み書き用にオープンされ、コマンド /sbin/su (su(1M) のマニュアルページを参照) が呼び出されます。init または telinit のどちらか一方を使用して、システムの実行レベルを変更します。(ファイルの終わり (EOF) を使用して) シェルを終了させたが、/etc/inittab が存在しない場合、init は再びシングルユーザー状態に戻すだけです。
	0-6	init は対応する実行レベルに移行します。実行レベル 0、5、および 6 はシステム停止用に予約されています。実行レベル 2、3、および 4 は、マルチユーザー操作用に使用で

init(1M)

きます。

電源投入後に、init が最初のシングルユーザー状態以外の実行レベルに移行する場合、init はまず、`/etc/inittab` を走査して `boot` および `bootwait` エントリを探します (`inittab(4)` のマニュアルページを参照)。移行する実行レベルがエントリの実行レベルと一致している場合は、`/etc/inittab` の他のプロセスが実行される前に、これらのエントリが実行されます。このようにすると、ファイルシステムのマウントなど、オペレーティングシステムの特異な初期化を先に終えてから、ユーザーにシステムを使用させることができます。次に init は、`/etc/inittab` を走査し、その実行レベルで処理すべき他のすべてのエントリを実行します。

`/etc/inittab` の各プロセスを生成するために、init は各エントリを読み込み、再生成が必要なエントリごとに、子プロセスを生成します。`/etc/inittab` で指定された全プロセスを生成すると、init は、以下の状態のために待機します。子孫プロセスの1つが消滅する、`powerfail` シグナルを受けると、システムの実行レベルの変更を要求する他の init または `telinit` プロセスから送られるシグナルを受けると。これらの状態のいずれか1つが発生すると、init は `/etc/inittab` を再検査します。

inittab に関するその他の事項

`/etc/inittab` にはいつでも新しいエントリを追加できますが、init は上記の3つの状態のいずれかが発生するまで待機し続けるので、`/etc/inittab` を再検証しません。この状況を回避するには、`init Q` または `init q` コマンドを使用して、init が `/etc/inittab` をただちに再検査するようにします。

起動時に init が呼び出されたとき、また、システムがシングルユーザー状態から別の実行状態に移行するたびに、init はコンソールの `ioct1(2)` 状態をファイル `/etc/ioct1.syscon` に格納されているモードに設定します。init はシングルユーザー状態に移行するたびに、このファイルに書き込みを行います。

実行レベルの変更

実行レベルの変更が要求されると、init は目標の実行レベルで定義されていないすべてのプロセスに警告シグナル (`SIGTERM`) を送ります。init は5秒間待機したのち、終了シグナル (`SIGKILL`) を送信することによって、これらのプロセスを強制終了させます。

init によって生成されたプロセスが終了したことを通知するシグナルを受信すると、init は、その事実と原因を `/var/adm/utmpx` と `/var/adm/wtmpx` (存在していれば) に記録します (`who(1)` を参照)。生成されたプロセスの履歴は `/var/adm/wtmpx` に記録されます。

init は `powerfail` シグナル (`SIGPWR`) を受信した場合、`/etc/inittab` を走査し、タイプが `powerfail` と `powerwait` の特異なエントリを探します。これらのエントリが呼び出されてから (実行レベルが許可する場合)、その後の処理が実行されます。このように、init はオペレーティングシステムの停止中にさまざまなクリーンアップおよび記録機能を実行します。

`/etc/defaults/init` の環境変数

タイムゾーンや文字の書式といった環境変数のデフォルト値を `/etc/default/init` で設定できます。環境変数の一覧については、「ファイル」の項を参照してください。

telinit	<code>/sbin/init</code> にリンクされている <code>telinit</code> は、 <code>init</code> のアクションを指示する場合に使用します。 <code>telinit</code> は、1 文字の引数を取り、適切なアクションを実行するように、 <code>init</code> にシグナルを送ります。																				
セキュリティ	<p><code>init</code> は <code>pam(3PAM)</code> を使用してセッションを管理します。<code>/etc/pam.conf</code> に記述されている PAM 構成ポリシーにより、<code>init</code> に使用されるセッション管理モジュールが指定されます。以下に、<code>pam.conf</code> ファイルの抜粋を示します。UNIX セッション管理モジュールを使用する <code>init</code> のエントリが指定されています。</p> <pre>init session required pam_unix_session.so.1</pre> <p><code>init</code> サービスに対応するエントリがない場合、「other」のサービスのエントリが使用されます。</p>																				
オプション	<table border="0"> <tr> <td style="vertical-align: top; padding-right: 10px;">0</td> <td>ファームウェアモードに移行します。</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;">1</td> <td>システムをシステム管理者モードにします。すべてのローカルファイルシステムがマウントされます。いくつかの重要なコアプロセスだけが実行を続けます。このモードはオプションのユーティリティパッケージをインストールする場合など、管理作業を行うためのものです。すべてのファイルにアクセスできます。ユーザーはシステムにログインできません。</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;">2</td> <td>システムをマルチユーザーモードにします。すべてのマルチユーザー環境用の端末プロセスおよびデーモンが生成されます。一般に、この状態を、マルチユーザー状態といいます。</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;">3</td> <td>ネットワークを介してローカル資源を使用できるようにすることで、マルチユーザーモードを拡張します。</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;">4</td> <td>代替のマルチユーザー環境構成として定義できます。システム運用上は必要なく、通常は使用されません。</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;">5</td> <td>電源を切断しても問題が起きないように、マシンを停止させます。可能であれば、マシンの電源を自動的に切断させます。</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;">6</td> <td>オペレーティングシステムを停止し、<code>/etc/inittab</code> の <code>initdefault</code> エントリで定義された状態で再起動します。</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;">a, b, c</td> <td><code>/etc/inittab</code> エントリに a、b、または c という実行レベルが設定されているときだけ処理します。これらは擬似状態であり、特定のコマンドを実行するように定義できますが、現在の実行レベルが変更されるわけではありません。</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;">Q, q</td> <td><code>/etc/inittab</code> を再検査します。</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;">S, s</td> <td>シングルユーザーモードに移行します。適切な形式の <code>/etc/inittab</code> ファイルがなくてもかまわない唯一の実行レベルです。このファイルが存在しない場合、デフォルトで <code>init</code> が入ることができる唯一の正当な実行レベルはシングルユーザーモードだけです。シングルユーザーモードでは、基本システム動作に必要なファイルシステムがマウントされます。システムがシングルユーザーモードに移行すると、これらのファイルシステムは (リ</td> </tr> </table>	0	ファームウェアモードに移行します。	1	システムをシステム管理者モードにします。すべてのローカルファイルシステムがマウントされます。いくつかの重要なコアプロセスだけが実行を続けます。このモードはオプションのユーティリティパッケージをインストールする場合など、管理作業を行うためのものです。すべてのファイルにアクセスできます。ユーザーはシステムにログインできません。	2	システムをマルチユーザーモードにします。すべてのマルチユーザー環境用の端末プロセスおよびデーモンが生成されます。一般に、この状態を、マルチユーザー状態といいます。	3	ネットワークを介してローカル資源を使用できるようにすることで、マルチユーザーモードを拡張します。	4	代替のマルチユーザー環境構成として定義できます。システム運用上は必要なく、通常は使用されません。	5	電源を切断しても問題が起きないように、マシンを停止させます。可能であれば、マシンの電源を自動的に切断させます。	6	オペレーティングシステムを停止し、 <code>/etc/inittab</code> の <code>initdefault</code> エントリで定義された状態で再起動します。	a, b, c	<code>/etc/inittab</code> エントリに a、b、または c という実行レベルが設定されているときだけ処理します。これらは擬似状態であり、特定のコマンドを実行するように定義できますが、現在の実行レベルが変更されるわけではありません。	Q, q	<code>/etc/inittab</code> を再検査します。	S, s	シングルユーザーモードに移行します。適切な形式の <code>/etc/inittab</code> ファイルがなくてもかまわない唯一の実行レベルです。このファイルが存在しない場合、デフォルトで <code>init</code> が入ることができる唯一の正当な実行レベルはシングルユーザーモードだけです。シングルユーザーモードでは、基本システム動作に必要なファイルシステムがマウントされます。システムがシングルユーザーモードに移行すると、これらのファイルシステムは (リ
0	ファームウェアモードに移行します。																				
1	システムをシステム管理者モードにします。すべてのローカルファイルシステムがマウントされます。いくつかの重要なコアプロセスだけが実行を続けます。このモードはオプションのユーティリティパッケージをインストールする場合など、管理作業を行うためのものです。すべてのファイルにアクセスできます。ユーザーはシステムにログインできません。																				
2	システムをマルチユーザーモードにします。すべてのマルチユーザー環境用の端末プロセスおよびデーモンが生成されます。一般に、この状態を、マルチユーザー状態といいます。																				
3	ネットワークを介してローカル資源を使用できるようにすることで、マルチユーザーモードを拡張します。																				
4	代替のマルチユーザー環境構成として定義できます。システム運用上は必要なく、通常は使用されません。																				
5	電源を切断しても問題が起きないように、マシンを停止させます。可能であれば、マシンの電源を自動的に切断させます。																				
6	オペレーティングシステムを停止し、 <code>/etc/inittab</code> の <code>initdefault</code> エントリで定義された状態で再起動します。																				
a, b, c	<code>/etc/inittab</code> エントリに a、b、または c という実行レベルが設定されているときだけ処理します。これらは擬似状態であり、特定のコマンドを実行するように定義できますが、現在の実行レベルが変更されるわけではありません。																				
Q, q	<code>/etc/inittab</code> を再検査します。																				
S, s	シングルユーザーモードに移行します。適切な形式の <code>/etc/inittab</code> ファイルがなくてもかまわない唯一の実行レベルです。このファイルが存在しない場合、デフォルトで <code>init</code> が入ることができる唯一の正当な実行レベルはシングルユーザーモードだけです。シングルユーザーモードでは、基本システム動作に必要なファイルシステムがマウントされます。システムがシングルユーザーモードに移行すると、これらのファイルシステムは (リ																				

init(1M)

モートファイルサーバーが提供している場合でも) マウントされたままになります。他のローカルファイルシステムもマウントされたままになります。シングルユーザーモードへの切り替え時に、init または init.d によって開始された、マルチユーザーモードでしか実行してはならないプロセスはすべて強制終了されます。さらに、utmpx エントリが設定されているプロセスもすべて強制終了されます。この最後の状態では、SAC が起動したすべてのポートモニターが確実に強制終了され、これらのポートモニターによって起動されたサービスも、ttymon login ログインサービスを合わせてすべて強制終了されます。

ファイル	/dev/console	システムコンソール装置
	/etc/default/init	環境変数とそのデフォルト値を指定します。たとえば、タイムゾーン変数 TZ の場合、TZ=US/Pacific のように指定できます。変数は次のとおりです。
	TZ	タイムゾーン情報 (ctime(3C) のマニュアルページを参照) またはタイムゾーン情報ファイル /usr/share/lib/zoneinfo の名前のどちらか一方を指定します。
	CMASK	init が使用し、すべてのプロセスが init プロセスから継承するマスク (umask(1) のマニュアルページを参照)。設定されていない場合、init はカーネルから継承したマスクを使用します。CMASK の設定に関係なく、init は必ず、022 で umask の適用を試みてからファイルを作成します。
	LC_CTYPE	文字の種類の情報
	LC_MESSAGES	翻訳メッセージ
	LC_MONETARY	通貨の書式情報
	LC_NUMERIC	数値の書式情報
	LC_TIME	時刻の書式情報
	LC_ALL	設定されている場合、他のすべての LC_* 環境変数でこの値が使用されます。
	LANG	LC_ALL が設定されてなく、かつ特定の LC_* も設定されていない場合は、これら環境変数に LANG の値が使用されます。
	/etc/initpipe	内部通信用の名前付きパイプ

/etc/inittab	init によるプロセスディスパッチ制御
/etc/ioctl.syscon	シングルユーザー状態に移行したときに init によって保存された、コンソールの ioctl (入出力制御) 状態
/var/adm/utmpx	ユーザーアクセスおよび管理情報
/var/adm/wtmpx	ユーザーアクセスおよび管理情報の履歴

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 login(1), sh(1), stty(1), who(1), shutdown(1M), su(1M), ttymon(1M), ioctl(2), kill(2), ctime(3C), pam(3PAM), inittab(4), pam.conf(4), utmpx(4), attributes(5), pam_authtok_check(5), pam_authtok_get(5), pam_authtok_store(5), pam_dhkeys(5), pam_passwd_auth(5), pam_unix(5), pam_unix_account(5), pam_unix_auth(5), pam_unix_session(5), termio(7I)

診断 2 分間に 10 回以上、/etc/inittab 内の 1 つのエントリが再起動されていることを検出すると、init はそのエントリのコマンド文字列に誤りがあるとみなして、システムコンソール上にエラーメッセージを表示します。さらに 5 分経過するか、あるいはユーザーが生成した init または telinit からシグナルを受信するまで、init はそのエントリを再起動することを拒否します。このようにすることで、inittab ファイルに入力ミスがあった場合や /etc/inittab で参照されているプログラムが削除された場合でも、init がシステム資源を使い果たすのを防止できます。

注意事項 init および telinit を実行できるのは、特権ユーザーだけです。

/etc/inittab で、s または S 状態をむやみに使用してはなりません。このファイルを変更するときには注意すべきことは、この状態を initdefault 以外の行に追加しないことです。

/etc/inittab の initdefault エントリでデフォルトの状態が指定されていない場合は、状態 6 になります。その結果、システムはファームウェアへのアクセスと再起動を繰り返すループに陥ります。

システムの起動時に utmpx ファイルを作成できない場合、システムは /etc/inittab の initdefault エントリで指定されている状態に関係なく、“s” の状態で起動します。/var ファイルシステムにアクセスできない場合には、この状況が発生することがあります。

システムが s または S 状態へ移行するときに、/etc/nologin ファイル (nologin(4) のマニュアルページを参照) が作成されます。その後、実行レベル 2 に移行すると、/etc/rc2.d ディレクトリにあるスクリプトによってこのファイルが削除されます。

init(1M)

init は内部通信に名前付きパイプ /etc/initpipe を使用します。

pam_unix(5) モジュールは、将来のリリースではサポートされなくなる可能性があります。同様の機能は、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)、および pam_unix_session(5) で提供されています。

名前	installer – Solaris Web Start インストールユーティリティ	
形式	installer [-locales <i>list</i>] [-nodisplay] [-noconsole] [-debug]	
機能説明	<p>installer ユーティリティは、Solaris Web Start のウィザードを起動します。ユーザーは、表示される一連のウィザードの指示に沿ってインストールを行います。</p> <p>installer ユーティリティは、Solaris に同梱されている別ソフトウェアの CD のトップディレクトリに含まれています。</p> <p>installer が含まれている CD にデスクトップのファイルマネージャからアクセスしている場合は、installer のアイコンをダブルクリックすることによって、ウィザードを起動することができます。スーパーユーザーになっていない場合は、スーパーユーザーのパスワードを入力するように指示メッセージが表示されます。</p> <p>installer ユーティリティは、他の UNIX スクリプトから実行することもできます。スクリプトから実行する場合は通常、installer ユーティリティを -nodisplay オプション付きで実行するようにします。対話形式でないスクリプトを使用する場合は、-nodisplay オプションを追加します。</p>	
オプション	<p>次のオプションを指定できます。</p> <p>-locales <i>list</i> インストールする製品のロケールを選択します。<i>list</i> に指定されたロケールがインストールメディアに提供されていれば、そのロケールの製品がインストールされます。ロケールは、-locales オプションの後にカンマで区切って指定します。たとえば次のように指定すると、</p> <p style="padding-left: 40px;">installer -locales fr,de,it フランス語 (fr)、ドイツ語 (de)、イタリア語 (it) ロケールの製品がインストールされます。</p> <p>-nodisplay GUI (グラフィカルユーザーインタフェース) を使用せずにインストールを行います。-locales オプションでロケールを指定していない場合は、デフォルトのロケール (英語) の製品をインストールします。</p> <p>-noconsole 対話型テキストコンソールデバイスを使用せずにインストールを行います。対話型でない UNIX スクリプトで使用する場合に、-nodisplay オプションと組み合わせて使用すると便利です。</p> <p>-debug インストール中の処理状況を示す情報を出力します。このオプションは、おもにインストール処理を診断したいときに使用します。</p>	
ファイル	/var/sadm/install/logs	インストールログファイルの保存ディレクトリ
関連項目	prodreg(1M)	

install_scripts(1M)

名前	install_scripts, add_install_client, add_to_install_server, modify_install_server, rm_install_client, setup_install_server, check – Solaris ソフトウェアをインストールするためのスクリプト
形式	<pre>cdrom-mnt-pt/Solaris_XX/Tools/add_install_client [-i IP_address] [-e Ethernet_address] [-s server_name : path] [-c server_name : path] [-n [server] : name_service [(netmask)]] [-p server_name : path] [-f boot_file_name] host_name platform_group cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server_name:path] [-c server_name:path] [-p server_name:path] [-t install_boot_image_path] [-f boot_file_name] platform_name platform_group cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server_name:path] [-c server_name:path] [-p server_name:path] [-t install_boot_image_path] [-f boot_file_name] -e Ethernet_address [-b property=value] platform_group cdrom-mnt-pt/Solaris_XX/Tools/add_to_install_server [-s] [-p product_image_path] install_server_path cdrom-mnt-pt/Solaris_XX/Tools/jumpstart_sample/check [-p install_dir_path] [-r rulesfile] cdrom-mnt-pt/modify_install_server [-p] install_dir_path installer_miniroot_path cdrom-mnt-pt/Solaris_XX/Tools/rm_install_client host_name cdrom-mnt-pt/Solaris_XX/Tools/setup_install_server [-b] [-t install_boot_image_path] [-w wanboot_image_path] install_dir_path</pre>
機能説明	<p>これらのコマンドは、Solaris Software and Solaris Installer CD のスライス 0 に置かれています。Solaris CD をすでにローカルディスクにコピーしている場合、<code>cdrom_mnt_pt</code> はコピーされている Solaris CD へのパスになります。これらのコマンドは、さまざまなインストール作業に使用できます。</p> <p>Solaris_XX の XX は、使用している Solaris リリースのバージョン番号です。</p> <p><code>add_install_client</code> コマンドには 3 種類の形式があります。「形式」の項を参照してください。</p> <p>ネットワークを使用してインストールするクライアントを追加する場合は、次の形式の <code>add_install_client</code> を使用します (これらのコマンドは <code>bootparams(4)</code> ファイルを更新します)。<code>add_install_client</code> コマンドは、インストールサーバーの Solaris インストールイメージ (マウントした CD またはディスクにコピーした Solaris CD) あるいはブートサーバーの起動ディレクトリ (ブートサーバーが必要な場合) から実行する必要があります。Solaris インストールイメージまたはブート専用ディレクトリの Solaris リリースは、クライアントにインストールする予定の Solaris リリースと同じでなければなりません。</p>

```
cdrom-mnt-pt/Solaris_XX/Tools/add_install_client [-i IP_address]
[-e Ethernet_address] [-s server_name : path] [-c server_name : path]
[-n [server] : name_service [ ( netmask)]] [-p server_name : path]
host_name platform_group
```

プラットフォームグループ内のあるプラットフォームのインスタンスをインストールサーバーに追加する場合は、次の形式の `add_install_client` コマンドを使用します。このグループは DHCP を使用して起動および構成されます。スクリプトは、必要な構成作業をサーバー上で実行し、ユーザーがそのグループのために DHCP サーバーに追加しなければならないデータを出力します。

```
cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server:path]
[-c server:path] [-p server:path] [-t install boot image path]
[-f boot file name] platform_name platform_group
```

インストールサーバーに単一のクライアントを追加する場合は、次の形式の `add_install_client` コマンドを使用します。このクライアントは DHCP を使用して起動および構成されます。スクリプトは、必要な構成作業をサーバー上で実行し、ユーザーがそのクライアントのために DHCP サーバーに追加しなければならないデータを出力します。すでに使用しているものにも、上記のように `-f` フラグを追加する必要があります。`-f` を使用すると、ユーザーは所定のクライアントに使用する起動ファイルの名前を指定できます。

```
cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d
[-s server_name:path] [-c server_name:path] [-p server_name:path]
[-t install_boot_image_path] [-f boot_file_name] -e Ethernet_address
platform_group
```

注 - x86 アーキテクチャの Pre-boot eXecution Environment (PXE) クライアントを登録する場合は、必ず `-d` オプションを使用してください。x86 PXE クライアントは構成に DHCP を使用します。

他の Solaris CD とネットワークインストールサーバー上の既存のイメージをマージする場合は、`add_to_install_server` を使用します。マージ可能な各 CD (現時点では OS CD 2 および Language CD) には専用の `add_to_install_server` スクリプトが含まれています。配布された CD に含まれているもの以外の `add_to_install_server` スクリプトを使用しないでください。

`rules` ファイル (カスタム JumpStart インストールを使用する場合に限り必要) のルールを検証する場合は、`check` を使用します。

既存のネットワークインストールサーバーのミニルートを Solaris Installation CD のミニルートに置き換える場合は、`modify_install_server` を使用します。ネットワークインストールサーバーのインストール時のユーザーインタフェースが、Solaris Installation CD の Web Start ユーザーインタフェースに変更されます。

install_scripts(1M)

modify_install_server コマンドを使用する場合は、既存のインストールイメージ (setup_install_server を使用して作成) があらかじめ存在していなければなりません。

ネットワークインストール用のクライアントを削除する場合は、rm_install_client を使用します (このコマンドは bootparams(4) ファイルを更新します)。

Solaris CD を (インストールサーバーを設定するために) ディスクにコピーする、(WANboot インストールサーバーを設定するために) WANboot ミニルートイメージを構築する、または Solaris CD のブートソフトウェアだけを (ブートサーバーを設定するために) ディスクにコピーする場合は、setup_install_server を使用します。ネットワーク経由でクライアントをインストールするには、インストールサーバーが必要です。インストールサーバーとインストールするクライアントが異なるサブネットに属している場合、ネットワークインストールを行うには、ブートサーバーも必要です (ブートサーバーはクライアントのサブネットに配置する必要があります)。

オプション add_install_client には、次のオプションを指定できます。

-b *property=value*

ブートサーバーの TFTP ディレクトリ (デフォルトは /tftpboot) にあるクライアント特有の bootenv.rc ファイルにプロパティ値を設定します。クライアントに特有のブートプロパティを設定する場合に、このオプションを使用します。bootpath プロパティを使用すると、デバイス構成補助を使用して手動で選択しなくても、ブートデバイスを自動的に選択できます。boot-args プロパティを使用すると、JumpStart インストールを自動的に選択できます。eeprom(1M) を参照してください。

このオプションは x86 クライアントにのみ使用できます。このオプションを使用するときには、必ず -d オプションと -e を同時に指定してください。

-c *server_name:path*

このオプションが必要なのは、カスタム JumpStart インストール用の JmpStart ディレクトリを指定する場合だけです。server_name は、JumpStart ディレクトリが置かれているサーバーのホスト名です。path は JumpStart ディレクトリの絶対パスです。

-d

DHCP クライアントを指定します。

-e *Ethernet_address*

インストールするシステムの Ethernet アドレスを指定します。

-f

インストールするクライアントの boot_file_name を指定します。

-i *IP_address*

インストールするクライアントの IP アドレスを指定します。

-n [*server*]: *name_service*[(*netmask*)]

このオプションでは、システム構成時に使用するネームサービスを指定します。bootparams(4) ファイル内の ns キーワードを設定します。

name_service

有効なエントリは nis、nisplus、および none です。

netmask

数字を 4 つずつピリオドで区切って並べたもので、IP アドレスのどの部分がネットワーク部分で、どれがホスト部分かを指定します。

server

サーバーの名前または指定したネームサービスの IP アドレス。指定したサーバーが異なるサブネット上にある場合は、クライアントからサーバーにアクセスするために、*netmask* が必要なことがあります。

-p *server_name* : *path*

このオプションは、システム情報またはネットワーク情報を前もって設定するために必要な、ユーザーが定義した *sysidcfg* ファイルの位置です。*server_name* は有効なホスト名または IP アドレスです。*path* は JumpStart ディレクトリの絶対パスです。

-s *server_name* : *path*

このオプションが必要なのは、ブートサーバーから *add_install_client* を使用する場合だけです。サーバー名およびこのインストールで使用する Solaris インストールイメージの絶対パスを指定します。*path* はマウントされている Solaris CD のパスまたは Solaris CD のコピーが置かれているディレクトリのパスです。

add_to_install_server コマンドには、次のオプションを指定できます。

-p

コピーする CD (補助的な製品が収められている) の位置を指定します。

-s

ユーザーが、インストールする必要がある製品だけをリストから選択できるようにします。

check コマンドには、次のオプションを指定できます。

-p *install_dir_path*

使用中のシステムの *check* スクリプトではなく、指定された Solaris インストールイメージの *check* スクリプトを使用することによって、ルールファイルを検査します。*install_dir_path* は、ローカルディスク上またはマウントした Solaris CD 上の Solaris インストールイメージのパスです。

旧バージョンの Solaris が稼働しているシステムでは、このオプションを使用して、最新バージョンの *check* を実行できます。

-r *rulesfile*

rules 以外の名前のルールファイルを指定します。このオプションを使用すると、ルールの妥当性を検証してから、ルールファイルに組み込むことができます。*check* はルールが有効であるかどうかを報告するだけで、カスタム JumpStart インストールに必要な *rules.ok* ファイルは作成しません。

modify_install_server コマンドには、次のオプションを指定できます。

install_scripts(1M)

-p
このオプションは、*install_dir_path/Solaris_XX/Tools/Boot.orig* にすでに存在しているイメージのミニルートを保持します。

setup_install_server コマンドには、次のオプションを指定できます。

-b
サーバーの設定を、ブートサーバー専用にします。

-t
代替ミニルートを指定できるようにします。

-w
WANboot ミニルートイメージを構築します。

オペランド add_install_client コマンドには、次のオペランドを指定できます。

host_name
インストールするクライアントの名前です。

platform_group
特定のソフトウェアを配布するために、ベンダーが定義したハードウェアプラットフォームグループ。以下に有効なプラットフォームグループの例を示します。

システム	プラットフォームグループ
x86	i86pc
Sun Fire 4800	sun4c

システムのプラットフォームグループを調べるには、(-m オプションを指定して) `uname(1)` コマンドを使用します。

platform_name
システムのプラットフォーム名を調べるには、-i オプションを指定して `uname(1)` コマンドを使用します。

次に、`uname command` コマンドを使用して Ultra 1 のシステムプラットフォーム名を調べる例を示します。

```
uname -i
システムは、次のように応答します。
```

```
SUNW,Ultra-1
システムのプラットフォーム名は SUNW,Ultra1 です。
```

次のコマンドは、Ultra 1 用の `add_install_client` を呼び出します。

```
add_install_client -d SUNW,Ultra-1 sun4u
```

IA32 プラットフォームでは、プラットフォーム名はつねに `SUNW.i86pc` です。

次のコマンドは、IA32 プラットフォーム用の `add_install_client` を呼び出します。

```
add_install_client -d SUNW.i86pc i86pc
```

`rm_install_client` コマンドには、次のオペランドを指定できます。

host_name

削除するクライアントの名前です。

`setup_install_server` コマンドには、次のオペランドを指定できます。

install_dir_path

Solaris ソフトウェアのコピー先となるディレクトリの絶対パスです。このディレクトリは空でなければなりません。

wanboot_image_path

WANboot ミニルートイメージを含むファイルが作成されるディレクトリの絶対パスです。

使用例 例 1 `add_install_client` の使用

次の `add_install_client` コマンドは、マウント済みの Solaris CD からネットワークインストール用のクライアントを Solaris 8 のインストールサーバーに追加します。

```
example# cd /cdrom/cdrom0/s0/Solaris_9/Tools
example# ./add_install_client system_2/sun4u
```

例 2 `add_install_client` の使用

次の `add_install_client` コマンドは、ネットワークインストール用のクライアントをインストールサーバーに追加します。カスタム JumpStart インストールを実行するためのルールファイルおよびプロファイルファイルが置かれている JumpStart のサーバーとディレクトリのパスを `-c` オプションで指定します。Solaris CD は Solaris 8 上の `/export/install` ディレクトリにコピーされています。

```
example# cd /export/install/Solaris_8/Tools
example# ./add_install_client -c install_server:/jumpstart system_1 i86pc
example# ./add_install_client -c install_server:/jumpstart system_2 i86pc
```

例 3 `add_install_client` の使用

次の `add_install_client` コマンドは、起動ファイル `sun4u.solaris8` を使用する特定の sun4u プラットフォームマシン (8:0:20:99:88:77) に対するサポートを追加します。

```
example# add_install_client -d -f sun4u.solaris8 -e 8:0:20:99:88:77 sun4u
```

例 4 `add_install_client` の使用

次の `add_install_client` コマンドは、PXE 規格を使用してネットワークから起動する x86 クライアントを追加します。

```
example# add_install_client -d -s svrname:/mnt/export/root SUNW.i86pc i86p
```

install_scripts(1M)

例 4 add_install_client の使用 (続き)

例 5 add_to_install_server の使用

次の add_to_install_server コマンドは、CD 上のすべての製品ディレクトリにあるパッケージを Solaris 8 上の既存のインストールサーバーにコピーします。

```
example# cd /cdrom/cdrom0/s0
example# ./add_to_install_server /export/Solaris_8
```

例 6 check の使用

次の check コマンドは、カスタム JumpStart インストールに使用するルールファイルの構文を検査します。

```
example# cd jumpstart_dir_path
example# ./check -p /cdrom/cdrom0/s0
```

例 7 modify_install_server の使用

次の modify_install_server コマンドは、上記の setup_install_server を使用して作成されたミニルートを Boot.orig に移し、Solaris Installer CD 上のミニルートに置き換えます。

```
example# cd /cdrom/cdrom0/s0
example# ./modify_install_server -p /export/install /cdrom/cdrom0/s1
```

例 8 modify_install_server の使用

次の modify_install_server コマンドは、上記の setup_install_server を使用して作成されたミニルートを Solaris Installer CD 上のミニルートに置き換えます。

```
example# cd /cdrom/cdrom0/s0
example# ./modify_install_server /export/install /cdrom/cdrom0/s1
```

例 9 rm_install_client の使用

次の rm_install_client コマンドは、Solaris 8 上のネットワークインストール用クライアントを削除します。

```
example# cd /export/install/Solaris_8/Tools
example# ./rm_install_client holmes
example# ./rm_install_client watson
```

例 10 setup_install_server の使用

次の setup_install_server コマンドは、マウント済みの Solaris CD を Solaris 8 のローカルディスク上にある /export/install というディレクトリにコピーします。

```
example# cd /cdrom/cdrom0/s0/Solaris_8/Tools
example# ./setup_install_server /export/install
```


例 10 setup_install_server の使用 (続き)**例 11** setup_install_server の使用

次の setup_install_server コマンドは、マウント済みの Solaris CD のブートソフトウェアを、Solaris 8 上のサブネット用ブートサーバーになるシステムの /boot_dir というディレクトリにコピーします。

```
example# cd /cdrom/cdrom0/s0/Solaris_8/Tools
example# ./setup_install_server -b /boot_dir
```

例 12 setup_install_server の使用

setup_install_server はデフォルトで、マウント済みの Solaris 配布ディスク上の Solaris ../Tools/Boot にあるインストールブートディレクトリを検索します。

以前に ./setup_install_server -b /boot_dir コマンドを使用して、ネットワークブートサーバー上に作成したディレクトリと別のディレクトリが必要な場合は、-t オプションを使用できます。

```
example# cd /cdrom/cdrom0/s0/Solaris_9/Tools
example# ./setup_install_server -t /boot_dir /export/install
```

例 13 setup_install_server と WANboot オプションの使用

次の setup_install_server コマンドは WANboot ミニルートファイルシステムのイメージを作成し、それを /wanboot_dir/miniroot に保存します。

```
example# cd /cdrom/cdrom0/s0/Solaris_9/Tools
example# ./setup_install_server -w /wanboot_dir /export/install
```

例 14 x86: ネットワークインストール時に使用するシリアルコンソールの指定 (CD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するシリアルコンソールを指定する方法を示します。この例では、インストールクライアントを次の方法で設定します。

- -d オプションは、クライアントが DHCP を使用してインストールパラメータを設定することを示します。
- -e オプションは、Ethernet アドレスが 00:07:e9:04:4a:bf であるクライアントでのみインストールが行われることを示します。
- 1 つめと 2 つめの -b オプションは、インストールプログラムが入力デバイスおよび出力デバイスとしてシリアルポート ttya を使用することを示します。

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
i86pc
```

-b オプションで使用できるブートプロパティの変数とその値について詳細は、eeprom(1M) を参照してください。

install_scripts(1M)

例 15 ネットワークインストール時に使用するブートデバイスの指定 (CD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するブートデバイスを指定する方法を示します。インストールクライアントを設定するときにブートデバイスを指定する場合は、インストール時にデバイス構成補助によって出されるこの情報は表示されません。

この例では、次の方法でインストールクライアントを設定します。

- `-a` オプションは、クライアントが DHCP を使用してインストールパラメータを設定することを示します。
- `-e` オプションは、Ethernet アドレスが `00:07:e9:04:4a:bf` であるクライアントでのみインストールが行われることを示します。
- 1 つめと 2 つめの `-b` オプションは、インストールプログラムが入力デバイスおよび出力デバイスとしてシリアルポート `ttya` を使用することを示します。
- 3 つめの `-b` オプションは、インストールプログラムがインストール時に特定のブートデバイスを使用することを示します。
- ブートデバイスのパスは、使用しているハードウェアによって異なります。
- `i86pc` というプラットフォーム名は、クライアントが x86 ベースのシステムであることを示します。

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
-b "bootpath=/pci@0,0/pci108e,16a8@8" i86pc
```

`-b` オプションで使用できるブートプロパティの変数とその値については、[eeprom\(1M\)](#) を参照してください。

終了ステータス 次の終了ステータスが返されます。

```
0      正常終了
1      エラーが発生した
```

属性 次の属性については、[attributes\(5\)](#) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	Solaris CD

関連項目 [uname\(1\)](#), [bootparams\(4\)](#), [attributes\(5\)](#)

『Solaris 9 インストールガイド』

kdmconfig(1M)

名前	kdmconfig – OpenWindows および国際化対応のキーボード、ディスプレイ、およびマウスオプションの設定または設定解除
形式	kdmconfig kdmconfig [-fv] [-s <i>hostname</i>] -c -t -u -d <i>filename</i>
機能説明	<p>kdmconfig プログラムは、Solaris ソフトウェアが稼動する x86 システム上で、クライアントマシンに関連するキーボード、ディスプレイ、およびマウス情報を、<code>/etc/openwin/server/etc/OWconfig</code> ファイルに設定または設定解除します。また、kdmconfig は、サーバマシン上の <code>bootparams(4)</code> データベースの <code>display</code>、<code>pointer</code>、および <code>keyboard</code> エントリ、または <code>sysidcfg(4)</code> ファイルの <code>monitor</code>、<code>keyboard</code>、<code>display</code>、および <code>pointer</code> キーワードを設定する場合にも使用できます。kdmconfig を実行できるのは、スーパーユーザーだけです。デバイス選択が完了すると、kdmconfig はユーザーに設定をテストするように要求します。このテストはウィンドウシステムを実行することによって行います。</p> <p>kdmconfig プログラムは通常、インストール時および再起動時に実行されますが、システムのインストール後にコマンド行から実行することもできます。最初のインストール時または再構成による再起動時のクライアント設定のとき、<code>sysidconfig(1M)</code> プログラムが <code>-c</code> オプションを指定して kdmconfig を呼び出します。また、ユーザーが <code>sys-unconfig(1M)</code> プログラムを実行すると、<code>-u</code> オプションを指定した kdmconfig が実行されます。同様に、コマンド行から kdmconfig を実行する場合も、<code>-u</code> オプションを使用することで、既存の OpenWindows 構成を設定解除できます。また、<code>-cf</code> オプションを指定して kdmconfig を再実行すると、新しい OpenWindows 構成を作成できます。既存の構成を編集する場合は、オプションを指定しないで、コマンド行から kdmconfig を実行します。再起動のたびに、自動構成機能を確認するために、<code>-t</code> (テストモード) オプションを指定した kdmconfig がシステムによって呼び出され、現在の構成と <code>OWconfig</code> ファイルに記録されている構成間に矛盾がないかどうか調べられます。</p>
オプション	<p>次のオプションを指定できます。</p> <p><code>-c</code> 構成モードでプログラムを実行します。このモードは、<code>OWconfig</code> ファイルを作成または更新する場合に使用します。このオプションを指定して呼び出された kdmconfig はまず、<code>bootparams(4)</code> データベース内の関連する構成情報を探します。また、<code>-s</code> オプションが同時に使用されていないかぎり、デバイスプロンプトから戻された情報も考慮に入れます。クライアントが利用できる <code>bootparams(4)</code> データベースは、サーバマシンが <code>bootparamd(1M)</code> デーモンを実行している、クライアントと同じサブネットにあるサーバ上のすべての <code>/etc/bootparams</code> ファイルです。<code>sysidconfig(1M)</code> によって呼び出される kdmconfig には、<code>-c</code> オプションが指定されます。</p> <p><code>-d filename</code> <code>sysidcfg(4)</code> ファイルを設定します。このオプションを指定した場合、<code>-c</code> オプションと同じ画面が表示されますが、指定した情報は <code>sysidcfg(4)</code> キーワード (<code>monitor</code>、<code>keyboard</code>、<code>display</code>、および <code>pointer</code>) として保存されます。そのため、<code>sysidcfg(4)</code> ファイ</p>

kdmconfig(1M)

	<p>ルを使用して、システムデバイス情報を事前に設定しておく、インストール時の kdmconfig を省略できます。</p> <p><i>filename</i> は作成される sysidcfg(4) ファイルです。パスを指定しない場合、kdmconfig が実行されるディレクトリに作成されず。指定したディレクトリに <i>filename</i> がすでにある場合、既存ファイルにキーワードが付加されます。</p>
-f	<p>強制的に画面モードにします。このオプションを指定して呼び出すと、ネットワークプロンプトは実行されません。クライアントの構成環境をデバッグする場合に便利です。-s オプションは -f オプションを使用することを意味し、サーバーの設定時にネットワークプロンプトが省略されます。</p>
-s <i>hostname</i>	<p>当該マシン上で、指定されたクライアント用の bootparams(4) データベースを設定します。このオプションを使用すると、クライアント側で実行した場合と同じ画面が表示されますが、情報は /etc/bootparams ファイルに書き込まれます。また、-s オプションは -f オプションを使用することを意味します。つまり、このオプションを指定すると、プログラムは必ずユーザーに画面を表示します。このオプションは nsswitch.conf(4) ファイルを再構成し、ローカルサーバー上で bootparams(4) データベースを探します。このオプションを使用できるのは、スーパーユーザーだけです。</p>
-t	<p>テストモードでプログラムを実行します。このモードでは、kdmconfig はデバイスプロンプト情報を使用して、キーボード、ディスプレイ、およびマウスに関する、最新かつ完全な情報が OWconfig ファイルに含まれているかどうかを調べます。情報が正確な場合、kdmconfig はそのまま終了します。情報に問題がある場合、kdmconfig はスーパーユーザーパスワードを要求し、(オプションを指定しないで実行された場合と同様に) 通常の編集セッションに進みます。</p>
-u	<p>システムの設定を解除し、「初期状態」に戻します。この状態では、/etc/openwin/server/etc/OWconfig ファイルからデバイス構成エントリが削除されているため、デフォルトのキーボード、マウス、およびディスプレイが選択されます。これはディスプレイサーバーにとって、不適切な構成である可能性があります。</p>
-v	<p>詳細表示モードを有効にします。kdmconfig は通常、どのような出力も生成しません。このオプションは kdmconfig が実行したさまざまなアクションを標準エラー出力に記録するので、デバッグ時に役立ちます。</p>
オプションを指定しない場合	<p>オプションを指定しないで実行すると、現在の構成を編集する目的で kdmconfig を使用できます。kdmconfig は OWconfig ファイルの情報、bootparams(4) ファイルから得た情報、およびデバイスプロンプトから得た情報を使用します。それ以外は、-c オプションを使用した場合と同様です。</p>

kdmconfig(1M)

- ファイル /etc/openwin/server/etc/OWconfig
OpenWindows の構成ファイル
- /etc/bootparams
ディスクレスクライアントが起動時に使用するクライアントリスト
- /etc/nsswitch.conf
ネームサービスの構成リスト
- 「x86」のみ /dev/openprom インストールされているデバイスおよび属性
- 属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
アーキテクチャ	x86
使用条件	SUNWos86r

- 関連項目 bootparamd(1M), sys-unconfig(1M), sysidconfig(1M), bootparams(4), nsswitch.conf(4), sysidcfg(4), attributes(5)

lpmove(1M)

名前	lpmove – 印刷要求の移動
形式	lpmove [<i>request-ID</i>] <i>destination</i> lpmove <i>destination1 destination2</i>
機能説明	<p>lpmove コマンドは、lp(1) または lpr(1B) によって待ち行列に入れられた印刷要求を宛先間で移動します。</p> <p>第 1 の形式では、指定の印刷要求 (<i>request-ID</i>) を、指定の宛先 (<i>destination</i>) に移します。</p> <p>第 2 の形式では、ある宛先 (<i>destination1</i>) から他の宛先 (<i>destination2</i>) にすべての印刷要求を移します。この形式では lpmove は、新たに発生する <i>destination1</i> 宛の印刷要求も受け付けなくなります。</p> <p>lpmove が個々の要求または待ち行列全体を移動できるのは、ローカルプリンタ間またはリモートプリンタ間のどちらか一方であり、ローカルプリンタとリモートプリンタ間では移動できません。また、移動できる要求はまだサーバーに転送されていない要求だけです。</p> <p>要求を移動する際、lpmove は、印刷要求の移動先の受け入れ状態をチェックしません (accept(1M) を参照)。なお、オプション (たとえば、内容タイプ、特定の用紙の要求) つきの要求に関しては、新たな宛先がそのオプションを扱えなければ、lpmove は移動を行いません。</p>
オペランド	<p>以下のオペランドを指定できます。</p> <p><i>request-ID</i> 移動対象になる特定の印刷要求。lpstat が示す印刷要求に対応する識別子を <i>request-ID</i> に指定します。lpstat(1) のマニュアルページを参照してください。</p> <p><i>destination</i> lpmove が指定された印刷要求を移動するプリンタ名またはプリンタのクラス名 (lpadmin(1M) を参照)。名前、POSIX スタイル名 (<i>server:destination</i>)</p> <p><i>destination1</i> lpmove がすべての印刷要求を移動する移動元の名前。名前、POSIX スタイル名 (<i>server:destination</i>)</p> <p><i>destination2</i> lpmove がすべての印刷要求を移動する移動先の名前。名前、POSIX スタイル名 (<i>server:destination</i>)</p> <p>名前や FNS 名の命名規約については printers.conf(4) を、POSIX については standards(5) を参照してください。</p>
終了ステータス	<p>以下の終了ステータスが返されます。</p> <p>0 正常終了</p> <p>non-zero エラーが発生した</p>
ファイル	/var/spool/print/* LP 印刷待ち行列
属性	次の属性については attributes(5) のマニュアルページを参照してください。

lpmove(1M)

属性タイプ	属性値
使用条件	SUNWpcu

関連項目

lp(1), lpr(1B), lpstat(1), accept(1M), lpadmin(1M), lpsched(1M),
printers.conf(4), attributes(5), standards(5)

lpsched(1M)

名前	lpsched - LP 印刷サービスの起動				
形式	lpsched [-f <i>num_filters</i>] [-n <i>num_notifiers</i>] [-p <i>fd_limit</i>] [-r <i>reserved_fds</i>]				
機能説明	<p>lpsched コマンドは LP 印刷サービスを起動または再起動します。</p> <p>lpshut コマンドは LP 印刷サービスを停止します。lpsched でプリンタを再起動すれば、lpshut で印刷を中止した要求に関しては (始めから) 再印刷できます (lpshut(1M) を参照)。</p>				
オプション	<p>以下のオプションを指定できます。</p> <p>-f <i>num_filters</i> 印刷サーバー上で実行できる、並行するスローフィルタ数を指定します。何も指定しない場合は、デフォルトとして 1 が使用されます。サーバーの構成によっては値を 1 にすると、サーバーにジョブの待ち行列が存在していても、プリンタがアイドル状態のままになります。</p> <p>-n <i>num_notifiers</i> 印刷サーバー上で実行できる、並行する通知プロセス数を指定します。何も指定しない場合は、デフォルトとして 1 が使用されます。</p> <p>-p <i>fd_limit</i> lpsched プロセスで使用するファイル記述子のリソース制限値を指定します。何も指定しない場合は、デフォルトとして 4096 が使用されます。非常に大型でアクティブな印刷サーバー上では、この値を増やす必要はありません。</p> <p>-r <i>reserved_fds</i> 大量ロード下で、スケジューラが内部通信に予約するファイル記述子の数を指定します。何も指定しない場合は、デフォルトとして 2 が使用されます。高速ロード下での問題を障害追跡する場合に、その解析を指示しない限りは、この値を修正する必要はありません。</p>				
終了ステータス	<p>以下の終了ステータスが返されます。</p> <p>0 正常終了</p> <p>0 以外 エラーが発生した</p>				
ファイル	/var/spool/lp/*LP 印刷待ち行列				
属性	次の属性については attributes(5) のマニュアルページを参照してください。				
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">属性タイプ</th> <th style="text-align: center;">属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWpsu</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWpsu
属性タイプ	属性値				
使用条件	SUNWpsu				
関連項目	lp(1), lpstat(1), lpadmin(1M), lpmove(1M), lpshut(1M), attributes(5)				

『Solaris のシステム管理 (基本編)』

lpshut(1M)

名前 lpshut - LP 印刷サービスの停止

形式 **lpshut**

機能説明 lpshut コマンドは LP 印刷サービスを停止します。

lpshut が実行されると、プリンタはただちに印刷を中止します。プリンタを起動または再起動するには、lpsched(1M) を使用します。

終了ステータス 以下の終了ステータスが返されます。

0 正常終了

0 以外 エラーが発生した

ファイル /var/spool/lp/* LP 印刷待ち行列

属性 次の属性については attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWpsu

関連項目 lp(1), lpstat(1), lpadmin(1M), lpmove(1M), lpsched(1M), attributes(5)

『Solaris のシステム管理 (基本編)』

名前	lu – Live Upgrade 機能を使用するための FMLI ベースのインタフェース
形式	/usr/sbin/lu
機能説明	<p>lu プログラムは、Solaris オペレーティング環境の Live Upgrade 機能を提供するコマンド群の 1 つです。Live Upgrade 機能の説明については、live_upgrade(5) のマニュアルページを参照してください。</p> <p>lu プログラムは、FMLI (Forms and Menu Language Interpreter) ベースのユーザーインタフェースです (FMLI の説明は、fmli(1) のマニュアルページを参照してください)。lu プログラムを使用すると、ブート環境 (BE) の作成やアップグレード、また、BE に関する管理タスクを行うことができます。ただし、lu プログラムのできるのは、Live Upgrade コマンド行ユーティリティの一部の機能です。</p> <p>lu コマンドを使用する場合は、下記のことを知っておく必要があります。</p> <ul style="list-style-type: none"> ■ lu は推奨されないインタフェースです。将来はほかのコマンドに置き換えられるので、重要な機能については使用しないでください。 ■ 新しいすべての Live Upgrade 機能は、現在、Live Upgrade コマンド行ユーティリティに実装されています。lu コマンドに新しい機能が追加される予定はありません。 ■ lu コマンドは国際化されていません。将来のリリースでも国際化される予定はありません。 <p>lu は学習または実験のためだけに使用してください。製品用に使用したり、または Live Upgrade の全機能を使用したりする場合は、Live Upgrade コマンド行ユーティリティを使用してください。</p> <p>lu コマンドを実行するには、root 権限が必要です。</p> <p>lu コマンドには引数はありません。lu を呼び出すと、次のオプションが表示されます。</p> <p>Activate ブート環境をアクティブにします。このオプションを使用すると、システムは次回のリブート時に、指定した BE から起動されません。このオプションは、luactivate(1M) コマンド行ユーティリティと同等です。</p> <p>Cancel コピージョブを取り消します。Live Upgrade を使用すると、コピーやアップグレード、フラッシュ機能 (下記の説明を参照) をスケジュールし、後で実行することができます。取り消し機能により、スケジュールしたジョブを取り消すことができます。このオプションは、lucancel(1M) コマンド行ユーティリティと同等です。</p> <p>Compare BE の内容を比較します。2 つの BE の詳細な比較情報が得られます。このオプションは、lucompare(1M) コマンド行ユーティリティと同等です。</p> <p>Copy コピーを開始またはスケジュールします。この機能は、ある BE の内容を他の BE にコピーします。このオプションは、lumake(1M)</p>

lu(1M)

	コマンド行ユーティリティと同等です。Live Upgrade 操作は、同時に 1 つしかスケジュールできません。
Create	ブート環境を作成します。このオプションは、lucreate(1M) コマンド行ユーティリティの一部の機能を実行します。
Current	現在のブート環境の名前を表示します。このオプションは、lucurr(1M) コマンド行ユーティリティと同等です。
Delete	ブート環境を削除します。このオプションは、ludelete(1M) コマンド行ユーティリティと同等です。
List	ブート環境のファイルシステムを一覧表示します。このオプションは、lufslist(1M) コマンド行ユーティリティと同等です。
Rename	ブート環境の名前を変更します。このオプションは、lurename(1M) コマンド行ユーティリティと同等です。
Status	すべてのブート環境の状態を一覧表示します。このオプションは、lustatus(1M) コマンド行ユーティリティと同等です。
Upgrade	ブート環境をアップグレードします。または、非アクティブな BE 上の OS をアップグレードします。このオプションを使用すると、新しいオペレーティングシステムにアップグレードしたり、指定した BE に新しいパッケージやパッチをインストールすることができます。このオプションは、luupgrade(1M) コマンド行ユーティリティの一部の機能を実行します。CD を 2 枚以上使用してアップグレードを行う場合は、-i オプションを使用して luupgrade を実行する必要があります。
Flash	ブート環境をフラッシュします。このオプションを使用すると、フラッシュアーカイブから BE 上にオペレーティングシステムをインストールできます。luupgrade(1M) でも同じ操作が行えます。
Help	ヘルプ情報を表示します。この他にも、多数のオプション用のヘルプ画面を利用できます。
Exit	lu を終了します。

ナビゲーション

lu の画面を切り替えるには、矢印キーとファンクションキー (通常は、Sun デスクトップシステムのキーボード上の F2 から F9 キー) を使用します。使用できるキーの機能が lu 画面の下部に表示されます。Ctrl-F と数字キーを組み合わせると、ファンクションキーの代わりに使用できます。たとえば、F2 キーの代わりに、Ctrl-F と数字キー 2 を押します。

特定のオプション画面で Esc キーを押すと、そのオプションのヘルプ画面が表示されます。

表示に関する注意 tip 回線などを介してリモートから FMLI インタフェースを使用する場合は、TERM 環境変数を VT220 に設定しなければならない場合があります。CDE 環境で FMLI インタフェースを使用する場合は、TERM 変数の値として、xterm ではなく dtterm を使用します。

lu コマンドは、シングルバイト環境でのみ使用できます。

共通の機能 上記のほとんどのオプションでは、次の機能が利用できます。これらの機能は、画面下部に表示されるファンクションキーを使って実行できます。

Choice 入力するフィールドがあるときに使用できます。Choice ファンクションキーを押すと、ポップアップ画面に選択肢が表示されます。たとえば、BE のコピーやアップグレードを伴うオプションでは、使用可能な BE のリストが表示されるので、矢印キーとファンクションキーを使ってポップアップから必要な BE を選択します。この機能には、無効な値を選択することがないという利点があります。たとえば、BE のコピーやアップグレードを伴うオプションで、コピーやアップグレード操作に利用できない BE を選択することはありません。なぜなら、アップグレード中の BE は、選択できないようになっているからです。

Cancel 操作を取り消します。

Save 操作を続けます。

その他の機能 上記の「Create」オプションでは、次の機能を使用できます。

Split ファイルシステムを分割します。たとえば、/ ファイルシステムを /、/usr、および /var に分割できます。ファイルシステムを分割する場合には、分割した個々のファイルシステムをマウントするディスクスライスが必要です。使用できるディスクスライスがない場合、lu は format(1M) ユーティリティを呼び出すので、partition オプションを使って新しいディスクスライスを作成します。

Merge 1 つまたは複数のファイルシステムを、それぞれの親ファイルシステムと結合します。たとえば、/、/usr、および /var という別々のファイルシステムが存在するソース BE をターゲット BE 上で / の下にこれらのファイルシステムをマージできます。

ファイル /etc/lutab システム上にある BE のリスト

属性 次の属性の説明については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWluu

lu(1M)

関連項目 luactivate(1M), lucancel(1M), lucompare(1M), lucreate(1M), lucurr(1M),
ludelete(1M), ludesc(1M), lufslist(1M), lumake(1M), lumount(1M),
lurename(1M), lustatus(1M), luupgrade(1M), lutab(4), attributes(5),
live_upgrade(5)

Solaris のインストールマニュアル

警告 lu コマンドは推奨されないインタフェースです。機能説明 を参照してください。

名前	luactivate – ブート環境のアクティブ化
形式	/usr/sbin/luactivate [-l <i>error_log</i>] [-o <i>outfile</i>] [-s] [<i>BE_name</i>] [-X]
機能説明	<p>luactivate コマンドは、Solaris オペレーティング環境の Live Upgrade 機能を提供するコマンド群の 1 つです。Live Upgrade 機能の説明については、live_upgrade(5) のマニュアルページを参照してください。</p> <p>引数を指定せずに luactivate コマンドを実行すると、システムを次回リブート時にアクティブになるブート環境 (BE) の名前が表示されます。引数として BE を指定すると、その BE がアクティブになります。</p> <p>luactivate は、BE のルートパーティションをブート可能にすることによって、その BE をアクティブにします。x86 マシンでは、luactivate の実行後に、いくつかの手順を実行しなければならないことがあります。その場合、luactivate は必要となる手順を表示します。</p> <p>BE をアクティブにするためには、BE が次の条件を満たしていなければなりません。</p> <ul style="list-style-type: none"> ■ lustatus(1M) で表示される BE の状態が complete でなければならない ■ BE が現在の BE でない場合、その BE 上に lumount(1M) や mount(1M) でマウントされたパーティションがあってはならない ■ アクティブにしたい BE が lucompare(1M) 操作の対象であってはならない <p>luactivate は、指定された BE をアクティブにした後に、次回のリブート時に問題が発生した場合にとるべきフォールバック手順を表示します。それらの手順を記録しておき、必要であればその手順に従います。</p> <p>新たに作成した BE を初めてブートすると、Live Upgrade ソフトウェアは、その BE と最後にアクティブだった BE の同期をとります (アクティブだった BE は、新たに作成した BE のソースであるとは限りません)。ここでの「同期をとる」とは、特定のシステムファイルおよびディレクトリを、最後にアクティブだった BE から、ブートする BE にコピーすることを指します。最初のブート以後は、-s オプションを指定しない限り、同期はとられません (詳細は -s オプションの説明を参照)。</p> <p>同期をとったファイル間に衝突が検出された場合、luactivate は警告を發しますが、衝突のあったファイルの同期をとることはしません。衝突があった場合でも、アクティブ化が成功することがあります。衝突は、ある BE を別のオペレーティングシステム用にアップグレードしたり、BE 上のシステムファイル (/etc/passwd など) を変更した場合に発生する可能性があります。</p> <p>luactivate コマンドには root 権限が必要です。</p>
オプション	<p>luactivate コマンドには、次のオプションを指定できます。</p> <p>-l <i>error_log</i> エラーメッセージと状態メッセージを、現在の環境での書き込み先だけでなく、<i>error_log</i> にも書き込みます。</p> <p>-o <i>outfile</i> すべてのコマンド出力を、現在の環境での書き込み先だけでなく、<i>outfile</i> にも書き込みます。</p>

luactivate(1M)

-s	<p>指定した BE に対する次のブートが、その BE に対する最初のブートでない場合でも、同期をとります(「機能説明」の項を参照)。このオプションは慎重に使用してください。アクティブだった BE で、このコマンドの実行者自身が管理または関知しない変更が行われている可能性があります。</p> <p>-s を使用する場合、アクティブだった BE にインストールされている Solaris より古いバージョンの Solaris でブートするときには、特に注意が必要です。たとえば、アクティブだった BE には Solaris 9 が含まれていて、これを Solaris 2.6 を含む BE でブートする場合を考えてみましょう。-s オプションを使用して強制的に同期をとると、Solaris 2.6 を含む BE のファイルは、Solaris 9 と互換性はあるが、Solaris 2.6 上では動作しない可能性のあるファイルと同期がとられることとなります。</p>				
-X	<p>XML 出力を有効にします。XML の特性は DTD (/usr/share/lib/xml/dtd/lu_cli.dtd.<num>) に定義されています。<num> は、各 DTD ファイルのバージョン番号を示します。</p>				
オペラント	<p><i>BE_name</i> アクティブにする BE の名前を指定します。</p>				
終了ステータス	<p>次の終了ステータス値が返されます。</p> <p>0 正常終了</p> <p>>0 エラーが発生した</p>				
ファイル	<p>/etc/lutab システム上にある BE のリスト</p> <p>/usr/share/lib/xml/dtd/lu_cli.dtd.<num> Live Upgrade の DTD (-X オプションを参照)</p>				
属性	<p>次の属性の説明については、attributes(5) のマニュアルページを参照してください。</p>				
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">属性タイプ</th> <th style="text-align: center;">属性値</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">使用条件</td> <td style="text-align: center;">SUNWluu</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWluu
属性タイプ	属性値				
使用条件	SUNWluu				
関連項目	<p>lu(1M), lucancel(1M), lucompare(1M), lucreate(1M), lucurr(1M), ludelete(1M), ludesc(1M), lufslslist(1M), lumake(1M), lumount(1M), lurename(1M), lustatus(1M), luupgrade(1M), lutab(4), attributes(5), live_upgrade(5)</p>				

名前	lucancel – スケジュールされた Live Upgrade 機能でのコピー/作成操作の取り消し				
形式	<code>/usr/sbin/lucancel [-l <i>error_log</i>] [-o <i>outfile</i>] [-X]</code>				
機能説明	<p>lucancel コマンドは、Solaris オペレーティング環境の Live Upgrade 機能を提供するコマンド群の 1 つです。Live Upgrade 機能の説明については、live_upgrade(5) のマニュアルページを参照してください。</p> <p>lucancel コマンドは、FMLI ベースのインタフェースである lu(1M) を使ってスケジュールされた、ブート環境 (BE) の作成またはアップグレードを取り消します。あるいは、lumake(1M) を使ってスケジュールされた BE の再生成を取り消します。lucancel は、アクティブな (つまり、作成中や再生成中の) ジョブは取り消しません。</p> <p>lucancel コマンドには root 権限が必要です。</p>				
オプション	<p>lucancel コマンドには、次のオプションを指定できます。</p> <p><code>-l <i>error_log</i></code> エラーメッセージと状態メッセージを、現在の環境での書き込み先だけでなく、<i>error_log</i> にも書き込みます。</p> <p><code>-o <i>outfile</i></code> すべてのコマンド出力を、現在の環境での書き込み先だけでなく、<i>outfile</i> にも書き込みます。</p> <p><code>-X</code> XML 出力を有効にします。XML の特性は DTD (<code>/usr/share/lib/xml/dtd/lu_cli.dtd.<num></code>) に定義されています。<num> は、各 DTD ファイルのバージョン番号を示します。</p>				
終了ステータス	<p>次の終了ステータス値が返されます。</p> <p>0 正常終了</p> <p>>0 エラーが発生した</p>				
ファイル	<code>/etc/lutab</code> システム上にある BE のリスト				
属性	次の属性の説明については、attributes(5) のマニュアルページを参照してください。				
	<table border="1"> <thead> <tr> <th>属性タイプ</th> <th>属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWluu</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWluu
属性タイプ	属性値				
使用条件	SUNWluu				
関連項目	lu(1M), luactivate(1M), lucompare(1M), lucreate(1M), lucurr(1M), ludelete(1M), ludesc(1M), lufslist(1M), lumake(1M), lumount(1M), lurename(1M), lustatus(1M), luupgrade(1M), lutab(4), attributes(5), live_upgrade(5)				

lucompare(1M)

名前	lucompare - 2つのブート環境の比較				
形式	<pre>/usr/sbin/lucompare [-i infile -t] [-o outfile] BE_name [-X] /usr/sbin/lucompare [-c file [-o outfile]] [-X]</pre>				
機能説明	<p>lucompare コマンドは、Solaris オペレーティング環境の Live Upgrade 機能を提供するコマンド群の1つです。Live Upgrade 機能の説明については、live_upgrade(5)のマニュアルページを参照してください。</p> <p>lucompare コマンドは、現在のブート環境 (BE) の内容と他の BE の内容を比較します。lucompare に -c オプションを指定すると、ファイルの統計情報が比較されます。これによって、指定した時刻 (BE の作成時刻など) 以降に BE 上のどのファイルが変更されたかを知ることができます。指定する BE は非アクティブで、その状態が complete でなければなりません。BE の状態は、lustatus(1M) コマンドで知ることができます。また、BE は、スケジュールされたコピージョブであってはなりません (これも lustatus(1M) で検査できます)。指定する BE には、lumount(1M) や mount(1M) でマウントされたパーティションがあってはなりません。</p> <p>lucompare は、指定された BE に定義されているファイルシステムごとに、そのすべてのファイルを、現在の BE にある同じパス名を持つファイルと比較します。アクティブな BE にはあるが、指定した BE にはないファイルと、その逆の場合のファイルが報告されます。比較するファイルのリストを指定するオプションも利用できます。</p> <p>現在の BE とターゲット BE の完全な比較を行う代わりに -c オプションを指定すると、指定した BE のファイルと、任意のファイルに記録されているファイルリストが比較されます。BE が作成されたときに、lucreate(1M) は /etc/lu/compare の下に :<BE_name> というファイルを作成します。-c オプションを指定すると、指定した BE のファイルと /etc/lu/compare にあるこのスナップショットを比較できます。あるいは、-o オプションですでに作成してあるファイルと比較することもできます。BE と、/etc/lu/compare にあるそれ自身のスナップショットを比較することにより、BE の作成時以降に変更されたファイルを判別できます。</p> <p>デフォルトでは、lucompare の出力は標準出力に書き込まれます。-c オプションを指定するとき、-o オプションで出力ファイルを指定する必要があります。lucompare の出力には、アクセス権、所有者、グループ、またはチェックサムが異なるファイルのリストと、その相違理由が含まれます。出力形式は次のとおりです。</p> <pre>> active BE < BE_name reason > file_name:owner:group:number_of_links:mode:type:size or major_minor number:checksum < file_name:owner:group:number_of_links:mode:type:size or major_minor number:checksum</pre> <p>上記の各フィールドは、ファイルの stat(2) 構造体から得られたものです。</p> <p>type フィールドは次のいずれかです。</p> <table><tr><td>SYMLINK</td><td>シンボリックリンク</td></tr><tr><td>FIFO</td><td>FIFO ファイル</td></tr></table>	SYMLINK	シンボリックリンク	FIFO	FIFO ファイル
SYMLINK	シンボリックリンク				
FIFO	FIFO ファイル				

CHRSPC	文字型特殊ファイル
BLKSPC	ブロック型特殊ファイル
DIR	ディレクトリ
REGFIL	通常ファイル
UNKNOW	不明なファイルタイプ

lucompare は、指定した BE 上のファイルとアクティブな BE 上の対応するファイルが上記のすべてのフィールドで一致する場合にのみ、チェックサムを計算します。そして、チェックサムが一致しない場合、lucompare は比較したそれぞれのファイルのエントリにそのチェックサムを追加します。

lucompare コマンドには root 権限が必要です。

オプション lucompare コマンドには、次のオプションを指定できます。

-C <i>file</i>	BE のファイル統計情報と、 <i>file</i> に記録されている統計情報を比較します。 <i>file</i> は、BE の作成時に作成されたスナップショット (/etc/lu/compare/:<BE_name>) か、-o オプションですでに作成されているファイルです。このオプションを使用するときには、-o オプションを指定する必要があります。
-i <i>infile</i>	<i>infile</i> にリストされているファイルを比較します。これらのファイルは、絶対パス名で指定しなければなりません。ファイル内のエントリがディレクトリの場合には、比較はそのディレクトリについて再帰的に行なわれます。-t オプションを同時に指定することはできません。
-o <i>outfile</i>	相違点を <i>outfile</i> に出力します。-c オプションを使用する場合は、このオプションも指定する必要があります。
-t	非バイナリファイルだけを比較します。この処理は、ツリー上の各ファイルに対して file(1) コマンドを実行し、テキストファイルだけを比較することによって行なわれます。-i オプションを同時に指定することはできません。
-X	XML 出力を有効にします。XML の特性は DTD (/usr/share/lib/xml/dtd/lu_cli.dtd.<num>) に定義されています。<num> は、各 DTD ファイルのバージョン番号を示します。

オペランド *BE_name* アクティブな BE の比較対象となる BE の名前を指定します。他の Live Upgrade 操作に使われている BE や、lumount(1M) または mount(1M) でパーティションがすでにマウントされている BE を指定することはできません。

lucompare(1M)

使用例 例 1 例 1 BE 作成時と現時点の相違点を検査する

次のコマンドは、BE s8u5 の作成時と現時点における相違点を一覧表示します。

```
# lucompare -C /etc/lu/compare/:s8u5 -o /var/tmp/compare.out s8u5
```

/etc/lu/compare/:s8u5 は、BE の作成時に lucreate が作成したファイルです。相違点の一覧は、/var/tmp/compare.out に書き込まれます。

終了ステータス 次の終了ステータスが返されます。

0 正常終了
>0 エラーが発生した

ファイル /etc/lutab
 システム上にある BE のリスト

/usr/share/lib/xml/dtd/lu_cli.dtd.<num>
 Live Upgrade の DTD (-x オプションを参照)

属性 次の属性の説明については、attributes(5) のマニュアルページを参照してください。

属性タイプ	使用条件
属性値	SUNWluu

関連項目 lu(1M), luactivate(1M), lucancel(1M), lucreate(1M), lucurr(1M), ludelete(1M), ludesc(1M), lufslis(1M), lumake(1M), lumount(1M), lurename(1M), lustatus(1M), luupgrade(1M), lutab(4), attributes(5), live_upgrade(5)

注意事項 lucompare コマンドが BE 間の相違点を修正することはありません。

名前	lucreate – 新しいブート環境の作成
形式	<pre> /usr/sbin/lucreate [-A BE_description] [-c BE_name] [-C (boot_device -)] -n BE_name [-f exclude_list_file] [-I] [-l error_log] [-o outfile] [-s (- source_BE_name)] [[-M slice_list_file [-M...]] [-m mount_point:device [, volume]:fs_options [-m...]]] [-x exclude [-x...]] [-X] [-y include [-y...]] [-Y include_list_file] [-z filter_list] </pre>
機能説明	<p>lucreate コマンドは、Solaris オペレーティング環境の Live Upgrade 機能を提供するコマンド群の 1 つです。Live Upgrade 機能の説明と関連用語については、live_upgrade(5) のマニュアルページを参照してください。</p> <p>lucreate コマンドではコマンド行オプションを指定することにより、次の操作が行えます。</p> <ul style="list-style-type: none"> ■ 現在のブート環境 (BE) から新しい BE を作成する ■ 現在の BE 以外の BE から新しい BE を作成する ■ BE 上のファイルシステムを新しい BE 上で分割または結合する。たとえば、/var と /opt を /の下に結合したり、これらのディレクトリを分割して異なるディスクスライスの下にマウントしたりすることができます。 ■ BE 上に、ファイルシステムを作成するが、ファイルは作成しないでおく。 <p>これらの操作は、lucreate コマンド行オプションだけを使って行うこともできますが、-m および -M オプション (後述の説明を参照) を省略して FMLI ベースのインタフェースを自動的に呼び出し、curses ベースの画面を使って Live Upgrade 管理を行うこともできます。FMLI ベースのインタフェースは、lucreate でサポートされる Live Upgrade 機能をすべてサポートするわけではないことに注意してください。また、Sun は FMLI ベースのインタフェースの開発を継続していくことを保証しているわけではありません。</p> <p>BE の作成には、BE のすべてのマウントポイントに対してディスクまたはデバイススライスの選択も含まれます。スライスには物理ディスク、または Solaris ポリユーママネージャのポリユームのような論理デバイスを選択できます。BE のマウントポイントは、FMLI ベースの構成画面で SPLIT または MERGE 機能を使って変更することもできます。</p> <p>BE が適切に作成されると、lustatus(1M) を使って BE の状態を表示したり、lufslist(1M) を使って BE のファイルシステムを表示したりすることができます。さらに、luupgrade(1M) を使ってその BE の OS をアップグレードしたり、luactivate(1M) を使って BE をアクティブにしたりすることができます。BE をアクティブにすると、システムは次のブート時にその BE からブートします。</p> <p>lucreate コマンドは、OS を含むファイルシステム (/、./usr、/var、/opt など) と、OS を含まないファイルシステム (/export、/home などのユーザー定義ファイルシステム) を区別します。OS を含むファイルシステムを、ソース BE と新しい BE の間で共有することはできません。これらのファイルシステムは、常にソース BE からターゲット BE にコピーされます。これに対して、ユーザー定義のファイルシステムはデフォルトで共有されます。Live Upgrade では、OS を含むファイルシステムを</p>

lucreate(1M)

共有不能 (または重要な) ファイルシステム、その他のファイルシステムを共有可能ファイルシステムと呼びます。ソース BE の `vfstab` にリストされている共有不能ファイルシステムは、新しい BE にコピーされます。共有可能ファイルシステムは、宛先スライスを指定した場合にのみコピーされます。宛先スライスを指定しない場合は、コピーされません。

`lucreate` コマンドは、Solaris ボリュームマネージャ機能のいくつかのサブセットをサポートします。たとえば、`lucreate` コマンドに `-m` オプションを使用すると、下記のことが可能です。

- ミラーを作成する
- ミラーから既存の Solaris ボリュームマネージャ連結を切り離す。同様に、ミラーに既存の Solaris ボリュームマネージャの連結を接続する。これらは、Solaris ボリュームマネージャまたは `lucreate` を使って作成されたミラーである。
- 単一スライス連結を作成し、それに単一のディスクスライスを接続する。
- 単一スライス連結から単一のディスクスライスを切り離す。
- 単一のミラーに複数の単一スライス連結を接続する。`lucreate` では、Solaris ボリュームマネージャで許可されている個数の連結を接続できる。

`lucreate` では、単一の連結に複数のディスクスライスまたは記憶装置を接続することはできません。同様に、単一の連結から複数のスライスまたは装置を切り離すことはできません。

ブート環境に Solaris ボリュームマネージャのボリュームを使用する場合は、ボリュームの操作には Solaris ボリュームマネージャのコマンドよりも `lucreate` コマンドが推奨されます。Solaris ボリュームマネージャソフトウェアはブート環境を認識しません。一方、`lucreate` コマンドには、たとえば Solaris ボリュームマネージャのボリュームを上書きしたり削除したりしてブート環境を破壊することを避ける検査機能が含まれています。

すでに Solaris ボリュームマネージャソフトウェアを使用して複雑な Solaris ボリュームマネージャのボリューム (たとえば、RAID-5 ボリューム) を作成した場合は、`Live Upgrade` がその使用をサポートします。ただし、これらの複雑なオブジェクトを作成したり操作したりするためには、Solaris ボリュームマネージャソフトウェアを使用する必要があります。上で述べたとおり、`lucreate` コマンドではなく Solaris ボリュームマネージャを使用すると、ブート環境を破壊する危険が伴います。それでも Solaris ボリュームマネージャソフトウェアを使用するという場合は、`lufslist(1M)` を使用して、ブート環境に使用されているデバイスを特定してください。

以下に説明するように、`-s` オプションを特に使用する場合を除き、新しい BE の作成にはソース BE が必要です。デフォルトでは、ソース BE は現在の BE ですが、`-s` オプションを使えば、現在の BE 以外の BE を指定できます。

新しく BE を作成する場合は、`lucreate` を使用するとソース BE からファイルを除外したり、あるいは含めたりすることができます。この動作は、以下で説明する `-f`、`-x`、`-y`、`-Y`、および `-z` オプションを使用して行います。これらのオプションの組み合わせについては、以下の「オプション」を参照してください。

デフォルトでは、ソース BE 上のすべてのスワップパーティションは、ソース BE とターゲット BE 間で共有されます。-m オプションを使えば (下記を参照)、ソース BE 上のスワップパーティションの追加セットまたは新しいセットをターゲット BE と共有できます。

lucreate コマンドを使用すると、BE に説明を追加できます。説明の指定は任意で、書式や長さに制限はありません。たとえば、テキスト文字列でもバイナリデータでもかまいません。BE の作成後は、ludesc(1M) ユーティリティを使用して BE の説明を変更できます。

lucreate コマンドにはスーパーユーザー (root) 権限が必要です。

オプション

lucreate コマンドには、以下のオプションを指定できます。BE 名は、30 文字を超えてはならず、使用できる文字は英数字とその他の ASCII 文字 (UNIX シェルにとって特殊な意味を持つ文字は除く) だけです。これについては、sh(1) の「クォート」セクションを参照してください。また、BE 名に使用できるのは 8 ビットで表現できるシングルバイトの文字だけです。空白文字を含めることはできません。

lucreate コマンド行で -m オプションまたは -M オプション (下記を参照) を省略すると、FMLI ベースのインタフェースが呼び出され、このインタフェースを介して BE 用のディスクスライスまたはデバイススライスを指定できます。

-A *BE_description*

BE に、説明 (*BE_description*) を追加します。*BE_description* には、テキスト文字列および、UNIX コマンド行に入力可能な任意の文字を使用できます。BE の説明に関する詳細は、ludesc(1M) を参照してください。

-c *BE_name*

現在の BE に *BE_name* という名前を割り当てます。このオプションは、省略可能であり、最初の BE を作成するときだけに使用できます。lucreate を初めて実行する際に -c を省略した場合、以下の規則に従ってデフォルト名が決定されます。

1. 物理ブートデバイスを決定できる場合、そのデバイスのベース名に基づいて新しいブート環境が命名されます。たとえば、物理ブートデバイスが /dev/dsk/c0t0d0s0 であった場合、新しいブート環境は「c0t0d0s0」と命名されます。
2. 物理ブートデバイスを決定できない場合、オペレーティングシステム名 (「uname -s」を使用) とオペレーティングシステムのリリースレベル (「uname -r」を使用) を組み合わせて新しいブート環境が命名されます。たとえば、「uname -s」から「SunOS」が返され、「uname -r」から「5.9」が返される場合、「SunOS5.9」という名前が新しいブート環境に割り当てられます。
3. ブートデバイス名、オペレーティングシステム名のいずれも決定できない場合、「current」という名前が新しいブート環境に割り当てられます。

最初のブート環境の作成後に -c オプションを使用した場合、指定した名前が現在のブート環境名と同じである場合にはそのオプションは無視され、異なる場合にはエラーメッセージが表示されて処理が終了します。

lucreate(1M)

-C (*boot_device* | -)

lucreate が、どの物理記憶装置がブートデバイスであるか認識できない場合があります。たとえば、x86 マシン上のソース BE にミラー化されたルートデバイスがある場合、このようなことが発生します。-c は、ソース BE のブートに使用する物理ブートデバイスを指定します。このオプションを指定しないと lucreate は BE のブートに使用された物理デバイスを検出しようとします。ルート (/) ファイルシステムのあるデバイスが物理ディスクでない場合 (例: ルートが Solaris ポリウムマネージャのボリューム上にある場合)、lucreate は、その物理デバイスが妥当と推測し、次のような質問メッセージを出力します。

```
Is the physical device devname the boot device for
the logical device devname?
```

y を入力すると、処理が続けられます。

-C *boot_device* を使用すると、lucreate は物理デバイスを検索せずに、指定されたブートデバイスを使用します。-c オプションに - (ハイフン) を指定すると、lucreate が検出したものをブートデバイスとして処理が続けられます。デバイスが特定できない場合は、デバイス名を入力するよう求めるプロンプトが表示されません。

-c を省略した場合や、-C *boot_device* を指定したが lucreate が指定されたブートデバイスを検出できなかった場合は、エラーメッセージが返されます。

-C - の場合、lucreate は、正しいブートデバイスを検出するか、以降の質問メッセージでブートデバイスを指定するように求めるので、この形式を使用する方が安全です。

-f *exclude_list_file*

exclude_list_file の内容を使用して、新しく作成した BE から特定のファイル (ディレクトリを含む) を除外します。*exclude_list_file* にはファイル名およびディレクトリ名が行単位のリストで記載されています。行の項目がファイル名の場合、そのファイルだけが除外されます。項目がディレクトリ名の場合は、該当するディレクトリとその下にあるすべてのファイル (サブディレクトリを含む) が除外されます。

-I

完全性の検査を無視します。新しい BE を作成する前に、BE から誤って重要なシステムファイルを除外しないように lucreate は完全性検査を実行します。完全性検査を上書きする場合もこのオプションを使用します。このオプションを使用することによる長所は、-I を使用すると BE をより速く作成できるということです。それに対して短所は、BE が期待した通りには動作しない危険があるということです。

-l *error_log*

エラーメッセージとステータスメッセージを、現在の環境での書き込み先だけでなく、*error_log* にも書き込みます。

-m *mount_point:device[,volume]:fs_option*
[-m *mount_point:device: fs_option*] ...

新しい BE の `vfstab(4)` 情報を指定します。-m オプションの引数として指定するファイルシステムは、同じディスク上にあっても、複数のディスクにまたがって存在していてもかまいません。

`mount_point` には、有効なマウントポイントを指定します。- (ハイフン) は、スワップパーティションであることを示します。`device` フィールドには、次のいずれかを指定します。

- ディスクスライスの名前 (書式は `/dev/dsk/cnumtnumdnumsnum`)。
- Solaris ボリュームマネージャのボリュームの名前 (書式は `/dev/md/dsk/dnum`)。
- Veritas ファイルシステムの名前 (書式は `/dev/md/vxfs/dnum`)。
- キーワード `merged`。指定したマウントポイントのファイルシステムがその親とマージされることを示します。
- キーワード `shared`。ソース BE のすべてのスワップパーティションが新しい BE と共有されることを示します。

物理ディスクデバイス名および Solaris ボリュームマネージャのボリューム名は、デバイスを一意に示す最短の名前に短縮できます。たとえば、マシンがデバイス `/dev/dsk/c0t0d0s0` に対して 1 つのディスクコントローラと 1 つのディスクドライブだけを持っている場合、`/dev/dsk/c0t0d0` は省略して `s0` という名前を使用できます。マシンが 1 つのコントローラと複数のディスクを持っている場合は `t0d0s0` という名前を、また、複数のコントローラを持っている場合は `c0t0d0s0` という名前を使用できます。Solaris ボリュームマネージャのボリュームは、その `dnum` という指定から判断できます。たとえば、`/dev/md/dsk/d10` は単純に `d10` で表わすことができます。

-m オプションを指定すると、物理ディスクデバイスを Solaris ボリュームマネージャの単一スライス連結に接続したり、Solaris ボリュームマネージャのボリュームをミラーに接続したりできます。どちらの操作も、後述する `attach` キーワードを使って実行できます。このオプションでは、特定の連結またはミラーを指定することができ、または `lucreate` に自動的に選択させることもできます。特定の連結またはミラーを指定するには、論理デバイスを接続するデバイス名の後にコンマと Solaris ボリュームマネージャの論理デバイス名を付加します。この指定を省略した場合、`lucreate` によって、空いているデバイスのリストから特定の連結またはミラーが選択されます。「使用例」を参照してください。

`fs_option` フィールドには、下記に示すキーワードの 1 つまたは複数指定できます。最初に示す 2 つのキーワードは、ファイルシステムの種類を指定します。ほかのキーワードは、ファイルシステムに対するアクションを指定します。複数のキーワードを指定する場合は、コンマで区切ります。

ufs

ファイルシステムを UFS ボリュームとして作成します。

vxfs

ファイルシステムを Veritas デバイスとして作成します。

preserve

指定した物理記憶装置のファイルシステムの内容を保存します。このキーワードを使用すると、デバイスのファイルシステムとその内容が、指定したマウントポ

イントに適切であると仮定されます。指定したマウントポイントに保存するデバイスとして指定できるのは1つだけです。このキーワードを指定すると、指定した記憶装置に新しいファイルシステムを作成し、ファイルシステムの内容を、ソース BE から指定したデバイスにコピーするというデフォルトの手順を省略できます。preserve、lucreate を使用すると、記憶装置の内容が指定したファイルシステムに適切かどうか検査します。この検査には限界があり、適合性を保証するものではありません。

mirror

指定した記憶装置上にミラーを作成します。指定する記憶装置は、正しく名前をついた (たとえば、/dev/md/dsk/d10、d10 など) ミラーとして働く論理デバイスである必要があります。後続の -m オプションでは、attach を指定して (下記を参照)、少なくとも1つの物理デバイスを新しいミラーに接続する必要があります。

attach

ボリュームに含まれる物理記憶装置を、指定したマウントポイントに関連するミラーまたは単一スライス連結に接続します。attach を使用する場合、ディスクを特定のミラーまたは連結に接続するときは、デバイス名の後ろにコンマ、続けて論理デバイスの名前を付与します。コンマと連結名を省略すると、lucreate はその記憶装置のコンテナボリュームとして空いているミラーまたは単一スライス連結を選択します。「使用例」を参照してください。

lucreate は、1つの物理ドライブを含む連結だけを作成して、そのような連結を最大4つまでミラーに接続できるようにします。

detach

指定したマウントポイントに関連するミラーまたは連結から物理記憶装置の接続を解除します。

ルートには、少なくとも1つのディスクまたはデバイススライスを指定する必要があります。この指定には、-m または -M オプション (下記を参照) を使用するか、FMLI ベースのインタフェースを使用します。新しい BE にファイルシステムを作成する場合は、ファイルシステムごとに -m 引数を指定する必要があります。たとえば、ソース BE にある3つのファイルシステム (たとえば、/、/usr、および /var) を新しい BE 上に個別のファイルシステムとして作成する場合は、-m 引数を3回指定します。-m 引数を1回だけ指定した場合、/、/usr、および /var は、1つのファイルシステムとして新しい BE の / の下にマージされます。

-m オプションを使ってスワップパーティションを指定する場合は、いずれかの BE で現在スワップに使用しているデバイスおよび未使用のデバイスを指定できます。スワップ割り当てについては、次の方法から選択できます。

- ソース BE に関連するすべてのスワップデバイスを新しい BE が使用するように割り当てる場合は、スワップデバイスの指定をすべて省略できます。
- 新しい BE が、指定したスワップデバイスだけを使用してソース BE に関連するスワップデバイスを自動的に共有しないように割り当てる場合は、1つまたは複数のスワップデバイスを指定します。

- 新しい BE が、指定したスワップデバイスを使用してソース BE とスワップデバイスを共有するように割り当てる場合は、構文 `-m -:shared:swap` を使用して 1 つまたは複数のスワップデバイスを指定します。

「使用例」を参照してください。

-M *slice_list*

slice_list ファイルには、`-m` オプションのリストが含まれています。リストに指定する引数の形式は、`-m` オプションに指定する引数の形式と同じです。# 文字で始まる行は、コメント行で無視されます。`-M` オプションは、BE に多数のファイルシステムを指定するときに便利です。`-m` と `-M` オプションは同時に指定できます。たとえば、スワップパーティションを *slice_list* 内に指定し、/ と /usr スライスを `-m` オプションで指定することができます。

`-m` と `-M` オプションには、1 つのマウントポイントに対して複数のスライスを指定できます。lucreate は、存在しないスライスを無視し、使用可能な最初のスライスを選択します。

-n *BE_name*

作成する BE の名前を指定します。*BE_name* は、システムにおいて一意の名前にする必要があります。

-o *outfile*

すべてのコマンド出力を、現在の環境での書き込み先だけでなく、*outfile* にも書き込みます。

-s (- | *BE_name*)

新しい BE を作成する際のソースを指定します。このオプションにより、新しい BE を作成するソースとして現在の BE 以外のものを指定できます。引数としてハイフン (-) を指定すると、新しい BE が作成されますが、ファイルシステム内にファイルは作成されません。この BE 内にファイルを作成するには `lumake(1M)` を使用し、BE 上にフラッシュアーカイブをインストールするには `luupgrade(1M)` を使用する必要があります。このオプションは、特に、フラッシュアーカイブをインストールするときに便利です (`flar(1M)` のマニュアルページを参照)。

-x *exclude*

新しく作成した BE から *exclude* ファイルまたは *exclude* ディレクトリを除外します。*exclude* がディレクトリ名の場合、lucreate はそのディレクトリと、その下にあるすべてのファイル (サブディレクトリを含む) を除外します。

-X

XML 出力を有効にします。XML の特性は DTD (`/usr/share/lib/xml/dtd/lu_cli.dtd.<num>`) に定義されています。`<num>` は、各 DTD ファイルのバージョン番号を示します。

-y *include*

新しく作成した BE にファイル *include* またはディレクトリ *include* を取り込みます。*include* がディレクトリ名の場合、lucreate はそのディレクトリと、その下にあるすべてのファイル (サブディレクトリを含む) を取り込みます。

lucreate(1M)

ファイルの取り込みオプションと除外オプションの組み合わせ

-Y include_list_file
include_list_file の内容を使用して、新しく作成した BE に特定のファイル (ディレクトリを含む) を取り込みます。 *include_list_file* にはファイル名およびディレクトリ名が行単位のリストで記載されています。行の項目がファイル名の場合、そのファイルが取り込まれます。行の項目がディレクトリ名の場合、該当するディレクトリとその下にあるすべてのファイル (サブディレクトリを含む) が取り込まれます。

-z filter_list_file
filter_list_file にはファイル名およびディレクトリ名が行単位の項目リストで記載されています。各項目の先頭に + 記号が付いた場合は新しい BE に取り込む項目を、 - が付いた場合は新しい BE から除外する項目を示します。

lucreate コマンドを使用すると、新しい BE を作成したときに特定のファイルおよびディレクトリを取り込みまたは除外できます。ファイルおよびディレクトリの取り込みは、次のオプションを使用します。

- **-y include** オプション
- **-Y include_list_file** オプション
- **-z filter_list** オプション。リストファイルの項目名の先頭に + を付ける

ファイルおよびディレクトリの除外は、次のオプションを使用します。

- **-x exclude** オプション
- **-f exclude_list_file** オプション
- **-z filter_list** オプション。リストファイルの項目名の先頭に - を付ける

除外される項目の親ディレクトリが、取り込みオプション (たとえば、 **-y include**) で取り込まれる場合は、 **exclude** で指定された特定のファイルまたはディレクトリだけが除外されます。逆に、取り込まれるファイルの親ディレクトリが除外オプションで指定されている場合は、 **include** で指定されたファイルだけが取り込まれます。たとえば、次のように指定します。

```
-x /a -y /a/b
```

/a/b を除いたすべての /a が除外されます。次のように指定すると、

```
-y /a -x /a/b
```

/a/b を除いたすべての /a が取り込まれます。

使用例 lucreate コマンドはさまざまな出力を生成します。以下の例では、説明のために必要な場合を除き、出力例は省略しています。

例 1 新しいブート環境を初めて作成する

次のコマンドシーケンスを使って、ブート環境がまだ作成されていないマシン上に新しい BE を作成します。共有できないすべての (クリティカルな) ファイルシステムは / の下にマウントされます。

```
# lucreate -c first_disk -m /:/dev/dsk/c0t4d0s0:ufs -n second_disk  
(出力略)
```

```
lucreate: Creation of Boot Environment <second_disk> successful.
```

例 1 新しいブート環境を初めて作成する (続き)

次のコマンドは、先のコマンドと同様、BE が作成されたことのないマシン上に新しいブート環境を作成します。ただし、次のコマンドは、先のコマンドと2つの点で異なります。-c オプションは省略され、/usr ファイルシステムは / からは分離して自身のディスクスライス上にマウントされます。

```
# lucreate -m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s1:ufs \
-n second_disk
lucreate: Please wait while your system configuration is determined.
(出力略)
lucreate: Creation of Boot Environment c0t4d0s0 successful.
```

-c オプションが指定されなかった場合、lucreate は、ルートデバイスのベース名である「c0t4d0s0」を、新しいブート環境に割り当てます。

次に -c オプションを指定して同じコマンドを実行します。

```
# lucreate -c first_disk -m /:/dev/dsk/c0t4d0s0:ufs \
-m /usr:/dev/dsk/c0t4d0s1:ufs -n second_disk
(出力略)
lucreate: Creation of Boot Environment <second_disk> successful.
```

BE を作成した後、luupgrade(1M) を使って新しい BE 上で OS をアップグレードし、luactivate(1M) を使ってその BE をアクティブにします。マシンは次回リブートするときに、この BE からブートされます。first_disk のスワップパーティションとすべての共有可能ファイルシステムは、second_disk から使用 (共有) できます。

```
# luupgrade -u -n second_disk \
-s /net/installmachine/export/solarisX/OS_image
(出力略)
luupgrade: Upgrade of Boot Environment <second_disk> successful.

# luactivate second_disk
```

これらのコマンドの説明については、luupgrade(1M) と luactivate(1M) のマニュアルページを参照してください。

例 2 現在の BE 以外のソースを使って BE を作成する

-s オプションを使って、現在の BE 以外の BE をソース BE として指定します。

```
# lucreate -s third_disk -m /:/dev/dsk/c0t4d0s0:ufs \
-m /usr:/dev/dsk/c0t4d0s1:ufs -n second_disk
(出力略)
lucreate: Creation of Boot Environment <second_disk> successful.
```

例 3 BE をフラッシュアーカイブから作成する

このためには、-s - オプションを指定した lucreate と luupgrade を実行する必要があります。

```
# lucreate -s - -m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s1:ufs \
-n second_disk
```

lucreate(1M)

例 3 BE をフラッシュアーカイブから作成する (続き)

brief messages

```
lucreate: Creation of Boot Environment <second_disk> successful.
```

-s オプションを指定した lucreate コマンドは、数秒で処理を終了します。この時点で luupgrade を実行して、フラッシュアーカイブをインストールできます。

```
# luupgrade -f -n second_disk \  
-s /net/installmachine/export/solarisX/OS_image \  
-J "archive_location http://example.com/myflash.flar"
```

このコマンドの説明については、luupgrade(1M) のマニュアルページを参照してください。

例 4 スワップパーティションの共有および追加

もっとも単純な例として、lucreate コマンドでスワップパーティションを1つも指定しない場合、ソース BE のすべてのスワップパーティションは新しい BE と共有されます。たとえば、現在の BE がスワップパーティションとして /dev/dsk/c0t4d0s7 を使用すると仮定します。次のコマンドを入力します。

```
# lucreate -n second_disk -m /:/dev/dsk/c0t4d0s0:ufs  
(出力略)  
lucreate: Creation of Boot Environment <second_disk> successful.
```

このコマンドの結果、その BE がアクティブになり、起動したとき、パーティション /dev/dsk/c0t4d0s7 は BE second_disk によって使用されます。

新しい BE にソース BE とは異なるスワップパーティションを使用したい場合は、1つまたは複数の -m オプションを入力して新しいパーティションを指定します。ここで、現在の BE がスワップパーティションとして /dev/dsk/c0t4d0s7 を使用していると仮定します。次のコマンドを入力します。

```
# lucreate -m /:/dev/dsk/c0t0d0s0:ufs -m -:/dev/dsk/c0t4d0s1:swap \  
-m -:/dev/dsk/c0t4d0s2:swap -n second_disk  
(出力略)  
lucreate: Creation of Boot Environment <second_disk> successful.
```

BE がアクティブになり起動すると、新しい BE second_disk は /dev/dsk/c0t4d0s1 および /dev/dsk/c0t4d0s2 を使用し、/dev/dsk/c0t4d0s7 は使用しません。スワップパーティションはソース BE が使用します。

新しい BE second_disk がソース BE のスワップパーティションを共有していて、さらにスワップパーティションを追加したいと仮定します。次のように入力します。

```
# lucreate -m /:/dev/dsk/c0t0d0s0:ufs -m -:/dev/dsk/c0t4d0s1:swap \  
-m -:shared:swap -n second_disk  
(出力略)  
lucreate: Creation of Boot Environment <second_disk> successful.
```

例 4 スワップパーティションの共有および追加 (続き)

BE がアクティブになり起動すると、新しい BE `second_disk` はスワップに `/dev/dsk/c0t4d0s7` を使用して、ソース BE と共有し、さらに `/dev/dsk/c0t4d0s1` を使用します。

例 5 複数のディスク上にあるスワップパーティションを共有する

次のコマンドは、BE を 2 台目のディスクに作成し、1 台目と 2 台目の両方のディスク上にあるスワップパーティションを共有できるようにします。

```
# lucreate -m /:/dev/dsk/c0t4d0s0:ufs -m -:/dev/dsk/c0t4d0s1:swap \  
-m -:/dev/dsk/c0t0d0s1:swap -n second_disk  
(出力略)  
lucreate: Creation of Boot Environment <second_disk> successful.
```

上記のコマンドが完了すると、`second_disk` という BE は、`/dev/dsk/c0t0d0s1` と `/dev/dsk/c0t4d0s1` をスワップパーティションとして使用します。ただし、このスワップ割当は、`second_disk` からブートが行なわれるまで有効になりません。スワップパーティションが多数ある場合は、次の例のように `-M` オプションを使用すると便利です。

例 6 `-m` と `-M` オプションを組み合わせて使用する

この例では、スワップパーティションのリストを `/etc/lu/swapslices` ファイルに指定します。このファイルの場所と名前はユーザーが定義できます。`/etc/lu/swapslices` ファイルの内容を以下に示します。

```
-:/dev/dsk/c0t3d0s2:swap  
-:/dev/dsk/c0t3d0s2:swap  
-:/dev/dsk/c0t4d0s2:swap  
-:/dev/dsk/c0t5d0s2:swap  
-:/dev/dsk/c1t3d0s2:swap  
-:/dev/dsk/c1t4d0s2:swap  
-:/dev/dsk/c1t5d0s2:swap
```

上記のファイルは次のコマンドで指定されます。

```
# lucreate -m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s1:ufs \  
-M /etc/lu/swapslices -n second_disk  
(出力略)  
lucreate: Creation of Boot Environment <second_disk> successful.
```

BE `second_disk` は、`/etc/lu/swapslices` に指定されたパーティションをスワップとして使用します。

例 7 コピーと共有

次のコマンドは、現在の BE にあるユーザーファイルシステム `/home` (および、共有可能なファイルシステム `/` と `/usr`) を新しい BE にコピーします。

```
# lucreate -m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s1:ufs \  
-m /home:/dev/dsk/c0t4d0s4:ufs -n second_disk
```

例7 コピーと共有 (続き)

前述のコマンドとは異なり、次のコマンドには、/home の宛先を指定する -m オプションが指定されていません。コマンドを実行すると、/home は現在の BE と BE second_disk の間で共有されます。

```
# lucreate -m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s1:ufs \
-n second_disk
```

例8 Solaris ボリュームマネージャのボリュームの使用

以下のコマンドがあります。

1. ミラー d10 を作成して、このミラーをルートファイルシステムを受容体として確立します。
2. c0t0d0s0 および c0t1d0s0 を単一スライス連結 d1 および d2 にそれぞれ接続します。これらのボリュームの指定は任意であることに注意してください。
3. c0t0d0s0 および c0t1d0s0 に関連する連結をミラー d10 に接続します。
4. 現在の BE のルートファイルシステムをミラー d10 にコピーして、d10 の内容をすべて上書きします。

```
# lucreate -m /:/dev/md/dsk/d10:ufs,mirror \
-m /:/dev/dsk/c0t0d0s0,d1:attach \
-m /:/dev/dsk/c0t1d0s0,d2:attach -n newBE
```

以下のコマンドは、物理記憶装置用の連結が指定されていないという点のみ、先のコマンドと異なります。この例では、lucreate は空いている名前のリストから連結名を選択し、これらのボリュームを最初の -m オプションで指定されたミラーに接続します。

```
# lucreate -m /:/dev/md/dsk/d10:ufs,mirror \
-m /:/dev/dsk/c0t0d0s0:attach \
-m /:/dev/dsk/c0t1d0s0:attach -n newBE
```

次のコマンドは、作成するミラーにデバイスを接続する前に、ミラーから物理ディスクの1つを切り離すという点で、先のコマンドとは異なります。また、物理ディスクの1つの内容が保存されます。このコマンドは次のことを行います。

1. ミラー d10 を作成して、このミラーをルートファイルシステムを受容体として確立します。
2. c0t0d0s0 を現在接続されているミラーから切り離します。
3. c0t0d0s0 および c0t1d0s0 を連結 d1 および d2 にそれぞれ接続します。これらの連結の指定は任意であることに注意してください。
4. c0t0d0s0 の内容を保存します。c0t0d0s0 には現在の BE のルートファイルシステムの有効なコピーが含まれていると仮定します。
5. c0t0d0s0 および c0t1d0s0 に関連する連結 (d1 および d2) をミラー d10 に接続します。

```
# lucreate -m /:/dev/md/dsk/d10:ufs,mirror \
-m /:/dev/dsk/c0t0d0s0,d1:detach,attach,preserve \
-m /:/dev/dsk/c0t1d0s0,d2:attach -n newBE
```


例 8 Solaris ボリュームマネージャのボリュームの使用 (続き)

上のコマンドは、次のように短縮できます。

```
# lucreate -m /:d10:ufs,mirror \  
-m /:c0t0d0s0:detach,attach,preserve \  
-m /:c0t1d0s0:attach -n newBE
```

上の例では、デバイス名 (物理デバイス名および論理デバイス名) が短縮され、連結 (d1 および d2) の指定子が省略されていることに注意してください。

以下のコマンドは、この一連の例で最初に紹介したコマンドに続くコマンドです。このコマンドは、連結 (c0t0d0s0 を含む) を 1 つのミラー (最初のコマンドでは d10) から切り離し、別のミラー (d20) に接続してその内容を保存します。

```
# lucreate -m /:/dev/md/dsk/d20:ufs,mirror \  
-m /:/dev/dsk/c0t0d0s0:detach,attach,preserve -n nextBE
```

次のコマンドは、2 つのミラーを作成し、一方のミラー上には新しい BE の / ファイルシステムを、もう一方のミラー上には新しい BE の /opt ファイルシステムを置きます。

```
# lucreate -m /:/dev/md/dsk/d10:ufs,mirror \  
-m /:/dev/dsk/c0t0d0s0,d1:attach \  
-m /:/dev/dsk/c1t0d0s0,d2:attach \  
-m /opt:/dev/md/dsk/d11:ufs,mirror \  
-m /opt:/dev/dsk/c2t0d0s1,d3:attach \  
-m /opt:/dev/dsk/c3t1d0s1,d4:attach -n anotherBE
```

例 9 FMLI ベースのインタフェースの呼び出し

次のコマンドには -m と -M オプションのいずれも指定されていないため、Live Upgrade 操作を行う FMLI ベースのインタフェースが呼び出されます。このインタフェースの説明については、1u(1M) のマニュアルページを参照してください。

```
# lucreate -n second_disk
```

上記のコマンドは、ターゲット BE second_disk のソース BE として現在の BE を使用します。FMLI インタフェースで、second_disk のターゲットスライスを指定します。次のコマンドは上記のコマンドを変更したものです。

```
# lucreate -n second_disk -s third_disk
```

上記のコマンドには、ターゲット BE のソースが指定されています。前述の例と同様に、FMLI インタフェースが表示されるので、新しい BE のターゲットスライスを指定します。

例 10 ファイルシステムのマージ

次のコマンドは、ファイルシステム /usr/opt をファイルシステム /usr にマージします。まず以下の例では、BE first_disk にあるディスクスライスを -m オプションの引数形式で表したものです。

lucreate(1M)

例 10 ファイルシステムのマージ (続き)

```
/:/dev/dsk/c0t4d0s0:ufs
/usr:/dev/dsk/c0t4d0s1:ufs
/usr/opt:/dev/dsk/c0t4d0s3:ufs
```

次のコマンドは、BE `second_disk` を作成し、`/usr/opt` をその親の `/usr` にマージします。

```
# lucreate -m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s1:ufs \
-m /usr/opt:merged:ufs -n second_disk
```

例 11 ファイルシステムの分割

`/`、`/usr`、および `/var` のすべてが同じディスクスライス上にマウントされているソース BE があるとします。次のコマンドは、`/`、`/usr`、および `/var` がそれぞれ異なるディスクスライスにマウントされた BE `second_disk` を作成します。

```
# lucreate -m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s1:ufs \
/var:/dev/dsk/c0t4d0s3:ufs -n second_disk
```

ファイルシステム (ルートなど) のコンポーネントを異なるディスクスライスに分割するこのような操作を、ファイルシステムの分割といいます。

例 12 代替スライスの指定

次のコマンドは、新しい BE `second_disk` の代替ディスクスライスとして複数の `-m` オプションを使用します。

```
# lucreate -m /:/dev/dsk/c0t4d0s0:ufs -m /:/dev/dsk/c0t4d0s1:ufs \
-m /:/dev/dsk/c0t4d0s5:ufs -n second_disk
(出力略)
lucreate: Creation of Boot Environment <second_disk> successful.
```

上記のコマンドは、`/` ファイルシステムのディスクスライスの候補として `s0`、`s1`、`s5` を指定しています。lucreate は、この 3 つのスライスの中から他の BE によって使用されていない最初のスライスを選択します。`-s` オプションが省略されているため、新しい BE は現在の BE をソースとして作成されます。

終了ステータス 次の終了ステータスが返されます。

```
0      正常終了
>0     エラーが発生した
```

ファイル `/etc/lutab`
システム上にある BE のリスト

```
/usr/share/lib/xml/dtd/lu_cli.dtd.<num>
Live Upgrade の DTD (-x オプションを参照)
```

属性 次の属性の説明については、`attributes(5)` のマニュアルページを参照してください。

属性タイプ	使用条件
属性値	SUNWluu

- 関連項目** lu(1M), luactivate(1M), lucancel(1M), lucompare(1M), lucurr(1M), ludelete(1M), ludesc(1M), lufslist(1M), lumake(1M), lumount(1M), lurename(1M), lustatus(1M), luupgrade(1M), lutab(4), attributes(5), live_upgrade(5)
- 注意事項** すべての Solaris オペレーティング環境のアップグレードに当てはまりますが (Live Upgrade の機能は除きます)、ディレクトリを複数のマウントポイントに分割すると、ハードリンクは解除されます。たとえば、/usr/test1/buglist と /usr/test2/buglist がハードリンクされているとします。/usr/test1 と /usr/test2 を別々のファイルシステムに分割すると、これらのファイル間のリンクは切断されます。lucreate が複数のファイルシステムにまたがるハードリンクを検出すると、コマンドは警告メッセージを出力し、失われたハードリンクに代わるシンボリックリンクが作成されます。
- lucreate は、共有不能なファイルシステムについて無効な構成が作成されても検出できません。たとえば、/ と /kernel を別々のファイルシステムとして作成する lucreate コマンドを実行することは可能です。しかし、作成した BE はブートできません。ブート環境用のファイルシステムを作成するときは、Solaris オペレーティング環境用のファイルシステムを作成する場合と同じ規則が適用されます。
- これまでの説明をふまえて、以下のことに留意してください。
- ソース BE では、コピーしたり新しい BE と共有する各ファイルシステムの有効な vfstab エントリが必要です。
 - 重複するパーティション (つまり、同じ物理ディスク領域を共有するパーティション) には、新しい BE を作成することはできません。このようなディスクに BE を作成すると、lucreate コマンドは問題を検出しませんが、作成された BE はブートできません。
- 注 -m オプションの説明にあるとおり、Solaris ボリュームマネージャのボリュームをブート環境に使用する場合は、Solaris ボリュームマネージャのコマンドではなく lucreate を使用してこれらのボリュームを操作してください。Solaris ボリュームマネージャソフトウェアはブート環境を認識しません。lucreate コマンドには、たとえば誤って Solaris ボリュームマネージャのボリュームを上書きしたり削除したりしてブート環境を破壊することを避ける検査機能が含まれています。
- Live Upgrade は、配布中のリリース、および最大 3 つのリリースをさかのぼってサポートします。たとえば、Solaris 9 版 (Solaris 9 Upgrade 版を含む) の Live Upgrade バージョンを入手した場合、Solaris 9 に加えて Solaris バージョン 2.6、Solaris 7、および Solaris 8 版をサポートします。Solaris 2.6 より前のバージョンの Solaris をサポートする Live Upgrade バージョンは存在しません。

lucreate(1M)

Solaris Live Upgrade を正しく操作するためには、指定された OS バージョン用の制限付きパッチリビジョンのセットがインストールされている必要があります。Live Upgrade をインストールする、または実行する前に、制限付きパッチリビジョンのセットをインストールする必要があります。<http://sunsolve.sun.com> を参照して、最新のパッチリストがあることを確認してください。SunSolve Web サイトで `infodoc 72099` を検索してください。

名前	lucurr - アクティブなブート環境名の表示
形式	<code>/usr/sbin/lucurr [-l <i>error_log</i>] [-m <i>mount_point</i>] [-o <i>outfile</i>] [-X]</code>
機能説明	<p>lucurr コマンドは、Solaris オペレーティング環境の Live Upgrade 機能を提供するコマンド群の 1 つです。Live Upgrade 機能については、live_upgrade(5) のマニュアルページを参照してください。</p> <p>lucurr コマンドは、現在実行中のブート環境 (BE) の名前を表示します。システム上に BE が構成されていない場合は、'No Boot Environments are defined' というメッセージを表示します。lucurr で表示されるのは現在の BE の名前だけであり、次のリポート時にアクティブになる BE の名前は表示されません。次のリポート時にアクティブになる BE の情報が必要な場合は、lustatus(1M) を使用してください。</p> <p>lucurr コマンドを実行するには、root 権限が必要です。</p>
オプション	<p>lucurr コマンドには、次のオプションを指定できます。</p> <p><code>-l <i>error_log</i></code> エラーメッセージとステータスメッセージを、現在の環境での書き込み先だけでなく、<i>error_log</i> にも書き込みます。</p> <p><code>-m <i>mount_point</i></code> <i>mount_point</i> (BE のルートファイルシステムのマウントポイント) を所有している BE の名前を返します。<i>mount_point</i> には、現在の BE のマウントポイントまたはそれ以外の BE のマウントポイントのどちらでも指定できます。後者の場合、このオプションを使用する前に、lumount(1M) または mount(1M) を使って BE のファイルシステムをマウントしておく必要があります。</p> <p><code>-o <i>outfile</i></code> すべてのコマンド出力を、現在の環境での書き込み先だけでなく <i>outfile</i> にも書き込みます。</p> <p><code>-X</code> XML 出力を有効にします。XML の特性は DTD (<code>/usr/share/lib/xml/dtd/lu_cli.dtd.<num></code>) に定義されています。<num> は、各 DTD ファイルのバージョン番号を示します。</p>
終了ステータス	<p>次の終了ステータスが返されます。</p> <p>0 正常終了</p> <p>>0 エラーが発生した</p>
ファイル	<p><code>/etc/lutab</code> システム上にある BE のリスト</p> <p><code>/usr/share/lib/xml/dtd/lu_cli.dtd.<num></code> Live Upgrade の DTD (-X オプションを参照)</p>
属性	次の属性については、attributes(5) のマニュアルページを参照してください。

lucurr(1M)

属性タイプ	属性値
使用条件	SUNWluu

関連項目

lu(1M), luactivate(1M), lucancel(1M), lucompare(1M), lucreate(1M),
ludelete(1M), ludesc(1M), lufslis(1M), lumake(1M), lumount(1M),
lurename(1M), lustatus(1M), luupgrade(1M), lutab(4), attributes(5),
live_upgrade(5)

名前	ludelete – ブート環境の削除
形式	/usr/sbin/ludelete [-l <i>error_log</i>] [-o <i>outfile</i>] <i>BE_name</i> [-X]
機能説明	<p>ludelete コマンドは、Solaris オペレーティング環境の Live Upgrade 機能を提供するコマンド群の 1 つです。Live Upgrade 機能については、live_upgrade(5) のマニュアルページを参照してください。</p> <p>ludelete コマンドは、BE 上でステータスが完了になっているブート環境 (BE) に関連する、すべてのレコードを削除します。完了している BE とは、lucreate(1M)、luupgrade(1M)、または lucompare(1M) 操作が進行中でない BE のことです。BE のステータスを確認するには、lustatus(1M) を使用します。アクティブな BE と、次のリポート時にアクティブになる BE は削除できません。また、lumount(1M) または mount(1M) でファイルシステムがマウントされている BE は削除できません。</p> <p>ludelete は、削除される BE 上のファイルは変更しません。</p> <p>ludelete コマンドを実行するには、root 権限が必要です。</p>
オプション	<p>ludelete コマンドには、次のオプションを指定できます。</p> <p>-l <i>error_log</i> エラーメッセージとステータスメッセージを、現在の環境での書き込み先だけでなく、<i>error_log</i> にも書き込みます。</p> <p>-o <i>outfile</i> すべてのコマンド出力を、現在の環境での書き込み先だけでなく、<i>outfile</i> にも書き込みます。</p> <p>-X XML 出力を有効にします。XML の特性は DTD (<i>/usr/share/lib/xml/dtd/lu_cli.dtd.<num></i>) に定義されています。<num> は、各 DTD ファイルのバージョン番号を示します。</p>
オペラント	<i>BE_name</i> 削除する BE の名前を指定します。
終了ステータス	<p>次の終了ステータスが返されます。</p> <p>0 正常終了</p> <p>>0 エラーが発生した</p>
ファイル	<p><i>/etc/lutab</i> システム上にある BE のリスト</p> <p><i>/usr/share/lib/xml/dtd/lu_cli.dtd.<num></i> Live Upgrade の DTD (-X オプションを参照)</p>
属性	次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWluu

ludelete(1M)

関連項目	lu(1M), luactivate(1M), lucancel(1M), lucompare(1M), lucreate(1M), lucurr(1M), ludesc(1M), lufslis(1M), lumake(1M), lumount(1M), lurename(1M), lustatus(1M), luupgrade(1M), lutab(4), attributes(5), live_upgrade(5)
------	--

名前	lufslist - ブート環境の構成の一覧表示
形式	<code>/usr/sbin/lufslist [-l error_log] [-o outfile] BE_name [-X]</code>
機能説明	<p>lufslist コマンドは、Solaris オペレーティング環境の Live Upgrade 機能を提供するコマンド群の 1 つです。Live Upgrade 機能については、live_upgrade(5) のマニュアルページを参照してください。</p> <p>lufslist コマンドは、ブート環境 (BE) の構成を一覧表示します。出力内容は、BE のマウントポイントごとのディスクスライス (ファイルシステム)、ファイルシステムタイプ、およびファイルシステムのサイズです。</p> <p>lufslist の出力例を以下に示します。</p> <pre># lufslist BE_name Filesystem fstype size(Mb) Mounted on ----- /dev/dsk/c0t0d0s1 swap 512.11 - /dev/dsk/c0t4d0s3 ufs 3738.29 / /dev/dsk/c0t4d0s4 ufs 510.24 /opt</pre> <p>ファイルシステムタイプは ufs、swap、vxfs (Veritas ファイルシステム) のいずれかです。Filesystem の見出しの下には、ディスクスライスまたは論理デバイス (ボリューム管理ソフトウェアが使用するディスクメタデバイスなど) が表示されます。</p> <p>lufslist コマンドを実行するには、root 権限が必要です。</p>
オプション	<p>lufslist コマンドには、次のオプションを指定できます。</p> <p><code>-l error_log</code> エラーメッセージとステータスメッセージを、現在の環境での書き込み先だけでなく、<code>error_log</code> にも書き込みます。</p> <p><code>-o outfile</code> すべてのコマンド出力を、現在の環境での書き込み先だけでなく、<code>outfile</code> にも書き込みます。</p> <p><code>-X</code> XML 出力を有効にします。XML の特性は DTD (<code>/usr/share/lib/xml/dtd/lu_cli.dtd.<num></code>) に定義されています。<num> は、各 DTD ファイルのバージョン番号を示します。</p>
オペランド	<p><code>BE_name</code> ファイルシステムの情報を表示する BE の名前を指定します。その他の Live Upgrade 操作で使用されている BE は指定できません。</p>
終了ステータス	<p>次の終了ステータスが返されます。</p> <p>0 正常終了</p> <p>>0 エラーが発生した</p>
ファイル	<p><code>/etc/lutab</code> システム上にある BE のリスト</p> <p><code>/usr/share/lib/xml/dtd/lu_cli.dtd.<num></code> (XML の特性を参照)</p>
属性	次の属性については、attributes(5) のマニュアルページを参照してください。

lufslis(1M)

属性タイプ	属性値
使用条件	SUNWluu

関連項目 lu(1M), luactivate(1M), lucancel(1M), lucompare(1M), lucreate(1M), lucurr(1M), ludelete(1M), ludesc(1M), lumake(1M), lumount(1M), lurename(1M), lustatus(1M), luupgrade(1M), lutab(4), attributes(5), live_upgrade(5)

名前	lumake – ブート環境の生成
形式	<code>/usr/sbin/lumake [-l error_log] [-o outfile] [-s source_BE] -n BE_name [-X]</code> <code>lumake [-l error_log] [-o outfile] [-s source_BE] -n BE_name</code>
機能説明	<p>lumake コマンドは、Solaris オペレーティング環境の Live Upgrade 機能を提供するコマンド群の 1 つです。Live Upgrade 機能については、live_upgrade(5) のマニュアルページを参照してください。</p> <p>lumake コマンドは、ファイルをコピーすることにより、指定のブート環境 (BE) のファイルシステムを作成します。具体的には、アクティブな BE またはソース BE (-s) のファイルシステムからファイルをコピーします。ターゲット BE 上の既存のデータはすべて削除されます。ターゲット BE 上のすべてのファイルシステムは再作成されます。</p> <p>ターゲット BE には、必ず既存の BE を指定します。新しい BE を作成するには、lucreate(1M) を使用します。</p> <p>lumake コマンドを実行するには、root 権限が必要です。</p>
オプション	<p>lumake コマンドには、次のオプションを指定できます。</p> <p><code>-n BE_name</code> 生成する BE の名前を指定します。</p> <p><code>-s source_BE</code> オプションとしてソース BE の名前を指定します。このオプションを省略すると、現在の BE がソースとして使用されます。コピーを行う前に、BE のステータスが complete になっていることを確認してください。BE のステータスを確認するには、lustatus(1M) を使用します。</p> <p><code>-l error_log</code> エラーメッセージとステータスメッセージを、現在の環境での書き込み先だけでなく、<code>error_log</code> にも書き込みます。</p> <p><code>-o outfile</code> すべてのコマンド出力を、現在の環境での書き込み先だけでなく、<code>outfile</code> にも書き込みます。</p> <p><code>-t time</code> 指定の時刻に指定の BE を生成するバッチジョブを設定します。時刻の指定には、<code>at(1)</code> と同じ形式を使用します。複数の Live Upgrade 操作を同時にスケジュールすることはできません。lucancel(1M) を使用すると、スケジュール済みの lumake 操作を取り消すことができます。</p> <p><code>-m email_address</code> lumake コマンドの完了時に、その出力内容を電子メールで指定アドレスに送信できるようにします。<code>email_address</code> の検査は行われません。このオプションを使用するときは、同時に <code>-t</code> オプションも指定してください。</p>

lumake(1M)

-X XML 出力を有効にします。XML の特性は DTD (/usr/share/lib/xml/dtd/lu_cli.dtd.<num>) に定義されています。<num> は、各 DTD ファイルのバージョン番号を示します。

終了ステータス 次の終了ステータスが返されます。

0 正常終了
>0 エラーが発生した

ファイル /etc/lutab システム上にある BE のリスト

/usr/share/lib/xml/dtd/lu_cli.dtd.<num> Live Upgrade の DTD (-X オプションを参照)

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWluu

関連項目 lu(1M), luactivate(1M), lucancel(1M), lucompare(1M), lucreate(1M), lucurr(1M), ludelete(1M), ludesc(1M), lufslis(1M), lumount(1M), lurename(1M), lustatus(1M), luupgrade(1M), lutab(4), attributes(5), live_upgrade(5)

lumount(1M)

名前	lumount, luumount – ブート環境のすべてのファイルシステムのマウントまたはマウント解除
形式	<pre> /usr/sbin/lumount [-l error_log] [-o outfile] BE_name [mount_point] [-X] /usr/sbin/luumount [-f] { [-n] BE_name [-m] mount_point block_device} [-l error_log] [-o outfile] [-X] </pre>
機能説明	<p>lumount コマンドと luumount コマンドは、Solaris オペレーティング環境の Live Upgrade 機能を提供するコマンドです。Live Upgrade 機能については、live_upgrade(5) のマニュアルページを参照してください。</p> <p>lumount コマンドと luumount コマンドを使用すると、ブート環境 (BE) のすべてのファイルシステムをマウントまたはマウント解除できます。これにより、BE がアクティブでないときに、その BE のファイルを検査したり変更したりすることができます。デフォルトで lumount コマンドは、<code>/.alt.BE_name</code> という形式のマウントポイントにファイルシステムをマウントします。ここで、<code>BE_name</code> は、ファイルシステムをマウントする BE の名前です。詳細は「注意事項」の項を参照してください。</p> <p>lumount コマンドと luumount コマンドを実行するには、root 権限が必要です。</p>
オプション	<p>lumount コマンドと luumount コマンドには、次のオプションを指定できます。</p> <p>-f (luumount でのみ使用) マウント解除に失敗した場合に、強制的に BE のファイルシステムのマウントを解除します。このオプションは、umount(1M) の <code>-f</code> オプションと似ています。</p> <p>-l error_log エラーメッセージとステータスメッセージを、現在の環境での書き込み先だけでなく、<code>error_log</code> にも書き込みます。</p> <p>-m mount_point マウントポイント (<code>mount_point</code>) を所有している BE のファイルシステムのマウントを解除します。「オペランド」の項にある <code>mount_point</code> の説明を参照してください。マウントポイントを指定して luumount を実行する場合は、<code>-m</code> オプションは指定しなくてもかまいません。</p> <p>-n BE_name <code>BE_name</code> に指定した BE のファイルシステムのマウントを解除します。「オペランド」の項にある <code>BE_name</code> の説明を参照してください。BE の名前を指定して luumount を実行する場合は、<code>-n</code> オプションは指定しなくてもかまいません。</p> <p>-o outfile すべてのコマンド出力を、現在の環境での書き込み先だけでなく、<code>outfile</code> にも書き込みます。</p> <p>-X XML 出力を有効にします。XML の特性は DTD (<code>/usr/share/lib/xml/dtd/lu_cli.dtd.<num></code>) に定義されています。<code><num></code> は、各 DTD ファイルのバージョン番号を示します。</p>

lumount(1M)

luumount コマンドに引数を指定して、`-m` と `-n` オプションのどちらも指定しなかった場合、luumount コマンドは、その引数が BE の名前、マウントポイント、ブロックデバイスのうちのどれに該当するかを判別します。引数が3つのうちのどれかに該当し、ファイルシステムをマウントしている BE に関連している場合、luumount は、その BE のファイルシステムのマウントを解除します。そうでない場合は、エラーを返します。

オペランド	<p><i>BE_name</i> ファイルシステムをマウントまたはマウント解除する BE の名前を指定します。現在のシステム上の非アクティブな BE を指定してください。lumount コマンドと luumount コマンドを正しく完了するには、<code>lustatus(1M)</code> を使って BE のステータスが <code>complete</code> であることをあらかじめ確認しておきます。なお、<code>mount(1M)</code> を使って BE のディスクスライスをマウントすることはできません。</p> <p><i>mount_point</i> lumount コマンドに使用する場合は、デフォルトのマウントポイント <code>/.alt.BE_name</code> に代わるマウントポイントを指定します。<i>mount_point</i> が存在しない場合は、新たに作成されます。luumount に使用する場合は、ファイルシステムのマウントを解除したい BE に関連しているマウントポイントを指定します。luumount を使用してマウントを解除すると、デフォルトのマウントポイントは、自動的に削除されます。指定したマウントポイントは削除されません。</p> <p><i>block_device</i> (luumount でのみ使用) BE のルートスライス (例: <code>/dev/dsk/c0t4d0s0</code>) を指定します。luumount は、<i>block_device</i> に指定したブロックデバイスに関連している BE のファイルシステムのマウントを解除します。</p>
使用例	<p>例 1 マウントポイントを指定する</p> <p>次のコマンドは、マウントポイント <code>/test</code> を作成し、BE <code>second_disk</code> のファイルシステムを <code>/test</code> にマウントします。</p> <pre># lumount second_disk /test /test</pre> <p><code>second_disk</code> のファイルシステムを確認するには、<code>/test</code> ディレクトリに移動します。マウントポイントとして <code>/test</code> を指定しなかった場合、lumount は、<code>/.alt.second_disk</code> という名前のデフォルトのマウントポイントを作成します。</p> <p>例 2 ファイルシステムをマウント解除する</p> <p>次のコマンドは、BE <code>second_disk</code> のファイルシステムをマウント解除します。この例では、<code>second_disk</code> のファイルシステム外で操作を行うために、<code>/</code> に移動しています。</p> <pre># cd / # luumount second_disk #</pre>

例 2 ファイルシステムをマウント解除する (続き)

/dev/dsk/c0t4d0s0 が second_disk のルートスライスである場合は、次のように指定すると、前述のコマンドと同じ効果が得られます。

```
# cd /
# luumount /dev/dsk/c0t4d0s0
#
```

終了ステータス 次の終了ステータスが返されます。

```
0          正常終了
>0        エラーが発生した
```

ファイル /etc/lutab
システム上にある BE のリスト

/usr/share/lib/xml/dtd/lu_cli.dtd.<num>
Live Upgrade の DTD (-x オプションを参照)

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWluu

関連項目 lu(1M), luactivate(1M), lucancel(1M), lucompare(1M), lucreate(1M), lucurr(1M), ludelete(1M), ludesc(1M), lufslis(1M), lumake(1M), lurename(1M), lustatus(1M), luupgrade(1M), lutab(4), attributes(5), live_upgrade(5)

注意事項 BE の名前にスラッシュ (/) 記号が含まれている場合、そのスラッシュ記号は、デフォルトのマウントポイント名ではコロン(:) に置き換えられます。以下に例を示します。

```
# lumount 'first/disk'
/.alt.first:disk
```

lurename(1M)

名前	lurename – ブート環境名の変更
形式	<code>/usr/sbin/lurename -e BE_name -n new_name [-l error_log] [-o outfile] [-X]</code>
機能説明	<p>lurename コマンドは、Solaris オペレーティング環境の Live Upgrade 機能を提供するコマンド群の 1 つです。Live Upgrade 機能については、live_upgrade(5) のマニュアルページを参照してください。</p> <p>lurename コマンドは、ブート環境 (BE) の名前を <i>BE_name</i> から <i>new_name</i> に変更します。</p> <p><i>new_name</i> は 30 文字を超えてはならず、使用できる文字は、英数字とその他の ASCII 文字 (UNIX シェルにとって特殊な意味を持つ文字は除く) だけです。sh(1) の「クォート」セクションを参照してください。BE 名に使用できるのは、8 ビットで表現できるシングルバイトの文字だけです。空白文字を含めることはできません。<i>new_name</i> は、システム上で一意のものでなければなりません。</p> <p>BE 名を変更する前に、BE のステータスが complete になっていることを確認してください。BE のステータスを確認するには、lustatus(1M) を使用します。なお、lumount(1M) または mount(1M) でファイルシステムがマウントされている BE の名前は変更できません。</p> <p>BE 名の変更は、BE を Solaris の新しいリリースにアップグレードするときに有用です。たとえば、オペレーティングシステムのアップグレードに伴って、BE 名を solaris7 から solaris8 に変更できます。</p> <p>lurename コマンドを実行するには、root 権限が必要です。</p>
オプション	<p>lurename コマンドには、次のオプションを指定できます。</p> <p><code>-e BE_name</code> 変更したい BE の名前を指定します。</p> <p><code>-l error_log</code> エラーメッセージとステータスメッセージを、現在の環境での書き込み先だけでなく、<i>error_log</i> にも書き込みます。</p> <p><code>-n new_name</code> 新しい BE 名を指定します。<i>new_name</i> は、システム上で一意のものでなければなりません。</p> <p><code>-o outfile</code> すべてのコマンド出力を、現在の環境での書き込み先だけでなく、<i>outfile</i> にも書き込みます。</p> <p><code>-X</code> XML 出力を有効にします。XML の特性は DTD (<code>/usr/share/lib/xml/dtd/lu_cli.dtd.<num></code>) に定義されています。<num> は、各 DTD ファイルのバージョン番号を示します。</p>
終了ステータス	<p>次の終了ステータスが返されます。</p> <p>0 正常終了</p> <p>>0 エラーが発生した</p>

lurename(1M)

ファイル

/etc/lutab
システム上にある BE のリスト

/usr/share/lib/xml/dtd/lu_cli.dtd.<num>
Live Upgrade の DTD (-x オプションを参照)

属性

次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWluu

関連項目

lu(1M), luactivate(1M), lucancel(1M), lucompare(1M), lucreate(1M),
lucurr(1M), ludelete(1M), ludesc(1M), lufslis(1M), lumake(1M),
lumount(1M), lustatus(1M), luupgrade(1M), lutab(4), attributes(5),
live_upgrade(5)

lustatus(1M)

名前	lustatus - ブート環境のステータスの表示																				
形式	<code>/usr/sbin/lustatus [-l error_log] [-o outfile] [BE_name] [-X]</code>																				
機能説明	<p>lustatus コマンドは、Solaris オペレーティング環境の Live Upgrade 機能を提供するコマンド群の 1 つです。Live Upgrade 機能については、live_upgrade(5) のマニュアルページを参照してください。</p> <p>lustatus コマンドは、ブート環境 (BE) <i>BE_name</i> のステータスを表示します。BE を指定しない場合は、システム上のすべての BE のステータスを表示します。</p> <p>lustatus コマンドでは、次の見出しの下に情報が出力されます。</p> <p>BE_name BE の名前です。</p> <p>Complete BE がブート可能かどうかを示します。現在何らかの操作が行われている場合や、lucreate(1M) または luupgrade(1M) の操作に失敗した場合、BE は未完了のステータスになります。たとえば、BE 上でコピー操作が実行中であつたり、コピー操作がスケジュールされている場合、その BE のステータスは未完了と見なされま</p> <p>Active BE が現在アクティブであるかどうかを示します。「アクティブな」BE とは、現在ブートされている BE です。</p> <p>ActiveOnReboot BE が次のシステムのリブート時にアクティブになるかどうかを示します。</p> <p>CopyStatus BE の作成や再生成がスケジュールされているかどうか、あるいは、アクティブ (実行中) かどうかを示します。この見出しの下に、ACTIVE、COMPARING (lucompare(1M) を参照)、UPGRADING、または SCHEDULED というステータスが表示されている場合、Live Upgrade 機能を使ったコピー、名前の変更、アップグレードは実行できません。</p> <p>lustatus の出力例を以下に示します。</p> <table><thead><tr><th>BE_name</th><th>Complete</th><th>Active</th><th>ActiveOnReboot</th><th>CopyStatus</th></tr></thead><tbody><tr><td>disk_a_S7</td><td>yes</td><td>yes</td><td>yes</td><td>-</td></tr><tr><td>disk_b_S7database</td><td>yes</td><td>no</td><td>no</td><td>UPGRADING</td></tr><tr><td>disk_b_S8</td><td>no</td><td>no</td><td>no</td><td>-</td></tr></tbody></table> <p>未完了である disk_b_S8 のコピー、名前の変更、アップグレードは実行できません。同じく、Live Upgrade 操作が保留状態になっている disk_b_S7database のコピー、名前の変更、アップグレードも実行できないことに注意してください。</p> <p>lustatus コマンドを実行するには、root 権限が必要です。</p>	BE_name	Complete	Active	ActiveOnReboot	CopyStatus	disk_a_S7	yes	yes	yes	-	disk_b_S7database	yes	no	no	UPGRADING	disk_b_S8	no	no	no	-
BE_name	Complete	Active	ActiveOnReboot	CopyStatus																	
disk_a_S7	yes	yes	yes	-																	
disk_b_S7database	yes	no	no	UPGRADING																	
disk_b_S8	no	no	no	-																	
オプション	<p>lustatus コマンドには、次のオプションを指定できます。</p> <p><code>-l error_log</code> エラーメッセージとステータスメッセージを、現在の環境での書き込み先だけでなく、<i>error_log</i> にも書き込みます。</p>																				

- o *outfile* すべてのコマンド出力を、現在の環境での書き込み先だけでなく、*outfile* にも書き込みます。
- x XML 出力を有効にします。XML の特性は DTD (/usr/share/lib/xml/dtd/lu_cli.dtd.<num>) に定義されています。<num> は、各 DTD ファイルのバージョン番号を示します。
- オペランド *BE_name* ステータスを表示する BE の名前を指定します。*BE_name* を省略すると、システム内のすべての BE のステータスが表示されます。
- 終了ステータス 次の終了ステータスが返されます。
 - 0 正常終了
 - >0 エラーが発生した
- ファイル */etc/lutab* システム上にある BE のリスト
/usr/share/lib/xml/dtd/lu_cli.dtd.<num>
Live Upgrade の DTD (-x オプションを参照)
- 属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWluu

関連項目 *lu(1M)*, *luactivate(1M)*, *lucancel(1M)*, *lucompare(1M)*, *lucreate(1M)*, *lucurr(1M)*, *ludesc(1M)*, *ludelete(1M)*, *lufslis(1M)*, *lumake(1M)*, *lumount(1M)*, *lurename(1M)*, *luupgrade(1M)*, *lutab(4)*, *attributes(5)*, *live_upgrade(5)*

luumount(1M)

名前	lumount, luumount – ブート環境のすべてのファイルシステムのマウントまたはマウント解除
形式	<pre>/usr/sbin/lumount [-l error_log] [-o outfile] BE_name [mount_point] [-X] /usr/sbin/luumount [-f] { [-n] BE_name [-m] mount_point block_device} [-l error_log] [-o outfile] [-X]</pre>
機能説明	<p>lumount コマンドと luumount コマンドは、Solaris オペレーティング環境の Live Upgrade 機能を提供するコマンドです。Live Upgrade 機能については、live_upgrade(5) のマニュアルページを参照してください。</p> <p>lumount コマンドと luumount コマンドを使用すると、ブート環境 (BE) のすべてのファイルシステムをマウントまたはマウント解除できます。これにより、BE がアクティブでないときに、その BE のファイルを検査したり変更したりすることができます。デフォルトで lumount コマンドは、<code>/.alt.BE_name</code> という形式のマウントポイントにファイルシステムをマウントします。ここで、<code>BE_name</code> は、ファイルシステムをマウントする BE の名前です。詳細は「注意事項」の項を参照してください。</p> <p>lumount コマンドと luumount コマンドを実行するには、root 権限が必要です。</p>
オプション	<p>lumount コマンドと luumount コマンドには、次のオプションを指定できます。</p> <ul style="list-style-type: none">-f (luumount でのみ使用) マウント解除に失敗した場合に、強制的に BE のファイルシステムのマウントを解除します。このオプションは、umount(1M) の -f オプションと似ています。-l <i>error_log</i> エラーメッセージとステータスメッセージを、現在の環境での書き込み先だけでなく、<i>error_log</i> にも書き込みます。-m <i>mount_point</i> マウントポイント (<i>mount_point</i>) を所有している BE のファイルシステムのマウントを解除します。「オペランド」の項にある <i>mount_point</i> の説明を参照してください。マウントポイントを指定して luumount を実行する場合は、-m オプションは指定しなくてもかまいません。-n <i>BE_name</i> <i>BE_name</i> に指定した BE のファイルシステムのマウントを解除します。「オペランド」の項にある <i>BE_name</i> の説明を参照してください。BE の名前を指定して luumount を実行する場合は、-n オプションは指定しなくてもかまいません。-o <i>outfile</i> すべてのコマンド出力を、現在の環境での書き込み先だけでなく、<i>outfile</i> にも書き込みます。-X XML 出力を有効にします。XML の特性は DTD (<code>/usr/share/lib/xml/dtd/lu_cli.dtd.<num></code>) に定義されています。<code><num></code> は、各 DTD ファイルのバージョン番号を示します。

luumount コマンドに引数を指定して、`-m` と `-n` オプションのどちらも指定しなかった場合、luumount コマンドは、その引数が BE の名前、マウントポイント、ブロックデバイスのうちのどれに該当するかを判別します。引数が 3 つのうちのどれかに該当し、ファイルシステムをマウントしている BE に関連している場合、luumount は、その BE のファイルシステムのマウントを解除します。そうでない場合は、エラーを返します。

オペランド	<i>BE_name</i>	ファイルシステムをマウントまたはマウント解除する BE の名前を指定します。現在のシステム上の非アクティブな BE を指定してください。lumount コマンドと luumount コマンドを正しく完了するには、lustatus(1M) を使って BE のステータスが <code>complete</code> であることをあらかじめ確認しておきます。なお、mount(1M) を使って BE のディスクスライスをマウントすることはできません。
	<i>mount_point</i>	lumount コマンドに使用する場合は、デフォルトのマウントポイント <code>/.alt.BE_name</code> に代わるマウントポイントを指定します。 <i>mount_point</i> が存在しない場合は、新たに作成されます。luumount に使用する場合は、ファイルシステムのマウントを解除したい BE に関連しているマウントポイントを指定します。luumount を使用してマウントを解除すると、デフォルトのマウントポイントは、自動的に削除されます。指定したマウントポイントは削除されません。
	<i>block_device</i>	(luumount でのみ使用) BE のルートスライス (例: <code>/dev/dsk/c0t4d0s0</code>) を指定します。luumount は、 <i>block_device</i> に指定したブロックデバイスに関連している BE のファイルシステムのマウントを解除します。

使用例 例 1 マウントポイントを指定する

次のコマンドは、マウントポイント `/test` を作成し、BE `second_disk` のファイルシステムを `/test` にマウントします。

```
# lumount second_disk /test
/test
```

`second_disk` のファイルシステムを確認するには、`/test` ディレクトリに移動します。マウントポイントとして `/test` を指定しなかった場合、lumount は、`/.alt.second_disk` という名前のデフォルトのマウントポイントを作成します。

例 2 ファイルシステムをマウント解除する

次のコマンドは、BE `second_disk` のファイルシステムをマウント解除します。この例では、`second_disk` のファイルシステム外で操作を行うために、`/` に移動しています。

```
# cd /
# luumount second_disk
#
```

luumount(1M)

例 2 ファイルシステムをマウント解除する (続き)

/dev/dsk/c0t4d0s0 が second_disk のルートスライスである場合は、次のように指定すると、前述のコマンドと同じ効果が得られます。

```
# cd /
# luumount /dev/dsk/c0t4d0s0
#
```

終了ステータス 次の終了ステータスが返されます。

```
0          正常終了
>0         エラーが発生した
```

ファイル /etc/lutab
システム上にある BE のリスト

```
/usr/share/lib/xml/dtd/lu_cli.dtd.<num>
Live Upgrade の DTD (-x オプションを参照)
```

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWluu

関連項目 lu(1M), luactivate(1M), lucancel(1M), lucompare(1M), lucreate(1M), lucurr(1M), ludelete(1M), ludesc(1M), lufslis(1M), lumake(1M), lurename(1M), lustatus(1M), luupgrade(1M), lutab(4), attributes(5), live_upgrade(5)

注意事項 BE の名前にスラッシュ (/) 記号が含まれている場合、そのスラッシュ記号は、デフォルトのマウントポイント名ではコロン(:) に置き換えられます。以下に例を示します。

```
# lumount 'first/disk'
/.alt.first:disk
```

luupgrade(1M)

名前	luupgrade – ブート環境上のソフトウェアのインストール、アップグレード、およびその他の機能の実行
形式	<code>/usr/sbin/luupgrade [-iIufpPtTcC] [options]</code>
機能説明	<p>luupgrade コマンドは、Solaris オペレーティング環境の Live Upgrade 機能を提供するコマンド群の 1 つです。Live Upgrade 機能については、live_upgrade(5) のマニュアルページを参照してください。</p> <p>luupgrade コマンドは、指定したブート環境 (BE) にソフトウェアをインストールできるようにします。具体的な機能は次のとおりです。</p> <ul style="list-style-type: none"> ■ BE 上のオペレーティングシステムイメージをアップグレードする (-u オプション)。Solaris フラッシュアーカイブなど、有効な Solaris インストール媒体をイメージのソースとして指定できる ■ インストーラプログラムを実行し、インストール媒体からソフトウェアをインストールする (-i オプション) ■ BE 上に Solaris フラッシュアーカイブを抽出する (-f オプション)。flar(1M) のマニュアルページを参照 ■ BE にパッケージを追加する (-p)。または、BE からパッケージを削除する (-P) ■ BE にパッチを追加する (-t)。または、BE からパッチを削除する (-T) ■ パッケージに関する情報をチェックします (-I)。または、パッケージの情報を確認する (-c) ■ オペレーティングシステムのインストール媒体を確認する (-c)。 <p>luupgrade を使用する前に、lucreate(1M) コマンドまたは、FMLI ベースのユーザーインタフェース lu(1M) を使って BE を作成しておきます。アップグレードできるのは、現在の BE 以外の BE です。</p> <p>これらの機能では、それぞれに固有のオプションを使用できます。各機能で使用可能なオプションについてはその機能の説明を参照してください。</p> <p>luupgrade を正しく完了するには、lustatus(1M) で取得される BE のステータスが、complete でなければならないことに注意してください。なお、lumount(1M) または mount(1M) でディスクスライスがマウントされている BE に対しては、luupgrade を実行できません。</p> <p>luupgrade を使用すると、luupgrade を呼び出すマシンで実行しているリリースとは別のマーケティングリリースの Solaris オペレーティングシステムイメージをインストールできます。この機能を使用するには、次の条件があります。</p> <ul style="list-style-type: none"> ■ 以前のリリースを実行しているマシンに、指定した Solaris オペレーティングシステムのリリースから Live Upgrade パッケージ (SUNWluu および SUNWlur) をインストールできます。これらのパッケージは、Live Upgrade パッケージのリリースより 3 つ前のリリースまでの Solaris を実行しているマシン上にインストールできます。Live Upgrade は Solaris 2.6 より前のリリースではサポートされていません。そのため、たとえば、Solaris 2.9 パッケージは Solaris 2.8、2.7、および 2.6 マシンにインストールできます。

luupgrade(1M)

- マシンにインストールされている Live Upgrade パッケージと同じリリースの Solaris オペレーティングシステムリリースにアップグレードできます。この機能により、マーケティングリリース範囲内の Solaris アップグレードリリースにアップグレードできます。たとえば、マシンに Solaris 9 FCS Live Upgrade パッケージがインストールされている場合は、luupgrade を使用して BE を Solaris オペレーティングシステムの Update 3 リリースにアップグレードできます。

Live Upgrade パッケージのインストール方法については、『Solaris 9 インストールガイド』を参照してください。

luupgrade コマンドを実行するには、root 権限が必要です。

常に指定可能なオプション

次のオプションは、どのような目的で luupgrade を実行する場合でも指定できません。

- l *error_log* エラーメッセージと状態メッセージを、現在の環境での書き込み先だけでなく、*error_log* にも書き込みます。
- o *outfile* すべてのコマンド出力を、現在の環境での書き込み先だけでなく、*outfile* にも書き込みます。
- N コマンド引数の指定が正しいかどうかを確認できる dry-run モードです。-c (媒体の検査) には適用されません。
- X XML 出力を有効にします。XML の特性は DTD (`/usr/share/lib/xml/dtd/lu_cli.dtd.<num>`) に定義されています。<num> は、各 DTD ファイルのバージョン番号を示します。

オペレーティングシステムイメージのアップグレード

オペレーティングシステムイメージをアップグレードするには、luupgrade コマンドに -u を指定します。この場合の構文は次のとおりです。

```
luupgrade -u -n BE_name -s os_image_path [ -j profile_path ]  
[ -l error_log ] [ -o outfile ] [-N]
```

最初のオプション -u は、OS イメージのインストールを実行することを示します。この luupgrade を使用する場合に指定可能なその他のオプションは、次のとおりです。

- n *BE_name* OS をアップグレードする BE の名前のどちらでも指定できます。
- s *os_image_path* OS イメージが格納されているディレクトリのパス名を指定します。CD-ROM などのインストール媒体上のディレクトリか、NFS または UFS ディレクトリのどちらでも指定できます。
- j *profile_path* JumpStart プロファイルのパスを指定します。luupgrade で呼び出すプロファイルで使用する有効なキーワードのリストについては、後述のセクション「JumpStart プロファイルのキーワード」を参照してください。JumpStart ソフトウェアについては、pfinstall(1M) のマニュアルページと Solaris のインストールマニュアルを参照してください。

luupgrade(1M)

-D 指定した BE のディスク構成に対して **-j** で指定されたプロファイルの値をテストします。このオプションの結果は、プロファイルをテストする事前実行です。luupgrade はその出力に指定されたログファイルを作成するので、コマンドの結果を調べることができます。

複数のコンポーネントから成る媒体からアップグレードを行う場合 (複数の CD からアップグレードを行う場合など) は、**-i** オプションを指定して、2 番目以降の媒体からソフトウェアをインストールできるようにします。具体的には、次の項を参照してください。

インストーラプログラムの実行

インストーラプログラムを実行するには、luupgrade コマンドに **-i** を指定します。この場合の構文は次のとおりです。

```
luupgrade -i -n BE_name -s installation_medium [ -N ]  
[ -O "installer_options" ] [ -l error_log ] [ -o outfile ]
```

最初のオプション **-i** は、**-s** で指定されたインストール上のインストーラプログラムを実行することを示します。この luupgrade を使用する場合に指定可能なその他のオプションは次のとおりです。

-n BE_name ソフトウェアのインストール先となる BE の名前を指定します。

-O "installer_options" Solaris インストーラプログラムに直接渡されるオプションを指定します。インストーラオプションについては、`installer(1M)` を参照してください。

-s installation_medium インストール媒体のパス名を指定します。CD などのインストール媒体上のディレクトリか、NFS または UFS ディレクトリのどちらでも指定できます。

-i オプションを指定すると、luupgrade は指定した媒体上でインストールプログラムを検索し、実行します。

-i オプションを前述の **-u** オプションと組み合わせて指定すると、複数のコンポーネントから成る媒体 (複数の CD など) からインストールを行えます。

Solaris フラッシュアーカイブからのインストール

Solaris フラッシュアーカイブからオペレーティングシステムをインストールするには、luupgrade コマンドに **-f** を指定します。アーカイブをインストールすると、ターゲットとなる BE のすべてのファイルが上書きされることに注意してください。この場合の構文は次のとおりです。

```
luupgrade -f -n BE_name -s os_image_path ( -a archive | -j profile_path  
| -J "profile" ) [ -l error_log ] [ -o outfile ] [ -N ]
```

最初のオプション **-f** は、Solaris フラッシュアーカイブから OS のインストールを行うことを示します。この luupgrade を使用する場合に指定可能なその他のオプションは次のとおりです。

-n BE_name OS のインストール先となる BE の名前のどちらでも指定できます。

luupgrade(1M)

<code>-s os_image_path</code>	OS イメージが格納されているディレクトリのパス名を指定します。CD-ROM などのインストール媒体上のディレクトリか、NFS または UFS ディレクトリのどちらでも指定できます。
<code>-a archive</code>	ローカルファイルシステム上でアーカイブが使用可能な状態になっているときは、Solaris フラッシュアーカイブのパスを指定します。-a、-j、または -J のいずれかを指定する必要があります。
<code>-j profile_path</code>	Solaris フラッシュインストール用に構成された JumpStart プロファイルのパスを指定します。luupgrade で呼び出すプロファイルで使用する有効なキーワードのリストについては、後述のセクション「JumpStart プロファイルのキーワード」を参照してください。JumpStart ソフトウェアについては、pfinstall(1M) のマニュアルページと Solaris のインストールマニュアルを参照してください。-a、-j、または -J のいずれかを指定する必要があります。
<code>-J "profile"</code>	Solaris フラッシュインストール用に構成された JumpStart プロファイルのエントリを指定します。このオプションに有効な唯一のキーワードは、archive_location です。JumpStart ソフトウェアについては、pfinstall(1M) のマニュアルページと Solaris のインストールマニュアルを参照してください。-a、-j、または -J のいずれかを指定する必要があります。
<code>-D</code>	指定した BE のディスク構成に対して -j または -J で指定されたプロファイルの値をテストします。このオプションの結果は、プロファイルをテストする事前実行です。luupgrade はその出力に指定されたログファイルを作成するので、コマンドの結果を調べることができます。

-s で指定した OS イメージのバージョンは、-a、-j、-J のいずれかのオプションで指定した Solaris フラッシュアーカイブ内の OS のバージョンと同じでなければなりません。

パッケージの追加と削除

パッケージを追加する場合は、luupgrade コマンドに -p を指定します。パッケージを削除する場合は -P を指定します。構文は次のとおりです。

パッケージの追加:

```
luupgrade -p -n BE_name -s packages_path [ -l error_log ] [ -o outfile ]  
[ -O "pkgadd_options" ] [ -a admin ] [ pkginst [ pkginst... ] ] [ -N ]
```

パッケージの削除:

```
luupgrade -P -n BE_name [ -l error_log ][ -o outfile ]
[ -O "pkgadm_options" ] [ pkginst [ pkginst... ] ] [ -N ]
```

最初のオプションは、パッケージを追加する場合は `-p`、削除する場合は `-P` を指定します。この `luupgrade` を使用する場合に指定可能なその他のオプションは次のとおりです。

`-n BE_name`

パッケージの追加先または削除先となる BE の名前を指定します。

`-s packages_path`

追加するパッケージが格納されているディレクトリのパス名を指定します (パッケージの追加の場合のみ)。`-d` オプションで代用することもできます。`-d` は、`pkgadd(1M)` との互換性を確保する目的でサポートされています。

`-d packages_path`

`-s` オプションと同じです。`-s` オプションの使用をお勧めします。

`-O "pkgadd_options" または "pkgadm_options"`

パッケージを追加する場合 (`-p`) は `pkgadd` に直接渡されるオプション、削除する場合 (`-P`) は `pkgadm` に直接渡されるオプションを指定します。これらのコマンドのオプションについては、`pkgadd(1M)` のマニュアルページと `pkgadm(1M)` のマニュアルページを参照してください。

`-a admin`

`admin` ファイルのパスを指定します (パッケージの追加の場合のみ)。`pkgadd` の `-a` オプションと同じ意味になります。ここでの `-a` オプションは、`-O "-a admin"` と同等です。

`pkginst [pkginst...]`

追加または削除するゼロ個以上のパッケージを指定します。パッケージを追加する場合、上記の `-s` オプションで指定したすべてのパッケージが追加されます (デフォルト)。複数のパッケージ名は空白文字で区切って指定します。

追加するパッケージは、すべて『SVR4 Advanced Packaging Guidelines』に準拠していなければなりません。「警告」の項を参照してください。

パッチの追加と削除

パッチを追加する場合は、`luupgrade` コマンドに `-t` を指定します。パッチを削除する場合は `-T` を指定します。構文は次のとおりです。

パッチの追加:

```
luupgrade -t -n BE_name -s patch_path [ -l error_log ][ -o outfile ]
[ -O "patchadd_options" ] [ patch_name [ patch_name... ] ] [ -N ]
```

パッチの削除:

```
luupgrade -T -n BE_name [ -l error_log ][ -o outfile ]
[ -O "patchrm_options" ] [ patch_name [ patch_name... ] ] [ -N ]
```

最初のオプションは、パッチを追加する場合は `-t`、削除する場合は `-T` を指定します。この `luupgrade` を使用する場合に指定可能なその他のオプションは、次のとおりです。

luupgrade(1M)

パッケージ情報の
確認と表示

`-n BE_name`
パッチの追加先となる BE の名前、またはパッチを削除する BE の名前を指定します。

`-s patch_path`
追加するパッチが格納されているディレクトリのパス名を指定します (パッチの追加の場合のみ)。

`-o "patchadd_options" または "patchrm_options"`
パッチを追加する場合 (`-p`) は `patchadd` に直接渡されるオプション、削除する場合 (`-P`) は `patchrm` に直接渡されるオプションを指定します。これらのコマンドのオプションについては、`patchadd(1M)` のマニュアルページまたは `patchrm(1M)` のマニュアルページを参照してください。

`patch_name [patch_name...]`
追加または削除するゼロ個以上のパッチを指定します。パッチを追加する場合、上記の `-s` オプションで指定したすべてのパッチが追加されます (デフォルト)。複数のパッチ名は空白文字で区切って指定します。

追加するパッケージは、すべて『SVR4 Advanced Packaging Guidelines』に準拠していなければなりません。「警告」の項を参照してください。

`-c` を指定すると、BE 上のすべてのパッケージまたは指定したパッケージに対して `pkgchk(1M)` が実行されます。`-I` オプションを指定すると、`pkginfo(1)` が実行されます。

`pkgchk` の実行:

```
luupgrade -C -n BE_name [ -l error_log ][ -o outfile ]  
[ -O "pkgchk_options" ][ pkginst [ pkginst... ] ] [ -N ]
```

`pkginfo` の実行:

```
luupgrade -I -n BE_name [ -l error_log ][ -o outfile ]  
[ -O "pkginfo_options" ][ pkginst [ pkginst... ] ] [ -N ]
```

最初のオプションは、`pkgchk` の場合は `-C`、`pkginfo` の場合は `-I` を指定します。この `luupgrade` を使用する場合に指定可能なその他のオプションは、次のとおりです。

`-n BE_name`
パッケージを確認する BE、またはパッケージ情報を表示する BE の名前を指定します。

`-o "pkgchk_options" または "pkginfo_options"`
`pkgchk` に直接渡されるオプション (`-C`) または `pkginfo` に直接渡されるオプション (`-I`) を指定します。これらのオプションについては、`pkgchk(1M)` のマニュアルページまたは `pkginfo(1)` のマニュアルページを参照してください。

`pkginst [pkginst...]`
チェックするゼロ個以上のパッケージ名、または情報を表示するゼロ個以上のパッケージ名を指定します。パッケージ名を省略すると、BE 上のすべてのパッケージ情報が返されます。複数のパッケージ名は空白文字で区切って指定しま

OS のインストール
媒体の確認

JumpStart プロ
ファイルのキー
ワード

す。

-c オプションを指定すると、CD などのローカル媒体または遠隔媒体が有効なインストール媒体であるかどうかを確認できます。-c オプションは、指定した媒体に関する有用な情報を返します。この場合の luupgrade の構文は次のとおりです。

```
luupgrade -c -s path_to_medium [ -l error_log ] [ -o outfile ]
```

最初のオプション -c は、これから実行する処理 (インストール媒体のチェック) を示します。上記の -s オプションの働きは次のとおりです。

-s CD-ROM などのインストール媒体のパス名を指定します。

path_to_medium

このセクションでは、luupgrade のプロファイルで使用できる Solaris JumpStart キーワードを指定します。-j オプションに -u (アップグレード) オプション、または -f (フラッシュ) オプションを組み合わせて使用します。-u オプションの場合、必須のキーワードはありません。-f オプションは、完全なフラッシュアーカイブの場合は値を `install_type: flash_install` に、差分フラッシュアーカイブの場合は値を `flash_update` に指定する必要があります。-j オプションと -f オプションを合わせて使用する場合は、-a (アクティブな位置) オプションを指定するか、またはプロファイル中で `archive_location` キーワードを指定する必要があります。

`archive_location` キーワードは -J オプションに有効な唯一の引数です。

以下のキーワードオプションは、-u オプションおよび -f オプションで使用するプロファイルに使用することがあります。

`cluster`

システムに追加するソフトウェアグループを指定します。

`geo`

システムにインストール、または追加する地域ロケールまたはロケールを指定します。使用可能な値の一覧については、『Solaris のインストール』を参照してください。

`isa_bits`

64 ビットまたは 32 ビットのどちらのパッケージをインストールするかを指定します。有効な値は 64 と 32 です。

`locale`

システムにインストールまたは追加するロケールパッケージを指定します。使用可能な値の一覧については、『Solaris のインストール』を参照してください。

`package`

システムに追加する、またはシステムから削除するパッケージを指定します。

次のキーワードは、luupgrade で使用するプロファイルに使用してはなりません。

- `boot_device`
- `dontuse`
- `fdisk`
- `filesys`
- `layout_constraint`

luupgrade(1M)

- noreboot
- partitioning
- root_device
- usedisk

すべての JumpStart プロファイルキーワードの説明、および JumpStart プロファイルを作成する手順については『Solaris のインストール』を参照してください。

使用例 例 1 パッケージを削除して追加する

次の例は、ブート環境から複数のパッケージを削除して、追加し直します。

```
# luupgrade -P -n second_disk SUNWabc SUNWdef SUNWghi
```

同じパッケージを追加します。

```
# luupgrade -p -n second_disk -s /net/installmachine/export/packages \
SUNWabc SUNWdef SUNWghi
```

次の例では、上記のコマンドに `-o` オプションを追加することにより、引数を直接 `pkgadd` に渡しています。

```
# luupgrade -p -n second_disk -s /net/installmachine/export/packages \
-o "-r /net/testmachine/export/responses" SUNWabc SUNWdef SUNWghi
```

このコマンドのオプションについては、`pkgadd(1M)` のマニュアルページを参照してください。

例 2 結合イメージを使用して、新しい OS にアップグレードする

次の例は、ブート環境上のオペレーティング環境をアップグレードします。ソースイメージは、リモートディスクまたは DVD 上に結合イメージとしてあります。

```
# luupgrade -u -n second_disk \
-s /net/installmachine/export/solarisX/OS_image
```

上記のコマンドに続けて次のコマンドを実行すると、アップグレードした BE がアクティブになります。

```
# luactivate second_disk
```

次のリポート時に、`second_disk` が現在のブート環境になります。`luactivate(1M)` のマニュアルページを参照してください。

例 3 複数の CD を使用して、新しい OS にアップグレードする

次の例は、これまでに紹介した例の応用です。OS のアップグレードは CD 2 枚に格納されています。SPARC マシン上でアップグレードを開始するには、次のコマンドを入力します。

```
# luupgrade -u -n second_disk -s /dev/cdrom/cdrom0/s0
```

x86 マシンでは、`-s` の引数 `s0` を `s2` で置き換えます。

1 枚目の CD の内容の処理が完了したら、ドライブに 2 枚目の CD を挿入し、次のコマンドを入力します。

例 3 複数の CD を使用して、新しい OS にアップグレードする (続き)

```
# luupgrade -i -n second_disk -s /dev/cdrom/cdrom0 \
-O "-nodisplay -noconsole"
```

継続してインストールするときは、`-u` オプションよりも `-i` を使用してください。3 枚目以降も同様に `-i` オプションを使用してください。上記の `-o` オプションは、`installer(1M)` に渡されます。これらのオプションを省略すると、CD の挿入と読み取りが完了した時点でグラフィカルユーザーインターフェースが起動します。`-o` オプションについては `installer(1M)` を参照してください。

複数の CD からアップグレードを行う場合、すべての CD に対して `luupgrade` コマンドを入力し、その実行が完了した時点でアップグレードが終了します。各 CD からパッケージをインストールした後、次のようなメッセージが出力される場合があります。

```
WARNING: <num> packages must be installed on boot environment <disk_device>.
```

これは、上記の例のように、複数の CD からパッケージのインストールが必要であるということを示すものです。全パッケージのインストールが完了していないと、アップグレードの完了した BE を `luactivate` を使用して有効化(ブート先として指定)することはできません。

例 4 JumpStart プロファイルを使用したアップグレード

次のコマンド例は、プロファイル `/home2/profiles/test.profile` をテストするのに `-D` オプションを使用しています。

```
# luupgrade -u -n second_disk \
-s /net/installmachine/export/solarisX/OS_image \
-j /home2/profiles/test.profile -D
```

このコマンドの結果が満足できる内容の場合は、上記のコマンドから `-D` オプションを省略して、アップグレードを実行します。

例 5 Solaris フラッシュアーカイブから新しい OS をインストールする

次の例は、Solaris フラッシュアーカイブを使ってブート環境上のオペレーティング環境をアップグレードします。`-J` オプションで指定するファイルは、フラッシュインストールを指定する JumpStart プロファイルです。

```
# luupgrade -f -n second_disk \
-s /net/installmachine/export/solarisX/OS_image \
-J "archive_location http://example.com/myflash.flar"
```

次のコマンドが上記のコマンド例と異なるのは、`-j` オプションが `-J` オプションに置き換わっている点だけです。どちらかのコマンドに `-D` オプションを追加すると、実際にフラッシュインストールを実行する前にプロファイルをテストできます。

```
# luupgrade -f -n second_disk \
-s /net/installmachine/export/solarisX/OS_image \
-j /net/example/flash_archives/flash_gordon
```

luupgrade(1M)

例 5 Solaris フラッシュアーカイブから新しい OS をインストールする (続き)

先の 2 つのコマンドは、完全フラッシュインストール、または差分フラッシュインストールとして動作します。差分フラッシュインストールか完全フラッシュインストールかは、プロファイル中の `install_type` キーワードの値で決まります。前述の「JumpStart プロファイルキーワード」を参照してください。

例 6 パッケージ情報を取得する

次の例は、`pkgchk` に `-v` オプションを指定して、`SUNWluu` と `SUNWlur` の 2 つのパッケージに対して `pkgchk` を実行する例です。

```
# luupgrade -C -n second_disk -O "-v" SUNWluu SUNWlur
```

次のコマンドは、上記の 2 つのパッケージに対して `pkginfo` を実行します。

```
# luupgrade -I -n second_disk -O "-v" SUNWluu SUNWlur
```

どちらのコマンドの場合でも、パッケージ名を省略すると、指定した BE の全パッケージの情報が返されます。これらのコマンドのオプションについては、`pkgchk(1M)` のマニュアルページと `pkginfo(1)` のマニュアルページを参照してください。

終了ステータス 次の終了ステータスが返されます。

```
0      正常終了
>0     エラーが発生した
```

ファイル `/etc/lutab`
システム上にある BE のリスト
`/usr/share/lib/xml/dtd/lu_cli.dtd.<num>`
Live Upgrade の DTD (「常に指定可能なオプション」の `-x` オプションを参照)

属性 次の属性については、`attributes(5)` のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWluu

関連項目 `installer(1M)`, `lu(1M)`, `luactivate(1M)`, `lucancel(1M)`, `lucompare(1M)`, `lucreate(1M)`, `lucurr(1M)`, `ludelete(1M)`, `ludesc(1M)`, `lufslis(1M)`, `lumake(1M)`, `lumount(1M)`, `lurename(1M)`, `lustatus(1M)`, `lutab(4)`, `attributes(5)`, `live_upgrade(5)`

警告 パッケージまたはパッチを追加する場合は (`-p`、`-P`、`-t`、`-T` のいずれかを指定)、『SVR4 Advanced Packaging Guidelines』および『Solaris 9 インストールガイド』の付録 C に記載されているガイドラインに準拠したパッケージまたはパッチを指定してください。これらに準拠したパッケージまたはパッチは、`pkgadd(1M)` または

luupgrade(1M)

patchadd(1M) の -R オプション (説明については各ユーティリティのマニュアルページを参照) に準拠することになります。ほぼすべての Sun のパッケージおよびパッチがこのガイドラインに準拠していますが、Sun 以外のベンダーのパッケージはガイドラインに準拠していない場合があります。また、Sun の古いパッケージおよびパッチの中には、-R オプションに対応していないものがあります。そのようなパッケージまたはパッチを見つけた場合は、Sun に報告してください。準拠していないパッケージまたはパッチを指定すると、luupgrade のパッケージまたはパッチ追加ソフトウェアを実行したときに問題が発生したり、現在の BE が変更されてしまう可能性があります。

注意事項 Live Upgrade は、配布中のリリース、および最大 3 つのリリースをさかのぼってサポートします。たとえば、Solaris 9 版 (Solaris 9 Upgrade 版を含む) の Live Upgrade バージョンを入手した場合、Solaris 9 に加えて Solaris バージョン 2.6、Solaris 7、および Solaris 8 版をサポートします。Solaris 2.6 より前のバージョンの Solaris をサポートする Live Upgrade バージョンは存在しません。

Solaris Live Upgrade を正しく操作するためには、指定された OS バージョン用の制限付きパッチリビジョンのセットがインストールされている必要があります。Live Upgrade をインストールする、または実行する前に、制限付きパッチリビジョンのセットをインストールする必要があります。<http://sunsolve.sun.com> を参照して、最新のパッチリストがあることを確認してください。SunSolve Web サイトで infodoc 72099 を検索してください。

luxadm(1M)

名前	luxadm – Sun StorEdge A5000、RSM、SPARCstorage Array (SSA)サブシステムおよび各 Fiber Channel Arbitrated Loop (FC_AL) デバイスの管理プログラム
形式	luxadm [<i>options...</i>] <i>subcommand</i> [<i>options...</i>] <i>enclosure</i> [<i>,dev</i>] <i>pathname...</i>
機能説明	<p>luxadm プログラムは、Sun StorEdge A5000、RSM、SPARCstorage Array サブシステムおよび各 FC_AL デバイスの管理コマンドです。luxadm は、コマンド行に指定される引数やオプションに応じて、さまざまな制御処理や照会処理を実行します。</p> <p>コマンド行にはサブコマンドの指定が必要です。コマンド行には、サブコマンドに応じてオプションやパラメタを指定することができます。オプションには、通常 1 つ以上の格納装置名またはパス名を指定します。指定する文字数は、サブコマンドを一意に識別するのに必要なだけでかまいません。</p> <p>サブコマンドが作用するデバイスをパス名で指定します。Sun StorEdge A5000 サブシステムでは、パス名の代わりに、デバイスまたはデバイスに対するポートのワールドワイド名 (WWN) を入力することによってディスクデバイスまたは格納装置サービスコントローラを指定することができます。また、Sun StorEdge A5000 の格納装置名および格納装置内の特定のデバイスを示す識別子 (オプション) を入力することによってデバイスを指定することもできます。各 FC_AL デバイスを指定するには、デバイスの WWN またはポートを入力します。</p>
パス名	<p>デバイスまたはコントローラを、完全な物理パス名または完全な論理パス名で指定します。</p> <p>Sun StorEdge A5000 に対するデバイスの一般的な物理パス名を以下に示します。</p> <pre>/devices/sbus@1f,0/SUNW,socal@1,0/sf@0,0/ssd@w2200002037000f96, 0:a,raw</pre> <p>または</p> <pre>/devices/io-unit@f,e0200000/sbi@0,0/SUNW,socal@2,0/sf@0,0/ssd@34, 0:a,raw</pre> <p>システム上のすべての Sun StorEdge A5000 IB (Interface Board) に対しては、物理パスへの論理リンクが /dev/es ディレクトリに格納されます。論理リンクの例として、/dev/es/ses0 などがあります。</p> <p>FC_AL デバイスや Sun StorEdge A5000 サブシステム IB を選択するために、パス名の代わりに WWN を使用することもできます。WWN は、デバイスを使用するためのポートまたはデバイスそのものを表す 16 桁の 16 進数の値です。一般的な WWN の値を次に示します。</p> <pre>2200002037000f96</pre> <p>WWN の形式については、「注意事項」を参照してください。</p> <p>SPARCstorage Array コントローラに対する一般的な物理パス名を次に示します。</p>

```
/devices/. . . /. . . /SUNW,soc@3,0/SUNW,pln@
xxxxxx,xxxxxx:ctlr
```

RSM の一般的な物理パス名を以下に示します。

```
/devices/sbus@1f,0/QLGC,isp@1,10000:devctl
```

SPARCstorage Array または RSM コントローラの指定を簡単にするために、*cN* の形式による論理パス名が使用可能になっています。Nには、論理コントローラ番号を指定します。luxadm は、SPARCstorage Array または RSM コントローラに接続されているディスクの `/dev/rdisk` ディレクトリの中にあるエントリを見つけるために、*cN* 名を使用します。次に、`/dev/rdisk` エントリは SPARCstorage Array または RSM コントローラの物理名を調べるために使用されます。

SPARCstorage Array ディスクに対する一般的なパス名を次に示します。

```
/devices/. . . /. . . /SUNW,soc@3,0/SUNW,
pln@xxxxxx,xxxxxx/ssd@0,0:c,raw
```

次に、一般的な論理パス名を示します。

```
/dev/rdisk/c1t0d0s2
```

RSM の一般的なパス名を以下に示します。

```
/devices/sbus@1f,0/QLGC,isp@1,10000/sd@8,0:c,raw
```

次に、一般的な論理パス名を示します。

```
/dev/rdisk/c2t8d0s2
```

各 FC_AL デバイスの一般的なパス名を示します。

```
/devices/sbus@3.0/SUNW,socal@d,10000/sf@0,0/ssd@w2200002037049fc3,0:a,raw
```

次に、一般的な論理パス名を示します。

```
/dev/rdisk/c1t0d0s2
```

格納装置

Sun StorEdge A5000 では、デバイスは格納装置名とスロット名で特定することができます。

```
box_name[, fslot_number]
box_name[, rslot_number]
```

luxadm(1M)

	<p><i>box_name</i> は、Sun StorEdge A5000 の格納装置名で、<i>enclosure_name</i> サブコマンドで指定します。オプションの <i>slot_number</i> パラメタを指定しないと、<i>box_name</i> には、Sun StorEdge A5000 サブシステム IB が指定されます。</p> <p><i>f</i> または <i>r</i> は、Sun StorEdge A5000 の格納装置の前面のスロット(<i>f</i>) か、背面のスロット (<i>r</i>) かを指定します。</p> <p><i>slot_number</i> は、Sun StorEdge A5000 の格納装置のデバイスのスロット番号を指定します。指定することができる番号の範囲は 0～6 または 0～10 です。</p> <p>ディスクおよびサブシステムの論理名については、<i>disks(1M)</i> および <i>devlinks(1M)</i> を参照してください。</p>
オプション	<p>次のオプションはすべてのサブコマンドに対して指定することができます。</p> <p>-e エキスパート (Expert) モードです。このオプションは初心者の方にはお勧めしません。</p> <p>-v 詳細表示 (Verbose) モードです。</p> <p>特定のサブコマンドだけに指定するオプションについては、「使用方法」の各サブコマンドの説明を参照してください。</p>
オペランド	<p>次のオペランドを指定することができます。</p> <p><i>enclosure</i> Sun StorEdge A5000 の <i>box_name</i> を指定します。</p> <p><i>fibre_channel_HBA_port</i> ホストのコントローラポートへのパスを指定します。一般的なパスは、下記のとおりです。</p> <pre>/devices/pci@8,600000/pci@1/SUNW,qlc@4/fp@0,0:devctl</pre> <p><i>pathname</i> Sun StorEdge A5000 IB、SPARCstorage Array、RSM コントローラの名前 (<i>cN</i>)、またはディスクデバイスの論理パスまたは物理パスを指定します。<i>pathname</i> は、Sun StorEdge A5000 IB、Sun StorEdge A5000 ディスク、または各 FC_AL デバイスの WWN でも指定することができます。</p>
使用方法	
サブコマンド	<pre>display enclosure[<i>dev</i>]... <i>pathname</i> ... display -p <i>pathname</i> ... display -r enclosure[<i>dev</i>]... <i>pathname</i> ... display -v enclosure[<i>dev</i>]... <i>pathname</i> ...</pre> <p>格納装置またはデバイス固有のデータを表示します。</p> <p>サブシステムのデータは、格納装置の環境検知情報、およびディスクの状態を含む、すべてのサブシステムのデバイスの状態で構成されます。</p> <p>ディスクデータは、照会、容量、および設定情報で構成されます。</p>

-p
pathname で指定されたデバイスまたはサブシステムの性能情報を表示します。
 このオプションは、性能情報を保存するサブシステムに対してのみ指定することができます。

-r
pathname で指定された FC_AL デバイスのエラー情報を表示します。また、パス名が Sun StorEdge A5000 の場合は、そのループ上のすべてのデバイスのエラー情報を表示します。**-r** オプションは Sun StorEdge A5000 サブシステムまたは各 FC_AL デバイスに対してのみ指定することができます。

-v
 モード検知データを含む、詳細表示モードで表示します。

download [**-s**] [**-w** *WWN*] [**-f** *filename_path*] *enclosure...*

prom イメージを、*filename_path* で指定された Sun StorEdge A5000 サブシステムのインタフェースボード装置か、*enclosure* または *pathname* で指定された SPARCstorage Array コントローラにダウンロードします。SPARCstorage Array は、ダウンロードコードを使用するためにリセットする必要があります。

Sun StorEdge A5000 のダウンロードが終了すると、Sun StorEdge A5000 はリセットされ、ダウンロードコードが実行されます。ファイル名が指定されていない場合は、デフォルトの *prom* イメージが使用されます。SPARCstorage Array コントローラ用のデフォルトの *prom* イメージは、
 /usr/lib/firmware/ssa/ssafirmware ディレクトリに格納されます。Sun StorEdge A5000 用のデフォルトの *prom* イメージは
 /usr/lib/locale/C/LC_MESSAGES ディレクトリに *ibfirmware* というファイル名で格納されます。

-s
 保存オプションです。**-s** オプションはダウンロードしたファームウェアを FEPROM に保存します。**-s** を省略すると、ダウンロードしたファームウェアは保存されず、電源の再投入後に消えてしまいます。**-s** オプションは SPARCstorage Array コントローラには指定することができません。SPARCstorage Array コントローラは常に FEPROM にファームウェアをダウンロードするため、**-s** オプションは SPARCstorage Array コントローラには適用されません。**-s** オプションを使用すると、ダウンロードサブコマンドがサブシステム上の FEPROM を変更するため、**-s** オプションの使用には注意が必要です。

-w *WWN*
 SPARCstorage Array コントローラのワールドワイド名を変更します。*WWN* は、先頭が 0 で始まる 12 桁の 16 進数です。**-w** オプションは、SPARCstorage Array に対してのみ指定することができます。新しい SPARCstorage Array コントローラのイメージは、8 バイトのワールドワイド名のうち、下位の 6 バイトが *WWN* で指定された値に変更されたものになります。

enclosure_name new_name enclosure | pathname

enclosure または *pathname* で指定された単数または複数の格納装置名を変更します。新しい名前は 16 文字以下の英数字で *new_name* に指定します。このサブコマンドは Sun StorEdge A5000 に対してだけ指定することができます。

```
fc_s_download [-F] [-f fcode-file]
```

fcode-file で指定されたファイルに含まれる *fcode* をすべての FC/S SBus カードにダウンロードします。このコマンドは対話型で、*fcode* をダウンロードする前にユーザーに対して確認を促します。

fc_s_download は、シングルユーザーモードでだけ使用してください。入出力操作が行われているホストアダプタに対して、そのアダプタを更新する目的で *fc_s_download* を指定すると、アダプタのリセットの原因となります。新しく更新した FCode は、システムを再起動したときに実行され、表示できるようになります。

-f fcode-file

-f fcode-file オプションを省略すると、各 FC/S SBus カードの中の現在のバージョンの *fcode* が表示されます。

-F

オプションを指定すると、*fcode* は強制的にダウンロードされますが、ダウンロードの前にユーザーに対して確認を促します。現在のバージョンのオペレーションシステムと共にリリースされた FC/S SBus カード *fcode* のバージョンは `/usr/lib/firmware/fc_s` ディレクトリに *fc_s_fcode* の名前で格納されています。

```
fcal_s_download [-f fcode-file]
```

fcode-file に指定されたファイルに含まれる *fcode* をすべての FC100/S SBus カードにダウンロードします。このコマンドは対話型で、*fcode* をダウンロードする前にユーザーに対して確認を促します。

fcal_s_download は、シングルユーザーモードでだけ使用してください。入出力操作があるホストアダプタに対して、そのアダプタを更新する目的で *fcal_s_download* を指定すると、アダプタのリセットの原因となります。新しく更新した *fcode* は、システムを再起動したときに実行され、表示できるようになります。

-f fcode-file

-f オプションを省略すると、各 FC100/S SBus カード中の現在のバージョンの *fcode* が表示されます。現在のバージョンのオペレーションシステムと共にリリースされた FC100/S SBus カード *fcode* のバージョンは `/usr/lib/firmware/fc_s` ディレクトリに *fcal_s_fcode* の名前で保存されています。

```
fcode_download [-p] [-d dir-name]
```

インストール済みの FC/S、FC100/S、FC100/P、または FC100/2P ホストバスアダプタカードを検出し、*dir-name* の *fcode* を適切なカードにダウンロードします。このコマンドは対話型で、各ファイルタイプに対して適切なカードを決定します。各デバイスに *fcode* をダウンロードする前にユーザーに対して確認を促します。

`fcode_download` は、シングルユーザーモードで `fcode` だけを読み込むときに使用します。入出力操作があるホストアダプタに対して、そのアダプタを更新する目的で `fcode_download` を指定すると、アダプタのリセットの原因となります。新しく更新した `fcode` は、システムを再起動したときに実行され、表示できるようになります。

`-d dir-name`

`dir-name` ディレクトリに格納されている `fcode` ファイルを適切なアダプタカードにダウンロードします。 `-d` オプションを省略すると、デフォルトの `/usr/lib/firmware/fc_s` ディレクトリが使用されます。

`-p`

各カードに読み込まれている現在のバージョンの `fcode` を表示します。ダウンロードは行われません。

`inquiry enclosure[,dev]... | pathname...`

`enclosure` または `pathname` で指定されたデバイスに対する照会情報を表示します。

`insert_device [enclosure,dev ...]`

このコマンドを使用して、電源を入れたまま 1 つまたは複数のデバイスを追加することができます。ホットプラグ操作中の制限に関しては「注意事項」を参照してください。このサブコマンドは Sun StorEdge A5000、RSM および各 FC_AL デバイスに対してのみ指定することができます。Sun StorEdge A5000 に対して複数の格納装置が指定された場合は、複数のバスに対して同時に追加が行われます。このサブコマンドの引数を省略すると、すべての格納装置または各 FC_AL デバイスが追加されます。RSM に対して指定することができるコントローラは 1 つだけです。Sun StorEdge A5000 に対しては、このサブコマンドは、電源を入れたまま 1 つまたは複数のデバイスを追加するすべての過程をユーザーとの対話によって行います。複数のディスクが指定された場合は、それらのディスクが正しいかどうかの確認が行われ、ユーザーは継続するか中止するかを選択することができます。その後、ディスクまたは格納装置の追加を実行するかどうかの確認が行われ、それらのデバイスの論理バス名が作成および表示されます。

RSM に対しては、以下の手続きが行われます。

- バスを休止させます。
- デバイスが安全に追加できることをユーザーに通知します。
- デバイスが追加されたことを確認するメッセージを表示します。
- バスを休止解除します。
- 新規デバイスに対する論理デバイス名を作成します。

`led enclosure,dev... | pathname...`

`enclosure` または `pathname` で指定されたディスクに対応している LED の現在の状態を表示します。このサブコマンドはこの機能を持つサブシステムに対してのみ指定することができます。

`led_blink enclosure,dev... | pathname...`

`enclosure` または `pathname` で指定されたディスクに対応している LED の点滅を開始するようサブシステムに指示します。このサブコマンドはこの機能を持つサブシステムに対してのみ指定することができます。

luxadm(1M)

`led_off enclosure,dev... | pathname...`

`enclosure` または `pathname` で指定されたディスクに対応している LED を消灯させるようサブシステムに指示します。Sun StorEdge A5000 サブシステムでは、LED の消灯や点滅の停止ができる状態ときかない状態があります。『Sun StorEdge A5000 設置・サービスマニュアル』(805-4111) を参照してください。このサブコマンドはこの機能を持つサブシステムに対してのみ指定することができます。

`led_on pathname...`

`pathname` で指定されたディスクに対応している LED を点灯させるようサブシステムに指示します。このサブコマンドはこの機能を持つサブシステムに対してだけ指定することができます。

`power_off [-F] enclosure[,dev]... | pathname...`

`power_off pathname [enclosure-port] ... | controller tray-number`

Sun StorEdge A5000 に対してこのオプションを指定した場合は、Sun StorEdge A5000 サブシステムが省電力モードに切り替わります。Sun StorEdge A5000 ドライブは、省電力モードでは使用することができません。SPARCstorage Array 中の格納装置サービスカードに対してこのオプションを指定した場合は、RSM トレーの電源が切れます。Sun StorEdge A5000 中のドライブに対してこのオプションを指定した場合は、そのドライブがドライブ off/unmated モードに設定されます。ドライブ off/unmated 状態では、ドライブが停止し、バイパスモードになります。

-F

強制オプションは Sun StorEdge A5000 のみに適用されます。このオプションを指定すると、luxadm は、1 つまたは複数のデバイスを、それらがホストによって使用されていても電源切断しようとします。

警告 : 現在使用されているデータを含むデバイスの電源を切断すると、予想可能な結果を引き起こします。デバイスの電源を切断する際は、まず、通常の方法を (-F を指定せずに) 試してください。このオプションは、通常の確認を無効にすることによる結果を理解した上で使用してください。

`power_on enclosure [,dev]..`

Sun StorEdge A5000 に対してこのオプションを指定した場合は、Sun StorEdge A5000 サブシステムが省電力モードから抜けます。Sun StorEdge A5000 ドライブは、省電力モードでは使用することができません。SPARCstorage Array RSM トレーの電源をプログラムによって投入する方法はありません。Sun StorEdge A5000 中のドライブに対してこのオプションを指定した場合は、そのドライブが通常の起動状態に設定されます。

`probe [-p]`

接続された Sun StorEdge A5000 サブシステムおよび各 FC_AL デバイスすべてに関する情報を検索して表示します。この情報には、論理パス名、WWN このサブコマンドは、同じ格納装置名を持つ異なる Sun StorEdge A5000 を見つけると、ユーザーに対して警告を發します。

-p

物理パス名を表示に含めます。

`qlgc_s_download [-f fcode-file]`

fcode-file ファイルに格納されている *fcode* をすべての FC100/P、FC100/2P PCI ホストアダプタカードにダウンロードします。このコマンドは対話型で、*fcode* を各ドライブにダウンロードする前にユーザーに対して確認を促します。

`qlgc_s_download` は、シングルユーザーモードでだけ使用してください。入出力操作があるホストアダプタに対して、そのアダプタを更新する目的で

`fcsl_s_download` を指定すると、アダプタのリセットの原因となります。新しく更新した *fcode* は、システムを再起動したときに実行され、表示できるようになります。

`-f fcode-file`

`-f` オプションを省略すると、各 FC100/P、FC100/2P PCI カードの、現在のバージョンの *fcode* が表示されます。

`release pathname`

指定されたディスクの予約を解除します。パス名は、ディスクの物理または論理パス名でなければなりません。SPARCstorage Array コントローラのパス名を指定した場合、SPARCstorage Array 中のすべてのディスクの予約が解除されます。

このサブコマンドは、履歴および診断目的以外では使用しないでください。

`remove_device [-F] enclosure[,dev]... | pathname ...`

このコマンドを使用して、電源を入れたまま 1 つまたは複数のデバイスを削除することができます。このサブコマンドはすべての格納装置を削除する場合にも指定することができます。このサブコマンドは Sun StorEdge A5000、RSM、および各 FC_AL デバイスに対して指定することができます。ホットプラグ操作中の制限に関しては「注意事項」を参照してください。Sun StorEdge A5000 および各 FC_AL デバイスに対しては、このサブコマンドは、電源を入れたまま 1 つまたは複数のデバイスを削除するすべての過程をユーザーとの対話によって行います。複数のディスクが指定された場合は、それらのディスクが正しいかどうかの確認が行われ、ユーザーは継続するか中止するかを選択することができます。その後で、ディスクまたは格納装置の削除を実行するかどうかの確認が行われ、それらのデバイスの論理パス名が作成および表示されます。

RSM に対しては、以下の手続きが行われます。

- デバイスをオフラインにします。
- バスを休止させます。
- デバイスが安全に削除できることをユーザーに通知します。
- デバイスが削除されたことを確認するメッセージを表示します。
- バスを休止解除します。
- (取り外された) デバイスをオンラインに戻します。
- 削除されたデバイスの論理デバイス名を削除します。

`-F`

1 つまたは複数のデバイスに対してホットプラグを適用するように `luxadm` に指示します。これらのデバイスがホストで使用されている場合 (*busy* または *reserved* の場合) にも適用されます。ホットプラグ オペレーションが強制的に実行されます。

警告: 現在使用されているデータが格納されているデバイスを取り外すと、予期しない結果が発生します。通常は、`-F` を指定しないでホットプラグを適用します。`-F` は、通常のホットプラグ検査を無効にしたときの結果がわかっているときにだけ使用します。

`replace_device [-F] pathname`

このコマンドは RSM に対してのみ指定することができます。ホットプラグ操作中の制限に関しては「注意事項」を参照してください。このサブコマンドは、電源を入れたままデバイスを交換するすべての過程をユーザーとの対話によって行います。

RSM に対しては、以下の手続きが行われます。

- デバイスをオフラインにします。
- バスを休止させます。
- デバイスが安全に削除できることをユーザーに通知します。
- デバイスが削除されたことを確認するメッセージを表示します。
- バスを休止解除します。
- デバイスをオンラインに戻します。

`-F`

1 つまたは複数のデバイスに対してホットプラグを適用するように `luxadm` に指示します。これらのデバイスがホストで使用されている場合 (`busy` または `reserved` の場合) にも適用されます。ホットプラグオペレーションが強制的に実行されます。

警告: 現在使用されているデータが格納されているデバイスを取り外すと、予期しない結果が発生します。通常は、`-F` を指定しないでホットプラグを適用します。`-F` は、通常のホットプラグ検査を無効にしたときの結果がわかっているときにだけ使用します。

`reserve pathname`

指定されたディスクを、ホストによる排他的利用のために予約します。パス名は、ディスクの物理または論理パス名でなければなりません。`pathname` が SPARCstorage Array コントローラのパス名の場合、SPARCstorage Array 中のすべてのディスクが予約されます。

このサブコマンドは、履歴および診断目的以外では使用しないでください。

`set_boot_dev [-y] pathname`

システム PROM 中のブートデバイス変数を、`pathname` で指定された物理デバイス名に設定します。指定する物理デバイス名は、ブロック型特殊デバイス、または起動ファイルシステムのマウント先のディレクトリのパス名です。通常、このコマンドは、PROM 中のデフォルト起動デバイスを設定するためにユーザーに対して確認を促します。`-y` オプションを指定すると、ユーザーに対して確認を要求しません。

`start [-t tray-number] pathname ...`

指定されたディスクを起動します。`pathname` に SPARCstorage Array コントローラを指定した場合は、SPARCstorage Array 中のすべてのディスクを起動します。

SPARCstorage
Array のサブコマ
ンド

-t tray-number で指定された番号のトレイ中のすべてのディスクを起動します。*pathname* には SPARCstorage Array コントローラを指定する必要があります。

stop [-t tray-number] *pathname* ...
指定されたディスクを停止します。*pathname* に SPARCstorage Array コントローラを指定した場合は、SPARCstorage Array 中のすべてのディスクを停止します。

-t tray-number で指定された番号のトレイ中のすべてのディスクを停止します。*pathname* には SPARCstorage Array コントローラを指定する必要があります。

fast_write [-s] -c *pathname*
fast_write [-s] -d *pathname*
fast_write [-s] -e *pathname*
SPARCstorage Array の書き込みの性能を向上させるために NVRAM の利用を有効・無効にします。*pathname* には SPARCstorage Array コントローラか、個々のディスクを指定します。

-s 電源を切断する前に SPARCstorage Array が変更を保存するように指定します。

-c 同期書き込みの場合にのみ高速書き込みを有効にします。

-d 高速書き込みを無効にします。

-e 高速書き込みを有効にします。

nvr_data *pathname*
指定されたディスクの NVRAM 中にある高速書き込みデータの量を表示します。このコマンドは 1 つのディスクに対してのみ使用することができます。

perf_statistics -d *pathname*
perf_statistics -e *pathname*
指定された SPARCstorage Array コントローラに関する性能統計情報を保存するかどうかを指定します。性能統計情報の保存は、display -p サブコマンドを使用する前に有効になっている必要があります。このサブコマンドは、SPARCstorage Array コントローラに対してのみ指定することができます。

-d 性能統計情報の保存を無効にします。

-e 性能統計情報の保存を有効にします。

purge *pathname*
あるディスクの NVRAM から高速書き込みデータをディスクに書き出します。コントローラを指定した場合は、そのコントローラ中のすべてのディスクが対象となります。このオプションの使用時には注意が必要で、通常はドライブに障害が発生した場合にだけ使用します。

luxadm(1M)

格納装置サービス
カードのサブコマ
ンド

**Sun StorEdge
A5000** および各
FC_AL デバイスの
エキスパートモー
ドコマンド

sync_cache pathname

指定されたディスクの NVRAM から、未処理の書き込みすべてをディスクに書き出します。pathname にコントローラを指定した場合は、その SPARCstorage Array サブシステム中のすべてのディスクが操作の対象となります。

env_display および *alarm** サブコマンドは、SPARCstorage Array の RSM トレーの中にある格納装置サービスカード (SES) のみが対象となります。RSM トレーは、SES デバイスの論理パス名または物理パス名で指定するか、コントローラにトレイ番号を続けて指定します。コントローラは cN または SSA コントローラの物理パス名で指定します。

alarm pathname | controller tray_number

警告音の現在の状態を表示します。

alarm_off pathname | controllerr tray_number

指定された RSM トレーの警告音を無効にします。

alarm_on pathname | controller tray_number

指定された RSM トレーの警告音を有効にします。

alarm_set controller-pathname | controller tray_number [seconds]

警告音の長さを秒単位で指定します。

env_display pathname | controller tray_number

指定された装置の環境情報を表示します。

以下のサブコマンドは、経験が豊富なユーザーのみが使用することができます。また、Sun StorEdge A5000 およびファイバチャネルループのみを対象にしています。これらのコマンドは、Sun StorEdge A5000 サブシステムとファイバチャネルループの知識が豊富なユーザーだけが使用することができます。

バスを操作するエキスパートサブコマンドにディスクを指定した場合は、そのサブコマンドは、指定したディスクに接続されたバスを操作します。

-e bypass [-ab] enclosure,dev

-e bypass -f enclosure

エンクロージャサービスコントローラに対して、指定したポートおよびデバイスの LRC (ループ冗長回路) をバイパス状態に設定するよう要求します。

このサブコマンドは、以下のオプションをサポートします。

-a

指定したデバイスのポート a をバイパスします。

-b

指定したデバイスのポート b をバイパスします。

-e dump_map fibre_channel_HBA_port

指定したファイバチャネルポート上にあるターゲットデバイスまたはホストバスアダプタの WWN データを表示します。指定したポートにターゲットデバイスが存在しない場合、エラーが返されます。

```
-e enable [-ab] enclosure,dev
-e enable -f enclosure
```

エンクローチャサービスコントローラに対して、指定したポートおよびデバイスの LRC (ループ冗長回路) を有効な状態に設定するよう要求します。

このサブコマンドは、以下のオプションをサポートします。

```
-a
  指定したデバイスのポート a を有効にします。
-b
  指定したデバイスのポート b を有効にします。
-e forcelip enclosure[, dev] ... | pathname ...
```

ループ初期化基本式 (LIP) の処理を使ってリンクを強制的に最初期化します。 enclosure または pathname には、ループ上のどのデバイスでも指定することができます。複数のループ構成に対して特定のパスを指定する場合は、パス名を使用します。

このコマンドは経験者向けのコマンドで、使用には注意が必要です。このコマンドは、ループ上のすべてのポートをリセットします。

```
-e rdls enclosure[, dev] ... | pathname ...
```

enclosure または pathname で指定されたデバイスを含むループにある、すべての使用可能なデバイスのリンクエラー状態情報を読み込み、表示します。

ホットプラグ操作中の制限に関しては「注意事項」を参照してください。以下のサブコマンドは、経験が豊富なユーザーのみが使用することができます。

```
-e bus_getstate pathname
  指定されたバスの状態を取得、表示します。
-e bus_quiesce pathname
  指定されたバスを休止します。
-e bus_reset pathname
  指定されたバスのみをリセットします。
-e bus_resetall pathname
  指定されたバスおよびすべてのデバイスをリセットします。
-e bus_unquiesce pathname
  指定されたバスを休止解除します。
-e dev_getstate pathname
  指定されたデバイスの状態を取得、表示します。
-e dev_reset pathname
  指定されたデバイスをリセットします。
-e offlinepathname
  指定されたデバイスをオフラインにします。
-e online pathname
  指定されたデバイスをオンラインにします。
```

その他のエキス
パートモードサブ
コマンド

luxadm(1M)

- 使用例**
- 例 1** システム上にあるすべての Sun StorEdge A5000 および FC_AL の表示
- システム上にあるすべての Sun StorEdge A5000 および FC_AL デバイスを検索、表示する例を次に示します。
- ```
example% luxadm probe
```
- 例 2** SSA の表示
- SSA を表示する例を次に示します。
- ```
example% luxadm display c1
```
- 例 3** SENA の表示
- Sun StorEdge A5000 を表示する例を次に示します。
- ```
example% luxadm display /dev/es/ses0
```
- 例 4** 2つのサブシステムの表示
- 格納装置名を使って2つのサブシステムを表示する例を次に示します。
- ```
example% luxadm display BOB system1
```
- 例 5** 最初のディスクに関する情報の表示
- BOB,f1 という名前の格納装置の前面にある最初のディスクに関する情報を表示する例を示します。前面のディスクを指定する場合には f を指定します。背面のディスクを指定する場合は r を指定します。
- ```
example% luxadm display BOB,f0
```
- 例 6** Sun StorEdge A5000 ディスク、格納装置、または各 FC\_AL デバイスに関する情報の表示
- ポートの WWN が 2200002037001246 の Sun StorEdge A5000 ディスク、格納装置、または各 FC\_AL デバイスに関する情報を表示する例を次に示します。
- ```
example% luxadm display 2200002037001246
```
- 例 7** サブコマンドとして一意に認識するための文字列
- サブコマンドとして一意に認識できるだけの長さの文字列を使用する例を次に示します。
- ```
example% luxadm disp BOB
```
- 例 8** エラー情報の表示
- 格納装置 BOB があるループに関するエラー情報を表示する例を次に示します。
- ```
example% luxadm display -r BOB
```

例 8 エラー情報の表示 (続き)**例 9** インタフェースボードへの新しいファームウェアのダウンロード

格納装置 BOB のインタフェースボードに新しいファームウェアをダウンロードする例を次に示します (ダウンロードするファイルはデフォルトパスで指定されています)。

```
example% luxadm download -s BOB
```

例 10 SCSI 照会コマンドからの情報の表示

システム上の個々のディスクから SCSI 照会コマンドからの情報を表示する例を次に示します。サブコマンドとして一意に認識できる長さの文字列だけが使用されます。

```
example% luxadm inq /dev/rdisk/c?t?d?s2
```

例 11 ホットプラグによる取り付け

BOB,f1 という名前の格納装置の前面の最初のスロットに新しいドライブをホットプラグで取り付ける例を次に示します。

```
example% luxadm insert_device BOB,f0
```

例 12 エキスパートサブコマンドの実行

エキスパートサブコマンドを実行する例を次に示します。このサブコマンドは、格納装置 BOB のあるループを強制的に初期化します。

```
example% luxadm -e forcelp BOB
```

例 13 エキスパートモードのホットプラグサブコマンドの使用

エキスパートモードのホットプラグサブコマンドを使用して SSA 上のディスクを削除する例を以下の示します。ホットプラグ操作中の制限に関しては「注意事項」を参照してください。最初の手順では、SCSI デバイスが 2 つ目の SCSI バスによってアクセスされないように、そのデバイスを予約します。

```
example# luxadm reserve /dev/rdisk/c1t8d0s2
```

例 14 ディスクをオフラインにする

次の 2 つの手順では、ディスクをオフラインにし、バスを休止します。

```
example# luxadm -e offline /dev/rdisk/c1t8d0s2
example# luxadm -e bus_quiesce /dev/rdisk/c1t8d0s2
```

例 15 バスの休止解除

ユーザーは、この時点でディスクを取り外し、バスを休止解除し、ディスクをオンラインに戻し、ディスクの予約を解除します。

```
example# luxadm -e bus_unquiesce /dev/rdisk/c1t8d0s2
example# luxadm -e online /dev/rdisk/c1t8d0s2
example# luxadm release /dev/rdisk/c1t8d0s2
```

luxadm(1M)

例 15 バスの休止解除 (続き)

環境 luxadm の実行に影響のある環境変数 LANG に関しては environ(5)を参照してください。

終了ステータス 次のいずれかの値が終了時に返されます。

0 正常終了

-1 異常終了

ファイル usr/lib/firmware/fc_s/fcal_s_fcode

usr/lib/firmware/fc_s/fc_s_fcode

usr/lib/firmware/ssa/ssafirmware

usr/lib/locale/C/LC_MESSAGES/ibfirmware

属性 以下の属性については、attributes(5) を参照してください。

usr/sbin

属性の種類	属性の値
使用条件	SUNWluxop

関連項目 devlinks(1M), disks(1M), ssaadm(1M), attributes(5), environ(5), ses(7D)

『Sun StorEdge A5000 設置・サービスマニュアル』(805-4111)

注意事項 Sun StorEdge A5000 に関するその他の情報については、『Sun StorEdge A5000 設置・サービスマニュアル』(805-4111)を参照してください。IEEE 拡張 WWW に関する情報については、『Tutorial for SCSI use of IEEE company_ID』(R. Snively 著)を参照してください。「関連項目」を参照してください。現在は、一部のデバイスドライバのみがホットプラグに対応しています。ホットプラグに対応していないディスクまたはバスにホットプラグが適用されると、以下のメッセージが表示されます。

```
luxadm: can't acquire "PATHNAME": No such file or directory
```

ルートファイルシステムまたは /usr ファイルシステムを含むバスやスワップデータを含むバスを休止する際は注意してください。そのようなバスを休止すると、デッドロックを引き起こす可能性があり、そのような場合は、システムの再起動が必要となります。

名前	m64config, SUNWm64_config – M64 グラフィックスアクセラレータの設定
形式	<pre> /usr/sbin/m64config [-defaults] [-depth 8 24] [-dev device-filename] [-file machine system] [-prconf] [-propt] [-res video-mode now try] [noconfirm nocheck] /usr/sbin/m64config [-prconf] [-propt] /usr/sbin/m64config [-help] [-res ?] </pre>
機能説明	<p>m64config は、M64 グラフィックスアクセラレータおよび M64 対応の X11 ウィンドウシステムのデフォルトの一部を設定します。</p> <p>形式の項に記された m64config の 1 番目の形式では、指定したオプションを OWconfig ファイルに保存します。これらのオプションは、次にウィンドウシステムをそのデバイスで実行するときに M64 デバイスを初期化するために使用されます。OWconfig ファイル内のオプションの更新は、異なるウィンドウセッションや再起動したシステムでも有効となります。</p> <p>-prconf、-propt、-help、-res ? オプションだけを起動する 2 番目と 3 番目の形式では、OWconfig ファイルは更新されません。また、3 番目の形式では、その他のオプションはすべて無視されます。</p> <p>オプションは、一度に 1 つの M64 デバイスに対してのみ指定することができます。複数の M64 デバイスに対してオプションを指定するには、m64config を複数回起動する必要があります。</p> <p>m64config で指定できるのは、M64 固有のオプションだけです。デフォルトの表示色数、デフォルトの画像表示形式クラスなどを指定する通常のウィンドウシステムのオプションは、openwin コマンド行のデバイス修飾子で指定してください (OpenWindows のマニュアルページの Xsun(1) を参照)。</p> <p>ユーザーは、更新する OWconfig ファイルを指定することもできます。デフォルトでは、/etc/openwin ディレクトリツリーにあるマシン固有のファイルが更新されます。別のファイルを指定するには、-file オプションを使用します。たとえば、/usr/openwin ディレクトリツリーにあるシステム共通の OWconfig ファイルを代わりに更新することができます。</p> <p>これらの標準 OWconfig ファイルのどちらもスーパーユーザーのみが書き込みを行えます。したがって、スーパーユーザーが所有する m64config プログラムは、setuid による root の権限で実行されます。</p>
オプション	<p>-defaults すべてのオプションの値をそれぞれのデフォルト値に戻します。</p> <p>-depth 8 24 表示色数を 8 または 24 ビット/ピクセルで指定します。変更を反映させるには、現在のウィンドウシステムのセッションからログアウトしてから再度ログインします。24 ビット/ピクセルでは、画面解像度を低くすることにより、ウィンドウシステムで TrueColor グラフィックスを実現できます。</p>

m64config(1M)

24 ビット/ピクセルで利用できる解像度の最大値は、PGX カードに搭載されているメモリー量に依存します。2 M バイトのメモリーを搭載した PGX カードで利用できる解像度の最大値は 800x600 です。4 M バイトのメモリーを搭載したカードで利用できる解像度の最大値は 1152x900 です。指定した解像度と色数の組み合わせに必要なメモリーが不足している場合、m64config はエラーメッセージを出力して終了します。

-dev *device-filename*

M64 特殊ファイルを指定します。デフォルトは /dev/fbs/m640 です。

-file *machine|system*

更新する OWconfig ファイルを指定します。machine を指定すると、/etc/openwin ディレクトリツリーにあるマシン固有の OWconfig ファイルが使用されます。system を指定すると、/usr/openwin ディレクトリツリーにある共通の OWconfig ファイルが使用されます。ファイルがない場合は、新たに生成されます。

-help

m64config コマンド行のオプションと機能の概要を一覧で表示します。

-prconf

M64 ハードウェア構成を表示します。次に表示例を示します。

```
--- Hardware Configuration for /dev/fbs/m640 ---
ASIC: version 0x41004754
DAC: version 0x0
PROM: version 0x0
Card possible resolutions: 640x480x60, 800x600x75, 1024x768x60
1024x768x70, 1024x768x75, 1280x1024x75, 1280x1024x76
1280x1024x60, 1152x900x66, 1152x900x76, 1280x1024x67
960x680x112S, 960x680x108S, 640x480x60i, 768x575x50i, 1280x800x76
1440x900x76, 1600x1000x66, 1600x1000x76, vga, svga, 1152, 1280
stereo, ntsc, pal
Monitor possible resolutions: 720x400x70, 720x400x88, 640x480x60
640x480x67, 640x480x72, 640x480x75, 800x600x56, 800x600x60
800x600x72, 800x600x75, 832x624x75, 1024x768x87, 1024x768x60
1024x768x70, 1024x768x75, 1280x1024x75, 1280x1024x76, 1152x900x66
1152x900x76, 1280x1024x67, 960x680x112S, vga, svga, 1152, 1280
stereo
Possible depths: 8, 24
Current resolution setting: 1280x1024x76
Current depth: 8
```

-propt

-file オプションで指定された OWconfig ファイルに書かれた M64 オプションの値のうち、-dev オプションで指定されたデバイスに対するものすべてを表示します。m64config の呼び出しが終了した後に、OWconfig ファイルに書き込まれるオプションの値を表示します。次に表示例を示します。

```
--- OpenWindows Configuration for /dev/fbs/m640 ---
OWconfig: machine
Video Mode: not set
```

```
-res video-mode [ now | try [ noconfirm | nocheck ] ]
```

指定した M64 デバイ스에 接続されている 모니터를 制御する 際 に 使われる 表示モードを 指定します。表示モードは あらかじめ 組み込まれています。表示モードの形式は *widthx-heightxrate* で、*width* はピクセル単位の画面幅、*height* はピクセル単位の画面の高さ、*rate* は画面を垂直方向に再描画する周期です。便宜上、*-res* にリフレッシュレートを指定する際は、値の直前に *x* の代わりに *@* を使用することができます。たとえば、1280x1024@76 のように指定することができます。

有効な表示モードのリストは、`m64config -res '?'` を実行することによって得ることができます。`?` は引用符で囲んで文字として扱う必要があります。すべてのビデオボードおよびモニターがすべての解像度に対応しているわけではありません。m64config は、ボードが対応していない解像度に設定することを許しません。モニターが対応していない解像度に設定しようとすると、確認を促すメッセージを表示します。

記号名

便宜上、上記の表示モードのいくつかには記号名が定義されています。*widthxheightxrate* の形式の代わりに、記号名を *-res* の引数として指定することができます。記号名 `none` は、ウィンドウシステムを実行すると、画面の解像度は現在デバイスにプログラムされている表示モードになることを意味します。

記号名	対応する表示モード
<code>svga</code>	1024x768x60
<code>1152</code>	1152x900x76
<code>1280</code>	1280x1024x76
<code>none</code>	(現在デバイスにプログラムされている表示モード)

-res オプションには、表示モードの直後に次の追加引数を指定することができます。追加引数は、単独でも複数でも指定することができます。

nocheck

このオプションを指定すると、モニターセンスコードに基づく通常のエラーチェックが行われません。ユーザーによって指定された表示モードは、現在接続されているモニターに適切かどうかにかかわらず受け付けられます。このオプションは、M64 デバイ스에 異なるモニターを接続する場合に便利です。このオプションを指定すると、`noconfirm` も指定されます。

noconfirm

-res オプションを指定した際に、システムが使用できない状態になり、表示出力がなくなる場合があります。このような状況は、特定のコードが読み込まれた際のモニターセンスコードにあいまいさがあった場合などに発生します。このような事態を避けるために m64config のデフォルトの動作では、この問題についての警告メッセージと、処理を継続するかどうかを確認するメッセージを表示します。`noconfirm` オプションを指定すると、m64config コマンドはこの確認をせずに、

m64config(1M)

要求のあった表示モードにプログラムします。このオプションは、m64config がシェルスクリプトから実行されている場合に便利です。

now

OWconfig ファイルの表示モードを更新するとともに、M64 デバイスが指定した表示モードにただちにプログラムされます (この機能は、ウィンドウシステムを開始する前に表示モードを変更する際に便利です)。

対象となるデバイスが稼働している間 (たとえば、ウィンドウシステムの稼働中) に、この追加オプションを m64config に指定することはお勧めしません。予期しない結果になることもあります。now オプションを指定して m64config コマンドを実行する場合は、最初にウィンドウシステムを終了してください。now オプションがウィンドウシステムのセッション中に使用された場合、表示モードはただちに変更されますが、画面の幅や高さはそのセッションが終了して次のセッションに入るまで変更されません。さらに、立体表示モードではシステムが変更を認識しないことがあります。したがって、ウィンドウシステムの稼働中には絶対に now オプションを指定しないでください。

try

このオプションを指定すると、指定した表示モードが試験的にプログラムされます。ユーザーは、指定した表示モードを使用する場合は、メッセージが表示されてから 10 秒以内に y と入力します。表示されたモードを使用しない場合は、10 秒以内に任意の文字を入力します。y または Return キー以外の文字の入力は、すべて「使用しない」とみなされ、以前の表示モードに戻され、OWconfig ファイル中の表示モードは書き換えられません (その他の指定されたオプションは有効となります)。Return キーの入力があつた場合は、新しい表示モードを保持するかどうかを yes または no で確認するメッセージが表示されます。このオプションを指定すると、now サブオプションも指定されます。(now サブオプションの注意を参照)。

デフォルト設定

m64config コマンド行で指定されていないオプションについては、対応する OWconfig ファイル中のオプションは更新されず、ファイル内の値がそのまま使用されます。

ウィンドウシステムを実行する際に、m64config による M64 オプションの指定がまったくなかった場合は、デフォルト値が使用されます。オプションのデフォルトを以下に示します。

オプション	デフォルト値
-dev	/dev/fbs/m640
-file	machine
-res	none

-res オプションのデフォルト値 none とは、ウィンドウシステムが実行された場合に、画面解像度がそのデバイスに現在プログラムされている表示モードになることを意味しています。

m64config(1M)

これによって、PROM によってデバイスの解像度を指定しているユーザーとの共用性が保てます。(GX などの)一部のデバイスでは、PROM が表示モードを指定する唯一の手段です。これは、デフォルトの M64 表示モードは、最終的に PROM によって決まることを意味しています。

使用例 例 1 モニターの種類の変更

モニターの種類を、垂直周波数 76 Hz で解像度 1280x1024 に変更する例を以下に示します。

```
example% /usr/sbin/m64config -res 1280x1024x76
```

ファイル /dev/fbs/m640
デバイス特殊ファイル

/usr/openwin/server/etc/OWconfig
システム設定ファイル

/usr/lib/fbconfig/SUNWm64_config
usr/sbin/m64config へのシンボリックリンク

属性 以下の属性については、attributes(5)を参照してください。

属性タイプ	属性値
使用条件	SUNWm64cf

関連項目 attributes(5), m64(7D)

『OpenWindows デスクトップ・リファレンスマニュアル』

mkfile(1M)

名前	mkfile – ファイルの作成				
形式	mkfile [-nv] size [g k b m] filename...				
機能説明	mkfile は、NFS マウント上のファイル、あるいはローカルファイルをスワップ領域として使用するのに適した 1 つまたは複数のファイルを作成します。root ユーザーとして mkfile() を実行すると、デフォルトでスティッキービットが設定され、ファイルはゼロでパディングされます。root 以外のユーザーは、mkfile() を実行するときに、chmod(1) を使用して、手動でスティッキービットを設定する必要があります。デフォルトでは size の単位はバイトですが、g、k、b、または m のフラグを使用すると、ギガバイト、キロバイト、ブロック、またはメガバイト単位でも指定できます。				
オプション	次のオプションを指定できます。 -n 空の filename を作成します。サイズは出力されますが、ディスクブロックはデータが書き込まれるまで割り当てられません。このオプションで作成したファイルは、ローカルの UFS マウント上でスワップできません。 -v 詳細表示。作成したファイルの名前とサイズを報告します。				
使用法	2G バイト (2 ³¹ バイト) 以上のファイルを検出した場合の mkfile の動作については、largefile(5) 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) のマニュアルページを参照してください。				
	<table border="1"><thead><tr><th>属性タイプ</th><th>属性値</th></tr></thead><tbody><tr><td>使用条件</td><td>SUNWcsu</td></tr></tbody></table>	属性タイプ	属性値	使用条件	SUNWcsu
属性タイプ	属性値				
使用条件	SUNWcsu				
関連項目	chmod(1), swap(1M), attributes(5), largefile(5)				

名前	mkfs – ファイルシステムの構築
形式	mkfs [-F <i>FSType</i>] [<i>generic_options</i>] [-o <i>FSType-specific_options</i>] <i>raw_device_file</i> [<i>operands</i>]
機能説明	<p>mkfs ユーティリティは、-F <i>FSType</i> で指定したファイルシステムタイプに固有の mkfs モジュールを呼び出すことによって、<i>raw_device_file</i> 上にファイルシステムを構築します。</p> <p>注: UFS ファイルシステムを作成するときは、通常 newfs(1M) コマンドを使用します。</p> <p><i>generic_options</i> はファイルシステムの種類によらない共通のオプションです。 <i>FSType-specific_options</i> は、<i>keyword=value</i> の組みを、空白を入れずにコンマで区切ったリストで、<i>FSType</i> に固有です。 <i>raw_device_file</i> は、ファイルシステムを作成するディスクパーティションです。この引数は必須で、(もしあれば) <i>specific_options</i> のすぐ後に指定する必要があります。 <i>operands</i> は <i>FSType</i> に固有の引数です。詳細については、mkfs の <i>FSType</i> に固有なマニュアルページを参照してください (たとえば、mkfs_ufs (1M))。</p>
オプション	<p>mkfs の汎用オプションは次のとおりです。</p> <ul style="list-style-type: none"> -F 構築する <i>FSType</i> を指定します。-F を指定しないと、<i>FSType</i> は /etc/vfstab から一致する <i>raw_device_file</i> を探すことによって、あるいは、/etc/default/fs に指定されているデフォルトを調べることによって決定されます。 -V コマンド行全体をエコーしますが、コマンドは実行しません。エコーされたコマンド行には、ユーザーが指定したオプションと引数、さらに、/etc/vfstab または /etc/default/fs から得られた情報が追加されます。このオプションは、コマンド行を確認および検証するときに使用します。 -m ファイルシステムを作成するのに使用したコマンド行を表示します。ファイルシステムは、あらかじめ存在していなければなりません。このオプションは、ファイルシステムを構築したときに使用したパラメータを調べるのに使用します。 -o <i>FSType</i> に固有なオプションを指定します。ファイルシステムタイプ (<i>FSType</i>) に固有な mkfs モジュールのマニュアルページを参照してください。
使用法	2G バイト (2 ³¹ バイト) 以上のファイルを検出した場合の mkfs の動作については、largefile(5) のマニュアルページを参照してください。
ファイル	<p>/etc/default/fs デフォルトのファイルシステムタイプ。デフォルト値は、/etc/default/fs 内で次のように設定されています。(例: LOCAL=ufs)</p> <p>LOCAL <i>FSType</i> を指定しない場合に、コマンドがデフォルトで使用するパーティション</p>

mkfs(1M)

/etc/vfstab 各ファイルシステム用のデフォルトのパラメータリスト

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 mkfs_ufs(1M), newfs(1M), vfstab(4), attributes(5), largefile(5)

mkfs の *FSType* に固有なモジュールのマニュアルページ

注意事項 このコマンドは、すべての *FSType* で使用できるわけではありません。

lofiadm を使用すると、mkfs コマンドには raw デバイスのように見えるファイルを作成できます。次に、mkfs コマンドを使用すると、そのデバイス上にファイルシステムを作成できます。mkfs_ufs(1M) と mkfs_pcfs(1M) を使用することにより、lofiadm で作成したデバイス上に UFS と PC (FAT) ファイルシステムを作成する例については、lofiadm(1M) のマニュアルページを参照してください。

modify_install_server(1M)

名前 | install_scripts, add_install_client, add_to_install_server, modify_install_server, rm_install_client, setup_install_server, check – Solaris ソフトウェアをインストールするためのスクリプト

形式 | `cdrom-mnt-pt/Solaris_XX/Tools/add_install_client [-i IP_address] [-e Ethernet_address] [-s server_name : path] [-c server_name : path] [-n [server] : name_service [(netmask)] [-p server_name : path] [-f boot_file_name] host_name platform_group`
`cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server_name:path] [-c server_name:path] [-p server_name:path] [-t install_boot_image_path] [-f boot_file_name] platform_name platform_group`
`cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server_name:path] [-c server_name:path] [-p server_name:path] [-t install_boot_image_path] [-f boot_file_name] -e Ethernet_address [-b property=value] platform_group`
`cdrom-mnt-pt/Solaris_XX/Tools/add_to_install_server [-s] [-p product_image_path] install_server_path`
`cdrom-mnt-pt/Solaris_XX/Tools/jumpstart_sample/check [-p install_dir_path] [-r rulesfile]`
`cdrom-mnt-pt/modify_install_server [-p] install_dir_path installer_miniroot_path`
`cdrom-mnt-pt/Solaris_XX/Tools/rm_install_client host_name`
`cdrom-mnt-pt/Solaris_XX/Tools/setup_install_server [-b] [-t install_boot_image_path] [-w wanboot_image_path] install_dir_path`

機能説明 | これらのコマンドは、Solaris Software and Solaris Installer CD のスライス 0 に置かれています。Solaris CD をすでにローカルディスクにコピーしている場合、`cdrom_mnt_pt` はコピーされている Solaris CD へのパスになります。これらのコマンドは、さまざまなインストール作業に使用できます。

Solaris_XX の XX は、使用している Solaris リリースのバージョン番号です。

`add_install_client` コマンドには 3 種類の形式があります。「形式」の項を参照してください。

ネットワークを使用してインストールするクライアントを追加する場合は、次の形式の `add_install_client` を使用します (これらのコマンドは `bootparams(4)` ファイルを更新します)。`add_install_client` コマンドは、インストールサーバーの Solaris インストールイメージ (マウントした CD またはディスクにコピーした Solaris CD) あるいはブートサーバーの起動ディレクトリ (ブートサーバーが必要な場合) から実行する必要があります。Solaris インストールイメージまたはブート専用ディレクトリの Solaris リリースは、クライアントにインストールする予定の Solaris リリースと同じでなければなりません。

`cdrom-mnt-pt/Solaris_XX/Tools/add_install_client [-i IP_address] [-e Ethernet_address] [-s server_name : path] [-c server_name : path]`

modify_install_server(1M)

```
[ -n [server] ] : name_service [ ( netmask) ] [ -p server_name : path ]
host_name platform_group
```

プラットフォームグループ内のあるプラットフォームのインスタンスをインストールサーバーに追加する場合は、次の形式の `add_install_client` コマンドを使用します。このグループは DHCP を使用して起動および構成されます。スクリプトは、必要な構成作業をサーバー上で実行し、ユーザーがそのグループのために DHCP サーバーに追加しなければならないデータを出力します。

```
cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server:path]
[-c server:path] [-p server:path] [-t install boot image path]
[-f boot file name] platform_name platform_group
```

インストールサーバーに単一のクライアントを追加する場合は、次の形式の `add_install_client` コマンドを使用します。このクライアントは DHCP を使用して起動および構成されます。スクリプトは、必要な構成作業をサーバー上で実行し、ユーザーがそのクライアントのために DHCP サーバーに追加しなければならないデータを出力します。すでに使用しているものにも、上記のように `-f` フラグを追加する必要があります。`-f` を使用すると、ユーザーは所定のクライアントに使用する起動ファイルの名前を指定できます。

```
cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d
[-s server_name:path] [-c server_name:path] [-p server_name:path]
[-t install_boot_image_path] [-f boot_file_name] -e Ethernet_address
platform_group
```

注 - x86 アーキテクチャの Pre-boot eXecution Environment (PXE) クライアントを登録する場合は、必ず `-d` オプションを使用してください。x86 PXE クライアントは構成に DHCP を使用します。

他の Solaris CD とネットワークインストールサーバー上の既存のイメージをマージする場合は、`add_to_install_server` を使用します。マージ可能な各 CD (現時点では OS CD 2 および Language CD) には専用の `add_to_install_server` スクリプトが含まれています。配布された CD に含まれているもの以外の `add_to_install_server` スクリプトを使用しないでください。

`rules` ファイル (カスタム JumpStart インストールを使用する場合に限り必要) のルールを検証する場合は、`check` を使用します。

既存のネットワークインストールサーバーのミニルートを Solaris Installation CD のミニルートに置き換える場合は、`modify_install_server` を使用します。ネットワークインストールサーバーのインストール時のユーザーインターフェースが、Solaris Installation CD の Web Start ユーザーインターフェースに変更されません。

`modify_install_server` コマンドを使用する場合は、既存のインストールイメージ (`setup_install_server` を使用して作成) があらかじめ存在していなければなりません。

modify_install_server(1M)

ネットワークインストール用のクライアントを削除する場合は、`rm_install_client` を使用します (このコマンドは `bootparams(4)` ファイルを更新します)。

Solaris CD を (インストールサーバーを設定するために) ディスクにコピーする、(WANboot インストールサーバーを設定するために) WANboot ミニルートイメージを構築する、または Solaris CD のブートソフトウェアだけを (ブートサーバーを設定するために) ディスクにコピーする場合は、`setup_install_server` を使用します。ネットワーク経由でクライアントをインストールするには、インストールサーバーが必要です。インストールサーバーとインストールするクライアントが異なるサブネットに属している場合、ネットワークインストールを行うには、ブートサーバーも必要です (ブートサーバーはクライアントのサブネットに配置する必要があります)。

オプション `add_install_client` には、次のオプションを指定できます。

-b *property=value*

ブートサーバーの TFTP ディレクトリ (デフォルトは `/tftpboot`) にあるクライアント特有の `bootenv.rc` ファイルにプロパティ値を設定します。クライアントに特有のブートプロパティを設定する場合には、このオプションを使用します。`bootpath` プロパティを使用すると、デバイス構成補助を使用して手動で選択しなくても、ブートデバイスを自動的に選択できます。`boot-args` プロパティを使用すると、JumpStart インストールを自動的に選択できます。 `eeeprom(1M)` を参照してください。

このオプションは x86 クライアントにのみ使用できます。このオプションを使用するときには、必ず `-d` オプションと `-e` を同時に指定してください。

-c *server_name:path*

このオプションが必要なのは、カスタム JumpStart インストール用の JumpStart ディレクトリを指定する場合だけです。`server_name` は、JumpStart ディレクトリが置かれているサーバーのホスト名です。`path` は JumpStart ディレクトリの絶対パスです。

-d

DHCP クライアントを指定します。

-e *Ethernet_address*

インストールするシステムの Ethernet アドレスを指定します。

-f

インストールするクライアントの `boot_file_name` を指定します。

-i *IP_address*

インストールするクライアントの IP アドレスを指定します。

-n [*server*]: *name_service*[(*netmask*)]

このオプションでは、システム構成時に使用するネームサービスを指定します。`bootparams(4)` ファイル内の `ns` キーワードを設定します。

name_service

有効なエントリは `nis`、`nisplus`、および `none` です。

modify_install_server(1M)

netmask

数字を4つずつピリオドで区切って並べたもので、IPアドレスのどの部分がネットワーク部分で、どれがホスト部分かを指定します。

server

サーバーの名前または指定したネームサービスのIPアドレス。指定したサーバーが異なるサブネット上にある場合は、クライアントからサーバーにアクセスするために、*netmask*が必要なことがあります。

-p *server_name*: *path*

このオプションは、システム情報またはネットワーク情報を前もって設定するために必要な、ユーザーが定義した *sysidcfg* ファイルの位置です。*server_name* は有効なホスト名またはIPアドレスです。*path* はJumpStart ディレクトリの絶対パスです。

-s *server_name*:*path*

このオプションが必要なのは、ブートサーバーから *add_install_client* を使用する場合だけです。サーバー名およびこのインストールで使用される Solaris インストールイメージの絶対パスを指定します。*path* はマウントされている Solaris CD のパスまたは Solaris CD のコピーが置かれているディレクトリのパスです。

add_to_install_server コマンドには、次のオプションを指定できます。

-p

コピーする CD (補助的な製品が収められている) の位置を指定します。

-s

ユーザーが、インストールする必要がある製品だけをリストから選択できるようにします。

check コマンドには、次のオプションを指定できます。

-p *install_dir_path*

使用中のシステムの *check* スクリプトではなく、指定された Solaris インストールイメージの *check* スクリプトを使用することによって、ルールファイルを検査します。*install_dir_path* は、ローカルディスク上またはマウントした Solaris CD 上の Solaris インストールイメージのパスです。

旧バージョンの Solaris が稼働しているシステムでは、このオプションを使用して、最新バージョンの *check* を実行できます。

-r *rulesfile*

rules 以外の名前のルールファイルを指定します。このオプションを使用すると、ルールの妥当性を検証してから、ルールファイルに組み込むことができます。*check* はルールが有効であるかどうかを報告するだけで、カスタム JumpStart インストールに必要な *rules.ok* ファイルは作成しません。

modify_install_server コマンドには、次のオプションを指定できます。

-p

このオプションは、*install_dir_path/Solaris_XX/Tools/Boot.orig* にすでに存在しているイメージのミニルートを保持します。

modify_install_server(1M)

setup_install_server コマンドには、次のオプションを指定できます。

- b
サーバーの設定を、ブートサーバー専用にします。
- t
代替ミニルート指定できるようにします。
- w
WANboot ミニルートイメージを構築します。

オペランド add_install_client コマンドには、次のオペランドを指定できます。

host_name

インストールするクライアントの名前です。

platform_group

特定のソフトウェアを配布するために、ベンダーが定義したハードウェアプラットフォームグループ。以下に有効なプラットフォームグループの例を示します。

システム	プラットフォームグループ
x86	i86pc
Sun Fire 4800	sun4c

システムのプラットフォームグループを調べるには、(-m オプションを指定して) `uname(1)` コマンドを使用します。

platform_name

システムのプラットフォーム名を調べるには、-i オプションを指定して `uname(1)` コマンドを使用します。

次に、`uname command` コマンドを使用して Ultra 1 のシステムプラットフォーム名を調べる例を示します。

```
uname -i
システムは、次のように応答します。
```

```
SUNW,Ultra-1
システムのプラットフォーム名は SUNW,Ultra1 です。
```

次のコマンドは、Ultra 1 用の `add_install_client` を呼び出します。

```
add_install_client -d SUNW,Ultra-1 sun4u
```

IA32 プラットフォームでは、プラットフォーム名はつねに `SUNW.i86pc` です。

次のコマンドは、IA32 プラットフォーム用の `add_install_client` を呼び出します。

```
add_install_client -d SUNW.i86pc i86pc
```

`rm_install_client` コマンドには、次のオペランドを指定できます。

modify_install_server(1M)

host_name

削除するクライアントの名前です。

setup_install_server コマンドには、次のオペランドを指定できます。

install_dir_path

Solaris ソフトウェアのコピー先となるディレクトリの絶対パスです。このディレクトリは空でなければなりません。

wanboot_image_path

WANboot ミニルートイメージを含むファイルが作成されるディレクトリの絶対パスです。

使用例 例 1 add_install_client の使用

次の add_install_client コマンドは、マウント済みの Solaris CD からネットワークインストール用のクライアントを Solaris 8 のインストールサーバーに追加します。

```
example# cd /cdrom/cdrom0/s0/Solaris_9/Tools
example# ./add_install_client system_2/sun4u
```

例 2 add_install_client の使用

次の add_install_client コマンドは、ネットワークインストール用のクライアントをインストールサーバーに追加します。カスタム JumpStart インストールを実行するためのルールファイルおよびプロファイルファイルが置かれている JumpStart のサーバーとディレクトリのパスを -c オプションで指定します。Solaris CD は Solaris 8 上の /export/install ディレクトリにコピーされています。

```
example# cd /export/install/Solaris_8/Tools
example# ./add_install_client -c install_server:/jumpstart system_1 i86pc
example# ./add_install_client -c install_server:/jumpstart system_2 i86pc
```

例 3 add_install_client の使用

次の add_install_client コマンドは、起動ファイル sun4u.solaris8 を使用する特定の sun4u プラットフォームマシン (8:0:20:99:88:77) に対するサポートを追加します。

```
example# add_install_client -d -f sun4u.solaris8 -e 8:0:20:99:88:77 sun4u
```

例 4 add_install_client の使用

次の add_install_client コマンドは、PXE 規格を使用してネットワークから起動する x86 クライアントを追加します。

```
example# add_install_client -d -s svrname:/mnt/export/root SUNW.i86pc i86p
```

例 5 add_to_install_server の使用

次の add_to_install_server コマンドは、CD 上のすべての製品ディレクトリにあるパッケージを Solaris 8 上の既存のインストールサーバーにコピーします。

例 5 add_to_install_server の使用 (続き)

```
example# cd /cdrom/cdrom0/s0
example# ./add_to_install_server /export/Solaris_8
```

例 6 check の使用

次の check コマンドは、カスタム JumpStart インストールに使用するルールファイルの構文を検査します。

```
example# cd jumpstart_dir_path
example# ./check -p /cdrom/cdrom0/s0
```

例 7 modify_install_server の使用

次の modify_install_server コマンドは、上記の setup_install_server を使用して作成されたミニルートを Boot.orig に移し、Solaris Installer CD 上のミニルートに置き換えます。

```
example# cd /cdrom/cdrom0/s0
example# ./modify_install_server -p /export/install /cdrom/cdrom0/s1
```

例 8 modify_install_server の使用

次の modify_install_server コマンドは、上記の setup_install_server を使用して作成されたミニルートを Solaris Installer CD 上のミニルートに置き換えます。

```
example# cd /cdrom/cdrom0/s0
example# ./modify_install_server /export/install /cdrom/cdrom0/s1
```

例 9 rm_install_client の使用

次の rm_install_client コマンドは、Solaris 8 上のネットワークインストール用クライアントを削除します。

```
example# cd /export/install/Solaris_8/Tools
example# ./rm_install_client holmes
example# ./rm_install_client watson
```

例 10 setup_install_server の使用

次の setup_install_server コマンドは、マウント済みの Solaris CD を Solaris 8 のローカルディスク上にある /export/install というディレクトリにコピーします。

```
example# cd /cdrom/cdrom0/s0/Solaris_8/Tools
example# ./setup_install_server /export/install
```

例 11 setup_install_server の使用

次の setup_install_server コマンドは、マウント済みの Solaris CD のブートソフトウェアを、Solaris 8 上のサブネット用ブートサーバーになるシステムの /boot_dir というディレクトリにコピーします。

modify_install_server(1M)

例 11 setup_install_server の使用 (続き)

```
example# cd /cdrom/cdrom0/s0/Solaris_8/Tools
example# ./setup_install_server -b /boot_dir
```

例 12 setup_install_server の使用

setup_install_server はデフォルトで、マウント済みの Solaris 配布ディスク上の Solaris ../Tools/Boot にあるインストールブートディレクトリを検索します。

以前に ./setup_install_server -b /boot_dir コマンドを使用して、ネットワークブートサーバー上に作成したディレクトリと別のディレクトリが必要な場合は、-t オプションを使用できます。

```
example# cd /cdrom/cdrom0/s0/Solairs_9/Tools
example# ./setup_install_server -t /boot_dir /export/install
```

例 13 setup_install_server と WANboot オプションの使用

次の setup_install_server コマンドは WANboot ミニルートファイルシステムのイメージを作成し、それを /wanboot_dir/miniroot に保存します。

```
example# cd /cdrom/cdrom0/s0/Solairs_9/Tools
example# ./setup_install_server -w /wanboot_dir /export/install
```

例 14 x86: ネットワークインストール時に使用するシリアルコンソールの指定 (CD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するシリアルコンソールを指定する方法を示します。この例では、インストールクライアントを次の方法で設定します。

- -d オプションは、クライアントが DHCP を使用してインストールパラメータを設定することを示します。
- -e オプションは、Ethernet アドレスが 00:07:e9:04:4a:bf であるクライアントでのみインストールが行われることを示します。
- 1 つめと 2 つめの -b オプションは、インストールプログラムが入力デバイスおよび出力デバイスとしてシリアルポート ttya を使用することを示します。

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
i86pc
```

-b オプションで使用できるブートプロパティの変数とその値については、eprom(1M) を参照してください。

例 15 ネットワークインストール時に使用するブートデバイスの指定 (CD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するブートデバイスを指定する方法を示します。インストールクライアントを設定するときにブートデバイスを指定する場合は、インストール時にデバイス構成補助によって出されるこの情報は表示されません。

例 15 ネットワークインストール時に使用するブートデバイスの指定 (CD) (続き)

この例では、次の方法でインストールクライアントを設定します。

- -d オプションは、クライアントが DHCP を使用してインストールパラメータを設定することを示します。
- -e オプションは、Ethernet アドレスが 00:07:e9:04:4a:bf であるクライアントでのみインストールが行われることを示します。
- 1 つめと 2 つめの -b オプションは、インストールプログラムが入力デバイスおよび出力デバイスとしてシリアルポート ttya を使用することを示します。
- 3 つめの -b オプションは、インストールプログラムがインストール時に特定のブートデバイスを使用することを示します。
- ブートデバイスのパスは、使用しているハードウェアによって異なります。
- i86pc というプラットフォーム名は、クライアントが x86 ベースのシステムであることを示します。

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
-b "bootpath=/pci@0,0/pci108e,16a8@8" i86pc
```

-b オプションで使用できるブートプロパティの変数とその値について詳細は、eeprom(1M) を参照してください。

終了ステータス 次の終了ステータスが返されます。

- 0 正常終了
- 1 エラーが発生した

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	Solaris CD

関連項目 uname(1), bootparams(4), attributes(5)

『Solaris 9 インストールガイド』

modinfo(1M)

名前	modinfo - ロードされているカーネルモジュールについての情報の表示				
形式	<code>/usr/sbin/modinfo [-c] [-w] [-i module-id]</code>				
機能説明	<p>modinfo ユーティリティは、ロードされているモジュールに関する情報を表示します。表示形式は次のとおりです。</p> <pre>Id Loadaddr Size Info Rev Module Name</pre> <p><i>Id</i> はモジュール ID、<i>Loadaddr</i> はテキスト開始位置のアドレス (16 進表記)、<i>Size</i> はテキストとデータと <i>bss</i> のバイト単位の合計サイズ (16 進表記)、<i>Info</i> はモジュール固有の情報、<i>Rev</i> はロード可能モジュールシステムのリビジョン番号、そして <i>Module Name</i> はモジュールのファイル名と内容の説明を表します。</p> <p>モジュール固有の情報として表示される内容は、モジュールの種類により異なります。すなわち、ドライバであればブロックメジャー番号とキャラクタメジャー番号、システムコールであればシステムコール番号、その他の種類であれば以下に示すようなカーネルテーブルへのインデックスが表示されます。</p> <pre>fmodsw STREAMS モジュールの場合 vfssw ファイルシステムの場合 class スケジューリングクラスの場合 execsw exec モジュールの場合</pre>				
オプション	<p>次のオプションを指定できます。</p> <pre>-c ロードされているモジュールのインスタンス数と、モジュールの現在の状態を表示します。 -i module-id 指定したモジュールに関する情報のみを表示します。 -w モジュールに関する情報を、80 文字の位置で切り落としません。</pre>				
使用例	<p>例 1 modinfo コマンドの使用</p> <p>次の例では、モジュール 3 に関する情報を表示しています。</p> <pre>example% modinfo -i 3 Id Loadaddr Size Info Rev Module Name 3 f5a7a000 3bc0 1 1 spedfs (specfs のファイルシステム)</pre>				
属性	<p>次の属性については <code>attributes(5)</code> のマニュアルページを参照してください。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">属性タイプ</th> <th style="text-align: center;">属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWcsu</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWcsu
属性タイプ	属性値				
使用条件	SUNWcsu				
関連項目	modload(1M), modunload(1M), attributes(5)				

名前	modload – カーネルモジュールのロード				
形式	modload [-p] [-e <i>exec_file</i>] <i>filename</i>				
機能説明	<p>modload コマンドは、<i>filename</i> で示すロード可能モジュールを、稼働中のシステムにロードします。</p> <p><i>filename</i> は、ld -rで生成されたオブジェクトファイルを示します。<i>filename</i> が絶対パス名の場合、その絶対パスが指定するファイルがロードされます。<i>filename</i> の先頭文字がスラッシュ (/) ではない場合、-p オプションが指定されていないと、現在のディレクトリに相対するパスを使って <i>filename</i> をロードします。</p> <p>カーネルのモジュールパス modpath 変数は、/etc/system ファイルを使って設定できます。この変数のデフォルト値は、オペレーティングシステムをロードした時のパスです。通常は、/kernel /usr/kernel となります。したがって、次のように入力した場合、カーネルは ./drv/foo ファイルを探します。</p> <pre>example# modload drv/foo</pre> <p>また、次のように入力した場合には、カーネルはまず /kernel/drv/foo を探し、存在しなければ /usr/kernel/drv/foo を探します。</p> <pre>example# modload -p drv/foo</pre>				
オプション	<p>次のオプションを試用できます。</p> <p>-p モジュールを検索するパスとして、カーネルの内部 modpath 変数を使用します。</p> <p>-e <i>exec_file</i> モジュールのロードが正常に終了した後で実行すべき シェルスクリプトまたは実行可能イメージの名前を指定します。そのスクリプトまたはイメージにはいくつかの引数が渡されます。第1引数は常にモジュール ID (10 進数) です。他の引数は、モジュールの種類により異なります。ドライバには、ブロックメジャー番号とキャラクタメジャー番号、システムコールにはシステムコール番号、その他のモジュールタイプにはそれぞれにあったカーネルテーブルへのインデックスが指定されます (modinfo(1M) を参照)。</p>				
属性	<p>次の属性については attributes(5) のマニュアルページを参照してください。</p> <table border="1"> <thead> <tr> <th>属性タイプ</th> <th>属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWcsu</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWcsu
属性タイプ	属性値				
使用条件	SUNWcsu				
関連項目	ld(1), add_drv(1M), kernel(1M), modinfo(1M), modunload(1M), system(4), attributes(5), modldrv(9S), modlinkage(9S), modlstrmod(9S), module_info(9S)				

modload(1M)

Writing Device Drivers

注意事項

デバイスドライバを追加するには、modloadではなく add_drv(1M) を使います。デバイスドライバの追加方法については、*Writing Device Drivers* を参照してください。

名前	modunload – モジュールのアンロード				
形式	modunload -i <i>module_id</i> [-exec <i>exec_file</i>]				
機能説明	modunload は、稼動中のシステムからロード可能モジュールをアンロードします。 <i>module_id</i> はアンロードするモジュールの ID で、これは modinfo(1M) の出力情報中に得られる値と同じです。ID として 0 を指定した場合、自動ローディングされたモジュールのうちアンロード可能なものがすべてアンロードされます。modload(1M) を使ってロードしたモジュールは対象とはなりません。				
オプション	次のオプションを指定できます。 <p>-e <i>exec_file</i> モジュールをアンロードする前に実行すべきシェルスクリプトまたは実行可能イメージの名前を指定します。そのスクリプトまたはイメージにはいくつかの引数が渡されます。第 1 引数は常にモジュール ID (10 進数) です。他の 2 つの引数は、モジュールの種類により異なります。ロード可能ドライバの場合は、ブロックメジャー番号が第 2 引数です。ロード可能システムコールの場合は、システムコール番号が第 2 引数です。ロード可能 exec クラスの場合は、execsw テーブルへのインデックスが第 2 引数です。ロード可能ファイルシステムの場合は、vfssw テーブルへのインデックスが第 2 引数です。ロード可能ストリームモジュールの場合は、fmodsw テーブルへのインデックスが第 2 引数です。ロード可能スケジューリングクラスの場合は、クラス配列へのインデックスが第 2 引数です。該当しない引数の値としては、-1 が渡されます。</p> <p>-i <i>module_id</i> アンロードすべきモジュールを指定します。</p>				
属性	次の属性については attributes(5) のマニュアルページを参照してください。				
	<table border="1"> <thead> <tr> <th>属性タイプ</th> <th>属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWcsu</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWcsu
属性タイプ	属性値				
使用条件	SUNWcsu				
関連項目	modinfo(1M), modload(1M), update_drv(1M), attributes(5)				
注意事項	modunload コマンドは、ドライバモジュールに対してしばしば関連するドライバ構成ファイルをシステムに再度読み込ませる目的で、使用されてきました。この方法は Solaris 9 でも機能しますが、将来のリリースではサポートされなくなる可能性があります。ドライバ構成ファイルの再読み込みには、update_drv(1M) コマンドを使用することをお勧めします。				

mount(1M)

名前	mount, umount – ファイルシステムとリモート資源のマウントおよびマウント解除
形式	<pre>mount [-p -v] mount [-F FSType] [generic_options] [-o specific_options] [-O] special mount_point mount [-F FSType] [generic_options] [-o specific_options] [-O] special mount_point mount -a [-F FSType] [-V] [current_options] [-o specific_options] [mount_point...] umount [-f] [-V] [-o specific_options] special mount_point umount -a [-f] [-V] [-o specific_options] [mount_point...]</pre>
機能説明	<p>mount は、ファイルシステムをファイルシステム階層内の <i>mount_point</i> (ディレクトリのパス名) に継ぎ足します。マウント操作を実行する前に <i>mount_point</i> の下に存在したファイルとディレクトリは、ファイルシステムのマウントを解除するまで見えなくなります。</p> <p>umount は、現在マウントされているファイルシステムをマウント解除します。マウントを解除するファイルシステムは <i>mount_point</i> または <i>special</i> (ファイルシステムが存在しているデバイス) のどちらかで指定できます。</p> <p>現在マウントされているファイルシステムのテーブルは、マウント済みファイルシステム情報ファイルにあります。このファイルは通常、<i>/etc/mnttab</i> 上にマウントされているファイルシステムによって提供されます。マウント済のファイルシステムについての情報は、<i>mnttab(4)</i> に格納されています。ファイルシステムをマウントすると、マウントテーブルにエントリが追加されます。(umount で) ファイルシステムをマウント解除すると、テーブルからエントリが削除されます。</p> <p><i>special</i> および <i>mount_point</i> 両方の引数と <i>-F</i> オプションを指定した場合、mount は <i>special</i> 以外のすべての引数の妥当性を検査し、適切な <i>FSType</i> に固有な mount モジュールを呼び出します。引数なしで呼び出した場合、mount は、マウントテーブル <i>/etc/mnttab</i> に記録されているすべてのマウント済みファイルシステムをリストします。引数を一部だけ、たとえば、<i>special</i> か <i>mount_point</i> のどちらか 1 つだけを指定した場合や、<i>special</i> と <i>mount_point</i> の両方を指定したが、<i>FSType</i> は指定しない場合、mount は <i>/etc/vfstab</i> を調べて、指定されていない引数を補完するエントリを探します。そのようなエントリが見つからず、また、<i>special</i> 引数が <i>/</i> で始まる場合、<i>/etc/default/fs</i> に指定されているデフォルトのローカルのファイルシステムタイプが使用されます。それ以外の場合、デフォルトのリモートファイルシステムタイプが使用されます。デフォルトのリモートファイルシステムタイプは、<i>/etc/dfs/fstypes</i> ファイルの最初のエントリによって決定されます。指定されていない引数を補完した後、mount は <i>FSType</i> に固有な mount モジュールを呼び出します。</p> <p>mount または umount を使用してファイルシステムをマウントまたはマウント解除できるのはスーパーユーザーだけです。ただし、マウントされているファイルシステムとリソースを一覧表示するだけであれば、だれでも mount コマンドを使用することができます。</p>

オプション 次のオプションを指定できます。

-F *FSType*

操作の対象となる *FSType* を指定します。*FSType* は、明示的に指定するか、あるいは、`/etc/vfstab` から決定できるか、`/etc/default/fs` または `/etc/dfs/fstypes` を調べることで決定できるものでなければなりません。

-a [*mount_points...*]

可能であれば、複数の `mount` または `umount` 操作を同時に実行します。

マウントポイントを指定しないと、`mount` は `/etc/vfstab` において `mount at boot` フィールドが `yes` に設定されているすべてのファイルシステムをマウントします。マウントポイントを指定すると、`/etc/vfstab` の `mount at boot` フィールドは無視されます。

マウントポイントを指定すると、`umount` は指定されたマウントポイントだけをマウント解除します。マウントポイントを指定しないと、`umount` は `/etc/mnttab` にあるすべてのファイルシステムのマウントを解除します。ただし、`/`、`/usr`、`/var`、`/var/adm`、`/var/run`、`/proc`、`/dev/fd`、および `/tmp` など、システムに必須のファイルシステムは除きます。

-f

ファイルシステムのマウントを強制的に解除します。

このオプションを指定しないと、`umount` は、ファイルシステム上のファイルがビジー状態である場合、ファイルシステムをマウント解除しません。このオプションを指定すると、オープンしているファイルのデータが失われる可能性があります。ファイルシステムをマウント解除した後にプログラムがファイルにアクセスしようとすると、エラー (EIO) が返されます。

-p

マウントされているファイルシステムの一覧を `/etc/vfstab` 形式で表示します。このオプションを指定するときは他のオプションを指定してはなりません。「使用上の留意点」を参照してください。

-v

マウントされているファイルシステムの一覧を詳細形式で表示します。このオプションを指定するときは他のオプションを指定してはなりません。

-V

コマンド行全体をエコーしますが、コマンドは実行しません。エコーされたコマンドには、ユーザーが指定したオプションと引数、さらに、`/etc/mnttab` から得られた情報が追加されます。このオプションは、コマンド行を確認および検証するときに使用します。

generic_options

ほとんどの *FSType* に固有なコマンドモジュールで使用できる共通のオプションです。次のオプションを指定できます。

-m

ファイルシステムをマウントしますが、`/etc/mnttab` にエントリを作成しません。

mount(1M)

- g
ファイルシステムを広域的にマウントします。クラスタ化されているシステムでは、クラスタ内にあるすべてのノード上でファイルシステムを広域的にマウントします。クラスタ化されていないシステムでは、このオプションは何の効果もありません。
- o
サブオプションとキーワード属性の組の並を、空白を入れずにコンマで区切った形式で、*FSType* 固有のオプションを指定します。これらのオプションはコマンドの *FSType* に固有なモジュールによって解釈されます (`mount_ufs(1M)` のマニュアルページを参照)。`/etc/vfstab` にエントリがあるファイルシステムで `-o` オプションを指定すると、`/etc/vfstab` 内でそのファイルシステムに指定されている `mount` コマンドのオプションはすべて無視されます。
- O
オーバーレイマウント。既存のマウントポイント上にファイルシステムをマウントできます。これより既存のマウントポイントのファイルシステムにはアクセスできなくなります。このフラグを指定せずに既存のマウントポイント上にファイルシステムをマウントしようとする、マウントは失敗して、`device busy` というエラーメッセージが表示されます。
- r
ファイルシステムを読み取り専用でマウントします。

使用法 2G バイト (2^{31} バイト) 以上のファイルを検出した場合の `mount` と `umount` の動作については、`largefile(5)` のマニュアルページを参照してください。

ファイル `/etc/mnttab` マウントされているファイルシステムのテーブル
`/etc/default/fs` デフォルトのローカルのファイルシステムタイプ。デフォルト値は、`/etc/default/fs` 内で次のように設定されています。(例: `LOCAL=ufs`)
 LOCAL: *FSType* を指定しない場合に、コマンドがデフォルトで使用するパーティション
`/etc/vfstab` 各ファイルシステム用のデフォルトのパラメータリスト

属性 次の属性については、`attributes(5)` のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 `mount_cachefs(1M)`, `mount_hsf(1M)`, `mount_nfs(1M)`, `mount_pcfs(1M)`, `mount_tmpfs(1M)`, `mount_ufs(1M)`, `mountall(1M)`, `umountall(1M)`, `mnttab(4)`, `vfstab(4)`, `attributes(5)`, `largefile(5)`, `lofs(7FS)`, `pcfs(7FS)`

注意事項	ファイルシステムがマウントされるディレクトリがシンボリックリンクの場合、ファイルシステムは、シンボリックリンク自身ではなく、シンボリックリンクが参照するディレクトリ上にマウントされます。
使用上の留意点	mount -p の出力は、cachefs については正しくありません。

mountall(1M)

名前	mountall, umountall – 複数のファイルシステムのマウントおよびマウント解除
形式	mountall [-F <i>FSType</i>] [-l -r] [<i>file_system_table</i>] umountall [-k] [-s] [-F <i>FSType</i>] [-l -r] [-n] umountall [-k] [-s] [-h <i>host</i>] [-n]
機能説明	<p>mountall は、ファイルシステムテーブルに指定されているファイルシステムをマウントするときに使用します。ファイルシステムテーブルは <i>vfstab</i>(4) 形式である必要があります。 <i>file_system_table</i> を指定しない場合は、 /etc/vfstab が使用されます。 <i>file_system_table</i> として - を指定すると、mountall は標準入力からファイルシステムテーブルを読み取ります。mountall は、 <i>file_system_table</i> において mount at boot フィールドが yes に設定されているファイルシステムだけをマウントします。</p> <p>ファイルシステムテーブル内のファイルシステムごとに、次のロジックが実行されます。つまり、 /usr/lib/fs/<i>FSType</i>/fsckall というファイル (<i>FSType</i> はファイルシステムのタイプ) が存在する場合、当該ファイルシステムをリストに保存しておき、後でまとめて /usr/lib/fs/<i>FSType</i>/fsckall スクリプトに引数として渡します。 /usr/lib/fs/<i>FSType</i>/fsckall スクリプトは、引数リスト内にあるファイルシステムをすべて検査して、安全にマウントできるかどうかを決定します。 <i>FSType</i> のファイルシステム用の /usr/lib/fs/<i>FSType</i>/fsckall スクリプトが存在しない場合、そのファイルシステムは fsck(1M) を使用して個々に検査されます。検査の結果、ファイルシステムがマウントできない状態であることが判明した場合、マウントを試行する前に fsck で修復されます。 fsckdev フィールドのエントリが - であるファイルシステムは事前の検査なしにマウントされます。</p> <p>umountall は、 root、 /usr、 /var、 /var/adm、 /var/run、 /proc、 および /dev/fd を除き、マウントされているファイルシステムをすべてマウント解除します。 <i>FSType</i> を指定すると、mountall と umountall のアクションは指定された <i>FSType</i> に制限されます。 -k を指定した場合でも、umountall がビジー状態のファイルシステムをマウント解除するかどうかは保証されません。</p>
オプション	<p>次のオプションを指定できます。</p> <p>-F マウントまたはマウント解除するファイルシステムの <i>FSType</i> を指定します。</p> <p>-h <i>host</i> <i>host</i> からリモートでマウントされている、 /etc/mnttab 内のファイルシステムをすべてマウント解除します。</p> <p>-k fuser -k <i>mount-point</i> コマンドを使用します。詳細については、 fuser(1M) のマニュアルページを参照してください。 -k オプションは SIGKILL シグナルをファイルを使用している各プロセスに送信します。このオプションはプロセスごとに終了シグナルを生成するので、終了メッセージがすぐに出力されないことがあります。 -k を指定した場合でも、umountall がビジー状態のファイルシステムをマウント解除するかどうかは保証されません。</p> <p>-l アクションをローカルのファイルシステムに制限します。</p>

mountall(1M)

- n 指定したオプションで実行されるアクションをリスト表示します。ただし、実際にはアクションを実行しません。-n オプションを指定せずにコマンドを繰り返すと、コマンドを繰り返す間に /etc/mnttab ファイルは変更されなかったと仮定して、リストされたアクションが実行されます。
- r アクションをリモートのファイルシステムタイプに制限します。
- s 複数の umount 操作を同時に実行しないようにします。

ファイル	/etc/mnttab	マウント済みファイルシステムテーブル
	/etc/vfstab	デフォルトのファイルシステムテーブル
	/usr/lib/fs/ <i>FSType</i> /fsckall	<i>FSType</i> タイプのファイルシステムをすべて検査するときに mountall が呼び出すスクリプト

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 fsck(1M), fuser(1M), mount(1M), mnttab(4), vfstab(4), attributes(5)

診断 ファイルシステムがマウント可能であり、クリーンな状態であれば、メッセージは出力されません。

エラーメッセージと警告メッセージは、fsck(1M) と mount(1M) の両方から送られます。

newfs(1M)

名前	newfs – 新しい UFS ファイルシステムの構築
形式	newfs [-NTv] [<i>mkfs-options</i>] <i>raw-device</i>
機能説明	<p>newfs は、ディスクパーティション上に UFS ファイルシステムを作成する mkfs(1M) プログラムをより使いやすいようにしたフロントエンドプログラムです。newfs は、最適化されたパラメータ値を算出して、mkfs を呼び出します。</p> <p>対話形式で実行する場合 (つまり、標準入力が tty である場合)、newfs はファイルシステムを作成する前に確認のプロンプトを出力します。</p> <p>-N オプションを指定せず、デバイスの i ノードがランダム化されていない場合、newfs は fsirand(1M) を呼び出します。</p> <p>このコマンドを使用するにはスーパーユーザーである必要があります。ただし、UFS ファイルシステムをフロッピーディスク上に作成するときは例外です (使用例を参照)。</p>
マルチテラバイトの UFS ファイルシステムの作成	<p>マルチテラバイトの UFS ファイルシステムを作成する場合、次の制限があることに注意してください。</p> <ul style="list-style-type: none">■ <i>nbpi</i> の値は、特別にこれより大きな値を設定しない限り、1M バイトに設定されています。マルチテラバイトの UFS ファイルシステムでは、<i>nbpi</i> を 1M バイトより小さな値に設定することはできません。■ <i>fragsize</i> は <i>bsize</i> と等しい値に設定されます。■ デフォルトで、ロギングは有効です。
オプション	<p>次のオプションを指定できます。</p> <p>-N ファイルシステムを作成するときに使用するファイルシステムのパラメータを出力しますが、ファイルシステムの作成は行いません。この場合、fsirand(1M) は呼び出されません。</p> <p>-T ファイルシステムの全体サイズが最終的に 1T バイトを超えることを許可するようにファイルシステムのパラメータを設定します。このオプションは、<i>fragsize</i> の値を <i>bsize</i> と同じに、また、<i>-i</i> オプションで値をさらに大きくするよう設定されていない場合は <i>nbpi</i> の値を 1M バイトに設定します。<i>-f</i> オプションまたは <i>-i</i> オプションを使用して、このオプションと両立しない <i>fragsize</i> または <i>nbpi</i> を指定すると、このオプションの <i>fragsize</i> または <i>nbpi</i> にユーザーが設定した値は無視されます。</p> <p> デフォルトでは、このオプションを使用して作成されたすべてのファイルシステムにロギングが有効です。</p> <p>-v 詳細表示。newfs はそのアクション (mkfs に渡されるパラメータも含む) を出力します。</p> <p><i>mkfs-options</i> デフォルトのパラメータを無効にするオプションは次のとおりです。</p>

- a *apc*** 不良ブロックの交換用に予約する、シリンダ当たりの代替セクター数 (SCSI デバイスのみ)。デフォルト値は 0 です。
- このオプションは、EFI ラベルに準拠したディスクには適用されず、無視されます。
- b *bsize*** ファイルシステムの論理ブロックのサイズ (バイト数)。4096 または 8192 で、デフォルトは 8192 です。sun4u アーキテクチャでは、4096 ブロックのサイズは使用できません。
- c *cgsize*** シリンダグループ当たりのシリンダ数 (16 から 256 まで)。デフォルトは、ファイルシステムのセクター数を 1G バイトのセクター数で割って、その結果に 32 をかけたものです。デフォルト値は常に 16 から 256 の範囲の値です。mkfs は、この値を無効にすることがあります。詳細については、mkfs_ufs(1M) のマニュアルページを参照してください。
- このオプションは、EFI ラベルに準拠したディスクには適用されず、無視されます。
- C *maxcontig*** 回転待ちが発生する前に連続して割り当てられる、1 つのファイルに属する論理ブロックの最大数。デフォルト値は、ディスクドライブの最大転送速度から決定されます。
- UFS は 1048576 バイト (1M バイト) までサポートします。論理ブロックのサイズが 4096 の場合、*maxcontig* 制限は 256 です。8192 の場合、128 になります。
- このパラメータは *tunefs(1M)* を使用して後で変更できます。
- このパラメータはクラスタリングも制御します。*gap* の値に関わらず、クラスタリングが有効なのは *maxcontig* が 1 よりも大きいときだけです。クラスタリングを使用すると、連続入出力の速度が向上します。詳細については、*tunefs(1M)* のマニュアルページを参照してください。
- d *gap*** 回転待ち。同じディスク上で転送完了割り込みから、新しい転送を開始するまでの予想待ち時間 (ミリ秒単位)。ファイル内の連続するブロック間にどれだけの回転間隔を置くべきかを決定するときに使用されます。このパラメータ

は `tunefs(1M)` コマンドを使用して後で変更できます。デフォルトは 0 で、このとき、ディスクドライブはそのトラックで先読みバッファリングを行います。

このオプションは、EFI ラベルに準拠したディスクには適用されず、無視されます。

`-f fragsize`

ファイルに割り当てるディスク容量の最小値 (バイト数)。論理ブロックのサイズが 4096 である場合、有効な値は 512、1024、2048、および 4096 です。論理ブロックのサイズが 8192 である場合、有効な値は 1024、2048、4096、および 8192 です。デフォルト値は 1024 です。

1T バイトより大きなファイルシステム、または `-t` オプションを使用して作成したファイルシステムの場合、`fragsize` は強制的にブロックサイズ (`bsize`) と同じ値に変更されます。

`-i nbpi`

`i` ノード当たりのバイト数。これは、ファイルシステム内にある `i` ノードの密度を指定します。この値をファイルシステムの合計サイズで割ることによって、作成できる `i` ノード数が決まります。この値は、ファイルシステム内にあるファイルの予想平均サイズを反映します。`i` ノードを減らしたい場合は大きい値を指定します。逆に、`i` ノードを増やしたい場合は小さい値を指定します。`nbpi` のデフォルト値は次のとおりです。

ディスクサイズ	密度
1G バイト未満	2048
2G バイト未満	4096
3G バイト未満	6144
3G バイト～ 1T バイト	8192
1T バイトより大きい。または -T を使用して作成した場合	1048576

`-m free`

ファイルシステム内で維持する空き領域の最小パーセンテージ (1% から 99% まで。1% と 99% も含む)。通常ユーザーはこの領域を設定できません。ファイルシステムがこのしきい値に達してしまったら、ファイルシステムに書き込みを続けられるのはスーパーユーザーだけです。このパラメータは `tunefs(1M)` コマンドを使用して後で変更できます。

newfs(1M)

	デフォルト値は ((64 M バイト/パーティションのサイズ) * 100) で、小数点以下は切り捨てられます。1% 以上、10% 以内に制限されます。
-n <i>nrpos</i>	シリンダグループを分割する回転位置の数。デフォルト値は 8 です。
-o <i>opt</i>	このオプションは、EFI ラベルに準拠したディスクには適用されず、無視されます。 (<i>space</i> または <i>time</i>)。ブロック割り当てにかかる時間を最小化するか、ディスク上のフラグメンテーションを最小化するかをファイルシステムに指示します。デフォルト値は <i>time</i> です。
-r <i>rpm</i>	分当たりのディスクの回転速度。デフォルト値はドライバまたはデバイスに固有です。 このオプションは、EFI ラベルに準拠したディスクには適用されず、無視されます。 <i>newfs</i> コマンドの場合は <i>rpm</i> を、 <i>mkfs</i> コマンドの場合は <i>rps</i> を指定することに注意してください。
-s <i>size</i>	ファイルシステムのサイズ (セクター単位)。デフォルトではパーティション全体を使用します。
-t <i>ntrack</i>	ディスク上のシリンダ当たりのトラック数。デフォルト値はディスクラベルから取得されます。 このオプションは、EFI ラベルに準拠したディスクには適用されず、無視されます。

オペランド 次のオペランドを指定できます。

raw-device ファイルシステムを作成する /dev/rdisk 上にある raw 特殊デバイスの名前 (たとえば、/dev/rdisk/c0t0d0s6)

使用法 2G バイト (2³¹ バイト) 以上のファイルを検出した場合の *newfs* の動作については、*largefile(5)* のマニュアルページを参照してください。

使用例 例 1 raw 特殊デバイスのパラメータを詳細に表示する

次の例は、raw 特殊デバイス *c0t0d0s6* のパラメータを詳細に表示します。この例では、新しいファイルシステムの作成は行いません。

```
example# newfs -Nv /dev/rdisk/c0t0d0s6
mkfs -F ufs -o N /dev/rdisk/c0t0d0s6 1112940 54 1
```

例1 raw 特殊デバイスのパラメータを詳細に表示する (続き)

```
5 8192 1024 16 10 60
2048 t 0 -1 8 /dev/rdisk/c0t0d0s6: 1112940 sectors in
1374 cylinders of 15 tracks, 54 sectors 569.8MB in 86 cyl
groups (16 c/g, 6.64MB/g, 3072 i/g) super-block backups
(for fsck -b #) at:
32, 13056, 26080, 39104, 52128, 65152, 78176, 91200, 104224, . . .
```

例2 UFS ファイルシステムを作成する

次の例は、ボリュームマネージャが管理するフロッピーディスク上に UFS ファイルシステムを作成します。

```
example% newfs /vol/dev/aliases/floppy0
newfs: construct a new file system /vol/dev/aliases/floppy0: (y/n)? y
/vol/dev/aliases/floppy0: 2880 sectors in 80 cylinders of 2 tracks,
18 sectors 1.4MB in 5 cyl groups (16 c/g, 0.28MB/g, 128 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
32, 640, 1184, 1792, 2336, . . .
```

例3 マルチテラバイトまでサイズを増大させる UFS ファイルシステムの作成

次の例は、サイズが増大して最終的にマルチテラバイトになる UFS ファイルシステムを作成します。

このコマンドは、ボリューム /dev/md/rdisk/d99 に 800G バイトのファイルシステムを作成します。

```
# newfs -T /dev/md/rdisk/d99
newfs: construct a new file system /dev/md/rdisk/d99: (y/n)? y
/dev/md/rdisk/d99: 1677754368 sectors in 45512 cylinders of
144 tracks, 256 sectors
819216.0MB in 1821 cyl groups (25 c/g, 450.00MB/g, 448 i/g) . . .
```

この後、ファイルシステムのボリュームサイズを増大させる場合は、growfs コマンドを使用してファイルシステムを拡張できます。この例では、ファイルシステムのサイズは 1.2T バイトに増大します。

```
# growfs -v /dev/md/rdisk/d99
/usr/lib/fs/ufs/mkfs -G /dev/md/rdisk/d99 2516631552 /dev/md/rdisk/d99:
2516631552 sectors in 68268 cylinders of 144 tracks, 256 sectors
1228824.0MB in 2731 cyl groups (25 c/g, 450.00MB/g, 448 i/g). . .
```

終了ステータス 次の終了ステータスが返されます。

```
0          正常終了
1, 10     使用方法のエラーまたは内部エラー。エラーを説明するメッセージが標準
          エラー出力に送られます。
```


newfs(1M)

newfs から呼び出された mkfs(1M) により、他の終了ステータスが返されることがあります。

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 fsck(1M), fsck_ufs(1M), fsirand(1M), mkfs(1M), mkfs_ufs(1M), tunefs(1M), attributes(5), largefile(5), fs_ufs(7FS)

診断 newfs: No such file or directory
指定されたデバイスが存在していないか、ディスクパーティションが指定されていません。

special: cannot open
このコマンドを使用するにはスーパーユーザーでなければなりません。

patchadd(1M)

名前	patchadd – Solaris オペレーティング環境を実現するシステムへのパッチ適用
形式	patchadd [-dun] [-B <i>backout_dir</i>] [-k <i>keystore</i>] [-P <i>passwd</i>] [-x <i>proxy</i>] [<i>target</i>] <i>source</i> [<i>dest</i>] patchadd -p [<i>target</i>]
機能説明	<p>patchadd は、Solaris 2.x オペレーティング環境、および Solaris 2.x と互換性がある、2.x 以降の Solaris オペレーティング環境 (Solaris 9 など) を稼動しているシステムにパッチパッケージを適用します。patchadd は、Solaris 1.x システム用のパッチを適用するためには使用できません。patchadd を実行するには、スーパーユーザーになる必要があります。</p> <p>patchadd コマンドには、次の使用方法があります。</p> <ul style="list-style-type: none">■ 1 つめの形式は、1 つまたは複数のパッチを、1 つのシステム、クライアント、サービス、またはネットインストールイメージの <i>miniroot</i> にインストールします。■ 2 つめの形式は、>クライアント、サービス、またはネットインストールイメージの <i>miniroot</i> にインストールされたパッチを表示します。
オプション	<p>以下のオプションを指定できます。</p> <p>-B <i>backout_dir</i> パッチのバックアウト (削除) 時に利用されるデータ (バックアウトデータ) を、パッケージデータベース以外のディレクトリに保存します。<i>backout_dir</i> は絶対パス名で指定してください。</p> <p>-d パッチが適用されるファイルのバックアップを作成しません。このオプションを指定すると、適用されたパッチを後で削除 (バックアウト) することはできません。</p> <p>-k <i>keystore</i> 各パッチ内に見つかったデジタル署名を検証するのに、認証局の信頼された証明書を入手するための場所を <i>keystore</i> で指定します。キーストアが指定されていない場合、デフォルトのキーストアの場所で信頼された有効な証明書を探します。詳細は、pkgadd(1M) のキーストアの場所を参照してください。</p> <p>-n 署名を無視してその検証を行いません。これは、パッチの内容が既知で信頼されている場合にのみ使用するべきです。本来、Solaris 8 のように、パッチ署名を検証する機能がないシステムにパッチを適用するためのオプションです。</p> <p>-p 2 つめの形式で使用され、現在適用されているパッチのリストを表示します。</p> <p>-P <i>passwd</i> 必要に応じて、-k で指定したキーストアを復号化するのに使用するパスワードを指定します。このオプションの引数の書式について詳細は、pkgadd(1M) のパスワードの引数を参照してください。</p>

-u	パッチ適用前にファイルを検証せずに、パッチを無条件に適用します。パッチの適用によって変更されるファイルがある場合でも、パッチを適用します。
-x <i>proxy</i>	パッケージをダウンロードする場合に使用する HTTP[S] プロキシを指定します。プロキシの書式は <i>host:port</i> で、ここで <i>host</i> は HTTP[S] プロキシのホスト名、 <i>port</i> はプロキシに関連するポート番号です。このスイッチは、プロキシを指定するほかのすべての方法より優先します。デフォルトのプロキシを指定する代わりにの方法について詳細は、 <code>pkgadd(1M)</code> の <code>環境</code> を参照してください。
オペランド	次のオペランドがサポートされています。
ソース	<code>patchadd</code> は、パッチを抽出するために、ソースを指定する必要があります。次のソースおよび構文が受け入れられます。
	<p><i>patch</i></p> <p><i>patch_id</i> の絶対パス名または署名付きパッチを指す URI。 <code>/var/sadm/spool/patch/104945-02</code> は <i>patch</i> の一例です。 <code>https://syrinx.eng:8887/patches/104945-02</code> は、署名付きパッチを指す URI の一例です。</p> <p>-M <i>patch_dir patch_id [patch_id...]</i> インストールするパッチをディレクトリの場所または URL、およびパッチ番号で指定します。</p> <p>ディレクトリの場所または URL、およびパッチ番号を使用するためには、<i>patch_dir</i> をスプールされるパッチを含むディレクトリの絶対パス名で指定します。スプールされたパッチを含むサーバー名およびパス名を URL で指定します。該当するパッチのパッチ番号は <i>patch_id</i> で指定します。複数の <i>patch_id</i> を指定することが推奨されます。<i>patch_id</i> は該当するパッチのパッチ番号です。104945-02 は <i>patch_id</i> の一例です。</p> <p>-M <i>patch_dir patch_list</i> インストールするパッチをディレクトリの場所または URL、およびパッチリストを含むファイル名で指定します。</p> <p>ディレクトリの場所または URL、およびパッチリストを含むファイル名を使用する場合、<i>patch_dir</i> はスプールされるパッチを含むディレクトリの絶対パス名を指定します。URL は、スプールされるパッチを含むサーバー名およびパス名を指定します。<i>patch_list</i> は、インストールするパッチを含むファイル名を指定します。</p>
宛先	デフォルトでは、 <code>patchadd</code> は指定した宛先にパッチを適用します。宛先が指定されていない場合、現在のシステム (そのルートファイルシステムが / にマウントされているシステム) がパッチの宛先と仮定されます。次の方法で宛先を指定することもできます。

patchadd(1M)

-C *net_install_image*

`setup_install_server` で作成されたネットインストールイメージ上のミニルート上に置かれたファイルにパッチを適用します。*net_install_image* には Solaris 8 またはそれと互換性のあるバージョンの起動ディレクトリへの絶対パス名を指定します。使用例 を参照してください。

ミニルートへのインストールに推奨されているパッチをインストールする場合のみ、`-c` オプションを使用してください。ミニルートへのインストールが推奨されているパッチは、通常、パッケージコマンド、Sun 製インストールツールおよびパッチインストールツールのようなインストール関連のパッチが含まれています。ミニルートにパッチをたくさん適用しすぎると、ミニルートが大きくなり、Solaris のネットインストール時にメモリが足りなくなる可能性があります。ミニルートが大きくなりすぎないように、`-B` オプションと `-c` オプションをいっしょに使用してください。上記の `-B` オプションの説明を参照してください。

-R *client_root_path*

`patchadd` で生成されたすべてのパッチファイルを *client_root_path* の下のディレクトリに配置します。*client_root_path* は、サーバーから見たクライアントのブート可能なルートを含むディレクトリです。*client_root_path* にはディレクトリツリーの先頭の絶対パスを指定します。この下に `patchadd` で生成されたすべてのパッチファイルがあります。`-R` オプションは `-s` オプションと いっしょに指定することはできません。注意事項 を参照してください。

-S *service*

代わりとなるサービスを指定します (たとえば、Solaris 8)。このサービスはサーバーモデルおよびクライアントモデルの一部で、サーバーのコンソールからのみ使用可能です。サーバーには、ホストマネージャで作成された共有の `/usr` ファイルシステムを含むことができます。これらのサービス領域は、それらが扱うクライアントに使用可能にすることができます。`-s` オプションは `-R` オプションと いっしょに指定することはできません。注意事項 を参照してください。

キーストアの場所

詳細は、`pkgadd(1M)` の「キーストアの場所」を参照してください。

キーストアおよび
証明書の形式
使用例

詳細は、`pkgadd(1M)` 「キーストアおよび証明書の形式」を参照してください。

以下に示す例では、`/usr/sbin` ディレクトリのコマンドを使用しているものとします。

例 1 スタンドアロンマシンに 1 つのパッチをインストールする

スタンドアロンマシンに 1 つのパッチをインストールする例を示します。

```
example# patchadd /var/spool/patch/104945-02
```

例 2 サーバーのコンソールからクライアントに 1 つのパッチをインストールする

サーバーのコンソールからクライアントに 1 つのパッチをインストールする例を示します。

```
example# patchadd -R /export/root/client1 /var/spool/patch/104945-02
```

例 2 サーバーのコンソールからクライアントに1つのパッチをインストールする (続き)

例 3 サーバーのコンソールからサービスに1つのパッチをインストールする

サーバーのコンソールからサービスに1つのパッチをインストールする例を示します。

```
example# patchadd -S Solaris_8 /var/spool/patch/104945-02
```

例 4 patchadd を1回実行して複数のパッチをインストールする

1回の patchadd の実行で複数のパッチをインストールする例を示します。

```
example# patchadd -M /var/spool/patch 104945-02 104946-02 102345-02
```

例 5 パッチのリストが記述されているファイルを指定して複数のパッチをインストールする

インストールするパッチのリストが記述されたファイルを指定して、複数のパッチをインストールする例を示します。

```
example# patchadd -M /var/spool/patch patchlist
```

例 6 クライアントに複数のパッチをインストールし、バックアウトデータをデフォルト以外のディレクトリに保存する

クライアントに複数のパッチをインストールし、パッチのバックアウト時に利用されるデータ (バックアウトデータ) をデフォルト以外のディレクトリに保存する例を示します。

```
example# patchadd -M /var/spool/patch -R /export/root/client1
-B /export/backoutrepository 104945-02 104946-02 102345-02
```

例 7 Solaris 8 およびその互換バージョンのネットインストールイメージにパッチをインストールする

Solaris 8 およびその互換バージョンのネットインストールイメージにパッチをインストールする例を示します。

```
example# patchadd -C /export/Solaris_8/Tools/Boot
/var/spool/patch/104945-02
```

例 8 クライアントにインストールされているパッチを表示する

クライアント上にインストールされているパッチを表示する例を示します。

```
example# patchadd -R /export/root/client1 -p
```

終了ステータス 以下の終了ステータスが返されます。

0 正常終了

>0 エラーが発生した

属性 次の属性については attributes(5) のマニュアルページを参照してください。

patchadd(1M)

属性タイプ	属性値
使用条件	SUNWswmt, SUNWcsu
インタフェースの安定性	開発中

診断 パッチのインストール時によく発生する問題、出力されるエラーメッセージ、その対処方法について説明します。

パッチのインストールエラー

メッセージ

```
The prepatch script exited with return code retcode.
patchadd is terminating.
```

説明・対処法

パッチに付属している prepatch スクリプトが 0 以外の終了コードで終了しました。prepatch スクリプトのトレースを実行して、prepatch スクリプトが不正な終了コードで終了した原因を調べてください。問題を修正するには、prepatch スクリプトの先頭行に -x オプションを追加して再度 patchadd を実行してください。

メッセージ

```
The signature on patch patch_id was unable to be verified.
patchadd is terminating.
```

説明・対処法

使用しているキーストアおよびパッチの署名が与えられたのですが、パッチのデジタル署名が検証できませんでした。パッケージの署名を検証するのに必要なトラストアンカーがあるかどうか、またパッケージが勝手に変更されていないかどうか、キーストアを確認してください。

メッセージ

```
The postpatch script exited with return code retcode.
Backing out patch.
```

説明・対処方法

パッチに付属している postpatch スクリプトが 0 以外の終了コードで終了しました。このスクリプトはおもに、パッチパッケージのオブジェクトに対応しないファイルを一掃するため(つまりパッケージの所有権およびアクセス権に問題があるとき)に使用されます。出力された検証エラーをすべて確認し、それぞれについて適切な処置を行なった後、-u オプション付きで再度 patchadd を実行してください。検証エラーを無視してパッチがインストールされます。

メッセージ

```
Insufficient space in /var/sadm/patch to save old files.
(Solaris 2.4 およびそれ以前のシステム)
```

説明・対処方法

パッチ適用前のファイルを保存するための容量が /var/sadm/patch ディレクトリにありません。3つの対処法があります。(1) -B オプション付きで patchadd を実行して、指定したファイルシステムにバックアウトデータ(パッチのバックアウト時に利用されるデータ)を保存します。(2) 不要なファイ

patchadd(1M)

ルを削除することによってディスク容量を確保します。(3) `-d` オプション付きで `patchadd` を実行して、パッチ適用前のファイルを保存しないようにします。

パッチ適用前のファイルを保存しないように選択した場合、後に `patchrm` を使用してパッチを削除することはできなくなります。以前に適用したパッチの保存領域を削除することによって、システム領域を確保するという方法もあります。後にパッチを削除する可能性はないと判断した場合は、`patchadd` によって保存されたファイルを削除しても構いません。パッチ `patch_id` について保存されたファイルを削除するには、次のように実行してください。

```
cd /var/sadm/patch/patch_id
rm -r save/*
rm .oldfilessaved
```

上記のコマンドを実行後は、パッチ `patch_id` は削除できなくなります。

メッセージ

```
Insufficient space in /var/sadm/pkg/PKG/save to save old files.
(Solaris 2.5 およびそれ以降のシステム)
```

説明・対処方法

`/var/sadm/pkg/PKG/save` ディレクトリに容量が不足しているため、パッチ適用前のファイルを保存できません。次の3つの対処方法があります。(1) `-B` オプション付きで `patchadd` を実行して、指定したファイルシステムにバックアウトデータ (パッチのバックアウト時に利用されるデータ) を保存します (1つ前のメッセージに関する説明を参照してください)。(2) 不要なファイルを削除することによってディスク領域を確保します。(3) `-d` オプション付きで `patchadd` を実行して、パッチ適用前のファイルを保存しないようにします。

パッチ適用前のファイルを保存しないように選択した場合、後に `patchrm` を使用してパッチを削除することはできなくなります。以前に適用したパッチの保存領域を削除することによって、システム領域を確保するという方法もあります。後にパッチを削除する可能性がないと判断した場合は、`patchadd` によって保存されたファイルを削除しても構いません。`patch_id` について保存されたファイルを削除するには、次のように実行してください。

```
cd /var/sadm/pkg/pkgabbrev/save
rm -r patch_id
```

上記のコマンドを実行後は、パッチ `patch_id` は削除できなくなります。

メッセージ

```
Save of old files failed.
(Solaris 2.4 およびそれ以前のシステム)
```

説明・対処方法

パッチを適用する前に、パッチインストールスクリプトは `cpio` を使用してパッチ適用前のファイルを保存します。このエラーメッセージは、`cpio` が失敗したことを示します。このエラーメッセージの前に `cpio` からの出力が表示されているはずですが、`cpio` の失敗を修正するために適切な処置を行う必要があります。次の2つの対処方法があります。(1) 不要なファイルを削除してディスク領域を確保します。(2) `-d` オプション付きで `patchadd` を実行してパッチ適用前のファイルを保存しないようにします。ただし、パッチ適用前のファイルを保存

patchadd(1M)

しないように選択した場合、パッチを削除することはできなくなります。

メッセージ

```
Pkgadd of pkgname package failed with error code code.  
See /tmp/log.patch_id for reason for failure.
```

説明・対処方法

メッセージ中に示されたパッチパッケージのインストールに失敗しました。patchadd はパッチをバックアウトして、システムをパッチ適用前の状態にします。ログファイルを参照してインストールに失敗した原因を確認し、必要な処置を行なって問題を解決した後、再度パッチを適用してください。

メッセージ

```
Pkgadd of pkgname package failed with error code code.  
Will not backout patch...patch re-installation.  
Warning: The system may be in an unstable state!  
See /tmp/log.patch_id for reason for failure.
```

説明・対処方法

メッセージ中に示されたパッチパッケージのインストールに失敗しました。patchadd はパッチをバックアウトしません。ログファイルを参照して pkgadd が失敗した原因を確認し、必要な処置を行なって問題を解決してください。その後、patchrm を使用して手動でパッチをバックアウトしてから、再度パッチを適用してください。

メッセージ

```
patchadd is unable to find the INST_RELEASE file. This file  
must be present for patchadd to function correctly.
```

説明・対処方法

システムに INST_RELEASE ファイルがありません。このファイルは初期インストール中またはアップグレード中に作成されます。

メッセージ

```
A previous installation of patch patch_id was invoked  
that saved files that were to be patched. Since files  
were saved, you must run this instance of patchadd  
without the -d option.
```

説明・対処方法

以前に -d オプションを使用しないでパッチがインストールされている場合、パッチを再インストールするときにも -d オプションなしで実行する必要があります。-d オプションを付けずに patchadd を実行してください。

メッセージ

```
A previous installation of patch patch_id was invoked  
with the -d option. (i.e. Do not save files that would  
be patched) Therefore, this invocation of patchadd  
must also be run with the -d option.
```


その他の診断
メッセージ

説明・対処方法

以前に `-d` オプションを使用してパッチがインストールされている場合、パッチを再インストールするときにも `-d` オプションを使用する必要があります。`-d` オプション付きで `patchadd` を実行してください。

以下に示すパッチインストール時のメッセージは、「説明・対処方法」で説明しているように必ずしもエラーではありませんが、パッチインストールのログファイルに記録されます。

メッセージ

```
Package not patched:
PKG=SUNxxxx
Original package not installed
```

説明・対処方法

メッセージに示されているパッチコンポーネントは、システムにインストールされていないパッケージに対するパッチです。これは必ずしもエラーではありません。1つのパッチが1つのバグを複数のパッケージに対して修正することもあります。

たとえば、オンラインバックアップと `fddi` パッケージの両方に対する1つのバグを修正するパッチを例として考えます。オンラインバックアップはインストールされているけれども `fddi` パッケージがインストールされていない場合、次のようなメッセージが出力されます。:

```
Package not patched:
PKG=SUNwbf
Original package not installed
```

システム上に `fddi` パッケージがインストールされている場合には、このメッセージをエラーとして対処する必要があります。必要な処置を行なってパッケージをインストールし、(パッチによって他のパッケージがインストールされている場合は) パッチをバックアウトし、再度パッチをインストールしてください。

メッセージ

```
Package not patched:
PKG=SUNxxx
ARCH=xxxxxxx
VERSION=xxxxxxx
Architecture mismatch
```

説明・対処方法

メッセージに示されているパッチコンポーネントは、ユーザーが使用しているシステムとは異なるアーキテクチャのパッケージに対するパッチです。これは必ずしもエラーではありません。アーキテクチャ固有のパッケージに対するパッチには、該当する各アーキテクチャごとに1つのコンポーネントが含まれている場合もあります。たとえば `sun4m` アーキテクチャのシステムを使用している場合に `SUNwcar` パッケージに対するパッチをインストールしようとする、次のようなメッセージが出力されます。

patchadd(1M)

```
Package not patched:
PKG=SUNWcar
ARCH=sparc.sun4c
VERSION=11.5.0,REV=2.0.18
Architecture mismatch
```

```
Package not patched:
PKG=SUNWcar
ARCH=sparc.sun4u
VERSION=11.5.0,REV=2.0.18
Architecture mismatch
```

```
Package not patched:
PKG=SUNWcar
ARCH=sparc.sun4e
VERSION=11.5.0,REV=2.0.18
Architecture mismatch
```

```
Package not patched:
PKG=SUNWcar
ARCH=sparc.sun4
VERSION=11.5.0,REV=2.0.18
Architecture mismatch
```

使用しているシステムのアーキテクチャを patchadd が正しく認識していない場合のみ、このメッセージはエラーを示しています。

メッセージ

```
Package not patched:
PKG=SUNxxxx
ARCH=xxxx
VERSION=xxxxxxx
Version mismatch
```

説明・対処方法

パッチが適用されるソフトウェアバージョンがシステムにインストールされていません。たとえば Solaris 8 を実行している場合に Solaris 9 に対するパッチをインストールしようとする、次のようなメッセージが出力されます。

```
Package not patched:
PKG=SUNWcsu
ARCH=sparc
VERSION=10.0.2
Version mismatch
```

これは必ずしもエラーではありません。パッチを適用するパッケージのバージョンが不一致の場合は、正しいバージョンのパッチを入手するか、または正しいバージョンのパッケージをインストールしてください。そのあと、必要な場合はパッチをバックアウトしてから、再度パッチを適用してください。

メッセージ

```
Re-installing Patch.
```

説明・対処方法

インストールしようとしているパッチはすでに適用されていますが、追加インストールされるパッケージが少なくとも1つパッチに含まれています。たとえば、AnswerBook がインストールされていないシステムに、OpenWindows と

patchadd(1M)

AnswerBook の両方のコンポーネントが含まれているパッチを適用すると、そのパッチの AnswerBook コンポーネント部分は適用されません。後に pkgadd を使って AnswerBook をインストールしてパッチを再度適用すると、そのパッチの AnswerBook コンポーネント部分がシステムに適用されます。

メッセージ

```
patchadd Interrupted.  
patchadd is terminating.
```

説明・対処方法

patchadd の実行が中断されました (通常 CTRL-c が押されたため)。patchadd は処理中のファイルを削除し、終了します。

メッセージ

```
patchadd Interrupted.  
Backing out Patch...
```

説明・対処方法

patchadd の実行が中断されました (通常 CTRL-c が押されたため)。patchadd は処理中のファイルを削除し、パッチをバックアウトし、終了します。

関連項目 `cpio(1)`, `pkginfo(1)`, `patchrm(1M)`, `pkgadd(1M)`, `pkgchk(1M)`, `pkgrm(1M)`, `smpatch(1M)`, `showrev(1M)`, `attributes(5)`

注意事項 クライアントまたはサーバーにパッチをインストールするには、patchadd を 2 回実行する必要があります。1 回は `-R` オプションを付けて実行し、もう 1 回は `-S` オプションを付けて実行します。これによって、パッチが `/usr` と `/` (ルート) パーティションの両方に確実にインストールされます。パッチに `/usr` と `/` (ルート) のパッケージが含まれている場合に、上記の方法を実行する必要があります。

patchadd を実行すると、pkgadd が起動され、`pkg/install` ディレクトリにあるインストールスクリプトを実行します。checkinstall スクリプトは、所有権を `install` というユーザー名に設定して実行されます。ユーザー名が `install` に設定されていない場合、pkgadd は checkinstall スクリプトを `nobody` として実行します。SVR4 ABI には、checkinstall は情報収集を行うスクリプトとしてのみ使用されると記述されています。checkinstall スクリプトのアクセス権を初期設定から変更すると、pkgadd はファイルをオープンして読み取ることができなくなり、次のようなメッセージが出力されてパッチのインストールが異常終了します。

```
pkgadd: ERROR: checkinstall script did not complete successfully.
```

このため、checkinstall スクリプトのアクセス権は変更しないでください。パッチが正しくインストールされたときのパッチインストールログファイルの内容は、patchadd が pkgadd の出力をリダイレクトしたのになります。パッチが正しくインストールされると、pkgadd は次のようなメッセージを出力し、それがログファイルに書き込まれます。

```
This appears to be an attempt to install the same architecture  
and version of a package which is already installed. This  
installation will attempt to overwrite this package.  
This message does not indicate a failure, it represents the  
correct behavior by pkgadd when a patch installs correctly.
```

patchadd(1M)

このメッセージは処理の失敗を示しているものではありません。パッチが正しくインストールされたときの pkgadd による正しい動作を示しています。

クライアント・サーバーマシンでは、既存のクライアントにまたはクライアントのルートディレクトリ (templates 領域) にパッチパッケージは適用 (インストール) されません。このため、すべてのクライアントマシンにおいて直接 patchadd を使用して、適切な時にクライアントマシンにパッチを適用する必要があります。パッケージユーティリティ (pkgadd, pkgrm, pkgchk) に影響するバグは、patchadd または patchrm の動作に影響する場合があります。patchadd および patchrm は、上記のパッケージユーティリティを使用してパッチパッケージをインストールおよびバックアウトしています。これらのパッケージユーティリティのバグを修正するパッチが提供されているかどうかを確認し、提供されている場合はそのパッチを先に適用してから、他のパッチを適用することをお勧めします。現在提供されている、パッケージユーティリティに対するパッチは、次のとおりです。

Solaris 2.5.1 (Sparc 版):
104578

Solaris 2.5.1 (Intel 版):
104579

Solaris 2.6 (Sparc 版):
106292

Solaris 2.6 (Intel 版):
106293

名前	patchrm – システムからのパッチ削除とパッチ適用前のファイル復元
形式	patchrm [-f] [-B <i>backout_dir</i>] [-C <i>net_install_image</i> -R <i>client_root_path</i> -S <i>service</i>] <i>patch_id</i>
機能説明	patchrm は、Solaris 2.x オペレーティング環境、および Solaris 2.x と互換性がある、2.x 以降の Solaris オペレーティング環境 (Solaris 8 など) を稼動しているシステムから、パッチパッケージを削除し、以前に保存しておいた (パッチ適用前の) ファイルを復元します。patchrm は Solaris 1.x システム用のパッチには使用できません。patchrm を実行するにはスーパーユーザーになる必要があります。
オプション	以下のオプションを指定できます。 <p>-B <i>backout_dir</i> パッケージデータベースとは別のディレクトリにバックアウトデータ (パッチのバックアウト時に利用されるデータ) が保存されているパッチを、バックアウトします。パッチのインストール時に patchadd コマンドでデフォルトのバックアウトデータのディレクトリを変更した場合のみ、このオプションを使用します。<i>backout_dir</i> には絶対パス名を指定してください。</p> <p>-C <i>net_install_image</i> setup_install_server によって作成された、ネットインストールイメージ上の miniroot にあるファイルに対して適用されたパッチを削除 (バックアウト) します。<i>net_install_image</i> には、Solaris 2.6 およびその互換バージョンのブートディレクトリへの絶対パス名を指定してください。</p> <p>-f 他のパッチに置き換えられているかどうかに関係なく、指定したパッチを削除 (バックアウト) します。</p> <p>-R <i>client_root_path</i> patchrm によって生成されるすべてのパッチファイルをディレクトリ <i>client_root_path</i> の下に置きます。<i>client_root_path</i> には、サーバー上のクライアントのルートディレクトリを絶対パス名で指定します。この -R オプションは -S オプションと同時に指定することはできません。</p> <p>-S <i>service</i> 代替サービス (たとえば Solaris_2.3) を指定します。ここでのサービスとは、クライアント・サーバーモデルの OS サービスのことを指しており、サーバーのコンソールでのみ使用できます。サーバーは、ホストマネージャで作成された /usr 共有ファイルシステムを持つことができ、登録された OS サービスをクライアントが利用できるようにすることが可能です。この -s オプションは -R オプションと同時に指定することはできません。</p>
オペランド	以下のオペランドを指定できます。 <p><i>patch_id</i> パッチ番号。たとえば 104945-02 のように指定します。</p>

patchrm(1M)

使用例 以下に示す例では、パッチ 104945-02 がシステムにインストールされていることを前提としています。また、/usr/sbin ディレクトリのコマンドを使用しているものとします。

例 1 スタンドアロンシステム上の 1 つのパッチを削除する

スタンドアロンシステム上のパッチを 1 つ削除 (バックアウト) する例を示します。

```
example# patchrm 104945-02
```

例 2 サーバーのコンソールからクライアントシステム上の 1 つのパッチを削除する

サーバーのコンソールから、クライアントシステム上のパッチを 1 つ削除 (バックアウト) する例を示します。

```
example# patchrm -R /export/root/client1 104945-02
```

例 3 サーバーの OS サービス領域上の 1 つのパッチを削除する

サーバーの OS サービス領域にあるパッチを 1 つ削除 (バックアウト) する例を示します。

```
example# patchrm -S Solaris_2.3 104945-02
```

例 4 ネットインストールイメージ上の 1 つのパッチを削除する

ネットインストールイメージ上のパッチを 1 つ削除 (バックアウト) する例を示します。

```
example# patchrm -C /export/Solaris_2.6/Tools/Boot 104945-02
```

終了ステータス 以下の終了ステータスが返されます。

0 正常終了

>0 エラーが発生した

属性 次の属性については `attributes(5)` のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWswmt, SUNWcsu

診断 パッチのバックアウト時によく発生する問題、出力されるエラーメッセージ、その対処方法について説明します。

メッセージ

```
prebackout patch exited with return code code.  
patchrm exiting.
```

説明・対処方法

パッチに付属している `prebackout` スクリプトが 0 以外の終了コードで終了しました。 `prebackout` スクリプトのトレースを作成して、 `prebackout` スクリプトが失敗した原因を調べてください。問題を修正するには、 `prebackout` スクリプトの先頭行に `-x` オプションを追加して再度 `patchrm` を実行してください。

メッセージ

```
postbackout patch exited with return code code.
patchrm exiting.
```

説明・対処方法

パッチに付属している `postbackout` スクリプトが 0 以外の終了コードで終了しました。 `postbackout` スクリプトを参照して `postbackout` スクリプトが失敗した原因を調べてください。問題を修正するには、 `postbackout` スクリプトの先頭行に `-x` オプションを追加して、必要な場合は `postbackout` スクリプトだけを再度実行してください。

メッセージ

```
Only one service may be defined.
```

説明・対処方法

複数の OS サービスからパッチをバックアウトしようとしています。複数の OS サービスからパッチをバックアウトするには、各 OS サービスごとに別々に `patchrm` を実行してください。

メッセージ

```
The -S and -R arguments are mutually exclusive.
```

説明・対処方法

ネイティブでない OS サービスと `client_root_path` からパッチをバックアウトしようとしています。これら 2 つの引数は互いに排他的です。ネイティブでない `usr` パーティションからパッチをバックアウト (削除) するには `-s` オプションを使用する必要があります。クライアントの (ネイティブまたは非ネイティブの) ルートパーティションからパッチをバックアウトするには `-R` オプションを使用する必要があります。

メッセージ

```
The service service cannot be found on this system
```

説明・対処方法

ネイティブでない OS サービスからパッチをバックアウト (削除) しようとしたが、指定された OS サービスはシステムにインストールされていません。正しい OS サービスを指定してください。

メッセージ

```
Only one client_root_path may be defined.
```

説明・対処方法

`-R` オプションを指定して複数の `client_root_path` を指定しています。1 回の `patchrm` の実行につき `-R` オプションは 1 回だけ使用できます。

patchrm(1M)

メッセージ

The *dir* directory cannot be found on this system.

説明・対処方法

-R オプションを使用して、マウントされていないまたはシステムに存在しないディレクトリを指定しています。正しいディレクトリ名を指定して、パッチのバックアウトを再度実行してください。

メッセージ

Patch *patch_id* has not been successfully installed to this system.

説明・対処方法

システムにインストールされていないパッチをバックアウト (削除) しようとしています。パッチが適用されたファイルをパッチ適用前のバージョンに復元するには、最初のインストール時に使用した CD から元のファイルを復元してください。

メッセージ

Patch *patch_id* has not been successfully applied to this system.
Will remove directory *dir*.

説明・対処方法

システムに適用されていないパッチをバックアウト (削除) しようとしています。パッチは適用されていませんが、(失敗した patchadd によって作成された) /var/sadm/patch/*patch_id* ディレクトリが残っています。パッチをバックアウトすることはできません。パッチが適用されたファイルをパッチ適用前のバージョンに復元するには、最初のインストール時に使用した CD から元のファイルを復元してください。

メッセージ

This patch was obsoleted by patch *patch_id*.
Patches must be backed out in the reverse order in which they were installed. Patch backout aborted.

説明・対処方法

不適切な順番でパッチをバックアウト (削除) しようとしています。パッチは正しい順序でバックアウトする必要があります。バックアウトしようとしているパッチ以外のパッチにも影響が及んでいる可能性があります。

メッセージ

Patch *patch_id* is required to be installed by an already installed *patch_id*.
It cannot be backed out until the required patch is backed out first.

説明・対処方法

インストール (適用) されている必要があるとメッセージ中に示されているパッチをバックアウト (削除) してから、目的のパッチをバックアウトしてください。

メッセージ

```
The installation of patch patch_id was interrupted.
```

説明・対処方法

以前に行なったパッチのインストールが中断されています。目的のパッチをバックアウト (削除) する前に、インストールが中断されたパッチをインストールする必要があります。

メッセージ

```
Patch patch_id was installed without backing up the original files.  
It cannot be backed out.
```

説明・対処方法

パッチの適用時に `patchadd` コマンドの `-d` オプションを指定したか、またはディスク領域を確保するためにパッチの保存領域が削除されています。このため、元のファイルが保存されていないので `patchrm` を使用できません。元のファイルはインストール CD からのみ復元できます。

メッセージ

```
pkgadd of pkgname package failed return code code.  
See /var/sadm/patch/patch_id/log for reason for failure.
```

説明・対処方法

メッセージに示されているパッチパッケージのインストールに失敗しています。ログファイルを参照して原因を確認してください。問題を修正後に、バックアウトスクリプトを再度実行してください。

メッセージ

```
Restore of old files failed.
```

説明・対処方法

パッチが適用されたファイルをパッチ適用前のバージョンに復元するために、バックアウトスクリプトが `cpio` コマンドを使用しています。上記のメッセージの前に `cpio` コマンドからの出力が表示されているはずですが、`cpio` が失敗した原因を調べて必要な処置を行なってください。これは、Solaris 2.4 およびそれ以前のバージョンのシステムで発生するエラーです。

関連項目 `cpio(1)`, `pkginfo(1)`, `patchadd(1M)`, `pkgadd(1M)`, `pkgchk(1M)`, `pkgrm(1M)`, `showrev(1M)`, `attributes(5)`

注意事項 クライアント・サーバーマシンでは、既存のクライアントからまたはクライアントのルートディレクトリ (`templates` 領域) からパッチパッケージは削除 (バックアウト) されません。このため、すべてのクライアントマシンにおいて直接 `patchrm` を使用して、クライアントマシンから適切な時にパッチを削除する必要があります。パッケージユーティリティ (`pkgadd`, `pkgrm`, `pkgchk`) に影響するバグは、`patchadd` または `patchrm` の動作に影響する場合があります。 `patchadd` および `patchrm` は、上記のパッケージユーティリティを使用してパッチパッケージをインストールおよびバックアウトしています。これらのパッケージユーティリティのバグを修正する

patchrm(1M)

パッチが提供されているかどうかを確認し、提供されている場合はそのパッチを先に適用してから、他のパッチを適用することをお勧めします。現在提供されている、パッケージユーティリティに対するパッチは、次のとおりです。

Solaris 2.1:

100901

Solaris 2.2:

101122

Solaris 2.3:

101331

Solaris 2.4 (SPARC 版):

102039

Solaris 2.4 (Intel 版):

102041

Solaris 2.5.1 (SPARC 版):

104578

Solaris 2.5.1 (Intel 版):

104579

Solaris 2.6 (SPARC 版):

106292

Solaris 2.6 (Intel 版):

106293

名前	pgxconfig, GFXconfig, TSIfxp_config – PGX32 (Raptor GFX) グラフィックスアクセラレータの設定
形式	<pre> /usr/sbin/pgxconfig [-dev device-filename] [-res video-mode [try noconfirm nocheck]] [-file machine system] [-depth 8 24] [-24only true false] [-cachedpixmap true false] [-defaults] /usr/sbin/pgxconfig [-propt] [-prconf] /usr/sbin/pgxconfig [-help] [-res ?] /usr/sbin/pgxconfig [-i] </pre>
機能説明	<p>pgxconfig ユーティリティーは PGX32 (Raptor GFX) グラフィックスアクセラレータの設定用コマンドで、これには X11 ウィンドウシステムの一部を PGX32 (Raptor GFX) 用にデフォルト設定する機能も含まれます。以前のバージョンでは、このユーティリティーは GFXconfig という名称でした。</p> <p>形式の項に記された pgxconfig の第 1 の形式では、指定したオプションを OWconfig ファイルに保存します。次回の PGX32 (Raptor GFX) デバイス上でのウィンドウシステム起動時に、ここで指定したオプションに従って PGX32 (Raptor GFX) デバイスを初期化します。OWconfig ファイルに保存されたオプションの更新内容は、異なる複数のウィンドウシステムセッションや再起動後のシステムでも有効となります。</p> <p>その他の形式 (2、3、4 番目の形式) を使用した場合はオプション -prconf, -propt, -help, -res ? を呼び出すだけで、OWconfig ファイルに保存されているオプションを更新することはありません。更に、3 番目の形式を使用した場合はこれ以外のオプションはすべて無視されます。</p> <p>-i オプションを使用すると、pgxconfig は、対話型モードで起動します。</p> <p>一度にオプションを指定できる PGX32 (Raptor GFX) デバイスの数は、1 つだけです。</p> <p>pgxconfig コマンドで指定できるのは、PGX32 (Raptor GFX) 固有のオプションだけです。デフォルトの表示色数、デフォルトのビジュアルクラスなどを指定する通常のウィンドウシステムのオプションは、openwin コマンド行のデバイス修飾子で指定してください。SUNWxwman パッケージに含まれている Xsun(1) マニュアルページを参照してください。</p> <p>ユーザーは、更新する OWconfig ファイルを指定することもできます。デフォルトでは /usr/openwin ディレクトリツリーにあるマシン固有のファイルが更新されます。別のファイルを指定するには、-file オプションを使用します。たとえば、/etc/openwin ディレクトリツリーにあるシステム共通の OWconfig ファイルを代わりに更新することができます。</p> <p>いずれの標準 OWconfig ファイルでも、書き込み権限があるのはスーパーユーザーだけです。</p>
オプション	次のオプションを指定できます。

pgxconfig(1M)

```
-cachedpixmap true | false
false に設定した場合、PGX32 (Raptor GFX) デバイスは、OpenWindows を実行
するときだけに 24 ビットを使用します。デフォルト値は true です。

アプリケーションの中には、ディスプレイデバイスに書き込むときにキャッシュさ
れたピクスマップを使用するものがあります。このような手法を使用すると、出
力が歪曲され、X サーバーがクラッシュする可能性があります。このような問題が
発生した場合は、-cachedpixmap オプションを false に設定してください。

-defaults
すべてのオプションの値をそれぞれのデフォルト値に戻します。

-depth 8 | 24
スクリーンデプスをピクセル当り 8 または 24 ビットに設定します。24 ビット/ピ
クセルに設定すると、ウィンドウシステムで TrueColor グラフィックスを使用でき
ます。

-dev device-filename
PGX32 (Raptor GFX) 特殊ファイルを指定します。デフォルトは
/dev/fbs/gfxp0、または使用可能な場合であれば /dev/fbs/raptor0 です。

-file machine | system
更新する OWconfig ファイルを指定します。ここで machine が指定された場合
は、/etc/openwin ディレクトリツリーにあるマシン固有の OWconfig ファイル
が使用されます。system が指定された場合は、/usr/openwin ディレクトリツ
リーにある共通の OWconfig ファイルが使用されます。ファイルがない場合は、新
たに生成されます。

-help
pgxconfig コマンド行のオプションと機能の概要を一覧で表示します。

-i
pgxconfig コマンドを対話型モードで起動します。

-prconf
PGX32 (Raptor GFX) のハードウェア構成を表示します。以下に表示例を示しま
す。

--- Hardware Configuration for /dev/fbs/gfxp0 ---
DAC: version 0x0
Type:
Board:
PROM: version 0x0
PROM Information:
RAM:
EDID Data:
Monitor Sense ID:
Card possible resolutions: 640x480x60, 800x600x75, 1024x768x60
1024x768x70, 1024x768x75, 1280x1024x75, 1280x1024x76
1280x1024x60, 1152x900x66, 1152x900x76, 1280x1024x67
960x680x112S, 960x680x108S, 640x480x60i, 768x575x50i,
1280x800x76, 1440x900x76, 1600x1000x66, 1600x1000x76,
vga, svga, 1152, 1280, stereo, ntsc, pal
Monitor possible resolutions: 720x400x70, 720x400x88, 640x480x60
640x480x67, 640x480x72, 640x480x75, 800x600x56,
```

```
800x600x60, 800x600x72, 800x600x75, 832x624x75,
1024x768x87, 1024x768x60, 1024x768x70, 1024x768x75,
1280x1024x75, 1280x1024x76, 1152x900x66, 1152x900x76,
1280x1024x67, 960x680x112S, vga, svga, 1152, 1280
stereo
```

```
Current resolution setting: 1280x1024x76
```

```
Possible depths:
```

```
Current depth: 8
```

-propt

-file オプションで指定された OWconfig ファイルに書かれた PGX32 (Raptor GFX) オプションの値のうち、-dev オプションで指定されたデバイスに対するものすべてを表示します。pgxconfig の呼び出しが終了した後に、OWconfig ファイルに書き込まれるオプションの値を表示します。以下に表示例を示します。

```
--- OpenWindows Configuration for /dev/fbs/gfxp0 ---
```

```
OWconfig: machine
```

```
Video Mode: not set
```

```
Depth: 8+24
```

-res *video-mode* [try | noconfirm | nocheck]

PGX32 (Raptor GFX) デバイスに接続されているモニターの制御に使用する組み込みのビデオモードを指定します。

video-mode には、以下のいずれかの書式で指定できます。

widthxheightxrate

width はピクセル単位の画面の幅、*height* はピクセル単位の画面の高さ、*rate* は垂直方向の画面再描画周期です。-res では、再描画周期 *rate* の前の *x* は、@ でも代用できます。つまり周期の指定に限っては、たとえば 1280x1024@76 のような形式にも対応しています。オプションに -res ? を付けて pgxconfig を実行すると (コマンド形式の項に記された 3 番目の形式)、ビデオモードの一覧が表示されます。ビデオボードとモニターの両方が、すべての解像度をサポートしているわけではありません。noconfirm または nocheck オプションを指定しないで、ボードのサポートしていない解像度を入力した場合には、pgxconfig はその要求を許可しません。また、nocheck オプションを指定しないで、モニターのサポートしていない解像度を指定した場合には、その値を適用する前に確認を求めてきます。

Symbolic names

便宜上、一部のビデオモードには記号名が定義されています。*widthxheightxrate* の形式の代わりに、記号名を -res の引数として指定することができます。記号名 none は、ウィンドウシステムを実行すると、画面の解像度が現在デバイスにプログラムされているビデオモードになることを意味します。

```
svga      1024x768x60
```

pgxconfig(1M)

1152	1152x900x76
1280	1280x1024x76
vga	640x480x60
none	デフォルトのコンソールの解像度

-res オプションには、ビデオモードの直後に次の追加引数を指定することができます。追加引数は、単独でも複数でも指定できます。

noconfirm -res オプションを指定した際に、システムが使用可能であっても、表示出力のない状態になる場合があります。このような状況は、特定のコードが読み込まれた際のモニターセンスコードにあいまいさがあった場合などに発生します。このような事態を避けるために pgxconfig のデフォルトの動作では、この問題についての警告メッセージと、処理を継続するかどうかを確認するメッセージを表示します。noconfirm オプションを指定すると、pgxconfig コマンドはこの確認をせずに、要求のあったビデオモードにプログラムします。このオプションは、pgxconfig がシェルスクリプトから実行されている場合に便利です。

nocheck このオプションを指定すると、モニターセンスコードに基づく通常のエラーチェックが行われません。ユーザーによって指定されたビデオモードは、現在接続されているモニターに適切かどうかにかかわらず受け付けられます。このオプションは、PGX32 (Raptor GFX) デバイスに異なるモニターを接続する場合に便利です。このオプションの指定は、noconfirm の指定も兼ねます。

try このオプションを指定すると、指定したビデオモードを適用する前にテストすることができます。まず、指定したモードに基づいたテストパターンが表示されます。テストパターンが正常に表示された場合は、「y」(次いでキャリッジリターン)を入力します。「y」以外の文字を入力(次いでキャリッジリターンを入力)した場合は、「no」となります。

-res ?
PGX32 およびモニターがサポートする解像度の一覧を表示します。

-24only
Openwindows の実行時に、PGX32 (Raptor GFX) デバイスが 24 ビットカラーのみを使用するようにします。

デフォルト pgxconfig コマンド行で指定されていないオプションについては、対応する OWconfig ファイル中のオプションは更新されず、ファイル内の値がそのまま使用されます。ただし、-depth と -24only については、その限りではありません。

ウィンドウシステムを実行する際に、pgxconfig による PGX32 (Raptor GFX) のオプションの指定がまったくなかった場合は、デフォルト値が使用されます。オプションのデフォルト値は次のとおりです。

```
-dev          /dev/fbs/gfxp0
-file         system
-res         none
```

-res オプションのデフォルト値 **none** とは、ウィンドウシステムが実行された場合に、画面解像度がそのデバイスに現在プログラムされているビデオモードになることを意味しています。

使用例 例 1 モニターの種類を変更する。

モニターの種類を、垂直周波数 76 Hz で解像度 1280 x 1024 に変更する例を以下に示します。

```
example# /usr/sbin/pgxconfig -res 1280x1024x76
```

ファイル /dev/fbs/gfxp0
デバイス特殊ファイル

/usr/openwin/server/etc/OWconfig
システム構成ファイル

/etc/openwin/server/etc/OWconfig
マシン構成ファイル

関連項目 PGX32 PCI グラフィックスカード インストールマニュアル

pkgadd(1M)

名前	pkgadd – システムへのソフトウェアパッケージの転送
形式	pkgadd [-nvi] [-a <i>admin</i>] [-d <i>device</i>] [-x <i>proxy</i>] [[-M] -R <i>root_path</i>] [-r <i>response</i>] [-k <i>keystore</i>] [-P <i>passwd</i>] [-V <i>fs_file</i>] [<i>source</i>] [<i>instances</i>] pkgadd -s [<i>source</i>] [<i>instances</i>]
機能説明	<p>pkgadd は、ソフトウェアパッケージの内容をインストール用の配布媒体またはディレクトリからシステムに転送します。-d オプションを指定しない場合、pkgadd はパッケージ用のデフォルトのプールディレクトリ (var/spool/pkg) を探します。-s オプションを指定すると、pkgadd はパッケージをインストールするのではなく、パッケージをプールディレクトリに書き込みます。</p> <p>pkgadd ユーティリティは、インストールされるパッケージと同じサイズの一時容量を必要とします。pkgadd は、\$TMPDIR 環境変数があるかどうかを検査して、使用する一時ディレクトリを決定します。\$TMPDIR が定義されていない場合、pkgadd は stdio.h に指定された P_tmpdir を使用します。P_tmpdir には、デフォルトの /var/tmp/ という値が定義されています。</p> <p>別製品や Sun 以外のパッケージの中には、最新バージョンの pkgadd との完全な互換性が確保されていないものがあります。そのようなパッケージでは、インストールの開始時だけでなく、インストール中もユーザーの関与が必要です。</p> <p>Solaris 2.4 より前にリリースされた古いパッケージをインストールするには、NONABI_SCRIPTS=TRUE 環境変数を設定する必要があります。</p> <p>この環境変数を設定した場合、インストール中にキーボードから pkgadd との対話が可能になります。</p>
オプション	<p>次のオプションを指定できます。</p> <p>-a <i>admin</i> デフォルトのインストール管理ファイルの代わりに使用するインストール管理ファイル <i>admin</i> を定義します。トークン none は <i>admin</i> ファイルの使用を無効にするため、ユーザーの関与が必要になります。完全なパス名を指定しない場合、pkgadd はまず現在の作業ディレクトリで管理ファイルを探します。指定した管理ファイルが現在の作業ディレクトリにないと、pkgadd は /var/sadm/install/admin ディレクトリで管理ファイルを探します。</p> <p>-d <i>device</i> <i>device</i> からパッケージをインストールまたはコピーします。<i>device</i> はディレクトリへのフルパス名 (たとえば、/var/tmp)、あるいは、テープ、フロッピーディスク、または着脱式ディスクの識別子 (たとえば、/floppy/floppy_name) のどちらでもかまいません。また、デバイスの別名 (たとえば、/floppy/floppy0)、あるいは、pkgtrans が作成したデータストリームでもかまいません (pkgtrans(1) のマニュアルページを参照)。</p> <p>-k <i>keystore</i> パッケージ内に見つかったデジタル署名を検証するのに、認証局の信頼された証明書を入手するための場所を <i>keystore</i> で指定します。キーストアが指定されていない場合、デフォルトのキースト</p>

- アの場所で信頼された有効な証明書を探します。詳細は、キーストアの場所 を参照してください。
- M** クライアントのマウントポイントを決定するときに、`$root_path/etc/vfstab` ファイルを使用しないようにします。このオプションは、マウントポイントがサーバー上で適切であり、そのサーバーが Solaris 2.5 以前のリリースで安定して動作しているものと仮定します。
- n** 非対話モードでインストールします。インストールされたファイルの一覧は出力されません。デフォルトは対話モードです。
- P *passwd*** 必要に応じて、**-k** で指定したキーストアを復号化するのに使用するパスワード。このオプションの引数の書式については、`パスフレーズの引数` を参照してください。
- r *response*** 直前の `pkgadd(1M)` セッションからの出力が入っているファイルまたはディレクトリを指定します。このファイルは、対話モードにおけるパッケージからの質問に対する応答を提供します。`response` はフルパス名でなければなりません。
- R *root_path*** `root_path` として使用するディレクトリのフルパス名を定義します。すべてのファイル (パッケージシステム情報ファイルを含む) は `root_path` から始まるディレクトリツリーに再配置されます。`root_path` は、サーバーからクライアントにインストールするときに指定します (たとえば、`/export/root/client1`)。
- s *spool*** パッケージをインストールするのではなく、`spool` ディレクトリに書き込みます。
- v** `pkgadd` によって実行されたすべてのスクリプトを追跡します。これらのスクリプトは `pkginst/install` ディレクトリにあります。このオプションは、手続き型スクリプトや非手続き型スクリプトをデバッグするときに使用します。
- V *fs_file*** クライアントのファイルシステム構成として `fs_file` を指定します。たとえば、`$root_path/etc/vfstab` ファイルが存在しない (あるいは、信頼できない) 場合に使用します。
- x *proxy*** パッケージをダウンロードする場合に使用する HTTP[S] プロキシを指定します。プロキシの書式は `host:port` で、ここで `host` は HTTP[S] プロキシのホスト名、`port` はプロキシに関連するポート番号です。このスイッチは、プロキシを指定するほかのすべての方法より優先します。デフォルトのプロキシを指定する代わりに方法については、`環境` を参照してください。

オプションまたはオペランドを指定しないで実行すると、`pkgadd` は `/var/spool/pkg` (デフォルトのスパールディレクトリ) を使用します。

オペランド 次のオペランドを指定できます。

pkgadd(1M)

ソース	<p>デフォルトでは、<code>pkgadd</code> はインストールまたはスプールするパッケージのインスタンスを探す場合、<code>/var/spool/pkg</code> ディレクトリを探します。任意で、インストールまたはスプールされたパッケージインスタンスのソースを指定することもできます。その場合、下記を指定します。</p> <p><code>-d device</code> <i>device</i> からパッケージをインストールまたはコピーします。<i>device</i> は、ディレクトリの完全パス、またはテープ、フロッピーディスク、またはリムーバブルディスクの識別子(たとえば、<code>/var/tmp</code> または <code>/floppy/floppy_name</code>) で指定できます。デバイス別名(たとえば <code>/floppy/floppy0</code>) または <code>pkgtrans</code> で作成されたデータストリーム (<code>pkgtrans(1)</code> を参照) で指定することもできます。<i>device</i> は、<code>pkgtrans</code> で作成されたデータストリームを指す URL で指定することもできます。</p>
インスタンス	<p>デフォルトでは、<code>pkgadd</code> は指定したソースを検索し、ユーザーが、ソースで見つかったどのパッケージをインストールするか選択できるような対話式メニューがあります。代わりの方法としては、次のオペランドを指定して、インストールするパッケージインスタンスを指定できます。</p> <p><code>pkginst</code> インストールするパッケージインスタンスまたはインスタンスのリストを指定します。ソースメディアで使用可能なすべてのパッケージを指定するには、トークン <code>all</code> を使用できます。パッケージのすべてのインスタンスを指定するには、書式 <code>pkginst.*</code> を使用できます。</p> <p>アスタリスク文字 (*) はあるシェルにとっては特殊文字なので、その場合はエスケープする必要があります。C シェルの場合、アスタリスクは単一引用符 (') で囲むか、前にバックスラッシュ (\) をつける必要があります。</p> <p><code>-Y category[,category...]</code> パッケージの <code>pkginfo(4)</code> ファイルに保存されている <code>CATEGORY</code> パラメータの値に基づいてパッケージをインストールします。<code>CATEGORY</code> の値が指定したカテゴリの 1 つに一致するソースメディア上のすべてのパッケージが、インストールまたはスプール用に選択されます。</p>
キーストアの場所	<p><code>pkgadd</code> または <code>patchadd</code> のようなパッケージツールまたはパッチツールは、パッケージまたはパッチで見つかったすべての署名の検証を行う場合、信頼された証明書のセットを使用します。パッケージまたはパッチに署名が 1 つも含まれていない場合は署名の検証が省略されます。証明書はいろいろな場所にある可能性があります。 <code>-k keystore</code> が指定され、<i>keystore</i> がディレクトリである場合、証明書を使用する基本ディレクトリとして <i>keystore</i> が仮定されます。 <i>keystore</i> がファイルの場合、このファイル自身が必要なキーおよび証明書をすべて持っているとして仮定されます。 <code>-k</code> が指定されない場合、基本ディレクトリとして <code>/var/sadm/security</code> が使用されます。</p> <p>指定した基本ディレクトリの中で、検索される保存場所は、検索を行なっているアプリケーション、および検索しているストアの種類に基づいて異なります。次のようなディレクトリが、指定された順番に検索されます。</p> <ol style="list-style-type: none">1. <code><store_dir>/<app_name>/<store_type></code>2. <code><store_dir>/<store_type></code>

ここで `<store_dir>` が `-k` で指定されたディレクトリである場合、`<app_name>` は検索を行なっているアプリケーションの名前、また、`<store_type>` は `keystore` (秘密鍵の場合)、`certstore` (信頼されていない公開鍵証明書の場合)、または `truststore` (認証局の信頼された証明書の場合) の 1 つです。

たとえば、`pkgadd` が `-k /export/certs` で実行された場合、信頼できる場所を検索するのに次の場所が順番に検索されます。

1. `/export/certs/pkgadd/truststore`
2. `/export/certs/truststore`

この検索順序により、管理者はほとんどのアプリケーションにそれぞれ 1 つの場所を、また、特定のアプリケーションに特別な証明書の場所を特定できます。

キーストアおよび
証明書の形式

`pkgtrans` および `patchadd` のようなパッケージ用ユーティリティおよびパッチ用ユーティリティは、パッケージおよびパッチに署名し、また必要に応じて検証するためにキーおよび証明書のセットにアクセスする必要があります。

キーストアの場所に指定された以下の検索パターンにしたがって見つかったキーストアファイルは、それぞれ自己保有型の PKCS#12 形式ファイルである必要があります。

`pkgtrans` を使用してパッケージに署名する場合、`certstore` に 1 つ以上の公開鍵証明書が含まれている場合、パッケージまたはパッチに署名するのに `-a` オプションで識別できる、または選択できるためには、各公開鍵には `friendlyName` 属性が含まれている必要があります。さらに、`-a` で選択した公開鍵証明書、および `certstore` で見つかった公開鍵証明書は、キーストア内に関連する秘密鍵を持っていなくてはなりません。

証明書およびキーを PKCS#12 キーストアにエクスポートおよびインポートするためには、いくつかのブラウザおよびユーティリティを使用できます。たとえば、OpenSSL ツールキットといっしょに `pkgadd` を使用するためには、信頼される証明書は Netscape からエクスポート可能で、さらに PKCS#12 キーストアにインポート可能です。

パスワードの引
数

`pkgtrans` および `pkgadd` は、パスワードの引数を受け入れます。一般的には `-p` がパスワードを指定します。これにより、パスワードをいろいろなソースから入手できます。これら 2 つのコマンドのオプションはどちらも 1 つの引数を使用し、その書式は下記で示します。パスワードの引数が 1 つも指定されていないがパスワードが必要な場合、ユーザーはパスワードを入力するよう促されます。一般的にこれはエコー機能がオフの状態、現在の端末から読みとられます。

`pass:password`

実際のパスワードは `password` です。パスワードは `ps` のようなユーティリティには見えるため、この書式はセキュリティがそれほど重要でない場合にのみ使用するべきです。

`env:var`

パスワードを環境変数 `var` から取得します。プラットフォームによっては、ほかのプロセスの環境が見えるため、このオプションは注意して使用する必要があります。

pkgadd(1M)

`file:pathname`

`pathname` に含まれる 1 行目がパスワードです。 `pathname` は通常のファイルを参照する必要がありません。というのは、デバイスまたは名前のついたパイプを参照できます。たとえば、標準入力からパスワードを読みとるためには、`file:/dev/stdin` を使用します。

`console`

`/dev/tty` からパスワードを読みとります。

使用例 例 1 Solaris CD-ROM からパッケージをインストールする

次の例は、Solaris CD-ROM からパッケージをインストールします。インストールするパッケージの名前を入力するように求められます。

```
example% pkgadd -d /cdrom/cdrom0/s0/Solaris_2.6
```

終了ステータス 次の終了ステータスが返されます。

- 0 正常終了
- 1 致命的なエラー
- 2 警告
- 3 割り込み
- 4 管理
- 5 管理。対話が必要。 `pkgadd -n` を使用してはならない
- 10 すべてのパッケージのインストール後に再起動する
- 20 当該パッケージのインストール後に再起動する

環境

`HTTPPROXY`

HTTP プロキシホストを指定します。管理ファイル設定、および `http_proxy` 環境変数より優先します。

`HTTPPROXYPORT`

`HTTPPROXY` で指定したホストに接続するとき使用するポートを指定します。`HTTPPROXY` が設定されていない場合は無視されます。

`http_proxy`

プロキシホストおよびプロキシポートを指定するための URL 書式です。管理ファイル設定より優先します。

属性

次の属性については、`attributes(5)` のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWpkgcmdsu
インタフェースの安定性	開発中

pkgadd(1M)

関連項目	<p>pkginfo(1), pkgmk(1), pkgparam(1), pkgproto(1), pkgtrans(1), installf(1M), pkgadm(1M), pkgask(1M), pkgrm(1M), removef(1M), admin(4), pkginfo(4), attributes(5)</p> <p><i>Application Packaging Developer's Guide</i></p> <p>http://www.openssl.org</p>
注意事項	<p>スプールディレクトリにパッケージを転送するときは、<code>-r</code>、<code>-n</code>、および <code>-a</code> オプションは使用できません。</p> <p><code>-r</code> オプションには、ファイル名と同様に、ディレクトリ名も指定できます。このディレクトリには、関連するパッケージと名前を共有する複数の応答ファイルを格納できます。これにより、1回の <code>pkgadd</code> 呼び出しで、複数の対話型パッケージを追加できます。複数の対話型パッケージを追加するには、パッケージごとに応答ファイルが必要です。パッケージと同じ名前 (たとえば、<code>pkinst1</code> や <code>pkinst2</code>) の応答ファイルを作成しておけば、<code>-r</code> オプションの後に、これらの応答ファイルが入ったディレクトリを指定するだけで済みます。</p> <p><code>-n</code> オプションを指定した場合、インストールを完了するのに対話が必要になると、インストールが中断します。</p> <p>デフォルトの <code>admin</code> ファイルの制限が厳しすぎる場合、パッケージのインストール時に非対話モードで操作できるように、管理ファイルを変更する必要があります。詳細については、<code>admin(4)</code> のマニュアルページを参照してください。</p> <p>パッケージストリームが <code>-d</code> で指定されている場合、ストリーム中にデジタル署名が見つかった場合、デフォルトの動作として、証明書と署名を検証しようとします。この動作は、<code>admin</code> ファイル設定で置き換えられます。詳細は、<code>admin(4)</code> を参照してください。</p>

pkgrm(1M)

名前	pkgrm – システムからのパッケージの削除
形式	pkgrm [-nv] [-a <i>admin</i>] [[-A -M] -R <i>root_path</i>] [-V <i>fs_file</i>] [pkginst... -Y <i>category</i> [, <i>category</i> ...]] pkgrm -s <i>spool</i> [pkginst... -Y <i>category</i> [, <i>category</i> ...]]
機能説明	<p>pkgrm はすでにインストールしてある、あるいは、部分的にインストールしたパッケージをシステムから削除します。このとき、削除するパッケージに対して依存性のあるパッケージが存在するかどうかを検査します。パッケージに依存性がある場合に行われる処理は <i>admin</i> ファイルに定義されています。</p> <p>コマンドのデフォルトの動作状態は対話モードです。つまり、処理中にプロンプトメッセージが表示されるので、管理者はどのような処理が行われるかを確認できます。非対話モードで処理を実行するには、-n オプションを使用します。</p> <p>-s オプションを使用すると、特定のディレクトリにスプールされているパッケージを削除できます。</p> <p>別製品や Sun 以外のパッケージの中には、最新バージョンの pkgrm との完全な互換性が確保されていないものがあります。このようなパッケージでは、削除の開始時だけでなく、削除処理中にもユーザーの関与が必要です。</p> <p>Solaris 2.4 より前にリリースされた古いパッケージをインストールするには、次の環境変数を設定します。</p> <pre>NONABI_SCRIPTS=TRUE</pre> <p>この環境変数を設定していると、削除処理の全工程で、キーボードから pkgrm との対話が可能になります。</p>
オプション	<p>次のオプションを指定できます。</p> <p>-a <i>admin</i> デフォルトのインストール管理ファイルの代わりに、インストール管理ファイル <i>admin</i> を使用します。pkgrm はまず現在の作業ディレクトリで管理ファイルを探します。指定した管理ファイルが現在の作業ディレクトリにないと、pkgrm は /var/sadm/install/admin ディレクトリで管理ファイルを探します。</p> <p>-A パッケージファイルをクライアントのファイルシステムから無条件に削除します。ファイルが他のパッケージによって共有されている場合、そのファイルは、デフォルトではクライアントのファイルシステムから削除されません。</p> <p>-M クライアントのマウントポイントを決定するときに、<i>\$root_path/etc/vfstab</i> ファイルを使用しないようにします。このオプションは、マウントポイントがサーバー上で適切であり、そのサーバーが Solaris 2.5 以前のリリースで安定して動作しているものと仮定します。</p> <p>-n 非対話モード。対話の必要がある場合、コマンドは終了します。</p>

pkgrm(1M)

このオプションを使用するには、コマンドを呼び出すときに、少なくとも1つのパッケージのインスタンスを指定しておく必要があります。非対話モードでパッケージを削除するには、特定の状態が存在しているか、制限のない `admin` ファイルを使用する必要があります。

- `-R root_path` `root_path` として使用するディレクトリのフルパス名を定義します。すべてのファイル (パッケージシステム情報ファイルを含む) は `root_path` から始まるディレクトリツリーに再配置されます。
- `-s spool` 指定したパッケージを `spool` ディレクトリから削除します。スプールされているパッケージのデフォルトディレクトリは `/var/sadm/pkg` です。
- `-v` `pkgrm` によって実行されたすべてのスクリプトを追跡します。これらのスクリプトは `pkginst/install` ディレクトリにあります。このオプションは、手続き型スクリプトや非手続き型スクリプトをデバッグするときに使用します。
- `-V fs_file` クライアントのファイルシステム構成として `fs_file` を指定します。たとえば、`$root_path/etc/vfstab` ファイルが存在しない (あるいは、信頼できない) 場合に使用します。
- `-Y category` インストールまたはスプールされているパッケージの `pkginfo(4)` ファイルに格納されている `CATEGORY` パラメータの値に基づいてパッケージを削除します。このオプションでは、`CATEGORY` が `system` に設定されているパッケージをファイルシステムから削除することはできません。

オペランド 次のオペランドを指定できます。

- `pkginst` 削除するパッケージを指定します。パッケージのすべてのインスタンスを削除するには、`pkginst.*` という形式を使用します。

シェルによっては、アスタリスク文字 (*) が特殊な意味を持つことがあります。* をエスケープしなければならない場合があります。Cシェルでは、* は単一引用符 (') で囲むか、バックスラッシュ (\) を前につける必要があります。

使用例 例1 SUNWjunk のすべてのインスタンスを `client1` から削除する

次の例は、SUNWjunk のすべてのインスタンスを `client1` から削除します。

```
example% pkgrm -R /export/root/client1 SUNWjunk*
```

終了ステータス 次の終了ステータスが返されます。

- 0 正常終了
- 1 致命的なエラー
- 2 警告

pkgrm(1M)

- 3 割り込み
- 4 管理
- 10 すべてのパッケージの削除後に再起動する
- 20 当該パッケージの削除後に再起動する

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 pkginfo(1), pkgmk(1), pkgparam(1), pkgproto(1), pkgtrans(1), installf(1M), pkgadd(1M), pkgask(1M), pkgchk(1M), removef(1M), admin(4), pkginfo(4), attributes(5)

Application Packaging Developer's Guide

名前	pmconfig – 電源管理システムの設定
形式	<code>/usr/sbin/pmconfig [-r]</code>
機能説明	<p>pmconfig ユーティリティは、電源管理システムと保存停止・復元再開機能を設定します。ユーザーは、<code>/etc/default/power</code> のキーワードである <code>PMCHANGEPERM</code> で許可されている場合にのみ pmconfig を使用して電源管理設定を変更する権限を持ちます。<code>/etc/default/power</code> のキーワード <code>CPRCHANGEPERM</code> で許可されている場合にのみ pmconfig を使用して保存停止・復元再開機能を変更する権限を持ちます。<code>/etc/default/power</code> のキーワードである <code>PMCHANGEPERM</code> と <code>CPRCHANGEPERM</code> の詳細については、以下に記述する「ファイル」セクションを参照してください。</p> <p>ユーザーの権限に基づいて、pmconfig は、最初に電源管理システムまたは保存停止・復元再開 (またはその両方) をリセットします。それから新しい電源管理システムまたは保存停止・復元再開 (またはその両方) の設定を <code>/etc/power.conf</code> から読み取り、その新しい設定を有効にするコマンドを発行します。pmconfig ユーティリティはシステムの起動時に実行されます。また、このユーティリティは、<code>/etc/power.conf</code> ファイルを手動で変更した後でコマンド行から実行することもできます。<code>/etc/power.conf</code> ファイルを編集した場合、その変更を有効にするには、pmconfig を実行する必要があります。</p> <p>電源管理および保存停止・復元再開の設定変更に、より便利なインターフェースは、<code>dtpower(1M)</code> です。</p>
オプション	<p>以下のオプションを使用することができます。</p> <p><code>-r</code> 電源管理システムと保存停止・復元再開状態をデフォルトの状態にリセットして終了します。このオプションを使用するには、電源管理システムと保存停止・復元再開の設定に対する権限が必要です。</p>
終了ステータス	<p>以下の終了ステータスが返されます。</p> <p>0 正常終了</p> <p>>0 エラーが発生</p>
ファイル	<p><code>/etc/power.conf</code> システムの電源管理設定ファイル</p> <p><code>/etc/default/power</code> システムの電源管理システムと保存停止・復元再開機能に対する権限を制御するファイル。<code>PMCHANGEPERM</code> キーワードは電源管理設定の権限を制御し、<code>CPRCHANGEPERM</code> キーワードは保存停止・復元再開機能の設定権限を制御します。</p> <p>以下に指定できる値を示します。</p> <p><code>all</code> すべてのユーザーが設定を変更できます。</p> <p><code>-</code> スーパーユーザーだけが設定を変更できます。</p> <p><code><user1, user2, ...></code> このユーザーリストに指定されているユーザーまたはスーパーユーザーは、設定の変更ができます。このリ</p>

pmconfig(1M)

	ストには、空白またはコンマ (,) を使用して複数のユーザーを指定できます。このリストは、<と> で囲む必要があります。						
	<code>console-owner</code> システムコンソールデバイスノードを所有するユーザーまたはスーパーユーザーだけが設定を変更できます。						
	デフォルトの値は、 <code>PMCHANGEPERM=console-owner</code> および <code>CPRCHANGEPERM=console-owner</code> です。						
属性	以下の属性については、 <code>attributes(5)</code> を参照してください。						
	<table border="1"><thead><tr><th>属性タイプ</th><th>属性値</th></tr></thead><tbody><tr><td>使用条件</td><td>SUNWpmu</td></tr><tr><td>インタフェースの安定性</td><td>不安定</td></tr></tbody></table>	属性タイプ	属性値	使用条件	SUNWpmu	インタフェースの安定性	不安定
属性タイプ	属性値						
使用条件	SUNWpmu						
インタフェースの安定性	不安定						
関連項目	<code>powerd(1M)</code> , <code>power.conf(4)</code> , <code>attributes(5)</code> , <code>cpr(7)</code> , <code>pm(7D)</code> 『電源管理システム ユーザーマニュアル』						
診断	プログラムが設定ファイルを開くことができない場合は、標準エラー出力にエラーメッセージを出力します。プログラムが設定ファイル内の構文エラーを検出した場合は、エラーメッセージと設定ファイル内のエラーの行番号を出力します。プログラムは、その行の残りの情報は処理せずに次の行を処理します。エラーを含んでいる行の設定情報の中で、すでに処理されているものは使用されます。電源管理システムおよび(または) 保存停止・復元再開機能の設定を変更する権限がない場合、また設定ファイル中にユーザーが権限を持っていないエントリがある場合、権限を持つエントリについてだけ処理が行われ、権限のない部分についてはエラーが出力されます。						

名前	powerd – 電源管理デーモン						
形式	<code>/usr/lib/power/powerd [-n]</code>						
機能説明	<p>powerd デーモンは pmconfig(1M) で起動し、システム動作の監視、および保存停止 - 復元再開機能を使用した自動停止を行います。システムが保存停止されると、電源が落とされる前に、完全な現在の状態がディスクに保存されます。リブート時に、システムは自動的に復元再開操作を開始し、システムが保存停止する直前と同じ状態に復元されます。</p> <p>システムの停止の直前に、powerd デーモンは停止について syslogd(1M) に通知し、syslogd(1M) はこれをブロードキャストします。</p>						
オプション	<p>以下のオプションを使用することができます。</p> <p><code>-n</code> 通知なし。デーモンは syslogd(1M) に通知しないでシステムを停止します。</p>						
ファイル	<code>/etc/power.conf</code> 電源管理設定情報ファイル						
属性	以下の属性については、attributes(5) を参照してください。						
	<table border="1"> <thead> <tr> <th>属性タイプ</th> <th>属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWpmu</td> </tr> <tr> <td>インタフェースの安定性</td> <td>不安定</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWpmu	インタフェースの安定性	不安定
属性タイプ	属性値						
使用条件	SUNWpmu						
インタフェースの安定性	不安定						
関連項目	<p>pmconfig(1M), syslogd(1M), power.conf(4), attributes(5), cpr(7), pm(7D)</p> <p>『電源管理システムユーザーマニュアル』</p>						

poweroff(1M)

名前	halt, poweroff - プロセッサの停止				
形式	<code>/usr/sbin/halt [-dlnqy]</code> <code>/usr/sbin/poweroff [-dlnqy]</code>				
機能説明	<p>halt および poweroff ユーティリティは、保留されている情報をディスクに書き出してから、プロセッサを停止させます。poweroff ユーティリティは、可能であれば、マシンの電源を自動的に切断します。</p> <p>halt および poweroff ユーティリティは通常、システム停止をシステムログデーモン syslogd(1M) に送信し、ログインアカウントファイル /var/adm/wtmpx に停止情報を記録します。ただし、-n または -q オプションが指定されている場合、これらのアクションは禁止されます。</p>				
オプション	<p>次のオプションを指定できます。</p> <ul style="list-style-type: none"> -d 再起動の前にシステムクラッシュダンプを強制的に実行します。システムクラッシュダンプの設定については、dumpadm(1M) のマニュアルページを参照してください。 -l halt を実行したユーザーに関するメッセージを、システムログデーモン syslogd(1M) に送信しないようにします。 -n 停止前の sync(1M) を行いません。 -q 即時停止。通常の停止手続きを実行しません。 -y ダイアルアップ端末からでも、システムを停止します。 				
ファイル	/var/adm/wtmpx ユーザーアクセスと管理情報の履歴				
属性	<p>次の属性については、attributes(5) のマニュアルページを参照してください。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">属性タイプ</th> <th style="text-align: center;">属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWcsu</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWcsu
属性タイプ	属性値				
使用条件	SUNWcsu				
関連項目	dumpadm(1M), init(1M), reboot(1M), shutdown(1M), sync(1M), syslogd(1M), inittab(4), attributes(5)				
注意事項	halt ユーティリティおよび poweroff ユーティリティは、/etc/rc<num>.d のスクリプトを実行しません。また、inittab(4) の停止アクションも実行しません。システムサービスを完全に停止させるためには、shutdown(1M) を使用するか、または init(1M) を使用して Solaris システムを再起動してください。				

名前	prodreg – Solaris Product Registry の管理
形式	prodreg [--help] [<i>subcommand operand ...</i>]
機能説明	<p>prodreg は、Solaris Product Registry のコンポーネントのブラウザ、登録解除、およびアンインストールに使用するユーティリティです。</p> <p>一部のインストーラは、libwsreg(3LIB) を使って情報を登録します。Solaris Product Registry には、このインストール済みソフトウェアに関する情報が含まれています。</p> <p>Solaris Product Registry のデータベースのパスは、インストール済みファイルシステムのルートを基点としています (通常、ファイルシステムのルートを基点とする場合はスラッシュ (/) で始まります)。ただし、Live Upgrade インストールの実行時には、代替ルートを基点とする異なった Solaris Product Registry インストールデータベースが使用されることがあります。live_upgrade(5) を参照してください。</p> <p>Registry データベースは、インストーラに、インストール済みソフトウェアの情報を伝達します。Registry や prodreg ユーティリティが直接インストールまたはアンインストールを実行することはありません。prodreg は、prodreg 自体またはその他の方法で起動され、外部で実行されるインストーラをサポートします。</p> <p>prodreg コマンドでは、サブコマンドを使って、コマンド行または GUI ビューアと同等の機能を使用できます。GUI ビューアには 2 つのバージョンがあります。1 つは Java Swing GUI (デフォルト)、もう 1 つは Java AWT GUI です。後者は、Java Swing がサポートされていない環境で使用します。</p> <p>unregister サブコマンドは、CLI (コマンド行インタフェース) からは実行できても GUI からは実行できない唯一の機能です。たとえば、アンインストーラを使用せずに手でソフトウェアを削除した場合、Product Registry に破損が生じる可能性があります。削除されたソフトウェアのエントリが、この後に実行されるインストーラを混乱させることがあるためです。このような場合は、unregister サブコマンドを使って無効なエントリを強制的に削除します。再帰オプションや強制オプションを使用してソフトウェアの登録を解除するときは、Registry 内の有効なエントリを誤って削除しないように注意してください。</p> <p>prodreg コマンドは、あくまでも、その時点の Registry の内容を表示するものです。これは、GUI から起動した場合もコマンド行インタフェースブラウザから起動した場合も変わりません。prodreg ビューアの起動後または起動と同時に、ソフトウェアをインストールまたはアンインストールした場合、コマンドの出力結果と実際の Solaris Product Registry の内容が一致しなくなることがあります。</p>
サブコマンド	<p>prodreg コマンドには、サブコマンドを指定せずにオプションを指定できます。サブコマンドの指定を省略すると、swing サブコマンドが指定されたものと見なされます。</p> <p>サポートされているサブコマンドは次のとおりです。</p> <p>awt Java awt GUI を起動します。</p>

prodreg(1M)

awt サブコマンドの形式は次のとおりです。

```
awt [-R alt_root | --help]
```

browse

コマンド行インタフェースから、テキスト形式で Solaris Product Registry の内容を表示します。この出力結果から、Product Registry ツリー内のすべてのコンポーネントの情報を、親コンポーネントおよび子コンポーネントの情報を含めて確認できます。このサブコマンドを繰り返し指定することにより、Product Registry 内を対話的にブラウズできます。

データベースコンポーネントはツリー形式で表され、1つ以上の子コンポーネントを持つ場合もあります。ルート以外のコンポーネントは、親コンポーネントを1つ持ちます。このサブコマンドを実行すると、Solaris Product Registry データベース内の指定のコンポーネントの親コンポーネントおよび子コンポーネントが表示されます。

prodreg browse サブコマンドを実行するたびに、Registry 内のコンポーネントとその子コンポーネント、さらにそのコンポーネントからルートに至る親(祖先)コンポーネントが表示されます。prodreg GUI を使用する場合は、任意のノードを選択し、クリックして展開します。コマンド行インタフェースでも同様の処理が可能です。あるノードの子ノードを順番にブラウズしていくことで、Registry を展開できます。

prodreg browse コマンドを実行して、まず Registry のルートをブラウズします。その後、コンポーネントを順に選択して、ブラウズする範囲を展開していきます。このように対話形式でブラウズする場合はブラウズ番号を指定すると便利ですが、スクリプト内に指定することはできません。ブラウズ番号は、セッションまたは使用するシステムごとに異なります。なぜなら、ブラウズ番号は、特定のシステム上で特定のユーザーによって最初に使用されるときに生成されるものだからです。

browse サブコマンドの形式は次のとおりです。

```
browse [-R alt_root] [-u uuid [-i instance | -p location]]  
browse [-R alt_root] -n bnum [-i instance | -p location]  
browse [-R alt_root] -m name  
browse --help
```

各コンポーネントについて、次の情報が出力されます。

BROWSE #	コンポーネントのブラウズ番号です。prodreg browse や info サブコマンドの引数として使用できます。
+/-/.	子コンポーネントがすべて非表示になっているコンポーネントは、「+」で表されます。1個以上の子コンポーネントが表示されているコンポーネントは、「-」で表されます。子コンポーネントを持たないコンポーネントは、「。」で表されます。下の世代のコンポーネントほど、フィールド内の記号の位置が右寄りに表示されます。
UUID	コンポーネントの一意の識別子です。

#	コンポーネントのインスタンス番号です。ソフトウェアコンポーネントを複数回インストールした場合、ソフトウェアレジストリによって、各コンポーネントに1つずつ一意のインスタンスが割り当てられます。
NAME	Solaris Product Registry データベース内の各コンポーネントに割り当てられている、各言語対応の名前です。この名前は、Registry 内で一意とは限りません。同じ名前を持つコンポーネントが複数存在している可能性があります。

`browse` サブコマンドで Registry データベースを表示するときは、4 種類のオプションを指定できます。あるコンポーネントに複数のインスタンスがある場合、どのインスタンスの情報が出力されたのか判別できないことがあります。要求は明確にする必要があります。-u オプションや -n オプションを使って、オペランド `instance` または `location` を指定すると、出力する情報を明確にできます。

- オペランド情報の指定がない場合は、レジストリツリーのルートとその子コンポーネントが表示されます。ここを起点にして、対話形式で Registry データベース全体をブラウズできます。
- ブラウズ番号が指定された場合は、該当コンポーネントが表示されます。
- `uuid` が指定された場合は、該当コンポーネントが表示されます。
- 名前が指定された場合は、該当コンポーネントが表示されます。

info

コンポーネントの識別情報を指定して、その属性を表示します。Solaris Product Registry 内のすべてのコンポーネントについて情報を表示できます。

Product Registry 内のコンポーネントは、`name` 文字列と単一の `value` 文字列から構成される属性を持っています。

このサブコマンドは、Solaris Product Registry 内のコンポーネントの属性情報を出力します。コンポーネントの指定方法は、UUID、名前、ブラウズ番号のいずれかを必ず指定するという点を除けば、`browse` サブコマンドの場合と同じです。

このサブコマンドで指定したコンポーネントが複数のインスタンスを持つ場合や、Registry 内に同じ名前のコンポーネントが複数存在する場合は、コンポーネントの指定があいまいになります。このような場合、属性情報ではなく、指定した条件に合致するコンポーネントの一覧が出力されます。

デフォルトでは、コンポーネントの属性が1行に1つずつ出力されます。まず属性名に続いてコロン(:)と空白文字が出力され、RETURN キーを押すと属性値が出力されます。その他、指定できるオプションとして、-a および -d があります。

`info` サブコマンドの形式は次のとおりです。

```
info --help
info [-R alt_root] -u uuid [-i instance | -p location]
info [-R alt_root] -n bnum [-i instance | -p location]
info [-R alt_root] -m name [-a attr | -d ]
```

`help` | `--help` | `-?`

ヘルプテキストを表示します。

prodreg(1M)

help サブコマンドの形式は次のとおりです。

```
help | --help | -?
```

swing

Java Swing GUI を起動します。Java Swing GUI を使用できない環境では失敗します。

swing サブコマンドの形式は次のとおりです。

```
swing [-R alt_root | --help]
```

version | --version | -V

現在のバージョン文字列を出力します。

version サブコマンドの形式は次のとおりです。

```
version | --version | -V
```

unregister

Registry 内のエントリの登録を解除します。

Solaris Product Registry からコンポーネントを削除します。-u オプションを使って、削除するコンポーネントの UUID を指定できます。ただし、同じ UUID のコンポーネントが複数存在する場合、削除処理は実行されません。代わりに、指定した UUID に一致するインスタンスの一覧が表示されます。この場合は、登録を解除するコンポーネントインスタンスを一意に指定できる -p または -i オプションを使って、サブコマンドを再発行してください。

Registry 内に、登録を解除しようとするコンポーネントに依存するコンポーネントが存在する場合、unregister サブコマンドは失敗します。

ユーザーが Registry に対する書き込み権を持っていない場合、unregister サブコマンドは失敗します。これについて

は、wsreg_can_access_registry(3WSREG) を参照してください。ユーザーが Solaris Product Registry に登録されたコンポーネントではなくシステムコンポーネントの登録を解除しようとした場合、unregister サブコマンドは失敗します。システムコンポーネントに該当するのは、PKG 属性を持つコンポーネントと、いくつかの特別な Registry ノードです。次にその一部を紹介します。

UUID	Name
root	System Registry
a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b	Solaris System Software
8f64eabf-1dd2-11b2-a3f1-0800209a5b6b	Unclassified Software
b96ae9a9-1dd1-11b2-a3f2-0800209a5b6b	System Software Localizations
b1c43601-1dd1-11b2-a3f2-0800209a5b6b	Additional System Software
a8dcab4f-1dd1-11b2-a3f2-0800209a5b6b	Software Localizations

unregister サブコマンドの -f オプションを使用するときは、これから登録を解除するコンポーネントに依存するコンポーネントがないかどうかを事前に注意深く確認してください。-r オプションは、-f オプション以上に注意を要するオプ

prodreg(1M)

ションです。このオプションを指定してコンポーネントの登録を解除すると、そのコンポーネントに依存するすべての子コンポーネントおよびソフトウェアコンポーネントの登録が解除されます。依存コンポーネントを確認するには、次のコマンドを使用します (*uuid* には UUID を指定)。

```
prodreg info -u uuid -a "Dependent Components"
```

必須コンポーネントを確認するには、次のコマンドを使用します。

```
prodreg info -u <uuid> -a "Required Components"
```

コンポーネントの名前、UUID、およびインスタンスが出力されます。

unregister サブコマンドの形式は次のとおりです。

```
unregister [-R alt_root] [-fr] -u uuid [-p location | -i instance]  
unregister --help
```

uninstall

アンインストールプログラムを起動します。

Registry 内のコンポーネントが個別にアンインストーラを備えている場合があります。-u オプションでコンポーネントを指定すると、そのアンインストーラが起動します。この指定は、アンインストーラを持たないコンポーネントに対しては無効です。-u オプションで指定された UUID を持つコンポーネントが複数存在する (そのコンポーネントのインスタンスが複数インストールされている) 場合、該当するコンポーネントインスタンスの一覧が出力されます。このように -u オプションで指定された UUID があいまいな場合は、-i または -p オプションを指定して、サブコマンドを再発行してください。ただし、これらからアンインストールするコンポーネントに依存するコンポーネントが存在する場合、コマンドは失敗します。

-x オプションを指定してアンインストーラを起動することもできます。この場合、依存コンポーネントを持つコンポーネントをアンインストールするかどうかの確認は行われません。

ユーザーが Registry に対する書き込み権を持っていない場合、uninstall コマンドは実行されません。これについては、`wsreg_can_access_registry(3WSREG)` を参照してください。

uninstall コマンドの形式は次のとおりです。

```
uninstall [-R alt_root] [-f] -u uuid -p location  
uninstall [-R alt_root] -i instance[arguments ...]  
uninstall --help
```

オプション **awt** サブコマンドは次のオプションをサポートします。

<code>--help</code>	ヘルプテキストを表示します。ビューアは起動しません。
<code>-R <i>alt_root</i></code>	指定された代替ルートを使って、GUI ビューアに表示するデータベースを検索します。
	<code><i>alt_root</i></code> の仕様については、「オペランド」を参照してください。

browse サブコマンドは次のオプションをサポートします。

prodreg(1M)

-help	ヘルプテキストを表示します。browse サブコマンドは実行しません。
-i instance	指定されたコンポーネントインスタンスを出力します。
-m name	指定された名前のコンポーネントインスタンスを出力します。
-n bnum	指定されたブラウザ番号のコンポーネントインスタンスを出力します。
-p location	指定された位置にインストールされているコンポーネントインスタンスを出力します。コンポーネントがインストールされている位置を確認するには、info サブコマンドを使用します。
-R alt_root	指定された代替ルートを使ってデータベースを検索します。
-u uuid	指定された UUID のコンポーネントインスタンスを出力します。

info サブコマンドは次のオプションをサポートします。

-a attr	オペランド attr で指定された属性だけを出力します。指定されたコンポーネントの属性をすべて出力するわけではありません。
-d	isDamaged という名前の属性だけを出力します。指定されたコンポーネントの属性をすべて出力するわけではありません。
--help	ヘルプテキストを出力します。info サブコマンドは実行しません。
-i instance	オペランド instance により、同じ uuid またはブラウザ番号を持つ複数のコンポーネントインスタンスを識別します。
-m name	オペランド name により、Registry 内の 1 つ以上のコンポーネントを指定します。
-n bnum	ブラウザ番号 bnum のコンポーネントインスタンスの属性を出力します。該当するインスタンスが複数存在する場合は、-i または -p オプションを指定して、あいまいさを解決する必要があります。
-p location	インストール先を指定して、同じ UUID またはブラウザ番号を持つ複数のコンポーネントインスタンスを識別します。
-R alt_root	指定された代替ルートを使って、データベースを検索します。
-u uuid	指定された UUID を持つコンポーネントインスタンスの属性を出力します。該当するインスタンスが複数存在する場合は、-i または -p オプションを指定して、あいまいさを解決する必要があります。

swing サブコマンドは次のオプションをサポートします。

--help	ヘルプテキストを出力します。swing サブコマンドは実行しません。
--------	------------------------------------

`-R alt_root` 指定された代替ルートを使って、データベースを検索します。

`uninstall` サブコマンドは次のオプションをサポートします。

`-f` 強制的にアンインストールを行います。指定された UUID を持つコンポーネントインスタンスが複数存在する場合は、該当するすべてのコンポーネントインスタンスをアンインストールします。

`--help` ヘルプテキストを出力します。`uninstall` サブコマンドは実行しません。

`-i instance` 指定された UUID のあいまいさを解決します。

`-p location` 指定された UUID のあいまいさを解決します。`location` には、ソフトウェアコンポーネントがインストールされている位置を指定します。

`-R alt_root` 指定された代替ルートを使って、データベースを検索します。

`-u uuid` 指定された UUID を持つコンポーネントの登録を解除します。同じ UUID を持つコンポーネントが複数回インストールされている場合は、`-i` または `-p` オプションを使って、指定のあいまいさを解決します。

`unregister` サブコマンドは次のオプションをサポートします。

`-f` 強制的に登録を解除します。指定されたコンポーネントに依存するコンポーネントが存在する場合も登録を解除します。

`--help` ヘルプテキストを出力します。`unregister` サブコマンドは実行しません。

`-i instance` 指定された UUID のあいまいさを解決します。

`-p location` 指定された UUID のあいまいさを解決します。`location` には、ソフトウェアコンポーネントがインストールされている位置を指定します。

`-r` 指定されたコンポーネントの子コンポーネントと依存コンポーネントを含めて、再帰的にコンポーネントの登録を解除します。

`-R alt_root` 指定された代替ルートを使って、データベースを検索します。

`-u uuid` 指定された UUID を持つコンポーネントの登録を解除します。同じ UUID を持つコンポーネントが複数回インストールされている場合は、`-i` または `-p` オプションを使って、指定のあいまいさを解決します。

オペランド 次のオペランドがサポートされています。

`alt_root` 代替ルートを示すファイルのパス名です。Solaris Product Registry データベースは、代替ルートを基点としています。この位置を基点とするデータベースが存在しない場合は、自動的に作成されます。

prodreg(1M)

<i>attr</i>	属性名を指定する info サブコマンド専用のオペランドです。attr がコンポーネントに関連している場合、属性名と値が表示されません。
<i>bnum</i>	ブラウズ番号です。 Solaris Product Registry の各コンポーネントには、対話形式でブラウズを行うユーザー向けにブラウズ番号が割り当てられます。ただし、この番号は、システムを再起動または再インストールすると変更されます。この番号を browse または info サブコマンドの補助以外の目的で使用または保存することは避けてください。ブラウズ番号は、prodreg browse サブコマンドを実行すると出力されます。これ以外の値を、browse または info サブコマンドの入力値として指定することはできません。
<i>instance</i>	ソフトウェアを複数の場所にインストールした場合、Solaris Product Registry は、それぞれに一意のインスタンス番号を割り当てます。Registry 内の各コンポーネントのインスタンス番号を表示するには、browse サブコマンドを使用します。このオペランドは、インストール済みソフトウェアの (おそらく同一ではない) 複数のコピーが存在する場合、これらを識別する目的で使用しません。
<i>location</i>	ファイルシステム内の特定のファイルまたはディレクトリのパスです。登録されたソフトウェアのインストール先を示します。たとえば、ソフトウェアのインストール先が /usr/local からの相対位置である場合、このオペランドの値は /usr/local になります。インストール先の情報は、インストーラによって使用されません。インストーラの位置を指定したり、ソフトウェアコンポーネントのインスタンスが複数存在する場合は、指定のあいまいさを解決する目的でも使用されます。
<i>name</i>	Solaris Product Registry 内の各ソフトウェアコンポーネントには、名前が割り当てられています。この名前は、browse サブコマンドで確認できます。また、一部のサブコマンドではオペランドとしてソフトウェアの名前を指定できます。ただし、こうした名前が一意であるとは限りません。指定された名前のコンポーネントが複数存在する場合、条件に合致するコンポーネントの一覧が出力されます。名前が各言語対応になっている場合、すなわち、言語設定によって名前が異なる場合もあります。
<i>uuid</i>	Solaris Product Registry 内の各ソフトウェアコンポーネントには、一意の識別子が割り当てられています。この識別子が、Registry データベース内のエントリにアクセスする際のハンドルになります。UUID は、インストールされているコンポーネントインスタンスの数やコンポーネントの各言語対応の名前に関係なく、そのコンポーネントに対応しています。

例 1 prodreg コマンドによるブラウズ

ブラウズを実行するには、prodreg の browse サブコマンドを使用します。この要求を繰り返し実行することにより、GUI を使ってコンポーネントおよびそれに含まれるコレクションを展開していく場合と同様に、ツリーの内容を確認することができます。ブラウズ番号を使ったブラウズは、こうした再帰的ブラウズ処理以外では実行しないでください。ブラウズ番号は、ブラウズ操作の結果として生成されるものです。

引数なしで browse サブコマンドを実行すると、Registry のトップからブラウズすることになります。出力内容は、そのシステムにインストールされているソフトウェアによって異なります。

```
$ prodreg browse
BROWSE # +/-/.  UUID                                     #  NAME
=====  =====  =====
1          -      root                                     1  System
                                           Registry
2          +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9
                                           System
                                           Software
3          +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                           Software
```

ルートコンポーネントとその子コンポーネントのブラウズ番号、UUID、インスタンス番号、および名前が出力されます。コンポーネントの祖先(ルートに至る親コンポーネント)も表示されます。「+/-/.」列は、ツリー内のコンポーネントが展開済みの親コンポーネントであるか(-)、子コンポーネントを持つ子コンポーネントであるか(+)、子コンポーネントを持たないコンポーネントであるか(.)を示します。

例 2 ツリー内のコンポーネントの情報を要求

UUID、名前、およびブラウズ番号の各フィールドを使って、ツリー内のコンポーネントの情報を要求できます。次の例では、UUID を指定してコンポーネントをブラウズします。

```
$ prodreg browse -u a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b
BROWSE # +/-/.  UUID                                     #  NAME
=====  =====  =====
1          -      root                                     1  System
                                           Registry
2          -      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9
                                           System
                                           Software
4          +      b96ae9a9-1dd1-11b2-a3f2-0800209a5b6b  1  System
                                           Software
                                           Localizations
5          +      SUNWCall                               1  Entire
                                           Distribution
```

例 3 名前を指定してノードをブラウズ

次の例では、名前を指定してノードをブラウズします。

prodreg(1M)

例 3 名前を指定してノードをブラウズ (続き)

```
$ prodreg browse -m "System Software Localizations"
BROWSE # +/-/. UUID # NAME
===== =====
1 - root 1 System Registry
2 - a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b 1 Solaris 9 System Software
4 - b96ae9a9-1dd1-11b2-a3f2-0800209a5b6b 1 System Software Localizations
316 . SUNWceuow 1 Central Europe OW Support
317 . SUNWcsfw 1 Simplified Chinese freeware message
318 . SUNWceuox 1 Central Europe 64-bit OS Support
```

例 4 再帰的ブラウズ

追加出力は省略されています。再帰的ブラウズには、ブラウズ番号を使用すると便利です。ブラウズ番号は、prodreg コマンドを実行するシステム、コマンドを実行するユーザー、およびコマンドを実行するログインセッションによって異なるため、保存できません。

```
$ prodreg browse -n 3
BROWSE # +/-/. UUID # NAME
===== =====
1 - root 1 System Registry
2 - a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b 1 Solaris 9 System Software
5 - SUNWCall 1 Entire Software Distribution
6 . SUNWrsmo 1 RSMPI Operations Registration Module
7 + SUNWCjvx 1 JavaVM (64-bit)
8 . SUNWrsmx 1 Remote Shared Memory (64-bit)
9 + SUNWCacc 1 System
```

例 4 再帰的ブラウズ (続き)

Accounting

例 5 あいまいな値によるブラウズ

要求された値があいまいな場合、条件に合致するインスタンスの一覧が表示されます。次の例では、同じ名前を持つ異なったソフトウェアコンポーネントが2つ存在します。

```
$ ./prodreg browse -m JavaVM
The request failed because multiple components correspond to the
criteria given. Use the list of possible components given below,
select one and try again.
```

BROWSE #	+/-/.	UUID	#	NAME
12	.	org.spybeam.javavm	1	JavaVM
51	.	SUNWCjv	1	JavaVM

次のいずれかの要求を再発行してください。

```
$ prodreg browse -u SUNWCjv
```

または

```
$ prodreg browse -u org.spybeam.javavm
```

例 6 複数回インストールされたソフトウェアのブラウズ

特定のソフトウェアコンポーネントが複数回インストールされている場合も、要求があいまいになります。次の例では、Example ソフトウェアが3回インストールされています。

```
$ prodreg browse -m Example
The request failed because multiple components correspond to the
criteria given. Use the list of possible components given below,
select one and try again.
```

BROWSE #	+/-/.	UUID	#	NAME
7	.	org.spybeam.example	2	Example
7	.	org.spybeam.example	3	Example
7	.	org.spybeam.example	1	Example

```
The component requested could not be found.
```

例 7 特定のインスタンスによるブラウズ

特定のインスタンスを指定して要求を繰り返し実行することにより、あいまいさを解決できます。-p オプションでインストール先を指定する方法もあります。次の例では、Example ソフトウェアの1番目のインスタンスをブラウズします。

```
$ prodreg browse -u org.spybeam.example -i 1
```

例 8 info サブコマンドの使用

コンポーネントのインストール先やその他の属性情報を取得したい場合は、`info` サブコマンドを使用します。このサブコマンドにも、あいまいさを解決するオプションを指定できます。実行すると、コンポーネントのすべての属性が 1 行に 1 つずつ出力されます。

```
$ prodreg info -m Example
The request failed because multiple components correspond to the
criteria given. Use the list of possible components given below,
select one and try again.
BROWSE # +/-/. UUID                                # NAME
=====
7      .      org.spybeam.example                    2 Example
7      .      org.spybeam.example                    3 Example
7      .      org.spybeam.example                    1 Example
The component requested could not be found.
```

次のコマンドでは、`Example` コンポーネントのインスタンス 1 の情報がすべて出力されます (出力結果は省略)。

```
$ prodreg info -u org.spybeam.example -i 1
```

例 9 インストール先の情報を取得

コンポーネントのインストール先やその他の属性情報を取得したい場合は、`info` サブコマンドを使用します。`browse` サブコマンドの場合と同様、`info` サブコマンドにも、あいまいさを解決するオプションを指定できます。`info` サブコマンドを実行すると、コンポーネントのすべての属性が 1 行に 1 つずつ出力されます。単一の属性を要求することもできます。

次のコマンドでは、インストール先の属性値が出力されます。

```
$ prodreg info -n 23 -a Location
```

例 10 損傷を受けたソフトウェアの特定および登録解除

所定のアンインストーラを使用せずにインストール済みのソフトウェアを削除すると、`Registry` 内のソフトウェアが損傷を受ける可能性があります。コンポーネントが損傷を受けた場合、すでに削除されたソフトウェアがまだインストールされているものと認識されます。所定のアンインストーラを実行せずにファイルやパッケージを直接削除すると、同様にコンポーネントが損傷を受けることがあります。通常、インストールプログラムを使ってインストールしたソフトウェアは、所定のアンインストールプログラムを使ってアンインストールすることになっています。

次の例では、損傷を受けたソフトウェアコンポーネントを特定し、再インストールできる状態に修復します。

`Examplesoft` をブラウズすると、次の内容が出力されます。

```
$ prodreg browse -m Examplesoft
BROWSE # +/-/. UUID                                # NAME
=====
```


例 10 損傷を受けたソフトウェアの特定および登録解除 (続き)

```

1      -      root                                1  System
                                           Registry
2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b 1  Solaris 9
                                           System
                                           Software
3      +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b 1  Unclassified
                                           Software
4      -      95842091-725a-8501-ef29-0472985982be 1  ExampleSoft
233    .      90209809-9785-b89e-c821-0472985982be 1  Example Doc
234    .      EXSOzzt                               1
235    .      EXSOblob                              1  Example Data

```

Examplesoft の子コンポーネント、EXSOzzt に注目してください。登録済みであるにもかかわらず、ソフトウェアのパッケージコンポーネント名が表示されていません。この場合、Examplesoft ソフトウェアが損傷を受けている可能性があります。次のコマンドで損傷の有無を確認します。

```

$ prodreg info -u 95842091-725a-8501-ef29-0472985982be \
-i 1 -d
isDamaged=TRUE

```

isDamaged が TRUE なので、Examplesoft の一部が損傷を受けています。次のコマンドで、Examplesoft を構成するパッケージを一覧表示します。

```

$ prodreg info \
-u 95842091-725a-8501-ef29-0472985982be\
-i 1 -a PKGS pkgs:
EXSOzzt EXSOblob

```

pkginfo コマンドで、EXSO がインストールされているかどうかを確認します。

```

$ pkginfo EXSOzzt
ERROR: information for "EXSOzzt" was not found
$ pkginfo EXSOblob
application EXSOblob      Example Data

```

一連のコマンドの実行結果から、EXSOzzt パッケージが、おそらく pkgrm コマンドによって削除されていることがわかります。この場合、Examplesoft ソフトウェアは正常に機能しません。ソフトウェアを修復するには、Examplesoft に登録されているアンインストーラを実行する必要があります。アンインストーラは、ソフトウェアの登録を解除し、pkgrm コマンドを実行します。この操作には root ユーザーのアクセス権が必要です。したがって、通常、アンインストーラの実行にも、root ユーザーのアクセス権が必要になります。

```

# prodreg uninstall -u 95842091-725a-8501-ef29-0472985982be -i 1
The install program requested could not be found.

```

アンインストールプログラムにアクセスして、ソフトウェアをアンインストールすることができませんでした。アンインストールプログラムが手動で削除されている可能性があります。そこで、uninstallprogram 属性を要求して、アンインストーラの位置を確認します。

```

$ prodreg info -m ExampleSoft -a uninstallprogram
uninstallprogram: /usr/bin/java -mx64m -classpath
/var/sadm/prod/org.example.ExampleSoft/987573587 uninstall_ExampleSoft

```

例 10 損傷を受けたソフトウェアの特定および登録解除 (続き)

登録された位置にアンインストーラがあるかどうかを確認します。

```
# ls /var/sadm/prod/org.example.ExampleSoft/987573587
/var/sadm/prod/org.example.ExampleSoft/987573587:
No such file or directory
```

所定の位置にアンインストーラがない場合、2通りの対処法があります。1つは、バックアップストレージからアンインストーラを読み込んで手動で実行する方法です。Registry に格納されているコマンド行を使用してください。

```
# /usr/bin/java -mmx64m -classpath \
  /var/sadm/prod/org.example.ExampleSoft/987573587 \
  uninstall_ExampleSoft
```

バックアップストレージにない場合は、手動でソフトウェアの登録を解除します。

```
# prodreg unregister -u 95842091-725a-8501-ef29-0472985982be -i 1
```

このコマンドでは、EXSOblob パッケージは削除されません。EXSOblob パッケージは手動で削除する必要があります。

```
# pkgrm EXSOblob
```

例 11 複数のコンポーネントの削除

コンポーネント A は子コンポーネント B および C を持ち、子コンポーネント C は子コンポーネント D および E を持っているとします。この例では、これらのコンポーネントを一括削除します。この処理は、コンポーネントの階層全体を再インストールする必要があるときに、アンインストーラが見つからない、あるいは実行できない場合に便利です。

```
$ prodreg browse -u UUID-of-C
BROWSE # +/-/.  UUID                                     #  NAME
===== =====
1      -      root                                     1  System
                                           Registry
2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9
                                           System
                                           Software
3      +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                           Software
1423   -      UUID-of-A                                     1  Example A
1436   .      UUID-of-B                                     1  Example B
1437   -      UUID-of-C                                     1  Example C
1462   .      UUID-of-D                                     1  Example D
1463   .      UUID-of-E                                     1  Example E

# prodreg uninstall -u UUID-of-A -i 1
```

uninstall サブコマンドが失敗することもあります。たとえば、Java クラスが削除されている、ユーザーのアクセス権が十分でない、システムに Java ソフトウェアがインストールされていない、などの条件下で失敗します。この問題には、再帰的登録解除のコマンドで対処できます。しかし、このコマンドは、対象コンポーネントの子コンポーネントおよび対象コンポーネントに依存するコンポーネントの登録をすべて解

例 11 複数のコンポーネントの削除 (続き)

除するため、使用に際してはきわめて慎重に臨む必要があります。誤って必要なコンポーネントの登録を解除してしまわないように、対象コンポーネントの全情報をあらかじめ確認しておくことをお勧めします。次の例では、UUID-of-A を指定してコンポーネントの情報を表示します。

```
$ prodreg info -u UUID-of-A
Title: Example A Software
Version: 5.8.0.2001.11.02
Location: /usr
Vendor: Example Vendor
uninstallprogram: /usr/bin/java -mx64m -classpath
/var/sadm/prod/org.example.ExampleA/90820965 uninstall_ExampleA
vendorurl: http://www.example.org
description: Example A Software has many uses
Supported Languages: en
```

```
Child Components:
Name                               UUID                               #
-----
Example B                           UUID-of-B                           1
Example C                           UUID-of-C                           1
```

```
Required Components:
Name                               UUID                               #
-----
Example B                           UUID-of-B                           1
Example C                           UUID-of-C                           1
```

「Dependent Components」フィールドが表示されていないので、Example A に依存するソフトウェアは存在しません。念のため、UUID-of-B と UUID-of-C の依存コンポーネントおよび子コンポーネント、UUID-of-B または UUID-of-C の子コンポーネントに依存するコンポーネントをすべて確認しておきます。

ブラウズツリーを表示して UUID-of-A のすべての子孫を確認できたら、今度は、Example A のすべての子孫の依存コンポーネント属性を確認します。

```
$ prodreg info -u UUID-of-B -i 1 -a "Dependent Components"
Dependent Components:
Name                               UUID                               #
-----
Example A                           UUID-of-A                           1

$ prodreg info -u UUID-of-C -i 1 -a "Dependent Components"
Dependent Components:
Name                               UUID                               #
-----
Example A                           UUID-of-A                           1

$ prodreg info -u UUID-of-D -i 1 -a "Dependent Components"
Dependent Components:
Name                               UUID                               #
-----
Example C                           UUID-of-C                           1
```

prodreg(1M)

例 11 複数のコンポーネントの削除 (続き)

```
$ prodreg info -u UUID-of-E -i 1 -a "Dependent Components"
Dependent Components:
Name                               UUID                               #
-----
Example C                           UUID-of-C                           1
```

Example A の再帰的登録解除を実行しても、Example A とその子孫の登録以外は解除されません。

```
# prodreg unregister -r -u UUID-of-A -i 1
```

例 12 損傷を受けたコンポーネントの再インストール

依存するソフトウェアを持つコンポーネント、Software ZZZ が損傷を受けたので、再インストールする必要があります。再インストールするには、Software ZZZ の登録を解除しなければなりません。

この場合、まず Software ZZZ の依存コンポーネントを確認します。

```
$ prodreg info -m "Software ZZZ" -a "Dependent Components"
Dependent Components:
Name                               UUID                               #
-----
Software Foobar                    d9723500-9823-1432-810c-0100e09832ff 1
```

通常、Software ZZZ の登録を解除するには、依存コンポーネントである Software Foobar をアンインストールする必要があります。しかし、Software Foobar の再インストールが不可能または不適切であることがわかっているため、Software ZZZ の再帰的登録解除は実行できません。再帰的登録解除を実行すると、Software Foobar の登録まで解除されてしまいます。この場合は、Software ZZZ の強制登録解除を実行します。Software ZZZ の UUID は、90843fb1-9874-3a20-9b88-984b32098432 です。

```
# prodreg unregister -f -u 90843fb1-9874-3a20-9b88-984b32098432 -i 1
```

続いて、Software ZZZ を再インストールします。

```
# /usr/bin/java -cp /usr/installers/org.example.softwarezzz
```

バグ ソフトウェアを手動で削除したり、pkgrm(1M) を使用して直接削除したりすると、Registry の情報が一致しくなくなります。Registry の損傷を防ぐため、インストールプログラムを使ってインストールしたソフトウェアはアンインストールプログラムを使ってアンインストールしてください。

環境変数 次の環境変数は、prodreg の実行に影響を及ぼします。

PKG_INSTALL_ROOT システムの PKG_INSTALL_ROOT パスとして使用するディレクトリの完全パス名を定義します。この環境変数が定義されている場合、すべての製品およびパッケージの情報ファイルは、最初に

prodreg(1M)

PKG_INSTALL_ROOT パスで検索されます。この環境変数が定義されていない場合、デフォルトのシステムパス / が使用されま
す。

終了ステータス 次の終了ステータスが返されます。

- 0 正常終了
- >0 エラーが発生した

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWwsr2, SUNWwsrv
インタフェースの安定性	開発中

関連項目 pkgadd(1M), pkgrm(1M), wsreg_can_access_registry(3WSREG),
libwsreg(3LIB), live_upgrade(5), attributes(5)

『Application Packaging Developer's Guide』

注意事項 prodreg の GUI やコマンド行インタフェースには、Solaris Product Registry と
パッケージデータベースが両方とも表示されます。どちらも Registry 内のコンポーネ
ントのように見えますが、一部、登録解除やアンインストールができないものがあり
ます。所定のアンインストーラを持たないパッケージは、prodreg uninstall コマ
ンドでアンインストールできません。Solaris パッケージは、prodreg unregister
コマンドで登録解除できません。GUI または CLI の prodreg ビューアに表示されな
いパッケージは、pkgrm(1M) コマンドで削除する必要があります。

ソフトウェアを削除するときは、pkgrm(1M) を使って個々のパッケージを削除するの
ではなく、付属のアンインストールソフトウェアを使用することをお勧めします。ア
ンインストールソフトウェアを使用すれば、Registry 内の情報の登録解除から
パッケージの削除まで、ソフトウェアの全リソースの削除を総合的に行うことができ
ます。

prodreg uninstall コマンドは、外部プログラムを起動します。Product Registry
の代替ルートを指定する際は、こうした外部プログラムのコマンド行の規約に従う必
要があります。インストールプログラムが prodreg と同じ環境で実行されるように、
PKG_INSTALL_ROOT 環境変数を使用する方法もあります。アンインストールプログ
ラムは、通常、Java のインストールを必要とする Java クラスです。Java ソフト
ウェアが Solaris のディストリビューションから削除された場合、または見つからない
場合、Java ベースのアンインストーラは実行できません。

root ユーザーのアクセス権がないと実行できないサブコマンドは、prodreg
unregister と prodreg uninstall の 2 つだけです。なぜなら、登録の解除には
Product Registry の変更が伴い、アンインストールにはパッケージの削除が伴うから
です。これ以外のコマンドは Registry の内容を読み取るだけなので、root ユーザーの

prodreg(1M)

アクセス権がなくても実行できます。インストーラが root ユーザーのアクセス権を必要とする pkgadd(1M) や pkgrm(1M) などのコマンドを実行する場合と同様に、prodreg uninstall コマンドの実行にも root ユーザーのアクセス権が必要になります。

コンポーネントの属性については、さまざまなマニュアルに説明が記載されていますが、詳しくは『*Application Packaging Developer's Guide*』を参照してください。Solaris Product Registry 自体の属性については、次の用語集を参照してください。

依存コンポーネント そのコンポーネントに依存するコンポーネント。

位置 ソフトウェアのインストール先からの相対位置。

pkgs コンポーネントのパッケージ。これらのパッケージは、コンポーネントの登録後に pkgadd コマンドで追加され、コンポーネントの登録解除前に pkgrm コマンドで削除されます。

必須コンポーネント そのコンポーネントが依存するコンポーネント。

ソース インストール媒体。

サポート言語 登録済みタイトルがあるロケール。

タイトル prodreg browse コマンドで指定される名前。シェルの実行ロケールの言語に対応させることができます。

固有名 以前のバージョンの Solaris Product Registry で使用されていた名前。通常、Registry 内のコンポーネントのパッケージ名に設定されます。

ベンダー コンポーネントの製造元ベンダー。

バージョン コンポーネントのバージョン文字列。

Registry 内に、実際のシステムにはインストールされていないソフトウェアコンポーネントが存在する場合があります。こうしたコンポーネントはいくつかの方法で検出できますが、一番簡単なのは、info サブコマンドを使用して、コンポーネントが損傷を受けているかどうかを調べる方法です。info サブコマンドで、ソフトウェアがインストールされた場所を調べ、現在もその場所にあるかどうかを確認する方法もあります。

名前	prtcnf – システム構成の出力
形式	
SPARC	<code>/usr/sbin/prtcnf [-V] [-F] [-x] [-vpPD]</code>
x86	<code>/usr/sbin/prtcnf [-V] [-x] [-vpPD]</code>
機能説明	prtcnf コマンドはシステム構成情報を出力します。この出力には、メモリーの総量やシステム周辺機器の構成 (デバイスツリー形式) が含まれます。
オプション	次のオプションを指定できます。 <ul style="list-style-type: none"> -D デバイスツリー内のシステム周辺機器ごとに、周辺機器を管理するために使用されているデバイスドライバの名前を表示します。 -F コンソールフレームバッファが存在する場合、そのデバイスパス名を返します。フレームバッファが存在しない場合、prtcnf はゼロ以外の終了ステータスを返します。このフラグは単独で使用する必要があります。prtcnf は、コンソールの名前、フレームバッファのデバイス、またはゼロ以外の終了ステータスだけを返します。たとえば、SPARCstation 1 上のコンソールフレームバッファが SBus スロット No.3 の cgthree であれば、prtcnf は <code>/sbus@1,f80000000/cgthree@3,0</code> を返します。このオプションは、<code>/dev/fb</code> から実際のコンソールデバイスへのシンボリックリンクを作成するときに使用できます。 -p SPARC プラットフォーム上のファームウェア (PROM) または x86 プラットフォーム上のブートシステムが提供するデバイスツリーから取得した情報を表示します。このオプションを使用して表示されるデバイスツリー情報は初期構成のスナップショットであり、後で行われた構成変更は正確に反映されない場合があります。 -P 疑似デバイスの情報も出力します。デフォルトでは、疑似デバイスの情報は出力されません。 -v 詳細表示モードを指定します。 -V プラットフォームに固有な PROM (SPARC プラットフォーム) またはブートシステム (x86 プラットフォーム) のバージョン情報を表示します。このフラグは単独で使用する必要があります。出力は文字列です。文字列の形式は決まっておらず、プラットフォームに固有です。 -x 当該システム上のファームウェアが 64 ビット対応であるかどうかを報告します。既存のプラットフォームの中には、64 ビットカーネルを実行するためにファームウェアをアップグレードする必要があるものもあります。この操作が当該プラットフォームで適用されない場合、つまり、ファームウェアがすでに 64 ビット対応である場合、prtcnf は何も表示せずに、ゼロの終了ステータスを返して終了します。この操作が当該プラットフォームで適用される場合、つまり、ファームウェアが 64 ビット対応でない場合、prtcnf は標準出力に説明メッセージを表示して、ゼロ以外の終了ステータスを返して終了します。64 ビットカーネルを実行するために

prtconf(1M)

ファームウェアをアップグレードする必要があるかどうかの詳細については、プラットフォームのハードウェアマニュアルを参照してください。

このフラグは他のすべてのフラグを無効にするので、単独で使用する必要があります。

終了ステータス 次の終了ステータスが返されます。

0 正常終了

0 以外 -F オプションを指定した場合 (SPARC のみ)、ゼロ以外の戻り値は、出力デバイスがフレームバッファではないことを意味します。-x オプションを指定した場合、ゼロ以外の戻り値は、ファームウェアが 64 ビット対応ではないことを意味します。上記以外の場合には、ゼロ以外の戻り値はエラーが発生したことを意味します。

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWesu (32 ビット)
	SUNWesxu (64 ビット)

関連項目 modinfo(1M), sysdef(1M), attributes(5)

Sun ハードウェアマニュアル

SPARC のみ openprom(7D)

注意事項 prtconf コマンドの出力はシステムにインストールされている PROM のバージョンに大きく依存します。したがって、出力は潜在的にあらゆる状況の影響を受けます。

driver not attached というメッセージは、現在、デバイスの当該インスタンスにドライバが接続されていないことを意味します。一般に、ドライバは必要に応じてロードおよびインストールされ、(そして、ハードウェアのインスタンスに接続され)、デバイスが使用されないときはアンインストールとアンロードされます。

名前	prtdiag – システムの診断情報の出力				
形式	<code>/usr/platform/platform-name/sbin/prtdiag [-v] [-l]</code>				
機能説明	<p>prtdiag は、sun4u システム上で、システム設定と診断情報を表示します。</p> <p>診断情報は、システム内で障害の発生した現場交換可能ユニット(FRU) を表示します。</p> <p>prtdiag に必要なインタフェース、出力、およびディレクトリ階層における位置は、まだ確定されていません。将来のリリースで変更される可能性があります。</p> <p><i>platform-name</i> は、実装するプラットフォームの名前です。uname(1) に <code>-i</code> オプションを付けて実行すると表示されます。</p> <p>prtdiag は、Sun Enterprise 10000 サーバーで実行された場合は、診断情報と環境状態を表示しません。それらの情報を得るには、システムサービスプロセッサ(SSP)上の <code>/var/opt/SUNWssp/adm/\${SUNW_HOSTNAME}/messages</code> ファイルを参照してください。</p>				
オプション	<p>以下のオプションを使用することができます</p> <p><code>-l</code> ログの出力。システムで障害またはエラーが発生した場合は、syslogd(1M) のみに対してこの情報を出力します。</p> <p><code>-v</code> 詳細表示モード。最近発生した AC 電源障害の時刻、最近発生した重大なハードウェアエラーの情報、および (必要に応じて) 環境状態を表示します。重大なハードウェアエラーの情報は、FRU を修理する場合や、詳細な診断を行う場合に有用です。</p>				
終了ステータス	<p>以下の終了値が返されます。</p> <p>0 システム内で何も異常が検出されなかった。</p> <p>1 システム内で異常が検出された。</p> <p>2 メモリー不足などの、内的な prtdiag のエラーが発生した。</p>				
属性	<p>以下の属性については、attributes(5) を参照してください。</p> <table border="1" data-bbox="461 1436 1430 1524"> <thead> <tr> <th>属性タイプ</th> <th>属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWkvm</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWkvm
属性タイプ	属性値				
使用条件	SUNWkvm				
関連項目	uname(1), modinfo(1M), prtconf(1M), psrinfo(1M), sysdef(1M), syslogd(1M), attributes(5), openprom(7D)				

raidctl(1M)

名前	raidctl - RAID ハードウェアユーティリティー
形式	raidctl -c disk1 disk2 raidctl -d disk1 raidctl [-f] -F filename controller... raidctl -l [controller...]
機能説明	<p>raidctl ユーティリティーは、LSI1030 HW RAID コントローラの RAID ボリュームを作成、削除、または表示します。</p> <p>raidctl ユーティリティーには、基本ファイルシステムのアクセス権で制御される特権が必要です。特権を持つユーザーだけが RAID システム構成を操作できます。特権のないユーザーが RAID ボリュームを作成または削除しようとする、コマンドは EPERM で失敗します。</p> <p>オプションを指定しないと、raidctl は、すべての既存のコントローラに関して現在の RAID 構成を表示します。</p>
オプション	<p>次のオプションがサポートされています。</p> <p>-c disk1 disk2 <i>disk1</i> と <i>disk2</i> を使用してミラーを作成します。<i>disk2</i> の内容を <i>disk1</i> の内容で置換します。<i>disk1</i> と <i>disk2</i> を標準的な形式 (たとえば、c0t0d0) で指定します。</p> <p>RAID ボリュームを作成するとき、RAID ボリュームはディスクペアの最初のターゲット (<i>disk1</i>) の識別情報を前提とします。2 番目のターゲット (<i>disk2</i>) はシステムから消失します。このため、RAID ボリュームは1つのディスクとして表示されます。</p> <p>RAID の作成時に、指定されたコントローラに RAID 構成がすでに存在している場合は失敗します。さらに、2 番目のディスクは、すべてのデータが消去されて、1 番目のディスクのデータで置換されるので、マウントしないでください。</p> <p>-d disk1 <i>disk1</i> として指定された RAID ボリュームを削除します。</p> <p><i>disk1</i> を標準的な形式 (たとえば、c0t0d0) で指定します。</p> <p>-f 強制的に更新します。プロンプトは出力しません。</p> <p>-F filename controller 指定のコントローラ (<i>controller</i>) で動作するファームウェアを更新します。</p> <p>-l [controller ...] システムの RAID 構成を一覧表示します。<i>controller</i> を指定した場合は、<i>controller</i> の RAID 構成が一覧表示されます。</p>

-l による出力は、次の情報を一覧表示します。

RAID Volume	論理 RAID ボリューム名を表示します。
RAID Status	RESYNCING (ディスクが同期している)、DEGRADED (RAID は機能を制限して稼動している)、OK (正常に稼動している)、または FAILED (機能していない) のいずれかの RAID ステータスを表示します。
RAID Disk	RAID ディスク名を表示します。
Disk Status	OK または FAILED のいずれかのディスクステータスを表示します。

使用例 例 1 RAID 構成の作成

次のコマンドにより RAID 構成を作成します。

```
# raidctl -c c0t0d0 c0t1d0
RAID Volume 'c0t0d0' created
```

例 2 RAID 構成の表示

次のコマンドにより RAID 構成を表示します。

```
# raidctl
RAID      RAID      RAID      Disk
Volume    Status    Disk      Status
-----
c0t0d0    RESYNCING c0t0d0    OK
                c0t1d0    OK
```

例 3 RAID 構成の削除

次のコマンドにより、RAID 構成を削除します。

```
# raidctl -d c0t0d0
RAID Volume 'c0t0d0' deleted.
```

例 4 コントローラのフラッシュイメージの更新

次のコマンドにより、コントローラのフラッシュイメージを更新します。

```
# raidctl -F lsi1030.fw 0
Update flash image on controller 0? (y/N): y
Flash updated successfully
```

終了ステータス 次の終了値が返されます。

raidctl(1M)

- 0 正常に終了しました。
- 1 不正なコマンド行入力です。
- 2 要求操作は失敗しました。

属性 以下の属性については、attributes(5)を参照してください。

属性の種類	属性の値
Availability	SUNWcsu

関連項目 attributes(5)
Solaris のシステム管理 (基本編)

名前	reboot – オペレーティングシステムの再起動
形式	<code>/usr/sbin/reboot [-dlmq] boot_arguments</code>
機能説明	<p>reboot ユーティリティはカーネルを再起動します。カーネルは PROM モニターによってメモリーに読み込まれ、読み込まれたカーネルに制御が渡されます。</p> <p>スーパーユーザーはいつでも reboot を実行できますが、通常は shutdown(1M) を使用して、これからサービスが停止されることを、すべてのユーザーに事前に警告する必要があります。詳細については、shutdown(1M) のマニュアルページを参照してください。</p> <p>reboot ユーティリティは sync(1M) 操作をディスクに実行して、マルチユーザー状態で再起動を実行します。詳細については、init(1M) のマニュアルページを参照してください。</p> <p>reboot ユーティリティは通常、再起動の記録をシステムログデーモン syslogd(1M) に送信し、シャットダウンの記録をログインアカウントファイル <code>/var/adm/wtmpx</code> に保存します。これらの処理を抑制するには、<code>-n</code> または <code>-q</code> オプションを指定します。</p> <p>通常、システムはパワーアップ時やクラッシュ後に再起動します。</p>
オプション	<p>次のオプションを指定できます。</p> <p><code>-d</code> 再起動する前に、システムクラッシュダンプを行います。システムクラッシュダンプを構成する方法については、dumpadm(1M) のマニュアルページを参照してください。</p> <p><code>-l</code> システムログデーモン syslogd(1M) へのメッセージ (誰が reboot を実行したかを示す) の送信を抑制します。</p> <p><code>-n</code> sync(2) の呼び出しは行わず、syslogd(1M) または <code>/var/adm/wtmpx</code> に再起動の記録を保存しないようにしますが、カーネルは再起動前にファイルシステムとの同期を取ろうとします。ただし、<code>-d</code> オプションも指定した場合は例外です。<code>-d</code> と <code>-n</code> を一緒に指定した場合、カーネルはファイルシステムとの同期を取りません。</p> <p><code>-q</code> 実行中のプロセスを停止せずに、ただちに再起動します。</p>
オペランド	<p>次のオペランドを指定できます。</p> <p><i>boot_arguments</i> オプションの <i>boot_arguments</i> 文字列を指定すると、uadmin(2) 関数に引数を指定できます。これらの引数は再起動時にブートプログラムとカーネルに渡されます。引数の形式と一覧については、boot(1M) のマニュアルページと kernel(1M) のマニュアルページを参照してください。複数の引数を指定する場合、シェルでそれらの引数をエスケープするために、それらの引数を引用符で囲む必要があります (「使用例」の項を参照)。<i>boot_arguments</i> がハイフン (-) で始まる場合、区切り文字列 (2 つのハイフン) を <i>boot_arguments</i> のハイフンの前に指定して、reboot 引数リストの</p>

reboot(1M)

終わりを示す必要があります (「使用例」の項を参照)。

使用例 例 1 -r と -v 引数を boot に渡す

次の例では、区切り文字列「—」(2つのハイフン)を使用して、reboot のオプションを reboot(1M) の引数から分離する必要があります。

```
example# reboot -dl — -rv
```

例 2 特定のディスクとカーネルを使用して再起動する

次の例は、特定のディスクとカーネルを使用して再起動します。複数のブート引数がクォートされていることに注意してください。

```
example# reboot "disk1 kernel.test/unix"
```

ファイル /var/adm/wtmpx ログインアカウントングファイル

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 mdb(1), boot(1M), dumpadm(1M), fsck(1M), halt(1M), init(1M), kernel(1M), shutdown(1M), sync(1M), syslogd(1M), sync(2), uadmin(2), reboot(3C), attributes(5)

注意事項 reboot ユーティリティは /etc/rcnum.d 内のスクリプトや inittab(4) 内の停止アクションを実行しません。システムサービスを完全に停止するためには、shutdown(1M) または init(1M) を使用して Solaris システムを再起動します。

名前	accept, reject – 印刷要求の受付または拒否						
形式	accept <i>destination</i> ... reject [-r <i>reason</i>] <i>destination</i> ...						
機能説明	accept コマンドは、指定した宛先への印刷要求を待ち行列に加えることを許可します。 reject コマンドは、指定した宛先への印刷要求を待ち行列に加えることができないようにします。 宛先が印刷要求を受け付けたか、または拒否したかを確認するには、lpstat -a を実行してください。 accept と request は印刷サーバー上で実行する必要があります。クライアントシステム上では意味をもちません。						
オプション	reject コマンドには、次のオプションを指定できます。 -r <i>reason</i> <i>destination</i> への印刷要求を抑止する理由を文字列として記述します。 <i>reason</i> に空白が含まれる場合は引用符で囲ってください。 <i>reason</i> は、プリンタの状況を調べる lpstat -a コマンドの出力中に表示されます。 <i>reason</i> のデフォルト値は、既存の宛先に関しては unknown reason、システムに追加されたばかりでまだ印刷要求を受け付けたことのない宛先に関しては new printer となります。						
オペランド	次のオペランドを指定できます。 <i>destination</i> 印刷要求を受け付ける、または拒否する宛先名。宛先にはプリンタ名やプリンタクラスを指定します (lpadmin(1M) 参照)。 <i>destination</i> は名前を使用して指定します。名前の命名規約については printers.conf(4) のマニュアルページを参照してください。						
終了ステータス	以下の終了ステータスが返されます。 0 正常終了 0 以外 エラーが発生した						
ファイル	/var/spool/lp/* LP 印刷待ち行列						
属性	次の属性については attributes(5) のマニュアルページを参照してください。						
	<table border="1"> <thead> <tr> <th>属性タイプ</th> <th>属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWpcu</td> </tr> <tr> <td>CSI</td> <td>対応済み (「注意事項」参照)</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWpcu	CSI	対応済み (「注意事項」参照)
属性タイプ	属性値						
使用条件	SUNWpcu						
CSI	対応済み (「注意事項」参照)						

reject(1M)

関連項目	enable(1), lp(1), lpstat(1), lpadmin(1M), lpsched(1M), printers.conf(4), attributes(5)
注意事項	<p>accept と reject は印刷サーバーのスプール用システムの待ち行列に対してのみ有効です。したがって、クライアントシステムから出された要求については、印刷サーバーのスプール用システムが取り消したり受け取ったりするまでは、クライアントシステムの印刷待ち行列に入れられたままの状態になります。</p> <p>accept は <i>destinations</i> 名を除いて CSI 対応が可能です。</p>

名前	rsh, restricted_shell – 制限付きシェルのコマンドインタプリタ
形式	<code>/usr/lib/rsh [-acefhiknprstuvx] [argument...]</code>
機能説明	<p>rsh は、標準のコマンドインタプリタである sh と比べて機能の一部が制限されており、ログインが許される実行環境も、sh の環境と比べて制限されています。機能の詳細や使用方法に関しては、sh(1) の説明を参照してください。</p> <p>シェルは、呼び出されると環境変数 SHELL を調べます。この環境変数が存在していて、その値のファイル名部分が rsh であれば、シェルは制限付きシェルとなります。</p> <p>rsh の機能は、以下の動作ができない点を除き sh と同一です。</p> <ul style="list-style-type: none"> ■ ディレクトリの変更 (cd(1) を参照) ■ \$PATH の値の設定 ■ / を含むパスまたはコマンド名の指定 ■ 出力先のリダイレクト (> および >>) <p>これらの制限は、.profile の解釈後に有効となります。</p> <p>制限付きシェルは、次のいずれかの方法で呼び出せます。</p> <ol style="list-style-type: none"> (1) /etc/passwd ファイルの最後のエントリのファイル名部分を rsh と記述する (passwd(4) を参照) (2) 環境変数 SHELL が存在していて、その値のファイル名の部分が rsh である。環境変数 SHELL は、.login ファイル内に設定されている必要がある (3) シェルの呼び出し時に、引数 0 のファイル名の部分が rsh である (4) シェルを -r オプション付きで呼び出す <p>実行するコマンドがシェル手続きである場合は、rsh は sh を呼び出して、コマンドを実行します。したがって、利用できるコマンドの種類には制限はありますが、一般ユーザーは標準シェルの全機能を利用できるシェル手続きを使用することができません。このスキーマは、一般ユーザーが同じディレクトリにおいて書き込み権と実行権を持っていないことを想定しています。</p> <p>.profile の作成者 (profile(4) を参照) が、確実な設定処理を実行してユーザーを適切なディレクトリ (おそらく、ログインディレクトリではない) に置くことにより、ユーザーの動作を完全に制御できるという点が、これらの規約の実際の効果となります。</p> <p>システム管理者は、制限付きシェルで安全に起動できるコマンドのディレクトリ (つまり /usr/rbin) を設定することがよくあります。システムによっては、制限付きエディタ red を提供するものもあります。</p>
終了ステータス	<p>構文エラーなどのエラーを検出した場合、シェルは 0 以外の終了ステータスを返します。シェルを対話型以外で使用している場合、シェルフファイルの実行は中止されます。対話型で使用している場合は、シェルは最後に実行されたコマンドの終了ステータスを返します。</p>

restricted_shell(1M)

属性 次の属性については `attributes(5)` のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 `cd(1)`, `intro(1)`, `login(1)`, `rsh(1)`, `sh(1)`, `exec(2)`, `passwd(4)`, `profile(4)`, `attributes(5)`

注意事項 制限付きシェル `/usr/lib/rsh` を、リモートシェル `/usr/bin/rsh` と混同しないように注意してください。リモートシェルに関しては `rsh(1)` を参照してください。

名前	install_scripts, add_install_client, add_to_install_server, modify_install_server, rm_install_client, setup_install_server, check – Solaris ソフトウェアをインストールするためのスクリプト
形式	<pre> cdrom-mnt-pt/Solaris_XX/Tools/add_install_client [-i IP_address] [-e Ethernet_address] [-s server_name : path] [-c server_name : path] [-n [server] : name_service [(netmask)]] [-p server_name : path] [-f boot_file_name] host_name platform_group cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server_name:path] [-c server_name:path] [-p server_name:path] [-t install_boot_image_path] [-f boot_file_name] platform_name platform_group cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server_name:path] [-c server_name:path] [-p server_name:path] [-t install_boot_image_path] [-f boot_file_name] -e Ethernet_address [-b property=value] platform_group cdrom-mnt-pt/Solaris_XX/Tools/add_to_install_server [-s] [-p product_image_path] install_server_path cdrom-mnt-pt/Solaris_XX/Tools/jumpstart_sample/check [-p install_dir_path] [-r rulesfile] cdrom-mnt-pt/modify_install_server [-p] install_dir_path installer_miniroot_path cdrom-mnt-pt/Solaris_XX/Tools/rm_install_client host_name cdrom-mnt-pt/Solaris_XX/Tools/setup_install_server [-b] [-t install_boot_image_path] [-w wanboot_image_path] install_dir_path </pre>
機能説明	<p>これらのコマンドは、Solaris Software and Solaris Installer CD のスライス 0 に置かれています。Solaris CD をすでにローカルディスクにコピーしている場合、<code>cdrom_mnt_pt</code> はコピーされている Solaris CD へのパスになります。これらのコマンドは、さまざまなインストール作業に使用できます。</p> <p>Solaris_XX の XX は、使用している Solaris リリースのバージョン番号です。</p> <p><code>add_install_client</code> コマンドには 3 種類の形式があります。「形式」の項を参照してください。</p> <p>ネットワークを使用してインストールするクライアントを追加する場合は、次の形式の <code>add_install_client</code> を使用します (これらのコマンドは <code>bootparams(4)</code> ファイルを更新します)。<code>add_install_client</code> コマンドは、インストールサーバーの Solaris インストールイメージ (マウントした CD またはディスクにコピーした Solaris CD) あるいはブートサーバーの起動ディレクトリ (ブートサーバーが必要な場合) から実行する必要があります。Solaris インストールイメージまたはブート専用ディレクトリの Solaris リリースは、クライアントにインストールする予定の Solaris リリースと同じでなければなりません。</p> <pre> cdrom-mnt-pt/Solaris_XX/Tools/add_install_client [-i IP_address] [-e Ethernet_address] [-s server_name : path] [-c server_name : path] </pre>

rm_install_client(1M)

```
[ -n [server] ] : name_service [ ( netmask) ] [ -p server_name : path ]
host_name platform_group
```

プラットフォームグループ内のあるプラットフォームのインスタンスをインストールサーバーに追加する場合は、次の形式の `add_install_client` コマンドを使用します。このグループは DHCP を使用して起動および構成されます。スクリプトは、必要な構成作業をサーバー上で実行し、ユーザーがそのグループのために DHCP サーバーに追加しなければならないデータを出力します。

```
cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server:path]
[-c server:path] [-p server:path] [-t install boot image path]
[-f boot file name] platform_name platform_group
```

インストールサーバーに単一のクライアントを追加する場合は、次の形式の `add_install_client` コマンドを使用します。このクライアントは DHCP を使用して起動および構成されます。スクリプトは、必要な構成作業をサーバー上で実行し、ユーザーがそのクライアントのために DHCP サーバーに追加しなければならないデータを出力します。すでに使用しているものにも、上記のように `-f` フラグを追加する必要があります。`-f` を使用すると、ユーザーは所定のクライアントに使用する起動ファイルの名前を指定できます。

```
cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d
[-s server_name:path] [-c server_name:path] [-p server_name:path]
[-t install_boot_image_path] [-f boot_file_name] -e Ethernet_address
platform_group
```

注 - x86 アーキテクチャの Pre-boot eXecution Environment (PXE) クライアントを登録する場合は、必ず `-d` オプションを使用してください。x86 PXE クライアントは構成に DHCP を使用します。

他の Solaris CD とネットワークインストールサーバー上の既存のイメージをマージする場合は、`add_to_install_server` を使用します。マージ可能な各 CD (現時点では OS CD 2 および Language CD) には専用の `add_to_install_server` スクリプトが含まれています。配布された CD に含まれているもの以外の `add_to_install_server` スクリプトを使用しないでください。

`rules` ファイル (カスタム JumpStart インストールを使用する場合に限り必要) のルールを検証する場合は、`check` を使用します。

既存のネットワークインストールサーバーのミニルートを Solaris Installation CD のミニルートに置き換える場合は、`modify_install_server` を使用します。ネットワークインストールサーバーのインストール時のユーザーインターフェースが、Solaris Installation CD の Web Start ユーザーインターフェースに変更されません。

`modify_install_server` コマンドを使用する場合は、既存のインストールイメージ (`setup_install_server` を使用して作成) があらかじめ存在していなければなりません。

ネットワークインストール用のクライアントを削除する場合は、`rm_install_client` を使用します (このコマンドは `bootparams(4)` ファイルを更新します)。

Solaris CD を (インストールサーバーを設定するために) ディスクにコピーする、(WANboot インストールサーバーを設定するために) WANboot ミニルートイメージを構築する、または Solaris CD のブートソフトウェアだけを (ブートサーバーを設定するために) ディスクにコピーする場合は、`setup_install_server` を使用します。ネットワーク経由でクライアントをインストールするには、インストールサーバーが必要です。インストールサーバーとインストールするクライアントが異なるサブネットに属している場合、ネットワークインストールを行うには、ブートサーバーも必要です (ブートサーバーはクライアントのサブネットに配置する必要があります)。

オプション `add_install_client` には、次のオプションを指定できます。

-b *property=value*

ブートサーバーの TFTP ディレクトリ (デフォルトは `/tftpboot`) にあるクライアント特有の `bootenv.rc` ファイルにプロパティ値を設定します。クライアントに特有のブートプロパティを設定する場合には、このオプションを使用します。`bootpath` プロパティを使用すると、デバイス構成補助を使用して手動で選択しなくても、ブートデバイスを自動的に選択できます。`boot-args` プロパティを使用すると、JumpStart インストールを自動的に選択できます。 `eeeprom(1M)` を参照してください。

このオプションは x86 クライアントにのみ使用できます。このオプションを使用するときには、必ず `-d` オプションと `-e` を同時に指定してください。

-c *server_name:path*

このオプションが必要なのは、カスタム JumpStart インストール用の JumpStart ディレクトリを指定する場合だけです。`server_name` は、JumpStart ディレクトリが置かれているサーバーのホスト名です。`path` は JumpStart ディレクトリの絶対パスです。

-d

DHCP クライアントを指定します。

-e *Ethernet_address*

インストールするシステムの Ethernet アドレスを指定します。

-f

インストールするクライアントの `boot_file_name` を指定します。

-i *IP_address*

インストールするクライアントの IP アドレスを指定します。

-n [*server*]: *name_service*[(*netmask*)]

このオプションでは、システム構成時に使用するネームサービスを指定します。`bootparams(4)` ファイル内の `ns` キーワードを設定します。

name_service

有効なエントリは `nis`、`nisplus`、および `none` です。

rm_install_client(1M)

netmask

数字を4つずつピリオドで区切って並べたもので、IPアドレスのどの部分がネットワーク部分で、どれがホスト部分かを指定します。

server

サーバーの名前または指定したネームサービスのIPアドレス。指定したサーバーが異なるサブネット上にある場合は、クライアントからサーバーにアクセスするために、*netmask*が必要なことがあります。

-p *server_name*: *path*

このオプションは、システム情報またはネットワーク情報を前もって設定するために必要な、ユーザーが定義した *sysidcfg* ファイルの位置です。*server_name* は有効なホスト名またはIPアドレスです。*path* はJumpStart ディレクトリの絶対パスです。

-s *server_name*:*path*

このオプションが必要なのは、ブートサーバーから *add_install_client* を使用する場合だけです。サーバー名およびこのインストールで使用する Solaris インストールイメージの絶対パスを指定します。*path* はマウントされている Solaris CD のパスまたは Solaris CD のコピーが置かれているディレクトリのパスです。

add_to_install_server コマンドには、次のオプションを指定できます。

-p

コピーする CD (補助的な製品が収められている) の位置を指定します。

-s

ユーザーが、インストールする必要がある製品だけをリストから選択できるようにします。

check コマンドには、次のオプションを指定できます。

-p *install_dir_path*

使用中のシステムの *check* スクリプトではなく、指定された Solaris インストールイメージの *check* スクリプトを使用することによって、ルールファイルを検査します。*install_dir_path* は、ローカルディスク上またはマウントした Solaris CD 上の Solaris インストールイメージのパスです。

旧バージョンの Solaris が稼働しているシステムでは、このオプションを使用して、最新バージョンの *check* を実行できます。

-r *rulesfile*

rules 以外の名前のルールファイルを指定します。このオプションを使用すると、ルールの妥当性を検証してから、ルールファイルに組み込むことができます。*check* はルールが有効であるかどうかを報告するだけで、カスタム JumpStart インストールに必要な *rules.ok* ファイルは作成しません。

modify_install_server コマンドには、次のオプションを指定できます。

-p

このオプションは、*install_dir_path/Solaris_XX/Tools/Boot.orig* にすでに存在しているイメージのミニルートを保持します。

setup_install_server コマンドには、次のオプションを指定できます。

- b
サーバーの設定を、ブートサーバー専用にします。
- t
代替ミニルート指定できるようにします。
- w
WANboot ミニルートイメージを構築します。

オペランド add_install_client コマンドには、次のオペランドを指定できます。

host_name

インストールするクライアントの名前です。

platform_group

特定のソフトウェアを配布するために、ベンダーが定義したハードウェアプラットフォームグループ。以下に有効なプラットフォームグループの例を示します。

システム	プラットフォームグループ
x86	i86pc
Sun Fire 4800	sun4c

システムのプラットフォームグループを調べるには、(-m オプションを指定して) uname(1) コマンドを使用します。

platform_name

システムのプラットフォーム名を調べるには、-i オプションを指定して uname(1) コマンドを使用します。

次に、uname command コマンドを使用して Ultra 1 のシステムプラットフォーム名を調べる例を示します。

```
uname -i
システムは、次のように応答します。
```

```
SUNW,Ultra-1
システムのプラットフォーム名は SUNW,Ultra1 です。
```

次のコマンドは、Ultra 1 用の add_install_client を呼び出します。

```
add_install_client -d SUNW,Ultra-1 sun4u
```

IA32 プラットフォームでは、プラットフォーム名はつねに SUNW.i86pc です。

次のコマンドは、IA32 プラットフォーム用の add_install_client を呼び出します。

```
add_install_client -d SUNW.i86pc i86pc
```

rm_install_client コマンドには、次のオペランドを指定できます。

rm_install_client(1M)

host_name

削除するクライアントの名前です。

setup_install_server コマンドには、次のオペランドを指定できます。

install_dir_path

Solaris ソフトウェアのコピー先となるディレクトリの絶対パスです。このディレクトリは空でなければなりません。

wanboot_image_path

WANboot ミニルートイメージを含むファイルが作成されるディレクトリの絶対パスです。

使用例 例 1 add_install_client の使用

次の add_install_client コマンドは、マウント済みの Solaris CD からネットワークインストール用のクライアントを Solaris 8 のインストールサーバーに追加します。

```
example# cd /cdrom/cdrom0/s0/Solaris_9/Tools
example# ./add_install_client system_2/sun4u
```

例 2 add_install_client の使用

次の add_install_client コマンドは、ネットワークインストール用のクライアントをインストールサーバーに追加します。カスタム JumpStart インストールを実行するためのルールファイルおよびプロファイルファイルが置かれている JumpStart のサーバーとディレクトリのパスを -c オプションで指定します。Solaris CD は Solaris 8 上の /export/install ディレクトリにコピーされています。

```
example# cd /export/install/Solaris_8/Tools
example# ./add_install_client -c install_server:/jumpstart system_1 i86pc
example# ./add_install_client -c install_server:/jumpstart system_2 i86pc
```

例 3 add_install_client の使用

次の add_install_client コマンドは、起動ファイル sun4u.solaris8 を使用する特定の sun4u プラットフォームマシン (8:0:20:99:88:77) に対するサポートを追加します。

```
example# add_install_client -d -f sun4u.solaris8 -e 8:0:20:99:88:77 sun4u
```

例 4 add_install_client の使用

次の add_install_client コマンドは、PXE 規格を使用してネットワークから起動する x86 クライアントを追加します。

```
example# add_install_client -d -s svrname:/mnt/export/root SUNW.i86pc i86p
```

例 5 add_to_install_server の使用

次の add_to_install_server コマンドは、CD 上のすべての製品ディレクトリにあるパッケージを Solaris 8 上の既存のインストールサーバーにコピーします。

例 5 add_to_install_server の使用 (続き)

```
example# cd /cdrom/cdrom0/s0
example# ./add_to_install_server /export/Solaris_8
```

例 6 check の使用

次の check コマンドは、カスタム JumpStart インストールに使用するルールファイルの構文を検査します。

```
example# cd jumpstart_dir_path
example# ./check -p /cdrom/cdrom0/s0
```

例 7 modify_install_server の使用

次の modify_install_server コマンドは、上記の setup_install_server を使用して作成されたミニルート Boot.orig に移し、Solaris Installer CD 上のミニルートに置き換えます。

```
example# cd /cdrom/cdrom0/s0
example# ./modify_install_server -p /export/install /cdrom/cdrom0/s1
```

例 8 modify_install_server の使用

次の modify_install_server コマンドは、上記の setup_install_server を使用して作成されたミニルートを Solaris Installer CD 上のミニルートに置き換えます。

```
example# cd /cdrom/cdrom0/s0
example# ./modify_install_server /export/install /cdrom/cdrom0/s1
```

例 9 rm_install_client の使用

次の rm_install_client コマンドは、Solaris 8 上のネットワークインストール用クライアントを削除します。

```
example# cd /export/install/Solaris_8/Tools
example# ./rm_install_client holmes
example# ./rm_install_client watson
```

例 10 setup_install_server の使用

次の setup_install_server コマンドは、マウント済みの Solaris CD を Solaris 8 のローカルディスク上にある /export/install というディレクトリにコピーします。

```
example# cd /cdrom/cdrom0/s0/Solaris_8/Tools
example# ./setup_install_server /export/install
```

例 11 setup_install_server の使用

次の setup_install_server コマンドは、マウント済みの Solaris CD のブートソフトウェアを、Solaris 8 上のサブネット用ブートサーバーになるシステムの /boot_dir というディレクトリにコピーします。

rm_install_client(1M)

例 11 setup_install_server の使用 (続き)

```
example# cd /cdrom/cdrom0/s0/Solaris_8/Tools
example# ./setup_install_server -b /boot_dir
```

例 12 setup_install_server の使用

setup_install_server はデフォルトで、マウント済みの Solaris 配布ディスク上の Solaris ../Tools/Boot にあるインストールブートディレクトリを検索します。

以前に ./setup_install_server -b /boot_dir コマンドを使用して、ネットワークブートサーバー上に作成したディレクトリと別のディレクトリが必要な場合は、-t オプションを使用できます。

```
example# cd /cdrom/cdrom0/s0/Solairs_9/Tools
example# ./setup_install_server -t /boot_dir /export/install
```

例 13 setup_install_server と WANboot オプションの使用

次の setup_install_server コマンドは WANboot ミニルートファイルシステムのイメージを作成し、それを /wanboot_dir/miniroot に保存します。

```
example# cd /cdrom/cdrom0/s0/Solairs_9/Tools
example# ./setup_install_server -w /wanboot_dir /export/install
```

例 14 x86: ネットワークインストール時に使用するシリアルコンソールの指定 (CD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するシリアルコンソールを指定する方法を示します。この例では、インストールクライアントを次の方法で設定します。

- -d オプションは、クライアントが DHCP を使用してインストールパラメータを設定することを示します。
- -e オプションは、Ethernet アドレスが 00:07:e9:04:4a:bf であるクライアントでのみインストールが行われることを示します。
- 1 つめと 2 つめの -b オプションは、インストールプログラムが入力デバイスおよび出力デバイスとしてシリアルポート ttya を使用することを示します。

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
i86pc
```

-b オプションで使用できるブートプロパティの変数とその値については、eprom(1M) を参照してください。

例 15 ネットワークインストール時に使用するブートデバイスの指定 (CD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するブートデバイスを指定する方法を示します。インストールクライアントを設定するときにブートデバイスを指定する場合は、インストール時にデバイス構成補助によって出されるこの情報は表示されません。

例 15 ネットワークインストール時に使用するブートデバイスの指定 (CD) (続き)

この例では、次の方法でインストールクライアントを設定します。

- -d オプションは、クライアントが DHCP を使用してインストールパラメータを設定することを示します。
- -e オプションは、Ethernet アドレスが 00:07:e9:04:4a:bf であるクライアントでのみインストールが行われることを示します。
- 1 つめと 2 つめの -b オプションは、インストールプログラムが入力デバイスおよび出力デバイスとしてシリアルポート ttya を使用することを示します。
- 3 つめの -b オプションは、インストールプログラムがインストール時に特定のブートデバイスを使用することを示します。
- ブートデバイスのパスは、使用しているハードウェアによって異なります。
- i86pc というプラットフォーム名は、クライアントが x86 ベースのシステムであることを示します。

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
-b "bootpath=/pci@0,0/pci108e,16a8@8" i86pc
```

-b オプションで使用できるブートプロパティの変数とその値について詳細は、eeprom(1M) を参照してください。

終了ステータス 次の終了ステータスが返されます。

```
0      正常終了
1      エラーが発生した
```

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	Solaris CD

関連項目 uname(1), bootparams(4), attributes(5)

『Solaris 9 インストールガイド』

rsh(1M)

名前	rsh, restricted_shell – 制限付きシェルのコマンドインタプリタ
形式	<code>/usr/lib/rsh [-acefhiknprstuvx] [argument...]</code>
機能説明	<p>rsh は、標準のコマンドインタプリタである sh と比べて機能の一部が制限されており、ログインが許される実行環境も、sh の環境と比べて制限されています。機能の詳細や使用方法に関しては、sh(1) の説明を参照してください。</p> <p>シェルは、呼び出されると環境変数 SHELL を調べます。この環境変数が存在していて、その値のファイル名部分が rsh であれば、シェルは制限付きシェルとなります。</p> <p>rsh の機能は、以下の動作ができない点を除き sh と同一です。</p> <ul style="list-style-type: none">■ ディレクトリの変更 (cd(1) を参照)■ \$PATH の値の設定■ / を含むパスまたはコマンド名の指定■ 出力先のリダイレクト (> および >>) <p>これらの制限は、.profile の解釈後に有効となります。</p> <p>制限付きシェルは、次のいずれかの方法で呼び出せます。</p> <ol style="list-style-type: none">(1) /etc/passwd ファイルの最後のエントリのファイル名部分を rsh と記述する (passwd(4) を参照)(2) 環境変数 SHELL が存在していて、その値のファイル名の部分が rsh である。環境変数 SHELL は、.login ファイル内に設定されている必要がある(3) シェルの呼び出し時に、引数 0 のファイル名の部分が rsh である(4) シェルを -r オプション付きで呼び出す <p>実行するコマンドがシェル手続きである場合は、rsh は sh を呼び出して、コマンドを実行します。したがって、利用できるコマンドの種類には制限はありますが、一般ユーザーは標準シェルの全機能を利用できるシェル手続きを使用することができます。このスキーマは、一般ユーザーが同じディレクトリにおいて書き込み権と実行権を持っていないことを想定しています。</p> <p>.profile の作成者 (profile(4) を参照) が、確実な設定処理を実行してユーザーを適切なディレクトリ (おそらく、ログインディレクトリではない) に置くことにより、ユーザーの動作を完全に制御できるという点が、これらの規約の実際の効果となります。</p> <p>システム管理者は、制限付きシェルで安全に起動できるコマンドのディレクトリ (つまり /usr/sbin) を設定することがよくあります。システムによっては、制限付きエディタ red を提供するものもあります。</p>
終了ステータス	構文エラーなどのエラーを検出した場合、シェルは 0 以外の終了ステータスを返します。シェルを対話型以外で使用している場合、シェルファイルの実行は中止されません。対話型で使用している場合は、シェルは最後に実行されたコマンドの終了ステータスを返します。

属性 次の属性については `attributes(5)` のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 `cd(1)`, `intro(1)`, `login(1)`, `rsh(1)`, `sh(1)`, `exec(2)`, `passwd(4)`, `profile(4)`, `attributes(5)`

注意事項 制限付きシェル `/usr/lib/rsh` を、リモートシェル `/usr/bin/rsh` と混同しないように注意してください。リモートシェルに関しては `rsh(1)` を参照してください。

scadm(1M)

名前	scadm – システムコントローラ (SC) の管理
形式	<code>/usr/platform/platform-name/sbin/scadm subcommand [option] [argument...]</code>
機能説明	<p>scadm ユーティリティは、システムコントローラ (SC) を管理します。このユーティリティを使用すると、ホストサーバーが SC と対話できるようになります。</p> <p>scadm ユーティリティは、スーパーユーザーとして実行しなければなりません。</p> <p>scadm のインタフェース、出力、およびディレクトリ階層内の位置は確定したのではなく、変更されることがあります。</p> <p><i>platform-name</i> はプラットフォームの実装名です。uname -i コマンドを使用して、プラットフォーム実装を識別します。uname(1) を参照してください。</p> <p>scadm ユーティリティには 15 個のサブコマンドがあります。一部のサブコマンドには、そのサブコマンドに関連付けられている特定のオプションと引数があります。「サブコマンド」、「オプション」、「オペランド」、および「使用例」を参照してください。</p>
サブコマンド	<p>コマンド行の scadm コマンドの後に SPACE を入力し、その後にサブコマンドを指定します。</p> <p>次のサブコマンドを指定できます。</p> <p>date SC の日付と時刻を表示します。</p> <p>date サブコマンドの書式を以下に示します。</p> <pre>scadm date</pre> <p>download SC のファームウェアをプログラムします。</p> <p>ファームウェアについては、ブートモニターとメインイメージの 2 つの部分があります。</p> <p>デフォルトでは、scadm コマンドのダウンロードを実行すれば、メインファームウェアイメージがプログラムされます。boot 引数を指定すると、ブートモニターのプログラミングが選択されます。</p> <p>download サブコマンドの書式を以下に示します。</p> <pre>scadm download [boot] file</pre> <p>help コマンドの一覧を表示します。</p> <p>help サブコマンドの書式を以下に示します。</p> <pre>scadm help</pre> <p>loghistory SC イベントログの最新のエントリを表示します。</p> <p>loghistory サブコマンドの書式を以下に示します。</p>

	<code>scadm loghistory</code>
<code>resetrsc</code>	<p>SC をリセットします。実行できるリセットには、ハードリセットとソフトリセットの 2 種類があります。デフォルトで実行されるのがハードリセットです。ソフトリセットを選択する場合は <code>-s</code> オプションを指定します。</p> <p><code>resetrsc</code> サブコマンドの書式を以下に示します。</p> <pre>scadm resetrsc [-s]</pre>
<code>send_event</code>	<p>テキストベースのイベントを手動で送信します。SC は、イベントを SC イベントログに転送できます。<code>-c</code> オプションを構成して、重要な警告を電子メール、ログイン中の SC ユーザー、および <code>syslog</code> に送信できます。重要なイベントは <code>syslog(3C)</code> に記録されます。関連テキストメッセージの文字数は 80 文字以内です。</p> <p><code>send_event</code> サブコマンドの書式を以下に示します。</p> <pre>scadm send_event [-c] "message"</pre>
<code>set</code>	<p>SC 構成変数の値を設定します。</p> <p>SC 構成変数には、SC IP アドレス <code>netsc_ipaddr</code> や、SC カスタマ情報 <code>sc_customerinfo</code> などがあります。SC 構成変数の全リストについては、<code>scadm help</code> コマンドの出力結果を参照してください。</p> <p><code>set</code> サブコマンドの書式を以下に示します。</p> <pre>scadm set variable value</pre>
<code>show</code>	<p>SC 構成変数の現在の設定を表示します。変数を指定しないで <code>scadm</code> を実行すると、すべての変数設定が表示されます。</p> <p><code>show</code> サブコマンドの書式を以下に示します。</p> <pre>scadm show [variable]</pre>
<code>shownetwork</code>	<p>SC の現在のネットワーク構成パラメタを表示します。</p> <p><code>shownetwork</code> サブコマンドの書式を以下に示します。</p> <pre>scadm shownetwork</pre>
<code>useradd</code>	<p>SC にユーザーアカウントを追加します。SC では 16 名までの個別ユーザーがサポートされます。</p> <p><code>useradd</code> サブコマンドの書式を以下に示します。</p> <pre>scadm useradd username</pre>
<code>userdel</code>	<p>SC からユーザーアカウントを削除します。</p> <p><code>userdel</code> サブコマンドの書式を以下に示します。</p>

scadm(1M)

	<code>scadm userdel username</code>
userpassword	<p>指定したユーザーアカウントのパスワードを設定します。このパスワードは、現在設定されている既存のパスワードがあった場合にはそれより優先されます。新しいパスワードを設定する前に、古いパスワードは検証されません。</p> <p>userpassword サブコマンドの書式を以下に示します。</p> <p><code>scadm userpassword username</code></p>
userperm	<p>ユーザーのアクセス権のレベルを設定します。</p> <p>userperm サブコマンドの書式を以下に示します。</p> <p><code>scadm userperm username [aucr]</code></p>
usershow	<p>指定したユーザーアカウントの詳細を表示します。ユーザー名を指定しないと、すべてのユーザーアカウントの詳細が表示されません。</p> <p>usershow サブコマンドの書式を以下に示します。</p> <p><code>scadm usershow username</code></p>
version	<p>SC のバージョン番号とそのコンポーネントを表示します。</p> <p>version サブコマンドの書式を以下に示します。</p> <p><code>scadm version [-v]</code></p>
オプション	<p>resetrsc、send_event、および version サブコマンドには、関連するオプションがあります。コマンド行のサブコマンドの後に SPACE を入力し、その後にオプションを指定します。</p> <p>resetrsc サブコマンドでは次のオプションを指定できます。</p> <p>-s ハードリセットの代わりにソフトリセットを実行します。ハードリセットは SC ハードウェアを物理的にリセットします。ソフトリセットの場合は、SC ソフトウェアがブートファームウェアにジャンプして、リセットをシミュレーションします。</p> <p>send_event サブコマンドでは次のオプションを指定できます。</p> <p>-c 重要なイベントを送信します。-c を指定しないで -send_event を実行すると、警告が送信されます。</p> <p>version サブコマンドでは次のオプションを指定できます。</p> <p>-v バージョン番号と関連情報の詳細な内容が表示されます。</p>
オペラント	<p>download, send_event, set, show, useradd, userdel, userperm, usershow, userpassword, userperm の各サブコマンドには関連する引数 (オペラント) があります。</p>

サブコマンドにオプションを指定する場合は、コマンド行のオプションの後に SPACE を入力し、その後に引数を指定します。サブコマンドにオプションを指定しない場合は、コマンド行のサブコマンドの後に SPACE を入力し、その後に引数を指定します。引数を 2 つ以上指定する場合は、それぞれの引数の後に SPACE を入力して区切ります。

download サブコマンドでは次の引数を指定できます。

boot フラッシュのブートモニター部分をプログラムします。フラッシュのメイン部分は、引数ファイルを使用せずにプログラムされます。

file ダウンロード用のブートまたはメインファームウェアイメージの格納場所へのパスを *file* で指定します。

file の例は、以下のとおりです。

```
/usr/platform/platform_type/lib/image/alommainfw
```

または

```
/usr/platform/platform_type/lib/image/alombootfw
```

send_event サブコマンドでは次の引数を指定できます。

"message" *message* に指定されているテキストを使用して、イベントを記述します。*message* は引用符で囲みます。

set サブコマンドでは次の引数を指定できます。

variable SC 構成 *variable* を設定します。

value SC 構成変数の *value* を設定します。

show サブコマンドでは次の引数を指定できます。

variable 指定した特定の変数の値を表示します。

useradd サブコマンドでは次の引数を指定できます。

username 新しい SC アカウント *username* を追加します。

userdel サブコマンドでは次の引数を指定できます。

username SC アカウント *username* を削除します。

userperm サブコマンドでは次の引数を指定できます。

aucr SC ユーザーアカウントのアクセス権を設定します。アクセス権を指定しない場合、4 つのアクセス権はすべて無効になり、読み取り専用アクセス権が割り当てられます。

アクセス権の定義を以下に示します。

scadm(1M)

- a ユーザーは SC 構成変数を管理または変更できません。
- u ユーザーは、ユーザーコマンドを使用して SC アカウントを変更できます。
- c ユーザーはコンソールに接続できます。
- r ユーザーは SC のリセットと、ホストの電源投入および切断ができます。

username SC アカウント *username* に対するアクセス権を変更します。

usershow サブコマンドでは次の引数を指定できます。

username SC アカウント *username* に関する情報を表示します。 *username* を指定しないと、すべてのアカウントの情報が表示されます。

userpassword サブコマンドでは次の引数を指定できます。

username *username* に対する SC パスワードを設定します。

userperm サブコマンドでは次の引数を指定できます。

username *username* に対する SC アクセス権を変更します。

使用例 例 1 SC の日付と時刻を表示する

次のコマンドは、SC の日付と時刻を表示します。

```
scadm date
```

例 2 SC の構成変数を設定する

次のコマンドは、SC の構成変数 `netsc_ipaddr` を 192.168.1.2 に設定します。

```
scadm set netsc_ipaddr 192.168.1.2
```

例 3 SC の現在の構成設定を表示する

次のコマンドは、SC の現在の構成設定を表示します。

```
scadm show
```

例 4 変数の現在の設定を表示する

次のコマンドは、`sys_hostname` という変数の現在の設定を表示します。

```
scadm show sys_hostname
```

例 4 変数の現在の設定を表示する (続き)**例 5** テキストベースの重要なイベントを送信する

次のコマンドは、重要なイベントを SC イベントログに送信し、現在の SC ユーザーに警告するとともに、イベントを syslog(3C) に送信します。

```
scadm send_event -c "The UPS signaled a loss in power"
```

例 6 参照用のテキストベースのイベントを送信する

次のコマンドは、重要ではない参照用のテキストベースのイベントを SC イベントログに送信します。

```
scadm send_event "The disk is close to full capacity"
```

例 7 ユーザーを SC に追加する

次のコマンドは、ユーザー rscroot を SC に追加します。

```
scadm useradd rscroot
```

例 8 SC からユーザーを削除する

次のコマンドは、ユーザー olduser を SC から削除します。

```
scadm userdel olduser
```

例 9 ユーザーの詳細を表示する

次のコマンドは、すべてのユーザーアカウントの詳細を表示します。

```
scadm usershow
```

例 10 特定のユーザーの詳細を表示する

次のコマンドは、ユーザーアカウント rscroot の詳細を表示します。

```
scadm usershow rscroot
```

例 11 ユーザーのアクセス権のレベルを設定する

次のコマンドは、ユーザー rscroot に全アクセス権 aucr を設定します。

```
scadm userperm rscroot aucr
```

例 12 ユーザーのアクセス権のレベルを設定する

次のコマンドは、ユーザー newuser にコンソールアクセス権 c だけを設定します。

```
scadm userperm newuser c
```

scadm(1M)

例 12 ユーザーのアクセス権のレベルを設定する (続き)

例 13 ユーザーのアクセス権のレベルを設定する

次のコマンドは、ユーザー newuser のアクセス権を読み取り専用を設定します。

```
scadm userperm newuser
```

例 14 現在のネットワークパラメタを表示する

次のコマンドは、SC の現在のネットワーク構成パラメタを表示します。

```
scadm shownetwork
```

例 15 ログ履歴を表示する

次のコマンドは、SC イベントログの最新のエントリを表示します。

```
scadm loghistory
```

例 16 詳細情報を表示する

次のコマンドは、SC とそのコンポーネントの詳細なバージョン情報を表示します。

```
scadm version -v
```

終了ステータス

次の終了値が返されます。

0 正常に終了。

0 以外 エラーが発生。

属性

以下の属性については、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	SUNWkvm

関連項目

uname(1)、syslog(3C)、attributes(5)

setup_install_server(1M)

名前 | install_scripts, add_install_client, add_to_install_server, modify_install_server,
rm_install_client, setup_install_server, check – Solaris ソフトウェアをインストールす
るためのスクリプト

形式 | `cdrom-mnt-pt/Solaris_XX/Tools/add_install_client [-i IP_address]`
 `[-e Ethernet_address] [-s server_name : path] [-c server_name : path]`
 `[-n [server] : name_service [(netmask)] [-p server_name : path]`
 `[-f boot_file_name] host_name platform_group`

`cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d`
 `[-s server_name:path] [-c server_name:path] [-p server_name:path]`
 `[-t install_boot_image_path] [-f boot_file_name] platform_name platform_group`

`cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d`
 `[-s server_name:path] [-c server_name:path] [-p server_name:path]`
 `[-t install_boot_image_path] [-f boot_file_name] -e Ethernet_address`
 `[-b property=value] platform_group`

`cdrom-mnt-pt/Solaris_XX/Tools/add_to_install_server [-s]`
 `[-p product_image_path] install_server_path`

`cdrom-mnt-pt/Solaris_XX/Tools/jumpstart_sample/check`
 `[-p install_dir_path] [-r rulesfile]`

`cdrom-mnt-pt/modify_install_server [-p]`
 `install_dir_path installer_miniroot_path`

`cdrom-mnt-pt/Solaris_XX/Tools/rm_install_client host_name`

`cdrom-mnt-pt/Solaris_XX/Tools/setup_install_server [-b]`
 `[-t install_boot_image_path] [-w wanboot_image_path] install_dir_path`

機能説明 | これらのコマンドは、Solaris Software and Solaris Installer CD のスライス 0 に置かれ
ています。Solaris CD をすでにローカルディスクにコピーしている場合、
cdrom_mnt_pt はコピーされている Solaris CD へのパスになります。これらのコマン
ドは、さまざまなインストール作業に使用できます。

Solaris_XX の XX は、使用している Solaris リリースのバージョン番号です。

add_install_client コマンドには 3 種類の形式があります。「形式」の項を参照
してください。

ネットワークを使用してインストールするクライアントを追加する場合は、次の形式
の add_install_client を使用します (これらのコマンドは bootparams(4)
ファイルを更新します)。add_install_client コマンドは、インストールサー
バーの Solaris インストールイメージ (マウントした CD またはディスクにコピーした
Solaris CD) あるいはブートサーバーの起動ディレクトリ (ブートサーバーが必要な場
合) から実行する必要があります。Solaris インストールイメージまたはブート専用
ディレクトリの Solaris リリースは、クライアントにインストールする予定の Solaris
リリースと同じでなければなりません。

`cdrom-mnt-pt/Solaris_XX/Tools/add_install_client [-i IP_address]`
 `[-e Ethernet_address] [-s server_name : path] [-c server_name : path]`

setup_install_server(1M)

```
[ -n [server] ] : name_service [ ( netmask) ] [ -p server_name : path ]
host_name platform_group
```

プラットフォームグループ内のあるプラットフォームのインスタンスをインストールサーバーに追加する場合は、次の形式の `add_install_client` コマンドを使用します。このグループは DHCP を使用して起動および構成されます。スクリプトは、必要な構成作業をサーバー上で実行し、ユーザーがそのグループのために DHCP サーバーに追加しなければならないデータを出力します。

```
cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d [-s server:path]
[-c server:path] [-p server:path] [-t install boot image path]
[-f boot file name] platform_name platform_group
```

インストールサーバーに単一のクライアントを追加する場合は、次の形式の `add_install_client` コマンドを使用します。このクライアントは DHCP を使用して起動および構成されます。スクリプトは、必要な構成作業をサーバー上で実行し、ユーザーがそのクライアントのために DHCP サーバーに追加しなければならないデータを出力します。すでに使用しているものにも、上記のように `-f` フラグを追加する必要があります。`-f` を使用すると、ユーザーは所定のクライアントに使用する起動ファイルの名前を指定できます。

```
cdrom-mnt-pt/Solaris_XX/Tools/add_install_client -d
[-s server_name:path] [-c server_name:path] [-p server_name:path]
[-t install_boot_image_path] [-f boot_file_name] -e Ethernet_address
platform_group
```

注 - x86 アーキテクチャの Pre-boot eXecution Environment (PXE) クライアントを登録する場合は、必ず `-d` オプションを使用してください。x86 PXE クライアントは構成に DHCP を使用します。

他の Solaris CD とネットワークインストールサーバー上の既存のイメージをマージする場合は、`add_to_install_server` を使用します。マージ可能な各 CD (現時点では OS CD 2 および Language CD) には専用の `add_to_install_server` スクリプトが含まれています。配布された CD に含まれているもの以外の `add_to_install_server` スクリプトを使用しないでください。

`rules` ファイル (カスタム JumpStart インストールを使用する場合に限り必要) のルールを検証する場合は、`check` を使用します。

既存のネットワークインストールサーバーのミニルートを Solaris Installation CD のミニルートに置き換える場合は、`modify_install_server` を使用します。ネットワークインストールサーバーのインストール時のユーザーインターフェースが、Solaris Installation CD の Web Start ユーザーインターフェースに変更されません。

`modify_install_server` コマンドを使用する場合は、既存のインストールイメージ (`setup_install_server` を使用して作成) があらかじめ存在していなければなりません。

setup_install_server(1M)

ネットワークインストール用のクライアントを削除する場合は、`rm_install_client` を使用します (このコマンドは `bootparams(4)` ファイルを更新します)。

Solaris CD を (インストールサーバーを設定するために) ディスクにコピーする、(WANboot インストールサーバーを設定するために) WANboot ミニルートイメージを構築する、または Solaris CD のブートソフトウェアだけを (ブートサーバーを設定するために) ディスクにコピーする場合は、`setup_install_server` を使用します。ネットワーク経由でクライアントをインストールするには、インストールサーバーが必要です。インストールサーバーとインストールするクライアントが異なるサブネットに属している場合、ネットワークインストールを行うには、ブートサーバーも必要です (ブートサーバーはクライアントのサブネットに配置する必要があります)。

オプション `add_install_client` には、次のオプションを指定できます。

-b *property=value*

ブートサーバーの TFTP ディレクトリ (デフォルトは `/tftpboot`) にあるクライアント特有の `bootenv.rc` ファイルにプロパティ値を設定します。クライアントに特有のブートプロパティを設定する場合には、このオプションを使用します。`bootpath` プロパティを使用すると、デバイス構成補助を使用して手動で選択しなくても、ブートデバイスを自動的に選択できます。`boot-args` プロパティを使用すると、JumpStart インストールを自動的に選択できます。 `eeeprom(1M)` を参照してください。

このオプションは x86 クライアントにのみ使用できます。このオプションを使用するときには、必ず `-d` オプションと `-e` を同時に指定してください。

-c *server_name:path*

このオプションが必要なのは、カスタム JumpStart インストール用の JumpStart ディレクトリを指定する場合だけです。`server_name` は、JumpStart ディレクトリが置かれているサーバーのホスト名です。`path` は JumpStart ディレクトリの絶対パスです。

-d

DHCP クライアントを指定します。

-e *Ethernet_address*

インストールするシステムの Ethernet アドレスを指定します。

-f

インストールするクライアントの `boot_file_name` を指定します。

-i *IP_address*

インストールするクライアントの IP アドレスを指定します。

-n [*server*]: *name_service*[(*netmask*)]

このオプションでは、システム構成時に使用するネームサービスを指定します。`bootparams(4)` ファイル内の `ns` キーワードを設定します。

name_service

有効なエントリは `nis`、`nisplus`、および `none` です。

setup_install_server(1M)

netmask

数字を4つずつピリオドで区切って並べたもので、IPアドレスのどの部分がネットワーク部分で、どれがホスト部分かを指定します。

server

サーバーの名前または指定したネームサービスのIPアドレス。指定したサーバーが異なるサブネット上にある場合は、クライアントからサーバーにアクセスするために、*netmask*が必要なことがあります。

-p *server_name*: *path*

このオプションは、システム情報またはネットワーク情報を前もって設定するために必要な、ユーザーが定義した *sysidcfg* ファイルの位置です。*server_name* は有効なホスト名またはIPアドレスです。*path* はJumpStart ディレクトリの絶対パスです。

-s *server_name*:*path*

このオプションが必要なのは、ブートサーバーから *add_install_client* を使用する場合だけです。サーバー名およびこのインストールで使用する Solaris インストールイメージの絶対パスを指定します。*path* はマウントされている Solaris CD のパスまたは Solaris CD のコピーが置かれているディレクトリのパスです。

add_to_install_server コマンドには、次のオプションを指定できます。

-p

コピーする CD (補助的な製品が収められている) の位置を指定します。

-s

ユーザーが、インストールする必要がある製品だけをリストから選択できるようにします。

check コマンドには、次のオプションを指定できます。

-p *install_dir_path*

使用中のシステムの *check* スクリプトではなく、指定された Solaris インストールイメージの *check* スクリプトを使用することによって、ルールファイルを検査します。*install_dir_path* は、ローカルディスク上またはマウントした Solaris CD 上の Solaris インストールイメージのパスです。

旧バージョンの Solaris が稼働しているシステムでは、このオプションを使用して、最新バージョンの *check* を実行できます。

-r *rulesfile*

rules 以外の名前のルールファイルを指定します。このオプションを使用すると、ルールの妥当性を検証してから、ルールファイルに組み込むことができます。*check* はルールが有効であるかどうかを報告するだけで、カスタム JumpStart インストールに必要な *rules.ok* ファイルは作成しません。

modify_install_server コマンドには、次のオプションを指定できます。

-p

このオプションは、*install_dir_path/Solaris_XX/Tools/Boot.orig* にすでに存在しているイメージのミニルートを保持します。

setup_install_server(1M)

setup_install_server コマンドには、次のオプションを指定できます。

- b
サーバーの設定を、ブートサーバー専用にします。
- t
代替ミニルート指定できるようにします。
- w
WANboot ミニルートイメージを構築します。

オペランド add_install_client コマンドには、次のオペランドを指定できます。

host_name

インストールするクライアントの名前です。

platform_group

特定のソフトウェアを配布するために、ベンダーが定義したハードウェアプラットフォームグループ。以下に有効なプラットフォームグループの例を示します。

システム	プラットフォームグループ
x86	i86pc
Sun Fire 4800	sun4c

システムのプラットフォームグループを調べるには、(-m オプションを指定して) `uname(1)` コマンドを使用します。

platform_name

システムのプラットフォーム名を調べるには、-i オプションを指定して `uname(1)` コマンドを使用します。

次に、`uname command` コマンドを使用して Ultra 1 のシステムプラットフォーム名を調べる例を示します。

```
uname -i
システムは、次のように応答します。
```

```
SUNW,Ultra-1
システムのプラットフォーム名は SUNW,Ultra1 です。
```

次のコマンドは、Ultra 1 用の `add_install_client` を呼び出します。

```
add_install_client -d SUNW,Ultra-1 sun4u
```

IA32 プラットフォームでは、プラットフォーム名はつねに `SUNW.i86pc` です。

次のコマンドは、IA32 プラットフォーム用の `add_install_client` を呼び出します。

```
add_install_client -d SUNW.i86pc i86pc
```

`rm_install_client` コマンドには、次のオペランドを指定できます。

setup_install_server(1M)

host_name

削除するクライアントの名前です。

setup_install_server コマンドには、次のオペランドを指定できます。

install_dir_path

Solaris ソフトウェアのコピー先となるディレクトリの絶対パスです。このディレクトリは空でなければなりません。

wanboot_image_path

WANboot ミニルートイメージを含むファイルが作成されるディレクトリの絶対パスです。

使用例 例 1 add_install_client の使用

次の add_install_client コマンドは、マウント済みの Solaris CD からネットワークインストール用のクライアントを Solaris 8 のインストールサーバーに追加します。

```
example# cd /cdrom/cdrom0/s0/Solaris_9/Tools
example# ./add_install_client system_2/sun4u
```

例 2 add_install_client の使用

次の add_install_client コマンドは、ネットワークインストール用のクライアントをインストールサーバーに追加します。カスタム JumpStart インストールを実行するためのルールファイルおよびプロファイルファイルが置かれている JumpStart のサーバーとディレクトリのパスを -c オプションで指定します。Solaris CD は Solaris 8 上の /export/install ディレクトリにコピーされています。

```
example# cd /export/install/Solaris_8/Tools
example# ./add_install_client -c install_server:/jumpstart system_1 i86pc
example# ./add_install_client -c install_server:/jumpstart system_2 i86pc
```

例 3 add_install_client の使用

次の add_install_client コマンドは、起動ファイル sun4u.solaris8 を使用する特定の sun4u プラットフォームマシン (8:0:20:99:88:77) に対するサポートを追加します。

```
example# add_install_client -d -f sun4u.solaris8 -e 8:0:20:99:88:77 sun4u
```

例 4 add_install_client の使用

次の add_install_client コマンドは、PXE 規格を使用してネットワークから起動する x86 クライアントを追加します。

```
example# add_install_client -d -s svrname:/mnt/export/root SUNW.i86pc i86p
```

例 5 add_to_install_server の使用

次の add_to_install_server コマンドは、CD 上のすべての製品ディレクトリにあるパッケージを Solaris 8 上の既存のインストールサーバーにコピーします。

例 5 add_to_install_server の使用 (続き)

```
example# cd /cdrom/cdrom0/s0
example# ./add_to_install_server /export/Solaris_8
```

例 6 check の使用

次の check コマンドは、カスタム JumpStart インストールに使用するルールファイルの構文を検査します。

```
example# cd jumpstart_dir_path
example# ./check -p /cdrom/cdrom0/s0
```

例 7 modify_install_server の使用

次の modify_install_server コマンドは、上記の setup_install_server を使用して作成されたミニルートを Boot.orig に移し、Solaris Installer CD 上のミニルートに置き換えます。

```
example# cd /cdrom/cdrom0/s0
example# ./modify_install_server -p /export/install /cdrom/cdrom0/s1
```

例 8 modify_install_server の使用

次の modify_install_server コマンドは、上記の setup_install_server を使用して作成されたミニルートを Solaris Installer CD 上のミニルートに置き換えます。

```
example# cd /cdrom/cdrom0/s0
example# ./modify_install_server /export/install /cdrom/cdrom0/s1
```

例 9 rm_install_client の使用

次の rm_install_client コマンドは、Solaris 8 上のネットワークインストール用クライアントを削除します。

```
example# cd /export/install/Solaris_8/Tools
example# ./rm_install_client holmes
example# ./rm_install_client watson
```

例 10 setup_install_server の使用

次の setup_install_server コマンドは、マウント済みの Solaris CD を Solaris 8 のローカルディスク上にある /export/install というディレクトリにコピーします。

```
example# cd /cdrom/cdrom0/s0/Solaris_8/Tools
example# ./setup_install_server /export/install
```

例 11 setup_install_server の使用

次の setup_install_server コマンドは、マウント済みの Solaris CD のブートソフトウェアを、Solaris 8 上のサブネット用ブートサーバーになるシステムの /boot_dir というディレクトリにコピーします。

setup_install_server(1M)

例 11 setup_install_server の使用 (続き)

```
example# cd /cdrom/cdrom0/s0/Solaris_8/Tools
example# ./setup_install_server -b /boot_dir
```

例 12 setup_install_server の使用

setup_install_server はデフォルトで、マウント済みの Solaris 配布ディスク上の Solaris ../Tools/Boot にあるインストールブートディレクトリを検索します。

以前に ./setup_install_server -b /boot_dir コマンドを使用して、ネットワークブートサーバー上に作成したディレクトリと別のディレクトリが必要な場合は、-t オプションを使用できます。

```
example# cd /cdrom/cdrom0/s0/Solairs_9/Tools
example# ./setup_install_server -t /boot_dir /export/install
```

例 13 setup_install_server と WANboot オプションの使用

次の setup_install_server コマンドは WANboot ミニルートファイルシステムのイメージを作成し、それを /wanboot_dir/miniroot に保存します。

```
example# cd /cdrom/cdrom0/s0/Solairs_9/Tools
example# ./setup_install_server -w /wanboot_dir /export/install
```

例 14 x86: ネットワークインストール時に使用するシリアルコンソールの指定 (CD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するシリアルコンソールを指定する方法を示します。この例では、インストールクライアントを次の方法で設定します。

- -d オプションは、クライアントが DHCP を使用してインストールパラメータを設定することを示します。
- -e オプションは、Ethernet アドレスが 00:07:e9:04:4a:bf であるクライアントでのみインストールが行われることを示します。
- 1 つめと 2 つめの -b オプションは、インストールプログラムが入力デバイスおよび出力デバイスとしてシリアルポート ttya を使用することを示します。

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
i86pc
```

-b オプションで使用できるブートプロパティの変数とその値については、eprom(1M) を参照してください。

例 15 ネットワークインストール時に使用するブートデバイスの指定 (CD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するブートデバイスを指定する方法を示します。インストールクライアントを設定するときにブートデバイスを指定する場合は、インストール時にデバイス構成補助によって出されるこの情報は表示されません。

例 15 ネットワークインストール時に使用するブートデバイスの指定 (CD) (続き)

この例では、次の方法でインストールクライアントを設定します。

- -d オプションは、クライアントが DHCP を使用してインストールパラメータを設定することを示します。
- -e オプションは、Ethernet アドレスが 00:07:e9:04:4a:bf であるクライアントでのみインストールが行われることを示します。
- 1 つめと 2 つめの -b オプションは、インストールプログラムが入力デバイスおよび出力デバイスとしてシリアルポート ttya を使用することを示します。
- 3 つめの -b オプションは、インストールプログラムがインストール時に特定のブートデバイスを使用することを示します。
- ブートデバイスのパスは、使用しているハードウェアによって異なります。
- i86pc というプラットフォーム名は、クライアントが x86 ベースのシステムであることを示します。

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
-b "bootpath=/pci@0,0/pci108e,16a8@8" i86pc
```

-b オプションで使用できるブートプロパティの変数とその値について詳細は、eeprom(1M) を参照してください。

終了ステータス 次の終了ステータスが返されます。

- 0 正常終了
- 1 エラーが発生した

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	Solaris CD

関連項目 uname(1), bootparams(4), attributes(5)

『Solaris 9 インストールガイド』

share(1M)

名前	share - ローカル資源をリモートシステムからマウントできるようにする
形式	share [-F <i>FSType</i>] [-o <i>specific_options</i>] [-d <i>description</i>] [<i>pathname</i>]
機能説明	share コマンドはローカル資源をエクスポートして、 <i>FSType</i> のリモートファイルシステムからマウントできるようにします。オプション -F <i>FSType</i> を指定しない場合、 <code>/etc/dfs/fstypes</code> 内の最初のファイルシステムタイプがデフォルトとして使用されます。NFS に固有なオプションについては、 <code>share_nfs(1M)</code> のマニュアルページを参照してください。 <i>pathname</i> は共有するディレクトリのパス名です。引数をなにも指定しないと、share は共有しているすべてのファイルシステムを表示します。
オプション	<p>次のオプションを指定できます。</p> <p>-F <i>FSType</i> ファイルシステムタイプを指定します。</p> <p>-o <i>specific_options</i> <i>specific_options</i> を使用すると、共有する資源へのアクセスを制御できます。NFS に固有なオプションについては、<code>share_nfs(1M)</code> のマニュアルページを参照してください。<i>specific_options</i> は次のとおりです。</p> <p>rw すべてのクライアントが <i>pathname</i> を読み書き両方で共有します。これはデフォルトの動作です。</p> <p>rw=<i>client</i>[:<i>client</i>]... 指定されたクライアントだけが <i>pathname</i> を読み書き両方で共有します。他のシステムは <i>pathname</i> にアクセスできません。</p> <p>ro すべてのクライアントが <i>pathname</i> を読み取り専用で共有します。</p> <p>ro=<i>client</i>[:<i>client</i>]... 指定されたクライアントだけが <i>pathname</i> を読み取り専用で共有します。他のシステムは <i>pathname</i> にアクセスできません。</p> <p>複数のオプションはコロンで区切ります。</p> <p>-d <i>description</i> -d オプションは、共有する資源の説明に使用します。</p>
使用例	<p>例 1 ファイルシステムを読み取り専用で共有する</p> <p>次の例では、起動時に、(すべてのクライアントが) <code>/disk</code> ファイルシステムを読み取り専用で共有します。</p>

例1 ファイルシステムを読み取り専用で共有する (続き)

```
share -F nfs -o ro /disk
```

例2 複数のオプションを呼び出す

次の例では、netgroup_name 内のメンバーは読み取り専用で、hostname 内のユーザーは読み書き両用で、/export/manuals ファイルシステムを共有します。

```
share -F nfs -o ro=netgroup_name:rw=hostname /export/manuals
```

ファイル	/etc/dfs/dfstab	起動時に実行する share コマンドのリスト
	/etc/dfs/fstypes	ファイルシステムタイプのリスト (デフォルトは NFS)
	/etc/dfs/sharetab	共有するファイルシステムのシステム記録
属性	次の属性については、attributes(5) のマニュアルページを参照してください。	

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 mountd(1M), nfsd(1M), share_nfs(1M), shareall(1M), unshare(1M), attributes(5)

注意事項 エクスポート (古い用語): SunOS 4.x では、ファイルシステムの共有は「エクスポート」と呼ばれていました。そのため、share コマンドは exportfs(1B) または /usr/sbin/exportfs として呼び出されました。

share コマンドを同じファイルシステムに複数回呼び出した場合、最後に呼び出した share コマンドが前の share コマンドに優先します。つまり、最後の share コマンドで設定したオプションが前のオプションに置き換わります。たとえば、/somefs を usera に読み書き両用のアクセス権を与えた場合、/somefs を userb にも読み書き両用のアクセス権が与えられます。

```
example% share -F nfs -o rw=usera:userb /somefs
```

この動作はルートファイルシステムの共有だけに制限されず、すべてのファイルシステムに適用されます。

shareall(1M)

名前 shareall, unshareall – 複数リソースの共有または共有解除

形式 **shareall** [-F *FSType* [,*FSType*...]] [-| *file*]
unshareall [-F *FSType* [,*FSType*...]]

機能説明 引数を指定せずに **shareall** を使用すると、*file* (share コマンド行のリストを含む) 内に指定されているすべてのリソースが共有されます。オペランドにハイフン (-) を指定すると、share コマンド行は標準入力から読み取られます。*file* とハイフンのどちらも指定しないと、ファイル /etc/dfs/dfstab がデフォルトで使用されます。

複数のファイルシステムタイプをコンマで区切り、-F の引数として指定すると、特定のファイルシステムタイプのみリソースが共有されます。

unshareall は、現在共有されているすべてのリソースの共有を解除します。-F フラグを指定しないと、すべてのファイルシステムタイプのリソースについて共有が解除されます。

オプション -F *FSType* ファイルシステムを指定します。デフォルトは、
/etc/dfs/fstypes 内の最初のエントリです。

ファイル /etc/dfs/dfstab

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 share(1M), unshare(1M), attributes(5)

shutdown(1M)

名前	shutdown – システムの停止とシステム状態の変更
形式	<code>/usr/sbin/shutdown [-y] [-g grace-period] [-i init-state] [message]</code>
機能説明	<p>shutdown は、マシンの状態を変更するために、スーパーユーザーが実行します。通常は、マルチユーザー状態 (state 2) からほかの状態に移行するために使用されます。</p> <p>デフォルトでは、shutdown は、コンソール以外はオペレーティングシステムにアクセスできないシステム状態にします。この状態はシングルユーザー状態と呼ばれます。</p> <p>デーモンを停止させたりプロセスを強制終了させたりする前に、shutdown は警告メッセージを出力するとともに、デフォルトでは確認のための最終メッセージも出力します。message は、標準の警告メッセージ 'The system will be shut down in ...' に続いて出力される文字列です。この文字列に 2 つ以上の語句を含める場合は、一重引用符 (') または二重引用符 (") で囲む必要があります。</p> <p>この警告メッセージとユーザーが作成した message は、shutdown コマンドが開始されるまでの残り時間が 7200 秒、3600 秒、1800 秒、1200 秒、600 秒、300 秒、120 秒、60 秒、および 30 秒の時点で出力されます。「使用例」の項を参照してください。</p> <p>各システム状態の定義は次のとおりです。</p> <p>state 0 オペレーティングシステムを停止します。</p> <p>state 1 state 1 は管理状態と呼ばれます。state 1 では、マルチユーザーの操作に必要なファイルシステムがマウントされ、マルチユーザーファイルシステムにアクセスする必要があるログインを使用できます。システムがファームウェアモードから state 1 に移行する時はコンソールだけがアクティブであり、ほかのマルチユーザー (state 2) サービスは使用できません。マルチユーザー状態から state 1 への移行時にすべてのユーザープロセスが停止されるわけではないことに注意してください。</p> <p>state s, S state s (または S) はシングルユーザー状態と呼ばれます。この状態への移行時にはすべてのユーザープロセスが停止されます。シングルユーザー状態では、マルチユーザーログインに必要なファイルシステムのマウントが解除され、システムへのアクセスはコンソールを介してしか行えません。マルチユーザーファイルシステムへアクセスする必要があるログインは使用できません。</p> <p>state 5 電源を切っても安全なようにマシンを停止し、可能であれば電源を自動的に切断します。rc0 プロシージャがこのために呼び出されます。</p> <p>state 6 オペレーティングシステムを停止したあと、/etc/inittab の initdefault エントリに定義されている状態でリポートします。rc6 プロシージャがこのために呼び出されます。</p>
オプション	次のオプションを指定できます。

shutdown(1M)

-y ユーザーの介入なしでコマンドが実行されるように、確認の問い合わせにあらかじめ応答します。

-g *grace-period* デフォルトの 60 秒を、スーパーユーザーが変更できるようにします。秒数を指定します。

-i *init-state* `init` が移行する状態を指定します。デフォルトは、システム状態 `s` です。

使用例 例 1 shutdown の使用

この例では、ホスト `foo` で `shutdown` が実行されており、120 秒後に停止がスケジュールされています。警告メッセージは、最終の確認メッセージの 2 分前、1 分前、および 30 秒前に出力されます。

```
example# shutdown -i S -g 120 "==== disk replacement ====="
Shutdown started.  Tue Jun  7 14:51:40 PDT 1994

Broadcast Message from root (pts/1) on foo Tue Jun  7 14:51:41. . .
The system will be shut down in 2 minutes
==== disk replacement =====
Broadcast Message from root (pts/1) on foo Tue Jun  7 14:52:41. . .
The system will be shut down in 1 minutes
==== disk replacement =====
Broadcast Message from root (pts/1) on foo Tue Jun  7 14:53:41. . .
The system will be shut down in 30 seconds
==== disk replacement =====
Do you want to continue? (y or n):
```

ファイル /etc/inittab `init` によるプロセスディスパッチを制御します。

属性 次の属性については、`attributes(5)` のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 `boot(1M)`, `halt(1M)`, `init(1M)`, `killall(1M)`, `reboot(1M)`, `ufsdump(1M)`, `init.d(4)`, `inittab(4)`, `nologin(4)`, `attributes(5)`

注意事項 システムが `s` (または `s`) 状態に移行するときは、`/etc/nologin` ファイル (`nologin(4)` のマニュアルページを参照) が作成されます。その後 `state 2` (マルチユーザー状態) へ移行する時点で、このファイルは `/etc/rc2.d` ディレクトリにあるスクリプトによって削除されます。

su(1M)

名前	su - スーパーユーザーまたは別のユーザーに変更
形式	su [-] [<i>username</i> [<i>arg...</i>]]
機能説明	<p>su コマンドを使用すると、ログオフをしないで別のユーザーまたは別の役割になることができます。デフォルトの <i>username</i> は root (スーパーユーザー) です。</p> <p>su を使用するときには、適切なパスワードを入力する必要があります (ユーザーがすでに root の場合は不要)。パスワードが正しければ、指定した <i>username</i> 用に設定されている実ユーザー ID、実効ユーザー ID、グループ ID、補助グループリストを持つ新しいシェルプロセスが生成されます。新しいシェルは、<i>username</i> のパスワードファイルエントリ (passwd(4) を参照) のシェルフールドで指定されたシェルです。シェルが指定されていないと、/usr/bin/sh (sh(1) を参照) になります。スーパーユーザー権限が必要な場合に、exec(2) によってスーパーユーザーのシェルを起動できないときは、代わりに /sbin/sh が使用されます。元のユーザー ID の権限に戻る場合は、EOF 文字 (CTRL-D) を入力して新しいシェルを終了します。</p> <p>コマンド行に引数を指定すると、その引数が新しいシェルに渡されます。sh などのプログラムを実行している場合は、<i>arg</i> (引数) に <i>-c string</i> と指定すると、シェルによって <i>string</i> が実行されます。また、引数に <i>-r</i> を指定すると、ユーザーに制限付きのシェルが与えられ提供されます。</p> <p>次の説明は、指定したユーザーのパスワードファイルのエントリに、ログインシェルとして /usr/bin/sh が指定されている場合と、何も指定されていない場合にだけのみ有効です。後者の場合、ログインシェルはデフォルトで /usr/bin/sh になります。su の最初の引数に <i>-</i> (ダッシュ) を指定すると、指定したユーザーとして実際にログインした場合と同じ環境が渡されます。最初の引数に <i>-</i> (ダッシュ) を指定しない場合、\$PATH 以外の環境が渡されます。\$PATH は /etc/default/su 中で PATH と SUPATH によって制御されます。また、引数に <i>-</i> (ダッシュ) を指定した場合は、ユーザーのプロジェクト ID が設定されます。settaskid(2) のマニュアルページを参照してください。</p> <p>su を実行して別のユーザーになる操作は、すべてログファイル /var/adm/sulog に記録されます (sulog(4) を参照)。</p>
セキュリティ	<p>su は、pam(3PAM) を使って、認証、アカウント管理、セッション管理を行います。su で使用するモジュールを指定する PAM 構成ポリシーは、/etc/pam.conf に記述されています。次の例は、su コマンド (UNIX 認証、アカウント管理、セッション管理モジュールを実行) エントリが記述されている pam.conf ファイルの抜粋を示します。</p> <pre> su auth requisite pam_authok_get.so.1 su auth required pam_dhkeys.so.1 su auth required pam_unix_auth.so.1 su account required pam_unix_roles.so.1 su account required pam_unix_projects.so.1 su account required pam_unix_account.so.1 su session required pam_unix_session.so.1 </pre>

使用例	<p>su サービスのエントリがない場合は <code>other</code> のサービスのエントリを使用します。複数の認証モジュールが記述されている場合、複数のパスワードが必要になることがあります。</p> <p>例 1 以前にエクスポートした環境を維持してユーザー <code>bin</code> になる</p> <p>以前にエクスポートした環境のままユーザー <code>bin</code> になるには、以下のコマンドを実行します。</p> <pre>example% su bin</pre> <p>例 2 ユーザー <code>bin</code> になり <code>bin</code> のログイン環境に変更する</p> <p>ユーザー <code>bin</code> になり、<code>bin</code> がログインした場合と同じ環境に切り替えるには、以下のコマンドを実行します。</p> <pre>example% su - bin</pre> <p>例 3 ユーザー <code>bin</code> の環境とアクセス権でコマンドを実行する</p> <p>ユーザー <code>bin</code> の一時的な環境とアクセス権を使ってコマンド <code>command</code> を実行する場合は、以下のように入力します。</p> <pre>example% su - bin -c "command args"</pre>
環境	<p>セキュリティ上の理由から、接頭辞 <code>LD_</code> が付いている環境変数は削除されました。このため、以前にユーザー <code>bin</code> でエクスポートした接頭辞 <code>LD_</code> 付きの環境変数は、<code>su bin</code> を実行しても使用できません。</p> <p><code>LC_*</code> 変数 (<code>LC_CTYPE</code>、<code>LC_MESSAGES</code>、<code>LC_TIME</code>、<code>LC_COLLATE</code>、<code>LC_NUMERIC</code>、<code>LC_MONETARY</code>) (<code>environ(5)</code> 参照) が設定されていない環境では、環境変数 <code>LANG</code> によって、各ロケールカテゴリの <code>su</code> の動作が決定します。もし、<code>LC_ALL</code> が設定されていれば、その内容が <code>LANG</code> 変数やその他の <code>LC_*</code> 変数より優先されます。上記の変数がどれも設定されていなければ、C ロケール (米国の形式) によって <code>su</code> の動作が決定します。</p> <p><code>LC_CTYPE</code> <code>su</code> が文字を処理する方法を決定します。<code>LC_CTYPE</code> に有効な値が設定されていると、そのロケールで有効な文字を含むテキストやファイル名を表示または処理できます。また、拡張 UNIX コード (EUC) の表示または処理も可能です (1 ~ 3 バイト幅の文字を使用)。このほか、1 カラム幅、2 カラム幅、またはそれ以上のカラム幅の EUC 文字も処理できます。C ロケールでは、ISO 8859-1 の文字だけが有効です。</p> <p><code>LC_MESSAGES</code> 診断メッセージや情報メッセージの表示方法を決定します。また、メッセージの言語とスタイル、肯定・否定の応答形式も決定します。C ロケールでは、メッセージはプログラム自身が使用しているデフォルトの形 (通常、米語) で表示されます。</p>
ファイル	<pre>\$HOME/.profile sh および ksh ユーザーのログインコマンド</pre> <pre>/etc/passwd システムのパスワードファイル</pre>

su(1M)

/etc/profile システム全体の sh と ksh のログインコマンド

/var/adm/sulog ログファイル

/etc/default/su このファイル中に指定できるデフォルトパラメータは以下のとおりです。

SULOG 定義されている場合、別のユーザーになるための su の試行はすべて、指定されたファイルに記録されます。

CONSOLE 定義されている場合、root になるための su の試行はすべて、コンソールに記録されます。

PATH デフォルトパス (/usr/bin:)

SUPATH root になるために su を起動しているユーザーのデフォルトパス (/usr/sbin:/usr/bin)

SYSLOG すべての su 試行を記録するのに syslog(3C) の LOG_AUTH を使うかどうかを指定します。
LOG_NOTICE メッセージは root への su に対して生成され、
LOG_INFO メッセージはその他のユーザーへの su に対して生成されます。
LOG_CRIT メッセージは、su に失敗した場合に生成されません。

SLEEPTIME 存在していれば、ログインに失敗したことが画面上に表示され、次のログインが入力できるまでの待ち時間を秒単位で設定します。デフォルト値は 4、最小値は 0、最大値は 5 です。

属性 次の属性については attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 csh(1), env(1), ksh(1), login(1), roles(1), sh(1), syslogd(1M), exec(2), getdefaultproj(3PROJECT), setproject(3PROJECT), pam(3PAM), syslog(3C), pam.conf(4), passwd(4), profile(4), sulog(4), attributes(5), environ(5),

su(1M)

pam_authtok_check(5), pam_authtok_get(5), pam_authtok_store(5),
pam_dhkeys(5), pam_passwd_auth(5), pam_unix(5), pam_unix_account(5),
pam_unix_auth(5), pam_unix_session(5)

注意事項

pam_unix(5) モジュールは、将来のリリースではサポートされなくなる可能性があります。同様の機能は、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)、および pam_unix_session(5) で提供されています。

SUNWafb_config(1M)

名前	afbconfig, SUNWafb_config – AFB グラフィックスアクセラレータの設定
形式	<pre>/usr/sbin/afbconfig [-dev <i>device-filename</i>] [-res <i>video-mode</i> [now try] [noconfirm nocheck]] [-file machine system] [-deflinear true false] [-defoverlay true false] [-overlayorder first last] [-expvis enable disable] [-sov enable disable] [-maxwinds <i>n</i>] [-extovl enable disable] [-g <i>gamma-correction-value</i>] [-gfile <i>gamma-correction-file</i>] [-propt] [-prconf] [-defaults] /usr/sbin/afbconfig [-propt] [-prconf] /usr/sbin/afbconfig [-help] [-res ?]</pre>
機能説明	<p>afbconfig は、AFB グラフィックスアクセラレータおよび AFB 対応の X11 ウィンドウシステムのデフォルトの一部を設定します。</p> <p>afbconfig の次の形式では、指定したオプションを OWconfig ファイルに保存します。</p> <pre>/usr/sbin/afbconfig [-dev<i>device-filename</i>] [-res <i>video-mode</i> [now try] [noconfirm nocheck]] [-file machine system] [-deflinear true false] [-defoverlay true false] [-overlayorderfirst last] [-expvisenable disable] [-sov enable disable] [-maxwinds<i>n</i>] [-extovl enable disable] [-g<i>gamma-correction-value</i>] [-gfile<i>gamma-correction-file</i>] [-propt] [-prconf] [-defaults]</pre> <p>これらのオプションは、次にウィンドウシステムをそのデバイスで実行するときに AFB デバイスを初期化するために使用されます。OWconfig ファイル内のオプションの更新は、異なるウィンドウセッションや再起動したシステムでも有効となります。</p> <p>-prconf、-propt、-help、-res ? オプションだけを起動する次の形式では、OWconfig ファイルは更新されません。</p> <pre>/usr/sbin/afbconfig [-propt] [-prconf] /usr/sbin/afbconfig [-help] [-res ?]</pre> <p>また、次の形式では、その他のオプションはすべて無視されます。</p> <pre>/usr/sbin/afbconfig [-help] [-res ?]</pre> <p>オプションは、一度に 1 つの AFB デバイスに対してのみ指定することができます。複数の AFB デバイスに対してオプションを指定するには、afbconfig コマンドを複数回起動する必要があります。</p> <p>afbconfig で指定できるのは、AFB 固有のオプションだけです。デフォルトの表示色数、デフォルトのビジュアルクラスなどを指定する通常のウィンドウシステムのオプションは、openwin コマンド行のデバイス修飾子で指定してください。</p>

ユーザーは、更新する OWconfig ファイルを指定することもできます。デフォルトでは、/usr/openwin ディレクトリツリーにあるマシン固有のファイルが更新されます。別のファイルを指定するには、-file オプションを使用します。たとえば、/etc/openwin ディレクトリツリーにあるシステム共通の OWconfig ファイルを代わりに更新することができます。

これらの標準 OWconfig ファイルのどちらもスーパーユーザーのみが書き込みを行えます。したがって、スーパーユーザーが所有する afbconfig プログラムは、setuid による root の権限で実行されます。

オプションのデフォルト値

afbconfig コマンド行で指定されていないオプションについては、対応する OWconfig ファイル中のオプションは更新されず、ファイル内の値がそのまま使用されます。ウィンドウシステムを実行する際に、afbconfig による AFB オプションの指定がまったくなかった場合は、デフォルト値が使用されます。オプションのデフォルトは次のとおりです。

```
-dev          /dev/fbs/afb0
-file         machine
-res          none
-deflinear    false
-defoverlay   false
-linearorder  last
-overlayorder last
-expvis       enabled
-sov          enabled
-maxwids      32
-extovl       enabled
-g            2.22
```

-res オプションのデフォルト値 none とは、ウィンドウシステムが実行された場合に、画面解像度がそのデバイスに現在プログラムされている表示モードになることを意味しています。

これによって、PROM によってデバイスの解像度を指定しているユーザーとの共用性が保てます。(Gx などの)一部のデバイスでは、PROM が表示モードを指定する唯一の手段です。これは、デフォルトの AFB 表示モードは、最終的に PROM によって決まることを意味しています。

オプション

次のオプションがサポートされています。

```
-defaults
  すべてのオプションの値をそれぞれのデフォルト値に戻します。
```

SUNWafb_config(1M)

-deflinear true | false

AFB には、2 種類の画像表示形式があります。リニア画像と非リニア画像です。リニア画像はガンマ補正され、非リニア画像は補正されません。リニア画像版も非リニア画像版も、ともに持つ画像表示形式が 2 つあります。24 ビットトゥルーカラーと 8 ビットスタティックグレーです。

true のときは、デフォルトの画像表示形式として、デフォルトで選択されたオプション (特に、Xsun(1) の defdepth および defclass オプション。詳細は OpenWindows のマニュアルページを参照) を満たすリニア画像がセットされます。

false のとき、または他のデフォルトで選択されたオプションを満たすリニア画像がないときは、これらの他のオプションを満たす、非リニア画像がデフォルトとして選択されます。AFB にはリニアオーバーレイ画像表示形式がないため、-defoverlay オプションが存在するときには、このオプションを使用することはできません。

-defoverlay true | false

AFB が、残りの AFB 画像から切り離されたピクセルを持つ 8 ビット疑似カラー画像を提供します。これを、オーバーレイ画像といいます。この画像表示形式で作成されたウィンドウは、他の画像表示形式で作成されたウィンドウに影響を与えません。逆に、他の画像表示形式で作成されたウィンドウは、オーバーレイウィンドウに影響を与えます。

この画像を使用して作成されたウィンドウで使用することができる色の数は -extov1 オプションの設定に依存します。-extov1 オプションが有効になっている場合は、256 種類の不透明カラーの値による拡張オーバーレイを使用することができます(-extov1 を参照)。-extov1 が無効になっている場合は、拡張オーバーレイを使用することはできず、この画像には、不透明カラーの (256 -maxwids) の値が使用されます (-maxwids を参照)。

-defoverlay の値が true である場合には、オーバーレイ画像がデフォルト画像になります。-defoverlay の値が false の場合には、他のデフォルトで選択された def、depth および defclass オプションを満たすオーバーレイでない画像表示形式が、デフォルトの画像表示形式として選択されます。詳細は、OpenWindows のマニュアルページを参照してください。

-defoverlay true オプションが使用される場合は、常に openwin コマンド行で選択されたデフォルトの深さとクラスは 8 ビット疑似カラーである必要があります。それ以外の場合は、警告メッセージが出力され、-defoverlay オプションは false として扱われます。

-deflinear オプションが存在するときには、AFB にはリニアオーバーレイ画像表示形式がないため、このオプションは使用することができません。

-dev device-filename

AFB 特殊ファイルを指定します。デフォルトは /dev/fbs/afb0 です。

-expvis enable | disable

enable にすると、OpenGL Visual Expansion が起動されます。選択された画像表示形式グループ (8 ビット PseudoColor、24 ビット TrueColor など) が画像表示形式リストに見つかります。

-extovl enable | disable

enable にすると、拡張オーバーレイを使用することができます。このオーバーレイ画像には 256 種類の不透明カラーがあります。SOV 画像には 255 種類の不透明カラーと 1 種類の透明カラーがあります。

また、このオプションは、ハードウェアによる透明カラーを有効にするため、SOV 画像を使用するウィンドウで、より高い性能が得られます。

-file machine | system

更新する OWconfig ファイルを指定します。machine が指定された場合は、/etc/openwin ディレクトリツリーにあるマシン固有の OWconfig ファイルが使用されます。system が指定された場合は、/usr/openwin ディレクトリツリーにある共通の OWconfig ファイルが使用されます。指定されたファイルがない場合は、新たに生成されます。

-g gamma-correction-value

ガンマ補正の値を変更することができます。すべてのリニア画像ではガンマ補正を使用することができます。デフォルトでは、*gamma-correction-value* は 2.22 です。0 より小さい値は無効 (不正) です。ガンマ補正の値はリニア画像に適用され、リニア画像の有効ガンマ値は 1.0 になります。これは、XSolarisGetVisualGamma(3) によって返される値です。この機能については、XSolarisGetVisualGamma(3) を参照してください。

このオプションは、ウィンドウシステムが稼動しているときに使用することができます。ガンマ補正の値を変更すると、リニア画像を使用して表示されているすべてのウィンドウが影響を受けます。

-gfile gamma-correction-file

指定されたファイル (*gamma-correction-file*) からガンマ補正表を読み込みます。*gamma-correction-file* は、各行が R、G、B チャネルの値を持つように書式化されている必要があります。それらの値は、16 進数で指定し、値と値の間は 1 つ以上のスペースで区切ります。また、*gamma-correction-file* は、そのような 3 つの値の組が 256 種類定義されます。

gamma-correction-file の例を以下に示します。

```
0x00 0x00 0x00
0x01 0x01 0x01
0x02 0x02 0x02
...
...
0xff 0xff 0xff
```

このオプションを使用することによって、ウィンドウシステムが稼動しているときにガンマ補正表を読み込むことができます。新しいガンマ補正は、このリニア画像によって表示されているすべてのウィンドウに影響を与えます。ユーザーが指定した表によってガンマ補正を行う際は、ガンマ補正の値は定義されません。デフォルト

SUNWafb_config(1M)

トでは、ウィンドウシステムはガンマ補正值として 2.22 を使用し、このガンマ補正值に対応してウィンドウシステムが作成したガンマ補正表を読み込みます。

-help

afbconfig コマンド行のオプションと機能の概要を一覧で表示します。

-linearorder first | last

first のときには、AFB スクリーン用の X11 スクリーン画像表示形式リスト上で、リニア画像が非リニア画像より前に表示されます。last のときには、非リニア画像は、リニア画像より前に表示されます。

-maxwidths *n*

ウィンドウ ID s (WIDs) として使用するために予約される最大数の AFB X チャンネルピクセル値を指定します。オーバーレイカラーマップのピクセル値の残りは、通常の X11 の未使用のカラーピクセルのために使用されます。確保された WIDs は、(XGL などの) 3 次元グラフィックスウィンドウ、MBX ウィンドウと、デフォルト以外の画像表示形式をもつウィンドウにより発生順に割り当てられます。X チャンネルコードの 0 から (255 - *n*) は、未使用のカラーピクセルです。(255 - *n* + 1) から 255 の X チャンネルコードは、WID として使用するために予約されます。適切な値は、1、2、4、8、16、32、64 です。

このオプションは -extov1 が無効になっている場合のみ使用することができます。

-overlayorder first | last

first のときには、AFB スクリーン用の X11 スクリーン画像表示形式リスト上で、8 ビット疑似カラーオーバーレイ画像が、非オーバーレイ画像より前に表示されます。last のときには、非オーバーレイ画像は、オーバーレイ画像より前に表示されます。

-propt

-file オプションで指定された OWconfig ファイルに書かれた AFB オプションの値のうち、-dev オプションで指定されたデバイスに対するものすべてを表示します。afbconfig の呼び出しが終了した後に、OWconfig ファイルに書き込まれるオプションの値を表示します。

次に表示例を示します。

```
--- OpenWindows Configuration for /dev/fbs/afb0 ---
OWconfig: machine
Video Mode: 1280x1024x76
Default Visual: Non-Linear Normal Visual
Visual Ordering: Linear Visuals are last
                  Overlay Visuals are last
OpenGL Visual Expansion: enabled
Server Overlay Visuals: enabled
Extended Overlay: enabled
Underlay WIDs: 64 (not configurable)
Overlay WIDs: 4 (not configurable)
Gamma Correction Value: 2.220
Gamma Correction Table: Available
```

-prconf

AFB ハードウェア構成を表示します。

次に表示例を示します。

```
--- Hardware Configuration for /dev/fbs/afb0 ---
Type: double-buffered AFB with Z-buffer
Board: rev 0 (Horizontal)
Number of Floats: 6
PROM Information: @(#)afb.fth x.xx xx/xx/xx
AFB ID: 0x101df06d
DAC: Brooktree 9070, version 1 (Pac2)
3DRAM: Mitsubishi 130a, version x
EDID Data: Available - EDID version 1 revision x
Monitor Sense ID: 4 (Sun 37x29cm RGB color monitor)
Monitor possible resolutions: 1024x768x77, 1024x800x84, 1
                             1152x900x76, 1280x1024x67, 1280x1024x76, 960x680xx108s
Current resolution setting: 1280x1024x76
```

-sov enable | disable

enable にすると、ルートウィンドウの `SERVER_OVERLAY_VISUALS` 属性が有効になります。SOV 画像が転送され、それらの透過タイプ、値、階層は、この属性によって参照することができます。**disable** にすると、`SERVER_OVERLAY_VISUALS` 属性は定義されません。SOV 画像は転送されません。

-res video-mode [now | try [noconfirm | nocheck]]

指定した AFB デバイスに接続されているモニターを制御する際に使われる表示モードを指定します。

組み込まれている表示モードの形式は次のとおりです。 `widthxheightxrate width` はピクセル単位のスクリーン幅、`height` はピクセル単位のスクリーンの高さ、`rate` は画面を垂直方向に再描画する周期です。

960x680x112s や 960x680x108s の `s` 接尾辞は、これらが立体表示モードであることを意味します。640x480x60i や 768x575x50i の `i` 接尾辞は、インタレース表示タイミングを有効にします。この接尾辞がない場合は、ノンインタレースタイミングが使用されます。

便宜上、`-res` にリフレッシュレートを指定する際、値の直前に `x` の代わりに `@` を使用できます。たとえば、1280x1024@76 のように指定できます。AFB が対応している一部の表示モードには、モニターが対応していない場合があります。また、AFB がサポートする表示モードにも、モニターがサポートしていないものがあります。AFB デバイスとモニターの両方がサポートしている表示モードのリストは、`-res ?` オプション付きの `afbconfig` (形式の項に記された 3 番目の形式) を実行することによって得ることができます。

AFB がサポートしている表示モードのリストを次に示します。

```
1024x768x60
1024x768x70
1024x768x75
```

SUNWafb_config(1M)

1024x768x77
1024x800x84
1152x900x66
1152x900x76
1280x800x76
1280x1024x60
1280x1024x67
1280x1024x76
960x680x112s (立体表示)
960x680x108s (立体表示)
640x480x60
640x480x60i (インタレース)
768x575x50i (インタレース)

便宜上、AFB がサポートしている表示モードのいくつかには記号名が定義されています。*widthxheightxrate* の形式の代わりに、記号名を *-res* オプションの引数として指定することができます。記号名 *none* は、ウィンドウシステムを実行すると、画面の解像度は現在デバイスにプログラムされている表示モードになることを意味します。

AFB がサポートしている表示モードのリストを次に示します。

記号名 対応する表示モード

svga 1024x768x60
1152 1152x900x76
1280 1280x1024x76
stereo 960x680x112s
ntsc 640x480x60i
pal 768x575x50i
none (上記参照)

-res オプションには、表示モードの直後に次の追加引数を指定することができます。追加引数は、単独でも複数でも指定できます。

noconfirm *-res* オプションを指定した際に、システムが使用不可であっても、表示出力のない状態になる場合があります。このような状況は、特定のコードが読み込まれた際のモニターセンスコードにあいまいさがあった場合などに発生します。このような事態を避けるために *afbconfig* のデフォルトの動作では、この問題についての警告メッセージと、処理を継続するかどうかを確認するメッセージを表示します。*noconfirm* オプションを指定すると、*afbconfig* コマンドはこの確認をせずに、要求のあった表示モードにプログラムします。このオプションは、*afbconfig* がシェルスクリプトから実行されている場合に便利です。

nocheck このオプションを指定すると、モニターセンスコードに基づく通常のエラーチェックが行われません。ユーザーによって指定された表示モードは、現在接続されているモニターに適切かど

うかにかかわらず受け付けられます。このオプションは、AFB デバイスに異なるモニターを接続する場合に便利です。このオプションの指定は、noconfirm の指定も兼ねます。

now OWconfig ファイルの表示モードを更新するとともに、AFB デバイスが指定した表示モードにただちにプログラムされます。この機能は、ウィンドウシステムを開始する前に表示モードを変更する際に便利です。

対象となるデバイスが稼働している間 (たとえば、ウィンドウシステムの稼働中) に、この引数をafbconfig に指定することはお勧めしません。予期しない結果になることもあります。now 引数を指定してafbconfig コマンドを実行する場合は、最初にウィンドウシステムを終了してください。now 引数がウィンドウシステムのセッション中に使用された場合、表示モードはただちに變更されますが、画面の幅や高さはそのセッションが終了して次のセッションに入るまで變更されません。さらに、立体表示モードではシステムが變更を認識しないことがあります。したがって、ウィンドウシステムの稼働中には絶対に now オプションを指定しないでください。

try このオプションを指定すると、指定した表示モードが試験的にプログラムされます。ユーザーは、指定した表示モードを使用する場合は、メッセージが表示されてから 10 秒以内に「y」を入力します。表示されたモードを使用しない場合は、10 秒以内に任意の文字を入力します。「y」または Return キー以外の文字の入力は、すべて「使用しない」と判断され、以前の表示モードに戻されるため、OWconfig ファイル中の表示モードは書き換えられません (その他の指定されたオプションは有効となります)。Return キーの入力があった場合は、新しい表示モードを保持するかどうかを yes または no で確認するメッセージが表示されます。このオプションの指定は、now 引数の指定も兼ねます (now 引数の注意を参照)。

使用例 例 1 モニターの種類の変更

モニターの種類を、垂直周波数 76 Hz で解像度 1280x1024 に変更する例を以下に示します。

```
example% /usr/sbin/afbconfig -res 1280x1024x76
```

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	SUNWafbcbf

SUNWafb_config(1M)

関連項目 | mmap(2), attributes(5)

名前	ffbconfig, SUNWffb_config – FFB グラフィックスアクセラレータの設定
形式	<pre> /usr/sbin/ffbconfig [-dev <i>device-filename</i>] [-res <i>video-mode</i> [now try] [noconfirm nocheck]] [-file machine system] [-deflinear true false] [-defoverlay true false] [-linearorder first last] [-overlayorder first last] [-expvis enable disable] [-sov enable disable] [-maxwids <i>n</i>] [-extovl enable disable] [-g <i>gamma-correction-value</i>] [-gfile <i>gamma-correction-file</i>] [-propt] [-prconf] [-defaults] /usr/sbin/ffbconfig [-propt] [-prconf] /usr/sbin/ffbconfig [-help] [-res ?] </pre>
機能説明	<p>ffbconfigは、FFB グラフィックスアクセラレータおよび FFB 対応の X11 ウィンドウシステムのデフォルトの一部を設定します。</p> <p>ffbconfigの1番目の形式では、指定したオプションを OWconfigファイルに保存します。これらのオプションは、次にウィンドウシステムをそのデバイスで実行するときに FFB デバイスを初期化するために使用されます。OWconfig ファイル内のオプションの更新は、異なるウィンドウセッションや再起動したシステムでも有効となります。</p> <p>-prconf、-propt、-help、-res? オプションだけを起動する2番目と3番目の形式では、OWconfig ファイルは更新されません。また、3番目の形式では、その他のオプションはすべて無視されます。</p> <p>オプションは、一度に1つの FFB デバイスに対してのみ指定することができます。複数の FFB デバイスに対してオプションを指定するには、ffbconfig を複数回起動する必要があります。</p> <p>ffbconfig で指定できるのは、FFB 固有のオプションだけです。デフォルトの表示色数、デフォルトの画像表示形式クラスなどを指定する通常のウィンドウシステムのオプションは、openwin コマンド行のデバイス修飾子で指定してください。詳細については、<i>OpenWindows</i> デスクトップ・リファレンスマニュアルを参照してください。</p> <p>ユーザーは、更新する OWconfig ファイルを指定することもできます。デフォルトでは、/etc/openwin ディレクトリツリーにあるマシン固有のファイルが更新されます。別のファイルを指定するには、-file オプションを使用します。たとえば、/usr/openwin ディレクトリツリーにあるシステム共通の OWconfig ファイルを代わりに更新することができます。</p> <p>これらの標準 OWconfig ファイルのどちらもスーパーユーザーのみが書き込みを行います。したがって、スーパーユーザーが所有する ffbconfig プログラムは、setuid による root の権限で実行されます。</p>
オプション	<pre> -dev <i>device-filename</i> FFB 特殊ファイルを指定します。デフォルトは/dev/fbs/ffb0 です。 -file <i>machine</i> <i>system</i> 更新する OWconfig ファイルを指定します。machine を指定すると、 /etc/openwin ディレクトリツリーにあるマシン固有の OWconfig ファイルが使 </pre>

SUNWffb_config(1M)

用されます。system を指定すると、/usr/openwin ディレクトリツリーにある共通のOWconfig ファイルが使用されます。ファイルがない場合は、新たに生成されます。

`-res video-mode [now | try [noconfirm | nocheck]]`

指定した FFB デバイスに接続されているモニターを制御する際に使われる表示モードを指定します。

表示モードの形式は *widthxheightxrate* で、*width* はピクセル単位の画面幅、*height* はピクセル単位の画面の高さ、*rate* は画面を垂直方向に再描画する周期です。

960x680x112s や 960x680x108s の s 接尾辞は、これらが立体表示モードであることを意味します。640x480x60i や 768x575x50i の i 接尾辞は、インタレース表示タイミングを有効にします。この接尾辞がない場合は、ノンインタレースタイミングが使用されます。

`-res` (「形式」に記されている 3 番目の形式) にリフレッシュレートを指定する際は、値の直前に *x* の代わりに *@* を使用することができます。たとえば、1280x1024@76 のように指定することができます。

一部の表示モードは、FFB の一部のバージョンのみが対応しています。また、FFB が対応している表示モードには、モニターが対応していないものもあります。FFB デバイスとモニターの両方が対応している表示モードのリストは、`-res ? オプション付きの ffbconfig` を実行することによって得ることができます。

FFB が対応している表示モードのリストを以下に示します。

表示モード	説明
1024x768x60	
1024x768x70	
1024x768x75	
1024x768x77	
1024x800x84	
1152x900x66	
1152x900x76	
1280x800x76	
1280x1024x60	
1280x1024x67	
1280x1024x76	
960x680x112s	(立体表示)

表示モード	説明
960x680x108s	(立体表示)
640x480x60	
640x480x60i	(インタレース)
768x575x50i	(インタレース)
1440x900x76	(高解像度)
1600x1000x66	(高解像度)
1600x1000x76i	(高解像度)
1600x1280x76	(高解像度)
1920x1080x72	(高解像度)
1920x1200x70	(高解像度)

記号名

便宜上、表示モードのいくつかには記号名が定義されています。 *widthxheightxrate* の形式の代わりに、記号名を *-res* の引数として指定することができます。記号名 *none* は、ウィンドウシステムを実行すると、画面の解像度は現在デバイスにプログラムされている表示モードになることを意味します。

記号名	対応する表示モード
svga	1024x768x60
1152	1152x900x76
1280	1280x1024x76
stereo	960x680x112s
ntsc	640x480x60i
pal	768x575x50i
none	(デバイスでプログラムされている表示モード)

-res オプションには、表示モードの直後に次の追加引数を指定することができます。追加引数は、単独でも複数でも指定することができます。

now

OWconfig ファイルの表示モードを更新するとともに、FFB デバイスが指定した表示モードにただちにプログラムされます (この機能は、ウィンドウシステムを開始する前に表示モードを変更する際に便利です)。

SUNWffb_config(1M)

対象となるデバイスが稼働している間 (たとえば、ウィンドウシステムの稼働中) に、この追加オプションを `ffbconfig` に指定することはお勧めしません。予期しない結果になることもあります。 `now` オプションを指定して `ffbconfig` コマンドを実行する場合は、最初にウィンドウシステムを終了してください。 `now` オプションがウィンドウシステムのセッション中に使用された場合、表示モードはただちに変更されますが、画面の幅や高さはそのセッションが終了して次のセッションに入るまで変更されません。さらに、立体表示モードではシステムが変更を認識しないことがあります。したがって、ウィンドウシステムの稼働中には絶対に `now` オプションを指定しないでください。

`noconfirm`

確認と警告メッセージを省略し、要求された表示モードにプログラムします。

`-res` オプションを指定した際に、システムが使用できない状態になり、表示出力がなくなる場合があります。このような状況は、特定のコードが読み込まれた際のモニターセンスコードにあいまいさがあった場合などに発生します。このような事態を避けるために `ffbconfig` のデフォルトの動作では、この問題についての警告メッセージと処理を継続するかどうかを確認するメッセージを表示します。このオプションは、`ffbconfig` がシェルスクリプトから実行されている場合に便利です。

`nocheck`

モニターセンスコードに基づく通常のエラーチェックが行われません。ユーザーによって指定された表示モードは、現在接続されているモニターに適切かどうかにかかわらず受け付けられます。このオプションは、FFB デバイスに異なるモニターを接続する場合に便利です。このオプションを指定すると、`noconfirm` も指定されます。

`try`

指定した表示モードに試験的にプログラムされます。ユーザーは、指定した表示モードを使用する場合は、メッセージが表示されてから 10 秒以内に `y` と入力します。表示されたモードを使用しない場合は、10 秒以内に任意の文字を入力します。 `y` または `Return` キー以外の文字の入力は、すべて「使用しない」とみなされ、以前の表示モードに戻され、`OWconfig` ファイル中の表示モードは書き換えられません。その他の指定されたオプションは有効となります。 `Return` キーの入力があった場合は、新しい表示モードを保持するかどうかを `yes` または `no` で確認するメッセージが表示されます。このオプションを指定すると、`now` サブオプションも指定されます。(`now` サブオプションの注意を参照)。

`-deflinear true | false`

FFB には、2 種類の画像表示形式があります。リニア画像と非リニア画像です。リニア画像はガンマ補正され、非リニア画像は補正されません。リニア画像版も非リニア画像版も、ともに持つ画像表示形式が 2 つあります。24 ビット TrueColor と 8 ビット StaticGray です。

`-deflinear true` を指定すると、デフォルトの画像表示形式として、デフォルトで選択されたオプション (特に、Xsun(1)の `defdepth` および `defclass` オプション。詳細は *OpenWindows* デスクトップ・リファレンスマニュアルのマニュアルページを参照) を満たすリニア画像を設定します。具体的には、デフォルトの画像表示形式の選択オプションは、Xsun(1) の `defdepth` および `defclass` オプションによって設定されたものです。

`-deflinear false` を指定すると、または他のデフォルトで選択されたオプションを満たすリニア画像がない場合は、これらの他のオプションを満たす非リニア画像がデフォルトとして選択されます。

FFB にはリニアオーバーレイ画像表示形式がないため、オプションが存在する場合は、このオプションを使用することはできません。

`-defoverlay true | false`

FFB が、残りの FFB 画像から切り離されたピクセルを持つ 8 ビット疑似カラー画像を提供します。これを、オーバーレイ画像といいます。この画像表示形式で作成されたウィンドウは、他の画像表示形式で作成されたウィンドウに影響を与えません。逆に、他の画像表示形式で作成されたウィンドウは、オーバーレイウィンドウに影響を与えます。この画像表示形式では、256 種類の不透明カラーの値による拡張オーバーレイを使用することができます。-maxwidts を参照してください。

`-defoverlay` に `true` を指定すると、オーバーレイ画像がデフォルト画像になります。`-defoverlay` に `false` を指定すると、他のデフォルトで選択された `defdepth` および `defclass` オプションを満たすオーバーレイでない画像表示形式が、デフォルトの画像表示形式として選択されます。詳細は、*OpenWindows* デスクトップ・リファレンスマニュアルのマニュアルページを参照してください。

`-defoverlay true` を使用する場合は、`openwin` コマンド行で選択されたデフォルトの深さとクラスは常に 8 ビット疑似カラーである必要があります。それ以外の場合は、警告メッセージが出力され、`-defoverlay` オプションは `false` として扱われます。`-deflinear` オプションが存在する場合、FFB にはリニアオーバーレイ画像表示形式がないため、このオプションは使用することができません。

`-linearorder first | last`

`first` を指定すると、FFB 画面用の X11 画面画像表示形式リスト上で、リニア画像が非リニア画像より前に表示されます。`last` を指定すると、非リニア画像は、リニア画像より前に表示されます。

`-overlayorder first | last`

`first` を指定すると、FFB 画面用の X11 画面画像表示形式リスト上で、8 ビット疑似カラーオーバーレイ画像が、非オーバーレイ画像より前に表示されます。`last` を指定すると、非オーバーレイ画像は、オーバーレイ画像より前に表示されます。

`-expvis enable | disable`

`enable` を指定すると、OpenGL Visual Expansion が起動されます。選択された画像表示形式グループ (8 ビット PseudoColor、24 ビット TrueColor など) は、画面画像表示形式リストで見つけることができます。

SUNWffb_config(1M)

`-sov enable | disable`

`enable` を指定すると、ルートウィンドウの `SERVER_OVERLAY_VISUALS` 属性が有効になります。SOV 画像が転送され、それらの透過タイプ、値、階層は、この属性によって参照することができます。`disable` を指定すると、`SERVER_OVERLAY_VISUALS` 属性は定義されません。SOV 画像は転送されません。

`-maxwids n`

ウィンドウ ID (WID) として使用するために予約される最大数の FFB チャンネルピクセル値を指定します。オーバーレイカラーマップのピクセル値の残りは、通常の X11 の未使用のカラーピクセルのために使用されます。確保された WID は、(XGL などの) 3次元グラフィックスウィンドウ、MBX ウィンドウと、デフォルト以外の画像表示形式をもつウィンドウにより発生順に割り当てられます。X チャンネルコードの 0 から (255-*n*) は、未使用のカラーピクセルです。(255-*n*+1) から 255 の X チャンネルコードは、WID として使用するために予約されます。FFB と FFB2 の有効な値は、1、2、4、8、16、32 です。FFB2+ の有効な値は、1、2、4、8、16、32、64 です。

`-extovl enable | disable`

このオプションは FFB2+ のみで使用することができます。`enable` を指定すると、拡張オーバーレイを使用することができます。このオーバーレイ画像には 256 種類の不透明カラーがあります。SOV 画像には 255 種類の不透明カラーと 1 種類の透明カラーがあります。このオプションは、ハードウェアによる透明カラーを有効にするため、SOV 画像を使用するウィンドウで、より高い性能が得られます。

`-g gamma-correction value`

このオプションは FFB2+ のみで使用することができます。このオプションによって、ガンマ補正の値を変更することができます。すべてのリニア画像ではガンマ補正を使用することができます。デフォルトでは、ガンマ補正の値は 2.22 です。0 より小さい値は無効 (不正) です。ガンマ補正の値はリニア画像に適用され、リニア画像の有効ガンマ値は 1.0 になります。これは、`XSolarisGetVisualGamma(3)` によって返される値です。この機能については、`XSolarisGetVisualGamma(3)` を参照してください。

このオプションは、ウィンドウシステムが稼動しているときに使用することができます。ガンマ補正の値を変更すると、リニア画像を使用して表示されているすべてのウィンドウが影響を受けます。

`-gfile gamma-correction file`

このオプションは FFB2+ のみで使用することができます。このオプションは、指定されたファイルからガンマ補正表を読み込みます。このファイルは、各行が R、G、B チャンネルの値を持つように書式化されている必要があります。それらの値は、16 進数で指定し、値と値の間は 1 つ以上の空白文字で区切ります。このファイルでは、3 つの値の組が 256 種類定義されます。このファイルの例を以下に示します。

```
0x00 0x00 0x00
0x01 0x01 0x01
0x02 0x02 0x02
...
...
```

```
0xff 0xff 0xff
```

このオプションを使用することによって、ウィンドウシステムが稼動しているときにガンマ補正表を読み込むことができます。新しいガンマ補正は、このリニア画像によって表示されているすべてのウィンドウに影響を与えます。ユーザーが指定した表によってガンマ補正を行う際は、ガンマ補正の値は定義されません。デフォルトでは、ウィンドウシステムはガンマ補正值として2.22を使用し、このガンマ補正值に対応してウィンドウシステムが作成したガンマ補正表を読み込みます。

-defaults

すべてのオプションの値をそれぞれのデフォルト値に戻します。

-propt

-file オプションで指定された OWconfig ファイルに書かれた FFB オプションの値のうち、-dev オプションで指定されたデバイスに対するものすべてを表示します。ffbconfig の呼び出しが終了した後に、OWconfig ファイルに書き込まれるオプションの値を表示します。次に表示例を示します。

```
--- OpenWindows Configuration for /dev/fbs/ffb0 ---
OWconfig: machine
Video Mode: NONE
Default Visual: Non-Linear Normal Visual
Visual Ordering: Linear Visuals are last
                  Overlay Visuals are last
OpenGL Visuals: disabled
SOV: disabled
Allocated WIDs: 32
```

-prconf

FFB のハードウェア構成を表示します。次に表示例を示します。

```
--- Hardware Configuration for /dev/fbs/ffb0 ---
Type: double-buffered FFB2 with Z-buffer
Board: rev x
PROM Information: @(#)ffb2.fth x.x xx/xx/xx
FBC: version x
DAC: Brooktree 9068, version x
3DRAM: Mitsubishi 1309, version x
EDID Data: Available - EDID version 1 revision x
Monitor Sense ID: 4 (Sun 37x29cm RGB color monitor)
Monitor possible resolutions: 1024x768x60, 1024x768x70,
                              1024x768x75, 1152x900x66, 1152x900x76,
                              1280x1024x67, 1280x1024x76, 960x680x112s,
                              640x480x60
Current resolution setting: 1280x1024x76
```

-help

ffbconfig コマンド行のオプションと機能の概要を一覧で表示します。

デフォルト設定

ffbconfig コマンド行で指定されていないオプションについては、対応する OWconfig ファイル中のオプションは更新されず、ファイル内の値がそのまま使用されます。

SUNWffb_config(1M)

ウィンドウシステムを実行する際に、ffbconfig による FFB オプションの指定がまったくなかった場合は、デフォルト値が使用されます。オプションのデフォルト値を以下に示します。

オプション	デフォルト値
-dev	/dev/fbs/ffb0
-file	machine
-res	none
-deflinear	false
-defoverlay	false
-linearorder	last
-overlayorder	last
-expvis	enabled
-sov	enabled
-maxwids	32

-res オプションのデフォルト値 none とは、ウィンドウシステムが実行された場合に、画面解像度がそのデバイスに現在プログラムされている表示モードになることを意味しています。

注: これによって、PROM によってデバイスの解像度を指定しているユーザーとの共用性が保てます。(gx などの)一部のデバイスでは、PROM が表示モードを指定する唯一の手段です。これは、デフォルトの FFB 表示モードは、最終的に PROM によって決まることを意味しています。

使用例 例 1 モニターの種類の変更

モニターの種類を、垂直周波数 76 Hz で解像度 1280 × 1024 に変更する例を以下に示します。

```
example% /usr/sbin/ffbconfig -res 1280x1024x76
```

ファイル /dev/fbs/ffb0 デバイス特殊ファイル

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	SUNWffbcf

関連項目 mmap(2), attributes(5), fbio(7I), ffb(7D)

OpenWindows デスクトップ・リファレンスマニュアルのマニュアルページ

SUNWm64_config(1M)

名前	m64config, SUNWm64_config – M64 グラフィックスアクセラレータの設定
形式	<pre>/usr/sbin/m64config [-defaults] [-depth 8 24] [-dev device-filename] [-file machine system] [-prconf] [-propt] [-res video-mode [now try] [noconfirm nocheck]] /usr/sbin/m64config [-prconf] [-propt] /usr/sbin/m64config [-help] [-res ?]</pre>
機能説明	<p>m64config は、M64 グラフィックスアクセラレータおよび M64 対応の X11 ウィンドウシステムのデフォルトの一部を設定します。</p> <p>形式の項に記された m64config の 1 番目の形式では、指定したオプションを OWconfig ファイルに保存します。これらのオプションは、次にウィンドウシステムをそのデバイスで実行するときに M64 デバイスを初期化するために使用されます。OWconfig ファイル内のオプションの更新は、異なるウィンドウセッションや再起動したシステムでも有効となります。</p> <p>-prconf、-propt、-help、-res ? オプションだけを起動する 2 番目と 3 番目の形式では、OWconfig ファイルは更新されません。また、3 番目の形式では、その他のオプションはすべて無視されます。</p> <p>オプションは、一度に 1 つの M64 デバイスに対してのみ指定することができます。複数の M64 デバイスに対してオプションを指定するには、m64config を複数回起動する必要があります。</p> <p>m64config で指定できるのは、M64 固有のオプションだけです。デフォルトの表示色数、デフォルトの画像表示形式クラスなどを指定する通常のウィンドウシステムのオプションは、openwin コマンド行のデバイス修飾子で指定してください (OpenWindows のマニュアルページの Xsun(1) を参照)。</p> <p>ユーザーは、更新する OWconfig ファイルを指定することもできます。デフォルトでは、/etc/openwin ディレクトリツリーにあるマシン固有のファイルが更新されません。別のファイルを指定するには、-file オプションを使用します。たとえば、/usr/openwin ディレクトリツリーにあるシステム共通の OWconfig ファイルを代わりに更新することができます。</p> <p>これらの標準 OWconfig ファイルのどちらもスーパーユーザーのみが書き込みを行います。したがって、スーパーユーザーが所有する m64config プログラムは、setuid による root の権限で実行されます。</p>
オプション	<p>-defaults すべてのオプションの値をそれぞれのデフォルト値に戻します。</p> <p>-depth 8 24 表示色数を 8 または 24 ビット/ピクセルで指定します。変更を反映させるには、現在のウィンドウシステムのセッションからログアウトしてから再度ログインします。24 ビット/ピクセルでは、画面解像度を低くすることにより、ウィンドウシステムで TrueColor グラフィックスを実現できます。</p>

SUNWm64_config(1M)

24 ビット/ピクセルで利用できる解像度の最大値は、PGX カードに搭載されているメモリー量に依存します。2 M バイトのメモリーを搭載した PGX カードで利用できる解像度の最大値は 800x600 です。4 M バイトのメモリーを搭載したカードで利用できる解像度の最大値は 1152x900 です。指定した解像度と色数の組み合わせに必要なメモリーが不足している場合、m64config はエラーメッセージを出力して終了します。

-dev *device-filename*

M64 特殊ファイルを指定します。デフォルトは /dev/fbs/m640 です。

-file *machine|system*

更新する OWconfig ファイルを指定します。machine を指定すると、/etc/openwin ディレクトリツリーにあるマシン固有の OWconfig ファイルが使用されます。system を指定すると、/usr/openwin ディレクトリツリーにある共通の OWconfig ファイルが使用されます。ファイルがない場合は、新たに生成されます。

-help

m64config コマンド行のオプションと機能の概要を一覧で表示します。

-prconf

M64 ハードウェア構成を表示します。次に表示例を示します。

```
--- Hardware Configuration for /dev/fbs/m640 ---
ASIC: version 0x41004754
DAC: version 0x0
PROM: version 0x0
Card possible resolutions:  640x480x60, 800x600x75, 1024x768x60
                            1024x768x70, 1024x768x75, 1280x1024x75, 1280x1024x76
                            1280x1024x60, 1152x900x66, 1152x900x76, 1280x1024x67
                            960x680x112S, 960x680x108S, 640x480x60i, 768x575x50i, 1280x800x76
                            1440x900x76, 1600x1000x66, 1600x1000x76, vga, svga, 1152, 1280
                            stereo, ntsc, pal
Monitor possible resolutions: 720x400x70, 720x400x88, 640x480x60
                              640x480x67, 640x480x72, 640x480x75, 800x600x56, 800x600x60
                              800x600x72, 800x600x75, 832x624x75, 1024x768x87, 1024x768x60
                              1024x768x70, 1024x768x75, 1280x1024x75, 1280x1024x76, 1152x900x66
                              1152x900x76, 1280x1024x67, 960x680x112S, vga, svga, 1152, 1280
                              stereo
Possible depths: 8, 24
Current resolution setting: 1280x1024x76
Current depth: 8
```

-propt

-file オプションで指定された OWconfig ファイルに書かれた M64 オプションの値のうち、-dev オプションで指定されたデバイスに対するものすべてを表示します。m64config の呼び出しが終了した後に、OWconfig ファイルに書き込まれるオプションの値を表示します。次に表示例を示します。

```
--- OpenWindows Configuration for /dev/fbs/m640 ---
OWconfig: machine
Video Mode: not set
```

SUNWm64_config(1M)

`-res video-mode [now | try [noconfirm | nocheck]]`

指定した M64 デバイスに接続されているモニターを制御する際に使われる表示モードを指定します。表示モードはあらかじめ組み込まれています。表示モードの形式は `widthx-heightxrate` で、`width` はピクセル単位の画面幅、`height` はピクセル単位の画面の高さ、`rate` は画面を垂直方向に再描画する周期です。便宜上、`-res` にリフレッシュレートを指定する際は、値の直前に `x` の代わりに `@` を使用することができます。たとえば、`1280x1024@76` のように指定することができます。

有効な表示モードのリストは、`m64config -res '??'` を実行することによって得ることができます。`?` は引用符で囲んで文字として扱う必要があります。すべてのビデオボードおよびモニターがすべての解像度に対応しているわけではありません。m64config は、ボードが対応していない解像度を設定することを許しません。モニターが対応していない解像度を設定しようとすると、確認を促すメッセージを表示します。

記号名

便宜上、上記の表示モードのいくつかには記号名が定義されています。`widthxheightxrate` の形式の代わりに、記号名を `-res` の引数として指定することができます。記号名 `none` は、ウィンドウシステムを実行すると、画面の解像度は現在デバイスにプログラムされている表示モードになることを意味します。

記号名	対応する表示モード
<code>svga</code>	1024x768x60
<code>1152</code>	1152x900x76
<code>1280</code>	1280x1024x76
<code>none</code>	(現在デバイスにプログラムされている表示モード)

`-res` オプションには、表示モードの直後に次の追加引数を指定することができます。追加引数は、単独でも複数でも指定することができます。

nocheck

このオプションを指定すると、モニターセンスコードに基づく通常のエラーチェックが行われません。ユーザーによって指定された表示モードは、現在接続されているモニターに適切かどうかにかかわらず受け付けられます。このオプションは、M64 デバイスに異なるモニターを接続する場合に便利です。このオプションを指定すると、`noconfirm` も指定されます。

noconfirm

`-res` オプションを指定した際に、システムが使用できない状態になり、表示出力がなくなる場合があります。このような状況は、特定のコードが読み込まれた際のモニターセンスコードにあいまいさがあった場合などに発生します。このような事態を避けるために m64config のデフォルトの動作では、この問題についての警告メッセージと、処理を継続するかどうかを確認するメッセージを表示します。`noconfirm` オプションを指定すると、m64config コマンドはこの確認をせずに、

要求のあった表示モードにプログラムします。このオプションは、m64config がシェルスクリプトから実行されている場合に便利です。

now

OWconfig ファイルの表示モードを更新するとともに、M64 デバイスが指定した表示モードにただちにプログラムされます (この機能は、ウィンドウシステムを開始する前に表示モードを変更する際に便利です)。

対象となるデバイスが稼働している間 (たとえば、ウィンドウシステムの稼働中) に、この追加オプションを m64config に指定することはお勧めしません。予期しない結果になることもあります。now オプションを指定して m64config コマンドを実行する場合は、最初にウィンドウシステムを終了してください。now オプションがウィンドウシステムのセッション中に使用された場合、表示モードはただちに変更されますが、画面の幅や高さはそのセッションが終了して次のセッションに入るまで変更されません。さらに、立体表示モードではシステムが変更を認識しないことがあります。したがって、ウィンドウシステムの稼働中には絶対に now オプションを指定しないでください。

try

このオプションを指定すると、指定した表示モードが試験的にプログラムされます。ユーザーは、指定した表示モードを使用する場合は、メッセージが表示されてから 10 秒以内に **y** と入力します。表示されたモードを使用しない場合は、10 秒以内に任意の文字を入力します。y または Return キー以外の文字の入力は、すべて「使用しない」とみなされ、以前の表示モードに戻され、OWconfig ファイル中の表示モードは書き換えられません (その他の指定されたオプションは有効となります)。Return キーの入力があつた場合は、新しい表示モードを保持するかどうかを **yes** または **no** で確認するメッセージが表示されます。このオプションを指定すると、now サブオプションも指定されます。(now サブオプションの注意を参照)。

デフォルト設定

m64config コマンド行で指定されていないオプションについては、対応する OWconfig ファイル中のオプションは更新されず、ファイル内の値がそのまま使用されます。

ウィンドウシステムを実行する際に、m64config による M64 オプションの指定がまったくなかった場合は、デフォルト値が使用されます。オプションのデフォルトを以下に示します。

オプション	デフォルト値
-dev	/dev/fbs/m640
-file	machine
-res	none

-res オプションのデフォルト値 **none** とは、ウィンドウシステムが実行された場合に、画面解像度がそのデバイスに現在プログラムされている表示モードになることを意味しています。

SUNWm64_config(1M)

これによって、PROM によってデバイスの解像度を指定しているユーザーとの共用性が保てます。(GX などの)一部のデバイスでは、PROM が表示モードを指定する唯一の手段です。これは、デフォルトの M64 表示モードは、最終的に PROM によって決まることを意味しています。

使用例 例 1 モニターの種類の変更

モニターの種類を、垂直周波数 76 Hz で解像度 1280x1024 に変更する例を以下に示します。

```
example% /usr/sbin/m64config -res 1280x1024x76
```

ファイル

`/dev/fbs/m640`
デバイス特殊ファイル

`/usr/openwin/server/etc/OWconfig`
システム設定ファイル

`/usr/lib/fbconfig/SUNWm64_config`
`usr/sbin/m64config` へのシンボリックリンク

属性 以下の属性については、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	SUNWm64cf

関連項目

`attributes(5)`, `m64(7D)`

『OpenWindows デスクトップ・リファレンスマニュアル』

名前	sys-unconfig – システム構成の解除
形式	/usr/sbin/sys-unconfig
機能説明	<p>sys-unconfig コマンドは、システムの構成を「出荷時設定」に戻し、再び構成可能な状態にします。システムの構成には、ホスト名、NIS (Network Information Service) ドメイン名、タイムゾーン、IP アドレス、IP サブネットマスク、および root パスワードが含まれます。これは、ブート時に sysidnet(1M)、sysidns(1M)、および sysidsys(1M) プログラムによって実行されるのと逆の処理です。sysidtool(1M) のマニュアルページを参照してください。</p> <p>sys-unconfig の処理内容は次のとおりです。</p> <ul style="list-style-type: none"> ■ 現在の /etc/inet/hosts ファイル情報を /etc/inet/hosts.saved に保存する ■ 現在の /etc/vfstab ファイルに NFS マウントのエントリが含まれている場合は、/etc/vfstab.orig に /etc/vfstab ファイルを保存する ■ デフォルトの /etc/inet/hosts ファイルを復元する ■ 実行時に構成されたすべてのインタフェースについて、/etc/hostname.interface ファイル内のデフォルトのホスト名を削除する。どのインタフェースが構成されたかを確認するには、コマンド <i>ifconfig-a</i> を実行します。生成される出力に示されているすべてのインタフェースに対応する /etc/hostname.interface ファイル (ループバックインタフェース (lo0) を除く) が削除されます。 ■ /etc/defaultdomain 内のデフォルトのドメイン名を削除する ■ /etc/TIMEZONE 内のタイムゾーンを PST8PDT に戻す ■ NIS (Network Information Service) または NIS+ (Network Information Service Plus) が構成されている場合、これらを無効にする ■ /etc/net/*/hosts からこのホストのエントリを削除する ■ ファイル /etc/inet/netmasks を削除する ■ ファイル /etc/defaultrouter を削除する ■ スーパーユーザー用に設定されているパスワードを /etc/shadow から削除する ■ ファイル /etc/.rootkey を削除する ■ すべてのシステム構成アプリケーションを実行する。これらのアプリケーションは、あらかじめ sysidconfig -a <i>application</i> を実行して定義されています (sysidconfig(1M) のマニュアルページを参照)。sys-unconfig の実行時に、すべてのシステム構成アプリケーションに 1 つの引数 -u が渡されます。 ■ ファイル /etc/resolv.conf を削除する ■ /var/ldap/ldap_client_cache、/var/ldap/ldap_client_file、/var/ldap/ldap_client_cred、および /var/ldap/cachemgr.log を削除することにより LDAP を無効にする <p>sys-unconfig は、その終了時にシステムを停止します。sys-unconfig は潜在的に危険なユーティリティであるため、スーパーユーザーだけが実行できます。</p>

sys-unconfig(1M)

ファイル	/etc/default/init	プロセス制御の初期化
	/etc/defaultdomain	
	/etc/defaultrouter	
	/etc/hostname. <i>interface</i>	
	/etc/inet/hosts	ホスト名データベース
	/etc/inet/netmasks	ネットワークマスクデータベース
	/etc/net/*/hosts	
	/etc/nodename	
	/etc/.rootkey	スーパーユーザーの秘密鍵
	/etc/shadow	シャドウパスワードファイル
	/etc/vfstab	仮想ファイルシステムテーブル
	/var/nis/NIS_COLD_START	
	/var/yp/binding/*/ypservers	

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWadmap

関連項目 `init(1M)`, `kdmconfig(1M)`, `sysidconfig(1M)`, `sysidtool(1M)`, `hosts(4)`, `netmasks(4)`, `shadow(4)`, `attributes(5)`

注意事項 `sys-unconfig` は、ディスクレスクライアントでは使用できません。

名前	init, telinit – プロセス制御の初期化	
形式	/sbin/init [0123456abcQqSs] /etc/telinit [0123456abcQqSs]	
機能説明	init は汎用のプロセス生成コマンドです。その主な役割は、ファイル /etc/inittab に格納されている情報に基づいてプロセスを生成することです。	
定義されている実行レベル	任意の時点で、システムは 8 つの実行レベルのいずれか 1 つにあります。実行レベルはソフトウェア構成で、ここでは選択されたプロセスグループだけが存在します。各実行レベルに対する、init によって生成されるプロセスは、/etc/inittab で定義されています。init は 8 つの実行レベル、つまり 0-6 および s または S (S と s は同じ) のいずれか 1 つを取ることができます。実行レベルは、特権ユーザーが /sbin/init を実行すると変更されます。/sbin/init は、起動時にオペレーティングシステムによって生成された元の init に適切なシグナルを送り、移行する実行レベルを通知します。	
init とシステムの起動	システムが起動されると、init が呼び出され、次のイベントが発生します。最初に init は、/etc/default/init を読み込み、環境変数を設定します。通常はここで、TZ (タイムゾーン) およびロケール関連の環境、すなわち LANG、LC_CTYPE など が設定されます (このページの最後にある「ファイル」の項を参照)。次に init は、/etc/inittab を調べ、initdefault エントリを探します (inittab(4) のマニュアルページを参照)。initdefault のエントリ	
	存在する場合	init は通常、このエントリで指定された実行レベルを、開始時の最初の実行レベルとして使用します。
	存在しない場合	init はユーザーに、システムコンソールから実行レベルを入力するように要求します。
	s または S	init はシングルユーザー状態に移行します。この状態では、システムコンソールデバイス (/dev/console) が読み書き用にオープンされ、コマンド /sbin/su (su(1M) のマニュアルページを参照) が呼び出されます。init または telinit のどちらか一方を使用して、システムの実行レベルを変更します。(ファイルの終わり (EOF) を使用して) シェルを終了させたが、/etc/inittab が存在しない場合、init は再びシングルユーザー状態に戻すだけです。
	0-6	init は対応する実行レベルに移行します。実行レベル 0、5、および 6 はシステム停止用に予約されています。実行レベル 2、3、および 4 は、マルチユーザー操作用に使用で

telinit(1M)

きます。

電源投入後に、init が最初のシングルユーザー状態以外の実行レベルに移行する場合、init はまず、/etc/inittab を走査して boot および bootwait エントリを探します (inittab(4) のマニュアルページを参照)。移行する実行レベルがエントリの実行レベルと一致している場合は、/etc/inittab の他のプロセスが実行される前に、これらのエントリが実行されます。このようにすると、ファイルシステムのマウントなど、オペレーティングシステムの特異な初期化を先に終えてから、ユーザーにシステムを使用させることができます。次に init は、/etc/inittab を走査し、その実行レベルで処理すべき他のすべてのエントリを実行します。

/etc/inittab の各プロセスを生成するために、init は各エントリを読み込み、再生成が必要なエントリごとに、子プロセスを生成します。/etc/inittab で指定された全プロセスを生成すると、init は、以下の状態のために待機します。子孫プロセスの1つが消滅する、powerfail シグナルを受けると、システムの実行レベルの変更を要求する他の init または telinit プロセスから送られるシグナルを受けると。これらの状態のいずれか1つが発生すると、init は /etc/inittab を再検査します。

inittab に関するその他の事項

/etc/inittab にはいつでも新しいエントリを追加できますが、init は上記の3つの状態のいずれかが発生するまで待機し続けるので、/etc/inittab を再検証しません。この状況を回避するには、init Q または init q コマンドを使用して、init が /etc/inittab をただちに再検査するようにします。

起動時に init が呼び出されたとき、また、システムがシングルユーザー状態から別の実行状態に移行するたびに、init はコンソールの ioct1(2) 状態をファイル /etc/ioct1.syscon に格納されているモードに設定します。init はシングルユーザー状態に移行するたびに、このファイルに書き込みを行います。

実行レベルの変更

実行レベルの変更が要求されると、init は目標の実行レベルで定義されていないすべてのプロセスに警告シグナル (SIGTERM) を送ります。init は 5 秒間待機したのち、終了シグナル (SIGKILL) を送信することによって、これらのプロセスを強制終了させます。

init によって生成されたプロセスが終了したことを通知するシグナルを受信すると、init は、その事実と原因を /var/adm/utmpx と /var/adm/wtmpx (存在していれば) に記録します (who(1) を参照)。生成されたプロセスの履歴は /var/adm/wtmpx に記録されます。

init は powerfail シグナル (SIGPWR) を受信した場合、/etc/inittab を走査し、タイプが powerfail と powerwait の特異なエントリを探します。これらのエントリが呼び出されてから (実行レベルが許可する場合)、その後の処理が実行されます。このように、init はオペレーティングシステムの停止中にさまざまなクリーンアップおよび記録機能を実行します。

/etc/defaults/init の環境変数

タイムゾーンや文字の書式といった環境変数のデフォルト値を /etc/default/init で設定できます。環境変数の一覧については、「ファイル」の項を参照してください。

telinit	<code>/sbin/init</code> にリンクされている <code>telinit</code> は、 <code>init</code> のアクションを指示する場合に使用します。 <code>telinit</code> は、1 文字の引数を取り、適切なアクションを実行するように、 <code>init</code> にシグナルを送ります。
セキュリティ	<p><code>init</code> は <code>pam(3PAM)</code> を使用してセッションを管理します。<code>/etc/pam.conf</code> に記述されている PAM 構成ポリシーにより、<code>init</code> に使用されるセッション管理モジュールが指定されます。以下に、<code>pam.conf</code> ファイルの抜粋を示します。UNIX セッション管理モジュールを使用する <code>init</code> のエントリが指定されています。</p> <pre>init session required pam_unix_session.so.1</pre> <p><code>init</code> サービスに対応するエントリがない場合、「other」のサービスのエントリが使用されます。</p>
オプション	<p>0 ファームウェアモードに移行します。</p> <p>1 システムをシステム管理者モードにします。すべてのローカルファイルシステムがマウントされます。いくつかの重要なコアプロセスだけが実行を続けます。このモードはオプションのユーティリティパッケージをインストールする場合など、管理作業を行うためのものです。すべてのファイルにアクセスできます。ユーザーはシステムにログインできません。</p> <p>2 システムをマルチユーザーモードにします。すべてのマルチユーザー環境用の端末プロセスおよびデーモンが生成されます。一般に、この状態を、マルチユーザー状態といいます。</p> <p>3 ネットワークを介してローカル資源を使用できるようにすることで、マルチユーザーモードを拡張します。</p> <p>4 代替のマルチユーザー環境構成として定義できます。システム運用上は必要なく、通常は使用されません。</p> <p>5 電源を切断しても問題が起きないように、マシンを停止させます。可能であれば、マシンの電源を自動的に切断させます。</p> <p>6 オペレーティングシステムを停止し、<code>/etc/inittab</code> の <code>initdefault</code> エントリで定義された状態で再起動します。</p> <p>a, b, c <code>/etc/inittab</code> エントリに a、b、または c という実行レベルが設定されているときだけ処理します。これらは擬似状態であり、特定のコマンドを実行するように定義できますが、現在の実行レベルが変更されるわけではありません。</p> <p>Q, q <code>/etc/inittab</code> を再検査します。</p> <p>S, s シングルユーザーモードに移行します。適切な形式の <code>/etc/inittab</code> ファイルがなくともかまわない唯一の実行レベルです。このファイルが存在しない場合、デフォルトで <code>init</code> が入ることができる唯一の正当な実行レベルはシングルユーザーモードだけです。シングルユーザーモードでは、基本システム動作に必要なファイルシステムがマウントされます。システムがシングルユーザーモードに移行すると、これらのファイルシステムは (リ</p>

telinit(1M)

モートファイルサーバーが提供している場合でも) マウントされたままになります。他のローカルファイルシステムもマウントされたままになります。シングルユーザーモードへの切り替え時に、init または init.d によって開始された、マルチユーザーモードでしか実行してはならないプロセスはすべて強制終了されます。さらに、utmpx エントリが設定されているプロセスもすべて強制終了されます。この最後の状態では、SAC が起動したすべてのポートモニターが確実に強制終了され、これらのポートモニターによって起動されたサービスも、ttymon login ログインサービスを含めてすべて強制終了されます。

ファイル	/dev/console	システムコンソール装置
	/etc/default/init	環境変数とそのデフォルト値を指定します。たとえば、タイムゾーン変数 TZ の場合、TZ=US/Pacific のように指定できます。変数は次のとおりです。
	TZ	タイムゾーン情報 (ctime(3C) のマニュアルページを参照) またはタイムゾーン情報ファイル /usr/share/lib/zoneinfo の名前のどちらか一方を指定します。
	CMASK	init が使用し、すべてのプロセスが init プロセスから継承するマスク (umask(1) のマニュアルページを参照)。設定されていない場合、init はカーネルから継承したマスクを使用します。CMASK の設定に関係なく、init は必ず、022 で umask の適用を試みてからファイルを作成します。
	LC_CTYPE	文字の種類の情報
	LC_MESSAGES	翻訳メッセージ
	LC_MONETARY	通貨の書式情報
	LC_NUMERIC	数値の書式情報
	LC_TIME	時刻の書式情報
	LC_ALL	設定されている場合、他のすべての LC_* 環境変数でこの値が使用されます。
	LANG	LC_ALL が設定されてなく、かつ特定の LC_* も設定されていない場合は、これら環境変数に LANG の値が使用されます。
	/etc/initpipe	内部通信用の名前付きパイプ

/etc/inittab	init によるプロセスディスパッチ制御
/etc/ioctl.syscon	シングルユーザー状態に移行したときに init によって保存された、コンソールの ioctl (入出力制御) 状態
/var/adm/utmpx	ユーザーアクセスおよび管理情報
/var/adm/wtmpx	ユーザーアクセスおよび管理情報の履歴

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 login(1), sh(1), stty(1), who(1), shutdown(1M), su(1M), ttymon(1M), ioctl(2), kill(2), ctime(3C), pam(3PAM), inittab(4), pam.conf(4), utmpx(4), attributes(5), pam_authtok_check(5), pam_authtok_get(5), pam_authtok_store(5), pam_dhkeys(5), pam_passwd_auth(5), pam_unix(5), pam_unix_account(5), pam_unix_auth(5), pam_unix_session(5), termio(7I)

診断 2 分間に 10 回以上、/etc/inittab 内の 1 つのエントリが再起動されていることを検出すると、init はそのエントリのコマンド文字列に誤りがあるとみなして、システムコンソール上にエラーメッセージを表示します。さらに 5 分経過するか、あるいはユーザーが生成した init または telinit からシグナルを受信するまで、init はそのエントリを再起動することを拒否します。このようにすることで、inittab ファイルに入力ミスがあった場合や /etc/inittab で参照されているプログラムが削除された場合でも、init がシステム資源を使い果たすのを防止できます。

注意事項 init および telinit を実行できるのは、特権ユーザーだけです。

/etc/inittab で、s または S 状態をむやみに使用してはなりません。このファイルを変更するときに注意すべきことは、この状態を initdefault 以外の行に追加しないことです。

/etc/inittab の initdefault エントリでデフォルトの状態が指定されていない場合は、状態 6 になります。その結果、システムはファームウェアへのアクセスと再起動を繰り返すループに陥ります。

システムの起動時に utmpx ファイルを作成できない場合、システムは /etc/inittab の initdefault エントリで指定されている状態に関係なく、“s” の状態で起動します。/var ファイルシステムにアクセスできない場合には、この状況が発生することがあります。

システムが s または S 状態へ移行するときに、/etc/nologin ファイル (nologin(4) のマニュアルページを参照) が作成されます。その後、実行レベル 2 に移行すると、/etc/rc2.d ディレクトリにあるスクリプトによってこのファイルが削除されます。

telinit(1M)

init は内部通信に名前付きパイプ /etc/initpipe を使用します。

pam_unix(5) モジュールは、将来のリリースではサポートされなくなる可能性があります。同様の機能は、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)、および pam_unix_session(5) で提供されています。

名前	pgxconfig, GFXconfig, TSIgfxp_config – PGX32 (Raptor GFX) グラフィックスアクセラレータの設定
形式	<pre> /usr/sbin/pgxconfig [-dev device-filename] [-res video-mode [try noconfirm nocheck]] [-file machine system] [-depth 8 24] [-24only true false] [-cachedpixmap true false] [-defaults] /usr/sbin/pgxconfig [-propt] [-prconf] /usr/sbin/pgxconfig [-help] [-res ?] /usr/sbin/pgxconfig [-i] </pre>
機能説明	<p>pgxconfig ユーティリティーは PGX32 (Raptor GFX) グラフィックスアクセラレータの設定用コマンドで、これには X11 ウィンドウシステムの一部を PGX32 (Raptor GFX) 用にデフォルト設定する機能も含まれます。以前のバージョンでは、このユーティリティーは GFXconfig という名称でした。</p> <p>形式の項に記された pgxconfig の第 1 の形式では、指定したオプションを OWconfig ファイルに保存します。次回の PGX32 (Raptor GFX) デバイス上でのウィンドウシステム起動時に、ここで指定したオプションに従って PGX32 (Raptor GFX) デバイスを初期化します。OWconfig ファイルに保存されたオプションの更新内容は、異なる複数のウィンドウシステムセッションや再起動後のシステムでも有効となります。</p> <p>その他の形式 (2、3、4 番目の形式) を使用した場合はオプション <code>-prconf</code>、<code>-propt</code>、<code>-help</code>、<code>-res ?</code> を呼び出すだけで、OWconfig ファイルに保存されているオプションを更新することはありません。更に、3 番目の形式を使用した場合はこれ以外のオプションはすべて無視されます。</p> <p><code>-i</code> オプションを使用すると、pgxconfig は、対話型モードで起動します。</p> <p>一度にオプションを指定できる PGX32 (Raptor GFX) デバイスの数は、1 つだけです。</p> <p>pgxconfig コマンドで指定できるのは、PGX32 (Raptor GFX) 固有のオプションだけです。デフォルトの表示色数、デフォルトのビジュアルクラスなどを指定する通常のウィンドウシステムのオプションは、openwin コマンド行のデバイス修飾子で指定してください。SUNWxwman パッケージに含まれている Xsun(1) マニュアルページを参照してください。</p> <p>ユーザーは、更新する OWconfig ファイルを指定することもできます。デフォルトでは /usr/openwin ディレクトリツリーにあるマシン固有のファイルが更新されます。別のファイルを指定するには、<code>-file</code> オプションを使用します。たとえば、/etc/openwin ディレクトリツリーにあるシステム共通の OWconfig ファイルを代わりに更新することができます。</p> <p>いずれの標準 OWconfig ファイルでも、書き込み権限があるのはスーパーユーザーだけです。</p>
オプション	次のオプションを指定できます。

TSIgfxp_config(1M)

```
-cachedpixmap true | false
false に設定した場合、PGX32 (Raptor GFX) デバイスは、OpenWindows を実行
するときだけに 24 ビットを使用します。デフォルト値は true です。

アプリケーションの中には、ディスプレイデバイスに書き込むときにキャッシュさ
れたピクスマップを使用するものがあります。このような手法を使用すると、出
力が歪曲され、X サーバーがクラッシュする可能性があります。このような問題が
発生した場合は、-cachedpixmap オプションを false に設定してください。

-defaults
すべてのオプションの値をそれぞれのデフォルト値に戻します。

-depth 8 | 24
スクリーンデプスをピクセル当り 8 または 24 ビットに設定します。24 ビット/ピ
クセルに設定すると、ウィンドウシステムで TrueColor グラフィックスを使用でき
ます。

-dev device-filename
PGX32 (Raptor GFX) 特殊ファイルを指定します。デフォルトは
/dev/fbs/gfxp0、または使用可能な場合であれば /dev/fbs/raptor0 です。

-file machine | system
更新する OWconfig ファイルを指定します。ここで machine が指定された場合
は、/etc/openwin ディレクトリツリーにあるマシン固有の OWconfig ファイル
が使用されます。system が指定された場合は、/usr/openwin ディレクトリツ
リーにある共通の OWconfig ファイルが使用されます。ファイルがない場合は、新
たに生成されます。

-help
pgxconfig コマンド行のオプションと機能の概要を一覧で表示します。

-i
pgxconfig コマンドを対話型モードで起動します。

-prconf
PGX32 (Raptor GFX) のハードウェア構成を表示します。以下に表示例を示しま
す。

--- Hardware Configuration for /dev/fbs/gfxp0 ---
DAC: version 0x0
Type:
Board:
PROM: version 0x0
PROM Information:
RAM:
EDID Data:
Monitor Sense ID:
Card possible resolutions: 640x480x60, 800x600x75, 1024x768x60
1024x768x70, 1024x768x75, 1280x1024x75, 1280x1024x76
1280x1024x60, 1152x900x66, 1152x900x76, 1280x1024x67
960x680x112S, 960x680x108S, 640x480x60i, 768x575x50i,
1280x800x76, 1440x900x76, 1600x1000x66, 1600x1000x76,
vga, svga, 1152, 1280, stereo, ntsc, pal
Monitor possible resolutions: 720x400x70, 720x400x88, 640x480x60
640x480x67, 640x480x72, 640x480x75, 800x600x56,
```



```
800x600x60, 800x600x72, 800x600x75, 832x624x75,
1024x768x87, 1024x768x60, 1024x768x70, 1024x768x75,
1280x1024x75, 1280x1024x76, 1152x900x66, 1152x900x76,
1280x1024x67, 960x680x112S, vga, svga, 1152, 1280
stereo
```

```
Current resolution setting: 1280x1024x76
```

```
Possible depths:
```

```
Current depth: 8
```

-propt

-file オプションで指定された OWconfig ファイルに書かれた PGX32 (Raptor GFX) オプションの値のうち、-dev オプションで指定されたデバイスに対するものすべてを表示します。pgxconfig の呼び出しが終了した後に、OWconfig ファイルに書き込まれるオプションの値を表示します。以下に表示例を示します。

```
--- OpenWindows Configuration for /dev/fbs/gfxp0 ---
```

```
OWconfig: machine
```

```
Video Mode: not set
```

```
Depth: 8+24
```

-res *video-mode* [try | noconfirm | nocheck]

PGX32 (Raptor GFX) デバイスに接続されているモニターの制御に使用する組み込みのビデオモードを指定します。

video-mode には、以下のいずれかの書式で指定できます。

widthxheightxrate

width はピクセル単位の画面の幅、*height* はピクセル単位の画面の高さ、*rate* は垂直方向の画面再描画周期です。-res では、再描画周期 *rate* の前の *x* は、@ でも代用できます。つまり周期の指定に限っては、たとえば 1280x1024@76 のような形式にも対応しています。オプションに -res ? を付けて pgxconfig を実行すると (コマンド形式の項に記された 3 番目の形式)、ビデオモードの一覧が表示されます。ビデオボードとモニターの両方が、すべての解像度をサポートしているわけではありません。noconfirm または nocheck オプションを指定しないで、ボードのサポートしていない解像度を入力した場合には、pgxconfig はその要求を許可しません。また、nocheck オプションを指定しないで、モニターのサポートしていない解像度を指定した場合には、その値を適用する前に確認を求めてきます。

Symbolic names

便宜上、一部のビデオモードには記号名が定義されています。*widthxheightxrate* の形式の代わりに、記号名を -res の引数として指定することができます。記号名 none は、ウィンドウシステムを実行すると、画面の解像度が現在デバイスにプログラムされているビデオモードになることを意味します。

```
svga      1024x768x60
```

TSIgfxp_config(1M)

1152	1152x900x76
1280	1280x1024x76
vga	640x480x60
none	デフォルトのコンソールの解像度

-res オプションには、ビデオモードの直後に次の追加引数を指定することができます。追加引数は、単独でも複数でも指定できます。

noconfirm	-res オプションを指定した際に、システムが使用可能であっても、表示出力のない状態になる場合があります。このような状況は、特定のコードが読み込まれた際のモニターセンスコードにあいまいさがあった場合などに発生します。このような事態を避けるために pgxconfig のデフォルトの動作では、この問題についての警告メッセージと、処理を継続するかどうかを確認するメッセージを表示します。noconfirm オプションを指定すると、pgxconfig コマンドはこの確認をせずに、要求のあったビデオモードにプログラムします。このオプションは、pgxconfig がシェルスクリプトから実行されている場合に便利です。
nocheck	このオプションを指定すると、モニターセンスコードに基づく通常のエラーチェックが行われません。ユーザーによって指定されたビデオモードは、現在接続されているモニターに適切かどうかにかかわらず受け付けられます。このオプションは、PGX32 (Raptor GFX) デバイスに異なるモニターを接続する場合に便利です。このオプションの指定は、noconfirm の指定も兼ねます。
try	このオプションを指定すると、指定したビデオモードを適用する前にテストすることができます。まず、指定したモードに基づいたテストパターンが表示されます。テストパターンが正常に表示された場合は、「y」(次いでキャリッジリターン)を入力します。「y」以外の文字を入力(次いでキャリッジリターンを入力)した場合は、「no」となります。

-res ?

PGX32 およびモニターがサポートする解像度の一覧を表示します。

-24only

Openwindows の実行時に、PGX32 (Raptor GFX) デバイスが 24 ビットカラーのみを使用するようにします。

デフォルト

pgxconfig コマンド行で指定されていないオプションについては、対応する OWconfig ファイル中のオプションは更新されず、ファイル内の値がそのまま使用されます。ただし、-depth と -24only については、その限りではありません。

ウィンドウシステムを実行する際に、pgxconfig による PGX32 (Raptor GFX) のオプションの指定がまったくなかった場合は、デフォルト値が使用されます。オプションのデフォルト値は次のとおりです。

```
-dev          /dev/fbs/gfxp0
-file         system
-res         none
```

-res オプションのデフォルト値 **none** とは、ウィンドウシステムが実行された場合に、画面解像度がそのデバイスに現在プログラムされているビデオモードになることを意味しています。

使用例 例 1 モニターの種類を変更する。

モニターの種類を、垂直周波数 76 Hz で解像度 1280 x 1024 に変更する例を以下に示します。

```
example# /usr/sbin/pgxconfig -res 1280x1024x76
```

ファイル /dev/fbs/gfxp0
デバイス特殊ファイル

/usr/openwin/server/etc/OWconfig
システム構成ファイル

/etc/openwin/server/etc/OWconfig
マシン構成ファイル

関連項目 PGX32 PCI グラフィックカード インストールマニュアル

ttymon(1M)

名前	ttymon – 端末ポートのポートモニター
形式	<pre>/usr/lib/saf/ttymon /usr/lib/saf/ttymon -g [-d device] [-h] [-t timeout] [-l ttylabel] [-p prompt] [-m modules] [-T termtype]</pre>
機能説明	<p>ttymon は STREAMS ベースの TTY ポートモニターです。その機能には、ポートの監視、ポートの端末モード、ボーレートおよび回線規約の設定、そしてポートに関連したサービスへのユーザーまたはアプリケーションの接続があります。通常、ttymon は、サービス・アクセス・ファシリティ (SAF) の一部として、サービス・アクセス・コントローラ、sac(1M) の制御下で実行するように構成されています。ttymon は sacadm(1M) コマンドを実行して構成されます。ttymon の各インスタンスは複数のポートを監視できます。ttymon のインスタンスによって監視されるポートは、ポートモニターの管理ファイルに指定されています。管理ファイルは pmadm(1M) コマンドおよび ttyadm(1M) コマンドを実行して構成されます。ttymon のインスタンスは、sac コマンドによって呼び出されると、ポートの監視を始めます。各ポートについて、ttymon は回線規約が指定されていれば、最初に回線規約を初期化してから、速度と端末設定を初期化します。/etc/logindevperm 中のエントリに指定されているポートには、デバイスのオーナー、グループ、アクセス権が設定されます (logindevperm(4) 参照)。初期化に使用される値は、TTY 設定ファイルの適切なエントリから取り込まれます。このファイルは sttydefs(1M) コマンドによって管理されています。通常、ポート上のデフォルトの回線規約は、オートプッシュファシリティの autopush(1M) コマンドによって設定されます。</p> <p>次に ttymon はプロンプトを書き込んでユーザーの入力を待ちます。ユーザーが BREAK キーを押して速度が適切でないことを知らせると、ttymon は次の速度を試行してプロンプトを再び書き込みます。正しい入力を受信すると、ttymon は、(もしあれば) ポートのサービスごとの構成ファイルを解釈し、必要に応じて utmpx エントリを作成します (utmpx(4) 参照)。そして、サービス環境を確立し、ポートに関連するサービスを呼び出します。有効な入力は、キャリッジリターンで終わり、少なくとも 1 つの復帰改行以外の文字が入った文字列からなります。サービスが終了すると、ttymon は utmpx エントリが存在していればそのエントリを消去して、ポートを初期状態に戻します。</p> <p><i>autobaud</i> がポートで動作可能であれば、ttymon は自動的にポートのボーレートを判別しようとします。ユーザーがキャリッジリターンを入力してからでないと、ttymon はボーレートを認識してプロンプトをプリントできません。現在のところ、<i>autobaud</i> が判別できるボーレートは 110、1200、2400、4800、および 9600 です。</p> <p>ポートが双方向性ポートとして構成されている場合、ttymon によって、ユーザーはサービスに接続できるようになり、またポートが使用されていなければ、uucico(1M)、cu(1C)、または ct(1C) はダイアルアウト用にポートを使用できるようになります。ポートが双方向性であれば、ttymon は文字の読み込みが終わるのを待ってからプロンプトをプリントします。</p> <p>ポート用に <i>connect-on-carrier</i> フラグをセットしている場合、ttymon は、接続要求を受け取るとただちにポートに関連付けられたサービスを呼び出します。プロンプトメッセージは送信されません。</p>

サービスの呼び出し	<p>ポートが動作禁止になっていると、ttymon はこのポートにおいてサービスを呼び出しません。動作禁止メッセージが指定されている場合には、ttymon は接続要求を受け取ると動作禁止メッセージを送信します。ttymon が動作禁止になっている場合、ttymon のインスタンスの制御下にあるすべてのポートが動作禁止になります。</p> <p>ポートに ttymon が呼び出すサービスは、ttymon 管理ファイルに指定されています。ttymon は、このポートに呼び出されるサービスを指示する文字列を走査して、%d または %% の2文字のシーケンスを捜します。%d を見つけると、ttymon は、これらの2文字をこのポートの完全パス名(デバイス名)に置き換えることによって、実行されるサービスコマンドを変更します。%% を見つけると、これらは単一の % に置き換えられます。</p>
セキュリティ	<p>サービスを呼び出すと、読み書きするためにポートデバイスに対して 0、1、および 2 の各ファイル記述子がオープンされます。サービスは、ttymon に登録したときに用いたユーザー名についてのユーザー ID、グループ ID、および現在のホームディレクトリを使用して呼び出されます。2つの環境変数、HOME および TTYPROMPT は、ttymon によってサービスの環境に追加されます。HOME はサービスを呼び出すときに用いるユーザー名のホームディレクトリに設定されています。TTYPROMPT はこのポートのサービスに構成されたプロンプト文字列に設定されています。ttymon が呼び出すサービスが、プロンプトが実際に ttymon によって出されていたかを判別し、出されている場合にはそのプロンプトが実際には何であったかを判断する機能を持つように、設定されています。</p> <p>サービスアクセスコントローラの制御下にある ttymon により監視されるポートに設定できるオプションについては、ttyadm(1M) を参照してください。</p> <p>ttymon は pam(3PAM) を使って、セッション管理を行います。PAM 構成ポリシーは ttymon で使用するモジュールを明記しています。このポリシーは /etc/pam.conf で見ることができます。以下に UNIX セッション管理モジュールを使用する ttymon コマンドのエントリの入った pam.conf ファイルの抜粋を示します。</p> <pre> ttymon session required /usr/lib/security/pam_unix.so.1ttymon </pre> <p>サービスのエントリがない場合には other のサービスのエントリを使用します。</p>
オプション	<p>-g ttymon の特殊な呼び出しは -g オプションを指定して行います。コマンドのこの書式を呼び出せるのは、ポートに正しいボーレートおよび端末設定を設定してから login サービスに接続する必要があるアプリケーションだけにする必要があります。SAC の制御下では前もって構成することはできません。-g は、以下のようなオプションの組み合わせとともに使用できます。</p> <p>-d <i>device</i> <i>device</i> は ttymon の接続先とすべきポートの完全パス名です。このオプションが指定されていない場合は、ファイル記述子 0 は TTY ポートに対する呼び出しプロセスによって設定される必要があります。</p> <p>-h -h フラグが指定されていないと、ttymon は、速度をデフォルト速度または指定速度に設定する前に、速度を 0 に設定することによって回線のハングアップを強制的に行います。</p>

ttymon(1M)

- l *ttylabel* *ttylabel* は `ttydefs` ファイルの速度および TTY 定義に対するリンクです。この定義によって、初期の実行速度、初期の TTY 設定の内容およびユーザーが **BREAK** キーを押して速度が適切でないことを指示する場合に、次に試行する速度が `ttymon` に通知されます。デフォルト速度は 9600 ボーです。
- m *modules* ポートを初期化すると、`ttymon` はポートのモジュールすべてをポップして、指定した順序で *modules* をプッシュします。*modules* はコンマで区切ったプッシュ可能なモジュールのリストです。通常、ポートのデフォルトのモジュールは、オートプッシュファシリティによって設定されます。
- p *prompt* ユーザーはこれを用いるとプロンプト文字列を指定できます。デフォルトのプロンプトは `Login:` です。
- t *timeout* プロンプトの送信後、*timeout* 秒内に何も入力がない場合には `ttymon` を終了します。
- T *termtyp* *termtyp* に `TERM` 環境変数を設定します。

環境

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

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

ファイル

`/etc/logindevperm`

属性

次の属性については `attributes(5)` のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目

`ct(1C)`、`cu(1C)`、`autopush(1M)`、`pmadm(1M)`、`sac(1M)`、`sacadm(1M)`、`sttydefs(1M)`、`ttyadm(1M)`、`uucico(1M)`、`pam(3PAM)`、`logindevperm(4)`、`pam.conf(4)`、`utmpx(4)`、`attributes(5)`、`environ(5)`、`pam_authtok_check(5)`、`pam_authtok_get(5)`、`pam_authtok_store(5)`、`pam_dhkeys(5)`、`pam_passwd_auth(5)`、`pam_unix(5)`、`pam_unix_account(5)`、`pam_unix_auth(5)`、`pam_unix_session(5)`

『Solaris のシステム管理 (基本編)』

ttymon(1M)

使用上の留意点

ポートが複数の ttymon によって監視されている場合は、ttymon は入力を争うような方法でプロンプトメッセージを送信できます。

pam_unix(5) モジュールは、将来のリリースではサポートされなくなる可能性があります。同様の機能は、pam_authok_check(5), pam_authok_get(5), pam_authok_store(5), pam_dhkeys(5), pam_passwd_auth(5), pam_unix_account(5), pam_unix_auth(5)、および pam_unix_session(5) で提供されています。

umount(1M)

名前	mount, umount – ファイルシステムとリモート資源のマウントおよびマウント解除
形式	<pre>mount [-p -v] mount [-F FSType] [generic_options] [-o specific_options] [-O] special mount_point mount [-F FSType] [generic_options] [-o specific_options] [-O] special mount_point mount -a [-F FSType] [-V] [current_options] [-o specific_options] [mount_point...] umount [-f] [-V] [-o specific_options] special mount_point umount -a [-f] [-V] [-o specific_options] [mount_point...]</pre>
機能説明	<p>mount は、ファイルシステムをファイルシステム階層内の <i>mount_point</i> (ディレクトリのパス名) に継ぎ足します。マウント操作を実行する前に <i>mount_point</i> の下に存在したファイルとディレクトリは、ファイルシステムのマウントを解除するまで見えなくなります。</p> <p>umount は、現在マウントされているファイルシステムをマウント解除します。マウントを解除するファイルシステムは <i>mount_point</i> または <i>special</i> (ファイルシステムが存在しているデバイス) のどちらかで指定できます。</p> <p>現在マウントされているファイルシステムのテーブルは、マウント済みファイルシステム情報ファイルにあります。このファイルは通常、<i>/etc/mnttab</i> 上にマウントされているファイルシステムによって提供されます。マウント済のファイルシステムについての情報は、<i>mnttab(4)</i> に格納されています。ファイルシステムをマウントすると、マウントテーブルにエントリが追加されます。(umount で) ファイルシステムをマウント解除すると、テーブルからエントリが削除されます。</p> <p><i>special</i> および <i>mount_point</i> 両方の引数と <i>-F</i> オプションを指定した場合、mount は <i>special</i> 以外のすべての引数の妥当性を検査し、適切な <i>FSType</i> に固有な mount モジュールを呼び出します。引数なしで呼び出した場合、mount は、マウントテーブル <i>/etc/mnttab</i> に記録されているすべてのマウント済みファイルシステムをリストします。引数を一部だけ、たとえば、<i>special</i> か <i>mount_point</i> のどちらか 1 つだけを指定した場合や、<i>special</i> と <i>mount_point</i> の両方を指定したが、<i>FSType</i> は指定しない場合、mount は <i>/etc/vfstab</i> を調べて、指定されていない引数を補完するエントリを探します。そのようなエントリが見つからず、また、<i>special</i> 引数が / で始まる場合、<i>/etc/default/fs</i> に指定されているデフォルトのローカルのファイルシステムタイプが使用されます。それ以外の場合、デフォルトのリモートファイルシステムタイプが使用されます。デフォルトのリモートファイルシステムタイプは、<i>/etc/dfs/fstypes</i> ファイルの最初のエントリによって決定されます。指定されていない引数を補完した後、mount は <i>FSType</i> に固有な mount モジュールを呼び出します。</p> <p>mount または umount を使用してファイルシステムをマウントまたはマウント解除できるのはスーパーユーザーだけです。ただし、マウントされているファイルシステムとリソースを一覧表示するだけであれば、だれでも mount コマンドを使用することができます。</p>

オプション 次のオプションを指定できます。

-F *FSType*

操作の対象となる *FSType* を指定します。*FSType* は、明示的に指定するか、あるいは、`/etc/vfstab` から決定できるか、`/etc/default/fs` または `/etc/dfs/fstypes` を調べることで決定できるものでなければなりません。

-a [*mount_points...*]

可能であれば、複数の `mount` または `umount` 操作を同時に実行します。

マウントポイントを指定しないと、`mount` は `/etc/vfstab` において `mount at boot` フィールドが `yes` に設定されているすべてのファイルシステムをマウントします。マウントポイントを指定すると、`/etc/vfstab` の `mount at boot` フィールドは無視されます。

マウントポイントを指定すると、`umount` は指定されたマウントポイントだけをマウント解除します。マウントポイントを指定しないと、`umount` は `/etc/mnttab` にあるすべてのファイルシステムのマウントを解除します。ただし、`/`、`/usr`、`/var`、`/var/adm`、`/var/run`、`/proc`、`/dev/fd`、および `/tmp` など、システムに必須のファイルシステムは除きます。

-f

ファイルシステムのマウントを強制的に解除します。

このオプションを指定しないと、`umount` は、ファイルシステム上のファイルがビジー状態である場合、ファイルシステムをマウント解除しません。このオプションを指定すると、オープンしているファイルのデータが失われる可能性があります。ファイルシステムをマウント解除した後プログラムがファイルにアクセスしようとすると、エラー (EIO) が返されます。

-p

マウントされているファイルシステムの一覧を `/etc/vfstab` 形式で表示します。このオプションを指定するときは他のオプションを指定してはなりません。「使用上の留意点」を参照してください。

-v

マウントされているファイルシステムの一覧を詳細形式で表示します。このオプションを指定するときは他のオプションを指定してはなりません。

-V

コマンド行全体をエコーしますが、コマンドは実行しません。エコーされたコマンドには、ユーザーが指定したオプションと引数、さらに、`/etc/mnttab` から得られた情報が追加されます。このオプションは、コマンド行を確認および検証するときに使用します。

generic_options

ほとんどの *FSType* に固有なコマンドモジュールで使用できる共通のオプションです。次のオプションを指定できます。

-m

ファイルシステムをマウントしますが、`/etc/mnttab` にエントリを作成しません。

umount(1M)

- g
ファイルシステムを広域的にマウントします。クラスタ化されているシステムでは、クラスタ内にあるすべてのノード上でファイルシステムを広域的にマウントします。クラスタ化されていないシステムでは、このオプションは何の効果もありません。
- o
サブオプションとキーワード属性の組の並を、空白を入れずにコンマで区切った形式で、*FSType* 固有のオプションを指定します。これらのオプションはコマンドの *FSType* に固有なモジュールによって解釈されます (`mount_ufs(1M)` のマニュアルページを参照)。`/etc/vfstab` にエントリがあるファイルシステムで `-o` オプションを指定すると、`/etc/vfstab` 内でそのファイルシステムに指定されている `mount` コマンドのオプションはすべて無視されます。
- O
オーバーレイマウント。既存のマウントポイント上にファイルシステムをマウントできます。これより既存のマウントポイントのファイルシステムにはアクセスできなくなります。このフラグを指定せずに既存のマウントポイント上にファイルシステムをマウントしようとする、マウントは失敗して、`device busy` というエラーメッセージが表示されます。
- r
ファイルシステムを読み取り専用でマウントします。

使用法 2G バイト (2^{31} バイト) 以上のファイルを検出した場合の `mount` と `umount` の動作については、`largefile(5)` のマニュアルページを参照してください。

ファイル

<code>/etc/mnttab</code>	マウントされているファイルシステムのテーブル
<code>/etc/default/fs</code>	デフォルトのローカルのファイルシステムタイプ。デフォルト値は、 <code>/etc/default/fs</code> 内で次のように設定されています。(例: <code>LOCAL=ufs</code>)
	LOCAL: <i>FSType</i> を指定しない場合に、コマンドがデフォルトで使用するパーティション
<code>/etc/vfstab</code>	各ファイルシステム用のデフォルトのパラメータリスト

属性 次の属性については、`attributes(5)` のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 `mount_cachefs(1M)`, `mount_hsfes(1M)`, `mount_nfs(1M)`, `mount_pcfs(1M)`, `mount_tmpfs(1M)`, `mount_ufs(1M)`, `mountall(1M)`, `umountall(1M)`, `mnttab(4)`, `vfstab(4)`, `attributes(5)`, `largefile(5)`, `lofs(7FS)`, `pcfs(7FS)`

umount(1M)

注意事項	ファイルシステムがマウントされるディレクトリがシンボリックリンクの場合、ファイルシステムは、シンボリックリンク自身ではなく、シンボリックリンクが参照するディレクトリ上にマウントされます。
使用上の留意点	mount -p の出力は、cachefs については正しくありません。

umountall(1M)

名前	mountall, umountall – 複数のファイルシステムのマウントおよびマウント解除
形式	<pre> mountall [-F <i>FSType</i>] [-l -r] [<i>file_system_table</i>] umountall [-k] [-s] [-F <i>FSType</i>] [-l -r] [-n] umountall [-k] [-s] [-h <i>host</i>] [-n] </pre>
機能説明	<p>mountall は、ファイルシステムテーブルに指定されているファイルシステムをマウントするときに使用します。ファイルシステムテーブルは <code>vfstab(4)</code> 形式である必要があります。 <code>file_system_table</code> を指定しない場合は、 <code>/etc/vfstab</code> が使用されます。 <code>file_system_table</code> として <code>-</code> を指定すると、mountall は標準入力からファイルシステムテーブルを読み取ります。mountall は、 <code>file_system_table</code> において <code>mount at boot</code> フィールドが <code>yes</code> に設定されているファイルシステムだけをマウントします。</p> <p>ファイルシステムテーブル内のファイルシステムごとに、次のロジックが実行されます。つまり、 <code>/usr/lib/fs/<i>FSType</i>/fsckall</code> というファイル (<i>FSType</i> はファイルシステムのタイプ) が存在する場合、当該ファイルシステムをリストに保存しておき、後でまとめて <code>/usr/lib/fs/<i>FSType</i>/fsckall</code> スクリプトに引数として渡します。 <code>/usr/lib/fs/<i>FSType</i>/fsckall</code> スクリプトは、引数リスト内にあるファイルシステムをすべて検査して、安全にマウントできるかどうかを決定します。 <i>FSType</i> のファイルシステム用の <code>/usr/lib/fs/<i>FSType</i>/fsckall</code> スクリプトが存在しない場合、そのファイルシステムは <code>fsck(1M)</code> を使用して個々に検査されます。検査の結果、ファイルシステムがマウントできない状態であることが判明した場合、マウントを試行する前に <code>fsck</code> で修復されます。 <code>fsckdev</code> フィールドのエントリが <code>-</code> であるファイルシステムは事前の検査なしにマウントされます。</p> <p>umountall は、 <code>root</code>、 <code>/usr</code>、 <code>/var</code>、 <code>/var/adm</code>、 <code>/var/run</code>、 <code>/proc</code>、および <code>/dev/fd</code> を除き、マウントされているファイルシステムをすべてマウント解除します。 <i>FSType</i> を指定すると、mountall と umountall のアクションは指定された <i>FSType</i> に制限されます。 <code>-k</code> を指定した場合でも、umountall がビジー状態のファイルシステムをマウント解除するかどうかは保証されません。</p>
オプション	<p>次のオプションを指定できます。</p> <ul style="list-style-type: none"> <code>-F</code> マウントまたはマウント解除するファイルシステムの <i>FSType</i> を指定します。 <code>-h <i>host</i></code> <i>host</i> からリモートでマウントされている、 <code>/etc/mnttab</code> 内のファイルシステムをすべてマウント解除します。 <code>-k</code> <code>fuser -k <i>mount-point</i></code> コマンドを使用します。詳細については、 <code>fuser(1M)</code> のマニュアルページを参照してください。 <code>-k</code> オプションは <code>SIGKILL</code> シグナルをファイルを使用している各プロセスに送信します。このオプションはプロセスごとに終了シグナルを生成するので、終了メッセージがすぐに出力されないことがあります。 <code>-k</code> を指定した場合でも、umountall がビジー状態のファイルシステムをマウント解除するかどうかは保証されません。 <code>-l</code> アクションをローカルのファイルシステムに制限します。

umountall(1M)

- n 指定したオプションで実行されるアクションをリスト表示します。ただし、実際にはアクションを実行しません。-n オプションを指定せずにコマンドを繰り返すと、コマンドを繰り返す間に /etc/mnttab ファイルは変更されなかったと仮定して、リストされたアクションが実行されます。
- r アクションをリモートのファイルシステムタイプに制限します。
- s 複数の umount 操作を同時に実行しないようにします。

ファイル	/etc/mnttab	マウント済みファイルシステムテーブル
	/etc/vfstab	デフォルトのファイルシステムテーブル
	/usr/lib/fs/ <i>FSType</i> /fsckall	<i>FSType</i> タイプのファイルシステムをすべて検査するときに mountall が呼び出すスクリプト

属性 次の属性については、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWcsu

関連項目 fsck(1M), fuser(1M), mount(1M), mnttab(4), vfstab(4), attributes(5)

診断 ファイルシステムがマウント可能であり、クリーンな状態であれば、メッセージは出力されません。

エラーメッセージと警告メッセージは、fsck(1M) と mount(1M) の両方から送られます。

unshare(1M)

名前	unshare - ローカルシステムをリモートシステムからマウント不可能にする設定				
形式	unshare [-F <i>FSType</i>] [-o <i>specific_options</i>] [<i>pathname</i> <i>resourcename</i>]				
機能説明	unshare コマンドは、共有されているローカルリソースを、ファイルシステムタイプ <i>FSType</i> として利用できないようにします。オプション <i>-FSType</i> を省略した場合、 <i>/etc/dfs/fstypes</i> ファイル内の最初のファイルシステムタイプがデフォルトで使用されます。 <i>Specific_options</i> 、および <i>resourcename</i> の意味は、個々の分散ファイルシステムによって異なります。				
オプション	<table border="0"> <tr> <td style="padding-right: 20px;"><i>-F FSType</i></td> <td>ファイルシステムタイプを指定します。</td> </tr> <tr> <td><i>-o specific_options</i></td> <td><i>-F</i> オプションによって指定されるファイルシステムに固有なオプションを指定します。</td> </tr> </table>	<i>-F FSType</i>	ファイルシステムタイプを指定します。	<i>-o specific_options</i>	<i>-F</i> オプションによって指定されるファイルシステムに固有なオプションを指定します。
<i>-F FSType</i>	ファイルシステムタイプを指定します。				
<i>-o specific_options</i>	<i>-F</i> オプションによって指定されるファイルシステムに固有なオプションを指定します。				
ファイル	<p><i>/etc/dfs/fstypes</i></p> <p><i>/etc/dfs/sharetab</i></p>				
属性	次の属性については、 <i>attributes(5)</i> のマニュアルページを参照してください。				
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">属性タイプ</th> <th style="text-align: center;">属性値</th> </tr> </thead> <tbody> <tr> <td>使用条件</td> <td>SUNWcsu</td> </tr> </tbody> </table>	属性タイプ	属性値	使用条件	SUNWcsu
属性タイプ	属性値				
使用条件	SUNWcsu				
関連項目	<i>share(1M)</i> , <i>shareall(1M)</i> , <i>attributes(5)</i>				
注意事項	<p>共有されている情報で <i>pathname</i> または <i>resourcename</i> のいずれかが見つからない場合、標準エラー出力にエラーメッセージが送られます。</p> <p>unshare コマンドが正常に終了すると、unshare コマンドに指定されたファイルシステムをマウントしているクライアントは、そのファイルシステムにアクセスできなくなります。</p>				

utmp2wtmp(1M)

名前	acct, acctdisk, acctdusg, accton, acctwtmp, closewtmp, utmp2wtmp – アカウンティングおよびいろいろなアカウントコマンドの概要
形式	<pre>/usr/lib/acct/acctdisk /usr/lib/acct/acctdusg [-u filename] [-p filename] /usr/lib/acct/accton [filename] /usr/lib/acct/acctwtmp reason filename /usr/lib/acct/closewtmp /usr/lib/acct/utmp2wtmp</pre>
機能説明	<p>アカウンティングソフトウェアは、アカウンティングシステムを構築するためのツール群です (C 言語のプログラムおよびシェルプロシージャから構成される)。acctsh(1M) では、C 言語のプログラムの一番上に組むシェルプロシージャ群を説明します。</p> <p>接続時間のアカウンティングは記録を /var/adm/wtmpx (utmpx(4) を参照) に書き込むプログラム群により管理されます。acctcon(1M) では、このファイルをセッションおよび課金記録に変換するプログラムを説明します。また、acctmerg(1M) では、この課金記録について集約します。</p> <p>プロセスアカウンティングは、システムカーネルによって行われます。プロセスを終了すると、1つのプロセスにつき1つのレコードが、あるファイル (通常は、/var/adm/pacct) に書き込まれます。acctprc(1M) では、課金のためにこのデータを集約します。コマンドの使用状況を集約するには、acctcms(1M) を使用します。acctcom(1) を用いて、現在のプロセスデータを調査することができます。</p> <p>acctmerg (acct(3HEAD) の tacct フォーマットを参照) を使うと、プロセスのアカウンティングレコードおよび接続時間のアカウンティングレコード (あるいは acct(3HEAD) で説明する tacct フォーマットのアカウンティングレコードのいずれか) をアカウンティングレコードの合計にマージしたり、集約したりすることができます。どのアカウンティングレコードにも、あるいはすべてのアカウンティングレコードにでも prtacct (acctsh(1M) を参照) をフォーマットに使用します。</p> <p>acctdisk は、ユーザー ID、ログイン名およびディスクブロック数がある行を読み取り、他のアカウンティングレコードとマージすることができるアカウンティングレコードの合計に、それらの情報を変換します。入力ファイルが壊れているか、または正しくフォーマットされていない場合は、acctdisk はエラーを返します。</p> <p>acctdusg は、その標準入力を (通常は、find / -print から) 読み取り、ログインによるディスクの資源消費 (間接ブロックを含む) を計算します。</p> <p>accton では、引数がない場合、プロセスアカウンティングをオフにします。filename を指定する場合は、そのファイルは、カーネルがプロセスのアカウンティングレコード (acct(2) および acct(3HEAD) を参照) を追加するための、既存のファイル名である必要があります。</p>

utmp2wtmp(1M)

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

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

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

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

オプション 以下のオプションを指定できます。

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

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

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

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

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

utmp2wtmp(1M)

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

属性 次の属性については attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	SUNWaccu

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

『Solaris のシステム管理 (基本編)』

名前	wall - すべてのユーザーへの通知				
形式	<code>/usr/sbin/wall [-a] [-g grpname] [filename]</code>				
機能説明	<p>wall はファイルの終わりまで標準入力を読み取り、現在ログインしているすべてのユーザーに以下の文字で始まるメッセージを送信します。</p> <pre>Broadcast Message from ...</pre> <p><i>filename</i> が指定されていれば、メッセージはこのファイルから読み込まれます。通常、リモートログインセッションに対応していない仮想端末は無視されます。したがって、ウィンドウシステムを使用しているときは、メッセージはコンソールウィンドウにだけに表示されます。ただし、<code>-a</code> オプションを指定すると、メッセージを仮想端末などにも送信します。</p> <p>一般的には、システムをシャットダウンする前に、すべてのユーザーに警告を与える場合に、このメッセージを用います。</p> <p>ユーザーが起動させた保護を無効にする場合、送信側はスーパーユーザーである必要があります (<code>mesg(1)</code> 参照)。</p> <p>wall は、他のユーザーの端末上で書き込み権を持つために、グループ ID <code>tty</code> に対して <code>setgid()</code> を実行します (<code>setuid(2)</code> 参照)。</p> <p>wall は、ユーザーの端末に送信する前に非表示可能文字を検出します。制御文字は、適切な ASCII 文字が後に続く ^ として表示されます。すなわち、高位ビットが設定された文字は meta 表記法で表示されます。たとえば、<code>\003</code> は ^c、また <code>\372</code> は M-z と表示されます。</p>				
オプション	<p>以下のオプションを指定できます。</p> <p><code>-a</code> コンソールと仮想端末にメッセージを送ります。</p> <p><code>-g grpname</code> グループデータベース (<code>group(4)</code> のマニュアルページを参照) ごとに、<code>grpname</code> で指定されたグループのユーザーだけにメッセージを送ります。</p>				
環境	<p><code>LC_*</code> 変数 (<code>LC_CTYPE</code>、<code>LC_TIME</code>、<code>LC_COLLATE</code>、<code>LC_NUMERIC</code>、<code>LC_MONETARY</code>) が環境に設定されていない場合は、それぞれ対応するロケールのカテゴリにおける wall の動作は、環境変数 <code>LANG</code> によって決定されます (<code>environ(5)</code> 参照)。もし、<code>LC_ALL</code> が設定されていれば、その内容が <code>LANG</code> 変数やその他の <code>LC_*</code> 変数より優先されます。上記の変数が環境にまったく設定されていない場合は、<code>C</code> ロケール (米国スタイル) が wall の動作を決定します。</p>				
ファイル	<code>/dev/tty*</code>				
属性	次の属性については <code>attributes(5)</code> のマニュアルページを参照してください。				
	<table border="1"> <thead> <tr> <th>属性タイプ</th> <th>属性値</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	属性タイプ	属性値		
属性タイプ	属性値				

wall(1M)

使用条件	SUNWcsu
------	---------

関連項目 `mesg(1)`, `write(1)`, `setuid(2)`, `attributes(5)`, `environ(5)`

注意事項 `wall` は、ユーザーの `tty` ファイルでオープンが失敗すると、`Cannot send to . . .` と表示します。