

Solaris のシステム管理 (基本編)



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

Part No: 819-0378-18
2009 年 10 月

Sun Microsystems, Inc. (以下 米国 Sun Microsystems 社とします) は、本書に記述されている製品に含まれる技術に関連する知的財産権を所有します。特に、この知的財産権はひとつかそれ以上の米国における特許、あるいは米国およびその他の国において申請中の特許を含んでいることがあります。それらに限定されるものではありません。

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。

U.S. Government Rights Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

この配布には、第三者によって開発された素材を含んでいることがあります。

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ョービマジッ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、Sun のロゴマーク、Solaris のロゴマーク、Java Coffee Cup のロゴマーク、docs.sun.com Java、JavaHelp、J2EE、JumpStart、Solstice、Sun Blade、SunSolve、SunSpectrum、Sun xVM hypervisor、ZFS、Java および Solaris は、米国およびその他の国における米国 Sun Microsystems 社の商標、登録商標もしくは、サービスマークです。

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。DLT は、米国およびその他の国において米国 Quantum Corporation の商標として使用されています。Netscape および Mozilla は米国およびその他の国における Netscape Communications Corporation の商標または登録商標です。

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。

Wnn8 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(C) OMRON Co., Ltd. 1995-2000. All Rights Reserved. Copyright(C) OMRON SOFTWARE Co., Ltd. 1995-2009 All Rights Reserved.

「ATOK for Solaris」は、株式会社ジャストシステムの著作物であり、「ATOK for Solaris」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。

「ATOK」および「推測変換」は、株式会社ジャストシステムの登録商標です。

「ATOK for Solaris」に添付す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』に添付のものを使用しています。

「ATOK for Solaris」に含まれる郵便番号辞書(7桁/5桁)は日本郵政公社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)。

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、OPEN LOOK のグラフィカル・ユーザインタフェースを実装するか、またはその他の方法で米国 Sun Microsystems 社との書面によるライセンス契約を遵守する、米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

本書で言及されている製品や含まれている情報は、米国輸出規制法で規制されるものであり、その他の国の輸出入に関する法律の対象となることがあります。核、ミサイル、化学あるいは生物兵器、原子力の海洋輸送手段への使用は、直接および間接を問わず厳しく禁止されています。米国が禁輸の対象としている国や、限定はされませんが、取引禁止顧客や特別指定国民のリストを含む米国輸出排除リストで指定されているものの輸出および再輸出は厳しく禁止されています。

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ない、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ないものとします。

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: System Administration Guide: Basic Administration

Part No: 817-1985-20

Revision A

目次

はじめに	17
1 Solaris 管理ツール (製品概要)	23
Solaris 管理ツールの新機能	23
Solaris 管理ツールとサポートされるリリースを示すマトリックス	25
Solaris 管理ツールの機能説明	26
Solaris 9 管理ツールの機能説明	27
Solaris 管理コマンドの有効性	28
Solaris 10 システム管理コマンド	28
Solaris 管理ツールの詳細情報	29
2 Solaris 管理コンソールの操作 (手順)	31
Solaris 管理コンソール (概要)	31
Solaris 管理コンソールについて	31
Solaris 管理コンソールツール	32
Solaris 管理コンソールを使用する理由	34
Solaris 管理コンソールの構成	35
Solaris 管理コンソールウィンドウの変更	36
Solaris 管理コンソールのマニュアル	36
役割によるアクセス制御の程度	36
スーパーユーザー (root) になるか役割を引き受ける	38
▼ スーパーユーザー (root) になるか役割を引き受ける方法	39
Solaris 管理ツールを RBAC と組み合わせて使用する (作業マップ)	40
コンソールに最初にログインした場合	41
プライマリ管理者の役割を作成する	42
▼ 最初の役割 (プライマリ管理者) を作成する方法	43
▼ プライマリ管理者の役割を引き受ける方法	44

Solaris 管理コンソールを起動する	45
▼ スーパーユーザーまたは役割としてコンソールを起動する方法	45
ネームサービス環境で Solaris 管理ツールを使用する (作業マップ)	47
RBAC セキュリティーファイル	47
ネームサービス環境で Solaris 管理コンソールを使用するための前提条件	49
管理範囲	49
/etc/nsswitch.conf ファイル	49
▼ 特定環境用のツールボックスを作成する方法	50
▼ ツールボックスにツールを追加する方法	52
▼ ネームサービス環境で Solaris 管理コンソールを起動する方法	53
Solaris 管理コンソールにツールを追加する	53
▼ レガシーツールをツールボックスに追加する方法	53
▼ 別製品のツールをインストールする方法	54
Solaris 管理コンソールのトラブルシューティング	55
▼ Solaris 管理コンソールのトラブルシューティングの方法	55
3 Sun Java Web Console の操作 (手順)	57
Java Web Console の管理に関する新機能	57
Java Web Console サーバーの管理	57
Java Web Console で使用できるアプリケーション	58
Java Web Console (概要)	58
Java Web Console について	59
Java Web Console の管理コマンド	60
サポートされている Web ブラウザ	60
Java Web Console を開始する (作業マップ)	61
Java Web Console を開始する	61
▼ Java Web Console の起動ページからアプリケーションを起動する方法	62
コンソールサービスの管理	64
▼ コンソールサービスを起動する方法	64
▼ システムの起動時に実行されるようコンソールサービスを有効にする方法	65
▼ コンソールサービスを停止する方法	65
▼ コンソールサービスを無効にする方法	66
Java Web Console を設定する	66
▼ Java Web Console のプロパティーを変更する方法	69
Java Web Console のユーザー ID	70

コンソールのデバッグトレースログを使用する	71
Java Web Console ソフトウェアのトラブルシューティング (作業マップ)	73
Java Web Console ソフトウェアのトラブルシューティング	74
コンソールの状態とプロパティの確認	74
コンソールへのアクセスに関連する問題	77
アプリケーションの登録に関連する問題	77
Java Web Console のリファレンス情報	83
Java Web Console のセキュリティに関する注意事項	83
authTypes タグを使って承認を指定する	85
Java Web Console へのリモートアクセスを使用可能にする	87
Java Web Console へのリモートアクセスを使用不可能にする	87
Java Web Console の内部パスワードを変更する	88
4 ユーザーアカウントとグループの管理 (概要)	91
ユーザーとグループの管理における新機能	91
ユーザーアカウントとグループアカウントの管理用のツール	92
ユーザーアカウントとグループとは	92
ユーザーアカウントの構成要素	93
ユーザー名、ユーザー ID、およびグループ ID を使用するガイドライン	100
ユーザーアカウントとグループ情報の格納場所	101
passwd ファイルのフィールド	102
デフォルトの passwd ファイル	102
shadow ファイルのフィールド	103
group ファイルのフィールド	104
デフォルトの group ファイル	104
ユーザーアカウントとグループを管理するツール	106
Solaris ユーザーおよびグループ管理ツールの作業	106
プロジェクトでユーザーおよびリソースを管理する	109
ユーザーの作業環境のカスタマイズ	110
サイト初期設定ファイルの使用方法	111
ローカルシステムへの参照を避ける	112
シェル機能	112
シェル環境	113
PATH 変数	116
ロケール変数	117

デフォルトのファイルアクセス権 (umask)	118
ユーザー初期設定ファイルとサイト初期設定ファイルの例	119
5 ユーザーアカウントとグループの管理(手順)	123
ユーザーアカウントの設定(作業マップ)	123
ユーザー情報の収集	124
▼ユーザー初期設定ファイルをカスタマイズする方法	125
▼Solaris 管理コンソールのグループツールを使ってグループを追加する方法	127
▼Solaris 管理コンソールのユーザーツールを使ってユーザーを追加する方法	128
コマンド行ツールを使ってグループやユーザーを追加する	129
Solaris 管理コンソールによるホームディレクトリの設定	130
▼ユーザーのホームディレクトリを共有する方法	131
▼ユーザーのホームディレクトリをマウントする方法	132
ユーザーアカウントの管理(作業マップ)	134
ユーザーアカウントを変更する	135
▼グループを変更する方法	136
▼グループを削除する方法	137
パスワードの管理	138
パスワード有効期限を設定する	138
▼ユーザーアカウントを無効にする方法	138
▼ユーザーのパスワードを変更する方法	139
▼ユーザーアカウントのパスワード有効期限を設定する方法	140
▼ユーザーアカウントを削除する方法	141
6 クライアントとサーバーサポートの管理(概要)	143
クライアントとサーバーサポートの管理に関する新機能	144
bootadm -p コマンドによるプラットフォーム指定のサポート	144
nfs4_domain キーワードがディスクレスクライアントのブートに与える影響	144
x86: GRUB ブート環境のディスクレスクライアントの変更点	145
x86: smdiskless コマンドの変更点	145
クライアントとサーバータスクの操作手順	146
サーバー、クライアント、およびアプライアンスとは	146
クライアントサポートとは	147
システムタイプの概要	148
サーバー	148

スタンドアロンシステム	149
ディスクレスクライアント	150
アプライアンス	150
システムタイプ選択のガイドライン	150
ディスクレスクライアント管理の概要	151
OS サーバーおよびディスクレスクライアントのサポート情報	152
ディスクレスクライアント管理機能	153
OS サーバーに必要なディスク容量	156
7 ディスクレスクライアントの管理 (手順)	159
ディスクレスクライアントの管理 (作業マップ)	159
ディスクレスクライアントの管理の準備	161
▼ x86: GRUB ベースのブート環境にディスクレスクライアントを追加するための準備方法	163
▼ Solaris 10 OS にディスクレスクライアントを追加するための準備方法	165
▼ ディスクレスクライアントサポートの OS サービスの追加方法	167
▼ x86: GRUB ベースのブート環境にディスクレスクライアントを追加する方法 ..	170
▼ Solaris 10 OS でディスクレスクライアントを追加する方法	173
▼ x86: GRUB を使用してディスクレスクライアントをブートする方法	176
▼ SPARC: Solaris 10 OS でディスクレスクライアントをブートする方法	177
▼ ディスクレスクライアントサポートの削除方法	178
▼ ディスクレスクライアントの OS サービスを削除する方法	178
ディスクレスクライアント OS サービスにパッチを適用する	179
ディスクレスクライアントの OS パッチの表示	180
▼ ディスクレスクライアントの OS パッチの追加方法	180
ディスクレスクライアント問題のトラブルシューティング	183
ディスクレスクライアントのインストールの問題	183
ディスクレスクライアントの一般的な問題のトラブルシューティング	187
8 システムのシャットダウンとブートの概要	193
システムのシャットダウンとブートに関する新機能	193
Solaris リリースのインストールおよびブートでの 2T バイトディスクのサポート	194
ZFS ブートのサポート	194
x86: 新しい findroot コマンド	194

bootadm コマンドによるプラットフォーム指定のサポート	195
再設計された Solaris SPARC ブートストラッププロセス	195
x86: 電源ボタンによるシステムシャットダウン開始のサポート	196
シャットダウンとブートについての参照先	196
シャットダウンとブートの用語	197
システムのシャットダウンに関するガイドライン	198
システムのブートに関するガイドライン	199
システムをシャットダウンする場合	199
システムをブートする場合	200
9 システムのシャットダウンとブート (概要)	203
Solaris ブートデザインの基礎	204
新しい Solaris SPARC ブートアーキテクチャーについて	205
ミニルートの圧縮と展開	206
ソフトウェアのインストールとアップグレード	207
インストールのメモリー要件	207
ネットワークブートサーバーの設定プロセスの変更点	207
複数の Solaris カーネルのブートのサポート	208
Solaris SPARC でのブートアーカイブの実装	208
x86: GRUB ブートローダーの管理	209
GRUB ベースのブートの動作	209
新しい findroot コマンドの GRUB サポート	210
ZFS ルートファイルシステムからの起動	211
ZFS の Solaris インストール要件	211
ZFS ルートファイルシステムからのブートの動作	212
SPARC: ZFS ルートファイルシステムからのブートをサポートするブートオプション	212
x86: ZFS ルートファイルシステムからのブートをサポートするブートオプション	213
10 システムのシャットダウン (手順)	215
システムのシャットダウン (作業マップ)	215
システムのシャットダウン	216
システムシャットダウンコマンド	216
システムのダウン時間をユーザーに通知する	217

▼ システムにログインしているユーザーを知る方法	218
▼ サーバーをシャットダウンする方法	218
▼ スタンドアロンシステムをシャットダウンする方法	222
すべてのデバイスの電源を落とす	224
▼ すべてのデバイスの電源を落とす方法	224
11 Solaris ブート動作の変更 (手順)	227
SPARC システムでのブート動作の変更 (作業マップ)	227
SPARC: ブート PROM の使用	228
▼ SPARC: システムの PROM リビジョン番号を確認する方法	229
▼ SPARC: システム上のデバイスを確認する方法	229
▼ SPARC: デフォルトのブートデバイスを調べる方法	231
▼ SPARC: ブート PROM を使用してデフォルトのブートデバイスを変更する方 法	231
▼ SPARC: eeprom コマンドを使用してデフォルトのブートデバイスを変更する方 法	233
SPARC: システムのリセット	234
▼ SPARC: ブート PROM を使用してデフォルトのカーネルを変更する方法	234
▼ SPARC: eeprom コマンドを使用してデフォルトのカーネルを変更する方法	234
x86 システムでの Solaris ブート動作の変更 (作業マップ)	235
x86 システムのブート動作を変更する	236
x86: ブート時に GRUB メニューを編集してブート動作を変更する	237
GRUB 編集メニューについて	238
ブート時に GRUB メニューを編集するときに指定できるブート引数	239
▼ x86: ブート時に GRUB メニューを編集してブート動作を変更する方法	240
x86: menu.lst ファイルを編集してブート動作を変更する	241
▼ x86: menu.lst ファイルを編集してブート動作を変更する方法	242
x86: アクティブな GRUB menu.lst ファイルを検出する	246
x86: findroot コマンドの実装	246
▼ x86: findroot コマンドを使用する GRUB メニューエントリを追加する方法	248
12 Solaris システムのブート (手順)	251
SPARC システムのブート (作業マップ)	251
SPARC システムのブート	253
▼ SPARC: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法	253

▼ SPARC: システムを実行レベル S (シングルユーザーレベル) でブートする方法	254
▼ SPARC: システムを対話式でブートする方法	255
▼ SPARC: デフォルトのカーネル以外の Solaris カーネルをブートする方法	257
SPARC システムで ZFS ルートファイルシステムからブートする	259
▼ SPARC: ZFS ルートプール内の使用可能なブート可能データセットのリストを表示する方法	260
▼ SPARC: ZFS ルートファイルシステムからブートする方法	261
SPARC システムでフェイルセーフアーカイブをブートする	264
▼ SPARC システムでフェイルセーフアーカイブをブートする方法	265
SPARC システムをネットワークからブートする	269
▼ SPARC: システムをネットワークからブートする方法	270
GRUB を使用して x86 システムをブートする (作業マップ)	271
▼ x86: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法	272
▼ x86: システムを実行レベル S (シングルユーザーレベル) でブートする方法	274
▼ x86: システムを対話式でブートする方法	277
x86 システムで ZFS ルートファイルシステムから起動する	279
▼ x86 システム上の使用可能な ZFS ブート環境のリストを表示する方法	279
▼ x86 システムで ZFS ルートファイルシステムからブートする方法	280
x86 システムでフェイルセーフアーカイブをブートする	284
▼ GRUB を使用して x86 システムでフェイルセーフアーカイブをブートする方法	284
▼ x86: フェイルセーフアーカイブをブートして、破壊されたブートアーカイブを強制的に更新する方法	286
x86 システムをネットワークからブートする	289
x86: DHCP マクロについて	290
▼ x86: GRUB ベースのブートをネットワークから実行する方法	292
13 Solaris システムのブートのトラブルシューティング (手順)	295
SPARC プラットフォームのブートのトラブルシューティング (作業マップ)	295
▼ SPARC: 復旧を目的としてシステムを停止する方法	296
SPARC: クラッシュダンプを強制してシステムをリブートする	297
▼ SPARC: 復旧を目的としてシステムをブートする方法	299
▼ SPARC: カーネルデバッグ (kmdb) を使ってシステムをブートする方法	301
x86 プラットフォームのブートのトラブルシューティング (作業マップ)	302
▼ x86: 復旧を目的としてシステムを停止する方法	302
x86: クラッシュダンプを強制してシステムをリブートする	303

▼ x86: GRUB ブート環境でカーネルデバッガ (kmdb) を使ってシステムをブートする方法	304
14 Solaris ブートアーカイブの管理 (手順)	307
Solaris ブートアーカイブの管理 (作業マップ)	307
Solaris ブートアーカイブの説明	308
boot-archive サービスの管理	309
▼ boot-archive サービスを有効または無効にする方法	310
▼ boot-archive サービスをクリアして矛盾したブートアーカイブを更新する方法	310
bootadm コマンドを使用してブートアーカイブを管理する	313
▼ ブートアーカイブを手動で更新する方法	313
▼ RAID-1 (ミラー) ボリューム上のブートアーカイブを手動で更新する方法	314
▼ ブートアーカイブの内容を一覧表示する方法	320
▼ x86: アクティブな GRUB メニューを検出し、現在のメニューエントリを一覧表示する方法	320
▼ x86: アクティブな GRUB メニューのデフォルトのブートエントリを設定する方法	321
15 x86: GRUB ベースのブート (参照情報)	323
x86: ブートプロセス	323
x86: システム BIOS	323
x86: カーネル初期化プロセス	324
x86: GRUB ブートローダーの Solaris サポート	324
x86: GRUB の用語	324
x86: GRUB の機能コンポーネント	326
GRUB で複数のオペレーティングシステムをサポートする方法	328
x86: サポートされている GRUB 実装	329
16 x86: GRUB を実装しないシステムのブート (手順)	333
x86: システムのブート (作業マップ)	333
x86: GRUB を実装しないシステムをブートする	335
▼ x86: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法	336
▼ x86: システムを実行レベル S (シングルユーザーレベル) でブートする方法	339
▼ x86: システムを対話式でブートする方法	340

x86: ネットワークからのブート	343
▼ x86: システムをネットワークからブートする方法	343
x86: Device Configuration Assistant (デバイス構成用補助) の使用	345
▼ x86: 復旧を目的としてシステムを停止する方法	346
▼ x86: 復旧を目的としてシステムをブートする方法	347
▼ x86: カーネルデバッガ (kmdb) を使ってシステムをブートする方法	349
x86: クラッシュダンプを強制してシステムをリブートする	352
x64: 64 ビットブートの失敗のトラブルシューティング	354
x86: ブートプロセス (参照情報)	355
x86: ブートサブシステム	355
x86: ブートプロセス	361
x86: ブートファイル	362
17 サービスの管理 (概要)	365
SMF について	365
SMF 使用時の動作に関する変更	367
SMF の概念	367
SMF サービス	367
サービス識別子	368
サービスの状態	369
SMF 目録	370
SMF プロファイル	370
サービス構成リポジトリ	371
SMF リポジトリのバックアップ	371
SMF スナップショット	371
SMF 管理およびプログラミングインタフェース	372
SMF コマンド行管理ユーティリティ	372
サービス管理構成ライブラリインタフェース	372
SMF コンポーネント	373
SMF マスターリスタータデーモン	373
SMF 委任リスタータ	373
SMF とブート	374
SMF 互換性	374
実行レベル	375
実行レベルとマイルストンの使い分け	376

システムの実行レベルを確認する	376
/etc/inittab ファイル	377
システムが実行レベル 3 になると実行される処理	378
18 サービスの管理 (手順)	379
サービスの管理 (作業マップ)	379
SMF サービスの監視	380
▼ サービスの状態を一覧表示する方法	380
▼ 特定のサービスに依存するサービスを表示する方法	382
▼ 特定のサービスが依存するサービスを表示する方法	382
SMF サービス (作業マップ)	383
SMF サービスの管理	384
RBAC 権利プロファイルを使って SMF を管理する	384
▼ サービスインスタンスを無効にする方法	384
▼ サービスインスタンスを有効にする方法	385
▼ サービスを再起動する方法	386
▼ 保守状態のサービスを復元する方法	386
▼ 別の SMF スナップショットに戻す方法	387
▼ SMF プロファイルを作成する方法	388
▼ SMF プロファイルを適用する方法	389
▼ ネットワークに提供するサービスを generic*.xml で変更する	390
SMF サービスの設定	391
▼ サービスを変更する方法	391
▼ サービスの環境変数を変更する方法	391
▼ inetd 制御サービスのプロパティーを変更する方法	392
▼ inetd 制御サービスのコマンド行引数を変更する方法	394
▼ inetd.conf エントリを変換する方法	395
実行制御スクリプトの使用 (作業マップ)	396
実行制御スクリプトの使用	396
▼ 実行制御スクリプトを使用して従来のサービスを起動または停止する方法	396
▼ 実行制御スクリプトを追加する方法	397
▼ 実行制御スクリプトを無効にする方法	398
サービス管理機能のトラブルシューティング	399
▼ 起動していないサービスをデバッグする	399
▼ 破壊されたりポジトリを修復する方法	400

▼ どのサービスも起動しないでブートする方法	402
▼ ブート中に <code>system/filesystem/local:default</code> サービスで障害が発生した場合に <code>sulogin</code> プロンプトを強制的に表示する方法	403
19 ソフトウェアの管理 (概要)	405
Solaris オペレーティングシステムでのソフトウェア管理における新機能	406
ゾーンをサポートするためのパッケージおよびパッチツールの拡張	407
遅延起動パッチ	408
Solaris OS に組み込まれた Common Agent Container	408
<code>patchadd -M</code> コマンドが複数のパッチを処理する方法の改善	408
パッケージおよびパッチツールの拡張	409
ソフトウェア管理作業についての参照先	409
ソフトウェアパッケージの概要	410
署名付きのパッケージ、パッチ、およびソフトウェア更新	410
ソフトウェアパッケージを管理するためのツール	414
ソフトウェアパッケージの追加または削除 (<code>pkgadd</code>)	415
ソフトウェアパッケージの追加にあたっての重要な注意点 (<code>pkgadd</code>)	416
パッケージの削除に関するガイドライン (<code>pkgrm</code>)	416
ゾーンに対応していない Solaris リリースでソフトウェアパッケージおよびパッチを 追加および削除する際の制限	417
パッケージ追加時のユーザーの対話操作を省略する (<code>pkgadd</code>)	418
管理ファイルの使用	418
応答ファイルの使用 (<code>pkgadd</code>)	419
20 Solaris システム管理ツールによるソフトウェアの管理 (手順)	421
Solaris Product Registry および Solaris インストール GUI ツール (ソフトウェア管理 用)	421
Solaris インストール GUI によるソフトウェアの追加	422
▼ Solaris インストール GUI プログラムを使ってソフトウェアをインストールする方 法	422
Solaris Product Registry の GUI によるソフトウェアの管理 (作業マップ)	423
▼ インストールまたはアンインストールしたソフトウェアの情報を Solaris Product Registry の GUI を使って表示する方法	426
▼ Solaris Product Registry の GUI を使ってソフトウェアをインストールする方法 ..	427
▼ Solaris Product Registry の GUI を使ってソフトウェアをアンインストールする方 法	428

Solaris Product Registry のコマンド行インタフェースによるソフトウェアの管理 (作業マップ)	429
Solaris Product Registry のコマンド行インタフェースによるソフトウェアの管理	430
▼ インストールまたはアンインストールしたソフトウェアの情報を表示する方法 (prodreg)	430
▼ ソフトウェア属性を表示する方法 (prodreg)	433
▼ ソフトウェアの依存関係を確認する方法 (prodreg)	436
▼ 損傷を受けたソフトウェア製品を特定する方法 (prodreg)	437
▼ ソフトウェアをアンインストールする方法 (prodreg)	440
▼ 損傷を受けたソフトウェアをアンインストールする方法 (prodreg)	444
▼ 損傷を受けたソフトウェアコンポーネントを再インストールする方法 (prodreg)	448
 21 パッケージコマンドによるソフトウェアの管理 (手順)	451
pkgadd コマンドによる署名付きパッケージの追加および削除 (作業マップ)	451
pkgadd コマンドによる署名付きパッケージの追加および削除	452
▼ 信頼される証明書を Java キーストアからインポートする方法 (pkgadm addcert)	452
▼ 証明書の情報を表示する方法 (pkgadm listcert)	454
▼ 証明書を削除する方法 (pkgadm removecert)	455
▼ プロキシサーバーをセットアップする方法 (pkgadd)	455
▼ 署名付きパッケージを追加する方法 (pkgadd)	457
パッケージコマンドによるソフトウェアパッケージの管理 (作業マップ)	458
パッケージコマンドによるソフトウェアパッケージの管理	459
▼ ソフトウェアパッケージを追加する方法 (pkgadd)	459
ソフトウェアパッケージをスプールディレクトリに追加する	462
▼ インストール済みのすべてのパッケージに関する情報を表示する方法 (pkginfo)	465
▼ インストール済みのソフトウェアパッケージの整合性を検査する方法 (pkgchk)	466
▼ インストール済みオブジェクトの整合性を検査する方法 (pkgchk -p、pkgchk -P)	467
ソフトウェアパッケージを削除する	469
▼ ソフトウェアパッケージを削除する方法 (pkgrm)	470

22	patchadd コマンドによる Solaris パッチの管理 (手順)	473
	パッチの種類	474
	署名付きパッチと署名なしパッチ	474
	Solaris パッチへのアクセス	475
	Solaris パッチの番号付け	475
	Solaris パッチの管理	475
	Solaris オペレーティングシステムでのパッチ管理	476
	システムに署名付きパッチ、署名なしパッチのいずれを適用するか決定	477
	Solaris パッチ管理の用語と定義	477
	patchadd コマンドによる Solaris パッチの管理 (作業マップ)	479
	▼ 信頼される証明書をパッケージキーストアにインポートする方法	480
	Java キーストアからのルート CA 証明書のエクスポート	481
	▼ Web プロキシを指定する方法	482
	patchadd -R を使って代替 root パスを作成する際の制限	483
	▼ Solaris パッチをダウンロードおよび適用する方法	483
	▼ Solaris パッチの情報を表示する方法	485
	▼ patchrm コマンドを使用して Solaris パッチを削除する方法	486
A	SMF サービス	487
	索引	493

はじめに

『Solaris のシステム管理 (基本編)』は、Solaris™ システム管理に関する重要な情報を提供するマニュアルの一部です。本書には、SPARC® および x86 の両方のシステムに関する情報が含まれています。

本書では、次のことを前提としています。

- SunOS™ 5.10 オペレーティングシステム (Solaris OS) がインストールされていること
- 使用するネットワークソフトウェアの設定が完了していること

システム管理者にとって重要と思われる Solaris 10 リリースの新機能については、各章の初めにある新機能に関する節を参照してください。

注 - この Solaris のリリースでは、SPARC および x86 系列のプロセッサアーキテクチャをサポートしています。サポートされるシステムについては、[Solaris OS: Hardware Compatibility Lists \(http://www.sun.com/bigadmin/hcl\)](http://www.sun.com/bigadmin/hcl) を参照してください。本書では、プラットフォームにより実装が異なる場合は、それを特記します。

本書の x86 に関連する用語については、以下を参照してください。

- 「x86」は、64 ビットおよび 32 ビットの x86 互換製品系列を指します。
- 「x64」は、特に 64 ビットの x86 互換 CPU に関連しています。
- 「32 ビット x86」は、x86 をベースとするシステムに関する 32 ビット特有の情報を指します。

サポートされるシステムについては、Solaris OS: Hardware Compatibility List を参照してください。

対象読者

このマニュアルは、Solaris 10 リリースが稼働しているシステムの管理者を対象としています。このマニュアルを読むには、UNIX® システム管理について 1-2 年の経験が必要です。UNIX システム管理のトレーニングコースに参加することも役に立ちます。

Solaris システム管理マニュアルセットの構成

システム管理マニュアルセットに含まれる各マニュアルとその内容は、次のとおりです。

マニュアルのタイトル	トピック
『Solaris のシステム管理 (基本編)』	ユーザーアカウントとグループ、サーバーとクライアントのサポート、システムのシャットダウンとブート、管理サービス、およびソフトウェアの管理 (パッケージとパッチ)
『Solaris のシステム管理 (上級編)』	端末とモデム、システムリソース (ディスク割り当て、アカウントティング、および crontab)、システムプロセス、および Solaris ソフトウェアの障害追跡
『Solaris のシステム管理 (デバイスとファイルシステム)』	リムーバブルメディア、ディスクとデバイス、ファイルシステム、およびデータのバックアップと復元
『Solaris のシステム管理 (IP サービス)』	TCP/IP ネットワーク管理、IPv4 と IPv6 アドレス管理、DHCP、IPsec、IKE、Solaris IP フィルタ、モバイル IP、IP ネットワークマルチのパス化 (IPMP)、および IPQoS
『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』	DNS、NIS、および LDAP のネーミングとディレクトリサービス (NIS から LDAP への移行、および NIS+ から LDAP への移行を含む)
『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』	NIS+ のネーミングとディレクトリサービス
『Solaris のシステム管理 (ネットワークサービス)』	Web キャッシュサーバー、時間関連サービス、ネットワークファイルシステム (NFS と Autofs)、メール、SLP、および PPP
『Solaris のシステム管理 (印刷)』	Solaris の印刷に関するトピックと作業、印刷サービスやプリンタを設定して管理するためのサービス、ツール、プロトコル、およびテクノロジーの使用法
『Solaris のシステム管理 (セキュリティサービス)』	監査、デバイス管理、ファイルセキュリティ、ティー、BART、Kerberos サービス、PAM、Solaris 暗号化フレームワーク、特権、RBAC、SASL、および Solaris Secure Shell
『Solaris のシステム管理 (Solaris コンテナ : 資源管理と Solaris ゾーン)』	リソース管理に関連する計画と作業、拡張アカウントティング、リソース制御、フェアシェアスケジューラ (FSS)、資源上限デーモン (rcapd) による物理メモリーの制御、および資源プール (Solaris Zones ソフトウェア区分技術と 1x ブランドゾーンによる仮想化)

マニュアルのタイトル	トピック
『Solaris ZFS 管理ガイド』	ZFS ストレージプールおよびファイルシステムの作成と管理、スナップショット、クローン、バックアップ、アクセス制御リスト (ACL) による ZFS ファイルの保護、ゾーンがインストールされた Solaris システム上での ZFS の使用、エミュレートされたボリューム、およびトラブルシューティングとデータ回復
『Solaris Trusted Extensions 管理の手順』	Solaris Trusted Extensions システム固有のシステム管理
『Solaris Trusted Extensions 構成ガイド』	Solaris 10 5/08 リリース以降での、Solaris Trusted Extensions の計画、有効化、および初期設定の方法

関連する Sun 以外の Web サイト情報

注 - このマニュアルで説明する Sun 以外の Web サイトの利用については、Sun は責任を負いません。こうしたサイトやリソース上の、またはこれらを通じて利用可能な、コンテンツ、広告、製品、その他の素材について、Sun は推奨しているわけではなく、Sun はいかなる責任も負いません。Sun は、これらのサイトあるいはリソースに関する、あるいはこれらのサイトから利用可能であるコンテンツ、製品、サービスのご利用あるいは信頼によって、あるいはそれに関連して発生するいかなる損害、損失、申し立てに対する一切の責任を負いません。

マニュアル、サポート、およびトレーニング

Sun の Web サイトでは、次のサービスに関する情報も提供しています。

- マニュアル (<http://jp.sun.com/documentation/>)
- サポート (<http://jp.sun.com/support/>)
- トレーニング (<http://jp.sun.com/training/>)

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 P-1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% su password:
AaBbCc123	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm <i>filename</i> と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「 」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% grep '^#define \ XV_VERSION_STRING'

コード例は次のように表示されます。

■ C シェル

```
machine_name% command y|n [filename]
```

■ C シェルのスーパーユーザー

```
machine_name# command y|n [filename]
```

■ Bourne シェルおよび Korn シェル

```
$ command y|n [filename]
```

■ Bourne シェルおよび Korn シェルのスーパーユーザー

```
# command y|n [filename]
```

[] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

|は区切り文字(セパレータ)です。この文字で分割されている引数のうち1つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します(例:Shift キーを押します)。ただし、キーボードによってはEnter キーがReturn キーの動作をします。

ダッシュ(-)は2つのキーを同時に押すことを示します。たとえば、Ctrl-D はControl キーを押したままD キーを押すことを意味します。

一般規則

- このマニュアルでは、英語環境での画面イメージを使っています。このため、実際に日本語環境で表示される画面イメージとこのマニュアルで使っている画面イメージが異なる場合があります。本文中で画面イメージを説明する場合には、日本語のメニュー、ボタン名などの項目名と英語の項目名が、適宜併記されています。

一般規則

このマニュアルでは次の規則が使用されています。

- このマニュアル中の手順を実行したり、例(コマンド入力、コードなど)を使用する場合には、二重引用符(")、左一重引用符(‘)、右一重引用符(’)をそれぞれ間違えないように注意してください。
- このマニュアル中で「Return キー」と表記しているキーは、キーボードによっては「Enter キー」という名前になっていることがあります。
- /sbin、/usr/sbin、/usr/bin、/etc ディレクトリにあるコマンドについては、このマニュアルでは絶対パス名で表記していない場合があります。ただし、それ以外のあまり一般的でないディレクトリにあるコマンドについては、このマニュアル中の例では絶対パスで表記します。

Solaris 管理ツール (製品概要)

この章では、Solaris 管理ツールの製品概要について説明します。

- 23 ページの「Solaris 管理ツールの新機能」
- 25 ページの「Solaris 管理ツールとサポートされるリリースを示すマトリックス」
- 26 ページの「Solaris 管理ツールの機能説明」
- 27 ページの「Solaris 9 管理ツールの機能説明」
- 29 ページの「Solaris 管理ツールの詳細情報」

注 - **Solaris 10 5/08:** この情報は Solaris 10 5/08 リリースで追加されたものですが、すべての Solaris 10 OS に当てはまります。使用している Solaris システムを登録するには、<https://inventory.sun.com/inventory/> にアクセスしてください。Sun Inventory を使用してハードウェア、ソフトウェア、およびオペレーティングシステムを登録する方法については、[Sun Inventory Information Center \(http://wikis.sun.com/display/SunInventory/Sun+Inventory\)](http://wikis.sun.com/display/SunInventory/Sun+Inventory) を参照してください。

Sun Ops Center を使ってデータセンターのシステムのプロビジョニング、更新、および管理を行う場合は、Ops Center のドキュメントを参照し、ソフトウェアを登録する方法についての情報を確認してください。

Solaris 管理ツールの新機能

この Solaris リリースで新たに導入された Solaris 管理ツールはありません。

Solaris 10 初期 3/05 リリースでは、次のツールが新たに追加または変更されました。

- `admintool` - Solaris 10 リリース以降、このツールは使用できなくなりました
- パッケージおよびパッチツールの拡張

Solaris の新機能の一覧および Solaris リリースについての説明は、『[Solaris 10 の概要](#)』を参照してください。

次の表では、Solaris 10 リリースで追加または変更された Solaris 管理ツールについて簡単に説明します。

表 1-1 この Solaris リリースで新たに追加または変更された Solaris 管理ツール

Solaris 管理ツール	説明	詳細
admintool	このツールは使用できなくなりました。 代わりに、次のツールを使用できます。 ■ Solaris 管理コンソール (ユーザーとグループの管理) ■ Solaris Product Registry (ソフトウェアの管理) ■ Solaris 印刷マネージャー (プリンタの管理) ■ Solaris 管理コンソール (端末とモデムの管理)	123 ページ の「ユーザーアカウント の設定 (作業マップ)」 423 ページの「Solaris Product Registry の GUI に よるソフトウェアの管理 (作業マップ)」 『Solaris のシステム管理 (印刷)』の第 4 章「プリ ンタの設定 (作業)」 『Solaris のシステム管理 (上級編)』の「シリアル ポートツールによる端末 とモデムの設定 (概要)」
パッケージツール とパッチツール	Solaris 10 以降のリリースでは、パッケージツール とパッチツールが拡張されています。grep -pattern /var/sadm/install/contents の代わり に、P オプションを指定して pkgchk コマンドを使 用してください。-P オプションでは部分パスが使 えます。	409 ページ の「パッケージおよび パッチツールの拡張」 第 22 章「patchadd コマ ンドによる Solaris パッチ の管理 (手順)」
Solaris 印刷マ ネージャー	Solaris 印刷マネージャーの拡張プリンタサポート には、次の機能があります。これらの機能 は、Solaris 10 リリースで導入されました。 ■ 「バナーを印刷しない」オプション ■ ラスターイメージプロセッサ (RIP) のサポート ■ PostScript プリンタ記述 (PPD) ファイルのサ ポート ■ lpadmin コマンドに新たに -n オプションが追加 されました。これにより、印刷待ち行列を新た に作成したり、既存の印刷待ち行列を変更した りするときに PPD ファイルを指定できます ■ lpstat コマンドの出力に、PPD ファイルを指定 して作成された印刷待ち行列の PPD が表示さ れます	『Solaris のシステム管理 (印刷)』の「印刷の新機 能」

Solaris 管理ツールとサポートされるリリースを示すマトリックス

この節では、主としてユーザー、グループ、クライアント、ディスク、プリンタ、シリアルポートの管理に使われるツールについて説明します。

次の表に、Solaris 管理用の各種 GUI ツールと、それらの現在のサポート状況を示します。

表 1-2 Solaris 管理ツールのサポート状況を示すマトリックス

	Solaris 8	Solaris 9	Solaris 10
admintool	サポートされています	サポートされています	サポートされていません
Solstice AdminSuite 2.3	サポートされていません	サポートされていません	サポートされていません
Solstice AdminSuite 3.0	サポートされています	サポートされていません	サポートされていません
Solaris Management Tools 1.0	サポートされています	サポートされていません	サポートされていません
Solaris Management Tools 2.0	サポートされています (Solaris 8 01/01、4/01、7/01、10/01、2/02 の各リリースのみ)	サポートされていません	サポートされていません
Solaris Management Tools 2.1	サポートされていません	サポートされています	サポートされています

表 1-3 Solaris 8 リリースでの Solaris 管理ツールのサポート状況を示すマトリックス

	Solaris 8
admintool	サポートされています
Solstice AdminSuite 2.3	サポートされていません
Solstice AdminSuite 3.0	サポートされています
Solaris Management Tools 1.0	サポートされています
Solaris Management Tools 2.0	サポートされています (Solaris 8 01/01、4/01、7/01、10/01、2/02 の各リリースのみ)
Solaris Management Tools 2.1	サポートされていません

コンソールにテキストベースの端末を使用しているシステムで管理作業を行う場合は、Solaris 管理コンソールのコマンドを使用してください。詳細は、[表 1-6](#)を参照してください。

Solaris 管理ツールの機能説明

次の表では、Solaris 10 リリースで使用可能なツールについて説明します。

表 1-4 Solaris 管理ツールの説明

機能またはツール	Solaris 管理コンソール 2.1 でのサポート
コンピュータとネットワークツール	サポートされています
ディスクレスクライアントサポート	ディスクレスクライアントのコマンド行インタフェースを使用できます
ディスクツール	サポートされています
拡張ディスクツール (Solaris ボリュームマネージャー)	サポートされています
ジョブスケジューラツール	サポートされています
ログビューアツール	サポートされています
メール別名サポート	サポートされています
マウントと共有ツール	サポートされています
ネームサービスサポート	ユーザー、グループ、ネットワーク情報に対してのみ
パフォーマンスツール	サポートされています
プリンタサポート	サポートされていません。ただし、独立したツールとして Solaris 印刷マネージャーを使用可能です
プロジェクトツール	サポートされています
役割によるアクセス制御 (Role-Based Access Control、RBAC) のサポート	サポートされています
RBAC ツール	サポートされています
シリアルポートツール	サポートされています
ソフトウェアパッケージツール	サポートされていません
システム情報ツール	サポートされています
ユーザー/グループツール	サポートされています

Solaris 9 管理ツールの機能説明

次の表に、Solaris 9 リリースで利用できるツールを示します。

表 1-5 Solaris 9 管理ツールの機能説明

機能またはツール	admintool でのサポート	Solaris 管理コンソール 2.1 でのサポート
コンピュータとネットワークツール	サポートされていません	サポートされています
ディスクレスクライアントサポート	サポートされていません	ディスクレスクライアントのコマンド行インタフェースを使用できます
ディスクツール	サポートされていません	サポートされています
拡張ディスクツール (Solaris ボリュームマネージャー)	サポートされていません	サポートされています
ジョブスケジューラツール	サポートされていません	サポートされています
ログビューアツール	サポートされていません	サポートされています
メール別名サポート	サポートされていません	サポートされています
マウントと共有ツール	サポートされていません	サポートされています
ネームサービスサポート	サポートされていません	ユーザー、グループ、ネットワーク情報に対してのみ
パフォーマンスツール	サポートされていません	サポートされています
プリンタサポート	サポートされています	サポートされていません。ただし、独立したツールとして Solaris 印刷マネージャーを使用可能です
プロジェクトツール	サポートされていません	サポートされています
RBAC サポート	サポートされていません	サポートされています
RBAC ツール	サポートされていません	サポートされています
シリアルポートツール	サポートされています	サポートされています
ソフトウェアパッケージツール	サポートされています	サポートされていません
システム情報ツール	サポートされていません	サポートされています
ユーザー/グループツール	サポートされています	サポートされています

Solaris 管理コマンドの有効性

次の一連の表に、Solaris 管理ツールと同じ作業を実行するコマンドを示します。ディスクレスクライアントサポートについては、第7章「ディスクレスクライアントの管理(手順)」を参照してください。

Solaris 10 システム管理コマンド

次の表に、Solaris 管理ツールと同じ機能を備えたコマンドを示します。これらのコマンドを使用する場合は、スーパーユーザーになるか、それと同等の役割を引き受ける必要があります。コマンドの中には、ローカルシステム専用のものもあります。また、ネームサービス環境で動作するものもあります。それぞれのマニュアルページで、-D オプションを参照してください。

表 1-6 Solaris 管理コマンドの説明

コマンド	説明	マニュアルページ
smc	Solaris 管理コンソールを起動します	smc(1M)
smcron	crontab ジョブを管理します	smcron(1M)
smdiskless	ディスクレスクライアントサポートを管理します	smdiskless(1M)
smexec	exec_attr データベースのエントリを管理します	smexec(1M)
smgroup	グループエントリを管理します	smgroup(1M)
smlog	WBEM ログファイルを管理/表示します	smlog(1M)
smmultiuser	複数のユーザーアカウントに対する一括操作を管理します	smmultiuser(1M)
smosservice	オペレーティングシステム (OS) サービスとディスクレスクライアントサポートを追加します	smosservice(1M)
smprofile	prof_attr および exec_attr データベースのプロファイルを管理します	smprofile(1M)
smrole	役割アカウントの役割とユーザーを管理します	smrole(1M)
smserialport	シリアルポートを管理します	smserialport(1M)

表 1-6 Solaris 管理コマンドの説明 (続き)

コマンド	説明	マニュアルページ
smuser	ユーザーエントリを管理します	smuser(1M)

次の表は、コマンド行から RBAC を管理するためのコマンドです。これらのコマンドを使用する場合は、スーパーユーザーになるか、それと同等の役割を引き受ける必要があります。なお、これらのコマンドは、ネームサービス環境で RBAC 情報を管理する場合には使用できません。

表 1-7 RBAC コマンドの説明

コマンド	説明	参照ページ
auths	ユーザーに付与されている権限を表示します	auths(1)
profiles	ユーザーの実行プロファイルを表示します	profiles(1)
roleadd	システムに新しい役割を追加します	roleadd(1M)
roles	ユーザーに付与されている役割を表示します	roles(1)

次の表は、コマンド行からユーザー、グループ、RBAC 機能を管理するためのコマンドです。これらのコマンドを使用する場合は、スーパーユーザーになるか、それと同等の役割を引き受ける必要があります。なお、これらのコマンドは、ネームサービス環境でユーザーやグループの情報を管理する場合には使用できません。

表 1-8 Solaris ユーザー/グループコマンドの説明

コマンド	説明	参照ページ
useradd, usermod, userdel	ユーザーを追加、変更、削除します	useradd(1M) 、 usermod(1M) 、 userdel(1M)
groupadd, groupmod, groupdel	グループを追加、変更、削除します	groupadd(1M) 、 groupmod(1M) 、 groupdel(1M)

Solaris 管理ツールの詳細情報

次の表に、Solaris 管理ツールの詳細情報の参照先を示します。

表 1-9 Solaris 管理ツールの詳細情報

ツール	対応するリリース	詳細
Solaris 管理コンソール 2.1 のツール群	Solaris 9 および 10 の各リリース	このマニュアルとコンソールのオンラインヘルプ
Solaris 管理コンソール 2.0 のツール群	Solaris 8 1/01、4/01、7/01、10/01、2/02 の各リリース	Solaris 管理コンソールのオンラインヘルプ
admintool	Solaris 9 および以前の Solaris リリース	admintool
AminSuite 3.0	Solaris 8、Solaris 8 6/00、Solaris 8 10/00 の各リリース	『Solaris Easy Access Server 3.0 のインストール』
ディスクレスクリアントのコマンド行インタフェース	Solaris 8 1/01、4/01、7/01、10/01、2/02、Solaris 9、Solaris 10 の各リリース	第 7 章「ディスクレスクリアントの管理 (手順)」

Solaris 管理コンソールの操作 (手順)

この章では、システム管理作業を行うための Solaris 管理ツールについて説明します。この章は、Solaris 管理コンソールの起動、コンソールで使用する役割によるアクセス制御 (RBAC) の設定、ネームサービス環境での Solaris 管理ツールの操作から構成されています。

Solaris 管理コンソールを使用したシステム管理作業の実行手順については、次の作業マップを参照してください。

- 40 ページの「Solaris 管理ツールを RBAC と組み合わせて使用する (作業マップ)」
- 47 ページの「ネームサービス環境で Solaris 管理ツールを使用する (作業マップ)」

Solaris 管理コンソールに関する問題のトラブルシューティングについては、55 ページの「Solaris 管理コンソールのトラブルシューティング」を参照してください。

Solaris 管理コンソール (概要)

この節では、Solaris 管理コンソールについて説明します。

Solaris 管理コンソールについて

Solaris 管理コンソールは、「ツールボックス」というコレクションに格納されている GUI ベースの管理ツール用のコンテナです。

このコンソールには、多数の基本管理ツールが含まれているデフォルトのツールボックスがあります。基本管理ツールには、次のものを管理するためのツールが含まれています。

- ユーザー
- プロジェクト

- ファイルシステムをマウント/共有するための cron ジョブ
- ディスクやシリアルポートを管理するための cron ジョブ

それぞれの Solaris 管理ツールについては、[表 2-1](#) を参照してください。

既存のツールボックスにツールを追加することも、新しいツールボックスを作成することもできます。

Solaris 管理コンソールには、主に次の 3 つの構成要素があります。

- **Solaris** 管理コンソールクライアント
「コンソール」と呼ばれるビジュアルインタフェースで、管理作業を行うための GUI ツールが含まれています。
- **Solaris** 管理コンソールサーバー
コンソールと同じマシン上またはリモートマシン上にあり、コンソールによる管理を可能にする「バックエンド」機能を提供します。
- **Solaris** 管理コンソールツールボックスエディタ
コンソールによく似たアプリケーションで、ツールボックスを追加または変更したり、ツールボックスにツールを追加したり、ツールボックスの適用範囲を広げたりする場合に使用します。たとえば、ネームサービスのドメインを管理するためのツールボックスを追加する場合に使用します。

コンソールを起動すると、デフォルトのツールボックスが表示されます。

Solaris 管理コンソールツール

次の表に、Solaris 管理コンソールのデフォルトのツールボックスに含まれているツールと、各ツールの説明の参照先を示します。

表 2-1 Solaris 管理コンソールツール群

カテゴリ	ツール	説明	詳細
システムの 状態	システム情報	日付、時間、時間帯などのシステム情報を監視および管理します	『Solaris のシステム管理 (上級編)』の第 5 章「システム情報の表示と変更 (手順)」
	ログビューア	Solaris 管理コンソールツールのログとシステムログを監視および管理します	『Solaris のシステム管理 (上級編)』の第 14 章「ソフトウェアの問題解決 (概要)」

表 2-1 Solaris 管理コンソールツール群 (続き)

カテゴリ	ツール	説明	詳細
システムの構成	プロセス数	システムプロセスを監視および管理します	『Solaris のシステム管理 (上級編)』の「プロセスとシステムのパフォーマンス」
	パフォーマンス	パフォーマンスを監視します	『Solaris のシステム管理 (上級編)』の第 11 章「システムパフォーマンスの管理 (概要)」
	ユーザー	ユーザー、権利、役割、グループ、メーリングリストを管理します	92 ページの「ユーザーアカウントとグループとは」および『Solaris のシステム管理 (セキュリティサービス)』の「役割によるアクセス制御 (概要)」
	プロジェクト	/etc/project データベースのエントリを作成および管理します	『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』の第 2 章「プロジェクトとタスク (概要)」
	コンピュータとネットワーク	コンピュータとネットワークの情報を作成および監視します	Solaris 管理コンソールのオンラインヘルプ
サービス	スケジュールされたジョブ	スケジュールされる cron ジョブを作成および管理します	『Solaris のシステム管理 (上級編)』の「システムタスクを自動的に実行する方法」
ストレージ	マウントと共有	ファイルシステムをマウントおよび共有します	『Solaris のシステム管理 (デバイスとファイルシステム)』の第 18 章「ファイルシステムのマウントとマウント解除 (手順)」
	ディスク	ディスクパーティションを作成および管理します	『Solaris のシステム管理 (デバイスとファイルシステム)』の第 10 章「ディスクの管理 (概要)」
	拡張ストレージ	ボリューム、ホットスเปアの集合、状態データベースの複製、ディスクセットを作成および管理します	『Solaris ボリュームマネージャの管理』
デバイスとハードウェア	シリアルポート	端末とモデムを設定します	『Solaris のシステム管理 (上級編)』の第 1 章「端末とモデムの管理 (概要)」

コンテキストヘルプは、ツールの起動後に使用できるようになります。コンテキストヘルプよりも詳細なオンライン情報については、拡張ヘルプトピックを参照してください。拡張ヘルプトピックは、コンソールの「ヘルプ」メニューから利用できます。

Solaris 管理コンソールを使用する理由

このコンソールには、管理者にさまざまなメリットをもたらすツールセットが用意されています。

コンソールは、次の処理を実行します。

- あらゆる経験レベルをサポートする

ダイアログボックス、ウィザード、コンテキストヘルプなどの GUI を使用すれば、経験の浅い管理者でも作業を完了することができます。また、経験を積んだ管理者であれば、何十または何百にも及ぶシステム上に分散している何百もの構成パラメータを管理する場合に、vi よりもこのコンソールを使用した方が便利で安全性が高いことに気がつきます。

- システムへのユーザーアクセスを管理する

デフォルトではどのユーザーもこのコンソールにアクセスできますが、初期設定を変更できるのはスーパーユーザーに限られます。『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[役割によるアクセス制御 \(概要\)](#)」に説明されているように、「役割」と呼ばれる特別なユーザーアカウントを作成して、特定のシステム変更を許可されているユーザー (通常は管理者) に割り当てることができます。

RBAC の主なメリットは、ユーザーがジョブの実行に必要な作業にしかアクセスできないように役割を制限できることです。Solaris 管理ツールを使用する場合に、RBAC は必須ではありません。何も変更せずスーパーユーザーとしてすべてのツールを実行できます。

- コマンド行インタフェースを提供する

管理者は、必要に応じてコマンド行インタフェース (CLI) から Solaris 管理ツールを操作することができます。コマンドの中には、ユーザーを管理するコマンドなどの、特に GUI ツールの機能をまねて作られたものもあります。表 1-6 に、これらの新しいコマンドの名前と簡単な説明が記載されています。各コマンドのマニュアルページもあります。

専用のコマンドを持たない Solaris 管理ツール (マウントツールや共有ツールなど) では、標準の UNIX コマンドを使用します。

RBAC の機能、メリット、ユーザーサイトでのそれらのメリットの活かし方については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[役割によるアクセス制御 \(概要\)](#)」を参照してください。

RBAC を Solaris 管理ツールと組み合わせて使用する方法については、[40 ページ](#)の「[Solaris 管理ツールを RBAC と組み合わせて使用する \(作業マップ\)](#)」を参照してください。

Solaris 管理コンソールの構成

次の図は、ユーザーツールが開いた状態でのコンソールです。



図 2-1 Solaris 管理コンソール - ユーザーツール

コンソールの主要部分は、次の 3 つの区画で構成されます。

- ナビゲーション区画 (左側) - ツール (ツールセット)、フォルダ、ほかのツールボックスへのアクセス用。ナビゲーション区画内にあるアイコンは「ノード」と呼ばれ、それらがフォルダやツールボックスである場合は拡張可能です。
- 表示区画 (右側) - ナビゲーション区画で選択したノードに関する情報の表示用。選択したフォルダの内容、従属ツール、選択したツールに関連付けられたデータが表示されます。
- 情報区画 (下側) - コンテキストヘルプまたはコンソールイベントの表示用。

Solaris 管理コンソールウィンドウの変更

コンソールウィンドウのレイアウトは、非常に柔軟に設定することができます。次の機能を使用して、コンソールウィンドウのレイアウトを変更できます。

- 「表示 (View)」メニュー – 「表示 (View)」メニューの「表示部分 (Show)」オプションを使用すると、オプションのバーと区画を表示または非表示にすることができます。「表示 (View)」メニューのほかのオプションは、表示区画内にあるノードの表示を制御します。
- 「コンソール (Console)」メニュー – 「設定の変更 (Preferences)」オプションを使用すると、次の設定ができます。初期ツールボックス、区画の向き、選択肢のクリックまたはダブルクリック、ツールバーのテキストまたはアイコン、フォント、デフォルトのツールロード、認証プロンプト、拡張ログイン。
- 「コンテキストヘルプ (Context Help)」と「コンソールイベント (Console Events)」のトグル – 情報区画の一番下にあるアイコンを使用すると、コンテキストヘルプとコンソールイベントの表示を切り替えることができます。

Solaris 管理コンソールのマニュアル

コンソールやそのツールの使用方法に関するマニュアルは、主にオンラインヘルプシステムから利用できます。オンラインヘルプは、コンテキストヘルプと拡張ヘルプトピックの2つの形式で利用できます。

- コンテキストヘルプは、コンソールツールの使用状況に対応しています。
タブ、入力フィールド、ラジオボタンなどでカーソルをクリックすると、該当するヘルプが情報区画に表示されます。情報区画を閉じたり、開き直したりするには、ダイアログボックスやウィザードの疑問符ボタンをクリックします。
- 拡張ヘルプトピックは、「ヘルプ (Help)」メニューから使用可能であり、一部のコンテキストヘルプから相互参照リンクをクリックしても使用できます。
これらのトピックは個別のビューアに表示され、コンテキストヘルプよりも詳細な情報が含まれています。具体的には、各ツールの概要、各ツールの機能説明、特定のツールで使用されるファイル、トラブルシューティングなどです。

各ツールの概要については、[表 2-1](#) を参照してください。

役割によるアクセス制御の程度

34 ページの「[Solaris 管理コンソールを使用する理由](#)」に説明されているように、Solaris 管理ツールを使用する主な利点は、役割によるアクセス制御 (RBAC) を使用できることです。RBAC を使用すると、管理者はジョブの実行に必要なツールとコマンドだけを使用できます。

組織のセキュリティ要件に応じて、さまざまなレベルの RBAC を使用できます。

RBAC の方法	説明	詳細
RBAC を使用しない	すべての作業をスーパーユーザーとして実行できます。ユーザーとしてログインできます。Solaris 管理ツールを選択するときは、ユーザーとして root を指定し、root のパスワードを入力します。	39 ページの「スーパーユーザー (root) になるか役割を引き受ける方法」
root を役割とする	匿名の root ログインを削除し、ユーザーが root としてログインできないようにします。この方法では、ユーザーが root の役割を持つ前にユーザー自身としてログインする必要があります。 この方法はほかの役割を使用しているかどうかに関係なく適用できるので注意が必要です。	『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の実装を計画する方法」
単一の役割のみ	プライマリ管理者の役割を使用します。これは、root アクセス権だけを持つ場合とほぼ同じです。	42 ページの「プライマリ管理者の役割を作成する」

RBACの方法	説明	詳細
推奨される役割	簡単に設定できる3つの役割(プライマリ管理者、システム管理者、オペレータ)を使用します。これらの役割は、責任の異なるレベルの管理者がいて、それぞれのジョブ機能が推奨される役割にほぼ合っている組織に適しています。	『Solarisのシステム管理(セキュリティサービス)』の「役割によるアクセス制御(概要)」
カスタムの役割	組織のセキュリティの必要性に応じて、独自の役割を追加できます。	『Solarisのシステム管理(セキュリティサービス)』の「RBACの管理」および『Solarisのシステム管理(セキュリティサービス)』の「RBACの実装を計画する方法」

スーパーユーザー (root) になるか役割を引き受ける

ほとんどの管理作業(ユーザー、ファイルシステム、プリンタの追加など)では、まず root (UID=0) としてログインするか、役割を引き受ける (RBAC を使用している場合) 必要があります。root アカウント (「スーパーユーザーアカウント」とも呼ばれる) は、システムを変更したり、緊急時にユーザーファイルの保護を無効にしたりする場合に使用します。

システムの変更がむやみに行われないように、スーパーユーザーアカウントと役割は、管理作業を実行するためだけに使用してください。スーパーユーザーアカウントに関連するセキュリティ問題は、小さな作業を行うときにでもユーザーがシステムへの完全なアクセス権を持つてしまうことにあります。

RBAC が実装されていない環境では、スーパーユーザーとしてシステムにログインすることも、su コマンドを使ってスーパーユーザーアカウントに変更することもできます。RBAC が実装されている場合は、コンソールを介して役割を引き受けることも、su を使用して役割を指定することもできます。

コンソールを使用して管理作業を行う場合は、次のいずれかを実行できます。

- ユーザーとしてコンソールにログインし、root ユーザー名とパスワードを入力する
- ユーザーとしてコンソールにログインし、役割を引き受ける

RBAC の大きなメリットは、特定の機能しか使用できないように役割を作成できることです。RBAC を使用している場合は、スーパーユーザーになるのではなく役割を引き受けることで、制限付きのアプリケーションを実行できます。

プライマリ管理者の役割の作成手順については、[43 ページの「最初の役割 \(プライマリ管理者\) を作成する方法」](#)を参照してください。RBAC の使用方法の概要については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の第 9 章「[役割によるアクセス制御の使用 \(手順\)](#)」を参照してください。

▼ スーパーユーザー (root) になるか役割を引き受ける方法

スーパーユーザーになるか役割を引き受ける場合は、次のいずれかの方法を使用します。各方法では、スーパーユーザーのパスワードまたは役割のパスワードを知っている必要があります。

- 1 スーパーユーザーになります。スーパーユーザーになる場合は、次のいずれかの方法を選択します。

- ユーザーとしてログインし、**Solaris** 管理コンソールを起動し、**Solaris** 管理ツールを選択して root としてログインする方法。

この方法では、コンソールから任意の管理作業を実行できます。

Solaris 管理コンソールの起動方法については、[53 ページの「ネームサービス環境で Solaris 管理コンソールを起動する方法」](#)を参照してください。

- システムコンソールでスーパーユーザーとしてログインする方法。

```
hostname console: root
Password: root-password
#
```

ポンド記号 (#) は、Bourne シェルにおける、スーパーユーザーアカウント用のプロンプトです。

この方法では、すべてのシステムコマンドとツールに完全にアクセスできます。

- ユーザーとしてログインし、コマンド行で su コマンドを実行してスーパーユーザーアカウントに変更する方法。

```
% su
Password: root-password
#
```

この方法では、すべてのシステムコマンドとツールに完全にアクセスできます。

- スーパーユーザーとしてリモートでログインする方法。

この方法は、デフォルトでは使用できません。システムコンソールの

スーパーユーザーとしてリモートログインするには、`/etc/default/login` ファイ

ルを変更する必要があります。このファイルの変更方法については、『Solaris のシステム管理 (セキュリティサービス)』の第 3 章「システムアクセスの制御 (作業)」を参照してください。

この方法では、すべてのシステムコマンドとツールに完全にアクセスできます。

2 役割になります。役割を引き受ける場合は、次のいずれかの方法を選択します。

- ユーザーとしてログインし、コマンド行で `su` コマンドを実行して役割に変更する方法。

```
% su role
Password: role-password
$
```

この方法では、設定した役割がアクセスできるすべてのコマンドとツールを使用できます。

- ユーザーとしてログインし、Solaris 管理コンソールを起動し、Solaris 管理ツールを選択して、役割を引き受ける方法。

Solaris 管理コンソールの起動方法については、45 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」を参照してください。

この方法では、引き受けた役割がアクセスできる Solaris 管理ツールを使用できます。

Solaris 管理ツールを RBAC と組み合わせて使用する (作業マップ)

この作業マップでは、スーパーユーザーアカウントではなく RBAC のセキュリティ機能を使用して管理作業を実行する場合に行う作業について説明します。

注 - この節の内容は、コンソールを RBAC と組み合わせて使用方法について書かれています。最初にコンソールを使って RBAC を設定する方法について説明するため、RBAC の概要や作業にも触れています。

RBAC の詳細や、RBAC をほかのアプリケーションと組み合わせて使用方法については、『Solaris のシステム管理 (セキュリティサービス)』の「役割によるアクセス制御 (概要)」を参照してください。

作業	説明	参照先
1. コンソールを起動します。	ユーザーアカウントをすでに設定してある場合は、まずユーザーとしてコンソールを起動し、次に <code>root</code> としてコンソールにログインします。ユーザーアカウントを設定していない場合は、まずスーパーユーザーになり、次にコンソールを起動します。	45 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」
2. 自分のユーザーアカウントを追加します。	アカウントが存在しない場合は、自分のユーザーアカウントを追加します。	Solaris 管理コンソールのオンラインヘルプ 41 ページの「コンソールに最初にログインした場合」
3. プライマリ管理者の役割を作成します。	プライマリ管理者の役割を作成し、自分をこの役割に追加します。	43 ページの「最初の役割 (プライマリ管理者) を作成する方法」
4. プライマリ管理者の役割を引き受けます。	プライマリ管理者の役割を作成後、その役割を引き受けます。	44 ページの「プライマリ管理者の役割を引き受ける方法」
5. (省略可能) <code>root</code> を役割にします。	<code>root</code> を役割にし、ほかのユーザーが <code>su</code> コマンドを使用して <code>root</code> にならないように <code>root</code> の役割に自分を追加します。	『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の実装を計画する方法」
6. (省略可能) ほかの管理役割を作成します。	ほかの管理役割を作成し、各役割に適切な権利を付与します。次に、各役割に該当するユーザーを追加します。	『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」

次の節では、Solaris 管理コンソールと RBAC のセキュリティ機能の使い方に関する概要とその手順について説明します。

コンソールに最初にログインした場合

管理者としてコンソールに最初にログインした場合は、まずユーザー (自分自身) としてコンソールを起動します。次に、スーパーユーザーとしてログインします。この方法では、コンソールのすべてのツールに完全にアクセスできます。

ここで、RBAC を使用しているかどうかに応じて、一般的な手順を示します。

- **RBAC を使用しない** – RBAC を使用しない場合は、スーパーユーザーとして作業を続けます。ほかのすべての管理者も、ジョブを実行するのに root アクセス権が必要になります。
- **RBAC を使用する** – 次の手順を実行する必要があります。
 - アカウントがない場合は、ユーザーアカウントを設定します。
 - プライマリ管理者という役割を作成します。
 - 作成中の役割にプライマリ管理者の権利を割り当てます。
 - この役割にユーザーアカウントを割り当てます。

プライマリ管理者の役割の作成手順については、[43 ページの「最初の役割 \(プライマリ管理者\) を作成する方法」](#)を参照してください。

RBAC の使用方法の概要については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「[役割によるアクセス制御の使用 \(手順\)](#)」を参照してください。

プライマリ管理者の役割を作成する

「管理役割」は、特殊なユーザーアカウントの 1 つです。この役割を引き受けたユーザーは、定義済みの管理作業を実行することができます。

プライマリ管理者の役割は、スーパーユーザーと同様に、すべての管理機能の実行が許可されています。

スーパーユーザー、またはプライマリ管理者の役割を引き受けたユーザーは、ほかの管理者が実行できる作業を定義することができます。「管理役割を追加」ウィザードを使用すると、役割を作成し、その役割に権利を付与し、その役割を引き受けられるユーザーを指定できます。「権利」とは、特定のアプリケーションを使用するためのコマンド (または承認) の集合に、名前を付けたものです。権利を使用すると、アプリケーション内にある特定の機能を実行できます。権利の使用は、管理者が付与または拒否できます。

プライマリ管理者の役割を作成するときは、次の情報の入力を求めるプロンプトが表示されます。

表 2-2 Solaris 管理コンソールを使用して役割を追加するためのフィールドの説明

フィールド名	説明
役割名	管理者が特定の役割にログインするために使用する名前を選択します。

表 2-2 Solaris 管理コンソールを使用して役割を追加するためのフィールドの説明 (続き)

フィールド名	説明
正式名	この役割の名前をフルネームでわかりやすく入力します。(省略可能)
説明	この役割の詳細な説明を入力します。
役割 ID 番号	この役割に割り当てられている ID 番号を選択します。この番号は、UID の ID セットと同じです。
役割のシェル	ユーザーが端末またはコンソールのウィンドウにログインするか、そのウィンドウで役割を引き受けるときに実行するシェルを選択します。
役割のメーリングリストを作成	項目をチェックすると、役割と同じ名前でメーリングリストを作成します。メーリングリストを使用すると、その役割に割り当てられているすべてのユーザーに電子メールを送信できます。
役割パスワードとパスワードを確認	役割のパスワードを設定および再入力します。
有効な権利と許可された権利	「有効な権利」のリストから権利を選択し、「許可された権利」のリストに追加することにより、この役割に権利を割り当てます。
ホームディレクトリの選択	この役割の専有ファイルが格納されるホームディレクトリサーバーを選択します。
ユーザーの役割への割り当て	特定のユーザーが特定の作業を行うための役割を持てるようにユーザーを役割に追加します。

役割によるアクセス制御の詳細と、役割を使用してよりセキュリティ保護された環境を作成する方法については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[役割によるアクセス制御 \(概要\)](#)」を参照してください。

▼ 最初の役割(プライマリ管理者)を作成する方法

この手順では、プライマリ管理者の役割を作成し、それをユーザーアカウントに割り当てる方法について説明します。ユーザーアカウントはすでに作成してあるものとしてします。

1 ユーザーとしてコンソールを起動します。

```
% /usr/sadm/bin/smc &
```

コンソールの起動方法については、[45 ページ](#)の「[スーパーユーザーまたは役割としてコンソールを起動する方法](#)」を参照してください。

ユーザーアカウントの作成方法については、コンソールのオンラインヘルプを参照してください。

- 2 ナビゲーション区画で「このコンピュータ」アイコンをクリックします。
 - 3 「システムの構成」→「ユーザー」→「管理役割」の順にクリックします。
 - 4 「アクション」→「管理役割を追加」の順にクリックします。
「管理役割を追加」ウィザードが開きます。
 - 5 次の手順に従って、「管理役割を追加」ウィザードでプライマリ管理者の役割を作成します。
 - a. 役割名、役割の正式名、備考欄、役割 ID 番号、役割シェル、役割のメーリングリストを作成するかどうかを指定します。「次へ (Next)」をクリックします。
 - b. 役割のパスワードを設定し、確認します。「次へ (Next)」をクリックします。
 - c. 「有効な権利」欄からプライマリ管理者の権利を選択後、「追加」をクリックし、「許可された権利」欄に追加します。「次へ (Next)」をクリックします。
 - d. 役割のホームディレクトリを選択します。「次へ (Next)」をクリックします。
 - e. この役割を引き受けることができるユーザーのリストに自分を割り当てます。「次へ (Next)」をクリックします。
- 必要に応じて、[表 2-2](#) で役割の各フィールドの説明を参照してください。
- 6 「完了」をクリックします。

▼ プライマリ管理者の役割を引き受ける方法

プライマリ管理者の役割を作成し終わったら、まずユーザー (自分自身) としてコンソールにログインし、次にプライマリ管理者の役割を引き受けます。

役割を引き受けるときは、その役割の権利を含むすべての属性を引き受けます。同時に、自分自身のユーザープロパティはすべて放棄します。

- 1 コンソールを起動します。

```
% /usr/sadm/bin/smc &
```

コンソールの起動方法については、[45 ページ](#)の「スーパーユーザーまたは役割としてコンソールを起動する方法」を参照してください。
- 2 ユーザー名とパスワードを使ってログインします。
引き受けることができる役割のリストが表示されます。

- 3 プライマリ管理者の役割にログインし、役割のパスワードを入力します。

Solaris 管理コンソールを起動する

次の手順では、コンソールを起動し、Solaris 管理ツールにアクセスする方法について説明します。

コンソールに最初にログインするユーザーが実行する手順については、[41 ページ](#)の「[コンソールに最初にログインした場合](#)」を参照してください。

▼ スーパーユーザーまたは役割としてコンソールを起動する方法

自分自身のユーザーアカウントを使ってユーザーとしてコンソールを起動する場合は、Solaris 管理ツールへのアクセスが制限されます。十分なアクセス権を得るために、まずユーザー (自分自身) としてログインし、次に引き受けることができる役割のいずれかでログインできます。プライマリ管理者の役割を引き受けることができる場合は、すべての Solaris 管理ツールにアクセスできます。この役割は、スーパーユーザーの役割と同等です。

- 1 CDE 環境などのウィンドウ環境で操作していることを確認します。
- 2 次のいずれかの方法でコンソールを起動します。
 - コマンド行から次のコマンドを入力します。

```
% /usr/sadm/bin/smc &
```

コンソールが初めて起動するときは12分かかります。
 - CDE フロントパネルの「ツール」メニューから起動します。
 - CDE のアプリケーションマネージャまたはファイルマネージャの「Solaris 管理コンソール」アイコンをダブルクリックします。
「Solaris 管理コンソール」ウィンドウが表示されます。

注 - Solaris 管理コンソールのスタートアップメッセージを表示する場合は、自分のウィンドウ環境でコンソールを開いてください。Solaris 管理コンソールを起動する前に、Solaris 管理コンソールサーバーを手動で起動しないでください。このサーバーは、Solaris 管理コンソールを起動すると、自動的に起動します。コンソールに関する問題のトラブルシューティングについては、[55 ページの「Solaris 管理コンソールのトラブルシューティング」](#)を参照してください。

- 3 ナビゲーション区画の「管理ツール」アイコンの下にある「このコンピュータ」アイコンをダブルクリックします。
カテゴリのリストが表示されます。
- 4 (省略可能)適切なツールボックスを選択します。
デフォルト以外のツールボックスを使用する場合は、ナビゲーション区画から該当するツールボックスを選択します。あるいは、コンソールメニューから「ツールボックスを開く」を選択し、任意のツールボックスをロードします。

各種ツールボックスの使用方法については、[50 ページの「特定環境用のツールボックスを作成する方法」](#)を参照してください。
- 5 カテゴリのアイコンをダブルクリックして、特定のツールにアクセスします。
特定の作業の実行方法を確認する場合は、オンラインヘルプを使用します。
- 6 ツールのアイコンをダブルクリックします。
「ログイン」ポップアップウィンドウが表示されます。
- 7 スーパーユーザーまたは役割のどちらでこのツールを使用するかを決めます。スーパーユーザーとしてログインしている場合は、root のパスワードを入力します。
- 8 ユーザー (自分自身) としてログインしている場合は、バックスペースキーを使って root のユーザー名を削除します。次に、適切なユーザー ID とユーザーパスワードを入力します。
ユーザーが引き受けることができる役割のリストが表示されます。
- 9 プライマリ管理者の役割、またはそれと同等の役割を選択し、役割のパスワードを入力します。
プライマリ管理者の役割の作成手順については、[43 ページの「最初の役割 \(プライマリ管理者\) を作成する方法」](#)を参照してください。
ツールのメインメニューが表示されます。

ネームサービス環境で **Solaris** 管理ツールを使用する (作業マップ)

デフォルトでは、Solaris 管理ツールはローカル環境で動作するように設定されます。たとえば、マウントと共有ツールを使用すると、特定のシステム上でディレクトリをマウントおよび共有できますが、NIS や NIS+ 環境ではできません。ただし、ネームサービス環境では、ユーザーツールやコンピュータとネットワークツールを使って情報を管理できます。

ネームサービス環境でコンソールのツールを使用する場合は、まずネームサービスのツールボックスを作成し、次にそのツールボックスにツールを追加する必要があります。

作業	説明	参照先
1. 前提条件を確認します。	ネームサービス環境でコンソールを使用する前に、前提条件が満たされていることを確認します。	49 ページの「ネームサービス環境で Solaris 管理コンソールを使用するための前提条件」
2. ネームサービス用のツールボックスを作成します。	新規ツールボックスウィザードを使用して、ネームサービスツール用のツールボックスを作成します。	50 ページの「特定環境用のツールボックスを作成する方法」
3. ネームサービスツールボックスにツールを追加します。	作成したネームサービスツールボックスにユーザーツール、またはほかのネームサービスツールを追加します。	52 ページの「ツールボックスにツールを追加する方法」
4. 直前に作成したツールボックスを選択します。	直前に作成したツールボックスを選択して、ネームサービス情報を管理します。	53 ページの「ネームサービス環境で Solaris 管理コンソールを起動する方法」

RBAC セキュリティーファイル

Solaris 管理コンソールで使用する RBAC セキュリティーファイルは、Solaris 9 以降のリリースへのアップグレードまたはそのインストール時に作成されます。Solaris 管理コンソールのパッケージをインストールしない場合、RBAC を使用するのに必要なデータがない状態で RBAC セキュリティーファイルがインストールされます。Solaris 管理コンソールのパッケージについては、[55 ページの「Solaris 管理コンソールのトラブルシューティング」](#)を参照してください。

Solaris 9 以降のリリースを実行している場合、RBAC セキュリティーファイルは、ネームサービス環境で Solaris 管理コンソールのツールを使用できるように、ネームサービスに組み込まれます。

ローカルサーバー上のセキュリティーファイルは、標準アップグレードの一環として、ypmake コマンド、nispopulate コマンド、または同様の LDAP コマンドによってネームサービス環境に作成されます。

サポートされているネームサービスは次のとおりです。

- NIS
- NIS+
- LDAP
- files

注 - NIS+ 環境では projects データベースはサポートされていません。

RBAC セキュリティーファイルは、Solaris 9 以降のリリースへのアップグレードまたはそのインストール時に作成されます。

次の表では、Solaris 9 以降のリリースが稼働するシステムにインストールされる、定義済みのセキュリティーファイルについて簡単に説明します。

表 2-3 RBAC セキュリティーファイル

ローカルファイル名	テーブルまたはマップ名	説明
/etc/user_attr	user_attr	ユーザーと役割を承認と権利プロファイルに関連付けます
/etc/security/auth_attr	auth_attr	承認とその属性を定義し、関連付けられたヘルプファイルを識別します
/etc/security/prof_attr	prof_attr	権利プロファイルを定義し、承認に割り当てられた権利プロファイルのリストを表示し、関連付けられたヘルプファイルを識別します
/etc/security/exec_attr	exec_attr	権利プロファイルに割り当てられている特権付きの操作を定義します

アップグレードに失敗した場合は、次のときに smattrpop コマンドで RBAC セキュリティーファイルを作成してください。

- 権利プロファイルを作成または変更するとき
- usr_attr ファイルをカスタマイズして、ユーザーと役割を追加する必要があるとき

詳細は、『Solaris のシステム管理 (セキュリティサービス)』の「役割によるアクセス制御 (概要)」を参照してください。

ネームサービス環境で Solaris 管理コンソールを使用するための前提条件

次の表に、ネームサービス環境で Solaris 管理コンソールを使用する前に行う必要のある操作を示します。

前提条件	詳細
Solaris 9 以降のリリースをインストールします。	『Solaris 10 インストールガイド (基本編)』
ネームサービス環境を設定します。	『Solaris のシステム管理 (ネーミングとディレクトリサービス:DNS、NIS、LDAP 編)』
管理範囲を選択します。	49 ページの「管理範囲」
ネームサービスデータにアクセスできるように /etc/nsswitch.conf ファイルが構成されていることを確認します。	49 ページの「/etc/nsswitch.conf ファイル」

管理範囲

Solaris 管理コンソールでは、選択した管理ツールで使用するネームサービス環境を指すときに「管理範囲」という用語を使用します。ユーザーツールおよびコンピュータとネットワークツールの管理範囲は、LDAP、NIS、NIS+、filesの中から選択します。

コンソールセッション時に選択する管理範囲は、/etc/nsswitch.conf ファイルで識別される一次ネームサービスに一致させるべきです。

/etc/nsswitch.conf ファイル

各システムの /etc/nsswitch.conf ファイルは、そのシステムのネームサービス検索 (ここからデータが読み取られる) のポリシーを示します。

注- コンソールからアクセスされるネームサービス (コンソールのツールボックスエディタで指定する) が /etc/nsswitch.conf ファイルの検索パスに含まれていることを確認する必要があります。指定のネームサービスがその検索パスにない場合、ツールが予測のつかない動作をしてエラーまたは警告が発生する可能性があります。

ネームサービス環境で Solaris 管理ツールを使用するときは、1 回の操作で多数のユーザーに影響を及ぼす可能性があります。たとえば、NIS ネームサービスのユーザーを削除すると、そのユーザーは NIS を使用しているすべてのシステムで削除されます。

ネットワーク内のさまざまなシステムで `/etc/nsswitch.conf` 構成が異なっていると、予期しない結果が生じる可能性があります。これを回避するために、Solaris 管理ツールで管理するすべてのシステムには、一貫したネームサービス構成を設定する必要があります。

▼ 特定環境用のツールボックスを作成する方法

Solaris オペレーティングシステムを管理するためのアプリケーションは「ツール」と呼ばれます。これらのツールは「ツールボックス」というコレクションに格納されます。ツールボックスは、ローカルサーバー (コンソールがある場所) にもリモートマシンにも配置できます。

ツールボックスエディタを使用すると、新しいツールボックスの追加、既存のツールボックスへのツールの追加、またはツールボックスの適用範囲の変更を行うことができます。たとえば、このツールを使ってドメインをローカルファイルからネームサービスに変更できます。

注-ツールボックスエディタは、通常のユーザーとして起動できます。ただし、変更を加えたり、変更内容をデフォルトのコンソールツールボックス `/var/sadm/smc/toolboxes` に保存する場合は、`root` としてツールボックスエディタを起動する必要があります。

- 1 ツールボックスエディタを起動します。
`# /usr/sadm/bin/smc edit &`
- 2 「ツールボックス」メニューから「開く」を選択します。
- 3 「ツールボックスを開く」ウィンドウで「このコンピュータ」アイコンを選択します。ウィンドウで[ホストの保守]リンクをクリックします。
- 4 「開く」をクリックします。
「このコンピュータ」ツールボックスがウィンドウに開きます。
- 5 ナビゲーション区画で「このコンピュータ」アイコンを再度選択します。
- 6 「アクション」メニューから「フォルダを追加」を選択します。

- 7 「フォルダウィザード」を使用して、ネームサービス環境用の新しいツールボックスを追加します。
 - a. 名前 (Name) と説明 (Description) – 「名前 (Full Name)」ウィンドウに名前を入力します。「次へ (Next)」をクリックします。
たとえば、NIS 環境の場合は「NIS tools」と入力します。
 - b. 「説明」ウィンドウに説明を入力します。「次へ (Next)」をクリックします。
たとえば、「tools for NIS environment」と入力します。
 - c. アイコン (Icons) – デフォルト値を使用します。「次へ (Next)」をクリックします。
 - d. 管理範囲 (Management Scope) – 「上書き (Override)」を選択します。
 - e. 「管理範囲」プルダウンメニューから該当するネームサービスを選択します。
 - f. 必要に応じて、ネームサービスのマスター名を「サーバー」フィールドに追加します。
 - g. サーバーが管理しているドメインを「ドメイン」フィールドに追加します。
 - h. 「完了」をクリックします。
左側のナビゲーション区画に新しいツールボックスが表示されます。
- 8 新しいツールボックスのアイコンを選択し、「ツールボックス」メニューから「別名保存」を選択します。
- 9 「ローカルツールボックス」ダイアログボックスの「フォルダ」からフォルダを選択し、「ファイル名」にツールボックスのファイル名を入力します。接尾辞 .tbx を使用します。
`/var/sadm/smc/toolboxes/this_computer/toolbox-name.tbx`
- 10 「Save」をクリックします。
コンソールウィンドウのナビゲーション区画に新しいツールボックスが表示されます。

参照 ネームサービスツールボックスを作成し終わったら、その中にネームサービスを入れることができます。詳細は、[52 ページの「ツールボックスにツールを追加する方法」](#)を参照してください。

▼ ツールボックスにツールを追加する方法

コンソールに添付されているデフォルトのツールのほかに、コンソールから起動できるツールがいくつか開発されています。これらのツールが使用可能になったら、既存のツールボックスに追加できます。

また、ローカル管理またはネットワーク管理用に新しくツールボックスを作成し、その新しいツールボックスにツールを追加することもできます。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 必要に応じて、ツールボックスエディタを起動します。

```
# /usr/sadm/bin/smc edit &
```

- 3 ツールボックスを選択します。

ネームサービス環境で作業する場合は、直前に作成したツールボックスをツールボックスエディタで選択します。詳細は、[50 ページの「特定環境用のツールボックスを作成する方法」](#)を参照してください。

- 4 「アクション」メニューから「ツールを追加」を選択します。

- 5 「ツールを追加」ウィザードを使用して新しいツールを追加します。

- a. 「サーバーの選択」-ネームサービスのマスターを「サーバー」ウィンドウに追加します。「次へ (Next)」をクリックします。

- b. 「ツールの選択」-追加するツールを「ツール」ウィンドウから選択します。「次へ (Next)」をクリックします。

このツールボックスがネームサービス用のツールボックスである場合は、ネームサービス環境で作業するツールを選択します。たとえば、ユーザーツールを選択します。

- c. 「ツールクラス名 (Name) と説明 (Description)」-デフォルト値を使用します。「次へ (Next)」をクリックします。

- d. 「アイコン (Icons)」-カスタムアイコンを作成していない場合は、デフォルト値を使用します。「次へ (Next)」をクリックします。

- e. 「管理範囲 (Management Scope)」-デフォルト値「親から継承 (Inherit from Parent)」を使用します。「次へ (Next)」をクリックします。

- f. 「ツールの読み込み (Tool Loading)」 - デフォルト値「選択された時にツールを読み込む (Load tool when selected)」を使用します。「完了」をクリックします。
- 6 「ツールボックス」メニューから「保存」を選択して、更新したツールボックスを保存します。
「ローカルツールボックス」ウィンドウが表示されます。

▼ ネームサービス環境で Solaris 管理コンソールを起動する方法

ネームサービスのツールボックスを作成し、そのツールボックスにツールを追加し終わったら、Solaris 管理コンソールを起動し、作成したツールボックスを開いてネームサービス環境を管理できます。

始める前に 次の前提条件を満たしていることを確認します。

- ログインしているシステムがネームサービス環境で動作するように設定されているかを確認します。
- /etc/nsswitch.conf ファイルがユーザーのネームサービス環境に適合するように設定されているかを確認します。

1 Solaris 管理コンソールを起動します。

詳細は、[45 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」](#)を参照してください。

2 ネームサービス用に作成したツールボックスを選択します。選択したツールボックスがナビゲーション区画に表示されます。

ネームサービス用のツールボックスの作成方法については、[50 ページの「特定環境用のツールボックスを作成する方法」](#)を参照してください。

Solaris 管理コンソールにツールを追加する

この節では、レガシーツールまたは別製品のツールをコンソールに追加する方法について説明します。これらのツールに認証を追加する場合は、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の管理](#)」を参照してください。

▼ レガシーツールをツールボックスに追加する方法

レガシーツールとは、特に Solaris 管理ツールとして設計されていないアプリケーションのことです。コンソールのツールボックスには、X アプリ

ケーション、コマンド行インタフェース、HTML の 3 種類のレガシーツールアプリケーションを追加できます。ツールボックスに追加した各ツールは、Solaris 管理コンソールから起動できます。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
- 2 必要に応じて、**Solaris** 管理コンソールのツールボックスエディタを起動します。
`# /usr/sadm/bin/smc edit &`
- 3 レガシーアプリケーションを追加するツールボックスを開きます。
選択したツールボックスはツールボックスエディタで開きます。
- 4 レガシーアプリケーションを追加するツールボックス内のノードを選択します。
レガシーアプリケーションは、ツールボックスの最上位のノードにも別のフォルダにも追加できます。
- 5 「アクション」→「従来のアプリケーションを追加」の順にクリックします。
「従来のアプリケーションのウィザード」の最初のパネル (基本) が表示されます。
- 6 ウィザードの指示に従います。
- 7 ツールボックスエディタでツールボックスを保存します。

▼ 別製品のツールをインストールする方法

Solaris 管理コンソールから起動できるツールパッケージを新たに追加する場合は、次の手順に従います。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
- 2 新しいツールパッケージをインストールします。
`# pkgadd ABCDtool`
- 3 コンソールを再起動して、コンソールに新しいツールを認識させます。
 - a. コンソールサーバーを停止します。
`# /etc/init.d/init.wbem stop`
 - b. コンソールサーバーを起動します。
`# /etc/init.d/init.wbem start`

- 4 コンソールを起動して、新しいツールが表示されることを確認します。
詳細は、[45 ページ](#)の「スーパーユーザーまたは役割としてコンソールを起動する方法」を参照してください。

Solaris 管理コンソールのトラブルシューティング

このトラブルシューティングの手順を使用する前に、次のパッケージがインストールされていることを `pkginfo(1)` コマンドで確認してください。

- SUNWmc – Solaris 管理コンソール 2.1 (サーバー構成要素)
- SUNWmcc – Solaris 管理コンソール 2.1 (クライアント構成要素)
- SUNWmccom – Solaris 管理コンソール 2.1 (共通の構成要素)
- SUNWmcdev – Solaris 管理コンソール 2.1 (開発キット)
- SUNWmcex – Solaris 管理コンソール 2.1 (例)
- SUNWwbmc – Solaris 管理コンソール 2.1 (WBEM 構成要素)

これらのパッケージには、基本的な Solaris 管理コンソール起動ツールが用意されています。Solaris 管理コンソールとそのすべてのツールを使用するには、SUNWCprog クラスタをインストールする必要があります。

▼ Solaris 管理コンソールのトラブルシューティングの方法

Solaris 管理コンソールを起動すると、クライアントとサーバーが自動的に起動します。

コンソールが表示可能で、ツールの実行に問題がある場合は、サーバーが実行していない可能性があります。あるいは、サーバーがエラー状態にあり、いったん停止してから再起動することによって解決できる可能性もあります。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
- 2 次のコマンドでコンソールサーバーが稼働しているかどうかを調べます。

```
# /etc/init.d/init.wbem status
```

コンソールサーバーが稼働している場合は、次のようなメッセージが表示されます。

```
SMC server version 2.1.0 running on port 898.
```

- 3 コンソールサーバーが稼働していない場合は、次のコマンドで起動します。

```
# /etc/init.d/init.wbem start
```

すぐに、次のようなメッセージが表示されます。

```
SMC server is ready.
```

- 4 サーバーが稼働しているのに問題が引き続き発生している場合は、いったんコンソールサーバーを停止してから、再起動します。

- a. コンソールサーバーを停止します。

```
# /etc/init.d/init.wbem stop
```

次のようなメッセージが表示されます。

```
Shutting down SMC server on port 898.
```

- b. コンソールサーバーを起動します。

```
# /etc/init.d/init.wbem start
```


Sun Java Web Console の操作 (手順)

この章では、システムにインストールおよび登録されている Web ベースの Sun システム管理アプリケーションを管理するための Sun Java™ Web Console について説明します。

この章の内容は次のとおりです。

- 57 ページの「Java Web Console の管理に関する新機能」
- 58 ページの「Java Web Console (概要)」
- 61 ページの「Java Web Console を開始する」
- 64 ページの「コンソールサービスの管理」
- 66 ページの「Java Web Console を設定する」
- 74 ページの「Java Web Console ソフトウェアのトラブルシューティング」
- 83 ページの「Java Web Console のリファレンス情報」

Java Web Console の使用に関連する手順については、61 ページの「Java Web Console を開始する (作業マップ)」および73 ページの「Java Web Console ソフトウェアのトラブルシューティング (作業マップ)」を参照してください。

Java Web Console の管理に関する新機能

この節では、この Solaris リリースの新機能について説明します。Solaris の新機能の一覧および Solaris リリースについての説明は、『[Solaris 10 の概要](#)』を参照してください。

Java Web Console サーバーの管理

Solaris 10 11/06: Java Web Console サーバーは、サービス管理機能 (SMF) によってサービスとして管理されます。SMF の詳細は、[第 17 章「サービスの管理 \(概要\)」](#)を参照してください。

Java Web Console で使用できるアプリケーション

Solaris 10 6/06: Java Web Console では ZFS™ Web ベース管理ツールを使用できます。このツールでは、コマンド行インタフェース (CLI) で実行できる管理タスクの大部分を実行できます。たとえば、パラメータを設定したり、さまざまなプールやファイルシステムを表示したり、それらを更新したりできます。

次に、このツールを使って実行できる主な手順の例を示します。

- 新規ストレージプールを作成する。
- 既存のプールに容量を追加する。
- 既存のプールを別のシステムに移動 (エクスポート) する。
- 以前にエクスポートしたストレージプールをインポートして、別のシステムでできるようにする。
- ストレージプールに関する情報を表示する。
- ファイルシステムを作成する。
- zvol (仮想ボリューム) を作成する。
- ファイルシステムまたは zvol ボリュームのスナップショットを取る。
- ファイルシステムを以前のスナップショットにロールバックする。

Solaris ZFS Web ベース管理ツールの使用方法の詳細は、『[Solaris ZFS 管理ガイド](#)』を参照してください。

注 - Sun Java Enterprise System ソフトウェアには、Java Web Console 内で実行される管理アプリケーションがいくつか含まれています。

Java Web Console (概要)

Java Web Console は、ユーザーが Web ベースのシステム管理アプリケーションにアクセスするための共通の場所を提供します。Web コンソールにアクセスするには、サポートされている Web ブラウザのいずれかを使用し、安全な https ポートを介してログインします。Web コンソールが提供する単一のエントリポイントを使用すれば、複数のアプリケーションの URL を覚える必要がなくなります。また、単一のエントリポイントにより、Web コンソールに登録されているすべてのアプリケーションのユーザー認証や承認も行うことができます。

Web コンソールベースのアプリケーションはすべて、同一のユーザーインタフェースのガイドラインに準拠しているため、より使いやすくなっています。また、Web コンソールは、登録されているすべてのアプリケーションに対し、ユーザーセッションの監査およびロギングサービスも提供します。

Java Web Console について

Java Web Console とは、システムにインストールおよび登録されている Sun システム管理の Web ベースのアプリケーションを得られる Web ページのことです。登録は、自動的にアプリケーションのインストールプロセスの一環となります。したがって、登録に管理者が介入する必要はありません。

Java Web Console には、次の機能が用意されています。

- ログインと、ブラウザベースのシステム管理アプリケーションの起動用の単一のエントリポイント

Java Web Console は、Sun のシステム管理アプリケーションの流れを汲んでいます。コンソールからは、アプリケーション名をクリックするだけでブラウザベースの管理アプリケーションを起動できます。Java Web Console と Solaris 管理コンソールとの間に互換性はありません。Java Web Console はブラウザを介して利用する Web アプリケーションで、Solaris 管理コンソールはコマンド行から起動する Java アプリケーションです。これらのコンソールは完全に独立しているため、両方のコンソールを同じシステム上で同時に実行することができます。

- 安全な https ポートを介したシングルサインオン

ここでのシングルサインオンは、Web コンソールに対して認証を行なったあとで、管理アプリケーションごとに認証を行う必要がないことを意味します。ユーザー名とパスワードの入力は、コンソールセッションにつき一度だけ行います。

- 動的に編成および集約されたアプリケーション

アプリケーションは、コンソールの起動ページで、もっとも適切な管理作業のカテゴリに分類されてインストールおよび表示されます。

カテゴリには次のものがあります。

- システム
 - ストレージ
 - サービス
 - デスクトップアプリケーション
 - その他
 - 共通の外観と操作性
- Web コンソールアプリケーションはすべて、同じユーザーインタフェース (UI) コンポーネントを使用し、同じ動きをするため、管理者の習得にかかる時間が短くなります。
- 標準の拡張可能な認証、承認、および監査機構
- Java Web Console では、プラグ接続できる認証モジュール (PAM)、役割に基づくアクセス制御 (RBAC) の役割、および基本セキュリティモジュール (BSM) の監査をサポートしています。

Java Web Console の管理コマンド

Java Web Console には、次の管理コマンドがあります。

- **smcwebserver** – このコマンドは、コンソールの Web サーバーを起動および停止します。
- **wcadmin** – **Solaris 10 11/06** リリース以降では、コンソールを設定したりコンソールアプリケーションを登録および配備したりする際にこのコマンドが使用されます。詳細は、[wcadmin\(1M\)](#) のマニュアルページを参照してください。
- **smreg** – **Solaris 10**、**Solaris 10 1/06**、および **Solaris 10 6/06 OS** では、すべてのコンソールアプリケーションを登録する際にこのコマンドが使用されます。

Solaris 10 11/06 リリース以降では、このコマンドを使用するのは、Java Web Console 3.0 より前のバージョンのコンソール用に作成されたレガシーアプリケーションを登録する場合だけにしてください。

これらのコマンドを使用して、この章で説明されているさまざまな作業を実行します。

各コマンドの詳細は、[smcwebserver\(1M\)](#)、[wcadmin\(1M\)](#)、および [smreg\(1M\)](#) のマニュアルページを参照してください。

サポートされている Web ブラウザ

Java Web Console は、Solaris OS の稼働中に、次のいずれかのブラウザで使用できます。

- Mozilla (バージョン 1.4 以降)
- Netscape (バージョン 6.2 以降)
- Firefox (バージョン 1.0 以降)

Java Web Console を開始する (作業マップ)

作業	説明	参照先
Java Web Console の起動ページからアプリケーションを起動します。	Java Web Console の起動ページには、ユーザーが使用する権限を持っている登録済みのすべてのシステム管理アプリケーションが表示されます。特定のアプリケーションに接続するには、そのアプリケーション名をクリックします。	62 ページの「Java Web Console の起動ページからアプリケーションを起動する方法」
コンソールサーバーの起動、停止、有効化、および無効化を行います。	コンソールと登録済みアプリケーションを実行するための Web サーバーを管理できます。	64 ページの「コンソールサービスを起動する方法」 65 ページの「システムの起動時に実行されるようコンソールサービスを有効にする方法」 65 ページの「コンソールサービスを停止する方法」 66 ページの「コンソールサービスを無効にする方法」
Java Web Console のプロパティを変更します。	Web コンソールのデフォルトのプロパティは変更する必要がありません。 変更できるプロパティは次のとおりです。 ■ コンソールのセッションタイムアウト ■ ログingleベル ■ 監査実装	69 ページの「Java Web Console のプロパティを変更する方法」

Java Web Console を開始する

Java Web Console の起動ページには、ユーザーが使用する権限を持つ登録済みのシステム管理アプリケーションのリストと、各アプリケーションの簡単な説明が表示されます。特定のアプリケーションに接続するには、そのアプリケーション名をクリックします。アプリケーション名は実際のアプリケーションへのリンクです。デフォルトでは、選択されたアプリケーションが Web コンソールのウィンドウで開きます。「各アプリケーションを新しいウィンドウで起動する」チェックボックスをクリックすると、アプリケーションを別のブラウザウィンドウで開くことができます。

す。アプリケーションを別のウィンドウで開くと、Web コンソールの起動ページを引き続き利用できるため、そのページに戻って1回のログインで複数のアプリケーションを起動できます。

コンソールの起動ページにアクセスするには、Web の場所を示すフィールドに次の形式の URL を入力します。

https://hostname.domain:6789

これらの内容は次のとおりです。

- https は Secure Socket Layer (SSL) 接続を指定します
- hostname.domain には、コンソールを提供するサーバーの名前とドメインを指定します
- 6789 は、コンソールに割り当てられているポート番号です

注 - 特定のシステムから初めて Java Web Console にアクセスしたときは、Web コンソールの起動ページが表示される前に、サーバーの証明書に同意する必要があります。

RBAC がシステム上で有効になっていて、かつユーザーの ID が役割に割り当てられている場合、ログイン成功後に役割のパスワードの入力を求められます。役割を引き受けた場合、その引き受けた役割の承認チェックが行われます。NO ROLE を選択して役割を引き受けないことも可能ですが、その場合、そのユーザーの ID に対して承認チェックが行われます。承認チェックに成功すると、Web コンソールの起動ページが表示されます。

▼ Java Web Console の起動ページからアプリケーションを起動する方法

- 1 **Java Web Console** と互換性のある Web ブラウザ (Mozilla 1.7 または Firefox 1.0 など) を起動します。

サポートされているブラウザの一覧は、[60 ページの「サポートされている Web ブラウザ」](#)を参照してください。

- 2 **Web ブラウザ**の場所を示すフィールドに、コンソールの URL を入力します。

たとえば、管理サーバーホストの名前が sailfish、ドメインが sw である場合、URL は https://sailfish.sw:6789 となります。この URL によって Web コンソールのログインページが表示されます。

3 サーバーの証明書を受け入れます。

サーバー証明書への同意は、ブラウザセッションにつき一度だけ行います。コンソールにログインしたりアプリケーションを起動したりするたびに行う必要はありません。

次の図に示すようなログインページが表示されます。



図 3-1 Java Web Console のログインページ

4 ユーザー名とパスワードを入力し、必要に応じてRBACの役割も入力します。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

コンソールサービスは、ユーザーを認証するために資格を調べ、ユーザーがコンソールと登録済みアプリケーションの使用を承認されていることを確認します。

5 アプリケーションを新しいウィンドウで開く場合は、「各アプリケーションを新しいウィンドウで起動する」チェックボックスをクリックします。

このオプションを選択しないと、アプリケーションはデフォルトのウィンドウ (起動ページと置き換わる) で実行されます。

6 実行するアプリケーションのリンクをクリックします。

ヒント-個々のアプリケーションを直接起動して、起動ページを省略することもできます。その場合は、次の構文を使用します。

```
https://hostname.domain:6789/app-context-name
```

ここでの *app-context-name* は、アプリケーションが配備されるときに使われる名前です。

アプリケーションのコンテキスト名を調べるには、次のいずれかを実行します。

- アプリケーションのマニュアルを読みます。
- `wcadmin list -a` コマンドまたは `smreg list -a` コマンドを実行して、配備済み Web アプリケーションとそのコンテキスト名の一覧を表示します。
- Web コンソールの起動ページからアプリケーションを実行して、アドレスの場所を示すフィールドに表示されている URL を書き留めます。次にそのアプリケーションを使用するときは、この URL を直接入力してアクセスできます。または、この場所をブックマークに追加して、ブックマークからアプリケーションにアクセスすることができます。

コンソールサービスの管理

Solaris 10 11/06: Java Web Console サービスは、サービス管理機能 (SMF) を介して管理されます。コンソールサービスの起動、停止、有効化、および無効化は、SMF のコマンドまたは `smcwebserver` スクリプトを使用して行うことができます。SMF でコンソールに使用される FMRI は `system/webconsole:console` です。

▼ コンソールサービスを起動する方法

この手順では、サーバーを一時的に起動します。サーバーがシステムのブート時に起動しないよう無効にされていた場合、サーバーは無効のままになります。サーバーが有効にされていた場合、サーバーは有効のままになります。

Solaris 10 11/06 リリース以降では、サーバーが無効の状態で稼働している場合、その時点の `enabled` の状態は `true (temporary)` と表示されます。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「RBAC の構成 (作業マップ)」を参照してください。
- 2 有効状態を変更せずに、サーバーをすぐに起動します。

```
# smcwebserver start
```


▼ システムの起動時に実行されるようコンソールサービスを有効にする方法

この手順では、システムの起動時に実行されるようコンソールサービスを有効にします。現在のセッションではコンソールは起動されません。

Solaris 10 11/06 リリース以降では、この手順によって SMF で `general/enabled` プロパティが `true` に設定されるため、システムのブート時にサーバーが起動されます。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。
- 2 システムのブート時に起動されるようサーバーを有効にします。

```
# smcwebserver enable
```

Solaris 10 11/06: また、サーバーをすぐに起動するとともに、システムのブート時に起動するようサーバーを有効にするには、次のコマンドを使用します。

```
# svcadm enable system/webconsole:console
```

注 - Solaris 10 11/06 リリースを実行している場合には、`smcwebserver` コマンドを使ってコンソールを有効にすることはできません。`svcadm` コマンドを使用する必要があります。

▼ コンソールサービスを停止する方法

この手順では、サーバーを一時的に停止します。サーバーがシステムのブート時に起動しないよう無効にされていた場合、サーバーは無効のままになります。サーバーが有効にされていた場合、サーバーは有効のままになります。

Solaris 10 11/06 リリース以降では、サーバーが有効の状態で停止している場合、その時点の `enabled` の状態は `false (temporary)` と表示されます。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。
- 2 有効状態を変更せずに、サーバーをすぐに停止します。

```
# smcwebserver stop
```

▼ コンソールサービスを無効にする方法

無効にされている場合、コンソールサーバーはシステムのブート時に起動しません。

Solaris 10 11/06 リリース以降では、この手順によって SMF でコンソールの `general/enabled` プロパティが `false` に設定されるため、システムのブート時にコンソールサーバーは起動しません。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。
- 2 システムのブート時に起動しないようサーバーを無効にします。

```
# smcwebserver disable
```

Solaris 10 11/06: また、サーバーをすぐに停止するとともに、システムのブート時に起動しないようサーバーを無効にするには、次のコマンドを使用します。

```
# svcadm disable system/webconsole:console
```

注 - Solaris 10 11/06 リリースを実行している場合には、`smcwebserver` コマンドを使ってコンソールを無効にすることはできません。`svcadm` コマンドを使用する必要があります。

Java Web Console を設定する

Java Web Console は、管理者が介入しなくても実行できるように事前設定されています。ただし、Web コンソールの設定プロパティを上書きすることで、コンソールのデフォルトの動作の一部を変更することもできます。

注 - **Solaris 10 11/06 OS** 以降では、これらのプロパティを変更するには `wcadmin` コマンドを使用する必要があります。以前は `smreg` コマンドが使用されていました。`wcadmin` コマンドの詳細は、`wcadmin(1M)` のマニュアルページを参照してください。

コンソールの設定ファイル内のプロパティは、コンソールの動作を制御します。この動作を変更するには、プロパティに新しい値を定義して、デフォルト値を上書きします。ほとんどのプロパティのデフォルト値は、独自のログインサービスを指定するなど、デフォルト値では満たせない特別なニーズがある場合を除き、上書きしないようにしてください。

通常、変更の対象となる可能性があるのは、次のプロパティー値です。

- コンソールのセッションタイムアウト

Web コンソールのセッションタイムアウト時間は、`session.timeout.value` プロパティーによって制御されます。このプロパティーでは、セッションが時間切れになるまで、ユーザーの対話的操作がない状態で Web コンソールページを表示できる時間を制御します。タイムアウトに達すると、ユーザーは再度ログインする必要があります。デフォルト値は 15 分間です。ユーザー自身のセキュリティポリシーに従って、新しい値を分単位で設定できます。ただし、このプロパティーはすべてのコンソールユーザーおよび登録されているすべてのアプリケーションのタイムアウト時間を制御するので注意が必要です。

セッションタイムアウトの変更方法の例については、[例 3-1](#) を参照してください。

- ログingleレベル

ログingleのプロパティーは、ログingleサービスを設定するために使用します。コンソールのログファイルは `/var/log/webconsole/console` ディレクトリに作成されます。`logging.default.level` プロパティーは、どのメッセージがログingleされるかを決定します。コンソールログは、問題のトラブルシューティングに役立つ情報を提供します。

このログingleレベルはログingleサービス経由で書き込まれるすべてのメッセージに適用されますが、ログingleサービスはデフォルトで Solaris リリースの syslog を使用します。syslog のログファイルは `/var/adm/messages` です。ファイル `/var/log/webconsole/console/console_debug_log` に含まれるログメッセージは、デバッグサービスを有効化した場合に書き込まれます。これを行うには、Using the Console Debug Trace Log の説明に従って [71 ページの「コンソールのデバッグトレースログを使用する」](#) プロパティーを設定します。デフォルトログingleサービスとデバッグログingleサービスは別個のものですが、syslog への Java Web Console ログingleメッセージはデバッグの助けになるように `console_debug_log` にも書き込まれます。一般に、`logging.default.level` で設定されるログingleサービスは、コンソールアプリケーションによるログingle用として常に有効にしておきます。`debug.trace.level` で設定されるデバッグログingleを有効にするのは、問題を調査する場合だけにしてください。

`logging.default.level` に使用できるプロパティー値は、次のとおりです。

- all
- info
- off
- severe
- warning

ログingleレベルの変更方法の例については、[例 3-2](#) を参照してください。

- 監査実装

監査とは、セキュリティ関連の管理イベントを生成および記録する処理です。イベントは、特定のユーザーがシステム上の管理情報を更新したことを示します。監査実装は、監査イベントを生成するサービスとアプリケーションで使用されます。

Web コンソールでは、次の監査イベントが定義されます。

- ログイン
- ログアウト
- 役割の引き受け

監査イベントが発生すると、そのイベントのレコードが監査ログに作成されます。監査ログの場所は、使用されている監査実装によって異なります。Web コンソールの監査サービスは、基になるオペレーティングシステムによって提供される監査実装を使用します。

Web コンソールは、次の3つの監査実装をサポートしています。Solaris、Log、および None です。これらのキーワードのいずれかを `audit.default.type` 設定プロパティの値として指定することで、監査実装を選択できます。一度に有効にできる監査実装は1つのみです。

サポートされている監査実装の種類は次のとおりです。

- Solaris

Solaris はデフォルトの実装です。この実装は、BSM 監査機構に対応しています。この監査機構では、`/var/audit` ディレクトリのシステムファイルに監査レコードが書き込まれます。

これらのレコードは、`praudit` コマンドを使って表示できます。イベントを取り込むためには、システムで BSM 監査機構を有効にする必要があります。また、`/etc/security/audit_control` ファイルに、どのイベントを生成するかを指示するエントリを入れておく必要があります。ユーザーごとにログインとログアウトのイベントを表示するために、`lo` イベントをフラグオプションとして設定する必要があります。詳細は、[praudit\(1M\)](#) と [bsmconv\(1M\)](#) のマニュアルページ、および『Solaris のシステム管理 (セキュリティサービス)』の [パート VII 「Solaris 監査」](#) を参照してください。

- Log

この実装を設定すると、システムの `syslog` サービスに書き込むことができます。ロギングサービスが `info` レベルで有効になっている場合は、監査メッセージがコンソールログに書き込まれます。詳細は、[例 3-2](#) を参照してください。

- None

監査イベントは生成されません。監査メッセージは、デバッグトレースログ (有効な場合) に書き込まれます。

監査実装の指定方法の例については、[例 3-5](#) を参照してください。

▼ Java Web Console のプロパティを変更する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 実行している Solaris リリースに応じて、選択したプロパティの値を次のように変更します。

- Solaris 10 11/06 以降のリリースを実行している場合は、次のコマンドを使用します。

```
# wcadmin add -p -a console name=value
```

-p オブジェクトタイプがプロパティであることを指定します。

-a console console というアプリケーションに対するプロパティ変更であることを指定します。コンソールのプロパティを変更する場合は、常に -a console オプションを使用する必要があります。

name=value プロパティ名とそのプロパティの新しい値を指定します。

- Solaris 10、Solaris 10 1/06、および Solaris 10 6/06 リリースの場合は、次のコマンドを使用します。

```
# smreg add -p -c name
```

- 3 (省略可能) コンソールのプロパティをそのデフォルト値にリセットします。

- Solaris 10 11/06 以降のリリースを実行している場合は、次のコマンドを使用します。

```
# wcadmin remove -p -a console name=value
```

- Solaris 10、Solaris 10 1/06、および Solaris 10 6/06 リリースの場合は、次のコマンドを使用します。

```
# smreg remove -p -c name
```

-p オブジェクトタイプがプロパティであることを指定します。

-c コンソールアプリケーションに対するプロパティ変更であることを指定します。コンソールのプロパティを変更する場合は、常に -c オプションを使用する必要があります。

name プロパティ名とそのプロパティの新しい値を指定します。

例 3-1 Java Web Console のセッションタイムアウトプロパティを変更する

この例では、セッションタイムアウトの値を5分に設定する方法を示します。

```
# wcadmin add -p -a console session.timeout.value=5
```

例 3-2 Java Web Console のロギングレベルを設定する

この例では、ロギングレベルを all に設定する方法を示します。

```
# wcadmin add -p -a console logging.default.level=all
```

例 3-3 Java Web Console のロギングレベルをデフォルト値にリセットする

この例では、ロギングレベルをデフォルトにリセットする方法を示します。

```
# wcadmin remove -p -a console logging.default.level
```

例 3-4 Java Web Console の Java バージョンを指定する

この例では、コンソールの Java バージョンを設定する方法を示します。

```
# wcadmin add -p -a console java.home=/usr/java
```

例 3-5 Java Web Console の監査実装を選択する

この例では、監査実装を None に設定する方法を示します。

```
# wcadmin add -p -a console audit.default.type=None
```

有効な監査タイプは次のとおりです。

None	監査なし
Log	syslog への監査メッセージの格納
Solaris	BSM への監査メッセージの格納

Java Web Console のユーザー ID

デフォルトでは、Web コンソールは noaccess というユーザー ID のもとで実行されます。ただし、システム構成によっては、noaccess ユーザーが無効にされていたり、noaccess ユーザーのログインシェルが無効なエントリに設定されてこのユーザー ID が使用できなくされていたりすることがあります。

noaccess ユーザーが使用できない場合は Web コンソールサーバーの起動や設定を行うことができないため、代替りのユーザー ID を指定する必要があります。最初の起動でコンソールサーバーが設定される前に、ユーザー ID を 1 回だけ変更することが理想的です。

コンソールを起動する前に次のいずれかのコマンドを使用して、root 以外の代替ユーザー ID のもとで実行されるように Web コンソールを設定できます。

```
# smcwebserver start -u username
```

このコマンドを実行すると、Web コンソールサーバーが指定のユーザー ID のもとで起動されます。コンソールの初回起動時にこのコマンドが発行された場合には、Web コンソールサーバーはそれ以降起動されるたびにこの ID で実行されます。

Solaris 10 11/06 以降のリリースを実行している場合は、次のコマンドも使用できます。

```
# wcadm add -p -a console com.sun.web.console.user=  
username
```

注 - Solaris 10 11/06 リリース以降では、システムの初回起動時にコンソールも起動し、noaccess の下で実行するように自動的に設定されます。その結果、ユーザー ID は、変更する機会が得られないまま noaccess に設定されてしまいます。その場合は、まず次のコマンドを使ってコンソールを初期の未設定状態にリセットしてください。続いて、コンソールの再起動時に別のユーザー ID を指定します。

```
# smcwebserver stop  
# /usr/share/webconsole/private/bin/wcremove -i console  
# smcwebserver start -u new_user_identity
```

Solaris 10、Solaris 10 1/06、および Solaris 10 6/06 リリースの場合は、次のコマンドを使用します。

```
# smreg add -p -c com.sun.web.console.user=username
```

このコマンドを実行すると、次回以降サーバーが起動されるたびに、Web コンソールサーバーが指定のユーザー ID のもとで起動されます。

コンソールのデバッグトレースログを使用する

デフォルトでは、コンソールはデバッグメッセージを記録しません。デバッグのロギングをオンにして、コンソールサービスの問題のトラブルシューティングに利用することができます。

デバッグのロギングをオンにするには、`debug.trace.level` プロパティを 0 以外の値に設定します。

可能な選択肢は次のとおりです。

- 1 - 重大エラーの可能性があるエラーを記録する場合は、この設定を使用します。
- 2 - 重要なメッセージやレベル 1 のエラーメッセージを記録する場合は、この設定を使用します。
- 3 - 考えられるすべてのメッセージを詳細な情報とともに記録するには、この設定を使用します。

Solaris 10、Solaris 10 1/06、および Solaris 10 6/06 リリースの場合、デバッグトレースログはデフォルトで `/var/log/webconsole` ディレクトリ内に作成されます。Solaris 10 11/06 リリース以降では、`/var/log/webconsole/console` ディレクトリ内にログが作成されます。ログファイルの名前は `console_debug_log` です。`console_debug_log.1` や `console_debug_log.2` などの履歴ログがこのディレクトリに格納されている場合もあります。このディレクトリには最高 5 つ (デフォルトの設定) の履歴ログを格納しておくことができ、そのあとはもっとも古いログが削除され、新しいログが作成されます。

例 3-6 コンソールのデバッグトレースログレベルを設定する

次のコマンドを使用して、デバッグトレースログレベルを 3 に設定します。

Solaris 10 11/06 リリースの場合は、次のコマンドを使用します。

```
# wcadmin add -p -a console debug.trace.level=3
```

Solaris 10、Solaris 10 1/06、および Solaris 10 6/06 リリースの場合は、次のコマンドを使用します。

```
# smreg add -p -c debug.trace.level=3
```

例 3-7 `debug.trace.level` プロパティの状態を確認する

`debug.trace.level` プロパティの状態を確認する場合は、`wcadmin list` コマンドまたは `smreg list` コマンドを使用します。

Solaris 10 11/06:

```
# wcadmin list -p | grep "debug.trace.level"
```

Solaris 10、Solaris 10 1/06、および Solaris 10 6/06 リリースの場合は、次のコマンドを使用します。

例 3-7 `debug.trace.level` プロパティの状態を確認する (続き)

```
# smreg list -p | grep "debug.trace.level"
```

Java Web Console ソフトウェアのトラブルシューティング (作業マップ)

作業	説明	参照先
コンソールが稼働しているか、および有効かどうかを確認します。	<code>smcwebserver</code> 、 <code>wcadmin</code> 、および <code>svcs</code> の各コマンドを使用して、コンソールが稼働しているか、および有効かどうかを確認します。この情報は問題のトラブルシューティングに役立ちます。	75 ページの「コンソールが稼働しているか、および有効かどうかを確認する方法」
コンソールのリソースとプロパティを一覧表示します。	トラブルシューティングの目的で、コンソールのリソースとプロパティに関する情報を収集できます。	75 ページの「コンソールのリソースとプロパティを一覧表示する方法」
アプリケーションがレガシーアプリケーションかどうかを判定します。	現行アプリケーションの登録と配備は、コンソールサーバーの稼働中に単一のコマンドで実行されます。レガシーアプリケーションの場合は、登録時にコンソールサーバーを停止する必要があります。アプリケーションの登録や登録解除を行うときは、まずアプリケーションがレガシーアプリケーションかどうかを判定する必要があります。	78 ページの「アプリケーションがレガシーアプリケーションかどうかを判定する方法」
登録済みのアプリケーションを一覧表示します。	Java Web Console に登録されているすべてのアプリケーションの一覧を表示できます。登録済みのアプリケーションを一覧表示すると、トラブルシューティングに役立つ情報が得られます。	78 ページの「配備済みのアプリケーションを一覧表示する方法」

作業	説明	参照先
レガシーアプリケーションを Java Web Console に登録します。	レガシーアプリケーションを使用する必要がある場合は、まずアプリケーションを Java Web Console に登録する必要があります。	80 ページの「レガシーアプリケーションを Java Web Console に登録する方法」
Java Web Console からレガシーアプリケーションを登録解除します。	レガシーアプリケーションを Java Web Console に登録しない場合は、手順に従ってレガシーアプリケーションを登録解除します。	81 ページの「Java Web Console からレガシーアプリケーションを登録解除する方法」
現行アプリケーションを Java Web Console に登録します。	新しいアプリケーションを使用するには、まずアプリケーションを Java Web Console に登録する必要があります。	81 ページの「現行アプリケーションを Java Web Console に登録する方法」
Java Web Console から現行アプリケーションを登録解除します。	場合によっては、Java Web Console から現行アプリケーションを登録解除する必要があることがあります。	82 ページの「Java Web Console から現行アプリケーションを登録解除する方法」
Java Web Console へのリモートアクセスを使用可能にします。	ほかのアクセス制限はそのまま、コンソールへのリモートアクセスだけを使用可能にすることができます。	87 ページの「Java Web Console へのリモートアクセスを使用可能にする方法」
コンソールの内部パスワードを変更します。	Java Web Console では内部パスワードが使用されます。セキュリティ侵入の可能性を低減するために、これらのパスワードを変更することができます。	88 ページの「コンソールの内部パスワードを変更する方法」

Java Web Console ソフトウェアのトラブルシューティング

次の情報は、Java Web Console ソフトウェアの使用時に検出される可能性がある問題のトラブルシューティングに役立ちます。

コンソールの状態とプロパティの確認

smcwebserver、wcadmin、および svcs の各コマンドを使用して、各種の情報を取得できます。これらの情報は問題のトラブルシューティングに役立つ場合があります。

▼ コンソールが稼働しているか、および有効かどうかを確認する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 サーバーの状態を確認します。

```
# smcwebserver status
Sun Java(TM) Web Console is running
```

- 3 **Solaris 10 11/06:** コンソールの SMF 状態と有効状態を確認します。

```
# svcs -l system/webconsole:console
fmri          svc:/system/webconsole:console
name          java web console
enabled       true
state         online
next_state    none
state_time    Wed 17 May 2006 01:22:32 PM EDT
logfile       /var/svc/log/system-webconsole:console.log
restarter     svc:/system/svc/restarter:default
contract_id   129
dependency    require_all/none svc:/milestone/multi-user (online)
```

smcwebserver コマンドを使用して、サーバーの有効または無効の状態を変更せずにサーバーを起動または停止すると、enabled プロパティは false (temporary) または true (temporary) と表示されます。

▼ コンソールのリソースとプロパティを一覧表示する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 コンソールのリソースとプロパティを一覧表示します。

Solaris 10 11/06 以降のリリースを実行している場合は、次のコマンドを使用します。

```
# wcadmin list
```

Deployed web applications (application name, context name, status):

```
console      ROOT          [running]
console      com_sun_web_ui [running]
```

console	console	[running]
console	manager	[running]
legacy	myapp	[running]

Registered jar files (application name, identifier, path):

console	audit_jar	/usr/lib/audit/Audit.jar
console	console_jars	/usr/share/webconsole/lib/*.jar
console	jato_jar	/usr/share/lib/jato/jato.jar
console	javahelp_jar	/usr/jdk/packages/javahelp-2.0/lib/*.jar
console	shared_jars	/usr/share/webconsole/private/container/shared/lib/*.jar

Registered login modules (application name, service name, identifier):

console	ConsoleLogin	userlogin
console	ConsoleLogin	rolelogin

Shared service properties (name, value):

ENABLE	yes
java.home	/usr/jdk/jdk1.5.0_06

注 - SMF は前の手順で示した独自の enabled プロパティを使用するため、この **ENABLE** プロパティは無視されます。ENABLE プロパティは、コンソールサーバーが SMF によって管理されない、以前の Solaris システムで使用されます。

Solaris 10、Solaris 10 1/06、および Solaris 10 6/06 リリースの場合は、次のコマンドを使用します。

smreg list

The list of registered plugin applications:

com.sun.web.console_2.2.4	/usr/share/webconsole/console
com.sun.web.ui_2.2.4	/usr/share/webconsole/com_sun_web_ui
com.sun.web.admin.example_2.2.4	/usr/share/webconsole/example

The list of registered jar files:

com_sun_management_services_api.jar	scoped to ALL
com_sun_management_services_impl.jar	scoped to ALL
com_sun_management_console_impl.jar	scoped to ALL
com_sun_management_cc.jar	scoped to ALL
com_sun_management_webcommon.jar	scoped to ALL
com_iplanet_jato_jato.jar	scoped to ALL
com_sun_management_solaris_impl.jar	scoped to ALL
com_sun_management_solaris_implx.jar	scoped to ALL

The list of registered login modules for service ConsoleLogin:

```
com.sun.management.services.authentication.PamLoginModule optional
use_first_pass="true" commandPath="/usr/lib/webconsole";
com.sun.management.services.authentication.RbacRoleLoginModule requisite
force_role_check="true" commandPath="/usr/lib/webconsole";
```

The list of registered server configuration properties:

```
session.timeout.value=15
authentication.login.cliservice=ConsoleLogin
logging.default.handler=com.sun.management.services.logging.ConsoleSyslogHandler
logging.default.level=info
logging.default.resource=com.sun.management.services.logging.resources.Resources
logging.default.filter=none
logging.debug.level=off
audit.default.type=None
audit.None.class=com.sun.management.services.audit.LogAuditSession
audit.Log.class=com.sun.management.services.audit.LogAuditSession audit.class.fail=none
authorization.default.type=SolarisRbac
authorization.SolarisRbac.class=
com.sun.management.services.authorization.SolarisRbacAuthorizationService
authorization.PrincipalType.class=
com.sun.management.services.authorization.PrincipalTypeAuthorizationService
debug.trace.level=0
.
.
.
No environment properties have been registered.
```

コンソールへのアクセスに関連する問題

コンソールへのアクセスに問題がある場合は、コンソールサーバーが有効になっていないか、セキュリティ設定の制限が厳しいことを示している可能性があります。詳細は、[74 ページ](#)の「[コンソールの状態とプロパティの確認](#)」および [83 ページ](#)の「[Java Web Console のセキュリティに関する注意事項](#)」を参照してください。

アプリケーションの登録に関連する問題

この節では、コンソールアプリケーションの登録で発生する可能性のある問題の解決方法について説明します。特定のコンソールアプリケーションの詳細は、そのアプリケーションのマニュアルを参照してください。

注-通常、コンソールアプリケーションはインストールプロセスの一環として登録されるため、通常はユーザー自身がアプリケーションを登録する必要はありません。

Solaris 10 11/06 リリース以降では、Web コンソールでのアプリケーション登録の方法が変更されましたが、以前のバージョンのコンソール用に開発されたアプリケーションも引き続きサポートされます。現行アプリケーションの登録と配備は、コンソールサーバーの稼働中に単一のコマンドで実行されます。以前のバージョンのコンソール用に開発されたアプリケーションは「レガシー」アプリケーションと呼ばれ、その登録時にはコンソールサーバーを停止する必要があります。アプリケーションの登録や登録解除を行うときは、次の手順に従って、まずアプリケーションがレガシーアプリケーションかどうかを判定する必要があります。

▼ アプリケーションがレガシーアプリケーションかどうかを判定する方法

- 1 アプリケーションの app.xml ファイルを表示します。

app.xml ファイルは、アプリケーションの WEB-INF ディレクトリにあります。

- 2 app.xml ファイル内の registrationInfo タグを調べます。

レガシーアプリケーションの場合、registrationInfo タグのバージョンは 2.x です。たとえば、registrationInfo version="2.2.4" となります。

現行アプリケーションの場合、registrationInfo タグのバージョンは 3.0 以上です。たとえば、registrationInfo version="3.0" となります。

▼ 配備済みのアプリケーションを一覧表示する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、[『Solaris のシステム管理 \(セキュリティサービス\)』の「RBAC の構成 \(作業マップ\)」](#)を参照してください。

- 2 配備済みのアプリケーションを一覧表示します。

Solaris 10 11/06 以降のリリースを実行している場合は、次のコマンドを使用します。

```
# wcadmin list -a
```

Deployed web applications (application name, context name, status):

```
console  ROOT                [running]
console  com_sun_web_ui         [running]
console  console                 [running]
```

```

console  manager      [running]
legacy   myapp         [running]

```

このコマンドは、登録済みかつ配備済みのアプリケーションをすべて表示します。レガシーアプリケーションは `legacy` というアプリケーション名で表示されます。詳細は、78 ページの「[アプリケーションがレガシーアプリケーションかどうかを判定する方法](#)」を参照してください。一覧に表示されるその他のアプリケーションはすべて現行アプリケーションであり、81 ページの「[現行アプリケーションを Java Web Console に登録する方法](#)」に説明されている方法で登録されます。

通常、アプリケーションに対して表示される状態には `running`、`stopped` のいずれかが含まれます。状態が `running` の場合、アプリケーションはその時点で読み込まれており、使用可能な状態になっています。状態が `stopped` の場合、アプリケーションはその時点で読み込まれておらず、使用不可能な状態になっています。アプリケーションのなかには、登録と配備は成功したが、アプリケーション内の問題により読み込まれないものもあります。その場合、アプリケーションの状態は `stopped` になります。`console_debug_log` をチェックし、そのアプリケーションの読み込み時にコンソールの背後の Web コンテナである Tomcat からのトレースバックを含むエラーが発生していないか確認してください。`console_debug_log` の詳細については、71 ページの「[コンソールのデバッグトレースログを使用する](#)」を参照してください。

コンソールアプリケーションを含むすべてのアプリケーションの表示が `stopped` になっている場合、それは通常、コンソールの Web コンテナが動作していないことを意味します。この場合、その Web コンテナに登録された静的な `context.xml` ファイルからアプリケーションの一覧が取得されます。

Solaris 10、Solaris 10 1/06、および Solaris 10 6/06 リリースの場合は、次のコマンドを使用します。

```
# smreg list -a
```

The list of registered plugin applications:

```

com.sun.web.console_2.2.4      /usr/share/webconsole/console
com.sun.web.ui_2.2.4           /usr/share/webconsole/com_sun_web_ui
com.sun.web.admin.yourapp_2.2.4 /usr/share/webconsole/yourapp

```

▼ レガシーアプリケーションを **Java Web Console** に登録する方法

注 - この手順は、Solaris 10、Solaris 10 1/06、および Solaris 10 6/06 リリースのすべてのコンソールアプリケーションに適用されます。Solaris 10 11/06 リリース以降では、この手順はレガシーアプリケーションと識別されるアプリケーションだけに適用されます。現行アプリケーションの登録手順については、[81 ページの「現行アプリケーションを Java Web Console に登録する方法」](#)を参照してください。[78 ページの「アプリケーションがレガシーアプリケーションかどうかを判定する方法」](#)も参照してください。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。
- 2 **Web** サーバーを停止します。

```
# smcwebserver stop
```
- 3 アプリケーションを登録します。

```
# smreg add -a /directory/containing/application-files
```

smreg コマンドは、Java Web Console の登録テーブル内の情報を管理します。さらに、このスクリプトはアプリケーションを配備するための作業も行います。このコマンドのその他のオプションについては、[smreg\(1M\)](#)のマニュアルページを参照してください。
- 4 **Web** サーバーを再起動します。

```
# smcwebserver start
```

例 3-8 レガシーアプリケーションを登録する

この例では、`/usr/share/webconsole/example` ディレクトリにファイルが置かれているレガシーアプリケーションを登録する方法を示します。レガシーアプリケーションの場合は、アプリケーションを登録する前にコンソールサーバーを停止し、アプリケーションの登録後に起動する必要があります。smreg はレガシーコンソールアプリケーションなので、これによって表示される警告は無視してかまいません。

```
# smcwebserver stop
# smreg add -a /usr/share/webconsole/example
```

```
Warning: smreg is obsolete and is preserved only for
compatibility with legacy console applications. Use wcadmin instead.
```



```
Type "man wcadmin" or "wcadmin --help" for more information.
```

```
Registering com.sun.web.admin.example_version.
```

```
# smcwebserver start
```

▼ Java Web Console からレガシーアプリケーションを登録解除する方法

注 - この手順は、Solaris 10、Solaris 10 1/06、および Solaris 10 6/06 リリースのすべてのコンソールアプリケーションに適用されます。Solaris 10 11/06 リリース以降では、この手順はレガシーアプリケーションと識別されるアプリケーションだけに適用されます。現在のアプリケーションの登録を解除する手順については、[82 ページ](#)の「[Java Web Console から現行アプリケーションを登録解除する方法](#)」を参照してください。

特定のレガシーアプリケーションを Web コンソールの起動ページに表示させたくないが、ソフトウェアのアンインストールは行いたくない場合、`smreg` コマンドを使用してアプリケーションの登録を解除できます。詳細は、[78 ページ](#)の「[アプリケーションがレガシーアプリケーションかどうかを判定する方法](#)」を参照してください。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 アプリケーションの登録を解除します。

```
# smreg remove -a app-name
```

例 3-9 Java Web Console からレガシーアプリケーションを登録解除する

この例では、`app-name` として `com.sun.web.admin.example` を持つレガシーアプリケーションを登録解除する方法を示します。

```
# smreg remove -a com.sun.web.admin.example
```

```
Unregistering com.sun.web.admin.example_version.
```

▼ 現行アプリケーションを Java Web Console に登録する方法

Solaris 10 11/06: この手順は、コンソールサーバーの停止や起動を行わずに登録および配備できる更新されたコンソールアプリケーション用です。レガシーアプリケーションと、Solaris 10、Solaris 10 1/06、Solaris 10 6/06 の各リリースで実行されるす

すべてのコンソールアプリケーションの登録手順については、80 ページの「レガシーアプリケーションを Java Web Console に登録する方法」を参照してください。78 ページの「アプリケーションがレガシーアプリケーションかどうかを判定する方法」も参照してください。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 アプリケーションの登録と配備を行います。

```
wcadmin deploy -a app-name -x app-context-name  
/full path/to/app-name
```

例 3-10 現行アプリケーションを登録する

この例では、現在の Web コンソール用に開発または更新されたアプリケーションの登録と配備を行う方法を示します。

```
# wcadmin deploy -a newexample_1.0 -x newexample /apps/webconsole/newexample
```

▼ Java Web Console から現行アプリケーションを登録解除する方法

Solaris 10 11/06: この手順は、コンソールサーバーの停止や起動を行わずに登録解除および配備解除できる更新されたコンソールアプリケーション用です。レガシーアプリケーションと、Solaris 10、Solaris 10 1/06、Solaris 10 6/06 の各リリースで実行されるすべてのコンソールアプリケーションの登録解除手順については、81 ページの「Java Web Console からレガシーアプリケーションを登録解除する方法」を参照してください。アプリケーションがレガシーアプリケーションか更新されたアプリケーションかを判定する方法については、78 ページの「配備済みのアプリケーションを一覧表示する方法」および 78 ページの「アプリケーションがレガシーアプリケーションかどうかを判定する方法」を参照してください。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 アプリケーションの配備解除と登録解除を行います。

```
# wcadmin undeploy -a newexample_1.0 -x newexample
```

Java Web Console のリファレンス情報

このリファレンスの節では、次の内容について説明します。

- 83 ページの「[Java Web Console のセキュリティに関する注意事項](#)」
- 85 ページの「[authTypes タグを使って承認を指定する](#)」

Java Web Console のセキュリティに関する注意事項

Java Web Console に含まれているアプリケーションを使用するときは、セキュリティに関して注意しなければならないことがいくつかあります。

セキュリティに関する注意事項は、次のとおりです。

- **Java Web Console** へのアクセス - ブラウザを介してコンソールに接続できるかどうか。
- アプリケーションへのアクセス - Java Web Console の起動ページに特定のアプリケーションを表示できるかどうか。
- アプリケーションのアクセス権 - アプリケーションの一部またはすべてを実行するために必要なアクセス権のレベル。
- リモートシステムへのアプリケーションのアクセス - セキュリティの資格がリモートシステムにどのように関連しているか。
- コンソールで使用される内部パスワード - Solaris 10 11/06 リリース以降、コンソールの内部で使用されるデフォルトパスワードの変更。

Java Web Console へのアクセス

通常、有効なユーザーはすべてログインできるように、Web コンソールの起動アプリケーションに対するアクセス権は開放されています。ただし、Web コンソールの app.xml ファイル内の authTypes タグに権利を指定することで、コンソールへのアクセスを制限できます。このファイルは

/usr/share/webconsole/webapps/console/WEB-INF ディレクトリにあります。詳細は、85 ページの「[authTypes タグを使って承認を指定する](#)」を参照してください。

非常に高いセキュリティで保護されるようシステム構成が設定され、コンソールや登録済みアプリケーションの URL に対するリモートシステムからの接続が拒否される場合もあります。システムがリモートアクセスを防止するように設定されている場合は、https://hostname.domain:6789 としてコンソールにアクセスしようとすると、ブラウザに次のようなメッセージが表示されます。

```
Connect to hostname.domain:6789 failed (Connection refused)
```

システムで有効になっている SMF プロファイルがアクセスを制限している場合もあります。プロファイルの詳細は、[370 ページの「SMF プロファイル」](#)を参照してください。リモートシステムからコンソールへのアクセスを許可する手順については、[87 ページの「Java Web Console へのリモートアクセスを使用可能にする」](#)を参照してください。

Java Web Console のアプリケーションへのアクセス

Web コンソールに正常にログインしたあとで、そのコンソールに登録されているすべてのアプリケーションに自動的にアクセスできるようになるとは限りません。通常、アプリケーションはすべてのユーザーがコンソールの起動ページでそれを表示できるようにインストールされます。管理者は、アプリケーションに対するアクセスを付与および制限できます。

アプリケーションへのアクセスを制限するには、アプリケーションの `app.xml` ファイルに含まれている `authTypes` タグに権利を指定します。アプリケーションの `app.xml` ファイルは、`installation-location/WEB-INF/` サブディレクトリにあります。通常、このディレクトリは `/usr/share/webconsole/webapps/app-context-name/WEB-INF` になります。

アプリケーションのファイルが通常の場合に存在しない場合にそれらの格納場所を見つけるには、次のコマンドを使用します。

```
wcadmin list --detail -a
```

このコマンドは配備済みのすべてのアプリケーションを一覧表示しますが、その際、各アプリケーションの配備時刻とベースディレクトリへのパスが表示されます。`app.xml` ファイルはそのベースディレクトリ内のサブディレクトリ `WEB-INF` に格納されています。

詳細は、[85 ページの「authTypes タグを使って承認を指定する」](#)を参照してください。

アプリケーションの特権

アプリケーションのリンクが Java Web Console の起動ページに表示される場合は、そのアプリケーションを実行できます。ただし、アプリケーションが認証ユーザーや役割の識別情報に基づいて承認チェックを別に行うことがあります。これらのチェックは、`authTypes` タグで制御されるのではなく、アプリケーション自体に明示的にコード化されます。たとえば、アプリケーションによって、読み取りアクセスはすべての認証ユーザーに付与されるが、更新アクセスは一部のユーザーまたは一部の役割に制限されるようにすることができます。

リモートシステムへのアプリケーションのアクセス

適切な資格をすべて持っているからといって、アプリケーションの操作の範囲内にあるすべてのシステムをそのアプリケーションで管理できるとは限りません。Java Web Console アプリケーションを使用して管理する各システムには、専用のセキュリティードメインがあります。Web コンソールシステムに対する読み取り/書き込み権を持っていても、それらの資格がほかのリモートシステムを管理するのに十分であるとは限りません。

一般に、リモートシステムへのアクセスは、セキュリティが Web アプリケーションにどのように実装されるかによって変わります。通常、Web アプリケーションは、アプリケーションに代わって処理を実行する「エージェント」を呼び出します。これらのアプリケーションはその Web コンソールの資格とアプリケーションがエージェントシステムで認識されるときに使われる資格に基づいて認証される必要があります。このエージェントによる認証がどのように行われるかによって、エージェントでもこの認証された識別情報に基づいて承認チェックが行われることがあります。

たとえば、リモート WBEM エージェントを使用する Web アプリケーションでは通常、最初に Java Web Console に対して認証されたユーザーや役割の識別情報を使って認証が行われます。エージェントシステムでこの認証に失敗すると、そのシステムへのアクセスは Web アプリケーションで拒否されます。エージェントシステムで認証に成功しても、エージェントがアクセス制御チェックを行なってアクセスを拒否した場合は、引き続きアクセスが拒否されます。ほとんどのアプリケーションは、ユーザーが Web コンソールでの認証に成功し、正しい役割を引き受けた場合はエージェントでの認証や承認のチェックが失敗することのないように作成されます。

コンソールで使用される内部パスワード

Solaris 10 11/06 リリース以降、Java Web Console は、パスワードで保護された内部ユーザー名を使用して、基になる Web サーバーに対する管理タスクを実行したり、キーストアファイルやトラストストアファイルを暗号化したりします。コンソールをインストールできるよう、パスワードは初期値に設定されます。セキュリティ侵入の可能性を低減するために、インストール後はパスワードを変更するようにしてください。詳細は、[88 ページの「Java Web Console の内部パスワードを変更する」](#)を参照してください。

authTypes タグを使って承認を指定する

ほとんどのシステム管理用の Web アプリケーションでは authTypes タグの使用に管理者が介入する必要はありませんが、場合によってはシステム管理者がこのタグの値を変更する必要があります。authTypes タグには、ユーザーが Java Web Console でアプリケーションを表示するのに必要な承認レベルが記述された情報の組が含まれています。Web コンソールでは、アプリケーション app.xml ファイル内の承認要件

に基づいて、特定のアプリケーションを表示する権限がユーザーにあるかどうかを判定します。各アプリケーションでは、ユーザーがアプリケーションを実行するための適切な承認を受ける必要があるかどうかを判定できます。この判定は、アプリケーションのインストールプロセスの一環として行われることがあります。あるいは、ユーザーが独自のセキュリティ要件に応じて情報を提供する必要がある場合もあります。アプリケーションの製品マニュアルには、特定のアクセス権を指定する必要があるかどうかを判断するのに必要な情報が記載されています。

authTypes タグの内部にいくつかの authType タグを入れ子にすることができます。

authTypes タグには、次の必要事項が指定された authType タグが少なくとも 1 つ含まれている必要があります。

- 実行する承認チェックのタイプ
- Permission サブクラス名
- Permission サブクラスのインスタンスの作成に必要なパラメータ

次の例では、authType タグに属性 name が指定されています。必要となる name 属性は、承認サービスタイプの名前です。それぞれの承認タイプでは、classType と permissionParam の各タグに異なる値が必要となる場合があります。

```
<authTypes>
  <authType name="SolarisRbac">
    <classType>
      com.sun.management.solaris.RbacPermission
    </classType>
    <permissionParam name="permission">
      solaris.admin.serialmgr.read
    </permissionParam>
  </authType>
</authTypes>
```

次の表に、authType タグの内部で入れ子にできるタグを示します。

表 3-1 入れ子の authType タグ

タグ	属性	説明
classType		Permission サブクラス名。このタグは必須です。
permissionParam	name	classType で指定されたクラスのインスタンスを作成するのに必要となるパラメータ。

authTypes タグと入れ子の authType タグは、app.xml ファイルに必ず含まれる要素です。だれでも使用できるアプリケーションを登録する場合は、次の例に示すように、空の authType タグを指定します。

```
<authTypes>
  <authType name="">
    <classType></classType>
    <permissionParam name=""></permissionParam>
  </authType>
</authTypes>
```

Java Web Console へのリモートアクセスを使用可能にする

コンソールが実行されているシステムにログインして `https://localhost:6789` の URL を使用する方法でしかコンソールに接続できない場合、そのシステムではリモートアクセスを防止する設定が使用されています。Solaris 10 11/06 リリース以降では、次の手順を使用して、ほかのアクセス制限はそのまま、コンソールへのリモートアクセスだけを使用可能にすることができます。

▼ Java Web Console へのリモートアクセスを使用可能にする方法

- 1 コンソールが実行されているシステムで、スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 コンソールサーバーがネットワーク要求に応答できるようにするためのプロパティを設定し、コンソールサーバーを再起動します。

```
# svccfg -s svc:/system/webconsole setprop options/tcp_listen = true
```

```
# smcwebserver restart
```

Java Web Console へのリモートアクセスを使用不可能にする

ユーザーがリモートシステムからコンソールに接続できないようにすることが可能です。Solaris 10 11/06 リリース以降では、次の手順を使用して、ほかのアクセス権はそのまま、コンソールへのリモートアクセスだけを使用不可能にすることができます。

▼ Java Web Console へのリモートアクセスを使用不可能にする方法

- 1 コンソールが実行されているシステムで、スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 コンソールサーバーがネットワーク要求に応答しないようにするためのプロパティを設定し、コンソールサーバーを再起動します。

```
# svccfg -s svc:/system/webconsole setprop options/tcp_listen = false
```

```
# smcwebserver restart
```

再起動後、コンソールは、コンソールサーバープロセスと同じシステム上のブラウザにしか応答しなくなります。ブラウザでプロキシを使用することはできません。使用できるのは直接接続のみです。また、<https://localhost:6789/> URL を使ってコンソールにアクセスすることもできます。

Java Web Console の内部パスワードを変更する

Solaris 10 11/06 リリース以降、コンソールではいくつかの内部ユーザー名とパスワードが使用されます。コンソールの内部ユーザー名とパスワードは、コンソールのフレームワークでのみ使用され、ユーザーやシステム管理者が直接使用することはありません。ただし、パスワードが知られると、悪意のあるユーザーがコンソールアプリケーションに干渉する可能性があります。そのようなセキュリティ侵入の可能性を低減するために、パスワードを変更するようにしてください。新しいパスワードはソフトウェアの内部で使用されるため、ユーザーがパスワードを覚える必要はありません。

▼ コンソールの内部パスワードを変更する方法

パスワードは、管理パスワード、キーストアパスワード、およびトラストストアパスワードと呼ばれます。デフォルトの初期値を知らなくてもパスワードを変更することができます。この手順では、3つのパスワードすべてを個別のコマンドで変更する方法について説明します。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 管理パスワードを変更します。

```
# wcadm password -a
```


新しいパスワードを2回入力するよう求められます。パスワードは8文字から32文字で指定してください。

3 キーストアパスワードを変更します。

```
# wcaadmin password -k
```

新しいパスワードを2回入力するよう求められます。パスワードは8文字から32文字で指定してください。

4 トラストストアパスワードを変更します。

```
# wcaadmin password -t
```

新しいパスワードを2回入力するよう求められます。パスワードは8文字から32文字で指定してください。

ユーザーアカウントとグループの管理 (概要)

この章では、ユーザーアカウントとグループを管理するためのガイドラインと管理計画の概要について説明します。また、ユーザーの作業環境のカスタマイズについても説明します。

この章の内容は以下のとおりです。

- 91 ページの「ユーザーとグループの管理における新機能」
- 92 ページの「ユーザーアカウントとグループとは」
- 101 ページの「ユーザーアカウントとグループ情報の格納場所」
- 106 ページの「ユーザーアカウントとグループを管理するツール」
- 110 ページの「ユーザーの作業環境のカスタマイズ」

ユーザーアカウントとグループの管理手順については、[第5章「ユーザーアカウントとグループの管理\(手順\)」](#)を参照してください。

ユーザーとグループの管理における新機能

この節では、この Solaris リリースで新たに追加または変更された、ユーザーとグループの管理機能について説明します。

この Solaris リリースで新たに追加または変更された機能はありません。

Solaris の新機能の一覧および Solaris リリースについての説明は、『[Solaris 10 の概要](#)』を参照してください。

ユーザーアカウントとグループアカウントの管理用のツール

次の表に、ユーザーアカウントとグループの管理に使用できるツールを示します。

表 4-1 ユーザーアカウントとグループの管理用のツール

ツール名	説明	詳細
Solaris 管理コンソール	ユーザー、グループ、役割、権利、メーリングリスト、ディスク、端末、およびモデムを管理するためのグラフィカルツール。	123 ページの「ユーザーアカウントの設定 (作業マップ)」
smuser、smrole、smgroup	ユーザー、グループ、および役割を管理するためのコマンド。これらのコマンドを使用するには、SMC サービスを実行しておく必要があります。	130 ページの「smgroup および smuser コマンドを使ってグループやユーザーを追加する」
useradd、groupadd、roleadd、usermod、groupmod、rolemod、userdel、groupdel、roledel	ユーザー、グループ、および役割を管理するためのコマンド。	130 ページの「groupadd および useradd コマンドを使ってグループやユーザーを追加する」

次のツールを使ってグループを追加できます。

Solaris 管理コンソールのグループツール

注 - この Solaris リリースでは Admintool は使用できません。

ユーザーアカウントとグループとは

基本的なシステム管理作業の 1 つに、サイトの各ユーザーにユーザーアカウントを設定することがあります。通常ユーザーアカウントには、ユーザーがスーパーユーザーのパスワードを知らなくても、システムにログインして、システムを使用するのに必要な情報が含まれています。ユーザーアカウント情報の構成要素については、[93 ページの「ユーザーアカウントの構成要素」](#)で説明します。

ユーザーアカウントを設定するときに、ユーザーをあらかじめ定義されたユーザーグループに追加できます。グループは一般に、ファイルまたはディレクトリへのグループアクセス権を設定して、グループ内のユーザーだけがファイルとディレクトリにアクセスできるようにするために使用されます。

たとえば、ごく少数のユーザーだけにアクセスさせたい機密ファイルを入れるディレクトリを作成できます。topsecret プロジェクトに携わるユーザーを含む topsecret という名前のグループを設定します。そして、topsecret ファイルの読み取り権を topsecret グループに対して設定します。こうすれば、topsecret グループ内のユーザーだけが、ファイルを読み取ることができます。

また、「役割」という特別な種類のユーザーアカウントは、指定したユーザーに特別な特権を与えるときに使用します。詳細は、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[役割によるアクセス制御 \(概要\)](#)」を参照してください。

ユーザーアカウントの構成要素

次の節では、ユーザーアカウントのそれぞれの構成要素について説明します。

ユーザー (ログイン) 名

ユーザーは、ユーザー名 (ログイン名とも呼ばれる) を使って、適切なアクセス権を持つ自分のシステムとリモートシステムにアクセスできます。作成するユーザーアカウントそれぞれに、ユーザー名を選択しなければなりません。

ユーザー名を探しやすいように、ユーザー名の標準的な割り当て方法を使用することを検討してください。また、ユーザー名はユーザーが覚えやすいものにしてください。単純な規則の例としては、ユーザーのファーストネームの頭文字とラストネームの最初の 7 文字を使用します。たとえば、Ziggy Ignatz は zignatz になります。ほかのユーザー名と重複する場合は、ユーザーのファーストネームの頭文字、ミドルネームの頭文字、ラストネームの最初の 6 文字を使用します。たとえば、Ziggy Top Ignatz は ztignatz になります。

さらに重複する場合、ユーザー名の作成には次の方法を検討してください。

- ファーストネームの頭文字、ミドルネームの頭文字、ラストネームの最初の 5 文字を使用します
- 固有の名前になるまで 1、2、3 などの数字を使用します

注 - それぞれの新しいユーザー名は、システムまたは NIS や NIS+ のドメインに登録されているメール別名 (エイリアス) とは異なるものでなければなりません。そうしないと、メールは実際のユーザーではなく別名に送られることがあります。

ユーザー (ログイン) 名の設定方法の詳しいガイドラインについては、[100 ページ](#)の「[ユーザー名、ユーザー ID、およびグループ ID を使用するガイドライン](#)」を参照してください。

ユーザー ID 番号

ユーザー名に関連するものとして、ユーザー識別 (UID) 番号があります。ユーザーがログインしようとするシステムは、UID 番号によってユーザー名を識別したり、ファイルとディレクトリの所有者を識別したりします。多数の異なるシステム上で、ある個人用にユーザーアカウントを作成する場合は、常に同じユーザー名と ID 番号を使用してください。そうすれば、そのユーザーは、所有権の問題を起こすことなく、システム間で簡単にファイルを移動できます。

UID 番号は、2147483647 以下の整数でなければなりません。UID 番号は、通常のユーザーアカウントと特殊なシステムアカウントに必要です。次の表に、ユーザーアカウントとシステムアカウントに予約されている UID 番号を示します。

表 4-2 予約済みの UID 番号

UID 番号	ユーザー/ログインアカウント	説明
0 - 99	root、daemon、bin、sys など	Solaris OS 用として予約されています
100 - 2147483647	通常のユーザー	汎用アカウント
60001 と 65534	nobody および nobody4	匿名ユーザー
60002	noaccess	アクセス権のないユーザー

0 - 99 の UID 番号を割り当てないでください。これらの UID は、Solaris オペレーティングシステムによる割り当て用として予約されています。システム上の定義により、root には常に UID 0、daemon には UID 1、擬似ユーザー bin には UID 2 が設定されます。また、UID が passwd ファイルの先頭にくるように、uucp ログインや、who、tty、ttytype などの擬似ユーザーログインには低い UID を指定するようにしてください。

UID の設定方法の詳しいガイドラインについては、[100 ページ](#)の「[ユーザー名、ユーザー ID、およびグループ ID を使用するガイドライン](#)」を参照してください。

ユーザー (ログイン) 名と同様に、固有の UID 番号を割り当てる方法を決めてください。企業によっては、従業員に固有の番号を割り当て、管理者がその従業員番号にある番号を加えて固有の UID 番号を作成している場合もあります。

セキュリティ上のリスクを最小限に抑えるため、削除したアカウントの UID を再利用することは避けてください。どうしても UID を再利用する必要がある場合、はじめから作りなおして、新しいユーザーが前のユーザーの属性に影響されないようにしてください。たとえば、前のユーザーがプリンタの拒否リストに含まれていたためプリンタにアクセスできなかった場合、その属性を新しいユーザーに適用することが正しいとは限りません。

大きな数値のユーザー ID とグループ ID の使用

UID とグループ ID (GID) には、符号付き整数の最大値 (つまり 2147483647) までの数値を割り当てることができます。

ただし、60000 より大きな数値の UID と GID は機能的に完全でなく、多くの Solaris の機能と互換性がありません。したがって、60000 を超える UID と GID は使わないようにしてください。

次の表では、Solaris の以前のリリースとの相互運用性に関する問題について説明します。

表 4-3 60000 より大きな数値の UID または GID の相互運用性に関する問題

カテゴリ	製品またはコマンド	問題
NFS 互換性	SunOS 4.0 NFS ソフトウェアとその互換リリース	NFS サーバーとクライアントのコードは、大きな UID と GID を 16 ビットに切り捨てます。この状況では、SunOS 4.0 およびその互換リリースが動作しているシステムが、大きな UID と GID を使用している環境で使用されると、セキュリティの問題が発生する可能性があります。この問題を回避するには、SunOS 4.0 およびその互換リリースが動作しているシステムにパッチをあてる必要があります。
ネームサービスの互換性	NIS ネームサービスおよびファイルベースのネームサービス	60000 より大きな数値の UID を持つユーザーは、Solaris 2.5 (およびその互換リリース) が動作しているシステムでは、ログインしたり、su コマンドを使用したりできますが、そのユーザーの UID と GID は 60001(nobody) に設定されます。
	NIS+ ネームサービス	60000 より大きな数値の UID を持つユーザーは、Solaris 2.5 (およびその互換リリース) と NIS+ ネームサービスが動作しているシステムではアクセスが拒否されます。

表 4-4 大きな UID または GID の制限の要約

UID または GID の値	制限
60003 以上	Solaris 2.5 (およびその互換リリース) と NIS ネームサービスまたはファイルネームサービスが動作しているシステムにログインしたユーザーは、nobody の UID および GID を取得します。

表 4-4 大きな UID または GID の制限の要約 (続き)

UID または GID の値	制限
65535 以上	■ Solaris 2.5 (およびその互換リリース) が NFS バージョン 2 ソフトウェアとともに動作しているシステムでは、UID が 16 ビットに切り捨てられるため、セキュリティ上の問題が発生する可能性があります。 ■ ユーザーがデフォルトのアーカイブフォーマットで <code>cpio</code> コマンドを使用してファイルをコピーすると、ファイルごとにエラーメッセージが表示されます。そして、UID と GID はアーカイブにおいて <code>nobody</code> に設定されます。 ■ x86 システム: ユーザーが SVR3 互換のアプリケーションを実行すると、一部のシステムコールから <code>EOVERFLOW</code> が返される場合があります。 ■ x86 システム: ユーザーがマウントされた System V ファイルシステムでファイルまたはディレクトリを作成しようとした場合、System V ファイルシステムは <code>EOVERFLOW</code> エラーを返します。
100000 以上	<code>ps -l</code> コマンドは最大 5 桁の UID を表示します。したがって、99999 より大きな UID または GID が含まれるときは、出力される列が揃いません。
262144 以上	ユーザーが <code>cpio</code> コマンドに <code>-H odc</code> を付けた形式または <code>pax -x cpio</code> コマンドを使用してファイルをコピーすると、ファイルごとにエラーメッセージが返されます。そして、UID と GID はアーカイブにおいて <code>nobody</code> に設定されます。
1000000 以上	ユーザーが <code>ar</code> コマンドを使用すると、そのユーザーの UID と GID はアーカイブにおいて <code>nobody</code> に設定されます。
2097152 以上	ユーザーが <code>tar</code> コマンド、 <code>cpio -H ustar</code> コマンド、または <code>pax -x tar</code> コマンドを使用すると、そのユーザーの UID と GID は <code>nobody</code> に設定されます。

UNIX グループ

「グループ」とは、ファイルやその他のシステム資源を共有できるユーザーの集合のことです。たとえば、同じプロジェクトで作業するユーザーはグループを構成することになります。グループは、従来の UNIX グループのことです。

各グループには、名前、グループ識別 (GID) 番号、およびそのグループに属しているユーザー名のリストが必要です。システムは GID 番号によって内部的にグループを識別します。

ユーザーは次の2つの種類のグループに所属できます。

- 一次グループ - オペレーティングシステムが、ユーザーによって作成されたファイルに割り当てるグループです。各ユーザーは、1つの一次グループに所属していなければなりません。
- 二次グループ - ユーザーが所属する1つまたは複数のグループです。ユーザーは、最大15個の二次グループに所属できます。

グループ名の設定方法の詳しいガイドラインについては、[100 ページ](#)の「[ユーザー名、ユーザー ID、およびグループ ID を使用するガイドライン](#)」を参照してください。

ユーザーの二次グループは、場合によっては重要でないことがあります。たとえば、ファイルの所有権は、一次グループだけが反映し、二次グループは反映しません。ただし、アプリケーションによってはユーザーの二次グループが関係することがあります。たとえば、ユーザーは以前の Solaris リリースで `Admintool` ソフトウェアを使用するとき `sysadmin` グループ (グループ 14) のメンバーでなければなりません。グループ 14 がそのユーザーの現在の一次グループであるかどうかは問題にはなりません。

`groups` コマンドを使用すると、ユーザーが所属しているグループのリストを表示できます。ユーザーは一度に1つの一次グループにしか所属できません。ただし、`newgrp` コマンドを使用して、ユーザーがメンバーとなっているほかのグループに一時的に一次グループを変更することはできます。

ユーザーアカウントを追加するときは、ユーザーに一次グループを割り当てるか、デフォルトの `staff` グループ (グループ 10) を使用する必要があります。一次グループは、すでに存在しているものでなければなりません。存在しない場合は、GID 番号でグループを指定します。ユーザー名は、一次グループに追加されません。ユーザー名が一次グループに追加されると、リストが長くなりすぎるからです。ユーザーを新しい二次グループに割り当てる前に、そのグループを作成し、それに GID 番号を割り当てなければなりません。

グループは、システムにとってローカルにすることも、ネームサービスを介して管理することもできます。グループ管理を簡単に行うには、NIS などのネームサービスや LDAP などのディレクトリサービスを使用する必要があります。これらのサービスを使用すると、グループのメンバーを一元管理できます。

ユーザーパスワード

ユーザーを追加するときにそのユーザーのパスワードを指定できます。または、ユーザーが最初にログインするときにパスワードを指定するよう強制できます。

ユーザーのパスワードは、次の構文に準拠している必要があります。

- パスワード長は、`/etc/default/passwd` ファイル内の `PASSLENGTH` 変数に指定された値に一致している必要があります。 `PASSLENGTH` にはデフォルトで6が設定されています。
- パスワードの最初の6文字には、2文字以上の英字と1文字以上の数字または特殊文字を含める必要があります。
- 最大パスワード長を9文字以上に増やすには、9文字以上に対応しているアルゴリズムで `/etc/policy.conf` ファイルを構成します。

ユーザー名は公表されますが、パスワードを知っているのは各ユーザーだけでなければなりません。各ユーザーアカウントには、パスワードを割り当てる必要があります。パスワードには、6-8文字の英数字と特殊文字を組み合わせることができます。

コンピュータシステムのセキュリティを強化するには、ユーザーのパスワードを定期的に変更する必要があります。高いレベルのセキュリティを確保するには、ユーザーに6週間ごとにパスワードを変更するよう要求してください。低いレベルのセキュリティなら、3ヵ月に1度で十分です。システム管理用のログイン (`root` や `sys` など) は、毎月変更するか、`root` のパスワードを知っている人が退職したり交替したりするたびに交換してください。

コンピュータセキュリティが破られる原因の多くは、正当なユーザーのパスワードが解読される場合です。ユーザーについて何か知っている人が簡単に推測できるような固有名詞、名前、ログイン名、パスワードを使わないよう各ユーザーに対して指示してください。

良いパスワードの例としては次のようなものが考えられます。

- 単語を組み合わせたフレーズ (たとえば、`beammeup`)。
- フレーズ内の各単語の頭文字だけを集めた、意味のない文字列。たとえば、`SomeWhere Over The RainBow` から取った `swotrB`。
- 文字を数字や記号に代えた単語。たとえば、`snoopy` から文字を代えた `sn00py`。

次のものは、パスワードに不適当です。

- 自分の名前そのもの、逆読み、飛ばし読みのもの
- 家族やペットの名前
- 免許証番号
- 電話番号
- 社会保険番号
- 従業員番号
- 趣味や興味に関連した単語
- 季節に関係のある名前 (たとえば12月に `Santa` を使うなど)
- 辞書にある単語

ホームディレクトリ

ホームディレクトリは、ユーザーが独自のファイルを格納するのに割り当てられるファイルシステムの一部分です。ホームディレクトリに割り当てる大きさは、ユーザーが作成するファイルの種類、サイズ、および数によって異なります。

ホームディレクトリは、ユーザーのローカルシステムまたはリモートファイルサーバーのどちらにでも配置できます。どちらの場合も、慣例により、ホームディレクトリは `/export/home/username` として作成します。大規模なサイトでは、ホームディレクトリをサーバーに格納してください。ホームディレクトリのバックアップおよび復元を簡単にするには、`/export/home` ディレクトリごとに別々のファイルシステムを使用してください。たとえば、`/export/home1` と `/export/home2` を使用します。

ホームディレクトリが配置される場所に関係なく、ユーザーは通常 `/home/username` という名前のマウントポイントを介してホームディレクトリにアクセスします。Autofs を使用してホームディレクトリがマウントされていると、どのシステムでも `/home` マウントポイントの下にディレクトリを作成することは許可されません。Autofs が使用されていると、システムはマウントされている `/home` を特別なものと認識します。ホームディレクトリを自動マウントする方法については、『[Solaris のシステム管理 \(ネットワークサービス\)](#)』の「[autofs 管理作業の概要](#)」を参照してください。

ネットワーク上の任意の場所からホームディレクトリを使用するには、`/export/home/username` ではなく、常に `$HOME` という環境変数の値によって参照するようにしてください。前者はマシンに固有の指定です。さらに、ユーザーのホームディレクトリで作成されるシンボリックリンクはすべて相対パス (たとえば `.././.././x/y/x`) を使用する必要があります。こうすることによって、そのリンクはどのシステムにホームディレクトリがマウントされても有効になります。

ネームサービス

大規模サイトのユーザーアカウントを管理する場合は、LDAP、NIS、NIS+ などのネームサービスまたはディレクトリサービスの使用を検討することをお勧めします。ネームサービスまたはディレクトリサービスを使うと、ユーザーアカウント情報を各システムの `/etc` 内のファイルに格納するのではなく、一元管理できます。ユーザーアカウントにネームサービスまたはディレクトリサービスを使用すると、サイト全体のユーザーアカウント情報をシステムごとにコピーしなくても、同じユーザーアカウントのままシステム間を移動できます。ネームサービスまたはディレクトリサービスを使用すると、ユーザーアカウント情報を集中化および一元化して管理できます。

ユーザーの作業環境

ファイルを作成して格納するホームディレクトリのほかに、ユーザーには仕事をするために必要なツールとリソースにアクセスできる環境が必要です。ユーザーがシ

システムにログインすると、初期設定ファイルによってユーザーの作業環境が決定されます。これらのファイルは、ユーザーの起動シェルによって定義されます。起動シェルは Solaris リリースによって異なる可能性があります。

ユーザーの作業環境を管理するのに便利な方法として、カスタマイズしたユーザー初期設定ファイル(.login、.cshrc、.profile など)をユーザーのホームディレクトリに置くという方法があります。

注 - システム初期設定ファイル (/etc/profile または /etc/.login) を使用してユーザーの作業環境を管理しないでください。これらのファイルはローカルシステムに存在するため、一元管理されません。たとえば、Autofs を使用してネットワーク上の任意のシステムからユーザーのホームディレクトリをマウントした場合、ユーザーがシステム間を移動しても環境が変わらないよう保証するには、各システムでシステム初期設定ファイルを修正しなければなりません。

ユーザー初期設定ファイルをユーザー用にカスタマイズする方法については、[110 ページの「ユーザーの作業環境のカスタマイズ」](#)を参照してください。

また、役割によるアクセス制御 (RBAC) でユーザーアカウントをカスタマイズする方法もあります。詳細は、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[役割によるアクセス制御 \(概要\)](#)」を参照してください。

ユーザー名、ユーザー ID、およびグループ ID を使用するガイドライン

ユーザー名、UID、および GID は、複数のドメインにまたがることもあるユーザーの組織内で一意でなければなりません。

ユーザー名または役割名、UID、および GID を作成するときは、次のガイドラインに従ってください。

- ユーザー名 - 2-8 文字の英数字を使用する必要があります。最初の文字は英字でなければならない、少なくとも 1 文字は小文字にする必要があります。

注 - ユーザー名にはピリオド (.)、下線 (_)、ハイフン (-) を使用できますが、これらの文字により障害が発生するソフトウェアもあるため、使用はお勧めできません。

- システムアカウント - デフォルトの /etc/passwd および /etc/group ファイルに含まれているユーザー名、UID、または GID を使用しないでください。0-99 の UID と GID は使用しないでください。これらの番号は、Solaris オペレーティングシス

テムによる割り当て用に予約されており、どのユーザーも使用してはいけません。この制限は、現在使用されていない番号にも適用されます。

たとえば、`gdm` は GNOME ディスプレイマネージャーデーモン用に予約されたユーザー名とグループ名であるため、ほかのユーザーは使用できません。デフォルトの `/etc/passwd` と `/etc/group` のエントリの全リストについては、[表 4-5](#) と [表 4-6](#) を参照してください。

`nobody` と `nobody4` のアカウントは、プロセスの実行には使用しないでください。これらの2つのアカウントは NFS 用に予約されています。これらのアカウントをプロセスの実行に使用すると、予期しないセキュリティ上のリスクにさらされる可能性があります。`root` 以外のユーザーとして実行する必要があるプロセスでは、`daemon` または `noaccess` のアカウントを使用してください。

- システムアカウントの設定 – デフォルトのシステムアカウントの設定は絶対に変更しないでください。この設定には、現在ロックされているシステムアカウントのログインシェルの変更が含まれています。ただし、`root` アカウントのパスワードとパスワード有効期限のパラメータ設定だけはこの規則に当てはまりません。

ユーザーアカウントとグループ情報の格納場所

ユーザーアカウントとグループ情報は、サイトの方針に応じて、次のようにローカルシステムの `/etc` ファイル、ネームサービス、またはディレクトリサービスに格納されます。

- NIS+ ネームサービス情報はテーブルに格納されます。
- NIS ネームサービス情報はマップに格納されます。
- LDAP ディレクトリサービス情報はインデックス付きのデータベースファイルに格納されます。

注 – 混乱を避けるために、ユーザーアカウントとグループ情報の格納場所は、「データベース」、「テーブル」、「マップ」という3種類の呼び方ではなく、単に「ファイル」と呼びます。

ほとんどのユーザーアカウント情報は、`passwd` ファイルに格納されます。パスワード情報は次のように格納されます。

- NIS または NIS+ を使用するときは `passwd` ファイルに
- `/etc` ファイルを使用するときは、`/etc/shadow` ファイルに
- LDAP を使用するときは、`people` コンテナに

パスワード有効期限は、NIS+ または LDAP を使用するときは利用できますが、NIS を使用するときは利用できません。

NIS、NIS+、および files の場合、グループ情報は group ファイルに格納されます。LDAP の場合、グループ情報は group コンテナに格納されます。

passwd ファイルのフィールド

passwd ファイルの各フィールドはコロンで区切られ、次のような情報が入っています。

```
username:password:uid:gid:comment:home-directory:login-shell
```

次に例を示します。

```
kryten:x:101:100:Kryten Series 4000 Mechanoid:/export/home/kryten:/bin/csh
```

passwd ファイルのフィールドの完全な説明については、[passwd\(1\)](#)のマニュアルページを参照してください。

デフォルトの passwd ファイル

Solaris のデフォルトの passwd ファイルには、標準のデーモン用のエントリが入っています。デーモンとは、通常ブート時に起動され、システム全体で有効な操作 (印刷、ネットワーク管理、ポートの監視など) を実行するプロセスのことです。

```
root:x:0:0:Super-User:/:/sbin/sh
daemon:x:1:1:/:
bin:x:2:2:/:usr/bin:
sys:x:3:3:/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
smmsp:x:25:25:SendMail Message Submission Program:/:
listen:x:37:4:Network Admin:/usr/net/nls:
gdm:x:50:50:GDM Reserved UID:/:
webserverd:x:80:80:WebServer Reserved UID:/:
postgres:x:90:90:PostgreSQL Reserved UID:/:usr/bin/pfksh
svctag:x:95:12:Service Tag UID:/:
nobody:x:60001:60001:NFS Anonymous Access User:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
```

表 4-5 デフォルトの `passwd` ファイルのエントリ

ユーザー名	ユーザー ID	説明
root	0	スーパーユーザーのアカウント
デーモン	1	ルーチンシステムタスクに関連するシステム包括デーモン
bin	2	ルーチンシステムタスクを実行するシステムバイナリの実行に関連する管理デーモン
sys	3	システムのログの記録や一時ディレクトリのファイルの更新に関連する管理デーモン
adm	4	システムのログの記録に関連する管理デーモン
lp	71	ラインプリンタのデーモン
uucp	5	uucp 関数に関連するデーモン
nuucp	6	uucp 関数に関連する別のデーモン
smmsp	25	Sendmail メッセージ送信プログラムデーモン
webserverd	80	WebServer アクセス用に予約されたアカウント
postgres	90	PostgreSQL アクセス用に予約されたアカウント
svctag	95	サービスタグレジストリアクセス
gdm	50	GNOME ディスプレイマネージャーデーモン
listen	37	ネットワーク監視デーモン
nobody	60001	匿名の NFS アクセス用に予約されたアカウント
noaccess	60002	あるアプリケーションを経由するが実際にログインせずに、システムにアクセスする必要があるユーザーまたはプロセスに割り当てられます。
nobody4	65534	SunOS 4.0 または 4.1 の nobody ユーザーアカウント

shadow ファイルのフィールド

shadow ファイルの各フィールドはコロンで区切られ、次のような情報が入っています。

username:password:lastchg:min:max:warn:inactive:expire

次に例を示します。

`rimmer:86Kg/MNT/dGu.:8882:0::5:20:8978`

shadow ファイルのフィールドの完全な説明については、[shadow\(4\)](#) と [crypt\(1\)](#) のマニュアルページを参照してください。

group ファイルのフィールド

group ファイルの各フィールドはコロンで区切られ、次のような情報が入っています。

group-name:group-password:gid:user-list

次に例を示します。

```
bin::2:root,bin,daemon
```

group ファイルのフィールドの完全な説明については、[group\(4\)](#) のマニュアルページを参照してください。

デフォルトの group ファイル

Solaris のデフォルトの group ファイルには、システム全体に有効な操作 (印刷、ネットワーク管理、電子メールなど) をサポートする次のようなシステムグループが記述されています。これらのグループの多くは、passwd ファイルのエントリに対応しています。

```
root::0:
other::1:root
bin::2:root,daemon
sys::3:root,bin,adm
adm::4:root,daemon
uucp::5:root
mail::6:root
tty::7:root,adm
lp::8:root,adm
nuucp::9:root
staff::10:
daemon::12:root
sysadmin::14:
smmsp::25:
gdm::50:
webservd::80:
postgres::90:
nobody::60001:
noaccess::60002:
nogroup::65534:
```


表 4-6 デフォルトの group ファイルのエントリ

グループ名	グループ ID	説明
root	0	スーパーユーザーのグループ
other	1	オプションのグループ
bin	2	システムバイナリの実行に関連する管理グループ
sys	3	システムのログの記録や一時ディレクトリに関連する管理グループ
adm	4	システムのログの記録に関連する管理グループ
uucp	5	uucp 関数に関連するグループ
mail	6	電子メールのグループ
tty	7	tty デバイスに関連するグループ
lp	8	ラインプリンタのグループ
nuucp	9	uucp 関数に関連するグループ
staff	10	一般的な管理グループ
デーモン	12	ルーチンシステムタスクに関連するグループ
sysadmin	14	従来の Admintool と Solstice AdminSuite ツールに関連する管理グループ
smmsp	25	Sendmail メッセージ送信プログラム用のデーモン
gdm	50	GNOME ディスプレイマネージャーデーモン用に予約されたグループ
webserverd	80	WebServer アクセス用に予約されたグループ
postgres	90	PostgreSQL アクセス用に予約されたグループ
nobody	60001	匿名の NFS アクセス用に割り当てられたグループ
noaccess	60002	あるアプリケーションを経由するが実際にログインをせずに、システムにアクセスする必要があるユーザーまたはプロセスに割り当てられるグループ
nogroup	65534	既知のグループのメンバーでないユーザーに割り当てられるグループ

ユーザーアカウントとグループを管理するツール

次の表に、ユーザーとグループを管理するための推奨ツールを示します。これらのツールは、Solaris 管理コンソールツール群に含まれています。Solaris 管理コンソールの起動および使用方法については、第2章「Solaris 管理コンソールの操作(手順)」を参照してください。

表 4-7 ユーザーとグループを管理するためのツール

Solaris 管理ツール	目的
ユーザー	ユーザーアカウントを管理します
ユーザーテンプレート	学生、技術者、教師のように特定のユーザーの属性セットを作成します
権限	RBAC 権限を管理します
管理役割	RBAC の管理役割を管理します
グループ	グループ情報を管理します
プロジェクト	プロジェクト情報を管理します
メーリングリスト	メーリングリストを管理します

これらの作業の実行方法については、Solaris 管理コンソールのオンラインヘルプを参照してください。

ユーザーアカウントとグループの管理に使用できる Solaris コマンドについては、表 1-6 を参照してください。これらのコマンドは、認証およびネームサービスサポートを含め、Solaris 管理ツールと同じ機能を提供します。

Solaris ユーザーおよびグループ管理ツールの作業

Solaris ユーザー管理ツールを使用すると、ローカルシステムまたはネームサービス環境のユーザーアカウントとグループを管理できます。

次の表で、ユーザーツールのユーザーアカウント機能を使って実行可能な作業について説明します。

表 4-8 ユーザーアカウントツールの作業の説明

作業	説明
ユーザーの追加	ユーザーをローカルシステムまたはネームサービスに追加します。

表 4-8 ユーザーアカウントツールの作業の説明 (続き)

作業	説明
ユーザーテンプレートの作成	学生、契約者、技術者など、同じグループのユーザーを作成するために、定義済みのユーザー属性のテンプレートを作成します。
ユーザーテンプレートを使用したユーザーの追加	テンプレートを使って、定義済みのユーザー属性を使用してユーザーを追加します。
ユーザーテンプレートの複製	定義済みのユーザー属性を少しだけ変更して使用する場合は、ユーザーテンプレートを複製します。次に、必要な属性のみを変更します。
ユーザープロパティの設定	ユーザーを追加する前にユーザープロパティを設定します。プロパティには、ユーザーの追加時にユーザーテンプレートを使用するかどうかや、ユーザーの削除時にデフォルトでホームディレクトリやメールボックスを削除するかどうかなどの指定があります。
複数ユーザーの追加	テキストファイルを指定する、それぞれの名前を入力する、または自動的に一連のユーザー名を生成することにより、ローカルシステムまたはネームサービスに複数のユーザーを追加します。
ユーザープロパティの表示および変更	ログインシェル、パスワード、またはパスワードオプションなどのユーザープロパティを表示または変更します。
ユーザーへの権限割り当て	特定の管理作業の実行を許可する RBAC 権限をユーザーに割り当てます。
ユーザーの削除	ユーザーをローカルシステムまたはネームサービスから削除します。オプションで、ユーザーのホームディレクトリまたはメールボックスも削除するかどうかを指定できます。ユーザーは、グループまたは役割からも削除されます。

ユーザーをローカルシステムまたはネームサービスに追加する方法については、[92 ページの「ユーザーアカウントとグループとは」](#) および [93 ページの「ユーザーアカウントの構成要素」](#) を参照してください。

表 4-9 権限ツールの作業の説明

作業	説明
権限の付与	管理者だけが実行できた特定のコマンドまたはアプリケーションの実行権限をユーザーに付与します。

表 4-9 権限ツールの作業の説明 (続き)

作業	説明
既存の権限のプロパティの表示および変更	既存の権限を表示または変更します。
承認の追加	承認、つまり役割またはユーザーに個別に付与できる権限を追加します。
承認の表示および変更	既存の承認を表示または変更します。

ユーザーに権限を付与する方法については、『Solaris のシステム管理 (セキュリティサービス)』の「[権利プロファイルの内容](#)」を参照してください。

表 4-10 管理役割ツールの作業の説明

作業	説明
管理の役割の追加	RBAC の承認役割を追加します。
管理の役割への権限の割り当て	作業を実行できるように、役割に特定の権限を割り当てます。
管理の役割の変更	役割に権限を追加したり、役割から権限を削除したりします。

管理役割の使用方法については、『Solaris のシステム管理 (セキュリティサービス)』の「[RBAC の実装を計画する方法](#)」を参照してください。

表 4-11 グループツールの作業の説明

作業	説明
グループの追加	ユーザーを追加する前にグループ名を使用できるように、ローカルシステムまたはネームサービスにグループを追加します。
グループへのユーザーの追加	グループが所有するファイルにユーザーがアクセスする場合、ユーザーをグループに追加します。
グループからのユーザーの削除	ユーザーがグループファイルにアクセスする必要がなくなった場合、グループからユーザーを削除します。

ユーザーをグループに追加する方法については、[96 ページ](#)の「[UNIX グループ](#)」を参照してください。

表 4-12 メーリングリストツールの作業の説明

作業	説明
メーリングリストの作成	メーリングリスト (電子メールのメッセージの宛先のリスト) を作成します。
メーリングリスト名の変更	メーリングリストの作成後、その内容を変更します。
メーリングリストの削除	不要になったメーリングリストを削除します。

メーリングリストの作成方法については、Solaris 管理コンソールのオンラインヘルプを参照してください。

表 4-13 プロジェクトツールの作業の説明

作業	説明
プロジェクトの作成および複製	新しいプロジェクトを作成するか、新しいプロジェクトに必要な属性が酷似している既存のプロジェクトがある場合は、そのプロジェクトを複製します。
プロジェクト属性の変更および表示	既存のプロジェクト属性を表示または変更します。
プロジェクトの削除	不要になったプロジェクトを削除します。

プロジェクトでユーザーおよびリソースを管理する

Solaris 9 リリース以降、ユーザーおよびグループを「プロジェクト」(システム使用率またはリソースアロケーションチャージバックの基礎として使用される、作業負荷の構成要素を示す識別子) のメンバーにすることができます。プロジェクトは、Solaris リソース管理機能の一部で、システムリソースの管理に使用されます。

Solaris 9 リリースを実行するシステムにログインするには、ユーザーはプロジェクトのメンバーになる必要があります。デフォルトでは、ユーザーは Solaris 9 リリースのインストール時に `group.staff` プロジェクトのメンバーになり、ほかのプロジェクト情報は設定されていません。

ユーザーのプロジェクト情報は、`/etc/project` ファイルに格納され、このファイルは、ローカルシステム (ファイル)、NIS ネームサービス、または LDAP ディレクトリサービスに保存できます。Solaris 管理コンソールを使用すると、プロジェクト情報を管理できます。

`/etc/project` ファイルは、ユーザーがログインするために必要ですが、プロジェクトを使用しない場合は管理する必要はありません。

プロジェクトの使用方法および設定方法については、『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』の第2章「プロジェクトとタスク (概要)」を参照してください。

ユーザーの作業環境のカスタマイズ

ユーザーのホームディレクトリの設定には、ユーザーのログインシェルにユーザー初期設定ファイルを提供することも含まれます。ユーザー初期設定ファイルは、ユーザーがシステムにログインしたあとにユーザーのために作業環境を設定するシェルスクリプトです。基本的にシェルスクリプトで行える処理はどれもユーザー初期設定ファイルで実行できます。主に、ユーザーの検索パス、環境変数、ウィンドウ機能の環境など、ユーザーの作業環境を定義します。次の表に示すように、各ログインシェルには、1つまたは複数の、固有のユーザー初期設定ファイルがあります。

表 4-14 Bourne、C、Korn シェルのユーザー初期設定ファイル

シェル	初期設定ファイル	目的
Bourne	<code>\$HOME/.profile</code>	ログイン時のユーザー環境を定義します
C	<code>\$HOME/.cshrc</code>	すべての C シェルに対するユーザー環境の定義で、ログインシェルのあとに起動されます
	<code>\$HOME/.login</code>	ログイン時のユーザー環境を定義します
Korn	<code>\$HOME/.profile</code>	ログイン時のユーザー環境を定義します
	<code>\$HOME/.ENV</code>	ログイン時のユーザー環境の定義で、Korn シェルの <code>ENV</code> 環境変数によって指定されます

Solaris 環境には、次の表に示すように、各システムの `/etc/skel` ディレクトリに、各シェル用のデフォルトのユーザー初期設定ファイルが用意されています。

表 4-15 デフォルトのユーザー初期設定ファイル

シェル	デフォルトファイル
C	<code>/etc/skel/local.login</code>
	<code>/etc/skel/local.cshrc</code>
Bourne または Korn	<code>/etc/skel/local.profile</code>

これらのファイルを変更して、すべてのユーザーに共通の作業環境を提供する標準のファイルセットを作成できます。または、異なるタイプのユーザーに作業環境を

提供することもできます。ユーザーツールを使って、カスタマイズしたユーザー初期設定ファイルを作成することはできませんが、指定された「スケルトン」ディレクトリ内のユーザー初期設定ファイルでユーザーのホームディレクトリを生成することができます。このためには、ユーザーテンプレートツールを使ってユーザーテンプレートを作成し、コピーするユーザー初期設定ファイルを保存するスケルトンディレクトリを指定します。

異なるタイプのユーザーにユーザー初期設定ファイルを作成する手順については、[125 ページの「ユーザー初期設定ファイルをカスタマイズする方法」](#)を参照してください。

ユーザーツールで新しいユーザーアカウントを作成して、ホームディレクトリを作成するオプションを選択すると、選択したログインシェルに合わせて次のファイルが作成されます。

表 4-16 ユーザーの追加時にユーザーツールによって作成されるファイル

シェル	作成されるファイル
C	/etc/skel/local.cshrc ファイルと /etc/skel/local.login ファイルがユーザーのホームディレクトリにコピーされ、それぞれ、.cshrc と .login という名前に変更されます。
Bourne と Korn	/etc/skel/local.profile ファイルがユーザーのホームディレクトリにコピーされ、.profile という名前に変更されます。

サイト初期設定ファイルの使用方法

ユーザー初期設定ファイルは、管理者とユーザーの両者によってカスタマイズできます。この重要な機能は、「サイト初期設定ファイル」と呼ばれる、大域的に配布されるユーザー初期設定ファイルによって実現します。サイト初期設定ファイルを使用して、ユーザーの作業環境に新しい機能を絶えず導入でき、しかもユーザーはユーザー初期設定ファイルをカスタマイズすることもできます。

ユーザー初期設定ファイルでサイト初期設定ファイルを参照するとき、サイト初期設定ファイルに対して行なったすべての更新は、ユーザーがシステムにログインするときかユーザーが新しいシェルを起動するとき自動的に反映されます。サイト初期設定ファイルは、ユーザーを追加したときにはなかったサイト全体の変更をユーザーの作業環境に配布するよう設計されています。

ユーザー初期設定ファイルでできるカスタマイズは、サイト初期設定ファイルでも行えます。これらのファイルは通常はサーバー、またはサーバーのグループにあり、ユーザー初期設定ファイルの最初の行に現れます。また、各サイト初期設定ファイルは、それを参照するユーザー初期設定ファイルと同じ型のシェルスクリプトでなければなりません。

C シェルのユーザー初期設定ファイルでサイト初期設定ファイルを参照するには、ユーザー初期設定ファイルの先頭に次のような行を挿入します。

```
source /net/machine-name/export/site-files/site-init-file
```

Bourne シェルまたは Korn シェルのユーザー初期設定ファイルでサイト初期設定ファイルを参照するには、ユーザー初期設定ファイルの先頭に次のような行を挿入します。

```
. /net/machine-name/export/site-files/site-init-file
```

ローカルシステムへの参照を避ける

ユーザー初期設定ファイルに、ローカルシステムへの個々の参照を追加しないでください。ユーザー初期設定ファイルの設定は、ユーザーがどのシステムにログインしても有効になる必要があります。

次に例を示します。

- ユーザーのホームディレクトリをネットワーク上の任意の位置で利用できるようにするには、常に環境変数の値 `$HOME` を使用してホームディレクトリを参照してください。たとえば、`/export/home/username/bin` ではなく `$HOME/bin` を使用してください。`$HOME` 変数は、ユーザーが別のシステムにログインする場合でも有効で、その場合ホームディレクトリは自動マウントされます。
- ローカルディスクのファイルにアクセスするには、`/net/system-name/directory-name` などの大域パス名を使用してください。システムが AutoFS を実行していれば、`/net/system-name` で参照されるディレクトリはすべてユーザーがログインする任意のシステムに自動的にマウントできます。

シェル機能

次の表に、各シェルの基本的な機能を示します。ユーザー初期設定ファイルを作成するときにどのシェルがどのような機能を提供するか参考にしてください。

表 4-17 Bourne、C、Korn シェルの基本機能

機能	Bourne	C	Korn
UNIX で標準シェルとして知られる	該当する	なし	なし
Bourne シェルと互換性がある構文	-	なし	該当する
ジョブ制御	該当する	該当する	該当する

表 4-17 Bourne、C、Korn シェルの基本機能 (続き)

機能	Bourne	C	Korn
履歴リスト	なし	該当する	該当する
コマンド行の編集	なし	該当する	該当する
別名(エイリアス)	なし	該当する	該当する
ログインディレクトリの1文字省略形	なし	該当する	該当する
ファイルの上書き保護(noclobber)	なし	該当する	該当する
CTRL+D 無視(ignoreeof)	なし	該当する	該当する
拡張 cd コマンド	なし	該当する	該当する
.profile とは別の初期設定ファイル	なし	該当する	該当する
ログアウトファイル	なし	該当する	なし

シェル環境

シェルは、login プログラム、システム初期設定ファイル、ユーザー初期設定ファイルによって定義される変数を含む環境を管理します。また、一部の変数はデフォルトで定義されます。

シェルには次の2種類の変数があります。

- 環境変数 - シェルによって生成されるすべてのプロセスにエクスポートされる変数。環境変数の設定値は env コマンドで表示できます。PATH などを含む環境変数の一部が、シェルそのものの動作に影響を与えます。
- シェル(ローカル)変数 - 現在使用中のシェルだけに関係する変数。C シェルの場合は、シェル変数は環境変数と特別に対応しています。これらのシェル変数は user、term、home、path です。シェル変数は、対応する環境変数の値によって初期設定されます。

C シェルでは、小文字を使って set コマンドでシェル変数を設定し、大文字を使って setenv コマンドで環境変数を設定します。シェル変数を設定すると、対応する環境変数が設定されます。同様に、環境変数を設定すると、対応するシェル変数も変更されます。たとえば、path シェル変数を新しいパスで更新すると、シェルは PATH 環境変数も新しいパスで更新します。

Bourne、Korn の両シェルでは、任意の大文字の変数名を使って、シェル変数と環境変数の両方を設定できます。その後に実行されるコマンドに対して変数を有効にするためには、export コマンドの実行も必要です。

すべてのシェルで、シェル変数と環境変数は一般的に大文字の名前で参照します。

ユーザー初期設定ファイルでは、ユーザーのシェル環境を、あらかじめ定義された変数の値を変更するか、変数を追加することによってカスタマイズできます。次の表に、ユーザー初期設定ファイルで環境変数を設定する方法を示します。

表 4-18 ユーザー初期設定ファイルでの環境変数の設定方法

シェルタイプ	ユーザー初期設定ファイルに追加する行
C シェル	<code>setenv VARIABLE value</code> 例: <code>setenv MAIL /var/mail/ripley</code>
Bourne または Korn シェル	<code>VARIABLE=value ; export VARIABLE</code> 例: <code>MAIL=/var/mail/ripley;export MAIL</code>

次の表では、ユーザー初期設定ファイルでカスタマイズできる環境変数とシェル変数について説明します。各シェルで使用される変数の詳細は、[sh\(1\)](#)、[ksh\(1\)](#)、[csh\(1\)](#) の各マニュアルページを参照してください。

表 4-19 シェル変数と環境変数の説明

変数	説明
CDPATH (C シェルでは <code>cdpath</code>)	cd コマンドで使用する変数を設定します。cd コマンドの対象ディレクトリを相対パス名で指定すると、cd コマンドは対象ディレクトリをまず現在のディレクトリ (.) 内で検索します。対象ディレクトリが見つからない場合、CDPATH 変数内のパス名のリストが順に検索され、見つかると、ディレクトリの変更が行われます。CDPATH で対象ディレクトリが見つからなかった場合は、現在の作業ディレクトリは変更されません。たとえば、CDPATH 変数が <code>/home/jean</code> に設定されており、 <code>/home/jean</code> の下に <code>bin</code> と <code>rje</code> の2つのディレクトリがあるとします。 <code>/home/jean/bin</code> ディレクトリ内で <code>cd rje</code> と入力した場合、フルパスを指定していなくても、ディレクトリが <code>/home/jean/rje</code> に変更されます。
history	C シェルの履歴を設定します。
HOME (C シェルでは <code>home</code>)	ユーザーのホームディレクトリへのパスを設定します。
LANG	ロケールを設定します。
LOGNAME	現在ログインしているユーザーの名前を設定します。LOGNAME のデフォルト値は、 <code>passwd</code> ファイルに指定されているユーザー名にログインプログラムによって自動的に設定されます。この変数は参照用のみ使用し、設定を変更してはいけません。

表 4-19 シェル変数と環境変数の説明 (続き)

変数	説明
LPDEST	ユーザーのデフォルトプリンタを設定します。
MAIL	ユーザーのメールボックスへのパスを設定します。
MANPATH	アクセスできるマニュアルページの階層を設定します。
PATH (C シェルでは path)	ユーザーがコマンドを入力したときに実行するプログラムについて、シェルが検索するディレクトリを順番に指定します。ディレクトリが検索パス上にない場合は、ユーザーはコマンドの絶対パス名を入力しなければなりません。 デフォルトの PATH は、ログインプロセスで .profile (Bourne または Korn シェル) あるいは .cshrc (C シェル) の指定どおりに自動的に定義され、設定されます。 検索パスの順序が重要です。同じコマンドが異なる場所にそれぞれ存在するときは、最初に見つかったコマンドが使用されます。たとえば、PATH が Bourne および Korn シェル構文で PATH=/bin:/usr/bin:/usr/sbin:\$HOME/bin のように定義されていて、sample というファイルが /usr/bin と /home/jean/bin の両方にあるものとします。ユーザーが sample コマンドを、その絶対パスを指定しないで入力した場合は、/usr/bin で見つかったバージョンが使用されます。
prompt	C シェルのシェルプロンプトを設定します。
PS1	Bourne または Korn シェルのシェルプロンプトを設定します。
SHELL (C シェルでは shell)	make、vi、その他のツールが使うデフォルトシェルを設定します。
TERMINFO	代替の terminfo データベースが保存されているディレクトリに名前を付けます。/etc/profile または /etc/.login ファイルで TERMINFO 変数を使用します。詳細は、terminfo(4) のマニュアルページを参照してください。 TERMINFO 環境変数を設定すると、システムはまずユーザーが定義した TERMINFO パスを調べます。ユーザーが定義した TERMINFO ディレクトリ内に端末の定義が見つからなかった場合は、システムはデフォルトディレクトリ /usr/share/lib/terminfo で定義を探します。どちらにも見つからなかった場合、端末は dumb として定義されます。
TERM (C シェルでは term)	端末を設定します。この変数は /etc/profile または /etc/.login ファイルで再設定する必要があります。ユーザーがエディタを起動すると、システムはこの環境変数で定義される名前と同じ名前のファイルを探します。システムは、TERMINFO が参照するディレクトリ内を探して端末の特性を知ります。
TZ	時間帯を設定します。時間帯は、たとえば ls -l コマンドで日付を表示する場合に使われます。TZ をユーザーの環境に設定しないと、システムの設定が使用されます。設定する場合、グリニッジ標準時が使用されます。

PATH 変数

ユーザーが絶対パス名でコマンドを入力すると、シェルはそのパス名を使ってコマンドを探します。ユーザーがコマンド名しか指定しないと、シェルは PATH 変数で指定されているディレクトリの順でコマンドを探します。コマンドがいずれかのディレクトリで見つければ、シェルはコマンドを実行します。

デフォルトのパスがシステムで設定されますが、大部分のユーザーはそれを変更してほかのコマンドディレクトリを追加します。環境の設定や、正しいバージョンのコマンドまたはツールへのアクセスに関連して発生するユーザーの問題の多くは、パス定義の誤りが原因です。

パスの設定のガイドライン

次に、効率的な PATH 変数を設定するためのガイドラインをいくつか示します。

- セキュリティーが特に問題とならない場合は、現在の作業ディレクトリ (.) をパスの最初に指定します。しかし、現在の作業ディレクトリをパスに入れると、セキュリティ上の問題となることがあり、特にスーパーユーザーにとって問題となります。
- 検索パスはできるだけ短くしておきます。シェルはパスで各ディレクトリを探します。コマンドが見つからないと、検索に時間がかかり、システムのパフォーマンスが低下します。
- 検索パスは左から右に読まれるため、通常使用するコマンドをパスの初めの方に指定するようにしてください。
- パスでディレクトリが重複していないか確認してください。
- 可能であれば、大きなディレクトリの検索は避けてください。大きなディレクトリはパスの終わりに指定します。
- NFS サーバーが応答しないときに「ハング」の可能性が小さくなるように、NFS がマウントするディレクトリより前にローカルディレクトリを指定します。この方法によって、不要なネットワークトラフィックも減少します。

ユーザーのデフォルトパスの設定

次の例は、ユーザーのデフォルトパスを設定する方法を示しています。

次の例は、ユーザーのデフォルトパスがホームディレクトリとほかの NFS マウントディレクトリを含むように設定する方法を示しています。現在の作業ディレクトリはパスの最初に指定されます。C シェルユーザー初期設定ファイルでは、次の行を追加してください。

```
set path=(. /usr/bin $HOME/bin /net/glrr/files1/bin)
```

Bourne シェルまたは Korn シェルユーザー初期設定ファイルでは、次の行を追加してください。

```
PATH=./usr/bin:/$HOME/bin:/net/grr/files1/bin
export PATH
```

ロケール変数

LANG と LC の各環境変数は、ロケール固有の変換と表記をシェルに指定します。指定できる変換と表記には、時間帯や照合順序、および日付、時間、通貨、番号の書式などがあります。さらに、ユーザー初期設定ファイルで stty コマンドを使って、端末のセッションが複数バイト文字をサポートするかどうかを指定できます。

LANG 変数は、ロケールのすべての変換と表記を設定します。ロケールの各種の設定を個別に行うには、次の LC 変数を使用します。
LC_COLLATE、LC_CTYPE、LC_MESSAGES、LC_NUMERIC、LC_MONETARY、および LC_TIME です。

次の表に、LANG と LC 環境変数の値の一部を示します。

表 4-20 LANG と LC 変数の値

値	ロケール
de_DE.ISO8859-1	ドイツ語
en_US.UTF-8	米国英語 (UTF-8)
es_ES.ISO8859-1	スペイン語
fr_FR.ISO8859-1	フランス語
it_IT.ISO8859-1	イタリア語
ja_JP.eucJP	日本語 (EUC)
ko_KR.EUC	韓国語 (EUC)
sv_SE.ISO8859-1	スウェーデン語
zh_CN.EUC	簡体字中国語 (EUC)
zh_TW.EUC	繁体字中国語 (EUC)

サポートされるロケールの詳細については、『国際化対応言語環境の利用ガイド』を参照してください。

例 4-1 LANG 変数によるロケールの設定

次の例は、LANG 環境変数を使ってロケールを設定する方法を示しています。C シェルユーザー初期設定ファイルでは、次の行を追加してください。

```
setenv LANG de_DE.ISO8859-1
```

例 4-1 LANG 変数によるロケールの設定 (続き)

Bourne シェルまたは Korn シェルユーザー初期設定ファイルでは、次の行を追加してください。

```
LANG=de_DE.ISO8859-1; export LANG
```

デフォルトのファイルアクセス権 (umask)

ファイルまたはディレクトリを作成したときに設定されるデフォルトのファイルアクセス権は、「ユーザーマスク」によって制御されます。ユーザーマスクは、初期設定ファイルで `umask` コマンドによって設定されます。現在のユーザーマスクの値は、`umask` と入力して Return キーを押すと表示できます。

ユーザーマスクは、次の 8 進値で構成されます。

- 最初の桁でそのユーザーのアクセス権を設定する
- 2 桁目でグループのアクセス権を設定する
- 3 桁目でその他 (「ワールド」とも呼ばれる) のアクセス権を設定する

最初の桁がゼロの場合、その桁は表示されません。たとえば、ユーザーマスクを 022 に設定すると、22 が表示されます。

設定する `umask` の値は、与えたいアクセス権の値を 666 (ファイルの場合) または 777 (ディレクトリの場合) から引きます。引いた残りが `umask` に使用する値です。たとえば、ファイルのデフォルトモードを 644 (`rw-r--r--`) に変更するとします。このとき 666 と 644 の差 022 が `umask` コマンドの引数として使用する値です。

また、次の表から `umask` 値を決めることもできます。この表は、`umask` の各 8 進値から作成される、ファイルとディレクトリのアクセス権を示しています。

表 4-21 `umask` 値のアクセス権

umask 8 進値	ファイルのアクセス権	ディレクトリアクセス権
0	rw-	rwX
1	rw-	rw-
2	r--	r-X
3	r--	r--
4	-w-	-wX
5	-w-	-w-

表 4-21 umask 値のアクセス権 (続き)

umask 8 進値	ファイルのアクセス権	ディレクトリアクセス権
6	--x	--x
7	--- (なし)	--- (なし)

次の例は、デフォルトのファイルアクセス権を `rw-rw-rw-` に設定します。

```
umask 000
```

ユーザー初期設定ファイルとサイト初期設定ファイルの例

ここでは、ユーザー自身の初期設定ファイルのカスタマイズを始める際に使用するユーザー初期設定ファイルとサイト初期設定ファイルの例を示します。例の中のシステム名やパス名は、実際のサイトに合わせて置き換えてください。

例 4-2 .profile ファイル

```
(Line 1) PATH=$PATH:$HOME/bin:/usr/local/bin:/usr/ccs/bin:..
(Line 2) MAIL=/var/mail/$LOGNAME
(Line 3) NNTPSERVER=server1
(Line 4) MANPATH=/usr/share/man:/usr/local/man
(Line 5) PRINTER=printer1
(Line 6) umask 022
(Line 7) export PATH MAIL NNTPSERVER MANPATH PRINTER
```

1. ユーザーのシェル検索パスを設定する
2. ユーザーのメールファイルへの検索パスを設定する
3. ユーザーの Usenet ニュースサーバーを設定する
4. マニュアルページへのユーザーの検索パスを設定する
5. ユーザーのデフォルトプリンタを設定する
6. ユーザーのデフォルトのファイル作成アクセス権を設定する
7. 指定された環境変数をエクスポートする

例 4-3 .cshrc ファイル

```
(Line 1) set path=($PATH $HOME/bin /usr/local/bin /usr/ccs/bin)
(Line 2) setenv MAIL /var/mail/$LOGNAME
(Line 3) setenv NNTPSERVER server1
(Line 4) setenv PRINTER printer1
(Line 5) alias h history
(Line 6) umask 022
(Line 7) source /net/server2/site-init-files/site.login
```

例4-3 .cshrc ファイル (続き)

1. ユーザーのシェル検索パスを設定する
2. ユーザーのメールファイルへの検索パスを設定する
3. ユーザーの Usenet ニュースサーバーを設定する
4. ユーザーのデフォルトプリンタを設定する
5. history コマンドの別名を作成する。これにより、h と入力するだけで history コマンドを実行できる
6. ユーザーのデフォルトのファイル作成アクセス権を設定する
7. 指定されたサイト初期設定ファイルを読み込む

例4-4 サイト初期設定ファイル

次のサイト初期設定ファイルの例では、ユーザーは特定のバージョンのアプリケーションを選択できます。

```
# @(#)site.login
main:
echo "Application Environment Selection"
echo ""
echo "1. Application, Version 1"
echo "2. Application, Version 2"
echo ""
echo -n "Type 1 or 2 and press Return to set your
application environment: "

set choice = $<

if ( $choice !~ [1-2] ) then
goto main
endif

switch ($choice)

case "1":
setenv APPHOME /opt/app-v.1
breaksw

case "2":
setenv APPHOME /opt/app-v.2
endsw
```

次のようにして、このサイト初期設定ファイルをユーザーの .cshrc ファイル (C シェルユーザーのみ使用可能) で参照させることができます。

例 4-4 サイト初期設定ファイル (続き)

```
source /net/server2/site-init-files/site.login
```

この行では、サイト初期設定ファイルは `site.login` という名前で、`server2` という名前のサーバー上にあります。また、自動マウンタがユーザーのシステムで実行されていることを前提としています。

ユーザーアカウントとグループの管理 (手順)

この章では、ユーザーアカウントとグループを設定し、管理する手順について説明します。

ユーザーアカウントとグループの設定および管理手順については、次の節を参照してください。

- 123 ページの「ユーザーアカウントの設定 (作業マップ)」
- 134 ページの「ユーザーアカウントの管理 (作業マップ)」

ユーザーアカウントとグループの管理の概要については、第 4 章「ユーザーアカウントとグループの管理 (概要)」を参照してください。

ユーザーアカウントの設定 (作業マップ)

作業	説明	参照先
ユーザー情報を収集します。	標準の書式を使ってユーザー情報を収集すると、情報を整理しやすくなります。	124 ページの「ユーザー情報の収集」
ユーザー初期設定ファイルをカスタマイズします。	新規ユーザーに一貫した環境を提供できるようにユーザー初期設定ファイル (.cshrc、.profile、.login) を設定します。	125 ページの「ユーザー初期設定ファイルをカスタマイズする方法」

作業	説明	参照先
グループを追加します。	Solaris コマンド行インタ フェースツール	127 ページの「Solaris 管理コン ソールのグループツールを 使ってグループを追加する方 法」 129 ページの「コマンド行 ツールを使ってグループや ユーザーを追加する」
ユーザーを追加します。	次のツールを使ってユーザーを 追加できます。 Solaris 管理コンソールの ユーザーツール Solaris コマンド行インタ フェースツール	128 ページの「Solaris 管理コン ソールのユーザーツールを 使ってユーザーを追加する方 法」 129 ページの「コマンド行 ツールを使ってグループや ユーザーを追加する」
ユーザーテンプレートを設定し ます。	類似したユーザープロパ ティをすべて手動で追加する 必要がないようにユーザーテン プレートを作成できます。	Solaris 管理コンソールオンラ インヘルプ
ユーザーへの権限または役割を 追加します。	特定のコマンドまたは操作を実 行できるようにユーザーに権限 または役割を追加します。	Solaris 管理コンソールオンラ インヘルプ
ユーザーのホームディレクトリ を共有します。	ユーザーのシステムからその ディレクトリをリモートでマウ ントして、ユーザーのホーム ディレクトリを共有できるよう にします。	131 ページの「ユーザーの ホームディレクトリを共有す る方法」
ユーザーのホームディレクトリ をマウントします。	ユーザーのホームディレクトリ をユーザーのシステムにマウン トします。	132 ページの「ユーザーの ホームディレクトリをマウン トする方法」

ユーザー情報の収集

ユーザーアカウントを追加する前に、ユーザーに関する情報を次のような書式で記録しておくことができます。

項目	説明
ユーザー名:	
役割名	

項目	説明
プロフィールまたは承認	
UID	
一次グループ	
二次グループ	
コメント:	
デフォルトシェル	
パスワードの状態と有効期限	
ホームディレクトリのパス名	
マウント方法	
ホームディレクトリのアクセス権	
メールサーバー	
所属部署	
部署管理者	
管理者	
従業員名	
役職	
資格	
従業員番号	
開始日	
メール別名への追加	
デスクトップシステム名	

▼ ユーザー初期設定ファイルをカスタマイズする方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、[『Solaris のシステム管理 \(セキュリティサービス\)』の「RBAC の構成 \(作業マップ\)」](#)を参照してください。

- 2 各タイプのユーザー用にスケルトンディレクトリを作成します。

```
# mkdir /shared-dir/skel/user-type
```

shared-dir ネットワーク上の別のシステムで利用できるディレクトリの名前。

user-type ユーザーのタイプに応じて初期設定ファイルを格納するディレクトリ
の名前。

- 3 デフォルトのユーザー初期設定ファイルを、異なるタイプのユーザー用に作成したディレクトリにコピーします。

```
# cp /etc/skel/local.cshrc /shared-dir/skel/user-type/.cshrc
```

```
# cp /etc/skel/local.login /shared-dir/skel/user-type/.login
```

```
# cp /etc/skel/local.profile /shared-dir/skel/user-type/.profile
```

注-アカウントにプロファイルが割り当てられている場合、プロファイルに割り当てられた(セキュリティ属性付きの)コマンドを使用するために、ユーザーは特別なバージョンのシェルであるプロファイルシェルを起動する必要があります。シェルの種類に合わせて、`pfsh` (Bourne シェル)、`pfcksh` (C シェル)、および `pfksh` (Korn シェル) の3つの「プロファイルシェル」があります。プロファイルシェルについては、『Solaris のシステム管理 (セキュリティサービス)』の「役割によるアクセス制御 (概要)」を参照してください。

- 4 各ユーザータイプ用にユーザー初期設定ファイルを編集し、必要に応じてカスタマイズします。

ユーザー初期設定ファイルをカスタマイズする方法については、[110 ページ](#)の「ユーザーの作業環境のカスタマイズ」を参照してください。

- 5 ユーザー初期設定ファイルのアクセス権を設定します。

```
# chmod 744 /shared-dir/skel/user-type/*
```

- 6 ユーザー初期設定ファイルのアクセス権が正しいことを確認します。

```
# ls -la /shared-dir/skel/*
```

例 5-1 ユーザー初期設定ファイルをカスタマイズする

次の例では、特定のタイプのユーザー向けの、`/export/skel/enduser` ディレクトリにある C シェルユーザー初期設定ファイルをカスタマイズします。`.cshrc` ファイルの例については、[例 4-3](#) を参照してください。

```
# mkdir /export/skel/enduser
```

```
# cp /etc/skel/local.cshrc /export/skel/enduser/.cshrc
```

(Edit `.cshrc` file)

```
# chmod 744 /export/skel/enduser/.*
```

▼ Solaris 管理コンソールのグループツールを使ってグループを追加する方法

グループを追加するときに既存のユーザーをグループに追加できます。また、グループだけを追加し、その後でユーザーを追加するときに既存のユーザーをそのグループに追加することもできます。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、[『Solaris のシステム管理 \(セキュリティサービス\)』の「RBAC の構成 \(作業マップ\)」](#)を参照してください。

- 2 Solaris 管理コンソールを起動します。

```
# /usr/sadm/bin/smc &
```

Solaris 管理コンソールの起動方法については、[45 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」](#)または [53 ページの「ネームサービス環境で Solaris 管理コンソールを起動する方法」](#)を参照してください。

- 3 ナビゲーション区画の「管理ツール」アイコンの下にある「このコンピュータ」アイコンをクリックします。

カテゴリのリストが表示されます。

- 4 (省略可能) ネームサービス環境に適したツールボックスを選択します。

- 5 「システムの構成」アイコンをクリックします。

- 6 「ユーザー」アイコンをクリックし、スーパーユーザーのパスワードまたは役割のパスワードを入力します。

- 7 「グループ」アイコンをクリックします。「アクション」メニューから「グループを追加」を選択します。

コンテキストヘルプを使用して、グループをシステムに追加します。

- 8 「グループ ID」の下にある「グループ名」プロンプトで、グループ名を指定します。

たとえば、`mechanoids` と指定します。

- 9 「グループ ID 番号」プロンプトで、グループ番号を指定します。
たとえば、GID 101 と指定します。
- 10 「了解 (OK)」をクリックします。

▼ Solaris 管理コンソールのユーザーツールを使ってユーザーを追加する方法

Solaris 管理コンソールのユーザーツールを使ってユーザーを追加するには、次の手順を実行します。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。
- 2 Solaris 管理コンソールを起動します。

```
# /usr/sadm/bin/smc &
```

Solaris 管理コンソールの起動方法については、[45 ページ](#)の「スーパーユーザーまたは役割としてコンソールを起動する方法」または [53 ページ](#)の「ネームサービス環境で Solaris 管理コンソールを起動する方法」を参照してください。
- 3 ナビゲーション区画の「管理ツール」アイコンの下にある「このコンピュータ」アイコンをクリックします。
カテゴリのリストが表示されます。
- 4 (省略可能)ネームサービス環境に適したツールボックスを選択します。
- 5 「システムの構成」アイコンをクリックします。
- 6 「ユーザー」アイコンをクリックし、スーパーユーザーのパスワードまたは役割のパスワードを入力します。
- 7 「ユーザー」アイコンをクリックします。
コンテキストヘルプを使用して、ユーザーをシステムに追加します。

- 8 「アクション」メニューから「ユーザーを追加」⇒「ウィザードを使用」を選択します。
- 次の手順に進むときは「次へ」をクリックしてください。
- a. 「ユーザー名」プロンプトで、ユーザー名またはログイン名を指定します。
たとえば、kryten と指定します。
 - b. (省略可能)「正式名」プロンプトで、ユーザーのフルネームを指定します。
たとえば、kryten series 3000 と指定します。
 - c. (省略可能)「備考欄」プロンプトで、このユーザーの詳細な説明を入力します。
 - d. 「ユーザーID 番号」プロンプトで、ユーザーIDを入力します。
たとえば、1001 と入力します。
 - e. 「ユーザーは最初のログイン時にこのパスワードを使用する必要があります」オプションを選択します。
「パスワード」プロンプトでユーザーのパスワードを入力し、「パスワードを確認」プロンプトでそのパスワードを確認します。
 - f. ユーザーの一次グループを選択します。
たとえば、mechanoids と指定します。
 - g. 「サーバー」と「パス」の各プロンプトでデフォルトを受け入れることにより、ユーザーのホームディレクトリを作成します。
 - h. メールサーバーを指定します。
 - i. 入力した情報を確認し、必要があれば前の画面に戻って情報を訂正します。訂正する必要がない場合は、「完了」をクリックします。

コマンド行ツールを使ってグループやユーザーを追加する

この節では、コマンド行ツールを使ってユーザーやグループを追加する例をいくつか紹介します。

groupadd および useradd コマンドを使ってグループやユーザーを追加する

次の例では、groupadd および useradd の各コマンドを使って、グループ `scutters` やユーザー `scutter1` をローカルシステムのファイルに追加します。なお、これらのコマンドは、ネームサービス環境でユーザーを管理する場合には使用できません。

```
# groupadd -g 102 scutters
# useradd -u 1003 -g 102 -d /export/home/scutter1 -s /bin/csh \
-c "Scutter 1" -m -k /etc/skel scutter1
64 blocks
```

詳細は、[groupadd\(1M\)](#) および [useradd\(1M\)](#) のマニュアルページを参照してください。

smgroup および smuser コマンドを使ってグループやユーザーを追加する

次の例では、smgroup および smuser の各コマンドを使って、グループ `gelfs` やユーザー `camille` をホスト `starlite` の NIS ドメイン `solar.com` に追加します。

```
# /usr/sadm/bin/smgroup add -D nis:/starlitesolar.com -- -g 103 -n gelfs
# /usr/sadm/bin/smuser add -D nis:/starlite/solar.com -- -u 1004
-n camille -c "Camille G." -d /export/home/camille -s /bin/csh -g gelfs
```

詳細は、[smgroup\(1M\)](#) および [smuser\(1M\)](#) のマニュアルページを参照してください。

Solaris 管理コンソールによるホームディレクトリの設定

Solaris 管理コンソールツールを使用してユーザーのホームディレクトリを管理するときは、次のことに注意してください。

- ユーザーツールの「ユーザーを追加」ウィザードを使用してユーザーアカウントを追加し、ユーザーのホームディレクトリを `/export/home/username` として指定すると、ホームディレクトリが自動マウントされるように自動的に設定され、次のエントリが `passwd` ファイルに追加されます。

```
/home/username
```

- ユーザーツールを使用して、ホームディレクトリを自動マウントしない
ユーザーアカウントを設定するには、この機能を無効にするユーザーアカウントのテンプレートを設定し、そのテンプレートを使ってユーザーを追加するほかに方法はありません。「ユーザーを追加」ウィザードでこの機能を無効にすることはできません。
- `-x autohome=N` オプションを指定して `smuser add` コマンドを使用すると、ユーザーのホームディレクトリを自動マウントしないでユーザーを追加できます。ただし、`smuser delete` コマンドには、ユーザーを追加した後でホームディレクトリを削除するオプションはありません。その場合は、ユーザーツールを使用して、ユーザーとユーザーのホームディレクトリを削除する必要があります。

▼ ユーザーのホームディレクトリを共有する方法

ユーザーのホームディレクトリを共有するには、次の手順を実行します。

- 1 ユーザーのホームディレクトリを含むシステムで、スーパーユーザーになるか、同等の役割になります。
- 2 次のように入力して、`mountd` デーモンが動作していることを確認します。
このリリースでは、`mountd` デーモンは NFS サーバーサービスの一環として起動されます。`mountd` デーモンが動作しているかどうかを確認するには、次のコマンドを入力します。


```
# svcs network/nfs/server
```

STATE	STIME	FMRI
online	Aug_26	svc:/network/nfs/server:default
- 3 `mountd` デーモンが動作していない場合は、`mountd` デーモンを起動します。


```
# svcadm network/nfs/server
```
- 4 システム上で共有されているファイルシステムをリスト表示します。


```
# share
```
- 5 ユーザーのホームディレクトリを含むファイルシステムがすでに共有されているかどうかによって、次のいずれかの手順を選択します。
 - a. ユーザーのホームディレクトリがすでに共有されている場合、手順8へ進みます。
 - b. ユーザーのホームディレクトリが共有されていない場合、手順6へ進みます。

- 6 /etc/dfs/dfstab ファイルを編集し、次の行を追加します。

```
share -F nfs /file-system
```

/file-system は、共有するユーザーのホームディレクトリを含むファイルシステムです。慣例上、このファイルシステムは /export/home になります。

- 7 /etc/dfs/dfstab ファイルで指定されたファイルシステムを共有します。

```
# shareall -F nfs
```

このコマンドは、/etc/dfs/dfstab ファイルにある share コマンドをすべて実行するので、システムをリブートする必要はありません。

- 8 ユーザーのホームディレクトリが共有されていることを確認します。

```
# share
```

例 5-2 ユーザーのホームディレクトリを共有する

次の例は、/export/home ディレクトリの共有方法を示しています。

```
# svcsv network/nfs/server
# svcadm network/nfs/server
# share
# vi /etc/dfs/dfstab
```

(The line share -F nfs /export/home is added.)

```
# shareall -F nfs
# share
```

```
-          /usr/dist          ro  ""
-          /export/home/user-name  rw  ""
```

参照 ユーザーのホームディレクトリがユーザーのシステム上にない場合、それが配置されているシステムから、ユーザーのホームディレクトリをマウントしなければなりません。詳細な手順については、[132 ページの「ユーザーのホームディレクトリをマウントする方法」](#)を参照してください。

▼ ユーザーのホームディレクトリをマウントする方法

ホームディレクトリを自動マウントする方法については、『Solaris のシステム管理 (ネットワークサービス)』の「autofs 管理作業の概要」を参照してください。

- 1 ユーザーのホームディレクトリが共有されていることを確認します。
詳細は、[131 ページ](#)の「ユーザーのホームディレクトリを共有する方法」を参照してください。
- 2 ユーザーのシステムにスーパーユーザーとしてログインします。
- 3 `/etc/vfstab` ファイルを編集して、次のようなユーザーのホームディレクトリ用のエントリを作成します。

```
system-name:/export/home/user-name - /export/home/username nfs - yes rw
```

<code>system-name</code>	ホームディレクトリが配置されているシステムの名前。
<code>/export/home/username</code>	共有されるユーザーのホームディレクトリの名前。慣例上、 <code>/export/home/username</code> にユーザーのホームディレクトリが含まれます。ただし、別のファイルシステムを使用しても構いません。
<code>-</code>	エントリに必要な可変部分。
<code>/export/home/username</code>	ユーザーのホームディレクトリがマウントされるディレクトリの名前。

`/etc/vfstab` ファイルにエントリを追加する方法については、『[Solaris のシステム管理 \(デバイスとファイルシステム\)](#)』の「[ファイルシステムのマウント](#)」を参照してください。

- 4 ユーザーのホームディレクトリのマウント先を作成します。

```
# mkdir -p /export/home/username
```

- 5 ユーザーのホームディレクトリをマウントします。

```
# mountall
```

現在の `vfstab` ファイルにあるすべてのエントリ (このファイルの「`mount at boot`」フィールドが `yes` に設定されている) がマウントされます。

- 6 ホームディレクトリがマウントされているかどうかを確認します。

```
# mount | grep username
```

例 5-3 ユーザーのホームディレクトリをマウントする

次の例は、ユーザー `ripley` のホームディレクトリをマウントする方法を示しています。

```
# vi /etc/vfstab
```

```
(The line venus:/export/home/ripley - /export/home/ripley
```

```
nfs - yes rw is added.)
# mkdir -p /export/home/ripley
# mountall
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror=panic/dev=...
/devices on /devices read/write/setuid/dev=46c0000 on Thu Jan 8 09:38:19 2004
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/onerror=panic/dev=...
/proc on /proc read/write/setuid/dev=4700000 on Thu Jan 8 09:38:27 2004
/etc/mnttab on mnttab read/write/setuid/dev=47c0000 on Thu Jan 8 09:38:27 2004
/dev/fd on fd read/write/setuid/dev=4800000 on Thu Jan 8 09:38:30 2004
/var/run on swap read/write/setuid/xattr/dev=1 on Thu Jan 8 09:38:30 2004
/tmp on swap read/write/setuid/xattr/dev=2 on Thu Jan 8 09:38:30 2004
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/xattr/onerror=...
/export/home/ripley on venus:/export/home/ripley remote/read/write/setuid/xattr/dev=...
```

ユーザーアカウントの管理 (作業マップ)

作業	説明	説明
グループを変更します。	グループツールを使用して、グループ名またはグループ内のユーザーを変更します。	136 ページの「グループを変更する方法」
グループを削除します。	不要な場合には、グループを削除できます。	137 ページの「グループを削除する方法」
ユーザーアカウントを変更します。	「ユーザーアカウントを無効にする」 将来必要な場合、ユーザーアカウントを一時的に無効にすることができます。 「ユーザーのパスワードを変更する」 ユーザーが忘れた場合、ユーザーのパスワードを変更する必要があります。 「パスワード有効期限を設定する」 ユーザーアカウントツールの「パスワード」オプションメニューを使えば、定期的にユーザーにパスワードを変更させることができます。	138 ページの「ユーザーアカウントを無効にする方法」 139 ページの「ユーザーのパスワードを変更する方法」 140 ページの「ユーザーアカウントのパスワード有効期限を設定する方法」

作業	説明	説明
ユーザーアカウントを削除します。	不要になった場合には、ユーザーアカウントを削除できます。	141 ページの「ユーザーアカウントを削除する方法」

ユーザーアカウントを変更する

既存のものと重複するユーザー名やUID 番号を定義しないかぎり、ユーザーアカウントのユーザー名やUID 番号を変更する必要はありません。

2つのユーザーアカウントが、同じユーザー名またはUID 番号を持つ場合、次の手順に従ってください。

- 2つのユーザーアカウントが同じUID 番号を持つ場合、ユーザーツールを使用して、どちらか一方のアカウントを削除し、もう一度、異なるUID 番号で追加します。ユーザーツールを使用して、既存のユーザーアカウントのUID 番号を変更することはできません。
- 2つのユーザーアカウントが同じユーザー名を持つ場合、ユーザーツールを使用して、どちらか一方のアカウントを修正し、ユーザー名を変更します。

ユーザーツールを使用してユーザー名を変更する場合、そのユーザーのホームディレクトリが存在すれば、ホームディレクトリの所有者も変更されます。

ユーザーアカウントの中で変更できる情報に、ユーザーのグループメンバーシップがあります。ユーザーツールの「アクション」メニューの「プロパティ」オプションを選択すると、ユーザーの二次グループを追加したり、削除したりできます。また、グループツールを使ってグループのメンバーリストを直接修正することもできます。

ユーザーアカウントの次の部分も変更できます。

- 備考欄(コメント)
- ログインシェル
- パスワードおよびパスワードオプション
- ホームディレクトリおよびホームディレクトリアクセス
- 権限および役割

ユーザーアカウントを無効にする

あるユーザーアカウントを一時的に、または永続的に無効にしなければならない場合があるかもしれません。ユーザーアカウントを無効にしたりロックしたりすると、無効なパスワード *LK* がユーザーアカウントに割り当てられ、それ以後ログインできなくなります。

もっとも簡単にユーザーアカウントを無効にする方法は、ユーザーツールを使用してアカウントのパスワードをロックすることです。

また、「ユーザープロパティ」画面の「有効期限」フィールドに有効期限を入力して、ユーザーアカウントの有効期間に制限を設けることもできます。

それ以外に、パスワード有効期限を設定するか、ユーザーのパスワードを変更することによって、ユーザーアカウントを無効にすることもできます。

ユーザーアカウントを削除する

ユーザーツールでユーザーアカウントを削除すると、passwd ファイルと group ファイル内のエントリが自動的に削除されます。また、ユーザーのホームディレクトリとメールディレクトリにあるファイルも削除されます。

▼ グループを変更する方法

グループを変更するには、次の手順を実行します。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、[『Solaris のシステム管理 \(セキュリティサービス\)』の「RBAC の構成 \(作業マップ\)」](#)を参照してください。
- 2 **Solaris 管理コンソール**を起動します。

```
# /usr/sadm/bin/smc &
```


Solaris 管理コンソールの起動方法については、[45 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」](#)または [53 ページの「ネームサービス環境で Solaris 管理コンソールを起動する方法」](#)を参照してください。
- 3 ナビゲーション区画の「管理ツール」アイコンの下にある「このコンピュータ」アイコンをクリックします。
カテゴリのリストが表示されます。
- 4 (省略可能) ネームサービス環境に適したツールボックスを選択します。
- 5 「システムの構成」アイコンをクリックします。
- 6 「ユーザー」アイコンをクリックします。
- 7 スーパーユーザーのパスワードまたは役割のパスワードを入力します。
- 8 「グループ」アイコンをクリックします。
- 9 変更するグループを選択します。
たとえば、scutters を選択します。

- 10 「グループ名」テキストボックスで、選択したグループを変更します。作業が終了したら、「了解」をクリックします。
たとえば、scutters を scutter に変更します。
これで、scutters グループに含まれていたユーザーがすべて scutter グループに入ります。

▼ グループを削除する方法

グループを削除するには、次の手順を実行します。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、[『Solaris のシステム管理\(セキュリティサービス\)』の「RBACの構成\(作業マップ\)」](#)を参照してください。
- 2 **Solaris 管理コンソールを起動します。**

```
# /usr/sadm/bin/smc &
```


Solaris 管理コンソールの起動方法については、[45 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」](#)または [53 ページの「ネームサービス環境で Solaris 管理コンソールを起動する方法」](#)を参照してください。
- 3 ナビゲーション区画の「管理ツール」アイコンの下にある「このコンピュータ」アイコンをクリックします。
カテゴリのリストが表示されます。
- 4 (省略可能) ネームサービス環境に適したツールボックスを選択します。
- 5 「システムの構成」アイコンをクリックします。
- 6 「ユーザー」アイコンをクリックします。
- 7 スーパーユーザーのパスワードまたは役割のパスワードを入力します。
- 8 「グループ」アイコンをクリックします。
- 9 削除するグループを選択します。
たとえば、scutter を選択します。
- 10 ポップアップウィンドウで「了解」をクリックします。
このグループのメンバーだったすべてのユーザーからグループが削除されます。

パスワードの管理

ユーザーツールを使用するとパスワードの管理を行えます。このツールには、次のような機能があります。

- ユーザーアカウントに通常のパスワードを指定する
- ユーザーが最初のログイン時にパスワードを作成できるようにする
- ユーザーアカウントを無効にするかロックする
- 有効期限とパスワード有効期限情報を指定する

注-パスワード有効期限は、NIS ネームサービスではサポートされません。

パスワード有効期限を設定する

NIS+ または /etc 内のファイルを使用してユーザーアカウント情報を格納する場合は、ユーザーのパスワードにパスワード有効期限を設定できます。Solaris 9 12/02 リリース以降は、LDAP ディレクトリサービスでもパスワードの有効期限がサポートされています。

パスワード有効期限の設定によって、ユーザーに定期的なパスワードの変更を強制したり、あるパスワードを保持するのに必要な最低日数以前にパスワードを変更するのを防止したりできます。不正ユーザーが、古くて使用されていないアカウントを使用して、発覚せずにシステムのアクセス権を得るような場合を防止するために、アカウントが無効になる日付を設定することができます。パスワードの有効期限属性を設定するには、passwd コマンドまたは Solaris 管理コンソールのユーザーツールを使用します。

Solaris 管理コンソールの起動方法については、[45 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」](#)を参照してください。

▼ ユーザーアカウントを無効にする方法

ユーザーアカウントを無効にする必要がある場合は、次の手順を実行します。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「RBAC の構成 (作業マップ)」を参照してください。
- 2 Solaris 管理コンソールを起動します。

```
# /usr/sadm/bin/smc &
```

Solaris 管理コンソールの起動方法については、[45 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」](#)または[53 ページの「ネームサービス環境で Solaris 管理コンソールを起動する方法」](#)を参照してください。

- 3 ナビゲーション区画の「管理ツール」アイコンの下にある「このコンピュータ」アイコンをクリックします。
カテゴリのリストが表示されます。
- 4 (省略可能) ネームサービス環境に適したツールボックスを選択します。
- 5 「システムの構成」アイコンをクリックします。
- 6 「ユーザー」アイコンをクリックし、スーパーユーザーのパスワードまたは役割のパスワードを入力します。
- 7 「ユーザー」アイコンをクリックします。
- 8 ユーザーをダブルクリックします。
たとえば、scutter2 を選択します。
- 9 「基本」タブ機能の「アカウントの有効/無効」セクションで「アカウントはロック状態」オプションを選択します。
- 10 「了解 (OK)」をクリックします。

▼ ユーザーのパスワードを変更する方法

パスワードを忘れてしまった場合は、次の手順を実行します。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、[『Solaris のシステム管理 \(セキュリティサービス\)』の「RBAC の構成 \(作業マップ\)」](#)を参照してください。
- 2 Solaris 管理コンソールを起動します。

```
# /usr/sadm/bin/smc &
```

Solaris 管理コンソールの起動方法については、[45 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」](#)または[53 ページの「ネームサービス環境で Solaris 管理コンソールを起動する方法」](#)を参照してください。

- 3 ナビゲーション区画の「管理ツール」アイコンの下にある「このコンピュータ」アイコンをクリックします。
カテゴリのリストが表示されます。
- 4 (省略可能) ネームサービス環境に適したツールボックスを選択します。
- 5 「システムの構成」アイコンをクリックします。
- 6 「ユーザー」アイコンをクリックします。
- 7 スーパーユーザーのパスワードまたは役割のパスワードを入力します。
- 8 「ユーザー」アイコンをクリックし、新しいパスワードを必要とするユーザーをダブルクリックします。
たとえば、scutter1 を選択します。
- 9 「パスワード」タブを選択し、「ユーザーは次のログイン時にこのパスワードを使用する必要があります」オプションを選択します。
- 10 ユーザーの新しいパスワードを入力し、「了解」をクリックします。

▼ ユーザーアカウントのパスワード有効期限を設定する方法

ユーザーアカウントのパスワード有効期限を設定するには、次の手順を実行します。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、[『Solaris のシステム管理 \(セキュリティサービス\)』の「RBAC の構成 \(作業マップ\)」](#)を参照してください。
- 2 **Solaris 管理コンソール**を起動します。

```
# /usr/sadm/bin/smc &
```

Solaris 管理コンソールの起動方法については、[45 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」](#)または[53 ページの「ネームサービス環境で Solaris 管理コンソールを起動する方法」](#)を参照してください。
- 3 ナビゲーション区画の「管理ツール」アイコンの下にある「このコンピュータ」アイコンをクリックします。
カテゴリのリストが表示されます。

- 4 (省略可能) ネームサービス環境に適したツールボックスを選択します。
- 5 「システムの構成」アイコンをクリックします。
- 6 「ユーザー」アイコンをクリックし、スーパーユーザーのパスワードまたは役割のパスワードを入力します。
- 7 「ユーザー」アイコンをクリックします。
- 8 ユーザーをダブルクリックし、「パスワードオプション」タブを選択します。
たとえば、scutter2 を選択します。
- 9 「パスワードオプション」タブを選択します。
- 10 「日数」オプションから該当するパスワードオプションを選択し、「了解」をクリックします。
たとえば、ユーザーがパスワードを変更しなければならない日付を設定するには、「次のパスワード変更までに必要な日数」を選択します。

▼ ユーザーアカウントを削除する方法

ユーザーアカウントを削除するには、次の手順を実行します。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、[『Solaris のシステム管理 \(セキュリティサービス\)』の「RBAC の構成 \(作業マップ\)」](#)を参照してください。
- 2 **Solaris 管理コンソール**を起動します。

```
# /usr/sadm/bin/smc &
```

Solaris 管理コンソールの起動方法については、[45 ページの「スーパーユーザーまたは役割としてコンソールを起動する方法」](#)または [53 ページの「ネームサービス環境で Solaris 管理コンソールを起動する方法」](#)を参照してください。
- 3 ナビゲーション区画の「管理ツール」アイコンの下にある「このコンピュータ」アイコンをクリックします。
カテゴリのリストが表示されます。
- 4 (省略可能) ネームサービス環境に適したツールボックスを選択します。
- 5 「システムの構成」アイコンをクリックします。

- 6 「ユーザー」アイコンをクリックします。
- 7 スーパーユーザーのパスワードまたは役割のパスワードを入力します。
- 8 「ユーザー」アイコンをクリックします。
- 9 削除するユーザーアカウントをダブルクリックします。
たとえば、scutter4 を選択します。
- 10 ユーザーアカウントを削除してもよい場合は、ポップアップウィンドウで「削除」をクリックします。
ユーザーのホームディレクトリとメールボックスの内容を削除するよう指示されます。

クライアントとサーバーサポートの管理 (概要)

この章では、ネットワーク環境でのサーバーとクライアントの管理について説明し、Solaris オペレーティングシステムでサポートされる各システム構成(「システムタイプ」と呼ぶ)に関する情報を紹介します。また、目的に合った最適なシステムを選択するためのガイドラインも示します。

注-ZFS ルートファイルシステムがインストールされているシステムでは、`smosservice` および `smdiskless` コマンドを使用できません。これは、ZFS ルートファイルシステムのインストールがサポートされているすべての Solaris リリースでの既知の問題です。

Solaris Flash インストール機能を使用すると、UFS ルートファイルシステムまたは ZFS ルートファイルシステムを実行するシステムをすばやくプロビジョニングできます。詳細は、『[Solaris ZFS Administration Guide](#)』の「[Installing a ZFS Root File System \(Flash Archive Installation\)](#)」を参照してください。

この章の内容は以下のとおりです。

- 144 ページの「クライアントとサーバーサポートの管理に関する新機能」
- 146 ページの「クライアントとサーバータスクの操作手順」
- 146 ページの「サーバー、クライアント、およびアプライアンスとは」
- 147 ページの「クライアントサポートとは」
- 148 ページの「システムタイプの概要」
- 151 ページの「ディスクレスクライアント管理の概要」

ディスクレスクライアントサポートの管理手順については、第7章「ディスクレスクライアントの管理(手順)」を参照してください。

クライアントとサーバーサポートの管理に関する新機能

この節では、この Solaris リリースで新たに追加または変更されたディスクレスクライアント機能について説明します。Solaris の新機能の一覧および Solaris リリースについての説明は、『[Solaris 10 の概要](#)』を参照してください。

bootadm -p コマンドによるプラットフォーム指定のサポート

bootadm コマンドに、**-p platform** 引数が新たに追加されました。このオプションを使用すると、クライアントプラットフォームがサーバープラットフォームと異なる場合 (ディスクレスクライアントを管理している場合など) に、クライアントシステムのプラットフォームまたはマシンのハードウェアクラスを指定できます。

詳細は、[bootadm\(1M\)](#) のマニュアルページを参照してください。

nfs4_domain キーワードがディスクレスクライアントのブートに与える影響

Solaris 10 OS で配布されていた **set_nfs4_domain** スクリプトは、NFSv4 ドメインの設定に使用されなくなりました。NFSv4 ドメインを設定するには、新しい **nfs4_domain** キーワードをディスクレスクライアントの **sysidcfg** ファイルに追加してください。**sysidcfg** ファイルに **nfs4_domain** キーワードがある場合は、ディスクレスクライアントの最初のブート時にドメインが適切に設定されます。

x86: GRUB ブート環境のディスクレスクライアントの変更点

新しいディスクレスブート方式では、次の機能が拡張されています。

- OS サーバーから複数の Solaris リリースを同時に提供できます。
新しいディスクレスブート方式で、`pxegrub` を使用してネットワークブートを実行すると、クライアントの GRUB メニューに複数のリリースを表示できます。
- ベンダー固有のオプションをブートアーカイブに指定できます。
以前のリリースでは、クライアント固有のブートプロパティは通常、`bootenv.rc` ファイルに定義されていて、DHCP を設定するときにベンダー固有のオプションを使用する方法で適用していました。このとき、必要な情報の合計長が DHCP 仕様の制限を超えることがよくありました。
新しいブート方式では、この情報はブートアーカイブに含まれるようになります。PXE/DHCP サーバーは、Site Option 150 を使用してサーバー IP アドレス、ブートファイル、`pxegrub`、およびクライアント固有のメニューファイル (使用する場合) を指定する場合にのみ必要となります。

x86: smdiskless コマンドの変更点

ディスクレスクライアントを設定するには `smdiskless` コマンドを使用します。以前の `smdiskless` コマンドは、ルート (`/`) および `/usr` ファイルシステムを設定してから、NFS 経由でこれらのファイルシステムをクライアントにエクスポートしていました。クライアントをブートするには、さらに `/tftpboot` 領域を手動で構成する必要がありました。この手動で構成する手順が、ディスクレスクライアントを設定するときに必要なくなりました。`smdiskless` コマンドが、`/usr/sadm/lib/wbem/config_tftp` ファイルのスクリプトを自動的に呼び出して、PXE ブートのための `/tftpboot` 領域を準備します。

`smdiskless` コマンドの実行後に、`/tftpboot/01 ethernet-address` ファイルが `pxegrub` および `/tftpboot/menu.lst.01 ethernet-address` ファイルへのリンクとして表示されます。このファイルには、GRUB メニューエントリが含まれます。この例の `ethernet-address` では、クライアントネットワークインタフェースの Ethernet アドレスの先頭に 01 が付いています。クライアントネットワークインタフェースの Ethernet アドレスを指定するときには、大文字を使用し、コロンは含めないでください。

ディスクレスクライアントのブートアーカイブは、シャットダウン時に自動的に更新されます。シャットダウン時にクライアントのブートアーカイブの期限が切れている場合は、OS サーバーから次のコマンドを実行して、ブートアーカイブを更新する必要があります。

```
# bootadm update-archive -f -R /export/root/host-name
```

host-name はクライアントシステムのホスト名です。

詳細は、[286 ページの「x86: フェイルセーフアーカイブをブートして、破壊されたブートアーカイブを強制的に更新する方法」](#)と、[bootadm\(1M\)](#) のマニュアルページを参照してください。

注 - この情報は、x86 ベースのクライアントを運用する SPARC ベースおよび x86 ベースの OS サーバーに適用されます。

DHCP の設定および構成の詳細については、『[Solaris のシステム管理 \(IP サービス\)](#)』の第 14 章「[DHCP サービスの構成 \(手順\)](#)」を参照してください。

GRUB ブート環境でのディスクレスクライアントの管理方法の詳細は、[271 ページの「GRUB を使用して x86 システムをブートする \(作業マップ\)」](#)を参照してください。

クライアントとサーバタスクの操作手順

サーバーとクライアントサポートの設定手順については、次の表を参照してください。

クライアントとサーバサービス	詳細
インストールまたは JumpStart クライアント	『 Solaris 10 インストールガイド (ネットワークインストール) 』
Solaris OS でのディスクレスクライアントシステム	151 ページの「ディスクレスクライアント管理の概要」 および 第 7 章「ディスクレスクライアントの管理 (手順)」
Solaris 7 OS 以前の Solaris リリースでのディスクレスクライアントシステム	『 Solstice AdminSuite 2.3 管理者ガイド 』

サーバー、クライアント、およびアプライアンスとは

通常、ネットワーク上のシステムは、次の表に示されたいずれかのシステムタイプに該当します。

システムタイプ	説明
サーバー	ネットワーク上のほかのシステムにサービスを提供するシステム。ファイルサーバー、ブートサーバー、Web サーバー、データベースサーバー、ライセンスサーバー、印刷サーバー、インストールサーバー、さらに、特定のアプリケーション用のサーバーなどもあります。この章では、「サーバー」とは、ネットワーク上のほかのシステムにブートサービスとファイルシステムを提供するシステムのことを意味します。
クライアント	サーバーから提供されるリモートサービスを利用するシステム。クライアントによっては、ディスクの記憶容量に限りがあったり、まったくディスクを持たないものもあり、このようなクライアントは、サーバーから提供されるリモートファイルシステムに依存しなければ機能できません。ディスクレスシステム、およびアプライアンスシステムは、このタイプのクライアントの一例です。 また、サーバーが提供するリモートサービス(インストールソフトウェアなど)を利用しながらも、サーバーに依存しなくても機能するクライアントもあります。このタイプのクライアントの例として、スタンドアロンシステムがあります。スタンドアロンシステムは、ルート(/)、/usr、/export/home の各ファイルシステムとスワップ空間を含む独自のハードディスクを備えています。
アプライアンス	Sun Ray TM アプライアンスのようなネットワークアプライアンスを使うと、アプリケーションや Solaris OS にアクセスできます。アプライアンスを使うと、サーバー管理を一元化できるので、クライアント管理またはアップグレードを行う必要がなくなります。Sun Ray アプライアンスには「ホットデスク」機能も備わっています。ホットデスク機能を使用すると、サーバーグループのどのアプライアンスからでもただちに、作業中のコンピュータセッション(正確に言うと作業を中断したところ)にアクセスできます。詳細は、 http://www.sun.com/software/index.jsp?cat=Desktop[amp]amp;tab=3[amp]amp;subcat=Sun%20Ray%20Clients を参照してください。

クライアントサポートとは

サポートには、次の項目が含まれます。

- システムをネットワークに認識させる(ホスト名とイーサネットアドレス情報)
- システムをリモートからインストールおよびブートできるインストールサービスを提供する
- ディスク容量に限りがある、またはディスク容量がないシステムに Solaris OS サービスやアプリケーションサービスを提供する

システムタイプの概要

システムタイプは、ルート (/) と /usr ファイルシステム (スワップ領域を含む) にアクセスする方法によって決まる場合があります。たとえば、スタンドアロンシステムとサーバーシステムでは、これらのファイルシステムをローカルディスクからマウントしていますが、その他のクライアントでは、これらのファイルシステムをリモートからマウントし、サーバーから提供されるサービスに依存しています。次の表にそれぞれのシステムタイプの特徴を示します。

表 6-1 システムタイプの特徴

システムタイプ	ローカルファイルシステム	ローカルスワップ空間	リモートファイルシステム	ネットワーク利用度	相対パフォーマンス
サーバー	ルート (/)	使用可能	使用不可	高	高
	/usr				
	/home				
	/opt				
	/export/home				
スタンドアロンシステム	ルート (/)	使用可能	使用不可	低	高
	/usr				
	/export/home				
OS サーバー	/export/root				
ディスクレスクライアント	使用不可	使用不可	ルート (/)	高	低
			スワップ領域	高	低
			/usr		
			/home		
アプライアンス	使用不可	使用不可	使用不可	高	高

サーバー

サーバーシステム上には、次のファイルシステムがあります。

- ルート (/) と /usr ファイルシステム、およびスワップ空間
- /export と /export/home ファイルシステム。クライアントシステムをサポートし、ユーザーにホームディレクトリを提供します

- アプリケーションソフトウェアを格納する /opt ディレクトリまたはファイルシステム

サーバー上には、ほかのシステムをサポートするために次のソフトウェアも格納できます。

- 別のリリースが動作するディスクレスシステム用の Solaris OS サービス



注意-GRUB を x86 プラットフォームで実装しているか、新しいブートアーキテクチャを SPARC プラットフォームで実装している Solaris リリースを実行しているシステムが 1 つだけの OS クライアントサーバー構成では、深刻な互換性の問題が発生する可能性があります。このため、ディスクレスクライアントサポートを追加する前に、ディスクレスシステムをインストールまたはアップグレードして、サーバー OS と同じリリースにすることをお勧めします。

GRUB ベースのブートは、Solaris 10 1/06 リリースで x86 プラットフォームに導入されました。新しい SPARC ブートアーキテクチャは、Solaris 10 10/08 リリースで導入されました。

- サーバーとは異なるプラットフォームを使用するクライアント
- ネットワークに接続されたシステムがリモートインストールを実行するのに必要な Solaris CD のイメージソフトウェアとブート用ソフトウェア
- ネットワークに接続されたシステムがカスタム JumpStart™ インストールを行うのに必要な JumpStart ディレクトリ

スタンドアロンシステム

「ネットワークに接続されたスタンドアロンシステム」は、ネットワーク上のほかのシステムと情報を共有できますが、ネットワークから切り離されても機能できます。

スタンドアロンシステムは、ルート (/)、/usr、/export/home の各ファイルシステムとスワップ空間を含むハードディスクを自ら持つため、独立して動作できます。つまり、スタンドアロンシステムは、OS ソフトウェア、実行可能ファイル、仮想メモリー空間、ユーザーが作成したファイルにローカルにアクセスできます。

注-スタンドアロンシステムには、必要なファイルシステムを保持できるだけのディスク容量が必要です。

「ネットワークに接続されないスタンドアロンシステム」は、ネットワークに接続されていない点を除き、ネットワークに接続されたスタンドアロンシステムと同じです。

ディスクレスクライアント

「ディスクレスクライアント」とは、ディスクが搭載されておらず、必要なすべてのソフトウェアおよび記憶装置をサーバーに依存しているシステムのことで、ディスクレスクライアントには、サーバーからリモートで、ルート (/)、/usr、および /home ファイルシステムがマウントされます。

ディスクレスクライアントでは、ネットワークを介して OS ソフトウェアや仮想メモリー空間に継続的にアクセスする必要があるため、多大なネットワークトラフィックが発生します。ディスクレスクライアントは、ネットワークから切り離されたり、そのサーバーが正しく機能しない場合は機能できません。

ディスクレスクライアントの概要については、[151 ページの「ディスクレスクライアント管理の概要」](#)を参照してください。

アプライアンス

アプライアンス (Sun Ray アプライアンスなど) は、管理を必要としない X ディスプレイデバイスです。このデバイスには、CPU、ファン、ディスクがなく、メモリーもわずかしが搭載されていません。アプライアンスは、Sun のディスプレイモニターに接続されていますが、アプライアンスユーザーのデスクトップセッションは、サーバーで実行され、その結果がユーザーのモニターに表示されます。

ユーザーの X 環境は自動的に設定されます。この環境には次の特徴があります。

- ほかのファイルシステムおよびソフトウェアアプリケーションへのアクセスは、サーバーに依存します
- ソフトウェアの一元管理およびリソース共有機能を提供します
- 永続的なデータがない、FRU (現場交換可能ユニット) として使用します

システムタイプ選択のガイドライン

次の特徴に基づいてそれぞれのシステムタイプを比較することにより、使用中の環境にどのシステムタイプが適切かを判断することができます。

一元管理

- システムを FRU (現場交換可能ユニット) として扱えるか。
これは、時間がかかるバックアップや復旧操作を必要とせず、またシステムデータを失わずに、障害が発生したシステムを直ちに新しいシステムと交換できることを意味します。
- システムをバックアップする必要があるか。
数多くのデスクトップシステムのバックアップを実行するには、時間とリソースの点で多大の費用コストがかかる場合があります。

- システムのデータは、中央サーバーから変更できるか。
 - クライアントシステムのハードウェアを操作せず、短時間で簡単にシステムを中央サーバーからインストールできるか。
- パフォーマンス
- この構成は、デスクトップで使用しても性能が低下しないか。
 - ネットワークにシステムを追加すると、既存のネットワーク上のシステムの性能に影響を与えるか。

ディスク使用率

この構成を効果的に導入するには、どれくらいのディスク容量が必要か。

次の表では、各システムタイプの点数を特徴別に表示しています。1は、もっとも効果があることを意味します。4は、もっとも効果が低いことを意味します。

表 6-2 システムタイプの比較

システムタイプ	一元管理	パフォーマンス	ディスク使用率
スタンドアロンシステム	4	1	4
ディスクレスクライアント	1	4	1
アプライアンス	1	1	1

ディスクレスクライアント管理の概要

次の節と第7章「ディスクレスクライアントの管理(手順)」では、Solaris オペレーティングシステム (Solaris OS) でのディスクレスクライアントサポートの管理方法について説明します。

「ディスクレスクライアント」とは、オペレーティングシステム、ソフトウェア、および記憶装置を「OS サーバー」に依存しているシステムのことです。ディスクレスクライアントは、そのルート (/)、 /usr、およびその他のファイルシステムを OS サーバーからマウントします。ディスクレスクライアントは独自の CPU と物理メモリーを持っており、データをローカルで処理することができます。しかしディスクレスクライアントは、ネットワークから切り離されたり、その OS サーバーが正しく機能しない場合は機能できません。ディスクレスクライアントは、ネットワークを経由して継続的に機能する必要があるため、多大なネットワークトラフィックを発生させます。

Solaris 9 以降のリリースでは、ディスクレスクライアントの smossservice コマンドと smdiskless コマンドを使って、OS サービスやディスクレスクライアントサポートを管理できるようになりました。Solaris 8 リリースでは、Solstice™ GUI 管理ツールでディスクレスクライアントが管理されていました。

OS サーバーおよびディスクレスクライアントのサポート情報



注意—一方のシステムは新しいブート方式を実装しているが、もう一方のシステムは実装していない OS クライアントサーバー構成を使用してディスクレスクライアントサポートを追加しようとする、深刻な互換性の問題が発生する可能性があります。新しいブート (GRUB) は、Solaris 10 1/06 リリースから x86 プラットフォームで実装され、Solaris 10 10/8 リリースから SPARC プラットフォームで実装されました。OS サーバー上で実行されているリリースよりも新しい Solaris リリースを実行しているシステムでのディスクレスサポートの追加も、サポートされていない構成です。問題が発生する可能性を回避するために、ディスクレスクライアントサポートを追加する前に最新の Solaris リリースをインストールすることをお勧めします。

smosservice および smdiskless コマンドでサポートされる Solaris リリースおよびアーキテクチャの種類は次のとおりです。

- **SPARC ベースのサーバー:** Solaris 8、Solaris 9、および Solaris 10 リリースでサポートされます。
- **SPARC ベースのクライアント:** Solaris 8、Solaris 9、および Solaris 10 リリースでサポートされます。
- **x86 ベースのサーバー:** Solaris 9 および Solaris 10 リリースでサポートされます。
- **x86 ベースのクライアント:** Solaris 9 および Solaris 10 リリースでサポートされます。

次の表に、smosservice および smdiskless コマンドでサポートされる x86 OS クライアントサーバー構成を示します。この情報は、Solaris 9 および Solaris 10 FCS (3/05) リリースに適用されます。

Solaris 10 1/06 リリース以降を実行している場合は、ディスクレスクライアントサポートを追加する前に、同じリリースをインストールするか、同じリリースにアップグレードすることをお勧めします。

表 6-3 クライアントとサーバーでの x86 OS のサポート

ディスクレスクライアント OS		
サーバー OS	Solaris 10 3/05	Solaris 9
Solaris 10 3/05	サポートされています	サポートされています
Solaris 9	サポートされていません	サポートされています

次の表に、smosservice コマンドと smdiskless コマンドでサポートされる SPARC OS クライアントサーバー構成を示します。この情報は、Solaris 8 および Solaris 9 リリース、Solaris 10 5/08 OS までの Solaris 10 リリースに適用されます。

Solaris 10 10/08 リリース以上を実行している場合は、ディスクレスクライアントサポートを追加する前に、同じリリースをインストールするか、同じリリースにアップグレードすることをお勧めします。

表 6-4 クライアントとサーバーでの SPARC OS のサポート

サーバー OS	ディスクレスクライアント OS		
	Solaris 10 3/05 から Solaris 10 5/08	Solaris 9	Solaris 8
Solaris 10 3/05 から Solaris 10 5/08	サポートされています	サポートされています	サポートされています
Solaris 9	サポートされていません	サポートされています	サポートされています
Solaris 8	サポートされていません	サポートされていません	サポートされています

ディスクレスクライアント管理機能

smosservice コマンドおよび smdiskless コマンドを使うと、ネットワークにディスクレスクライアントサポートを追加したり、維持したりすることができます。ネームサービスを使うと、システム情報を一元管理できるので、ホスト名などの重要なシステム情報をネットワーク上のすべてのシステムに複製する必要がありません。

smosservice コマンドおよび smdiskless コマンドを使うと、次の作業を実行できます。

- ディスクレスクライアントサポートを追加および変更します
- OS サービスを追加および削除します
- LDAP、NIS、NIS+、files の各ネームサービス環境でのディスクレスクライアント情報を管理します

x86 システム上で GRUB ベースのブートを実行する場合は、DHCP 構成を手動で設定する必要があります。詳細は、[163 ページの「x86: GRUB ベースのブート環境にディスクレスクライアントを追加するための準備方法」](#)を参照してください。

注-ディスクレスクライアントコマンドは、ディスクレスクライアントのブートの設定にのみ使用できます。このコマンドは、リモートインストールまたはプロファイルサービスなど、ほかのサービスの設定では使用できません。リモートインストールを設定するには、`sysidcfg` ファイルにディスクレスクライアント仕様を定義する必要があります。詳細については、『[Solaris 10 インストールガイド \(カスタム JumpStart/ 上級編\)](#)』を参照してください。

ディスクレスクライアントコマンドの使用

次の表のコマンドを使って独自のシェルスクリプトを記述すると、簡単にディスクレスクライアント環境を設定および管理できます。

表 6-5 ディスクレスクライアントコマンド

コマンド	サブコマンド	作業
<code>/usr/sadm/bin/smosservice</code>	<code>add</code>	OS サービスを追加します
	<code>delete</code>	OS サービスを削除します
	<code>list</code>	OS サービスをリスト表示します
	<code>patch</code>	OS サービスのパッチを管理します
<code>/usr/sadm/bin/smdiskless</code>	<code>add</code>	ディスクレスクライアントを OS サーバーに追加します
	<code>delete</code>	ディスクレスクライアントを OS サーバーから削除します
	<code>list</code>	OS サーバー上のディスクレスクライアントをリスト表示します
	<code>modify</code>	ディスクレスクライアントの属性を変更します

次に示す 2 種類の方法で、これらのコマンドに関するヘルプを参照することができます。

- 次の例に示すように、コマンド、サブコマンド、および必要なオプションを入力するときに `-h` オプションを使用します。

```
% /usr/sadm/bin/smdiskless add -p my-password -u my-user-name -- -h
```

- [smdiskless\(1M\)](#) および [smosservice\(1M\)](#) のマニュアルページを参照します。

ディスクレスクライアント管理に必要な RBAC 権限

`smoservice` コマンド および `smdiskless` コマンドはスーパーユーザーとして使用できます。役割によるアクセス制御 (RBAC) を使用している場合、割り当てられた RBAC 権限に応じて、すべてのディスクレスクライアントコマンドまたはそのサブセットのいずれかを使用できます。次の表にディスクレスクライアントコマンドを使用するのに必要な RBAC 権限を示します。

表 6-6 ディスクレスクライアント管理に必要な RBAC 権限

RBAC 権限	コマンド	作業
基本的な Solaris ユーザー、ネットワーク管理	<code>smoservice list</code>	OS サービスをリスト表示します
	<code>smoservice patch</code>	OS サービスのパッチリストを表示します
	<code>smdiskless list</code>	OS サーバー上のディスクレスクライアントのリストを表示します
Network Management	<code>smdiskless add</code>	ディスクレスクライアントを追加します
システム管理者	すべてのコマンド	すべての作業

OS サービスの追加

Solaris OS サーバーとは、ディスクレスクライアントシステムをサポートするオペレーティングシステム (OS) サービスを提供するサーバーのことです。OS サーバーを新たにサポートすることも、`smoservice` コマンドを使ってスタンドアロンシステムを OS サーバーに変換することもできます。

サポートする各プラットフォームグループおよび Solaris リリース用に、特定の OS サービスを OS サーバーに追加する必要があります。たとえば、Solaris 10 リリースが動作する SPARC sun-4u システムをサポートするには、`sun-4u/Solaris 10 OS サービス` を OS サーバーに追加する必要があります。サポートするディスクレスクライアントごとに、そのクライアント用の OS サービスを OS サーバーに追加する必要があります。たとえば、Solaris 9 リリースが動作する SPARC sun-4m システムまたは x86 システムは、それぞれ異なるプラットフォームグループに属しているため、それぞれのシステムをサポートするための OS サービスを追加する必要があります。

OS サービスを追加するには、該当する Solaris ソフトウェアの CD またはディスクイメージへのアクセス権が必要です。

OS サーバーにパッチが適用された場合に OS サービスを追加する

OS サーバーに OS サービスを追加しようとする、サーバー上の OS と追加するサービスの OS のバージョンが一致しないというエラーメッセージが表示される場合があります。このエラーメッセージは、OS サーバーにインストールされている OS のパッケージにパッチが適用されていても、追加しようとしている OS サービスのパッケージにはパッチが適用されていない(パッチがパッケージに組み込まれているため)場合に表示されます。

たとえば、Solaris の最新リリースまたは Solaris 10 リリースが動作するサーバーがあるとしします。このサーバーには、パッチがすでに適用された Solaris 9 SPARC sun-4m OS サービスなど、追加の OS サービスがロードされている可能性があります。このため、Solaris 8 SPARC sun-4u OS サービスを CD-ROM からこのサーバーに追加しようとする、次のエラーメッセージが表示されることがあります。

```
Error: inconsistent revision, installed package appears to have been
patched resulting in it being different than the package on your media.
You will need to backout all patches that patch this package before
retrying the add OS service option.
```

OS サーバーに必要なディスク容量

ディスクレスクライアント環境を設定する前に、ディスクレスクライアントの各ディレクトリに必要なディスク容量があるかどうかを確認する必要があります。

以前のバージョンの Solaris リリースでは、インストール中にディスクレスクライアントサポートについてのプロンプトが表示されました。Solaris 9 以降のリリースでは、手動で、インストール中に /export ファイルシステムを割り当てるか、インストール後に作成する必要があります。具体的なディスク容量要件については、次の表を参照してください。

表 6-7 Solaris OS サーバーおよびディスクレスクライアントのディスク容量に関する推奨事項

サーバー OS/アーキテクチャタイプ	ディレクトリ	必要なディスク容量
Solaris 10 SPARC の OS サーバー	/export	5 - 6.8G バイト
Solaris 10 x86 の OS サーバー	/export	5 - 6.8G バイト
Solaris 10 SPARC のディスクレスクライアント	/export	1 台のディスクレスクライアントにつき 200 - 300M バイトの容量を確保してください。
Solaris 10 x86 のディスクレスクライアント	/export	1 台のディスクレスクライアントにつき 200 - 300M バイトの容量を確保してください。

注- ディスク容量に関する推奨事項は、インストールされている Solaris リリースによって異なる場合があります。最新の Solaris リリースでのディスク容量に関する推奨事項の詳細については、[『Solaris 10 11/06 インストールガイド\(インストールとアップグレードの計画\)』](#)の「ソフトウェアグループごとの推奨ディスク容量」を参照してください。

ディスクレスクライアントの管理 (手順)

この章では、Solaris オペレーティングシステム (Solaris OS) でディスクレスクライアントを管理する方法について説明します。

注-ZFS ルートファイルシステムがインストールされているシステムでは、`smosservice` および `smdiskless` コマンドを使用できません。これは、ZFS ルートファイルシステムのインストールがサポートされているすべての Solaris リリースでの既知の問題です。

Solaris Flash インストール機能を使用すると、UFS ルートファイルシステムまたは ZFS ルートファイルシステムを実行するシステムをすばやくプロビジョニングできます。詳細は、『[Solaris ZFS Administration Guide](#)』の「[Installing a ZFS Root File System \(Flash Archive Installation\)](#)」を参照してください。

ディスクレスクライアントの管理手順については、[159 ページ](#)の「[ディスクレスクライアントの管理 \(作業マップ\)](#)」を参照してください。ディスクレスクライアントの管理に関連するインストールの問題については、[183 ページ](#)の「[ディスクレスクライアントのインストールの問題](#)」を参照してください。ディスクレスクライアントの管理の概要については、[第 6 章](#)「[クライアントとサーバーサポートの管理 \(概要\)](#)」を参照してください。

ディスクレスクライアントの管理 (作業マップ)

次の表に、ディスクレスクライアントの管理に必要な手順を示します。

作業	説明	参照先
1.(省略可能) Solaris 管理コンソールのログインを有効にしてディスクレスクライアントのエラーメッセージを表示します。	コンソールのメインウィンドウからログビューアを選択し、ディスクレスクライアントのエラーメッセージを表示します。	45 ページの「Solaris 管理コンソールを起動する」
2. ディスクレスクライアントを追加する準備をします。	サポートされているリリースを確認し、各ディスクレスクライアントのプラットフォーム、メディアパス、およびクラスタ (またはソフトウェアグループ) を確認します。	163 ページの「x86: GRUB ベースのブート環境にディスクレスクライアントを追加するための準備方法」 165 ページの「Solaris 10 OS にディスクレスクライアントを追加するための準備方法」
3. 必要な OS サービスを OS サーバーに追加します。	smosservice コマンドを使用して、サポートするディスクレスクライアントの OS サービスを追加します。サポートする各ディスクレスクライアントプラットフォームのプラットフォームおよびメディアパスを識別する必要があります。	167 ページの「ディスクレスクライアントサポートの OS サービスの追加方法」
4. OS サービスをサーバーに追加したときに失われた ARCH=all パッケージをすべて検出してインストールします。 注-これらのパッケージを各ディスクレスクライアントに個別に追加しなくてもいいように、ディスクレスクライアントサポートを追加する前にこの作業を行ってください。	smosservice add コマンドを実行しても、ARCH=all が指定されているルート (/) および /usr パッケージはどれもインストールされません。これらのパッケージは、OS サービスを OS サーバーに追加したあとに、手動でインストールする必要があります。 この作業は、Solaris 2.1 OS 以降で発生します。この作業は、SPARC および x86 の両方のプラットフォームに適用されます。失われる ARCH=all パッケージは、実行している Solaris OS によって異なります。	183 ページの「失われた ARCH=all パッケージを検出してインストールする方法」

作業	説明	参照先
5. ディスクレスクライアントを追加します。	smdiskless コマンドを使用してすべての必須情報を指定し、ディスクレスクライアントサポートを追加します。	170 ページの「 x86: GRUB ベースのブート環境にディスクレスクライアントを追加する方法 」 173 ページの「 Solaris 10 OS でディスクレスクライアントを追加する方法 」
6. ディスクレスクライアントをブートします。	ディスクレスクライアントをブートして、ディスクレスクライアントが正常に追加されたことを確認します。	176 ページの「 x86: GRUB を使用してディスクレスクライアントをブートする方法 」 177 ページの「 SPARC: Solaris 10 OS でディスクレスクライアントをブートする方法 」
7.(省略可能) ディスクレスクライアントサポートを削除します。	必要がなくなった場合には、ディスクレスクライアントのサポートを削除します。	178 ページの「 ディスクレスクライアントサポートの削除方法 」
8.(省略可能) ディスクレスクライアントの OS サービスを削除します。	必要がなくなった場合には、ディスクレスクライアントの OS サービスを削除します。	178 ページの「 ディスクレスクライアントの OS サービスを削除する方法 」
9.(省略可能) OS サービスにパッチを適用します。	ディスクレスクライアント OS サービスのパッチを追加、削除、リスト表示、または同期化します。	180 ページの「 ディスクレスクライアントの OS パッチの追加方法 」

ディスクレスクライアントの管理の準備

この節では、ディスクレスクライアントの管理に必要な準備作業について説明します。

ディスクレスクライアントを管理する場合には、次の点に注意してください。

- Solaris インストールプログラムでは、ディスクレスクライアントサポートのセットアップを促すメッセージが表示されません。ディスクレスクライアントをサポートするには、手動で /export パーティションを作成する必要があります。/export パーティションの作成は、インストール中またはインストール後に行なってください。
- /export パーティションには、サポートするクライアントの数に応じて、5G バイト以上の容量が必要です。詳細は、「OS サーバーに必要なディスク容量」を参照してください。

- `smosservice` コマンドまたは `smdiskless` コマンドで指定したネームサービスは、`/etc/nsswitch.conf` ファイルで指定された一次ネームサービスと一致していなければなりません。 `smdiskless` コマンドまたは `smosservice` コマンドにネームサービスを指定しない場合、デフォルトのネームサービスは `files` になります。

ネームサーバーを指定するには、`smosservice` コマンドおよび `smdiskless` コマンドの `-D` オプションを使用します。詳細は、[smosservice\(1M\)](#) および [smdiskless\(1M\)](#) のマニュアルページを参照してください。

Solaris 10 8/07 リリース以降では、Solaris 10 OS で配布されていた `set_nfs4_domain` スクリプトは、NFSv4 ドメインの設定に使用されなくなりました。NFSv4 ドメインを設定するには、`nfs4_domain` キーワードをディスクレスクライアントの `sysidcfg` ファイル (たとえば `server:/export/root/client/etc/sysidcfg`) に追加してください。

クライアントシステムの `sysidcfg` ファイルに `nfs4_domain` キーワードがある場合は、ディスクレスクライアントの最初のブート時にドメインが適切に設定されます。また、OS サーバーは起動して稼働している必要があり、ディスクレスクライアントの NFSv4 ドメイン設定は OS サーバーの `/var/run/nfs4_domain` ファイル内の設定と一致している必要があります。

詳細は、『[Solaris 10 インストールガイド\(ネットワークインストール\)](#)』の「[sysidcfg ファイルによる事前設定](#)」を参照してください。

- OS サーバーとディスクレスクライアントは、同じサブネット上になければなりません。
- EFI ラベル付きディスク上にある UFS ファイルシステムには OS やディスクレスクライアントのサービスを追加「できない」ため、マルチテラバイト UFS ファイルシステムではクライアントサービスを提供できません。

注 - EFI ラベル付きディスク上に存在している UFS ファイルシステムに OS およびディスクレスクライアントサービスを追加しようとする、ディスク容量が不足していることを示す次のようなエラーメッセージが表示されます。

```
The partition /export does not have enough free space.
1897816 KB (1853.34 MB) additional free space required.
Insufficient space available on
/dev/dsk/c0t5d0s0 /export
```

各ディスクレスクライアントについてプラットフォーム、メディアパス、およびクラスタを決定したら、OS サービスを追加する準備ができたことになります。

追加する各 OS サービスについて、次のディレクトリが作成され移植されます。

- `/export/Solaris_version/Solaris_version-instruction-set.all (/export/exec/Solaris_version/Solaris_version-instruction-set.all` へのシンボリックリンク)
- `/export/Solaris_version`

- `/export/Solaris_version/var`
- `/export/Solaris_version/opt`
- `/export/share`
- `/export/root/templates/Solaris_version`
- `/export/root/clone`
- `/export/root/clone/Solaris_version`
- `/export/root/clone/Solaris_version/ machine-class`

追加する各ディスクレスクライアントについて、次のデフォルトのディレクトリが OS サーバー上に作成されます。

- `/export/root/diskless-client`
- `/export/swap/diskless-client`
- `/tftpboot/diskless-client-ipaddress-in-hex/export/dump/diskless-client (-x dump オプションを指定した場合)`

注 - `smosservice` および `smdiskless` コマンドの `x` オプションを使うと、ルート (`/`)、`/swap-`、および `/dump` ディレクトリのデフォルト位置を変更することができます。ただし、`/export` ファイルシステムの下にはこれらのディレクトリを作成しないでください。

▼ x86: GRUB ベースのブート環境にディスクレスクライアントを追加するための準備方法

ディスクレスクライアントの追加の準備をするには、次の手順を実行します。この手順は、x86 システムでの一般的な説明になっています。

`smosservice add` コマンドを使用して OS サービスを追加する場合は、サポートする各ディスクレスクライアントプラットフォームのプラットフォーム、メディアパス、およびクラスタ (またはソフトウェアグループ) を指定する必要があります。

始める前に サポートしているリリースが、OS サービスに指定されたシステム上で実行されていることを確認します。さらに、OS サーバーリリースとディスクレスクライアントリリースの組み合わせがサポートされていることも確認します。詳細は、[152 ページ](#)の「OS サーバーおよびディスクレスクライアントのサポート情報」を参照してください。

- 1 次の形式で使用されるディスクレスクライアントプラットフォームを識別します。
`instruction-set.machine-class.Solaris-version`
次に例を示します。

i386.i86pc.Solaris_10

次に、利用できるプラットフォームのオプションを示します。

命令セット	マシンクラス	Solaris バージョン
sparc	sun4v	Solaris 10 1/06 OS 以降
	sun4u, sun4m, sun4d, および sun4c	Solaris 10, Solaris 9, および Solaris 8
i386	i86pc	Solaris 10, Solaris 9, および Solaris 8

注 - sun-4c アーキテクチャーは、Solaris 8、Solaris 9、Solaris 10 の各リリースではサポートされていません。sun-4d アーキテクチャーは、Solaris 9、Solaris 10 の各リリースではサポートされていません。sun-4m アーキテクチャーは、Solaris 10 リリースではサポートされていません。

- 2 メディアパスを特定します。
- メディアパスは、ディスクレスクライアントにインストールする OS を含むディスクイメージのフルパスです。
- Solaris OS は、複数の CD で配布されます。ただし、smosservice コマンドでは、複数の CD から OS を読み込むことはできません。ユーザーは、次の操作を行うために Solaris SOFTWARE CD (および付属する LANGUAGES CD) にあるスクリプトを実行する必要があります。
- 3 サーバー上にインストールイメージを作成します。インストールサーバーの設定方法については、[『Solaris 10 インストールガイド\(ネットワークインストール\)』](#)を参照してください。
- 4 CD イメージから必要な OS サービスを読み込みます。
- 次のいずれかのスクリプトを使用します。

- CD 1 - /cdrom/cdrom0/Solaris_10/Tools/setup_install_server
- その他の Solaris Software CD - /cdrom/cdrom0/Solaris_10/Tools/add_to_install_server
- Language CD - /cdrom/cdrom0/Solaris_10/Tools/add_to_install_server

たとえば、ローカルに接続された CD-ROM デバイス上の Solaris 10 SOFTWARE - 1 CD から setup_install_server スクリプトを使用する場合は、次のような構文になります。

```
# mkdir /export/install/sol_10_x86
# cd /cd_mount_point/Solaris_10/Tools
```

```
# ./setup_install_server /export/install/sol_10_x86
```

- 5 BootFile および BootSrvA DHCP オプションを DHCP サーバー構成に追加して、PXE ブートを有効にします。

次に例を示します。

```
Boot server IP (BootSrvA) : svr-addr
(BootFile) : 01client-macro
```

svr-addr は OS サーバーの IP アドレスです。*client-macro* は、クライアントの Ethernet タイプ (01) とクライアントの mac アドレスに基づいて作成された名前です。この番号は、インストールサーバーの /tftpboot ディレクトリで使用するファイルの名前としても使われます。

注 - *client-macro* は大文字で表記します。コロンは含めないでください。

これらのオプションは、コマンド行から、または DHCP Manager を使用して追加できます。詳細は、[例 7-4](#) を参照してください。

詳細は、292 ページの「x86: GRUB ベースのブートをネットワークから実行する方法」、『Solaris 10 インストールガイド (ネットワークインストール)』の「DHCP サービスによるシステム構成情報の事前設定 (作業)」、および『Solaris のシステム管理 (IP サービス)』のパート III 「DHCP」を参照してください。

- 6 ディスクに Solaris CD イメージをインストールしたあと、ディスクのメディアパスを書き留めておきます。次に例を示します。

```
/net/export/install/sol_10_x86
```

このディスクのメディアパスは、smosservice コマンドを使用するときに指定する必要があります。

- 7 OS サービスを追加するときに SUNWCxall クラスタを特定します。
同じシステム上で同じ OS が動作しているディスクレスクライアントには、同じクラスタを使用する必要があります。

注 - クラスタには常に SUNWCxall を指定してください。

▼ Solaris 10 OS にディスクレスクライアントを追加するための準備方法

smosservice add コマンドを使用して OS サービスを追加する場合は、サポートする各ディスクレスクライアントプラットフォームのプラットフォーム、メディアパス、およびクラスタ (またはソフトウェアグループ) を指定する必要があります。

始める前に サポートしているリリースが、OS サービスに指定されたシステム上で実行されていることを確認します。さらに、OS サーバーリリースとディスクレスクライアントリリースの組み合わせがサポートされていることも確認します。詳細は、[152 ページ](#)の「OS サーバーおよびディスクレスクライアントのサポート情報」を参照してください。

1 次の形式で使用されるディスクレスクライアントプラットフォームを識別します。

instruction-set.machine-class.Solaris- version

次に例を示します。

sparc.sun4u.Solaris_10

次に、利用できるプラットフォームのオプションを示します。

<i>instruction-set</i>	<i>machine-class</i>	<i>Solaris_version</i>
sparc	sun4v	Solaris 10 1/06 OS 以降
	sun4c、sun4d、sun4m、sun4u	Solaris_10、Solaris_9、および Solaris_8
i386	i86pc	Solaris_10、Solaris_9、および Solaris_8

注 - sun-4c アーキテクチャーは、Solaris 8、Solaris 9、Solaris 10 の各リリースではサポートされていません。sun-4d アーキテクチャーは、Solaris 9、Solaris 10 の各リリースではサポートされていません。sun-4m アーキテクチャーは、Solaris 10 リリースではサポートされていません。

2 メディアパスを特定します。

メディアパスは、ディスクレスクライアントにインストールする OS を含むディスクイメージのフルパスです。

Solaris OS は、複数の CD で配布されます。ただし、smosservice コマンドでは、複数の CD から OS を読み込むことはできません。ユーザーは、次の操作を行うために Solaris SOFTWARE CD (および付属する LANGUAGES CD) にあるスクリプトを実行する必要があります。

3 サーバー上にインストールイメージを作成します。インストールサーバーの設定方法については、『[Solaris 10 インストールガイド \(ネットワークインストール\)](#)』を参照してください。

4 CD イメージから必要な OS サービスを読み込みます。

次のいずれかのスクリプトを使用します。

- Solaris_10 SOFTWARE - 1 CD –
/cdrom/cdrom0/Solaris_10/Tools/setup_install_server
- その他の Solaris Software CD –
/cdrom/cdrom0/Solaris_10/Tools/add_to_install_server
- Language CD – /cdrom/cdrom0/Solaris_10/Tools/add_to_install_server

たとえば、ローカルに接続された CD-ROM デバイス上の Solaris 10 SOFTWARE - 1 CD から setup_install_server スクリプトを使用する場合は、次のような構文になります。

```
# mkdir /export/install/sparc_10
# cd /cd_mount_point/Solaris_10/Tools
# ./setup_install_server /export/install/sparc_10
```

- 5 ディスクに **Solaris** CD イメージをインストールしたあと、ディスクのメディアパスを指定します。次に例を示します。

```
/export/install/sparc_10
```

- 6 **OS** サービスを追加するときに SUNWCxall クラスタを特定します。
同じシステム上で同じ OS が動作しているディスクレスクライアントには、同じクラスタを使用する必要があります。

たとえば、次の Solaris 9 ディスクレスクライアントについて考えてみましょう。

- sparc.sun4m.Solaris_9
- sparc.sun4u.Solaris_9

sun4u および sun4m システムでは SUNWCxall クラスタを要求するため、これらのディスクレスクライアントをセットアップするには、各ディスクレスクライアントに SUNWCxall クラスタを指定する必要があります。また、同じシステム上で同じオペレーティングリリース (この例では Solaris_9) が動作しているディスクレスクライアントには、同じクラスタを使用する必要があります。

注 - sun4u システム、または 8 ビットの高速カラーメモリーフレームバッファ (cgsix) が搭載されたシステムを使用している場合には、クラスタに必ず SUNWCxall を指定してください。

▼ ディスクレスクライアントサポートの **OS** サービスの追加方法

ディスクレスクライアントの OS サービスをサーバーに追加するには、次の手順を実行します。

注 - smossservice add コマンドを使用して OS サービスを追加するときに、ARCH=all タイプのルート(/) パッケージと /usr パッケージはインストールされません。これらのパッケージは読み飛ばされます。警告やエラーメッセージは表示されません。OS サービスを OS サーバーに追加したあとに、失われたパッケージを手動でインストールする必要があります。手順については、「[失われた ARCH=all パッケージを検出してインストールする方法](#)」を参照してください。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 Solaris 管理コンソールサーバーが実行中であり、システムでディスクレスクライアントツールが使用できることを確認します。

```
# /usr/sadm/bin/smossservice list -H host-name:898 --
```

- 3 OS サービスを追加します。

```
# /usr/sadm/bin/smossservice add -H host-name
```

```
:898 -- -o
```

```
host-name
```

```
-x mediapath=path -x platform=
```

```
instruction-set.machine-class
```

```
.Solaris_version
```

```
-x cluster=cluster-name
```

```
-x locale=locale-name
```

```
add
```

指定された OS サービスを追加します。

```
-H host-name:898
```

接続するホスト名とポートを指定します。ポートを指定しない場合は、デフォルトポート 898 に接続されます。

注 - smossservice コマンドを使って OS サービスを追加する場合は、-H オプションは必要ありません。

```
--
```

これ以降のサブコマンド引数が始まることを示します。

```
-x mediapath=path
```

Solaris イメージのフルパスを指定します。

-x *platform=instruction-set.machine-class.Solaris_version*

追加する命令アーキテクチャー、マシンクラス、および Solaris バージョンを指定します。

-x *cluster=cluster-name*

インストールする Solaris クラスタを指定します。

-x *locale=locale-name*

インストールするロケールを指定します。

注 - サーバーの処理速度および選択した OS サービスの構成により、インストールプロセスには 45 分ほどかかることがあります。

詳細は、[smosservice\(1M\)](#) のマニュアルページを参照してください。

- 4 (省略可能) ほかの OS サービスを追加します。
- 5 OS サービスを追加し終わったら、OS サービスがインストールされているかどうか確認します。

```
# /usr/sadm/bin/smosservice list -H host-name:898 --
```

例 7-1 SPARC: ディスクレスクライアントサポートの OS サービスを追加する

この例では、サーバー jupiter で SPARC 版 Solaris 10 の OS サービスを追加する方法を示します。サーバー jupiter では Solaris 10 リリースが動作しています。SPARC 版 Solaris 10 OS の CD イメージは、インストールサーバー myway の /export/s10/combined.s10s_u2wos/61 にあります。

```
# /usr/sadm/bin/smosservice add -H jupiter:898 -- -o jupiter
-x mediapath=/net/myway/export/s10/combined.s10s_u2wos/61
-x platform=sparc.sun4u.Solaris_10
-x cluster=SUNWCXall -x locale=en_US
```

```
# /usr/sadm/bin/smosservice list - H jupiter:898
```

Authenticating as user: root

Type /? for help, pressing enter accepts the default denoted by []

Please enter a string value for: password :: xxxxxx

Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli

from jupiter:898

Login to jupiter as user root was successful.

Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from jupiter:898 was successful.

例 7-2 x86: ディスクレスクライアントサポートの OS サービスを追加する

この例では、サーバー orbit で x86 版 Solaris 10 の OS サービスを追加する方法を示します。サーバー orbit では Solaris10 リリースが動作しています。x86 版 Solaris 10 OS の CD イメージは、インストールサーバー seriously の /export/s10/combined.s10x_u2wos/03 にあります。

```
# /usr/sadm/bin/smosservice add -H orbit:898 -- -o orbit -x
mediapath=/net/seriously/export/s10u2/combined.s10x_u2wos/03 -x
platform=i386.i86pc.Solaris_10 -x cluster=SUNWCxall -x locale=en_US

# /usr/sadm/bin/smosservice list - H orbit:898
Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::
Starting Solaris Management Console server version 2.1.0.
endpoint created: :898
Solaris Management Console server is ready.
Loading Tool: com.sun.admin.osservmgr.cli.OsServerMgrCli from orbit:898
Login to orbit as user root was successful.
Download of com.sun.admin.osservmgr.cli.OsServerMgrCli from orbit:898 was successful.
Client          Root Area
                  Swap Area
                  Dump Area
-----
.
.
.
#
```

次の手順 smosservice add コマンドを実行して OS サービスを OS サーバーに追加したときに失われた ARCH=all パッケージをすべて検出し、それらをインストールします。詳細は、「[失われた ARCH=all パッケージを検出してインストールする方法](#)」を参照してください。

▼ x86: GRUB ベースのブート環境にディスクレスクライアントを追加する方法

Solaris 10 1/06 リリース以降では、OS サービスを追加したあとに、次の手順を実行してディスクレスクライアントを追加します。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

2 ディスクレスクライアントを追加します。

```
# /usr/sadm/bin/smdiskless add -- -i
ip-address -e ethernet-address
-n client-name -x os=
instruction-set.machine-class.Solaris_
version
-x root=/export/root/client-name
-x swap=/export/swap/client-name
-x swapsize=size -x tz=
time-zone -x locale=
locale-name
```

add

指定したディスクレスクライアントを追加します。

--

これ以降のサブコマンド引数が始まることを示します。

-i ip-address

ディスクレスクライアントのIPアドレスを指定します。

-e ethernet-address

ディスクレスクライアントのイーサネットアドレスを識別します。

-n client-name

ディスクレスクライアントの名前を指定します。

-x os=instruction-set.machine-class.Solaris_version

ディスクレスクライアントの命令アーキテクチャー、マシンクラス、OS、および Solaris バージョンを指定します。

-x root=/export/root/client-name

ディスクレスクライアントのルート (/) ディレクトリを指定します。

-x swap=/export/root/client-name

ディスクレスクライアントのスワップファイルを指定します。

-x swapsize=size

スワップファイルのサイズをメガバイト (M バイト) で指定します。デフォルトは 24M バイトです。

-x tz=time-zone

ディスクレスクライアントの時間帯を指定します。

-x locale=locale-name

ディスクレスクライアントをインストールするロケールを指定します。

詳細は、[smdiskless\(1M\)](#) のマニュアルページを参照してください。

- 3 まだ作成されていない場合は、DHCP サーバー構成に BootSrvA および BootFile DHCP オプションを追加して、PXE ブートを有効にします。

次に例を示します。

```
Boot server IP (BootSrvA) : svr-addr
Boot file (BootFile) : 01client-macro
```

svr-addr はサーバーの IP アドレスです。client-macro は、クライアントの Ethernet タイプ (01) とクライアントの mac アドレスに基づいて作成された名前です。この番号は、インストールサーバーの /tftpboot ディレクトリで使用されるファイルの名前としても使われます。

注 - client-macro は大文字で表記します。コロンは含めないでください。

次のファイルとディレクトリが /tftpboot ディレクトリに作成されます。

```
drwxr-xr-x  6 root sys      512 Dec 28 14:53 client-host-name

lrwxrwxrwx  1 root root      31 Dec 28 14:53 menu.lst.01ethernet-address

                                -> /tftpboot/client-host-name/grub/menu.lst

-rw-r--r--  1 root root 118672 Dec 28 14:53 01ethernet-address
```

- 4 コンソールがシリアルポート上にある場合は、/tftpboot/menu.lst.01ethernet-address ファイルを編集します。tty 設定を指定する行のコメントを解除します。

クライアントに作成されるデフォルトの menu.lst ファイルを変更する場合は、/usr/sadm/lib/wbem/config_tftp ファイルの echo 行を編集します。

詳細は、[289 ページの「x86 システムをネットワークからブートする」](#)を参照してください。

- 5 ディスクレスクライアントがインストールされたことを確認します。

```
# /usr/sadm/bin/smdiskless list -H host-name:898 --
```

- 6 (省略可能) smdiskless add コマンドを続けて使用して、各ディスクレスクライアントを追加します。

例 7-3 x86: GRUB ブート環境の x86 システムにディスクレスクライアントサポートを追加する

この例では、Solaris 10 x86 ベースのディスクレスクライアント mikey1 を追加する方法を示します。

```
rainy-01# /usr/sadm/bin/smdiskless add -H sdts-01-qfe0 -- -o sdts-01-qfe0
-n mikey1 -i 192.168.20.22 -e 00:E0:88:55:33:BC -x os=i386.i86pc.Solaris_10
```

```
-x root=/export/root/mikey1 -x swap=/export/swap/mikey1
```

```
Loading Tool: com.sun.admin.osservmgr.cli.OsServerMgrCli
from sdt01-qfe0
Login to rainy-01-qfe0 as user root was successful.
Download of com.sun.admin.osservmgr.cli.OsServerMgrCli from
rainy-01-qfe0 was successful.
```

```
# /usr/sadm/bin/smdiskless list -H mikey1:898 --
Loading Tool: com.sun.admin.osservmgr.cli.OsServerMgrCli from mikey1:898
Login to mikey1 as user root was successful.
Download of com.sun.admin.osservmgr.cli.OsServerMgrCli from mikey1:898 was
successful.
Platform
-----
i386.i86pc.Solaris_10
sparc.sun4us.Solaris_10
sparc.sun4u.Solaris_10
i386.i86pc.Solaris_9
```

例 7-4 x86: BootSrvA および BootFile DHCP オプションを DHCP サーバー構成に追加する

この例では、PXE ブートを有効にするために必要な BootSrvA および BootFile DHCP オプションを追加する方法を示します。

```
rainy-01# pntadm -A mikey1 -m 0100E0885533BC -f 'MANUAL+PERMANENT' \
-i 0100E0885533BC 192.168.0.101
```

```
rainy-01# dhtadm -A -m 0100E0885533BC -d \
":BootSrvA=192.168.0.1:BootFile=0100E0885533BC:"
```

この例のサーバーアドレスはサーバーの IP アドレスであり、クライアントマクロの名前はクライアントの Ethernet タイプ (01) と mac アドレスに基づいて作成されています。この番号は、インストールサーバーの /tftpboot ディレクトリで使用されるファイルの名前としても使われます。クライアントマクロの表記は大文字で構成されます。また、このマクロにコロンを含めないようにしてください。

▼ Solaris 10 OS でディスクレスクライアントを追加する方法

OS サービスを追加した後に、ディスクレスクライアントを追加するには、次の手順を実行します。特に記載している場合を除いて、この手順は、SPARC システムと x86 システムについての一般的な説明になっています。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 ディスクレスクライアントを追加します。

```
# /usr/sadm/bin/smdiskless add -- -i
ip-address -e ethernet-address
-n client-name -x os=
instruction-set.machine-class.Solaris_
version
-x root=/export/root/client-name
-x swap=/export/swap/client-name
-x swappsize=size -x tz=
time-zone -x locale=
locale-name
```

add

指定したディスクレスクライアントを追加します。

--

これ以降のサブコマンド引数が始まることを示します。

-i ip-address

ディスクレスクライアントの IP アドレスを指定します。

-e ethernet-address

ディスクレスクライアントのイーサネットアドレスを識別します。

-n client-name

ディスクレスクライアントの名前を指定します。

-x os=instruction-set.machine-class. Solaris_version

ディスクレスクライアントの命令アーキテクチャー、マシンクラス、OS、および Solaris バージョンを指定します。

-x root=/export/root/client-name

ディスクレスクライアントのルート (/) ディレクトリを指定します。

-x swap=/export/root/client-name

ディスクレスクライアントのスワップファイルを指定します。

-x swappsize=size

スワップファイルのサイズをメガバイト (M バイト) で指定します。デフォルトは 24M バイトです。

-x tz=time-zone

ディスクレスクライアントの時間帯を指定します。

`-x locale=locale-name`

ディスクレスクライアントをインストールするロケールを指定します。

詳細は、[smdiskless\(1M\)](#) のマニュアルページを参照してください。

- 3 (省略可能) `smdiskless add` コマンドを続けて使用して、各ディスクレスクライアントを追加します。
- 4 ディスクレスクライアントがインストールされたことを確認します。

```
# /usr/sadm/bin/smdiskless list -H host-name:898 --
```

例 7-5 SPARC: ディスクレスクライアントサポートを SPARC システムに追加する

この例では、サーバー `bearclaus` から Solaris 10 sun4u ディスクレスクライアント `starlite` を追加する方法を示します。

```
# /usr/sadm/bin/smdiskless add -- -i 172.20.27.28 -e 8:0:20:a6:d4:5b
-n starlite -x os=sparc.sun4u.Solaris_10 -x root=/export/root/starlite
-x swap=/export/swap/starlite -x swappsize=128 -x tz=US/Mountain
-x locale=en_US
```

```
# /usr/sadm/bin/smdiskless list -H starlite:898 --
Loading Tool: com.sun.admin.osservmgr.cli.OsServerMgrCli from line2-v480:898
Login to line2-v480 as user root was successful.
Download of com.sun.admin.osservmgr.cli.OsServerMgrCli from line2-v480:898 was
successful.
```

Platform

```
-----
i386.i86pc.Solaris_10
sparc.sun4us.Solaris_10
sparc.sun4u.Solaris_10
i386.i86pc.Solaris_9
sparc.sun4m.Solaris_9
sparc.sun4u.Solaris_9
sparc.sun4us.Solaris_9
```

`smdiskless list -H` コマンド出力には SPARC と x86 の両方のシステムが表示されます。

例 7-6 x86: Solaris 10 OS の x86 システムにディスクレスクライアントサポートを追加する

この例では、サーバー `bearclaus` から Solaris 10 x86 のディスクレスクライアント `mars` を追加する方法を示します。

```
# /usr/sadm/bin/smdiskless add -- -i 172.20.27.176 -e 00:07:E9:23:56:48
-n mars -x os=i386.i86pc.Solaris_10 -x root=/export/root/mars
-x swap=/export/swap/mars -x swappsize=128 -x tz=US/Mountain
-x locale=en_US
```

▼ x86: GRUB を使用してディスクレスクライアントをブートする方法

Solaris 10 1/06 OS 以降をインストールした場合、または Solaris 10 1/06 OS 以降にアップグレードした場合は、ディスクレスクライアントのブート手順が変更されています。GRUB を使用してディスクレスクライアントをブートする場合は、次の手順で行います。

注 - Solaris 10 6/06 リリース以降では、GRUB のフェイルセーフ対話操作が変更されています。フェイルセーフアーカイブをブートするときに、ブートアーカイブの自動更新を要求されることがなくなりました。ブートアーカイブの更新を要求されるのは、矛盾したブートアーカイブが検出された場合だけです。詳細は、[284 ページの「GRUB を使用して x86 システムでフェイルセーフアーカイブをブートする方法」](#)を参照してください。

始める前に システムをネットワークブートできるように、OS サーバーが次の要件を満たしていることを確認してください。

- ディスクレスクライアントおよび OS サービスの追加に使用するネームサービスがサーバーの /etc/nsswitch.conf ファイルのプライマリ名に一致していることを確認します。
- DHCP サービスと tftp ブートサービスが実行中であることを確認します。
- システムをネットワークブートするために、システム BIOS の PXE ROM オプションが有効になるように設定します。

PXE 対応ネットワークアダプタの中には、ブート時にしばらく表示されるプロンプトに対して特定のキーを押すと、PXE ブートを実行する機能を持つものがあります。ブートの優先順位を BIOS で設定する方法については、ハードウェアのマニュアルを参照してください。

- 1 適切なキーの組み合わせを入力して、ディスクレスクライアントをブートします。GRUB メニューが表示されます。

ネットワークインストールサーバーの設定によっては、使用するシステムに表示される GRUB メニューがここで示した GRUB メニューと異なる可能性があります。

- 2 矢印キーを使ってブートエントリを選択して、**Enter** キーを押します。
選択しない場合は、数秒後にデフォルトの OS インスタンスが自動的にブートします。
- ブート時に **GRUB** メニューを編集して **GRUB** カーネル動作を変更する必要がある場合には、矢印キーを使ってブートエントリを選択したあと、**e** と入力してそのエントリを編集します。

注-この例は、GRUB マルチブート実装を示しています。GRUB メニューは、実行している Solaris リリースに応じて異なります。

編集するブートコマンドが、GRUB 編集画面に表示されます。

ブート時にカーネルの動作を変更する方法については、[第 11 章「Solaris ブート動作の変更\(手順\)」](#)を参照してください。

- 編集を保存して **GRUB** メニューに戻るには、**Enter** キーを押します。
GRUB メニューが開き、ブートコマンドに加えた編集が表示されます。
- **b** キーを押して、ネットワークからシステムをブートします。

▼ SPARC: Solaris 10 OS でディスクレスクライアントをブートする方法

始める前に OS サーバーの次の必要条件を確認します。

- ディスクレスクライアントおよび OS サービスの追加に使用するネームサービスがサーバーの `/etc/nsswitch.conf` ファイルのプライマリ名に一致していることを確認します。
一致しない場合は、ディスクレスクライアントがブートしません。
- `rpc.bootparamd` デーモンが実行中かどうか確認します。実行中でない場合には起動します。

- ディスクレスクライアントをブートします。

`ok boot net`

▼ ディスクレスクライアントサポートの削除方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。
- 2 ディスクレスクライアントサポートを削除します。

```
# /usr/sadm/bin/smdiskless delete -- -o host-name  
:898 -n client-name
```
- 3 ディスクレスクライアントサポートが削除されたことを確認します。

```
# /usr/sadm/bin/smosservice list -H host-name:898 --
```

例 7-7 ディスクレスクライアントサポートを削除する

この例では、OS サーバー starlite からディスクレスクライアント holoship を削除する方法を示します。

```
# /usr/sadm/bin/smdiskless delete -- -o starlite:898 -n holoship  
  
Authenticating as user: root  
  
Type /? for help, pressing enter accepts the default denoted by [ ]  
Please enter a string value for: password ::  
Starting SMC server version 2.0.0.  
endpoint created: :898  
SMC server is ready.  
  
# /usr/sadm/bin/smosservice list -H starlite:898 --  
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from starlite  
Login to starlite as user root was successful.  
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starlite  
was successful.
```

▼ ディスクレスクライアントの OS サービスを削除する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 ディスクレスクライアントの OS サービスを削除します。

```
# /usr/sadm/bin/smosservice delete -H $HOST:$PORT -u root -p $PASSWD --
-x instruction-set.all.Solaris_version
```

注 - machine-class は all のみがサポートされています。

- 3 OS サービスが削除されたことを確認します。

```
# /usr/sadm/bin/smosservice list -H host-name:898 --
```

例 7-8 ディスクレスクライアントの OS サービスを削除する

次の例では、サーバー starlite からディスクレスクライアントの OS サービス (sparc.all.Solaris_10) を削除する方法を示します。

```
# /usr/sadm/bin/smosservice delete -H starlite:898 -u root
-p xxxxxx -- -x sparc.all.solaris_10
Authenticating as user: root
Type /? for help, pressing enter accepts the default denoted by [ ]
Please enter a string value for: password ::

# /usr/sadm/bin/smosservice list -H starlite:898 --
Loading Tool: com.sun.admin.osservmgr.cli.OsServerMgrCli from starlite:898
Login to starlite as user root was successful.
Download of com.sun.admin.osservmgr.cli.OsServerMgrCli from starlite:898
was successful
```

ディスクレスクライアント OS サービスにパッチを適用する

smosservice patch コマンドを使うと次の操作を実行できます。

- OS サーバー上に /export/diskless/Patches パッチスプールディレクトリを設定します。
- パッチスプールディレクトリにパッチを追加します。追加するパッチにより、スプールの既存のパッチが不要になると、不要なパッチは /export/diskless/Patches/Archive に移動されます。
- パッチスプールディレクトリからパッチを削除します。
- パッチスプールディレクトリ内のパッチをリスト表示します。
- スプールされたパッチをクライアントに同期させます。クライアントが更新されたパッチを認識できるように、同期させた各クライアントをリブートする必要があります。

注- 推奨 OS パッチを定期的にインストールして、OS サーバーを最新の状態に保ってください。

パッチのダウンロード方法については、[483 ページの「Solaris パッチをダウンロードおよび適用する方法」](#)を参照してください。

ディスクレスクライアントの OS パッチの表示

ディスクレスクライアントのパッチは、パッチのタイプに応じて、異なるディレクトリにロギングされます。

- カーネルパッチは、ディスクレスクライアントの `/var/sadm/patch` ディレクトリにロギングされます。カーネルパッチを表示するには、ディスクレスクライアントで次のコマンドを入力します。

```
% patchadd -p
```

注- このコマンドを実行するには、ディスクレスクライアントにログインしている必要があります。OS サーバー上で `patchadd -p` コマンドを実行すると、OS サーバー用のカーネルパッチだけが表示されます。

- `/usr` パッチは、OS サーバーの `/export/Solaris_version/var/patch` ディレクトリにロギングされます。パッチ ID ごとにディレクトリが作成されます。`/usr` パッチを表示するには、OS サーバーで次のコマンドを入力します。

```
% patchadd -S Solaris_version -p
```

```
Patch: 111879-01 Obsoletes: Requires: Incompatibles: Packages: SUNWwsr
```

OS およびアーキテクチャーによりスプールされたすべてのパッチを表示するには、`-P` オプションを指定した `smosservice` コマンドを使用します。

▼ ディスクレスクライアントの OS パッチの追加方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 ディスクレスクライアントシステムにログインし、シャットダウンします。

```
# init 0
```

- 3 パッチをスプールディレクトリに追加します。

```
# /usr/sadm/bin/smosservice patch -- -a /var/patches/  
patch-ID-revision
```

追加するパッチが別のパッチに依存する場合は、次のメッセージが表示され、パッチを追加できません。

```
The patch patch-ID-revision could not be added  
because it is dependent on other patches which have not yet been spooled.  
You must add all required patches to the spool first.
```

- 4 パッチがスプールされたことを確認します。

```
# /usr/sadm/bin/smosservice patch -- -P
```

- 5 スプールされたパッチをディスクレスクライアントにプッシュします。

```
# /usr/sadm/bin/smosservice patch -- -m -U
```

注-パッチをディスクレスクライアントにプッシュおよび同期させるには、1パッチあたり最大90分ほどかかることがあります。

- 6 パッチがディスクレスクライアントに適用されたことを確認します。

```
# /usr/sadm/bin/smosservice patch -- -P
```

例7-9 ディスクレスクライアントのOSパッチを追加する

この例では、Solaris 8 パッチ (111879-01) をサーバーのディスクレスクライアントのOS サービスに追加する方法を示します。

```
# /usr/sadm/bin/smosservice patch -- -a /var/patches/111879-01
```

```
Authenticating as user: root
```

```
Type /? for help, pressing <enter> accepts the default denoted by [ ]  
Please enter a string value for: password ::  
Loading Tool: com.sun.admin.osservmgr.cli.OsServerMgrCli from starlite  
Login to starlite as user root was successful.  
Download of com.sun.admin.osservmgr.cli.OsServerMgrCli from starlite  
was successful..
```

```
# /usr/sadm/bin/smosservice patch -- -P
```

```
Patches In Spool Area
```

```
Os Rel Arch  Patch Id  Synopsis
-----
8          sparc  111879-01 SunOS 5.8: Solaris Product Registry patch SUNWwsr

Patches Applied To OS Services
Os Service                                     Patch
-----
Solaris_8

Patches Applied To Clone Areas
Clone Area                                     Patch
-----
Solaris_8/sun4u          Patches In Spool Area
Os Rel Arch  Patch Id  Synopsis
-----
8          sparc  111879-01 SunOS 5.8: Solaris Product Registry patch SUNWwsr
.
.
.
# /usr/sadm/bin/smosservice patch -- -m -U
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.sun.admin.ossvermgr.cli.OsServerMgrCli from starlite
Login to starlite as user root was successful.
Download of com.sun.admin.ossvermgr.cli.OsServerMgrCli from starlite
was successful.

# /usr/sadm/bin/smosservice patch -- -P
Authenticating as user: root
.
.
.
Patches In Spool Area
Os Rel Arch  Patch Id  Synopsis
-----
8          sparc  111879-01 SunOS 5.8: Solaris Product Registry patch SUNWwsr

Patches Applied To OS Services
Os Service                                     Patch
-----
Solaris_8

Patches Applied To Clone Areas
Clone Area                                     Patch
-----
Solaris_8/sun4u
```

ディスクレスクライアント問題のトラブルシューティング

この節では、ディスクレスクライアントを管理するときに発生する問題と、それらに適用できる解決方法について説明します。

ディスクレスクライアントのインストールの問題

`smosservice add` コマンドを実行しても、ルート (/) および `/usr` ファイルシステム内の `ARCH=all` が指定されているパッケージは、いずれもインストールされません。つまり、これらのパッケージは読み飛ばされます。警告やエラーメッセージは表示されません。これらのパッケージは、新規に作成した Solaris OS サービスに手動で追加する必要があります。この作業は、Solaris 2.1 OS 以降で発生します。この作業は、SPARC および x86 の両方のクライアントに適用されます。失われるパッケージのリストは、稼働している Solaris OS によって異なります。

▼ 失われた `ARCH=all` パッケージを検出してインストールする方法

ここでは、Solaris OS サービスをサーバー上に作成したあとに、失われた `ARCH=all` パッケージを検出してインストールする方法について説明します。この手順で取り上げる例は、Solaris 10 6/06 OS を対象としています。

1 `ARCH=all` パラメータが指定されたすべてのパッケージを検出します。

- a. Solaris 10 イメージ用のメディアの Product ディレクトリに移動します。次に例を示します。

```
% cd /net/server/export/Solaris/s10u2/combined.s10s_u2wos/latest/Solaris_10/Product
```

- b. `pkginfo` ファイルにあるパッケージのうち、`ARCH=all` パラメータが指定されているパッケージの一覧を表示します。

```
% grep -w ARCH=all */pkginfo
```

引数のリストが長すぎるというエラーメッセージが表示される場合は、次のコマンドを代わりに実行してリストを生成できます。

```
% find . -name pkginfo -exec grep -w ARCH=all {} /dev/null \;
```

このコマンドは、結果の生成に少し時間がかかります。

出力は、次のようになります。

```
./SUNWjdmk-base/pkginfo:ARCH=all
./SUNWjhdev/pkginfo:ARCH=all
./SUNWjhrt/pkginfo:ARCH=all
```

```
./SUNWjhdem/pkginfo:ARCH=all
./SUNWjhdoc/pkginfo:ARCH=all
./SUNWmllibk/pkginfo:ARCH=all
```

このリストの情報を使用して、`/usr` ファイルシステムにインストールされているパッケージと、ルート (`/`) ファイルシステムにインストールされているパッケージを特定できます。

- c. 生成されたパッケージリストで、`SUNW_PKGTYPE` パラメータの値を確認します。

`/usr` ファイルシステムに属しているパッケージには、`pkginfo` ファイルで `SUNW_PKGTYPE=usr` が設定されています。ルート (`/`) ファイルシステムに属しているパッケージには、`pkginfo` ファイルで `SUNW_PKGTYPE=root` が設定されています。この出力では、すべてのパッケージが `/usr` ファイルシステムに属していません。

- 2 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 3 一時インストール管理ファイルを作成します。

ルート (`/`) ファイルシステムにインストールされているパッケージと、`/usr` ファイルシステムにインストールされているパッケージには、それぞれ個別にインストール管理ファイルを作成する必要があります。

- `/usr` ファイルシステムにインストールされている `ARCH=all` パッケージのために、次の一時インストール管理ファイルを作成します。

```
# cat >/tmp/admin_usr <<EOF
mail=
instance=unique
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
basedir=/usr_sparc.all
EOF
#
```

- ルート (`/`) ファイルシステムにインストールされている `ARCH=all` パッケージのために (パッケージが存在する場合)、次の一時インストール管理ファイルを作成します。


```
# cat >/tmp/admin_root <<EOF
mail=
instance=unique
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
EOF
#
```

4 失われた ARCH=all パッケージをインストールします。

- a. 現在のディレクトリが **Solaris 10** イメージ用のメディアの **Product** ディレクトリでない場合は、このディレクトリに移動します。次に例を示します。

```
# cd /net/server/export/Solaris/s10u2/combined.s10s_u2wos/latest/Solaris_10/Product
pwd コマンドを実行して、現在のディレクトリを確認できます。
```

- b. /usr ファイルシステム内の失われた ARCH=all パッケージをインストールします。

```
# pkgadd -R /export/Solaris_10 -a /tmp/admin_usr -d 'pwd' [
package-A package-B ...]
```

pkgadd コマンドを実行するときに、複数のパッケージを指定できます。

- c. ARCH=all パッケージがインストールされたことを確認します。

```
# pkginfo -R /export/Solaris_10 [package-A
package-B ...]
```

- d. ルート (/) ファイルシステム内の失われた ARCH=all パッケージをインストールします。

これらのパッケージが存在しないこともあります。

```
# pkgadd -R /export/root/clone/Solaris_10/sun4u -a /tmp/admin_root -d 'pwd' [
package-X package-Y ...]
```

- e. ARCH=all パッケージがインストールされたことを確認します。

```
# pkginfo -R /export/root/clone/Solaris_10/sun4u [
package-X package-Y ...]
```

5 失われた ARCH=all パッケージの追加が完了したら、一時インストール管理ファイルを削除します。

```
# rm /tmp/administration-file
```

例 7-10 失われた ARCH=all パッケージを検出してインストールする

この例では、失われた ARCH=all パッケージ SUNWjdmk-base を /usr ファイルシステムにインストールする方法を示します。

```
% uname -a
SunOS t1fac46 5.10 Generic_118833-02 sun4u sparc SUNW,UltraSPARC-IIi-cEngine

% cat /etc/release
Solaris 10 6/06 s10s_u2wos_03 SPARC
Copyright 2006 Sun Microsystems, Inc. All Rights Reserved.
Use is subject to license terms.
Assembled 06 February 2006

% cd /net/ventor/export/Solaris/s10u2/combined.s10s_u2wos/latest/Solaris_10/Product

% grep -w ARCH=all */pkginfo
Arguments too long

% find . -name pkginfo -exec grep -w ARCH=all {} /dev/null \;
./SUNWjdmk-base/pkginfo:ARCH=all
./SUNWjhdev/pkginfo:ARCH=all
./SUNWjhrt/pkginfo:ARCH=all
./SUNWjhdem/pkginfo:ARCH=all
./SUNWjhdoc/pkginfo:ARCH=all
./SUNWmllibk/pkginfo:ARCH=all

% grep -w SUNW_PKGTYPE=usr ./SUNWjdmk-base/pkginfo ./SUNWjhdev/pkginfo ...

./SUNWjdmk-base/pkginfo:SUNW_PKGTYPE=usr
./SUNWjhdev/pkginfo:SUNW_PKGTYPE=usr
./SUNWjhrt/pkginfo:SUNW_PKGTYPE=usr
./SUNWjhdem/pkginfo:SUNW_PKGTYPE=usr
./SUNWjhdoc/pkginfo:SUNW_PKGTYPE=usr

% grep -w SUNW_PKGTYPE=root ./SUNWjdmk-base/pkginfo ./SUNWjhdev/pkginfo ...

% su
Password: xxxxxx

# cat >/tmp/admin_usr <<EOF
mail=
instance=unique
partial=nocheck
runlevel=nocheck
```

```
idepend=nochec> k
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
basedir=/usr_sparc.all
EOF

# pwd
/net/ventor/export/Solaris/s10u2/combined.s10s_u2wos/latest/Solaris_10/Product

# pkginfo -R /export/Solaris_10 SUNWjdmk-base
ERROR: information for "SUNWjdmk-base" was not found

# pkgadd -R /export/Solaris_10 -a /tmp/admin_usr -d 'pwd' SUNWjdmk-base

Processing package instance <SUNWjdmk-base> </net/ventor/export/Solaris/s10u2/combined.s10s_u2wos...

Java DMK 5.1 minimal subset(all) 5.1,REV=34.20060120
Copyright 2005 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Using </export/Solaris_10/usr_sparc.all>
## Processing package information.
## Processing system information.

Installing Java DMK 5.1 minimal subset as <SUNWjdmk-base>

## Installing part 1 of 1.
2438 blocks

Installation of <SUNWjdmk-base> was successful.

# pkginfo -R /export/Solaris_10 SUNWjdmk-base
application SUNWjdmk-base Java DMK 5.1 minimal subset

# rm /tmp/admin_usr
```

ディスクレスクライアントの一般的な問題のトラブルシューティング

この節では、ディスクレスクライアントで発生する可能性がある一般的な問題とその解決策について説明します。

問題: /usr ファイルシステムが nobody によって所有されているときにログインを試みると、ディスクレスクライアントから Owner of the module /usr/lib/security/pam_unix_session.so.1 is not root というメッセージが出力される。

対処方法: この問題を修正するには、次の対処方法に従います。

1. テキストエディタを使用して、ディスクレスクライアントの server:/export/root/client/etc/default/nfs ファイルを変更します。
2. #NFSMAPID_DOMAIN=domain 行を次のように変更します。

```
NFSMAPID_DOMAIN=the_same_value_as_in_server's /var/run/nfs4_domain
```

3. OS サーバーとディスクレスクライアントが同じ nfsmapid ドメインを使用していることを確認します。この情報を確認するには、/var/run/nfs4_domain ファイルを調べます。



注意-ディスクレスクライアントの nfs4_domain ファイル内の値が、OS サーバーの /var/run/nfs4_domain ファイル内の値と異なっている場合は、ディスクレスクライアントのブート後にシステムにログインすることができません。

4. ディスクレスクライアントをリブートします。

詳細は、『Solaris カーネルのチューンアップ・リファレンスマニュアル』の第3章「NFS チューニング可能パラメータ」および nfsmapid(1M) のマニュアルページを参照してください。

問題: OS サーバーが次の処理に失敗する。

- クライアントの RARP (Reverse Address Resolution Protocol) 要求に対する応答
- クライアントの bootparam 要求に対する応答
- ディスクレスクライアントのルート (/) ファイルシステムのマウント

対処方法: ファイル環境で次の解決法を適用できます。

- OS サーバーの /etc/nsswitch.conf ファイルの hosts、ethers、および bootparams の最初の参照元として files が設定されていることを確認します。
- クライアントの IP アドレスが /etc/inet/hosts ファイルに含まれていることを確認します。

注 - Solaris 10 8/07 以降のリリースを実行していない場合、クライアントの IP アドレスが `/etc/inet/ipnodes` ファイルに含まれていることも確認する必要があります。

この Solaris リリースでは、2つの異なる `hosts` ファイルを持たなくなりました。`/etc/inet/hosts` ファイルが、IPv4 エントリと IPv6 エントリの両方を含む単一のファイルとなります。常に同期させる必要がある2つの `hosts` ファイルに、IPv4 エントリを保持する必要はありません。`/etc/inet/ipnodes` ファイルは、下位互換性のために、`/etc/inet/hosts` ファイルへの同名のシンボリックリンクに置き換えられています。詳細は、[hosts\(4\)](#) のマニュアルページを参照してください。

- クライアントのイーサネットアドレスが `/etc/ethers` ファイルに表示されていることを確認します。
- `/etc/bootparams` ファイルにクライアントのルート (*/*) ディレクトリとスワップ空間への次のパスが含まれていることを確認します。

```
client root=os-server:/export/root/
client swap=os-server:
/export/swap/client
```

スワップのサイズは、ディスクレスクライアントの追加時に `-x swapsize` オプションを指定したかどうかによって異なります。ディスクレスクライアントの追加時に `-x dump` オプションを指定した場合は、次の行が表示されます。

```
dump=os-server:/export/dump/client
dumpsize=512
```

ダンプサイズも、ディスクレスクライアントの追加時に `-x dumpsize` オプションを指定したかどうかによって異なります。

- OS サーバーの IP アドレスが `/export/root/client/etc/inet/hosts` ファイルに設定されていることを確認します。

問題: OS サーバーが次の処理に失敗する。

- クライアントの RARP 要求に対する応答
- クライアントの `bootparam` 要求に対する応答
- ディスクレスクライアントのルート (*/*) ファイルシステムのマウント

対処方法: ネームサービス環境で次の解決法を適用できます。

- OS サーバーとクライアントのイーサネットアドレスおよび IP アドレスが正しくマップされていることを確認します。
- `/etc/bootparams` ファイルにクライアントのルート (*/*) ディレクトリとスワップ空間へのパスが含まれていることを確認します。

```
client root=os-server:/export/
root/client swap=os-server:/export/
swap/client swapsize=24
```

スワップのサイズは、ディスクレスクライアントの追加時に `-x swapsize` オプションを指定したかどうかによって異なります。ディスクレスクライアントの追加時に `-x dump` オプションを指定した場合は、次の行が表示されます。

```
dump=os-server:/export/dump/
client dumpsize=24
```

ダンプサイズも、ディスクレスクライアントの追加時に `-x dumpsize` オプションを指定したかどうかによって異なります。

問題: ディスクレスクライアントパニック

対処方法: 次の内容を確認します。

- OS サーバーのイーサネットアドレスが IP アドレスに正しくマップされていることを確認します。システムをあるネットワークから別のネットワークに物理的に移動した場合、システムの新しい IP アドレスを再マップするのを忘れている可能性があります。
- クライアントの RARP、TFTP (Trivial File Transfer Protocol)、または `bootparam` 要求に応答する「同じサブネット」にある別のサーバーのデータベースに、クライアントのホスト名、IP アドレス、およびイーサネットアドレスが存在しないことを確認します。インストールサーバーから OS をインストールするために、しばしばテストシステムがセットアップされます。このような場合、インストールサーバーはクライアントの RARP または `bootparam` 要求に対して、正しくない IP アドレスを返します。この不正なアドレスにより、間違ったアーキテクチャーのブートプログラムをダウンロードしたり、クライアントのルート (/) ファイルシステムのマウントが失敗したりしている可能性があります。
- ディスクレスクライアントの TFTP 要求にインストールサーバー (または以前の OS サーバー) が応答しないことを確認します。このサーバーは不正なブートプログラムを転送しています。ブートプログラムのアーキテクチャーが異なる場合は、クライアントがただちにパニックになります。ブートプログラムが非 OS サーバーから読み込まれた場合、クライアントはそのルートパーティションを非 OS サーバー上に確保し、`/usr` パーティションを OS サーバー上に確保することがあります。この状況では、ルートおよび `/usr` パーティションのアーキテクチャーまたはバージョンが競合する場合、クライアントがパニックに陥ります。
- インストールサーバーと OS サーバーの両方を使用している場合は、`/etc/dfs/dfstab` ファイルに次のエントリがあることを確認してください。

```
share -F nfs -o -ro
/export/exec/Solaris_version- \
instruction-set.all/usr
```

ここで、*version* は 8、9、10 であり、*instruction-set* は *sparc* または *i386* です。

- ディスクレスクライアントのルート (*/*)、*/swap*、*/dump* (指定されている場合) の各パーティションに次の共有エントリがあることを確認します。

```
share -F nfs -o rw=client,root=client
/export/root/client
share -F nfs -o rw=client,root=client /export/swap/
client
share -F nfs -o rw=client,root=client /export/dump/
client
```

- OS サーバーで次のコマンドを入力し、共有されているファイルを確認します。

% **share**

OS サーバーでは、ディスクレスクライアントを追加したときに指定した */export/root/client* および */export/swap/client_name* (デフォルト)、またはルート、*/swap*、*/dump* の各パーティションを共有する必要があります。

次のエントリが */etc/dfs/dfstab* ファイルにあることを確認します。

```
share -F nfs -o ro /export/exec/Solaris_version-
instruction-set.all/usr
share -F nfs -o rw=client,root=client /export/root/
client
share -F nfs -o rw=client,root=client /export/swap/
client
```

問題: OS サーバーがディスクレスクライアントの RARP 要求に応答しない

対処方法: クライアントの目的の OS サーバーから、次のクライアントのイーサネットアドレスを使って *snoop* コマンドをスーパーユーザー (*root*) として実行します。

```
# snoop xx:xx:xx:xx:xx:xx
```

問題: ブートプログラムをダウンロードしたが、プロセスの初期にパニックが発生した

対処方法: *snoop* コマンドを使用して、対象の OS サーバーがクライアントの TFTP および NFS 要求に応答するかどうかを確認します。

問題: ディスクレスクライアントがハングアップする

対処方法: OS サーバーで次のデーモンを再起動します。

```
# /usr/sbin/rpc.bootparamd
# /usr/sbin/in.rarpd -a
```

問題: ディスクレスクライアントの RARP 要求に対するサーバーの応答が不正である
対処方法: OS サーバーで次のデーモンを再起動します。

```
# /usr/sbin/rpc.bootparamd  
# svcadm enable network/rarp
```


システムのシャットダウンとブートの概要

Solaris オペレーティングシステム (Solaris OS) は、電子メールとネットワークリソースをいつでも利用できるように、停止することなく動作するように設計されています。この章では、システムのシャットダウンとブートについて概要を説明します。

この章の内容は次のとおりです。

- 193 ページの「システムのシャットダウンとブートに関する新機能」
- 196 ページの「シャットダウンとブートについての参照先」
- 197 ページの「シャットダウンとブートの用語」
- 198 ページの「システムのシャットダウンに関するガイドライン」
- 199 ページの「システムのブートに関するガイドライン」
- 199 ページの「システムをシャットダウンする場合」
- 200 ページの「システムをブートする場合」

Solaris リリースで利用できるすべてのブート機能および方法の概要は、[第9章「システムのシャットダウンとブート \(概要\)」](#)を参照してください。

Solaris システムのブート手順については、[第12章「Solaris システムのブート \(手順\)」](#)を参照してください。

システムのシャットダウンとブートに関する新機能

この節では、この Solaris リリースでの新しいブート機能について説明します。Solaris の新機能の一覧および Solaris リリースについての説明は、『[Solaris 10 の概要](#)』を参照してください。

Solaris リリースのインストールおよびブートでの 2T バイトディスクのサポート

Solaris 10 10/09: 以前の Solaris リリースでは、1T バイトを超えるディスクに Solaris OS をインストールしてブートすることはできませんでした。この Solaris リリースからは、最大 2T バイトのディスクに Solaris OS をインストールしてブートできます。以前のリリースでは、1T バイトを超えるディスクには EFI ラベルを使用する必要がありました。このリリースでは、どのサイズのディスクでも VTOC ラベルを使用できます。ただし、VTOC ラベルでアドレス指定可能な領域は 2T バイトに制限されます。

詳細は、『[Solaris のシステム管理 \(デバイスとファイルシステム\)](#)』の「[ディスク管理の新機能](#)」を参照してください。

ZFS ブートのサポート

Solaris 10 10/08 リリースには、ZFS™ インストール、および ZFS ブートサポートが含まれます。ZFS ルートファイルシステムをインストールしてブートできるようになりました。この実装は、SPARC と x86 の両方のシステムに適用されます。ブート方法、システム操作、およびインストール手順は、この変更をサポートするように修正されました。

詳細は、[211 ページ](#)の「[ZFS ルートファイルシステムからの起動](#)」を参照してください。

x86: 新しい findroot コマンド

Solaris Live Upgrade など、すべての Solaris インストール方法では、ブート対象の x86 システム上のディスクスライスを指定する際に `findroot` コマンドを使用するようになりました。この実装は、UFS ルートだけでなく ZFS ルートを含むシステムのブートにも適用されます。以前は、`root` コマンド、`root (hd0.0.a)` を使ってブート対象のディスクスライスを明示的に指定していました。この情報は、GRUB で使用される `menu.lst` ファイルに格納されています。

GRUB `menu.lst` エントリのもっとも一般的な書式は次のとおりです。

```
findroot (pool_rpool,0,a)
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive
```

詳細は、[246 ページ](#)の「[x86: findroot コマンドの実装](#)」を参照してください。

bootadm コマンドによるプラットフォーム指定のサポート

bootadm コマンドに -p オプションが新たに追加されました。

このオプションを使用すると、クライアントプラットフォームがサーバープラットフォームと異なる場合 (ディスクレスクライアントを管理している場合など) に、クライアントシステムのプラットフォームまたはマシンのハードウェアクラスを指定できます。

注 - -p オプションは、-R オプションと組み合わせて使用する必要があります。

```
# bootadm -p platform -R [altroot]
```

次のいずれかのプラットフォームを指定します。

- i86pc
- sun4u
- sun4v

詳細は、[bootadm\(1M\)](#) のマニュアルページを参照してください。

再設計された Solaris SPARC ブートストラッププロセス

Solaris SPARC ブートストラッププロセスは、Solaris x86 ブートアーキテクチャーとの共通性を高めるために再設計されました。

そのほかの機能拡張として、ブートアーキテクチャーの改善があげられます。DVD、NFS、または HTTP からシステムをブートできるだけでなく、追加のファイルシステムタイプ (ZFS ファイルシステムなど) やインストール用の単一のミニルートからもブートできるようになりました。これらの機能拡張により、SPARC システムの柔軟性が向上し、保守の必要性が軽減されます。

この再設計の一貫として、以前は Solaris x86 プラットフォームでしか使用できなかった Solaris ブートアーカイブと bootadm コマンドが Solaris SPARC ブートアーキテクチャーに不可欠な要素となりました。

SPARC と x86 のブートアーキテクチャーの主な違いは、ブート時のブートデバイスおよびファイルの選択方法です。SPARC プラットフォームでは、引き続き OpenBoot™ PROM (OBP) を主要な管理インタフェースとして使用し、OBP コマンドを使ってブートオプションを選択します。x86 システムでは、BIOS と GRUB (GRand Unified Bootloader) メニューを介してブートオプションを選択します。

注 - Solaris SPARC ブートの実装は変更されましたが、SPARC システムをブートするための管理手順にはまったく影響はありません。システム管理者が行うブート作業は、ブートアーキテクチャーを設計し直す前と何も変わりません。

詳細は、[boot\(1M\)](#) および [bootadm\(1M\)](#) のマニュアルページを参照してください。

また、このマニュアルの [205 ページ](#) の「[新しい Solaris SPARC ブートアーキテクチャーについて](#)」も参照してください。

x86: 電源ボタンによるシステムシャットダウン開始のサポート

x86 システムで電源ボタンを押して離すと、システムの正常なシャットダウンが開始され、システムの電源が切れます。この機能は、`init 5` コマンドを使ってシステムをシャットダウンするのと同様です。一部の x86 システムでは、BIOS の設定によって、電源ボタンを押してもシャットダウンが開始されない可能性があります。電源ボタンを使ってシステムの正常なシャットダウンを実行できるようにするには、BIOS を設定し直します。

注 - 1999 年よりも前に製造され、かつ古い Solaris リリースが稼働する一部の x86 システム上では、電源ボタンを押すとすぐにシステムの電源が切れてしまい、正しくシャットダウンされません。`acpi-user-options` によって無効化された ACPI サポートを使用して稼働しているシステム上で電源ボタンを押しても、これと同じ動作が起きます。

`acpi-user-options` の詳細は、[eeprom\(1M\)](#) のマニュアルページを参照してください。

シャットダウンとブートについての参照先

システムをシャットダウンおよびブートする手順については、次を参照してください。

シャットダウンとブート作業	詳細
SPARC システムまたは x86 システムのシャットダウン	第 10 章「システムのシャットダウン (手順)」
ブート動作の変更	第 11 章「Solaris ブート動作の変更 (手順)」

シャットダウンとブート作業	詳細
SPARC システムまたは x86 システムのブート	第 12 章「Solaris システムのブート(手順)」
Solaris ブートアーカイブの管理	第 14 章「Solaris ブートアーカイブの管理(手順)」
SPARC または x86 システムでのブート動作のトラブルシューティング	295 ページの「SPARC プラットフォームのブートのトラブルシューティング(作業マップ)」

シャットダウンとブートの用語

この節では、シャットダウンとブートに関する用語について説明します。

実行レベルと `init` 状態 「実行レベル」とは、システムの状態を表す文字または数字のことで、どのシステムサービスを使用できるのかを示します。システムは常に定義済み実行レベルの 1 つで動作します。実行レベルは `init` プロセスによって維持されるため、実行レベルは「`init` 状態」と呼ばれることもあります。システム管理者は、`init` コマンドまたは `svcadm` コマンドを使用して、実行レベルを変更します。このマニュアルでは、`init` 状態を実行レベルと呼びます。

ブートオプション 「ブートオプション」とは、システムのブート方法を表します。

次のようなブートオプションがあります。

- 対話式ブート - システムのブート方法に関する情報 (カーネルやデバイスのパス名など) を入力するプロンプトが表示されます。
- 再構成ブート - システムが再構成され、新しく追加されたハードウェアや新しい擬似デバイスがサポートされるようになります。
- 回復ブート - システムがハング状態になったとき、無効なエントリがあるためシステムが正常にブートできないとき、またはユーザーがログインできないときに使用します。

GRUB ベースのブート特有の用語については、[324 ページの「x86: GRUB の用語」](#)を参照してください。

システムのシャットダウンに関するガイドライン

システムをシャットダウンするときは、次の点に注意してください。

- システムのシャットダウンには、`init` および `shutdown` コマンドを使用します。これらのコマンドは、すべてのシステムプロセスとサービスを正常に終了させてからシャットダウンします。

x86 のみ – Solaris 10 6/06 以降のリリースが稼働する x86 システムでは、電源ボタンを押して離すことで、システムの正常なシャットダウンを開始できます。この方法で x86 システムをシャットダウンすることは、`init 5` コマンドを使ってシステムをシャットダウンするのと同様です。一部の x86 システムでは、BIOS の設定によって、電源ボタンを押してもシステムのシャットダウンが開始されない可能性があります。電源ボタンを使用するには、BIOS を設定し直します。

- サーバーをシャットダウンする場合は、`shutdown` コマンドを使用します。ログインしているユーザーやサーバーのリソースをマウントしているシステムは、サーバーがシャットダウンされる前に通知を受けます。システムのシャットダウンについては、ユーザーが予定を立てられるようあらかじめ電子メールで知らせておくようにします。
- `shutdown` または `init` コマンドを使用してシステムをシャットダウンするには、スーパーユーザー権限が必要です。
- `shutdown` および `init` コマンドはどちらも実行レベルを引数に指定します。

もっともよく使用される実行レベルは次の3つです。

- **実行レベル 3** – すべてのシステムリソースを使用でき、ユーザーもログインできます。デフォルトでは、システムをブートすると実行レベル 3 になります。通常の運用で使用されます。この実行レベルは、NFS リソースを共有できるマルチユーザーレベルとも呼ばれます。
- **実行レベル 6** – オペレーティングシステムを停止して、`/etc/inittab` ファイルの `initdefault` エントリに定義されている状態でリブートします。
- **実行レベル 0** – オペレーティングシステムがシャットダウンされ、安全に電源が切断されます。システムの設置場所を変更したり、ハードウェアを追加または削除する場合は、システムを実行レベル 0 にする必要があります。

実行レベルの詳細は、[第 17 章「サービスの管理 \(概要\)」](#) を参照してください。

システムのブートに関するガイドライン

システムをブートするときは、次の点に注意してください。

- SPARC システムをシャットダウンしたあとでブートするには、PROM レベルで `boot` コマンドを使用します。
- x86 システムをシャットダウンしたあとでブートするには、GRUB メニューで OS インスタンスを選択します。
- Solaris 9 リリースと一部の Solaris 10 リリースでは、x86 システムをシャットダウンしたあとでブートするには、一次ブートサブシステムメニューで `boot` コマンドを使用します。
- 電源を切断した後には再投入すればシステムをリブートできます。



注意 - このシャットダウン方法をサポートしている Solaris リリースが稼働する x86 システムを使用するのではないかぎり、この方法は正常なシャットダウンとは言えません。196 ページの「[x86: 電源ボタンによるシステムシャットダウン開始のサポート](#)」を参照してください。このシャットダウン方法は、緊急の場合にだけ使用してください。システムサービスやプロセスが突然終了するため、ファイルシステムに損傷を与えることがあります。この種の損傷の修復作業には手間がかかり、さまざまなユーザーファイルやシステムファイルをバックアップコピーから復元する必要がある可能性があります。

- SPARC システムと x86 システムとでは、ブート時に使用するハードウェアが異なります。これらの違いについては、[第 15 章「x86: GRUB ベースのブート \(参照情報\)」](#)を参照してください。

システムをシャットダウンする場合

次の表に、システム管理作業とそれに伴って必要となるシャットダウンの種類を示します。

表 8-1 システムのシャットダウン

システムシャットダウンの理由	適切な実行レベル	詳細
停電のためシステムの電源を切断します	実行レベル 0。安全に電源を切れる状態	第 10 章「システムのシャットダウン (手順)」
<code>/etc/system</code> ファイル内のカーネルパラメータを変更します	実行レベル 6 (システムのリブート)	第 10 章「システムのシャットダウン (手順)」

表 8-1 システムのシャットダウン (続き)

システムシャットダウンの理由	適切な実行レベル	詳細
ファイルシステムを保守します (システムデータのバックアップや復元など)	実行レベル S (シングルユーザーレベル)	第 10 章「システムのシャットダウン (手順)」
/etc/system などのシステム構成ファイルを修復します	200 ページの「システムをブートする場合」を参照してください	なし
システムにハードウェアを追加します (または、システムからハードウェアを削除する)	再構成ブート (ハードウェアを追加または削除したら電源を切断する)	『Solaris のシステム管理 (デバイスとファイルシステム)』の「システムへ周辺デバイスを追加する」
ブート失敗の原因となっていた重要なシステムファイルを修復します	200 ページの「システムをブートする場合」を参照してください	なし
カーネルデバッグ (kmdb) をブートして、システムの障害を調査します	実行レベル 0 (可能な場合)	第 10 章「システムのシャットダウン (手順)」
ハング状態から回復させ、クラッシュダンプを強制します	200 ページの「システムをブートする場合」を参照してください	なし
カーネルデバッグ (kmdb) を使用してシステムをリブートします (実行時にデバッグを読み込めない場合)	実行レベル 6 (システムのリブート)	SPARC システムの場合: 301 ページの「SPARC: カーネルデバッグ (kmdb) を使ってシステムをブートする方法」 x86 システムの場合: 304 ページの「x86: GRUB ブート環境でカーネルデバッグ (kmdb) を使ってシステムをブートする方法」

サーバーまたはスタンドアロンシステムのシャットダウンの例については、[第 10 章「システムのシャットダウン \(手順\)」](#)を参照してください。

システムをブートする場合

次の表に、システム管理作業とそれに伴って必要となるブートオプションを示します。

表 8-2 システムのブート

システムリブートの理由	適切なブートオプション	SPARC システムの参照先	x86 システムの参照先
停電のためシステムの電源を切断します	システムの電源を再投入します	第 10 章「システムのシャットダウン (手順)」	第 10 章「システムのシャットダウン (手順)」
/etc/system ファイル内のカーネルパラメータを変更します	システムを実行レベル 3 でリブートします (NFS リソースを共有できるマルチユーザーレベル)	253 ページの「SPARC: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法」	272 ページの「x86: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法」
ファイルシステムを保守します (システムデータのバックアップや復元など)	実行レベル S で Control + D を押して、システムを実行レベル 3 に戻します	254 ページの「SPARC: システムを実行レベル S (シングルユーザーレベル) でブートする方法」	274 ページの「x86: システムを実行レベル S (シングルユーザーレベル) でブートする方法」
/etc/system などのシステム構成ファイルを修復します	対話式ブート	255 ページの「SPARC: システムを対話式でブートする方法」	277 ページの「x86: システムを対話式でブートする方法」
システムにハードウェアを追加します (または、システムからハードウェアを削除する)	再構成ブート (ハードウェアを追加または削除したら電源を投入する)	『Solaris のシステム管理 (デバイスとファイルシステム)』の「システムディスクまたは二次ディスクの追加 (作業マップ)」	『Solaris のシステム管理 (デバイスとファイルシステム)』の「システムディスクまたは二次ディスクの追加 (作業マップ)」
カーネルデバッグ (kmdb) を使用してシステムをブートし、システムの障害を調査します	kmdb をブートします	301 ページの「SPARC: カーネルデバッグ (kmdb) を使ってシステムをブートする方法」	304 ページの「x86: GRUB ブート環境でカーネルデバッグ (kmdb) を使ってシステムをブートする方法」
フェイルセーフモードでシステムをブートし、システムのブート失敗の原因となっている重要なシステムファイルを修復します	フェイルセーフアーカイブをブートします	265 ページの「SPARC システムでフェイルセーフアーカイブをブートする方法」	284 ページの「GRUB を使用して x86 システムでフェイルセーフアーカイブをブートする方法」
ハング状態から回復させ、クラッシュダンプを強制します。	回復ブート	297 ページの「SPARC: クラッシュダンプを強制してシステムをリブートする方法」	303 ページの「x86: クラッシュダンプを強制してシステムをリブートする方法」

システムのシャットダウンとブート (概要)

この章では、システムのブートについての概要を説明します。Solaris ブートデザイン、ブートプロセス、および Solaris OS でシステムをブートする各種方法について説明します。

この章で説明する情報は次のとおりです。

- 204 ページの「Solaris ブートデザインの基礎」
- 205 ページの「新しい Solaris SPARC ブートアーキテクチャーについて」
- 208 ページの「Solaris SPARC でのブートアーカイブの実装」
- 209 ページの「x86: GRUB ブートローダーの管理」
- 211 ページの「ZFS ルートファイルシステムからの起動」

Solaris システムのブート手順については、第 12 章「Solaris システムのブート (手順)」を参照してください。

GRUB を実装していない Solaris システムのブート手順については、第 16 章「x86: GRUB を実装しないシステムのブート (手順)」を参照してください。

システムのシャットダウンとブートに関する新機能については、193 ページの「システムのシャットダウンとブートに関する新機能」を参照してください。

ブートローダーの管理と Solaris ブート動作の変更に関する概要と手順については、第 11 章「Solaris ブート動作の変更 (手順)」を参照してください。

サービス管理機能 (SMF) によるブートサービスの管理については、374 ページの「SMF とブート」を参照してください。

Solaris ブートデザインの基礎

SPARC と x86 の両方のプラットフォーム向けの Solaris ブートデザインには、次の特徴があります。

- ブートアーカイブの使用

ブートアーカイブとは、システムのブートに必要なすべてのファイルが含まれている RAM ディスクイメージのことです。Solaris OS をインストールすると、一次アーカイブとフェイルセーフアーカイブの 2 つのブートアーカイブが作成されます。詳細については、[208 ページの「Solaris SPARC でのブートアーカイブの実装」](#)を参照してください。

bootadm コマンドも、SPARC プラットフォームでできるように変更されました。このコマンドは、Solaris x86 プラットフォームの場合と同じように機能します。bootadm コマンドは、アーカイブの更新と検証を細部にわたって自動的に処理します。インストールまたはシステムのアップグレード中に、bootadm コマンドによって初期ブートアーカイブが作成されます。正常なシステムシャットダウンの処理中に、シャットダウンプロセスで、ブートアーカイブの内容がルートファイルシステムと突き合わせてチェックされます。何らかの不一致が検出されると、システムはブートアーカイブを再構築して、リブート時に必ずブートアーカイブとルート (/) ファイルシステムの同期がとれるようにします。bootadm コマンドを使用してブートアーカイブを手動で更新することもできます。[313 ページの「bootadm コマンドを使用してブートアーカイブを管理する」](#)を参照してください。

注 - SPARC システムでは、bootadm コマンドの一部のオプションを使用できません。

詳細は、[bootadm\(1M\)](#) および [boot\(1M\)](#) のマニュアルページを参照してください。

- インストールおよびフェイルセーフ処理中に **RAM** ディスクイメージをルートファイルシステムとして使用

このプロセスは、Solaris SPARC および Solaris x86 の両方のプラットフォームで同じになりました。RAM ディスクイメージは、ブートアーカイブから生成されたあと、ブートデバイスからシステムに転送されます。

注 - SPARC プラットフォームでは、OpenBoot™ PROM を引き続き使用して、ブートデバイスにアクセスし、ブートアーカイブをシステムのメモリーに転送します。反対に、x86 プラットフォームでは、システムは最初に BIOS によって制御されます。BIOS を使用して、ネットワークデバイスからのブートアーカイブの転送を開始するか、ブートローダーを実行します。Solaris OS では、ディスクからのブートアーカイブの転送に使用する x86 ブートローダーは GRUB です。323 ページの「x86: ブートプロセス」を参照してください。

ソフトウェアをインストールする場合、RAM ディスクイメージは、インストールプロセス全体に使用されるルートファイルシステムになります。この目的で RAM ディスクイメージを使用すると、リムーバブルメディアからシステムをブートする必要がなくなります。RAM ディスクファイルシステムのタイプには、HSFS (High Sierra File System) または UFS を指定できます。

新しい Solaris SPARC ブートアーキテクチャーについて

Solaris SPARC プラットフォームのブートプロセスは、Solaris x86 ブート動作との共通性を高めるために再設計および改善されました。新しい Solaris SPARC ブートデザインでは、ブートチェーンの複数の部分に変更を加えなくても、新しいファイルシステムタイプなどの新機能を追加できます。変更には、ブートフェーズの独立性の実装も含まれます。

これらの改善の重要点は、次のとおりです。

- SPARC および Solaris x86 プラットフォームでのブートプロセスの共通性
- ネットワークブート動作の共通性
- さまざまなファイルシステムタイプからシステムをより簡単にブートできるようにする、ブートアーキテクチャーの柔軟性

次の4つのブートフェーズは、互いに独立しています。

1. OBP (Open Boot PROM) フェーズ

Solaris SPARC プラットフォームでのブートプロセスの OBP フェーズは変わっていません。

ディスクデバイスの場合、ファームウェアドライバは通常 OBP ラベルパッケージの「読み込み」メソッドを使用します。このメソッドは、ディスクの先頭にある VTOC ラベルを解析して、指定されたパーティションを見つけます。そして、パーティションのセクター 1-15 がシステムのメモリーに読み込まれます。この領域は一般にブートブロックと呼ばれ、通常はファイルシステムリーダーが格納されています。

2. ブーターフェーズ

このフェーズの間に、ブートアーカイブが読み取られ、実行されます。ブートファイルシステムの形式を認識しておく必要があるのは、ブートプロセスのこのフェーズだけです。場合によっては、ブートアーカイブがインストールミニルートでもあることがあります。ブートローダーとブートアーカイブの転送に使用されるプロトコルには、ローカルディスクアクセス、NFS、および HTTP があります。

3. RAM ディスクフェーズ

RAM ディスクは、カーネルモジュールまたはインストールミニルートから成るブートアーカイブです。

Solaris SPARC ブートアーカイブは、Solaris x86 ブートアーカイブと同じです。ブートアーカイブのファイルシステム形式は非公開です。そのため、システムのブート時に使用されるファイルシステムタイプ (HSFS または UFS ファイルシステムなど) の情報は、ブーターやカーネルには必要ありません。RAM ディスクはブートアーカイブからカーネルイメージを抽出して、それを実行します。RAM ディスク、特にシステムのメモリーに常駐するインストールミニルートのサイズを最小化するため、ミニルートの内容が圧縮されます。この圧縮はファイルレベルごとに行われ、個々のファイルシステム内で実装されます。ファイルを圧縮して、ファイルに圧縮済みのマークを付けるには、`/usr/sbin/fiocompress` ユーティリティが使用されます。

注 - このユーティリティには、ファイル圧縮ファイルシステム `dcfs` との非公開インタフェースが備わっています。

4. カーネルフェーズ

カーネルフェーズは、ブートプロセスの最終段階です。このフェーズの間に、Solaris OS が初期化され、ブートアーカイブから構築された RAM ディスク上に最小ルートファイルシステムがマウントされます。ブートアーカイブがインストールミニルートである場合、OS は引き続きインストールプロセスを実行します。それ以外の場合は、指定されたルートデバイス上にルートファイルシステムをマウントするために十分な 1 組のカーネルファイルおよびドライバが RAM ディスクに格納されます。

そして、カーネルはブートアーカイブから主モジュールの残りを抽出し、それ自体を初期化し、実際のルートファイルシステムをマウントして、ブートアーカイブを破棄します。

ミニルートの圧縮と展開

RAM ディスクベースのミニルートは、`root_archive` コマンドによって圧縮および展開されます。新しいブートアーキテクチャーをサポートする SPARC システムだけがミニルートの圧縮版を圧縮および展開できます。



注意 - Solaris 10 バージョンの `root_archive` ツールは、ほかの Solaris リリースに含まれているツールのバージョンと互換性がありません。そのため、RAM ディスクの操作はアーカイブと同じ Solaris リリースが動作するシステムでのみ行うようにしてください。

ミニルートの圧縮および展開の詳細については、`root_archive(1M)` のマニュアルページを参照してください。

ソフトウェアのインストールとアップグレード

Solaris OS をインストールまたはアップグレードするには、CD/DVD またはネットワークからミニルートをブートする必要があります。どちらの場合も、ミニルートのルートファイルシステムは RAM ディスクです。このプロセスを使用すると、システムをリブートしなくても Solaris ブート CD を取り出すことができます。ブートアーカイブにはミニルート全体が含まれています。インストール CD の構築は、HSFS ブートブロックを使用するように変更されました。ミニルートは、RAM ディスクとして読み込まれる単一の UFS ファイルに圧縮されます。ミニルートはすべての OS インストールタイプに使用されます。

インストールのメモリー要件

Solaris 10 リリースでは、システムをインストールする際の最小メモリー要件が 256M バイトから 384M バイトに増やされました。このメモリー容量では、テキストベースのインストールだけが可能になります。インストール GUI プログラムを実行するには、768M バイト以上のメモリーが必要です。

ネットワークブートサーバーの設定プロセスの変更点

ネットワークブートサーバーの設定プロセスが変更されました。ブートサーバーは、ブートストラッププログラムだけでなく、RAM ディスクにも対応するようになりました。RAM ディスクは、CD/DVD からブートしようと、NFS または HTTP を使ってネットワークインストールを実行しようと、すべてのインストールの単一のミニルートとしてダウンロードされ、ブートされます。NFS または `wanboot` プログラム (HTTP) によるネットワークブート用のネットワークブートサーバーの管理は何も変わりません。ただし、ネットワークブートプロセスの内部実装は、次のように変更されました。

1. ブートサーバーはブートアーカイブの形でブートストラップをターゲットシステムに転送する。

2. ターゲットシステムはブートアーカイブを RAM ディスクに展開する。
3. ブートアーカイブは読み取り専用の初期ルートデバイスとしてマウントされる。

SPARC システムのブートについては、[251 ページ](#)の「[SPARC システムのブート \(作業マップ\)](#)」を参照してください。

複数の Solaris カーネルのブートのサポート

SPARC システムでは、`ok` プロンプトで `boot` と入力すると、デフォルトのブートデバイスが自動的に選択されます。代替ブートデバイスを指定するには、`boot-device` NVRAM 変数を変更します。ブート時にコマンド行から代替ブートデバイスまたは代替カーネル (ブートファイル) を指定することもできます。[257 ページ](#)の「[SPARC: デフォルトのカーネル以外の Solaris カーネルをブートする方法](#)」を参照してください。

Solaris SPARC でのブートアーカイブの実装

以前は x86 プラットフォームでしか使用できなかった Solaris ブートアーカイブが Solaris SPARC ブートアーキテクチャーに不可欠な要素となりました。

`bootadm` コマンドは、Solaris SPARC プラットフォームでも使用できるように変更されました。このコマンドは、Solaris x86 プラットフォームの場合と同じように機能します。`bootadm` コマンドは、アーカイブの更新と検証を細部にわたって処理します。x86 プラットフォームでは、インストールまたはシステムのアップグレード中に、`bootadm` コマンドによって GRUB メニューが更新されます。ブートアーカイブを手動で管理する場合にも、`bootadm` コマンドを使用できます。

`boot-archive` サービスは、サービス管理機能 (SMF) によって管理されます。ブートアーカイブのサービスインスタンスは `svc:/system/boot-archive:default` です。このサービスを有効、無効、または更新するには、`svcadm` コマンドを使用します。SMF を使用したサービスの管理については、[第 17 章「サービスの管理 \(概要\)」](#)を参照してください。

サポートされている Solaris リリースでは、SPARC と x86 の両方のシステムに 2 種類のブートアーカイブが用意されています。

- 一次ブートアーカイブ
- フェイルセーフブートアーカイブ

Solaris SPARC ブートアーカイブに含まれるファイルは、`/platform` ディレクトリにあります。

/platformディレクトリの内容は2つのグループのファイルに分けられます。

- sun4u ブートアーカイブに必要なファイル
- sun4v ブートアーカイブに必要なファイル

ブートアーカイブの管理については、307 ページの「Solaris ブートアーカイブの管理 (作業マップ)」を参照してください。

x86: GRUB ブートローダーの管理

オープンソースの GRUB (GRand Unified Bootloader) は、x86 システムのデフォルトのブートローダーです。GRUB は、ブートアーカイブをシステムのメモリーに読み込むソフトウェアです。ブートアーカイブは、システムの起動中にルートファイルシステムがマウントされる前に必要な、重要なファイルの集まりです。ブートアーカイブとは、Solaris OS のブートに使用されるインタフェースのことです。GRUB の詳細は、<http://www.gnu.org/software/grub/grub.html> を参照してください。grub(5) のマニュアルページも参照してください。

GRUB ベースのブートの動作

x86 システムの電源を入れると、BIOS (Basic Input/Output System) により、CPU、メモリー、およびプラットフォームハードウェアが初期化されます。初期化フェーズが完了すると、BIOS が構成済みブートデバイスからブートローダーをロードし、システムの制御をブートローダーに移します。「ブートローダー」は、システムの電源を投入したあとに最初に動作するソフトウェアプログラムです。このプログラムによってブートプロセスが開始されます。

GRUB は、menu.lst ファイルと呼ばれる構成ファイルに定義済みのブートオプションが含まれているメニューインタフェースを実装します。また、デフォルトのブート動作の変更など、さまざまなブート機能の実行に使用できる GUI メニューインタフェースからアクセス可能なコマンド行インタフェースも備えています。Solaris OS では、GRUB の実装はマルチブート仕様に準拠しています。マルチブート仕様の詳細は、<http://www.gnu.org/software/grub/grub.html> を参照してください。

Solaris カーネルはマルチブート仕様に完全に準拠しているため、GRUB を使用して x86 システムをブートできます。GRUB を使用すると、単一の x86 システムにインストールされている各種オペレーティングシステムをブートできます。たとえば、ブート時に GRUB メニューでブートエントリを選択するか、デフォルトで特定の OS をブートするように menu.lst ファイルを設定することで、Solaris OS、Linux、または Windows を個別にブートできます。

GRUB では、ファイルシステムとカーネル実行可能ファイルの形式が認識されるため、ディスク上のカーネルの物理的な位置を記録しておかなくても、オペレーティングシステムを読み込むことができます。GRUB ベースのブートで

は、カーネルは、カーネルのファイル名、およびカーネルが存在するドライブとパーティションを指定することでロードされます。詳細は、[327 ページの「GRUB の構成に使用される命名規則」](#)を参照してください。

GRUB を使ってシステムをブートする手順については、[271 ページの「GRUB を使用して x86 システムをブートする \(作業マップ\)」](#)を参照してください。

また、次のマニュアルページも参照してください。

- [boot\(1M\)](#)
- [bootadm\(1M\)](#)
- [grub\(5\)](#)
- [installgrub\(1M\)](#)

新しい findroot コマンドの **GRUB** サポート

この Solaris リリースでは、新しい findroot コマンドの GRUB サポートが実装されました。findroot コマンドは、以前に GRUB が使用していた root コマンドと同様に機能しますが、ブートデバイスに関係なく、目的のディスクを発見する機能が強化されました。findroot コマンドは、ZFS ルートファイルシステムからのブートもサポートします。

このコマンドの menu.lst エントリのもっとも一般的な形式は次のとおりです。

```
title Solaris 10 10/08 s10x_u6wos_03 X86
findroot (pool_rpool,0,a)
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive

title Solaris failsafe
findroot (pool_rpool,0,a)
kernel /boot/multiboot kernel/unix -s      -B console=ttyb
module /boot/x86.miniroot-safe
```

詳細は、[246 ページの「x86: findroot コマンドの実装」](#)を参照してください。

GRUB の参照情報については、[第 15 章「x86: GRUB ベースのブート \(参照情報\)」](#)を参照してください。

ZFS ルートファイルシステムからの起動

ZFS ルートファイルシステムからのブートのサポートが Solaris OS に追加されました。Solaris インストールソフトウェアには、ZFS ルートを含むシステムのアップグレードやパッチ適用のサポートも含まれています。ブート方法、システム操作、およびインストール手順は、この変更をサポートするように修正されました。ブートに関する変更には、SPARC プラットフォームでの新しいブートアーキテクチャーの実装があります。新しい SPARC ブート設計では、Solaris x86 ブートアーキテクチャーとの共通性を高める機能強化が行われています。

この機能を使用する前に、『[Solaris 10 ご使用にあたって](#)』で既知の問題に関する情報を確認してください。

用語の完全なリストなど、ZFS の詳細は、『[Solaris ZFS 管理ガイド](#)』の「ZFS の用語」を参照してください。

ZFS の Solaris インストール要件

Solaris ソフトウェアの新規インストールを実行する前、または Solaris Live Upgrade を使って UFS ルートファイルシステムを ZFS ルートファイルシステムに移行する前に、次の要件が満たされていることを確認してください。

- **Solaris** リリース情報

ZFS ルートファイルシステムからインストールやブートを行う機能は、Solaris 10 10/09 リリースで使用できます。Solaris Live Upgrade 処理を実行して ZFS ルートファイルシステムに移行するには、Solaris 10 10/09 リリースをインストールしておくか、そのリリースにアップグレードしておく必要があります。

- **ZFS** ストレージプールの容量要件

ZFS ルート環境ではスワップデバイスとダンプデバイスが共有されないため、ブート可能な ZFS ルートファイルシステムに必要な空きプール領域の最小容量は、ブート可能な UFS ルートファイルシステムの場合よりも大きくなります。スワップボリュームのサイズは物理メモリーのサイズの 1/2 (ただし 2G バイト以下かつ 512M バイト以上) として計算されます。ダンプボリュームのサイズは、`dumpadm` の情報と物理メモリーのサイズに基づいて、カーネルによって計算されます。新しいサイズがシステムの運用をサポートする限り、JumpStart プロファイルで、または最初のインストール時に、スワップボリュームとダンプボリュームのサイズを好きなサイズに調整できます詳細は、『[Solaris ZFS 管理ガイド](#)』の「[スワップデバイスおよびダンプデバイスの ZFS サポート](#)」を参照してください。

。

ZFS ルートファイルシステムからのブートの動作

ZFS ルートファイルシステムからのブート動作は、UFS ファイルシステムからブートする場合とは異なります。ZFS はインストールやブートに新しい概念をいくつか適用するため、システムのブートに関する基本的な管理業務がいくつか変更されました。ZFS ルートファイルシステムからのブートと UFS ルートファイルシステムからのブートのもっとも重要な違いは、ZFS からのブートでは、デバイス識別子がルートファイルシステムを一意に特定しないことです。これはつまり、ブート環境 (BE) であると言えます。ZFS を使用した場合は、デバイス識別子によって「ストレージプール」が一意に識別されます。ストレージプールには、複数のブート可能データセット (ルートファイルシステム) が含まれていることがあります。そのため、ブートデバイスを指定するほかに、ブートデバイスによって識別されたプール内のルートファイルシステムも指定する必要があります。

x86 システムでは、GRUB によって識別されるブートデバイスに ZFS ストレージプールが含まれている場合、GRUB メニューの作成に使われる `menu.lst` ファイルがそのプールのデータセット階層のルートにあるデータセットに格納されています。このデータセットには、プールと同じ名前が付いています。各プールにそのようなデータセットが 1 つあります。

「デフォルトのブート可能データセット」は、ブート時にマウントされ、ルートプールの `bootfs` プロパティによって定義される、プールのブート可能データセットです。ルートプール内のデバイスがブートすると、このプロパティによって指定されるデータセットがルートファイルシステムとしてマウントされます。

新しい `bootfs` プールプロパティは、指定されたプールのデフォルトのブート可能データセットを指定するためにシステムが使用するメカニズムです。ルートプール内のデバイスが起動すると、デフォルトでルートファイルシステムとしてマウントされるデータセットは `bootfs` プールプロパティによって識別されるデータセットになります。

SPARC システムでは、`boot` コマンドの新しい `-Z dataset` オプションを使用することで、デフォルトの `bootfs` プールプロパティが上書きされます。

x86 システムでは、ブート時に GRUB メニューで代替ブート環境を選択することで、デフォルトの `bootfs` プールプロパティが上書きされます。

SPARC: ZFS ルートファイルシステムからのブートをサポートするブートオプション

SPARC プラットフォームでは、次の 2 つのブートオプションが新たに追加されました。

- `-L` オプション。システム上の使用可能なすべての BE のリストを表示します。

```
ok boot -L
```

注 -L オプションは、ok プロンプトから実行されます。このオプションは、システム上の使用可能な BE のリストを表示するだけです。システムをブートするには、-Z ブートオプションを使用します。

- -Z オプション。boot コマンドでこのオプションを使用すると、bootfs プロパティによって指定されるデフォルトのデータセット以外のブート可能データセットを指定できます。

```
ok boot -Z dataset
```

ZFS ブートローダーを備えたデバイスで -L オプションを使用したときに表示される BE のリストは、当該システムで使用可能な menu.lst エントリを反映しています。使用可能な BE のリストのほかに、BE を選択する方法や -Z オプションを使ってシステムをブートする方法も表示されます。メニュー項目の bootfs 値によって指定されたデータセットは、ブーターによって読み込まれる後続のすべてのファイル (/etc ディレクトリに格納されているブートアーカイブや各種の構成ファイルなど) に使用されます。その後、このデータセットはルートファイルシステムとしてマウントされます。

手順については、[259 ページの「SPARC システムで ZFS ルートファイルシステムからブートする」](#)を参照してください。

x86: ZFS ルートファイルシステムからのブートをサポートするブートオプション

x86 プラットフォームでは、新しい GRUB キーワード \$ZFS-BOOTFS が導入されました。x86 システムのブート時に、GRUB メニューエントリに対応するルートファイルシステムが ZFS データセットである場合は、GRUB メニューエントリに、デフォルトで \$ZFS-BOOTFS トークンを持つ -B オプションが含まれます。ZFS ブートローダーをサポートする Solaris リリースをシステムにインストールするか、そのリリースにアップグレードする場合は、GRUB menu.lst ファイルがこの情報で自動的に更新されます。デフォルトのブート可能データセットは、bootfs プロパティによって識別されます。

ZFS ブートローダーをサポートする Solaris リリースが稼働する x86 システムでは、この情報は GRUB メニューに組み込まれます。

ZFS ブートローダーをサポートする GRUB 実装のデフォルトの menu.lst ファイルの例を次に示します。

```
title Solaris 10 5/08 s10x_nbu6wos_nightly X86
findroot (pool_rpool,0,a)
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive

title Solaris failsafe
findroot (pool_rpool,0,a)
kernel /boot/multiboot kernel/unix -s      -B console=ttyb
module /boot/x86.miniroot-safe
```

ZFS からシステムをブートする手順については、[279 ページ](#)の「[x86 システムで ZFS ルートファイルシステムから起動する](#)」を参照してください。

システムのシャットダウン(手順)

この章では、システムのシャットダウン手順について説明します。この章で説明する手順は次のとおりです。

この章の内容は以下のとおりです。

- [216 ページの「システムシャットダウンコマンド」](#)
- [217 ページの「システムのダウン時間をユーザーに通知する」](#)
- [224 ページの「すべてのデバイスの電源を落とす」](#)

システムの実行レベルの概要については、[第 17 章「サービスの管理\(概要\)」](#)を参照してください。

実行レベルとブートファイルに関連した手順については、[215 ページの「システムのシャットダウン\(作業マップ\)」](#)を参照してください。

システムのシャットダウン(作業マップ)

作業	説明	参照先
システムにログインしているユーザーを確認します。	who コマンドを使用して、システムにログインしているユーザーを確認します。	218 ページの「システムにログインしているユーザーを知る方法」
サーバーをシャットダウンします。	適切なオプションで shutdown コマンドを使用して、サーバーをシャットダウンします。	218 ページの「サーバーをシャットダウンする方法」
スタンドアロンシステムをシャットダウンします。	init コマンドを使用し、適切な実行レベルを指定して、スタンドアロンシステムをシャットダウンします。	222 ページの「スタンドアロンシステムをシャットダウンする方法」

作業	説明	参照先
すべてのデバイスの電源を落とします。	システムの電源を落とすデバイスは次のとおりです。 ■ CPU ■ モニター ■ 外部デバイス(ディスク、テープ、プリンタなど)	224 ページの「すべてのデバイスの電源を落とす方法」

システムのシャットダウン

Solaris ソフトウェアは、電子メールやネットワークソフトウェアをいつでも利用できるように、停止することなく動作するように設計されています。しかし、システム管理作業を行う場合や緊急事態が発生した場合は、システムをシャットダウンして安全に電源を切断できる状態にする必要があります。場合によっては、システムを一部のシステムサービスしか利用できない中間の実行レベルまで移行する必要があります。

次のような場合がそれに該当します。

- ハードウェアを追加または削除する
- 予定された停電に備える
- ファイルシステムの保守を行う (バックアップなど)

システムのシャットダウンを必要とするシステム管理作業の詳細については、[第 9 章「システムのシャットダウンとブート \(概要\)」](#)を参照してください。

システムの電源管理機能を使用する方法については、[pmconfig\(1M\)](#) のマニュアルページを参照してください。

システムシャットダウンコマンド

システムをシャットダウンするもっとも基本的な方法は、`init` と `shutdown` の各コマンドを使用する方法です。どちらのコマンドも、システムを「正常な状態でシャットダウンする」ため、すべてのファイルシステムに対する変更がディスクに書き出され、すべてのシステムサービス、プロセス、オペレーティングシステムが正常に終了します。

システムのアボートキーシーケンスを使用したり、電源をオフにしてからオンにする方法では、システムサービスが突然終了してしまうので、正常なシャットダウン方法とはいえません。しかし、緊急時には、これらの方法が必要となる場合があります。システムの復旧方法については、[第 12 章「Solaris システムのブート \(手順\)」](#) および [第 14 章「Solaris ブートアーカイブの管理 \(手順\)」](#)を参照してください。

注 - Solaris 10 6/06 以降のリリースが稼働する x86 システムでは、電源ボタンを押して離すとシステムの正常なシャットダウンが開始されます。この方法は、`init 5` コマンドを使用するのと同様です。

次の表に、各種シャットダウンコマンドとその用途を要約します。

表 10-1 シャットダウンコマンド

コマンド	説明	用途
shutdown	<code>init</code> プログラムを呼び出してシステムをシャットダウンする実行可能なシェルスクリプト。デフォルトでは、システムは実行レベル 5 に移行します。	実行レベル 3 で動作しているサーバーで使用します。サーバーにログインしているユーザーに、サーバーが間もなくシャットダウンされることが通知されます。シャットダウンされるサーバーのリソースをマウントしているシステムにも通知されます。
init	すべてのアクティブなプロセスを終了し、ディスクを同期させてから実行レベルを変更する実行可能ファイル。	ほかのユーザーが影響を受けないスタンダオンシステムで使用します。ユーザーはまもなく行われるシャットダウンについて通知されないの、シャットダウンにかかる時間が短くて済みます。
reboot	ディスクを同期させ、ブート命令を <code>uadmin</code> システムコールに渡す実行可能ファイル。このシステムコールによってプロセッサが停止します。	<code>init</code> コマンドを使用するのが望ましい方法です。
halt、poweroff	ディスクを同期させ、プロセッサを停止する実行可能ファイル。	すべてのプロセスを正常にシャットダウンしないで、残りのファイルシステムをすべてマウント解除するため、推奨されていません。正常なシャットダウンを行わずにサービスを停止する操作は、緊急時またはほとんどのサービスがすでに停止している場合に限り行うべきです。

システムのダウン時間をユーザーに通知する

`shutdown` コマンドは起動時に、ログインしているすべてのユーザーと、システムリソースをマウントしているすべてのシステムに、警告と最終メッセージという形でシャットダウンを通知します。

サーバーをシャットダウンする必要がある場合に、`init` コマンドではなく `shutdown` コマンドを使用することを推奨するのはこのためです。どちらを使用するにしても、ユーザーには予定されているシャットダウンについてあらかじめ電子メールで知らせるようにしてください。

システム上のどのユーザーに通知する必要があるかを確認するには、`who` コマンドを使用します。このコマンドは、システムの現在の実行レベルを調べる場合にも役立ちます。詳細は、[376 ページの「システムの実行レベルを確認する」](#) と、[who\(1\)](#) のマニュアルページを参照してください。

▼ システムにログインしているユーザーを知る方法

- 1 シャットダウンするシステムにログインします。
- 2 システムにログインしているすべてのユーザーを表示します。

```
$ who
```

例 10-1 システムにログインしているユーザーを知る

次の例は、システムにログインしているユーザーを表示する方法を示しています。

```
$ who
holly      console      May  7 07:30
kryten     pts/0         May  7 07:35  (starlite)
lister     pts/1         May  7 07:40  (bluemidget)
```

- 1 列目のデータはログインしているユーザーのユーザー名を示します
- 2 列目のデータはログインしているユーザーの端末回線を示します
- 3 列目のデータはユーザーがログインした日時を示します
- 4 列目のデータ (省略可能) は、ユーザーがリモートシステムからログインしている場合のホスト名を示します

▼ サーバーをシャットダウンする方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。
- 2 システムにユーザーがログインしているか調べます。

```
# who
```

ログインしているすべてのユーザーがリスト表示されます。システムがシャットダウンされることを、メールかブロードキャストメッセージで知らせることをお勧めします。

3 システムをシャットダウンします。

shutdown -iinit-level -ggrace-period -y

-iinit-level システムをデフォルトのS以外のinitレベルにします。0、1、2、5、6のいずれかを指定できます。

実行レベル0および5は、システムのシャットダウン用に予約された状態です。実行レベル6はシステムをリブートします。実行レベル2はマルチユーザーオペレーティング状態として使用できます。

-ggrace-period システムがシャットダウンするまでの時間(秒)を示します。デフォルト値は60秒です。

-y ユーザーの介入なしにシャットダウンを継続します。このオプションを指定しないと、シャットダウンを継続するかどうか60秒後にたずねられます。

詳細は、[shutdown\(1M\)](#)のマニュアルページを参照してください。

4 シャットダウンを継続するかどうかたずねられたら、yを入力します。

Do you want to continue? (y or n): **y**

-y オプションを指定した場合、このプロンプトは表示されません。

5 プロンプトが表示されたら、スーパーユーザーのパスワードを入力します。

Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance): **xxxxxx**

6 システム管理作業を終了したら、**Control+D**を押してデフォルトの実行レベルに戻ります。

7 次の表を使用して、システムがshutdownコマンドで指定した実行レベルに移行したことを確認します。

指定した実行レベル	SPARC システムのプロンプト	x86 システムのプロンプト
S (シングルユーザーレベル)	#	#
0 (電源切断レベル)	ok または>	Press any key to reboot

指定した実行レベル	SPARC システムのプロンプト	x86 システムのプロンプト
実行レベル 3 (リモートリソースを共有できるマルチユーザーレベル)	<i>hostname console login:</i>	<i>hostname console login:</i>

例 10-2 SPARC: サーバーを実行レベル S にする

次の例では、shutdown コマンドを使用して、3 分後に、SPARC システムを実行レベル S (シングルユーザーレベル) にしています。

```
# who
root  console      Jun 14 15:49      (:0)

# shutdown -g180 -y

Shutdown started.      Mon Jun 14 15:46:16 MDT 2004

Broadcast Message from root (pts/4) on venus Mon Jun 14 15:46:16...
The system venus will be shut down in 3 minutes .
.
.
Broadcast Message from root (pts/4) on venus Mon Jun 14 15:46:16...
The system venus will be shut down in 30 seconds .
.
.
INIT: New run level: S
The system is coming down for administration.  Please wait.
Unmounting remote filesystems: /vol nfs done.
Shutting down Solaris Management Console server on port 898.
Print services stopped.
Jun 14 15:49:00 venus syslogd: going down on signal 15
Killing user processes: done.

Requesting System Maintenance Mode
SINGLE USER MODE

Root password for system maintenance (control-d to bypass): xxxxxx
single-user privilege assigned to /dev/console.
Entering System Maintenance Mode
#
```

例 10-3 SPARC: サーバーを実行レベル 0 にする

次の例では、shutdown コマンドを使用して、SPARC システムを 5 分後に実行レベル 0 にしています。確認用プロンプトが表示されないように -y オプションを指定しています。

```
# who
root      console      Jun 17 12:39
userabc   pts/4             Jun 17 12:39  (:0.0)
# shutdown -i0 -g300 -y
Shutdown started.    Thu Jun 17 12:40:25 MST 2004

Broadcast Message from root (console) on pretend Thu Jun 17 12:40:25...
The system pretend will be shut down in 5 minutes
.
.
.
Changing to init state 0 - please wait
#
INIT: New run level: 0
The system is coming down. Please wait.
System services are now being stopped.
.
.
.
The system is down.
syncing file systems... done
Program terminated
Type help for more information
ok
```

システムを実行レベル0にしてすべてのデバイスの電源を落とす場合は、[224 ページ](#)の「すべてのデバイスの電源を落とす方法」を参照してください。

例 10-4 SPARC: サーバーをリブートして実行レベル3にする

次の例では、shutdown コマンドを使用して SPARC システムをリブートし、2分後に実行レベル3にしています。確認用プロンプトが表示されないように y オプションを指定しています。

```
# who
root      console      Jun 14 15:49  (:0)
userabc   pts/4             Jun 14 15:46  (:0.0)
# shutdown -i6 -g120 -y
Shutdown started.    Mon Jun 14 15:46:16 MDT 2004

Broadcast Message from root (pts/4) on venus Mon Jun 14 15:46:16...
The system venus will be shut down in 2 minutes

Changing to init state 6 - please wait
#
INIT: New run level: 6
The system is coming down. Please wait.
```

```
.
.
.
The system is down.
syncing file systems... done
rebooting...
.
.
.
venus console login:
```

参照 システムをシャットダウンした理由にかかわらず、すべてのファイルリソースが利用可能かつユーザーがログイン可能な、実行レベル3にシステムが戻ることを想定しているでしょう。システムをマルチユーザーレベルに戻す手順については、[第12章「Solaris システムのブート\(手順\)」](#)を参照してください。

▼ スタンドアロンシステムをシャットダウンする方法

スタンドアロンシステムをシャットダウンする必要がある場合は、次の手順を実行します。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理\(セキュリティサービス\)](#)』の「[RBACの構成\(作業マップ\)](#)」を参照してください。

- 2 システムをシャットダウンします。

```
# init 5
```

詳細は、[init\(1M\)](#)のマニュアルページを参照してください。

- 別の方法として、uadmin コマンドを使ってシステムをシャットダウンすることもできます。

```
# uadmin 2 0
```

- **Solaris 10 6/06**以降のリリースが稼働する **x86** システムを使用する場合、電源ボタンを押して離すことで、システムの正常なシャットダウンの開始とシステムの電源切断を行えます。この機能は、init 5 コマンドを使ってシステムをシャットダウンするのと同様です。詳細については、[193 ページの「システムのシャットダウンとブートに関する新機能」](#)を参照してください。

- 3 次の表を使用して、システムが init コマンドで指定した実行レベルに移行したことを確認します。

指定した実行レベル	SPARC システムのプロンプト	x86 システムのプロンプト
S (シングルユーザーレベル)	#	#
2 (マルチユーザーレベル)	#	#
0 (電源切断レベル)	ok または >	Press any key to reboot
3 (NFS リソースを共有できるマルチユーザーレベル)	hostname console login:	hostname console login:

例 10-5 uadmin コマンドを使ってシステムをシャットダウンする

```
# uadmin 2 0
syncing file systems... done
Program terminated
```

例 10-6 スタンドアロンシステムを実行レベル 0 にする

次の例では、init コマンドを使用して、x86 スタンドアロンシステムを、電源を安全に落とせるレベルにします。

```
# init 0
#
INIT: New run level: 0
The system is coming down. Please wait.
.
.
.
The system is down.
syncing file systems... [11] [10] [3] done
Press any key to reboot
```

システムを実行レベル 0 にしてすべてのデバイスの電源を落とす場合は、[224 ページの「すべてのデバイスの電源を落とす方法」](#)を参照してください。

例 10-7 SPARC: スタンドアロンシステムを実行レベル S にする

次の例では、init コマンドを使用して、SPARC スタンドアロンシステムを実行レベル S (シングルユーザーレベル) にしています。

```
# init s
#
INIT: New run level: S
The system is coming down for administration. Please wait.
Unmounting remote filesystems: /vol nfs done.
```

```
Print services stopped.
syslogd: going down on signal 15
Killing user processes: done.

SINGLE USER MODE

Root password for system maintenance (control-d to bypass): xxxxxx
single-user privilege assigned to /dev/console.
Entering System Maintenance Mode
#
```

参照 システムをシャットダウンした理由にかかわらず、すべてのファイルリソースが利用可能かつユーザーがログイン可能な、実行レベル3にシステムが戻ることを想定しているでしょう。システムをマルチユーザーレベルに戻す手順については、[第12章「Solaris システムのブート\(手順\)」](#)を参照してください。

すべてのデバイスの電源を落とす

次のような場合は、すべてのシステムデバイスの電源を落とす必要があります。

- ハードウェアを置換または追加する。
- システムの設置場所を変更する。
- 予定された停電や自然災害(接近中の雷雨など)に備える。

電源を落とすシステムデバイスには、CPU、モニター、外部デバイス(ディスク、テープ、プリンタ)などがあります。

すべてのシステムデバイスの電源を落とす前に、前の節の説明に従って、システムを正常にシャットダウンする必要があります。

▼ すべてのデバイスの電源を落とす方法

- 1 次の方法のどちらかを選択して、システムをシャットダウンします。
 - サーバーをシャットダウンする場合は、[218 ページの「サーバーをシャットダウンする方法」](#)を参照してください。
 - スタンドアロンシステムをシャットダウンする場合は、[222 ページの「スタンドアロンシステムをシャットダウンする方法」](#)を参照してください。
- 2 システムをシャットダウンしたら、すべてのデバイスの電源を落とします。必要な場合は、電源ケーブルをコンセントから引き抜きます。

- 3 電源が回復したら、次の手順に従ってシステムとデバイスの電源を投入します。
 - a. 電源ケーブルをコンセントに差し込みます。
 - b. モニターの電源を入れます。
 - c. ディスクドライブ、テープドライブ、プリンタの電源を入れます。
 - d. CPUの電源を入れます。
システムが実行レベル3になります。

Solaris ブート動作の変更 (手順)

この章では、Solaris システムのブート動作の変更方法について説明します。

この章の内容は以下のとおりです。

- [227 ページの「SPARC システムでのブート動作の変更 \(作業マップ\)」](#)
- [235 ページの「x86 システムでの Solaris ブート動作の変更 \(作業マップ\)」](#)

ブートに関する新機能とブートプロセスの概要については、[第 8 章「システムのシャットダウンとブートの概要」](#)を参照してください。

Solaris システムのブート手順については、[第 12 章「Solaris システムのブート \(手順\)」](#)を参照してください。

SPARC システムでのブート動作の変更 (作業マップ)

作業	説明	参照先
PROM のリビジョン番号を確認します。	ok プロンプトで banner コマンドを使用して、システムの PROM リビジョン番号を表示します。	229 ページの「SPARC: システムの PROM リビジョン番号を確認する方法」
ブートできるシステム上のデバイスを確認します。	ブート PROM を使用してブート動作を変更する前に、システム上のデバイスを確認します。	229 ページの「SPARC: システム上のデバイスを確認する方法」
現在のブートデバイスを表示します。	この手順を使用して、システムのブート元となる現在のデフォルトのブートデバイスを調べます。	231 ページの「SPARC: デフォルトのブートデバイスを調べる方法」

作業	説明	参照先
デフォルトのブートデバイスを変更します。	デフォルトのブートデバイスを変更するには、次のいずれかの方法を使用します。 ■ ブート PROM で <code>boot-device</code> パラメータを変更する。 ■ <code>eeprom</code> コマンドを使用して <code>boot-device</code> パラメータを変更する。	231 ページの「SPARC: ブート PROM を使用してデフォルトのブートデバイスを変更する方法」 233 ページの「SPARC: eeprom コマンドを使用してデフォルトのブートデバイスを変更する方法」
システムをリセットします。	システムをリセットすると、ハードウェアの診断テストが実行されたあとで、システムがリブートします。	234 ページの「SPARC: システムのリセット」
デフォルトのブートファイルを変更します。	システムがブートするデフォルトのカーネルを変更するには、次のいずれかの方法を使用します。 ■ ブート PROM を使用して <code>boot-file</code> パラメータを変更する。 ■ <code>eeprom</code> コマンドを使用して <code>boot-file</code> パラメータを変更する。	234 ページの「SPARC: ブート PROM を使用してデフォルトのカーネルを変更する方法」 234 ページの「SPARC: eeprom コマンドを使用してデフォルトのカーネルを変更する方法」

SPARC: ブート PROM の使用

ブート PROM はシステムのブートに使用します。必要に応じて、システムのブート方法を変更します。たとえば、システムをマルチユーザーレベルにする前に、どのデバイスからブートするかを設定し直したり、ハードウェア診断プログラムを実行したりする場合があります。

システム管理者は、通常 PROM レベルを使ってシステムをブートします。PROM レベルでデフォルトのブートファイルやブートデバイスを変更することもできます。

次のいずれかの作業を行う必要がある場合は、デフォルトのブートデバイスを変更する必要があります。

- 新しいドライブを永久または一時的にシステムに追加します
- ネットワークブート方法を変更します
- スタンドアロンシステムを一時的にネットワークからブートします

すべての PROM コマンドについては、[monitor\(1M\)](#) または [eeprom\(1M\)](#) のマニュアルページを参照してください。

▼ SPARC: システムの PROM リビジョン番号を確認する方法

- システムの PROM リビジョン番号を表示するには、`banner` コマンドを使用します。

```
ok banner
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #number.
Ethernet address number, Host ID: number.
```

ハードウェア構成情報が表示され、その中に PROM のリビジョン番号があります。この例では、PROM のリビジョン番号は 3.15 です。

▼ SPARC: システム上のデバイスを確認する方法

システム上のデバイスを確認して、ブートに適したデバイスを見極めることが必要な場合があります。

始める前に `probe` コマンドを使用してシステムに接続されているデバイスを安全に確認するためには、次のことを行なっておく必要があります。

- PROM の `auto-boot?` パラメータの値を `false` に変更します。

```
ok setenv auto-boot? false
```

- `reset-all` コマンドを発行して、システムのレジスタをクリアします。

```
ok reset-all
```

`sifting probe` コマンドを使用すると、システム上で利用可能な `probe` コマンドを表示できます。

```
ok sifting probe
```

システムのレジスタをクリアしないで `probe` コマンドを実行すると、次のメッセージが表示されます。

```
ok probe-scsi
This command may hang the system if a Stop-A or halt command
has been executed. Please type reset-all to reset the system
before executing this command.
Do you wish to continue? (y/n) n
```

- 1 システム上のデバイスを確認します。

```
ok probe-device
```

- 2 (省略可能) 電源障害後や reset コマンドの使用後にシステムをリブートするには、`auto-boot?` パラメータを **true** にリセットします。

```
ok setenv auto-boot? true
auto-boot? = true
```

- 3 システムをマルチユーザーモードでブートします。

```
ok reset-all
```

例 11-1 SPARC: システム上のデバイスを確認する

次の例は、Ultra™ 10 システムに接続されているデバイスの確認方法を示しています。

```
ok setenv auto-boot? false
auto-boot? = false
ok reset-all
Resetting ...
```

```
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #10933339.
Ethernet address 8:0:20:a6:d4:5b, Host ID: 80a6d45b.
```

```
ok probe-ide
Device 0 ( Primary Master )
    ATA Model: ST34321A

Device 1 ( Primary Slave )
    Not Present

Device 2 ( Secondary Master )
    Removable ATAPI Model: CRD-8322B

Device 3 ( Secondary Slave )
    Not Present
```

```
ok setenv auto-boot? true
auto-boot? = true
```

別の方法として、`devalias` コマンドを使うと、システムに接続されている「可能性のある」デバイスの別名と関連パスを確認できます。次に例を示します。

```
ok devalias
screen          /pci@1f,0/pci@1,1/SUNW,m64B@2
net             /pci@1f,0/pci@1,1/network@1,1
cdrom           /pci@1f,0/pci@1,1/ide@3/cdrom@2,0:f
disk            /pci@1f,0/pci@1,1/ide@3/disk@0,0
disk3          /pci@1f,0/pci@1,1/ide@3/disk@3,0
```

```

disk2                /pci@1f,0/pci@1,1/ide@3/disk@2,0
disk1                /pci@1f,0/pci@1,1/ide@3/disk@1,0
disk0                /pci@1f,0/pci@1,1/ide@3/disk@0,0
ide                  /pci@1f,0/pci@1,1/ide@3
floppy               /pci@1f,0/pci@1,1/ebus@1/fdthree
ttyb                 /pci@1f,0/pci@1,1/ebus@1/se:b
ttya                 /pci@1f,0/pci@1,1/ebus@1/se:a
keyboard!            /pci@1f,0/pci@1,1/ebus@1/su@14,3083f8:forcemode
keyboard             /pci@1f,0/pci@1,1/ebus@1/su@14,3083f8
mouse                /pci@1f,0/pci@1,1/ebus@1/su@14,3062f8
name                 aliases

```

▼ SPARC: デフォルトのブートデバイスを調べる方法

- 1 システムを **ok PROM** プロンプトにします。

詳細は、[222 ページの「スタンドアロンシステムをシャットダウンする方法」](#)を参照してください。

- 2 `printenv` コマンドを使用して、デフォルトのブートデバイスを調べます。

```
ok printenv boot-device
```

`boot-device` ブートするデバイスを設定するパラメータを示します。

`device[n]` `boot-device` の値 (disk または net) を設定します。`n` はディスク番号です。

次のような形式で、デフォルトのブートデバイス (`boot-device`) が表示されます。

```
boot-device = /pci@1f,4000/scsi@3/disk@1,0:a
```

デフォルトのブートデバイスがネットワークブートデバイスの場合は、次のような出力が表示されます。

```
boot-device = /sbus@1f,0/SUNW,fas@e,88000000/sd@a,0:a \
/sbus@1f,0/SUNW,fas@e,88000000/sd@0,0:a disk net
```

▼ SPARC: ブート PROM を使用してデフォルトのブートデバイスを変更する方法

デフォルトのブートデバイスをほかのデバイスに変更するには、まずシステム上のデバイスを確認しておく必要があります。システム上のデバイスを確認する方法については、[229 ページの「SPARC: システム上のデバイスを確認する方法」](#)を参照してください。

- 1 実行レベル 0 に変更します。

```
# init 0
```

ok PROM プロンプトが表示されます。詳細は、[init\(1M\)](#) のマニュアルページを参照してください。

- 2 boot-device パラメータの値を変更します。

```
ok setenv boot-device device[n]
```

ディスク番号を確認する必要がある場合は、probe コマンドのいずれかを使用します。

- 3 デフォルトのブートデバイスが変更されていることを確認します。

```
ok printenv boot-device
```

- 4 新しい boot-device 値を保存します。

```
ok reset-all
```

新しい boot-device 値が PROM に書き込まれます。

例 11-2 SPARC: デフォルトのブートデバイスを変更する

この例では、デフォルトのブートデバイスをディスクに設定しています。

```
# init 0
#
INIT: New run level: 0
.
.
.
The system is down.
syncing file systems... done
Program terminated
ok setenv boot-device /pci@1f,4000/scsi@3/disk@1,0
boot-device = /pci@1f,4000/scsi@3/disk@1,0
ok printenv boot-device
boot-device /pci@1f,4000/scsi@3/disk@1,0
ok boot
Resetting ...

screen not found.
Can't open input device.
Keyboard not present. Using ttys for input and output.

Sun Enterprise 220R (2 X UltraSPARC-II 450MHz), No Keyboard
OpenBoot 3.23, 1024 MB memory installed, Serial #13116682.
```


Ethernet address 8:0:20:c8:25:a, Host ID: 80c8250a.

Rebooting with command: boot disk1

Boot device: /pci@1f,4000/scsi@3/disk@1,0 File and args:

この例では、デフォルトのブートデバイスをネットワークに設定しています。

```
# init 0
#
INIT: New run level: 0
.
.
.
The system is down.
syncing file systems... done
Program terminated
ok setenv boot-device net
boot-device = net
ok printenv boot-device
boot-device net disk
ok reset
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #number.
Ethernet address number, Host ID: number.
```

Boot device: net File and args:

```
.
.
.
```

pluto console login:

▼ SPARC: eeprom コマンドを使用してデフォルトのブートデバイスを変更する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
- 2 ブートする代替カーネルを指定します。
eeprom boot-device new-boot-device
- 3 新しいパラメータが設定されていることを確認します。
eeprom boot-device

この出力には、`boot-device` パラメータの新しい `eeprom` 値が表示されるはずです。

SPARC: システムのリセット

ok プロンプトで次のコマンドを実行します。

```
ok reset-all
```

自己診断プログラムが実行され、ハードウェアの診断テストが行われたあと、`auto-boot?` パラメータが `true` に設定されていれば、システムがリブートします。

▼ SPARC: ブート PROM を使用してデフォルトのカーネルを変更する方法

- 1 実行レベル 0 に変更します。

```
# init 0
```

ok PROM プロンプトが表示されます。詳細は、[init\(1M\)](#) のマニュアルページを参照してください。

- 2 `boot-file` プロパティを代替カーネルに設定します。

```
ok setenv boot-file boot-file
```

- 3 デフォルトのブートデバイスが変更されていることを確認します。

```
ok printenv boot-file
```

- 4 新しい `boot-file` 値を保存します。

```
ok reset-all
```

新しい `boot-file` 値が PROM に書き込まれます。

▼ SPARC: eeprom コマンドを使用してデフォルトのカーネルを変更する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

- 2 ブートする代替カーネルを指定します。

```
# eeprom boot-file new boot-file
```

次に例を示します。

```
# eeprom boot-file=kernel.name/sparcv9/unix
```

- 新しいパラメータが設定されていることを確認します。

```
# eeprom boot-file
```

この出力には、指定したパラメータの新しい eeprom 値が表示されるはずです。

x86 システムでの Solaris ブート動作の変更 (作業マップ)

表 11-1 x86: ブート動作の変更: 作業マップ

作業	説明	参照先
eeprom コマンドを使用してブートファイルパラメータを設定します。	eeprom コマンドを使用して、x86 システムのブート動作を変更します。eeprom コマンドを使って設定するブートオプションは、ブート時に GRUB メニューでカーネルの動作を変更してオプションを上書きしないかぎり、システムをリブートしても持続します。	236 ページの「x86: eeprom コマンドを使用してブート動作を変更する方法」
ブート時に GRUB メニューを編集してブート動作を変更します。	ブート時に GRUB メニューを編集してブート動作を変更します。GRUB メニューでブート動作を変更することで指定するブートオプションは、次にシステムをリブートするまでしか持続しません。	240 ページの「x86: ブート時に GRUB メニューを編集してブート動作を変更する方法」
手動で menu.lst ファイルを編集してブート動作を変更します。	menu.lst 構成ファイルを編集して新しい OS エントリを追加するか、コンソールをリダイレクトすることで、ブート動作を変更します。ファイルに加えた変更は、システムをリブートしても持続します。	242 ページの「x86: menu.lst ファイルを編集してブート動作を変更する方法」
menu.lst ファイルを、findroot コマンドをサポートするエントリを含むように変更します。	インストールまたはアップグレードが完了したら、findroot コマンドを使用するそのほかのメニューエントリを menu.lst ファイルメニューに追加できます。	248 ページの「x86: findroot コマンドを使用する GRUB メニューエントリを追加する方法」

x86 システムのブート動作を変更する

x86 システムのブート動作を変更する主な方法は、次のとおりです。

- `eeeprom` コマンドを使用する方法。

`eeeprom` コマンドを使用して、一連の標準プロパティに別の値を割り当てます。これらの値は、SPARC の OpenBoot PROM NVRAM 変数と等価で、`/boot/solaris/bootenv.rc` ファイルに格納されます。`eeeprom` コマンドを使用してブート動作に加えた変更は、システムをリブートしても持続し、ソフトウェアのアップグレード中も保持されます。これらの変更を上書きするには、ブート時に GRUB メニューを編集するか、`menu.lst` ファイルを編集します。詳細は、[eeeprom\(1M\)](#) のマニュアルページを参照してください。

注 - `bootenv.rc` ファイルを直接編集して加えた変更は、ソフトウェアのアップグレード中に必ずしも保持されるとは限りません。そのため、この方法はお勧めしません。この種類の変更を加える場合は、`eeeprom` コマンドを使用する方法をお勧めします。

- ブート時に GRUB メニューを編集する方法。

ブート時に GRUB カーネルの動作を変更して加えた変更は、`eeeprom` コマンドを使用して設定するオプションより優先されます。ただし、これらの変更の効果は、次にシステムをブートするまでしか持続しません。詳細は、[kernel\(1M\)](#) のマニュアルページを参照してください。

- 手動で GRUB `menu.lst` ファイルを編集する方法。



注意 - `menu.lst` エントリに加えられたシステム生成の変更はすべて、システムのアップグレード中に変更されるか、失われます。ただし、手動で追加した新しいブートエントリは、アップグレード後も維持されます。`eeeprom` の設定を上書きするには、ブート時に GRUB メニューを編集するか、`menu.lst` ファイルを編集します。ブート時に GRUB メニューを編集して加えた変更は持続しません。`menu.lst` ファイルに加えた変更は、システムをリブートしても持続します。

▼ x86: `eeeprom` コマンドを使用してブート動作を変更する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 指定したパラメータを変更するには、`eeeprom` コマンドに適切な引数を付けて入力します。

```
# eeeprom parameter=new-value
```

- 3 新しいパラメータが設定されていることを確認します。

```
# eeeprom parameter
```

この出力には、指定したパラメータの新しい `eeeprom` 値が表示されるはずです。

例 11-3 x86: `eeeprom` コマンドを使用して `boot-file` パラメータを設定する

次の例は、システムが 64 ビットカーネルをブートするように手動で指定する方法を示しています。64 ビット対応のシステムが必要です。

```
# eeeprom boot-file=kernel/amd64/unix
```

次の例は、64 ビット対応システムで 32 ビットカーネルを手動でブートする方法を示しています。

```
# eeeprom boot-file=kernel/unix
```

次の例は、システム上で自動検出されたデフォルトのブート動作を復元する方法を示しています。

```
# eeeprom boot-file=""
```

x86: ブート時に **GRUB** メニューを編集してブート動作を変更する

次の例は、ZFS ルートファイルシステムからのシステムのブートをサポートする Solaris リリースの GRUB メインメニューを示しています。このメニューは、`menu.lst` 構成ファイルの内容に基づいて、システム上のブート可能なすべての OS インスタンスのメニューエントリを含んでいます。特に指定がないかぎり、メニューの最初のエントリはデフォルトです。別のブートエントリをデフォルトとして指定するには、`default=n` コマンドを `menu.lst` ファイルに追加します。この場合の `n` は 0 (最初のブートエントリ) から始まる数値です。

```
GNU GRUB  version 0.95  (637K lower / 3144640K upper memory)
+-----+
be1)
be1 failsafe
be3
be3 failsafe
be2
```

```
be2 failsafe
+-----+
      Use the ^ and v keys to select which entry is highlighted.
      Press enter to boot the selected OS, 'e' to edit the
      commands before booting, or 'c' for a command-line.
```

注 - menu.lst ファイルに含まれる情報は、Solaris のリリースと使用したインストール方法によって異なります。

GRUB メニューのブートエントリを編集するには、矢印キーを使ってエントリを選択してから、e と入力します。

```
GNU GRUB  version 0.95  (637K lower / 3144640K upper memory)
+-----+
findroot (BE_be1,0,a)
bootfs rpool/ROOT/szboot_0508
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive
+-----+
      Use the ^ and v keys to select which entry is highlighted.
      Press enter to boot the selected OS, 'e' to edit the
      commands before booting, or 'c' for a command-line.
```

ブート時に GRUB メニューを編集する方法については、[240 ページの「x86: ブート時に GRUB メニューを編集してブート動作を変更する方法」](#)を参照してください。

GRUB 編集メニューについて

次の例は、さまざまな GRUB 実装の編集メニューを示しています。

GRUB ZFS をサポートしている場合

```
grub edit> kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS,prop=value
[.prop=value...][-asrvxk] [-m smf-options] [-i altinit]
```

注 - ZFS をサポートしているシステムでブート引数を追加するときは、デフォルトの -B \$ZFS-BOOTFS 引数の後ろに追加の -B オプションを追加するようにしてください。

GRUB UFS をサポートしている場合

```
grub edit> kernel /platform/i86pc/multiboot [-asrvxk] [-m smf-options]
[-i altinit][-B prop=value [.prop=value...]]
```

ブート時に **GRUB** メニューを編集するときに指定できるブート引数

次の一覧では、ブート時に GRUB メニューを編集して指定できるブート引数とオプションについて説明します。

<code>multiboot</code>	ブートするカーネルを指定します。
<code>-a</code>	構成情報の入力を求めるプロンプトを表示します。
<code>-s</code>	システムをシングルユーザーモードでブートします。
<code>-r</code>	再構成ブートを指定します。
	接続されているすべてのハードウェアデバイスを検索してから、実際に見つかったデバイスだけに、ファイルシステムのノードを割り当てます。
<code>-v</code>	詳細メッセージを有効にした状態でシステムをブートします。
<code>-x</code>	クラスタモードではブートしません。
<code>-k</code>	カーネルデバッグを有効にした状態でシステムをブートします。
<code>-m <i>smf-options</i></code>	サービス管理機能 (Service Management Facility、SMF) のブート動作を制御します。復元オプションとメッセージオプションという、2 種類のオプションがあります。
<code>-i altinit</code>	代替実行可能ファイルを原始プロセスとして指定します。 <code>altinit</code> は実行可能ファイルへの有効なパスです。
<code>-B <i>prop=value[,prop=value]...</i></code>	カーネルのブートプロパティを指定します。

次に、`-B prop=val` オプションを使用して GRUB メニューでブート動作を変更できる各種方法を示します。

<code>-B console=ttya</code>	出力先をコンソールから <code>ttya</code> に変更します。
<code>-B acpi-enum=off</code>	デバイスの ACPI (Advanced Configuration and Power Interface) 列挙を使用不可にします。
<code>-B console=ttya,acpi-enum=off</code>	出力先をコンソールから <code>ttya</code> に変更し、デバイスの ACPI 列挙を使用不可にします。
<code>-B acpi-user-options=0x2</code>	ACPI を完全に使用不可にします。

注 - プロパティを指定するときに、`eeeprom` コマンドと GRUB コマンド行を使用した場合には、GRUB コマンドが優先されます。

▼ x86: ブート時に GRUB メニューを編集してブート動作を変更する方法

ブート時に GRUB メニューを編集して GRUB カーネルの動作を変更すると、その変更はシステムをリブートするまでしか持続しません。次にシステムをブートすると、デフォルトのブート動作が復元されます。

- 1 システムを再起動します。
ブートシーケンスが始まると、GRUB メインメニューが表示されます。
- 2 矢印キーを使用して編集対象のブートエントリを選択してから、`e` と入力して **GRUB** 編集メニューにアクセスします。
- 3 このメニューで、矢印キーを使用して `kernel` 行または `kernel$` 行を選択します。
- 4 `e` と入力して、ブート引数をその行に追加します。
- 5 指定する追加のブート引数を入力します。
- 6 **Return** キーを押して変更を保存し、前のメニューに戻ります。

注 - 変更を保存せずに GRUB メインメニューに戻るには、`Escape` キーを押します。

- 7 システムをブートするために、`b` と入力します。
適用した変更は、システムをブートしたあとで有効になります。

例 11-4 x86: 64 ビット対応のシステムで 32 ビットのカーネルをブートする

64 ビット対応のシステムで 32 ビットのカーネルをブートするには、`kernel/unix` 引数を追加します。

```
grub edit> kernel /platform/i86pc/multiboot kernel/unix
```


例 11-5 x86: シリアルコンソールをリダイレクトする

シリアルコンソールを `ttyb` にリダイレクトするには、`-B console=ttyb` 引数を追加します。

```
grub edit> kernel /platform/i86pc/multiboot -B console=ttyb
```

また、次の例に示すように、`input-device/output-device` プロパティを使用することもできます。

```
grub edit> kernel /platform/i86pc/multiboot -B input-device=ttyb,output-device=ttyb
```

次の例は、シリアル回線速度を上書きする方法を示しています。

```
grub edit> kernel /platform/i86pc/multiboot -B ttyb-mode="115200,8,n,1,-"
```

注意: この例のプロパティ値には、プロパティの区切り文字としてのコンマも含まれています。プロパティの構文解析部と混同しないようにするため、二重引用符を使用してプロパティ値全体を囲んでください。

x86: menu.lst ファイルを編集してブート動作を変更する

GRUB メニューは、`menu.lst` 構成ファイルに基づいており、カスタマイズできます。Solaris リリースをインストールまたはアップグレードすると、`bootadm` コマンドによって自動的に `menu.lst` ファイルが更新されて、その GRUB 実装でサポートされているメニューエントリが反映されます。新しくインストールした OS は、このファイルに登録され、システムのリブート時に GRUB メニューのブートエントリとして表示されます。Solaris OS 以外のオペレーティングシステムをインストールしたときは、あとでそのメニューエントリを `menu.lst` ファイルに手動で追加する必要があります。

次の例は、`menu.lst` ファイルの内容に基づいた、一般的な GRUB メインメニューを示しています。GRUB メインメニューは、使用可能なブートエントリとフェイルセーフアーカイブで構成されます。

```
GNU GRUB  version 0.95  (631K lower / 2095488K upper memory)
```

```
+-----+
| Solaris 10.1 ... X86                               |
| Solaris failsafe                                   |
|                                                     |
+-----+
```

デフォルトで登録されている OS ブートでは、設定可能なタイムアウトを利用できません。ブート対象となるデフォルトの OS ブートエントリは、`default` コマンドによって設定可能です。Solaris のインストールソフトウェアでは通常、有効な Solaris ブートエントリのいずれかがブートされるようにこのコマンドを設定します。Solaris OS の別のインスタンスをブートするか (適切な場合)、別の OS をブートするには、矢印キーを使用して別のブートエントリを強調表示します。次に、Enter キーを押してそのエントリをブートします。`default` コマンドが設定されていない場合は、GRUB メニューの最初のブートエントリがブートされます。

システムのブートに使用されるのは、アクティブな `menu.lst` ファイルだけです。システムのブート時に表示される GRUB メニューを変更するには、アクティブな GRUB `menu.lst` ファイルを編集します。それ以外の `menu.lst` ファイルに変更を加えても、システムのブート時に表示されるメニューに影響はありません。アクティブな `menu.lst` ファイルの場所を調べるには、`bootadm` コマンドの `list-menu` サブコマンドを使用します。`bootadm` コマンドの使用法の詳細は、[313 ページの「bootadm コマンドを使用してブートアーカイブを管理する」](#)を参照してください。

Solaris OS のそれぞれの GRUB 実装での `menu.lst` ファイルの詳細は、[329 ページの「x86: サポートされている GRUB 実装」](#)を参照してください。

▼ x86: menu.lst ファイルを編集してブート動作を変更する方法

次のいずれかの理由により、`menu.lst` ファイルの変更が必要になる場合があります。

- 新しい OS エントリを追加するため
- GRUB コンソールのリダイレクション情報を追加するため

始める前に システムのブートにはアクティブな GRUB `menu.lst` ファイルしか使用できないため、必ず適切なファイルを編集するようにしてください。それ以外の GRUB `menu.lst` ファイルに変更を加えても、システムのブート時に表示されるメニューに影響はありません。

アクティブな `menu.lst` ファイルの場所は、稼働するシステムに UFS ルートと ZFS ルートのどちらが含まれるかで異なります。

- UFS ルートの場合、アクティブな `menu.lst` ファイルは `/boot/grub/menu.lst` です。
- ZFS ルートの場合、アクティブな `menu.lst` ファイルは `/pool-name/boot/grub/menu.lst` です。

アクティブな GRUB menu.lst ファイルの場所を調べるには、bootadm コマンドで list-menu サブコマンドを指定します。

```
# bootadm list-menu
```

bootadm コマンドの詳細は、[bootadm\(1M\)](#) のマニュアルページを参照してください。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
- 2 アクティブな menu.lst ファイルに新しい OS エントリを追加するには、テキストエディタを使ってファイルを変更します。
menu.lst ファイル内のコメントを読めば、新しい OS エントリを追加するのに必要な情報が得られます。

次の例は、ZFS ブート対応の Solaris リリースが稼働するシステムの menu.lst ファイルを示しています。menu.lst ファイルに含まれるブートエントリは、実行している Solaris リリースに応じて異なります。

```
#----- ADDED BY BOOTADM - DO NOT EDIT -----
title Solaris Solaris 10 s10x_nbu6wos_nightly X86
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive
#-----END BOOTADM-----
```



注意 - menu.lst ファイルの元の内容を直接編集しないでください。このファイルの OS エントリに変更を加えるには、ファイルを手動で編集して既存の内容を複製します。次に、複製した内容に変更を加えます。

また、手動でこのファイルに新しいユーザーエントリを追加するときは、「Added by bootadm」や「Added by Live Upgrade」など、システムで予約されている保護コメントを含めないでください。手動で追加するエントリに保護コメントを含めないことで、これらのエントリはソフトウェアアップグレード中にそのまま維持されます。

デフォルトのエントリ以外に、エントリを新しく追加した場合は、同等の変更を手動で行なってください。

[-B*] フラグと [*] フラグが元の menu.lst ファイルに存在する場合は、これらのフラグを保持する必要があります。また、failsafe エントリには常に -s フラグが付いているはずです。

- 3 必要な情報を追加したあとで、ファイルを保存します。
ファイルに加えた変更は、次にシステムをリブートしたときに有効になります。

ヒント - Linux OS が稼働しているマシンに Solaris OS をインストールすると、システムのリブート時に Linux OS のエントリは GRUB メニューに表示されません。Solaris ソフトウェアをインストールする前に、Linux 情報が含まれている menu.lst ファイルのコピーを保存します。インストールが完了したら、Solaris パーティションに新しく作成された menu.lst ファイルに Linux 情報を戻します。

menu.lst ファイルに加えた変更は、Solaris OS と直接の関係がありません。このため、eeprom コマンドを使用して変更を行うことはできません。ファイルを直接編集する必要があります。Solaris ソフトウェアのアップグレード処理では、menu.lst ファイルに加えた変更内容がすべて保持されます。



注意 - Solaris GRUB では、Linux OS と Solaris OS の両方の OS をブートできますが、Linux GRUB では Solaris OS をブートできません。

必ず次のいずれかの条件が満たされていることを確認してください。

- Solaris fdisk パーティションがアクティブになっていて、そこに GRUB がインストールされていること。また、menu.lst ファイルがアクティブな GRUB メニューであること
- Solaris GRUB がマスターブートレコード (MBR) にインストールされ、Solaris fdisk パーティションにある menu.lst を参照すること

各 Solaris リリースに関連する GRUB menu.lst の詳細は、[329 ページの「x86: サポートされている GRUB 実装」](#)を参照してください。

例 11-6 ZFS ブートローダーを備えたシステムの menu.lst ファイル

次の例は、ZFS ブートローダーを備えたシステムで menu.lst ファイルがどのように表示されるかを示しています。デフォルトでは、このシステムは ZFS ルートファイルシステムからブートします。このファイルの内容は、インストールの種類によって異なります。

新規インストールまたは標準アップグレードの場合:

```
title Solaris 10 s10x_nbu6wos_nightly X86
findroot (pool_rpool,0,a)
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive

title Solaris failsafe
findroot (pool_rpool,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttyb
module /boot/x86.miniroot-safe
```

Solaris Live Upgrade

```

title bel
findroot (BE_bel,0,a)
bootfs rpool/ROOT/szboot_0508
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive

title bel failsafe
findroot (BE_bel,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttyb
module /boot/x86.miniroot-sa

```

例 11-7 UFS ブートローダーを備えたシステムの menu.lst ファイル

次の例は、UFS ルートファイルシステムがインストールされているシステムで menu.lst ファイルがどのように表示されるかを示しています。デフォルトでは、このシステムは UFS ルートファイルシステムからブートします。

新規インストールまたは標準アップグレードの場合:

```

title Solaris 10 s10x_nbu6wos_nightly X86
findroot (rootfs0,0,a)
kernel /platform/i86pc/multiboot
module /platform/i86pc/boot_archive

title Solaris failsafe
findroot (rootfs0,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttyb
module /boot/x86.miniroot-safe

```

Solaris Live Upgrade の場合:

```

title bel
findroot (BE_bel,0,a)
kernel /platform/i86pc/multiboot
module /platform/i86pc/boot_archive

title bel failsafe
findroot (BE_bel,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttyb
module /boot/x86.miniroot-safe

```

x86: アクティブな GRUB menu.lst ファイルを検出する

ZFS ルートを含むシステムでは、アクティブな menu.lst ファイルは通常 `/pool-name/boot/grub/menu.lst` にあります。

UFS ルートを含むシステムでは、アクティブな menu.lst ファイルは通常 `/boot/grub/menu.lst` にあります。

アクティブな GRUB メニューを検出するには、`bootadm` コマンドで `list-menu` サブコマンドを指定します。

```
# bootadm list-menu
```

このコマンドを使用すると、アクティブな menu.lst ファイルの内容も表示できます。

```
# bootadm list-menu
```

```
The location for the active GRUB menu is: /pool-name/boot/grub/menu.lst
```

```
default 0
```

```
timeout 10
```

```
0 be1
```

```
1 be1 failsafe
```

```
2 be3
```

```
3 be3 failsafe
```

```
4 be2
```

```
5 be2 failsafe
```

`bootadm` コマンドの使用法の詳細は、[313 ページの「bootadm コマンドを使用してブートアーカイブを管理する」](#)を参照してください。

x86: findroot コマンドの実装

Solaris Live Upgrade など、すべての Solaris インストール方法では、ブート対象の x86 システム上のディスクスライスを指定する際に `findroot` コマンドを使用するようになりました。この実装は、UFS ルートだけでなく ZFS ルートを含むシステムのブートにも適用されます。この情報は、GRUB が使用する menu.lst ファイル内に格納されます。以前は、`root` コマンド `root(hd0.0.a)` を使ってブート対象のディスクスライスを明示的に指定していました。

インストール方法には、Solaris Live Upgrade、JumpStart、インストール GUI プログラムなどがあります。

`findroot` コマンドのほかに、スライス (*mysign*, 0, a) 上の署名ファイルも追加されます。この場合の *mysign* は `/boot/grub/bootsign` ディレクトリにある署名ファイルの名前です。ZFS ルートからシステムをブートすると、ZFS GRUB プラグインが ZFS ファイルシステムを探して、`fdisk` パーティション 0 のスライス a にマウントしようとします。

署名ファイルの名前は、使用したインストールの種類によって異なります。`findroot` コマンドで使用される命名規則についての詳細は、[327 ページ](#) の「`findroot` コマンドで使用される命名規則」を参照してください。

インストールまたはアップグレードが完了したら、`findroot` コマンドを使用するその他のメニューエントリを GRUB メニューに追加できます。方法については、[248 ページ](#) の「x86: `findroot` コマンドを使用する GRUB メニューエントリを追加する方法」を参照してください。



注意 - ブートの署名は一意にする必要があります。Solaris ソフトウェアの複数のインスタンス間で複製される、システム生成の署名またはユーザーの署名を使用したり削除したりしないでください。そのようにすると、正しくない OS インスタンスがブートされたり、システムがブートできなくなる場合があります。

`root` コマンドは、Windows のブートなどの特定の場合に `menu.lst` ファイルで引き続き使用できます。ただし、`findroot` コマンドが優先される場合、`root` コマンドの使用はお勧めできません。

例 11-8 x86: UFS ブートローダーをサポートするシステムのデフォルトの `menu.lst` ファイル

次の例は、`findroot` コマンドを実装する `menu.lst` ファイルエントリの形式を示しています。

```
title Solaris 10 s10x_nbu6wos_nightly X86
findroot (pool_rpool,0,a)
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive

title Solaris failsafe
findroot (pool_rpool,0,a)
kernel /boot/multiboot kernel/unix -s -B console=tttyb
module /boot/x86.miniroot-safe
```

例 11-9 x86: ZFS ブートローダーをサポートするデフォルトの menu.lst ファイル

次の例は、ZFS ブートローダーをサポートするシステムの menu.lst ファイルを示しています。Solaris Live Upgrade を実行すると、ZFS ルートファイルシステムからブートするための情報がこのファイルに自動的に追加されます。

```
title bel
findroot (BE_bel,0,a)
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive

title bel failsafe
findroot (BE_bel,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttyb
module /boot/x86.miniroot-safe
```

▼ x86: findroot コマンドを使用する GRUB メニューエントリを追加する方法

この手順は、findroot コマンドを使用するユーザー定義エントリで menu.lst ファイルを手動で更新する方法を示しています。通常、これらのエントリはインストールまたはアップグレード後に追加されます。findroot コマンドを使用するユーザー定義エントリを追加するためのガイドラインについては、[246 ページの「x86: findroot コマンドの実装」](#)を参照してください。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
- 2 ブート対象のルート (/) ファイルシステムまたはルートプールのブート署名ファイルを作成します。
 - ZFS プール *my-pool* の場合、ブート署名ファイルを */my-pool/boot/grub/bootsign* ディレクトリに作成します。

```
# touch /my-pool/boot/grub/bootsign/user-sign
```
 - UFS ファイルシステムの場合、ブート署名ファイルをブート対象のルートファイルシステムの */boot/grub/bootsign* ディレクトリに作成します。

```
# touch /boot/grub/bootsign/user-sign
```

注- ブート署名用に選択したファイル名が一意であることを確認してください。Solaris ソフトウェアの複数のインスタンス間で複製されるシステム生成の署名またはユーザーの署名の名前を使用しないでください。そのようにすると、システムがブートできなくなったり、間違った Solaris インスタンスがブートされたりする場合があります。

3 findroot コマンドを含むメニューエントリを追加します。

a. アクティブな menu.lst ファイルを検出します。

```
# bootadm list-menu
```

b. テキストエディタを使用して、アクティブな menu.lst ファイルを編集し、次のエントリを追加します。

```
title      User Solaris boot entry
findroot   (user-sign, 3, c)
kernel$    /platform/i86pc/multiboot
module     /platform/i86pc/boot_archive
```

前の例で、3は4番目の fdisk パーティション (パーティションは0から始まる) を表します。cはSolaris fdisk パーティション内のスライス (スライスはaから始まる) を表します。

4 システムを再起動します。

新しいエントリが GRUB メニューに表示され、指定の Solaris OS インスタンスをブートするために選択できます。

Solaris システムのブート (手順)

この章では、SPARC システムおよび x86 システムで Solaris リリースをブートするための手順について説明します。

この章の内容は次のとおりです。

- 251 ページの「SPARC システムのブート (作業マップ)」
- 271 ページの「GRUB を使用して x86 システムをブートする (作業マップ)」

ブートプロセスの概要については、第 9 章「システムのシャットダウンとブート (概要)」を参照してください。

注 - Solaris 10 1/06 リリース以降では、オープンソースの GRand Unified Bootloader (GRUB) が x86 システム上で実装されています。GRUB は、カーネルモジュールおよび構成ファイルが含まれているブートアーカイブをシステムのメモリーに読み込む役割を果たします。

GRUB ベースのブートを実装していない Solaris リリースで x86 システムをブートする方法については、第 16 章「x86: GRUB を実装しないシステムのブート (手順)」を参照してください。

SPARC システムのブート (作業マップ)

作業	説明	参照先
SPARC システムを実行レベル 3 でブートします。	このブート方法は、システムのシャットダウン後またはシステムのハードウェア保守作業の実行後に使用します。	253 ページの「SPARC: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法」

作業	説明	参照先
SPARC システムを実行レベル S でブートします。	このブート方法は、ファイルシステムのバックアップなど、システムの保守作業の実行後にシステムをブートする場合に使用します。このレベルでは、一部のファイルシステムだけがマウントされ、ユーザーはシステムにログインできません。	254 ページの「SPARC: システムを実行レベル S (シングルユーザーレベル) でブートする方法」
SPARC システムを対話式でブートします。	このブート方法は、テストのためにシステムファイルまたはカーネルを一時的に変更したあとで使用します。	255 ページの「SPARC: システムを対話式でブートする方法」
デフォルト以外の Solaris カーネルをブートします。	この手順は、デフォルトのカーネル以外の Solaris カーネルをブートする場合に使用します。 別の方法として、代替ブートファイルのコピーを取得し、デフォルトのカーネルを新しいカーネルに変更してから、新しいデフォルトのブートデバイスをブートするように <code>boot-file</code> パラメータを設定することもできます。	257 ページの「SPARC: デフォルトのカーネル以外の Solaris カーネルをブートする方法」
SPARC システム上の使用可能な ZFS ブート可能データセットのリストを表示します。	システム上の ZFS プール内の使用可能な BE のリストを表示する場合は、 <code>boot -L</code> コマンドを使用します。 注- このオプションは、ZFS プールを含むブートデバイスでのみサポートされています。	260 ページの「SPARC: ZFS ルートプール内の使用可能なブート可能データセットのリストを表示する方法」
SPARC システムを ZFS ルートファイルシステムからブートします。	指定の ZFS データセットをブートする場合は、 <code>boot -Z</code> オプションを使用します。 注- このオプションは、ZFS プールを含むブートデバイスでのみサポートされています。	261 ページの「SPARC: ZFS ルートファイルシステムからブートする方法」
SPARC システムでフェイルセーフアーカイブをブートします。	この手順を使用して、SPARC システムでフェイルセーフアーカイブをブートします。次に、 <code>bootadm</code> コマンドを実行してブートアーカイブを更新します。	265 ページの「SPARC システムでフェイルセーフアーカイブをブートする方法」

作業	説明	参照先
SPARC システムをネットワークからブートします。	このブート方法は、システムをネットワークからブートする場合に使用します。この方法は、ディスクレスクライアントのブートにも使用されます。	270 ページの「 SPARC: システムをネットワークからブートする方法 」

SPARC システムのブート

システムの電源を切ってから入れ直すと、マルチユーザーのブートシーケンスが開始されます。このあとに示す手順では、ok PROM プロンプトからさまざまな実行レベルでブートする方法を説明します。これらの手順では、特に指示がない限り、システムが正常な状態でシャットダウンしていることを前提とします。

who -r コマンドを使って、システムが指定した実行レベルになっていることを確認します。実行レベルについては、[第 17 章「サービスの管理 \(概要\)」](#)を参照してください。

▼ SPARC: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法

次の手順に従って、現時点で実行レベル 0 になっているシステムを実行レベル 3 でブートします。

- 1 システムを実行レベル 3 でブートします。

```
ok boot
```

自動ブート処理によって、一連の起動メッセージが表示され、システムが実行レベル 3 になります。詳細は、[boot\(1M\)](#) のマニュアルページを参照してください。

- 2 システムが実行レベル 3 になっていることを確認します。

ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示されます。

```
hostname console login:
```

例 12-1 SPARC: システムを実行レベル 3 (マルチユーザーレベル) でブートする

次の例は、システムを実行レベル 3 でブートしたときに表示されるメッセージを示しています。

```
ok boot
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz)
OpenBoot 3.15, 128 MB memory installed, Serial #number.
```

Ethernet address *number*, Host ID: *number*.

```
Rebooting with command: boot
Boot device: /pci@1f,0/pci@1,1/ide@3/disk@0,0:a File and args: kernel/sparcv9/unix
SunOS Release 5.10 Version s10_60 64-bit
Copyright 1983-2004 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
configuring IPv4 interfaces: hme0.
add net default: gateway 172.20.27.248
Hostname: starlite
The system is coming up. Please wait.
NIS domain name is example.com
starting rpc services: rpcbind keysern ypbind done.
Setting netmask of hme0 to 255.255.255.0
Setting default IPv4 interface for multicast: add net 224.0/4: gateway starlite
syslog service starting.The system is ready.
Starting Sun(TM) Web Console Version 2.1-dev..
volume management starting.
The system is ready.
starlite console login:
```

前の例では、*sparcv9*が例として使用されました。この文字列は、*isainfo -k* コマンドの出力と一致します。

▼ SPARC: システムを実行レベルS(シングルユーザーレベル)でブートする方法

次の手順に従って、現時点で実行レベル0になっているシステムを実行レベルSでブートします。この実行レベルは、ファイルシステムのバックアップなど、システムの保守作業に使用されます。

- 1 システムを実行レベルSでブートします。

```
ok boot -s
```

- 2 次のメッセージが表示されたら、スーパーユーザーのパスワードを入力します。

```
SINGLE USER MODE
```

```
Root password for system maintenance (control-d to bypass): xxxxxx
```

- 3 システムが実行レベルSになっていることを確認します。

```
# who -r
```

- 4 実行レベルをSに移行して行う必要がある保守作業を行います。

- 5 システム保守作業が完了したら、**Ctrl+D** キーを押してシステムをマルチユーザー状態にします。

例 12-2 SPARC: システムを実行レベル S (シングルユーザーレベル) でブートする

次の例は、システムを実行レベル S でブートしたときに表示されるメッセージを示しています。

```
ok boot -s
.
.
.
Sun Microsystems Inc.   SunOS 5.10 Version Generic_120012-14 32-bit
Copyright 1983-2003 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
configuring IPv4 interfaces: hme0.
Hostname: starlite

SINGLE USER MODE

Root password for system maintenance (control-d to bypass): xxxxxx
single-user privilege assigned to /dev/console.
Entering System Maintenance Mode
Oct 14 15:01:28 su: 'su root' succeeded for root on /dev/console
Sun Microsystems Inc.   SunOS 5.10
# who -r
.          run-level S  Sep 19 08:49      S      0  ?
      (Perform some maintenance task)
# ^D
```

▼ SPARC: システムを対話式でブートする方法

代替カーネルまたは `/etc/system` ファイルを指定する必要がある場合は、このブートオプションを使用します。

始める前に `boot -a` コマンドを使用して、SPARC システムを対話式でブートするときに代替ファイル `/etc/system` を指定するには、システムをブートする前に次の手順を実行する必要があります。

- 1. `/etc/system` ファイルと `boot/solaris/filelist.ramdisk` ファイルのバックアップコピーを作成します。


```
# cp /etc/system /etc/system.bak
# cp /boot/solaris/filelist.ramdisk /boot/solaris/filelist.ramdisk.orig
```
- 2. `etc/system.bak` ファイル名を `/boot/solaris/filelist.ramdisk` ファイルに追加します。

```
# echo "etc/system.bak" >> /boot/solaris/filelist.ramdisk
```

- ブートアーカイブを更新します。

```
# bootadm update-archive -v
```

- 1 システムを対話式でブートします。

```
ok boot -a
```

- 2 次のようにシステムプロンプトに応答します。

- a. プロンプトが表示されたら、ブートに使用するカーネルの名前を入力します。

デフォルトのカーネルファイル名を使用する場合は、Enter キーを押します。そうでない場合は、代替カーネルの名前を入力して、Enter キーを押します。

- b. プロンプトが表示されたら、modules ディレクトリの代替パスを入力します。

デフォルトのモジュールディレクトリを使用する場合は、Enter キーを押します。そうでない場合は、モジュールディレクトリへの代替パスを入力して、Enter キーを押します。

- c. プロンプトが表示されたら、代替システムファイルの名前を入力します。

/etc/system ファイルが破損している場合、/dev/null を入力します。

- d. プロンプトが表示されたら、root ファイルシステムを入力します。

ローカルディスクからのブートに UFS (デフォルト) を選択する場合は Enter キーを押し、ネットワークブートの場合は NFS と入力します。

- e. プロンプトが表示されたら、ルートデバイスの物理名を入力します。

代替デバイス名を入力します。デフォルトを使用する場合は Return キーを押します。

- 3 これらの質問に応答するためのプロンプトが表示されない場合は、boot -a コマンドを正しく入力しているかどうか確認してください。

例 12-3 SPARC: システムを対話式でブートする

次の例では、利用できるデフォルトの選択例 ([] で囲まれた部分) を示します。boot -a コマンドを使用して代替ファイルシステムをブートする方法と例については、[255 ページの「SPARC: システムを対話式でブートする方法」](#)を参照してください。

```
ok boot -a
```

```
.  
.   
.
```



```

Rebooting with command: boot -a
Boot device: /pci@1f,0/pci@1,1/ide@3/disk@0,0:a
File and args: -a
Enter filename [kernel/sparcv9/unix]:      Press Return
Enter default directory for modules [/platform/SUNW,Ultra-5_10/kernel
/platform/sun4u/kernel /kernel /usr/kernel]:      Press Return
Name of system file [etc/system]:      Press Return
SunOS Release 5.10 Version S10_60 64-bit
Copyright (c) 1983-2004 by Sun Microsystems, Inc. All rights reserved
Use is subject to license terms.
root filesystem type [ufs]:      Press Return
Enter physical name of root device
[/pci@1f,0/pci@1,1/ide@3/disk@0,0:a]:      Press Return
configuring IPv4 interfaces: hme0.
Hostname: starlite
The system is coming up. Please wait.
checking ufs filesystems
.
.
.
The system is ready.
starlite console login:

```

▼ SPARC: デフォルトのカーネル以外の **Solaris** カーネルをブートする方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
- 2 既存の **Solaris** カーネルのコピーを取得し、その名前を変更します。
- 3 手順2でコピーして名前を変更したカーネルを `/etc/boot/solaris/filelist.ramdisk` ファイルに追加します。

```
# echo "kernel.name" >> /boot/solaris/filelist.ramdisk
```

- 4 代替カーネルが `/etc/boot/solaris/filelist.ramdisk` ファイルに追加されていることを確認します。

```
# cat > /etc/boot/solaris/filelist.ramdisk
```

- 5 `bootadm` コマンドを使用してブートアーカイブを更新します。

```
# bootadm update-archive
```

- 6 実行レベル0に変更します。

```
# init 0
```

ok PROM プロンプトが表示されます。

7 代替カーネルをブートします。

```
ok boot alternate-kernel
```

次に例を示します。

```
ok boot kernel.mymame/sparcv9/unix
```

- デフォルトで代替カーネルをブートするには、次の手順に従います。

- a. **boot-file** パラメータを新しいカーネルに設定します。

```
ok setenv boot-file kernel.name/sparc9/unix
```

- b. **boot-file** プロパティーが変更されていることを確認します。

```
ok printenv boot-file
```

- c. システムを再起動します。

```
ok boot
```

8 システムのブートが完了したら、代替カーネルがブートされたことを確認します。

```
# prtconf -vp | grep whoami
```

例 12-4 デフォルトのブートファイルを変更して代替 Solaris カーネルをブートする

```
# cp -r /platform/sun4v/kernel /platform/sun4vu/kernel.caiobella
```

```
# echo "kernel.caiobella" >> /boot/solaris/filelist.ramdisk
```

```
# cat > /etc/boot/solaris/filelist.ramdisk
```

```
/platform/sun4v/kernel.caiobella
```

```
^D (control D)
```

```
ok setenv boot-file kernel.caiobells/sparcv9/unix
```

```
ok printenv boot-file
```

```
boot-file = kernel.caiobella/sparcv9/unix
```

```
ok boot
```

```
SC Alert: Host System has Reset
```

```
SC Alert: Host system has shut down.
```

```
Sun Fire T200, No KeyboardCopyright 2006 Sun Microsystems, Inc. All rights reserved.
```

```
OpenBoot 4.25.0.build_01***PROTOTYPE BUILD***, 32760 MB memory available, Serial
#69060038.
Ethernet address 0:x:4f:x:c5:c6, Host ID: 8xxc5c6.
```

```
Rebooting with command: boot
Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a File and
args: kernel.caiobella/sparcv9/unix
SunOS Release 5.10
Copyright 1983-2007 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
DEBUG enabled
misc/forthdebug (176650 bytes) loaded
Hostname: seasonz
NIS domain name is lab.domain.sun.com
Reading ZFS config: done.
```

```
seasonz console login:
Password:
Last login: Mon Nov 12 18:02:00 on console
Sun Microsystems Inc. SunOS 5.10
.
.
.
You have new mail.
#
#
# prtconf -vp | grep whoami
whoami: '/platform/sun4v/kernel.caiobella/sparcv9/unix'
```

SPARC システムで ZFS ルートファイルシステムからブートする

Solaris SPARC プラットフォームでの ZFS からのブートをサポートするために、2つの新しいブートオプションが追加されました。

-L ZFS プール内の使用可能なブート可能データセットのリストを表示します。

注 -boot -L コマンドは、コマンド行からではなく OBP から実行されます。

-Z dataset 指定の ZFS ブート可能データセットのルートファイルシステムをブートします。

ZFS ルートファイルシステムからシステムをブートする場合は、最初に OBP から `-L` オプションで `boot` コマンドを実行して、システム上の使用可能な BE のリストを表示します。次に、`-z` オプションを使用して指定の BE をブートします。

詳細は、[boot\(1M\)](#) のマニュアルページを参照してください。

▼ SPARC: ZFS ルートプール内の使用可能なブート可能データセットのリストを表示する方法

SPARC システムでは、`menu.lst` ファイルに次の 2 つの GRUB コマンドが含まれています。

- `title` – ブート環境のタイトルを指定する
- `bootfs` – ブート可能データセットのフルネームを指定する

ZFS プール内のブート可能データセットのリストを表示するには、次のどちらかの方法を選択します。

- `lustatus` コマンドを使用する。このコマンドを実行すると、指定の ZFS プール内のすべての BE のリストが表示されます。
`lustatus` コマンドは x86 システムでも使用できます。
- `boot -L` コマンドを使用する。このコマンドを実行すると、指定の ZFS プール内の使用可能な BE のリストが表示され、システムをブートするための方法も示されます。

次の手順では、`boot -L` コマンドを使用してシステム上の使用可能な BE のリストを表示する方法を示します。このコマンドの実行後に指定の BE をブートするには、画面に表示された指示に従います。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 システムを `ok` プロンプトにします。

```
# init 0
```

- 3 ZFS プール内の使用可能な BE のリストを表示します。

```
ok boot device-specifier -L
```

- 4 (省略可能) 表示されているいずれかのエントリをブートするには、そのエントリの番号を入力します。指定の BE をブートするには、画面に表示された指示に従います。方法については、[261 ページの「SPARC: ZFS ルートファイルシステムからブートする方法」](#)を参照してください。

例 12-5 SPARC:boot -L を使用してシステム上の使用可能な BE のリストを表示する

```
# init 0
# svc.startd: The system is coming down. Please wait.
svc.startd: 94 system services are now being stopped.
svc.startd: The system is down.
syncing file systems... done
Program terminated
ok boot -L
.
.
.
Boot device: /pci@1f,0/pci@1/scsi@8/disk@0,0 File and args: -L
zfs-file-system
Loading: /platformsun4u/bootlst
1.s10s_nbu6wos
2.zfs2BE
Select environment to boot: [ 1 - 2 ]: 2

to boot the selected entry, invoke:
boot [<root-device>] -Z rpool/ROOT/zfs2BE
```

参照 詳細は、『[Solaris ZFS 管理ガイド](#)』の第 5 章「ZFS ルートファイルシステムのインストールと起動」を参照してください。

▼ SPARC: ZFS ルートファイルシステムからブートする方法

ZFS からブートする方法は、UFS からブートする方法とは異なります。ZFS からブートするときは、デバイス識別子によって単一のルートファイルシステムではなく、ストレージプールが識別されます。ストレージプールには、複数のブート可能データセット (ルートファイルシステム) が含まれていることがあります。そのため、ZFS からブートするときは、ブートデバイスによってデフォルトと見なされるプール内のルートファイルシステムも識別する必要があります。デフォルトでは、デフォルトのブートデバイスはプールの `bootfs` プロパティによって識別され

ます。この手順では、ZFS ブート可能データセットを指定してシステムをブートする方法を示します。使用可能なすべてのブートオプションの詳細は、[boot\(1M\)](#) のマニュアルページを参照してください。

注 - `bootfs` プロパティを以前に正しく設定した場合 (`luactivate` コマンドを使用して BE をアクティブにした場合など)、システムは自動的に ZFS ルートをブートします。

詳細は、[zpool\(1M\)](#) のマニュアルページを参照してください。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。
- 2 システムを `ok` プロンプトにします。
`# init 0`
- 3 (省略可能) 使用可能な BE のリストを表示するには、`-L` オプションを指定して `boot` コマンドを実行します。
方法については、[260 ページ](#)の「[SPARC: ZFS ルートプール内の使用可能なブート可能データセットのリストを表示する方法](#)」を参照してください。
- 4 指定のエントリをブートするには、そのエントリの番号を入力し、**Return** キーを押します。
Select environment to boot: [1 - 2]:
- 5 システムをブートするには、画面に表示された指示に従います。
To boot the selected entry, invoke:
`boot [<root-device>] -Z rpool/ROOT/dataset`

`ok boot -Z rpool/ROOT/dataset`
次に例を示します。

`# boot -Z rpool/ROOT/zfs2BE`

- 6 システムのブートが完了したら、次のコマンドを入力してアクティブな **BE** を確認します。

```
# prtconf -vp | grep whoami
```

- アクティブな **BE** のブートパスを表示するには、次のコマンドを入力します。

```
# prtconf -vp | grep bootpath
```

- 別の方法として、`df -lk` コマンドを使用して、正しい **BE** がブートされたかどうかを確認することもできます。

例 12-6 SPARC: ZFS ルートファイルシステムからの起動

次の例は、`boot -Z` コマンドを使用して SPARC システムで ZFS データセットをブートする方法を示しています。

```
# init 0
# svc.startd: The system is coming down. Please wait.
svc.startd: 79 system services are now being stopped.
svc.startd: The system is down.
syncing file systems... done
Program terminated
ok boot -Z rpool/ROOT/zfs2BEe
Resetting
LOM event: =44d+21h38m12s host reset
g ...

rProcessor Speed = 648 MHz
Baud rate is 9600
8 Data bits, 1 stop bits, no parity (configured from lom)

Firmware CORE Sun Microsystems, Inc.
@(#) core 1.0.12 2002/01/08 13:00
software Power ON
Verifying nVRAM...Done
Bootmode is 0
[New I2C DIMM address]
.
.
.
Environment monitoring: disabled
Executing last command: boot -Z rpool/ROOT/zfs2BE
Boot device: /pci@1f,0/pci@1/scsi@8/disk@0,0 File and args: -Z rpool/ROOT/zfs2Be
zfs-file-system
Loading: /platform/SUNW,UltraAX-i2/boot_archive
Loading: /platform/sun4u/boot_archive
ramdisk-root hsfs-file-system
```

```
Loading: /platform/SUNW,UltraAX-i2/kernel/sparcv9/unix
Loading: /platform/sun4u/kernel/sparcv9/unix
.
.
.
Hostname: mallory
NIS domainname is boulder.Central.Sun.COM
Reading ZFS config: done.
Mounting ZFS filesystems: (6/6)

mallory console login:
```

参照 指定の ZFS ブート可能データセットのフェイルセーフアーカイブをブートする方法については、[265 ページの「SPARC システムでフェイルセーフアーカイブをブートする方法」](#)を参照してください。

SPARC システムでフェイルセーフアーカイブをブートする

ブートアーカイブであるルート (/) ファイルシステムイメージからシステムをブートし、そのあとでこのファイルシステムを実際のルートデバイスに再マウントすると、ブートアーカイブとルートファイルシステムが一致しなかったり、「矛盾」していたりすることがたまにあります。このような状況では、システムの適切な処理や完全性が損なわれます。システムでは、ルート (/) ファイルシステムがマウントされると、メモリー内のファイルシステムを放棄する前に、2つのファイルシステムを照合して整合性検査を行います。不一致が検出されると、通常のブートシーケンスが中断され、システムは「フェイルセーフモード」に戻ります。

カーネルファイルの更新の直後に、システム障害、電源障害、またはカーネルパニックが発生した場合も、ブートアーカイブとルート (/) ファイルシステムが同期していない可能性があります。矛盾したブートアーカイブを使用してシステムが引き続きブートする場合でも、フェイルセーフアーカイブをブートしてブートアーカイブを更新することをお勧めします。bootadm コマンドを使用してブートアーカイブを手動で更新することもできます。詳細は、[313 ページの「bootadm コマンドを使用してブートアーカイブを管理する」](#)を参照してください。

フェイルセーフアーカイブをブートできるのは、復旧のためか、SPARC と x86 の両方のプラットフォームでブートアーカイブを更新する場合です。

SPARC プラットフォームでは、フェイルセーフアーカイブは次のとおりです。

```
/platform/'uname -m'/failsafe
```


次の構文を使用してフェイルセーフアーカイブをブートします。

```
ok boot -F failsafe
```

フェイルセーフブートは、ZFS からブートされるシステムでもサポートされます。ZFS をルートとした BE からブートするときは、各 BE に独自のフェイルセーフアーカイブが用意されています。フェイルセーフアーカイブは、UFS をルートとした BE と同様に、ルート (/) ファイルシステムが置かれている場所にあります。デフォルトのフェイルセーフアーカイブは、デフォルトのブート可能ファイルシステムにあるアーカイブです。デフォルトのブート可能ファイルシステム (データセット) は、プールの `bootfs` プロパティの値で示されます。

x86 ベースのフェイルセーフアーカイブのブートについては、[284 ページの「x86 システムでフェイルセーフアーカイブをブートする」](#)を参照してください。

ブートアーカイブの更新に使用できるもう一つの方法は、`boot-archive` サービスをクリアすることです。ただし、ブートアーカイブを更新する場合は、フェイルセーフアーカイブをブートするか、`bootadm` コマンドを使用することをお勧めします。詳細は、[310 ページの「boot-archive サービスをクリアして矛盾したブートアーカイブを更新する方法」](#)を参照してください。

▼ SPARC システムでフェイルセーフアーカイブをブートする方法

次の手順に従って、SPARC システムでフェイルセーフアーカイブをブートします。ブートアーカイブが更新されたあとでシステムがブートしない場合は、シングルユーザーモードでシステムをブートする必要がある場合があります。詳細は、[254 ページの「SPARC: システムを実行レベル S \(シングルユーザーレベル\) でブートする方法」](#)を参照してください。

注 - この手順には、特定の ZFS データセットのフェイルセーフアーカイブをブートする方法も記載されています。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。
- 2 システムを `ok` プロンプトにします。
`# init 0`

3 フェイルセーフアーカイブをブートします。

- デフォルトのフェイルセーフアーカイブをブートするには、次のように入力します。

```
ok boot -F failsafe
```

- 特定の ZFS データセットのフェイルセーフアーカイブをブートするには、次のように入力します。

```
ok boot -F failsafe -Z dataset
```

次に例を示します。

```
ok boot -F failsafe -Z rpool/ROOT/zfsBE2
```

注-ブートするデータセットの名前を調べるには、最初に `boot -L` コマンドを使用して、システム上の使用可能な BE のリストを表示します。詳細は、[260 ページの「SPARC: ZFS ルートプール内の使用可能なブート可能データセットのリストを表示する方法」](#)を参照してください。

矛盾したブートアーカイブが検出されると、メッセージが表示されます。

4 ブートアーカイブを更新するには、y と入力して **Return** キーを押します。

```
An out of sync boot archive was detected on rpool.
The boot archive is a cache of files used during boot
and should be kept in sync to ensure proper system operation.
```

```
Do you wish to automatically update this boot archive? [y,n,?] y
```

アーカイブが正常に更新されると、メッセージが表示されます。

```
The boot archive on rpool was updated successfully.
```

例 12-7 SPARC: フェイルセーフアーカイブをブートする

次の例は、SPARC システムでフェイルセーフアーカイブをブートする方法を示しています。デバイスを指定しない場合は、デフォルトのブートデバイスのフェイルセーフアーカイブがブートされます。

```
ok boot -F failsafe
```

```
Resetting ...
```

```
screen not found.
```

```
Can't open input device. Keyboard not present. Using ttya for input and output.
```

```
Sun Enterprise 220R (2 X UltraSPARC-II 450MHz), No Keyboard
```

```
OpenBoot 3.23, 1024 MB memory installed, Serial #13116682.
```

```

Ethernet address 8:0:20:c8:25:a, Host ID: 80c8250a.

Rebooting with command: boot -F failsafe
Boot device: /pci@1f,4000/scsi@3/disk@1,0:a File and args: -F failsafe
SunOS Release 5.10t
Copyright 1983-2007 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Configuring /dev Searching for installed OS instances...

An out of sync boot archive was detected on /dev/dsk/c0t1d0s0.
The boot archive is a cache of files used during boot and
should be kept in syncto ensure proper system operation.

Do you wish to automatically update this boot archive? [y,n,?] y
Updating boot archive on /dev/dsk/c0t1d0s0.
The boot archive on /dev/dsk/c0t1d0s0 was updated successfully.

Solaris 5.10 was found on /dev/dsk/c0t1d0s0.
Do you wish to have it mounted read-write on /a? [y,n,?] n
Starting shell.
#

```

例 12-8 SPARC: 指定の ZFS データセットのフェイルセーフアーカイブをブートする

次の例は、ZFS データセットのフェイルセーフアーカイブをブートする方法を示しています。最初に `boot -L` コマンドを使用して、使用可能なブート環境のリストを表示します。このコマンドは、`ok` プロンプトで実行する必要があります。

```

ok boot -L
Rebooting with command: boot -L
Boot device: /pci@1f,4000/scsi@3/disk@1,0 File and args: -L
1 zfsBE2
Select environment to boot: [ 1 - 1 ]: 1

To boot the selected entry, invoke:
boot [<root-device>] -Z rpool/ROOT/zfsBE2

Program terminated
{0} ok

Resetting ...

```

```
screen not found.
Can't open input device.
Keyboard not present.  Using ttys for input and output.

Sun Enterprise 220R (2 X UltraSPARC-II 450MHz), No Keyboard
OpenBoot 3.23, 1024 MB memory installed, Serial #13116682.
Ethernet address 8:0:20:c8:25:a, Host ID: 80c8250a.
```

```
{0} ok boot -F failsafe -Z rpool/ROOT/zfsBE2
Boot device: /pci@1f,4000/scsi@3/disk@1,0 File and args: -F failsafe -Z
rpool/ROOT/zfsBE2
SunOS Release 5.10
Copyright 1983-2008 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Configuring /dev
Searching for installed OS instances...
```

```
ROOT/zfsBE2 was found on rpool.
Do you wish to have it mounted read-write on /a? [y,n,?] y
mounting rpool on /a
```

```
Starting shell.
```

```
#
#
#
```

```
# zpool list
```

NAME	SIZE	USED	AVAIL	CAP	HEALTH	ALTROOT
rpool	16.8G	6.26G	10.5G	37%	ONLINE	/a

```
#
```

```
# zpool status
```

```
pool: rpool
state: ONLINE
scrub: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM
rpool	ONLINE	0	0	0
c0t1d0s0	ONLINE	0	0	0

```
errors: No known data errors
```

```
#
```

```
# df -h
```

Filesystem	size	used	avail	capacity	Mounted on

/ramdisk-root:a	163M	153M	0K	100%	/
/devices	0K	0K	0K	0%	/devices
/dev	0K	0K	0K	0%	/dev
ctfs	0K	0K	0K	0%	/system/contract
proc	0K	0K	0K	0%	/proc
mnttab	0K	0K	0K	0%	/etc/mnttab
swap	601M	344K	601M	1%	/etc/svc/volatile
objfs	0K	0K	0K	0%	/system/object
sharefs	0K	0K	0K	0%	/etc/dfs/sharetab
swap	602M	1.4M	601M	1%	/tmp
/tmp/root/etc	602M	1.4M	601M	1%	/.tmp_proto/root/etc
fd	0K	0K	0K	0%	/dev/fd
rpool/ROOT/zfsBE2	16G	5.7G	9.8G	37%	/a
rpool/export	16G	20K	9.8G	1%	/a/export
rpool/export/home	16G	18K	9.8G	1%	/a/export/home
rpool	16G	63K	9.8G	1%	/a/rpool

SPARC システムをネットワークからブートする

次のような場合に、システムをネットワークからブートする必要があります。

- システムを最初にインストールする場合
- システムをローカルディスクからブートできない場合
- システムがディスクレスクライアントである場合

次の2つのネットワーク構成ブート方法が利用できます。

- 逆アドレス解決プロトコル (RARP) および ONC+™ RPC Bootparams プロトコル
- 動的ホスト構成プロトコル (DHCP)
- ネットワークデバイスの場合、ローカルエリアネットワーク (LAN) 上でブートするプロセスと広域ネットワーク (WAN) 上でブートするプロセスは多少異なっています。どちらのネットワークブートシナリオでも、PROM によってブーター (この場合 `inetboot`) がブートサーバーまたはインストールサーバーからダウンロードされます。

LAN 上でブートするときは、ファームウェアは RARP と BOOTP または DHCP を使用して、ブートサーバーまたはインストールサーバーを検出します。そして、ブーター (この場合は `inetboot`) のダウンロードには TFTP が使用されます。

WAN 上でブートするときは、ファームウェアは DHCP または NVRAM プロパティを使用して、システムをネットワークからブートするために必要なインストールサーバー、ルーター、およびプロキシを検出します。ブーターのダウンロードに使用されるプロトコルは HTTP です。さらに、定義済みの非公開鍵を使ってブーターの署名が確認される場合もあります。

▼ SPARC: システムをネットワークからブートする方法

ブートサーバーが利用できれば、どのようなシステムもネットワークからブートできます。たとえば、スタンドアロンシステムがローカルディスクからブートできない場合は、そのシステムをネットワークからブートしてみてください。デフォルトのブートデバイスを変更または再設定する方法については、[231 ページの「SPARC: ブート PROM を使用してデフォルトのブートデバイスを変更する方法」](#)を参照してください。

sun-4u システムでは、次の 2 つのネットワーク構成ブート方法を使用できます。

- RARP – Reverse Address Resolution Protocol and ONC+ RPC Bootparams Protocol
- DHCP – Dynamic Host Configuration Protocol

デフォルトのネットワークブート方法は RARP に設定されています。ネットワークで利用できるブートサーバーによって、RARP または DHCP を選択できます。

注-Sun Ultra システムで DHCP ネットワークブート方法を使用するには、PROM のバージョンが 3.25.nn 以上でなければなりません。PROM のバージョンの確認方法については、[229 ページの「SPARC: システムの PROM リビジョン番号を確認する方法」](#)を参照してください。

RARP と DHCP の両方のプロトコルを使用できる場合、どちらのプロトコルを使用するかを `boot` コマンドに一時的に指定できます。あるいは、NVRAM 別名を設定すれば、システムをリブートしても有効に PROM レベルでネットワークブート方法を永続的に保存することができます。次の `nvalias` コマンドの例では、Sun Ultra 10 システムにおいてデフォルトで DHCP でブートするように、ネットワークデバイスの別名を設定します。

```
ok nvalias net /pci@1f,4000/network@1,1:dhcp
```

結果として、`boot net` と入力するだけで、システムは DHCP ネットワークブート方法を使用してブートします。

注-nvalias コマンドと nvunalias コマンドの構文を十分理解するまで、nvalias コマンドで NVRAMRC ファイルを変更しないでください。これらのコマンドの使用方法については、『OpenBoot 3.x コマンド・リファレンスマニュアル』を参照してください。

始める前に どちらのプロトコルでブートする場合でも、RARP または DHCP のブートサーバーがすでにネットワークに設定されていなければなりません。

- 1 必要に応じてシステムをシャットダウンします。
- 2 ネットワークからブートする方法を決定し、次のどちらかを選択します。

a. DHCP を使用してシステムをネットワークからブートするには、次のように入力します。

ok boot net[:dhcp]

上記 nvalias の例のように、デフォルトで DHCP でブートするように PROM 設定を変更してある場合は、boot net と指定するだけで、システムは DHCP でブートします。

b. RARP を使用してシステムをネットワークからブートするには、次のように入力します。

ok boot net[:rarp]

RARP はデフォルトのネットワークブート方法です。このため、デフォルトで DHCP でブートするように PROM 値を変更してある場合にだけ、boot net:rarp と指定しなければなりません。
- GRUB を使用して x86 システムをブートする (作業マップ)
- | 作業 | 説明 | 参照先 |
|--|--|--|
| x86 システムを実行レベル 3 (マルチユーザーレベル) でブートします。 | このブート方法は、システムのシャットダウン後またはシステムのハードウェア保守作業の実行後にシステムをマルチユーザーレベルに戻す場合に使用します。 | 272 ページの「x86: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法」 |
| x86 システムをシングルユーザーモードでブートします。 | このブート方法は、ファイルシステムのバックアップなど、システムの保守作業を実行する場合に使用します。 | 274 ページの「x86: システムを実行レベル S (シングルユーザーレベル) でブートする方法」 |
| x86 システムを対話式でブートします。 | このブート方法は、テストのためにシステムファイルまたはカーネルを一時的に変更したあとで使用します。 | 277 ページの「x86: システムを対話式でブートする方法」 |
- 第 12 章 • Solaris システムのブート (手順)

271

作業	説明	参照先
x86 システム上の ZFS ブート可能データセットのリストを表示します。	ZFS ルートファイルシステムを含む x86 システム上の使用可能な BE を表示する場合は、次のどちらかの方法を使用します。 ■ <code>lustatus</code> ■ <code>bootadm list-menu</code>	279 ページの「x86 システム上の使用可能な ZFS ブート環境のリストを表示する方法」
x86 システムを ZFS ルートファイルシステムからブートします。	ZFS ブートローダーをサポートする Solaris リリースをシステムにインストールするか、そのリリースにシステムをアップグレードする場合は、デフォルトの ZFS BE の GRUB メニューエントリに、デフォルトで <code>-B \$ZFS-BOOTFS</code> ブート引数が含まれます。システムは ZFS から自動的にブートします。 注-このオプションは、ZFS プールを含むブートデバイスでのみサポートされています。	280 ページの「x86 システムで ZFS ルートファイルシステムからブートする方法」
x86 システムでフェイルセーフアーカイブをブートします。	この手順を使用して、x86 システムでフェイルセーフアーカイブをブートします。次に、 <code>bootadm</code> コマンドを実行してブートアーカイブを更新します。	284 ページの「GRUB を使用して x86 システムでフェイルセーフアーカイブをブートする方法」
x86 ベースのフェイルセーフアーカイブをブートして、破壊されたブートアーカイブを強制的に更新します。	この手順は、ブートアーカイブが破壊されていて、システムが正常にブートすることを拒否する場合、または矛盾したブートアーカイブを更新するように要求されない場合に使用します。	286 ページの「x86: フェイルセーフアーカイブをブートして、破壊されたブートアーカイブを強制的に更新する方法」
GRUB を使用して x86 システムをネットワークからブートします。	この方法は、PXE または非 PXE デバイスをデフォルトのネットワーク構成方法でネットワークからブートする場合に使用します。この方法は、ディスクレスクライアントをブートする場合にも使用します。	292 ページの「x86: GRUB ベースのブートをネットワークから実行する方法」

▼ x86: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法

次の手順に従って、現時点で実行レベル 0 になっているシステムを実行レベル 3 でブートします。

1 システムを再起動します。

reboot

Press any key to reboot プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。

あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。

ブートシーケンスが始まると、GRUB メニューが表示されます。

- 2 GRUB メニューが表示されたら、**Enter** キーを押してデフォルトの OS インスタンスをブートします。

エントリを 10 秒以内に選択しない場合、システムは自動的に実行レベル 3 でブートします。

ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示されます。

- 3 システムにログインします。

```
hostname console login:
```

- 4 システムが実行レベル 3 になっていることを確認します。

```
# who -r
system% who -r
.          run-level 3  Mar  2 09:44      3      0  S
```

例 12-9 x86: システムを実行レベル 3 (マルチユーザーレベル) でブートする

```
# reboot
```

```
Jul 24 11:29:52 bearskin reboot: rebooted by root
syncing file systems... done
rebooting...
```

```
Adaptec AIC-7899 SCSI BIOS v2.57S4
(c) 2000 Adaptec, Inc. All Rights Reserved.
```

```
Press <Ctrl><A> for SCSISelect(TM) Utility!
```

```
Ch B, SCSI ID: 0 SEAGATE ST336607LSUN36G 160
```

```
GNU GRUB version 0.95 (637K lower / 2096064K upper memory)
```

```
=====
Solaris 10 10/08 s10x_u6wos_03 X86
Solaris failsafe
```

```
=====
Use the  and  keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

```
SunOS Release 5.10 Version Generic_137138-04 32-bit
Copyright 1983-2008 Sun Microsystems, Inc. All rights reserved.
```

```
Use is subject to license terms.
Hostname: pups
NIS domain name is ....sfbay.sun.com
Reading ZFS config: done.
Mounting ZFS filesystems: (5/5)

pups console login:

# who -r
.          run-level 3  Jul 24 11:31    3      0  S
```

▼ x86: システムを実行レベル S (シングルユーザーレベル) でブートする方法

次の手順に従って、実行レベル 0 になっているシステムを実行レベル S でブートします。シングルユーザーレベルは、システムを保守する場合に使用します。

注 - この手順は、すべての GRUB 実装で使用できます。ただし、GRUB メインメニューに表示されるブートエントリは、実行している Solaris リリースに応じて異なります。

ブート時に GRUB メニューに指定できるすべてのカーネルオプションについては、[237 ページの「x86: ブート時に GRUB メニューを編集してブート動作を変更する」](#)を参照してください。

- 1 システムを再起動します。

```
# reboot
```

Press any key to reboot プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。

あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。

ブートシーケンスが始まると、GRUB メニューが表示されます。

- 2 GRUB メインメニューが表示されたら、e と入力して GRUB メニューを編集します。
- 3 実行しているリリースによっては、矢印キーを使用して kernel または kernel\$ 行を選択します。
矢印キーを使用できない場合は、キャレットキー (^) を使って上方へスクロールしたり、英字キー v を使って下方へスクロールしたりします。

- 4 e と再度入力してブートエントリを編集します。
ここから、kernel 行または kernel\$ 行にオプションと引数を追加できます。
- 5 システムをシングルユーザーモードでブートするには、ブートエントリ行の終わりに -s と入力します。次に、**Return** キーを押して前の画面に戻ります。
 - 他のブート動作を指定するには、-s オプションを適切なブートオプションに置き換えます。
この方法で、次の代替ブート動作を指定できます。
 - 再構成用ブートを実行します。
 - 64 ビット対応のシステムを 32 ビットモードでブートします。
 - カーネルデバッグを使ってシステムをブートします。
 - コンソールをリダイレクトします。

詳細は、[boot\(1M\)](#) のマニュアルページを参照してください。
- 6 システムをシングルユーザーモードでブートするには、b と入力します。
- 7 プロンプトが表示されたら、ルートパスワードを入力します。
- 8 システムが実行レベル S になっていることを確認します。


```
# who -r
.          run-level S  Jun 13 11:07      S      0  0
```
- 9 実行レベルを S に移行して行う必要があるシステム保守作業を行います。
- 10 システムの保守作業が完了したら、システムをリブートします。

例 12-10 x86: システムをシングルユーザーモードでブートする

```
# reboot
Jul  2 14:30:01 pups reboot: initiated by root on /dev/console
syncing files...

Press <Ctrl><A> forPSCSISelect(TM) Utility!

GNU GRUB  version 0.95  (637K lower / 2096064K upper memory)

=====
Solaris 10 10/08 s10x_u6wos_03 X86
Solaris failsafe

=====

Use the  and  keys to select which entry is highlighted.
```

```

        Press enter to boot the selected OS, 'e' to edit the
        commands before booting, or 'c' for a command-line.
=====

GNU GRUB  version 0.95  (637K lower / 2096064K upper memory)

=====
findroot (pool_rpool,0,a)
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive
=====

        Use the  and  keys to select which entry is highlighted.
        Press 'b' to boot, 'e' to edit the selected command in the
        boot sequence, 'c' for a command-line, 'o' to open a new line
        after ('O' for before) the selected line, 'd' to remove the
        selected line, or escape to go back to the main menu.

[ Minimal BASH-like line editing is supported.  For the first word, TAB
lists possible command completions.  Anywhere else TAB lists the possible
completions of a device/filename.  ESC at any time exits. ]

grub edit> kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS -s

GNU GRUB  version 0.95  (637K lower / 2096064K upper memory)

=====
findroot (pool_rpool,0,a)
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS -s
module /platform/i86pc/boot_archive
=====

        Use the  and  keys to select which entry is highlighted.
        Press 'b' to boot, 'e' to edit the selected command in the
        boot sequence, 'c' for a command-line, 'o' to open a new line
        after ('O' for before) the selected line, 'd' to remove the
        selected line, or escape to go back to the main menu.

.
.
.
SunOS Release 5.10
Copyright 1983-2008 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
Booting to milestone "milestone/single-user:default".
Hostname: pups Requesting System Maintenance Mode SINGLE USER MODE
Root password for system maintenance (control-d to bypass):
single-user privilege assigned to /dev/console.
Entering System Maintenance Mode
Jul  2 14:41:48 su: 'su root' succeeded for root on /dev/console Sun Microsystems Inc.
# who -r
who -r      .          run-level S  Jul  2 14:39      S      0  0  #

```

▼ x86: システムを対話式でブートする方法

代替のカーネルまたは代替の `/etc/system` ファイルを指定する必要がある場合は、次の手順に従ってシステムをブートします。

始める前に `boot -a` コマンドを使用して x86 システムを対話式でブートするときに、代替の `/etc/system` ファイルを指定するには、最初に次の手順を実行する必要があります。

- 1. `/etc/system` ファイルと `boot/solaris/filelist.ramdisk` ファイルのバックアップコピーを作成します。

```
# cp /etc/system /etc/system.bak
# cp /boot/solaris/filelist.ramdisk /boot/solaris/filelist.ramdisk.orig
```

- 2. `/etc/system.bak` ファイル名を `/boot/solaris/filelist.ramdisk` ファイルに追加します。

```
# echo "etc/system.bak" >> /boot/solaris/filelist.ramdisk
```

- ブートアーカイブを更新します。

```
# bootadm update-archive -v
```

- 1 システムを再起動します。

```
# reboot
```

Press any key to reboot プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。

あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。

ブートシーケンスが始まると、GRUB メインメニューが表示されます。

- 2 GRUB 編集メニューにアクセスするには、`e` と入力します。
- 3 矢印キーを使用して、`kernel` 行または `kernel$` 行を選択します。
- 4 `e` と入力してブートエントリ行を編集します。
- 5 `-a` と入力して、システムを対話式でブートします。次に、**Enter** キーを押して GRUB メインメニューに戻ります。
- 6 システムを対話式でブートするために、`b` と入力します。

- 7 モジュールのデフォルトディレクトリを入力するか、**Enter** キーを押してデフォルトを受け入れます。

Enter default directory for modules [/platform/i86pc/kernel /kernel /usr/kernel]:

- 8 代替システムファイル名 *alternate-file* を入力します。

Name of system file [etc/system]: **/etc/system.bak**

代替ファイルを入力しないで Enter キーを押すと、デフォルトが受け入れられます。

破損した /etc/system ファイルを修復します。

- 9 システムを実行レベル 3 でリブートします。

例 12-11 x86: システムを対話式でブートする

```
# reboot
```

```
syncing file systems... done
```

```
rebooting...
```

```
GNU GRUB version 0.95 (637K lower / 2096064K upper memory)
```

```
=====
```

```
Solaris 10 10/08 s10x_u6wos_03 X86
```

```
Solaris failsafe
```

```
=====
```

```
Use the and keys to select which entry is highlighted.
```

```
Press enter to boot the selected OS, 'e' to edit the
```

```
commands before booting, or 'c' for a command-line.
```

```
=====
```

```
GNU GRUB version 0.95 (637K lower / 2096064K upper memory)
```

```
=====
```

```
findroot (pool_rpool,0,a)
```

```
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
```

```
module /platform/i86pc/boot_archive
```

```
=====
```

```
Use the and keys to select which entry is highlighted.
```

```
Press 'b' to boot, 'e' to edit the selected command in the
```

```
boot sequence, 'c' for a command-line, 'o' to open a new line
```

```
after ('O' for before) the selected line, 'd' to remove the
```

```
selected line, or escape to go back to the main menu.
```

```
[ Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename. ESC at any time exits. ]
```

```
grub edit> kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS -a
```

```

GNU GRUB  version 0.95  (637K lower / 2096064K upper memory)

=====
findroot (pool_rpool,0,a)
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS -a
module /platform/i86pc/boot_archive
=====
.
.
.
Enter default directory for modules [/platform/i86pc/kernel /kernel /usr/kernel]:
Name of system file [/etc/system]:  /etc/system.bak
SunOS Release 5.10 Version Generic_137138-04 32-bit
Copyright 1983-2008 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
Hostname: pups
NIS domain name is ....sfbay.sun.com
Reading ZFS config: done.
Mounting ZFS filesystems: (5/5)
pups console login:

```

x86 システムで **ZFS** ルートファイルシステムから起動する

x86 プラットフォームで ZFS ルートファイルシステムのブートをサポートするために、新しい GRUB キーワード `$ZFS-BOOTFS` が導入されました。ルートデバイスに ZFS プールが含まれている場合は、このキーワードに値が割り当てられ、その値が `-B` オプションを使ってカーネルに渡されて、ブートするデータセットが識別されます。ZFS ブートローダーをサポートする Solaris リリースをシステムにインストールするか、そのリリースにアップグレードする場合は、`GRUB menu.lst` ファイルおよび GRUB ブートメニューにデフォルトでこの情報が含まれます。

▼ x86 システム上の使用可能な **ZFS** ブート環境のリストを表示する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
- 2 システム上の使用可能な **BE** のリストを表示するには、次のコマンドを入力します。

```
# lustatus
```

`lustatus` コマンドは、SPARC システムでも使用できます。

注-lustatus コマンドの実行時に次のエラーが表示される場合は、インストールを新たに実行したときに Solaris Live Upgrade が使用されなかったことを示しています。BE が lustatus の出力で認識されるようにするには、最初にシステムで BE を新しく作成する必要があります。

```
# lustatus
ERROR: No boot environments are configured on this system
ERROR: cannot determine list of all boot environment names
```

Solaris Live Upgrade を使用して UFS ルートファイルシステムを ZFS ルートファイルシステムに移行する方法の詳細は、『Solaris ZFS 管理ガイド』の「UFS ルートファイルシステムから ZFS ルートファイルシステムへの移行 (Solaris Live Upgrade)」を参照してください。

例 12-12 lustatus コマンドを使って使用可能な ZFS ブート可能データセットのリストを表示する

次の例では、lustatus コマンドの出力は 3 つの ZFS ブート可能データセットの状態を示しています。デフォルトのブート環境は be1 であるため、これを削除することはできません。

```
# lustatus
Boot Environment      Is      Active Active      Can      Copy
Name                  Complete Now    On Reboot Delete Status
-----
sl0s_nbu6wos          yes      no     no           yes      -
zfs2BE                yes      yes    yes          no       -
zfsbe3                no       no     no           yes      -
#
```

BE が作成され、ブート可能になっている場合は、Is Complete 列に「yes」が表示されます。BE が作成され、まだアクティブになっていない場合は、この列に「no」が表示されます。BE をアクティブにするには、luactivate コマンドを使用します。そのあとで lustatus コマンドを実行して、BE が正常にアクティブ化されたことを確認します。

詳細は、lustatus(1M) および luactivate(1m) のマニュアルページを参照してください。

▼ x86 システムで ZFS ルートファイルシステムからブートする方法

次の手順では、ZFS ブートローダーをサポートする x86 システムで ZFS ルートファイルシステムからブートする方法について説明します。

ZFS ブートローダーをサポートする Solaris リリースをシステムにインストールするか、そのリリースにアップグレードする場合は、GRUB メニューエントリに、デフォルトで `-B $ZFS-BOOTFS` ブート引数が含まれます。そのため、追加のブート引数を指定しなくてもシステムは ZFS からブートします。

1 システムを再起動します。

reboot

Press any key to reboot プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。

あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。

ブートシーケンスが始まると、GRUB メインメニューが表示されます。デフォルトのブートエントリが ZFS ファイルシステムの場合は、次のようなメニューが表示されます。

```
GNU GRUB version 0.95 (637K lower / 3144640K upper memory)
+-----+
| be1
| be1 failsafe
| be3
| be3 failsafe
| be2
| be2 failfafa
+-----+
      Use the ^ and v keys to select which entry is highlighted.
      Press enter to boot the selected OS, 'e' to edit the
      commands before booting, or 'c' for a command-line.
```

2 GRUB メニューが表示されたら、Enter キーを押してデフォルトの OS インスタンスをブートします。

エントリを 10 秒以内に選択しない場合、システムは自動的に実行レベル 3 でブートします。

3 別の BE をブートするには、矢印キーを使用して指定のブートエントリを強調表示します。

4 このエントリをブートする場合は b と入力し、このエントリを編集する場合は e と入力します。

e と入力してエントリを編集する場合は、ZFS ルートを含むシステムをブートするためのデフォルトのメニューが次のように表示されます。

```
findroot (BE_be10,0,a)
kernel$ /platform/i86pc/kernel/$ISADIR/unix -B $ZFS-BOOTFS
module$ /platform/i86pc/$ISADIR/boot-archive
```

ブート時の GRUB メニューエントリの詳細は、240 ページの「[x86: ブート時に GRUB メニューを編集してブート動作を変更する方法](#)」を参照してください。

例 12-13 x86:x86 システムで新しいブート環境をアクティブにする

次の例は、システム上でブート環境 `be10` をアクティブにするために従う手順を示しています。最初に `lustatus` コマンドを実行して、システム上のどの BE がアクティブになっているか、またどの BE をアクティブにする必要があるかを確認します。

```
# lustatus
Boot Environment      Is      Active Active      Can      Copy
Name                  Complete Now    On Reboot Delete Status
-----
be1                    yes      yes    yes        no
be10                   yes      no     no         yes

# luactivate be10
System has findroot enabled GRUB Generating boot-sign, partition and slice
information for PBE <be1>
WARNING: The following file s have change on both the current boot environment
<be1> zone <global> and the boot environment to be activitate <be10>
        /etc/zfs/zpool.cache
INFORMATION: The files listed above are in conflict between the current
boot environment <be1> zone <global> and the boot environment to be
activated <be10>. These files will not be automatically synchronized from
the current boot environment <be1> when boot environment <be10> is activated.

Setting failsafe console to <ttyb>
Generating boot-sign for ABE <be10>
Generating partition and slice information for ABE <be10>
Copied boot menu from top level dataset.
Generating direct boot menu entries for PBE.
Generating direct boot menu entries for ABE.
Disabling splashimage
Current GRUB menu default setting is not valid
title Solaris bootenv rc
No more bootadm entries. Deletion of bootadm entries is complete.
GRUB menu default setting is unchanged
Done eliding bootadm entries.
*****
The target boot environment has been activated. It will be used when you
reboot. NOTE: You MUST NOT USE the reboot, halt, or uadmin commands. You
MUST USE either the init or the shutdown command when you reboot. If you
do not use either init or shutdown, the system will not boot using the
target BE.
```

```

*****

', '

# reboot
May 30 09:52:32 pups reboot: initiated by root on /dev/console
syncing file systems... done
rebooting...

CE SDRAM BIOS P/N GR-xlint.007-4.330
*

BIOS Lan-Console 2.0
Copyright (C) 1999-2001 Intel Corporation
.
.
.
GNU GRUB version 0.95 (637K lower / 3144640K upper memory)
+-----+
| be1
| be1 failsafe
| be10
| be10 failsafe
+-----+
      Use the ^ and v keys to select which entry is highlighted.
      Press enter to boot the selected OS, 'e' to edit the
      commands before booting, or 'c' for a command-line.

SunOS Release 5.10 32-bit
Copyright 1983-2008 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.

Hostname: pups
NIS domain name is sunsoft.eng.sun.com
Reading ZFS config: done.
Mounting ZFS filesystems: (8/8)

pups console login:
# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                  Complete Now   On Reboot Delete Status
-----
be1                    yes     yes   yes     no
be10                   yes     yes   yes     no
#

```

x86 システムでフェイルセーフアーカイブをブートする

x86 システムでフェイルセーフアーカイブをブートするには、システムのブート中に GRUB メニューが表示されたら、フェイルセーフブートエントリを選択します。フェイルセーフブートの手順で、一次ブートアーカイブを更新するかどうかを確認されたら、y と入力します。

フェイルセーフブートは、ZFS からブートされるシステムでもサポートされます。UPS をルートとした BE からブートするときは、各 BE に独自のフェイルセーフアーカイブが用意されています。フェイルセーフアーカイブは、ZFS をルートとした BE と同様に、ルートファイルシステムが置かれている場所にあります。x86 システムでは、プール全体の GRUB メニューに各フェイルセーフアーカイブのエントリが 1 つあります。デフォルトのフェイルセーフアーカイブは、デフォルトのブート可能ファイルシステムにあるアーカイブです。デフォルトのブート可能ファイルシステム(データセット)は、プールの `bootfs` プロパティの値で示されます。

ブートアーカイブの更新に使用できるもう一つの方法は、`boot-archive` サービスをクリアすることです。310 ページの「[boot-archive サービスをクリアして矛盾したブートアーカイブを更新する方法](#)」を参照してください。ただし、ブートアーカイブを更新する場合は、フェイルセーフアーカイブをブートするか、`bootadm` コマンドを使用することをお勧めします。詳細は、第 14 章「[Solaris ブートアーカイブの管理\(手順\)](#)」を参照してください。

▼ GRUB を使用して x86 システムでフェイルセーフアーカイブをブートする方法

注 - 一部の Solaris では、GRUB フェイルセーフの対話操作により、矛盾したブートアーカイブが検出されるかどうかに関係なく、ブートアーカイブを更新するように要求されます。この Solaris リリースでは、矛盾したブートアーカイブが検出された場合にだけ、ブートアーカイブの更新を要求されます。

- 1 302 ページの「[x86: 復旧を目的としてシステムを停止する方法](#)」の手順で説明されている方法のいずれかを使用してシステムを停止します。

- 2 Press any key to reboot プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。

あるいは、リセットボタンを使用することもできます。あるいは、電源スイッチを使用してシステムをリブートします。

ブートシーケンスが始まると、GRUB メニューが表示されます。

```
GNU GRUB  version 0.95  (637K lower / 3144640K upper memory)
+-----+
| be1
```

```
| be1 failsafe
| be3
| be3 failsafe
| be2
| be2 failfafa
+-----+
      Use the ^ and v keys to select which entry is highlighted.
      Press enter to boot the selected OS, 'e' to edit the
      commands before booting, or 'c' for a command-line.
```

注 - 表示される GRUB メニューは、実行している Solaris リリースに応じて異なる場合があります。

- 3 矢印キーを使用して **GRUB** メニューをナビゲートし、**failsafe** エントリを選択します。

- 4 **Return** キーを押してフェイルセーフアーカイブをブートします。

インストール済みの OS インスタンスが検索されます。矛盾したブートアーカイブが検出されると、次のようなメッセージが表示されます。

```
Searching for installed OS instances...
```

```
An out of sync boot archive was detected on /dev/dsk/c0t0d0s0.
The boot archive is a cache of files used during boot and
should be kept in sync to ensure proper system operation.
```

```
Do you wish to automatically update this boot archive? [y,n,?]
```

- 5 y と入力して、ブートアーカイブを更新します。

矛盾したブートアーカイブが複数検出された場合は、矛盾したブートアーカイブごとに y と入力してアーカイブを更新するように要求されます。

アーカイブが正常に更新されるごとに、次のメッセージが表示されます。

```
Updating boot archive on /dev/dsk/c0t0d0s0.
```

```
The boot archive on /dev/dsk/c0t0d0s0 was updated successfully.
```

ブートアーカイブの更新が完了すると、インストール済みのすべての OS インスタンスが再度検索され、/a にマウントするデバイスの選択を要求されます。矛盾したブートアーカイブが検出されない場合にシステムが最初にブートすると、これと同じメッセージが表示されます。

```
Searching for installed OS instances...
```

```
Multiple OS instances were found. To check and mount one of them
read-write under /a, select it from the following list. To not mount
any, select 'q'.
```

```
1 pool10:13292304648356142148      ROOT/be10
```

```
2  rpool:14465159259155950256      ROOT/be01
```

Please select a device to be mounted (q for none) [?,??,q]:

- デバイスをマウントしないことを選択する場合は、q と入力してブートプロセスを続行します。
- デバイスをマウントすることを選択する場合は、次の手順を実行します。
 - a. デバイスの番号を入力して **Return** キーを押します。
デバイスが /a にマウントされ、シェルプロンプトに戻ります。
 - b. クリティカルなシステムリソースを修復します。
 - c. クリティカルなシステムリソースの修復が完了したら、デバイスのマウントを解除します。

```
# umount /a
```
 - d. システムを再起動します。

```
# reboot
```

▼ **x86:** フェイルセーフアーカイブをブートして、破壊されたブートアーカイブを強制的に更新する方法

次の手順では、ブートアーカイブの更新を要求されない場合、またはシステムのハングアップやループシーケンスが発生した場合に、矛盾したブートアーカイブまたは破壊されたブートアーカイブを再構築する方法を示します。

- 1 **302 ページの「x86: 復旧を目的としてシステムを停止する方法」**の手順で説明されている方法のいずれかを使用してシステムを停止します。
- 2 システムを再起動します。

```
# reboot
```

Press any key to reboot プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。

あるいは、リセットボタンを使用することもできます。

ブートシーケンスが始まると、GRUB メニューが表示されます。

```
+-----+
| Solaris 10.1... X86                               |
| Solaris failsafe                                   |
|                                                     |
|                                                     |
+-----+
Use the  and  keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

注-GRUB メニューの内容は、実行している Solaris リリースに応じて異なります。

- 3 矢印キーを使用して **GRUB** メニューをナビゲートし、**failsafe** エントリを選択します。**Return** キーを押してフェイルセーフアーカイブをブートします。
期限切れのブートアーカイブが存在する場合は、次のようなメッセージが表示されます。

```
Searching for installed OS instances...
```

```
An out of sync boot archive was detected on /dev/dsk/c0t0d0s0.
The boot archive is a cache of files used during boot and
should be kept in sync to ensure proper system operation.
```

```
Do you wish to automatically update this boot archive? [y,n,?]
```

- 4 y と入力してから **Enter** キーを押して、矛盾したブートアーカイブを更新します。
次のメッセージが表示されます。

```
Updating boot archive on /dev/dsk/c0t0d0s0.
```

```
The boot archive on /dev/dsk/c0t0d0s0 was updated successfully.
```

矛盾したブートアーカイブが見つからない場合は、次のようなメッセージが表示されます。

```
Searching for installed OS instances...
```

```
Solaris 10.1... X86 was found on /dev/dsk/c0t0d0s0.
Do you wish to have it mounted read-write on /a? [y,n,?]
```

このメッセージは、矛盾したブートアーカイブの更新に成功した場合にも表示されます。

- 5 ブートアーカイブが破壊されているデバイスを /a にマウントします。デバイスに対応する番号を入力してから、**Enter** キーを押します。

注- 前の手順で矛盾したアーカイブを更新した場合は、デバイスはすでに /a にマウントされています。手順6に進みます。

- 6 破壊されたブートアーカイブを強制的に更新する場合は、次のように入力します。

```
# bootadm update-archive -f -R /a
```

- 7 デバイスをマウント解除します。

```
# umount /a
```

- 8 システムを再起動します。

```
# reboot
```

例 12-14 x86: フェイルセーフアーカイブをブートして、破壊されたブートアーカイブを強制的に更新する

次の例では、フェイルセーフアーカイブをブートして、破壊されたブートアーカイブを強制的に更新する方法を示します。

```
GNU GRUB version 0.95 (635K lower / 523200K upper memory)
```

```
+-----+
| Solaris 10 1/06 s10x_u1wos_19a X86 |
| >Solaris failsafe< |
| |
| |
+-----+
```

```
Use the and keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

```
SunOS Release 5.10
```

```
Copyright 1983-2006 Sun Microsystems, Inc. All rights reserved.
```

```
Use is subject to license terms.
```

```
Booting to milestone "milestone/single-user:default".
```

```
Configuring devices.
```

```
Searching for installed OS instances...
```

```
Multiple OS instances were found. To check and mount one of them
read-write under /a, select it from the following list. To not mount
any, select 'q'.
```

```
1 /dev/dsk/c0t0d0s0 Solaris 10 1/06 s10x_u1wos_19a X86
2 /dev/dsk/c0t1d0s0 Solaris 10 5/08 X86
```



```

Please select a device to be mounted (q for none) [?,??,q]: 1
mounting /dev/dsk/c0t0d0s0 on /a

Starting shell.
# rm /a/platform/i86pc/boot_archive
# bootadm update-archive -f -R /a
Creating boot_archive for /a
updating /a/platform/i86pc/amd64/boot_archive
updating /a/platform/i86pc/boot_archive
# umount /a
# reboot
syncing file systems... done
rebooting...
.
.
.

```

x86 システムをネットワークからブートする

この節では、GRUB ベースのブートをネットワークから実行するときの要件と注意事項について説明します。

ブートサーバーが利用できれば、どのようなシステムもネットワークからブートできます。スタンドアロンのシステムがローカルディスクからブートできない場合、そのシステムを復旧目的でネットワークからブートしなければならないことがあります。Solaris OS x86 システムは、PXE ネットワークブートプロトコルをサポートするネットワークから直接ブートすることができます。

注 - PXE ネットワークブートは、Intel の PXE 仕様を実装しているデバイスでのみ動作します。

GRUB ベースで PXE ネットワークブートを実行するときには、デフォルトでは DHCP が使用されます。PXE 以外のデバイスの場合には、DHCP または RARP を使用してブートできます。使用するブート方法は、ネットワークで利用できるブートサーバーの種類によって異なります。PXE または DHCP サーバーを使用できない場合は、GRUB をフロッピーディスク、CD-ROM、またはローカルディスクから読み込むことができます。

GRUB ベースのネットワークブートを実行するには、PXE クライアント用に構成されている DHCP サーバーが必要です。また、tftp サービスを提供するブートサーバーも必要です。クライアントのネットワークインタフェースの構成に必要な情報は、DHCP サーバーから提供されます。

DHCP サーバーは、DHCP クラス PXEClient および GRUBClient に次の情報を返せるように設定されている必要があります。

- ファイルサーバーの IP アドレス
- ブートファイルの名前 (pxegrub)

Solaris OS の PXE ネットワークブートは、次の順序で実行されます。

1. BIOS が、ネットワークインタフェースからブートするように構成されます。
2. BIOS が DHCP 要求を送信します。
3. DHCP サーバーが、サーバーのアドレスとブートファイルの名前を返します。
4. BIOS が、tftp を使用して pxegrub をダウンロードし、pxegrub を実行します。
5. tftp を使用して GRUB メニューファイルがダウンロードされます。
このファイルによって、利用可能なブートメニューエントリが表示されます。
6. メニューエントリを選択すると、Solaris OS のロードが開始されます。

詳細は、『[Solaris のシステム管理 \(IP サービス\)](#)』の「[ネットワーク構成サーバーの設定方法](#)」を参照してください。

add_install_client コマンドを実行すると、/tftpboot_01ethernet-address ファイルが作成されます。このファイルは、pxegrub および /tftpboot/menu.lst.01ethernet-address ファイルにリンクされています。/tftpboot/menu.lst.01 ethernet-address ファイルは、GRUB メニューファイルです。このファイルが存在しないときには、pxegrub は、DHCP オプション 150 が指定されている場合はそのオプションを使用する状態に戻り、指定されていない場合は /tftpboot/boot/grub/menu.lst ファイルに戻ります。通常は、1 つのシステムで両方の機能を提供できるように設定します。このようなシステムでは、add_install_client コマンドを実行すると、/tftpboot ファイルと一緒に、適切に設定された pxegrub メニューファイルと Solaris ファイルが設定されます。DHCP サービスは、add_install_client コマンドを使用して、別の手順として設定します。設定は、クライアントごとに 1 回行うだけで完了します。詳細は、[290 ページ](#)の「[x86: DHCP マクロについて](#)」および [292 ページ](#)の「[x86: GRUB ベースのブートをネットワークから実行する方法](#)」を参照してください。

x86: DHCP マクロについて

add_install_client -d スクリプトを使用してインストールサーバーにクライアントを追加すると、このスクリプトから DHCP 設定情報が標準出力に報告されます。この情報は、ネットワークインストール情報をクライアントに伝えるために必要なオプションとマクロを作成する際に使用できます。

DHCP サーバーを使用して DHCP クライアントをネットワークインストールする場合は、DHCP オプションを作成する必要があります。この情報は、Solaris OS をインストールするときに必要になります。

クライアントから DHCP 要求を送信するときには、サーバーに次のようなクライアント情報を渡す必要があります。

- クライアントの ID。通常は Ethernet アドレス
- クライアント要求のクラス
- クライアントが存在するサブネット

Solaris DHCP サーバーは、応答を作成します。この応答は、クライアント要求に対応する次の「マクロ」に基づいて作成されます。

クラスマクロ	クラスマクロは、DHCP 要求に含まれる「クラス文字列」に基づいて作成されます。x86 システムでは、クラス PXEClient:Arch:00000:UNDI:002001 を含む DHCP 要求が、BIOS によってすでに作成されています。この名前のマクロが DHCP サーバー構成に定義されている場合には、そのマクロの内容が x86 クライアントに送信されます。
ネットワークマクロ	ネットワークマクロの名前は、クライアントが存在するサブネットの IP アドレスになっています。マクロ 129.146.87.0 が DHCP サーバーに定義されている場合には、そのマクロの内容がそのサブネット上のすべてのクライアントに送信されます。送信されるマクロの内容は、DHCP 要求に含まれるクラスに関係ありません。クラスマクロとネットワークマクロの両方に DHCP オプションが定義されている場合は、ネットワークマクロが優先されます。
IP マクロ	IP マクロの名前は、IP アドレスになっています。このマクロはほとんど使用されません。
クライアントマクロ	クライアントマクロの名前は、クライアントの種類 (Ethernet の場合は 01) およびクライアントの mac アドレスに基づいて、大文字の名前が付けられます。Ethernet アドレスが 0:0:39:fc:f2:ef のクライアントの場合には、対応するマクロの名前は 01000039FCEF になります。クライアントマクロにはコロンが付きません。

たとえば、サブネット 192.168.100.0 に存在し、Ethernet アドレスとして 0:0:39:fc:f2:ef が割り当てられ、クラス PXEClient の DHCP 要求を作成するクライアントの場合には、DHCP サーバー上に次のような対応するマクロが存在します。

```
PXEClient
  BootSrvA: 192.168.100.0
  BootFile: pxegrub
129.146.87.0
  Router: 129.146.87.1
  NISdmain: sunsoft.eng.sun.com
01000039FCEF
```

```
BootFile: 01000039FCEF
The actual DHCP response will be
BootSrvA: 192.168.100.0
BootFile: 01000039FCEF
Router: 129.146.87.1
NISdomain: sunsoft.eng.sun.com
```

クライアントマクロの BootFile は、クラスマクロの BootFile より優先されます。

詳細については、『Solaris 10 インストールガイド(ネットワークインストール)』の「DHCP サービスによるシステム構成情報の事前設定(作業)」を参照してください。

▼ x86: GRUB ベースのブートをネットワークから実行する方法

GRUB ベースのネットワークブートを実行するには、PXE クライアント用に構成されている DHCP サーバーが必要です。また、tftp サービスを提供するブートサーバーも必要です。この DHCP サーバーは、ファイルサーバーの IP アドレスとブートファイル(pxegrub)を取得するために、DHCP クラスの PXEClient と GRUBClient に応答できるように設定されている必要があります。デフォルトでは、メニューファイルは /tftpboot/menu.lst.01ethernet-address です。このファイルが存在しないときには、pxegrub は、DHCP オプション 150 が指定されている場合はそのオプションに戻り、指定されていない場合は /tftpboot/boot/grub/menu.lst ファイルに戻ります。

システムを Solaris SOFTWARE 1 CD または DVD からブートする場合、システムは自動的にブートします。

始める前に GRUB を使用して x86 システム上でネットワークブートを実行する前に、次の準備を行います。

- システムをネットワークからブートするために必要なコマンドをインストールサーバー上で実行します。
- クライアントシステムをインストールクライアントとして追加します。

詳細は、『Solaris 10 インストールガイド(ネットワークインストール)』の第7章「ネットワークインストールの準備(概要)」を参照してください。

- 1 DHCP サーバー上で、次の 2 つのオプションを使用して、DHCP サービスのクライアントマクロを作成します。

- BootSrvA: svr-addr

- **BootFile:** *client-macro*

dhtadm コマンドを実行するには、DHCP サーバー上のスーパーユーザー特権が必要です。

svr-addr はサーバーの IP アドレスです。*client-macro* は、クライアントの Ethernet タイプ (01) とクライアントの *mac* アドレスに基づいて作成された、大文字の名前です。この番号は、インストールサーバーの */tftpboot* ディレクトリで使用されるファイルの名前としても使われます。

注 - *client-macro* には、コロンを含めてはいけません。

クライアントマクロは、DHCP GUI またはコマンド行インタフェースから作成できます。

クライアントマクロをコマンド行から作成するには、次のように入力します。

```
# dhtadm -[MA] -m client macro -d
":BootFile=client-macro:BootSrvA=svr-addr:"
```

- 2 システムを再起動します。
- 3 ネットワークからブートするように **BIOS** に指示します。
 - 特定のキーストロークシーケンスを使用してネットワークブートするように設定されている場合は、BIOS 画面が表示されたときにそのキーストロークを入力します。
 - ネットワークブートのために BIOS 設定を手動で変更する必要がある場合は、BIOS 設定ユーティリティにアクセスするためのキーストロークシーケンスを入力します。その画面で、ネットワークブートのブート優先順位を変更します。
- 4 **GRUB** メニューが表示されたら、インストールするネットワークインストールイメージを選択します。

Solaris システムのブートのトラブルシューティング (手順)

この章では、SPARC システムおよび x86 システムで Solaris リリースをブートするための手順について説明します。

この章の内容は次のとおりです。

- 295 ページの「SPARC プラットフォームのブートのトラブルシューティング (作業マップ)」
- 302 ページの「x86 プラットフォームのブートのトラブルシューティング (作業マップ)」

SPARC プラットフォームのブートのトラブルシューティング (作業マップ)

作業	説明	参照先
復旧を目的としてシステムを停止します。	破損したファイルによってシステムのブートが正常に行えない場合は、最初にシステムを停止して復旧を試みます。	296 ページの「SPARC: 復旧を目的としてシステムを停止する方法」
強制的にシステムのクラッシュダンプを実行し、システムをリポートします。	トラブルシューティング手段としてシステムのクラッシュダンプとリポートを強制的に実行できます。	297 ページの「SPARC: クラッシュダンプを強制してシステムをリポートする方法」
復旧を目的として SPARC システムをブートします。	システムをブートして、システムの正常なブートを妨げている重要なシステムファイルを修復します。	299 ページの「SPARC: 復旧を目的としてシステムをブートする方法」

作業	説明	参照先
カーネルデバッグを使ってシステムをブートします。	カーネルデバッグを使ってシステムをブートして、ブート問題のトラブルシューティングを実行できます。システムのブートには <code>kmdb</code> コマンドを使用します。	301 ページの「SPARC: カーネルデバッグ (kmdb) を使ってシステムをブートする方法」

システムの正常なブートを妨げている問題のトラブルシューティングには、次の方法が 1 つ以上必要になることがあります。

- システムのブート時にエラーメッセージのトラブルシューティングを行います。
- システムを停止して復旧を試みます。
- 復旧を目的としてシステムをブートします。
- 強制的にクラッシュダンプを実行し、システムをリブートします。
- `kmdb` コマンドを使用してカーネルデバッグでシステムをブートします。

▼ SPARC: 復旧を目的としてシステムを停止する方法

- 1 システムのアボートキーシーケンスを入力します。
モニターに、`ok PROM` プロンプトが表示されます。
`ok`
アボートキーシーケンスは、キーボードのタイプによって異なります。たとえば、`Stop+A` または `L1+A` キーを押します。端末では、`Break` キーを押します。
- 2 ファイルシステムを同期させます。
`ok sync`
- 3 `syncing file systems...` というメッセージが表示されたら、アボートキーシーケンスをもう一度入力します。
- 4 該当する `boot` コマンドを入力して、ブートプロセスを起動します。
詳細は、[boot\(1M\)](#) のマニュアルページを参照してください。
- 5 システムが指定した実行レベルになっていることを確認します。

```
# who -r
.          run-level s  May  2 07:39    3      0  S
```

例 13-1 SPARC: 復旧を目的としてシステムを停止する

```
Press Stop-A
ok sync
syncing file systems...
```



```
Press Stop-A
ok boot
```

SPARC: クラッシュダンプを強制してシステムをリブートする

場合によっては、トラブルシューティングの目的でクラッシュダンプを強制実行し、システムをリブートする必要があります。デフォルトでは、`savecore` 機能を使用できます。

システムのクラッシュダンプについては、『[Solaris のシステム管理 \(上級編\)](#)』の第 17 章「[システムクラッシュ情報の管理 \(手順\)](#)」を参照してください。

▼ SPARC: クラッシュダンプを強制してシステムをリブートする方法

システムのクラッシュダンプを強制実行するには、次の手順を実行します。この手順のあとに紹介する例は、`halt -d` コマンドを使用してシステムのクラッシュダンプを強制実行する方法を示しています。このコマンドの実行後は、手動でシステムをリブートする必要があります。

- 1 システムのアボートキーシーケンスを入力します。
アボートキーシーケンスは、キーボードのタイプによって異なります。たとえば、`Stop + A` または `L1 + A` キーを押します。端末では、`Break` キーを押します。
`ok PROM` プロンプトが表示されます。
- 2 ファイルシステムを同期させ、クラッシュダンプを書き出します。

```
> n
ok sync
```


クラッシュダンプがディスクに書き出されると、システムはそのままりブートします。
- 3 システムが実行レベル 3 になっていることを確認します。
ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示されます。
`hostname console login:`

例 13-2 SPARC: `halt -d` コマンドを使用してシステムのクラッシュダンプとリブートを強制実行する

次の例は、`halt -d` と `boot` の各コマンドを使用して、システム `jupiter` のクラッシュダンプとリブートを強制実行する方法を示しています。システムのクラッシュダンプとリブートを強制実行する場合は、この方法を使用してください。

```
# halt -d
Jul 21 14:13:37 jupiter halt: halted by root

panic[cpu0]/thread=30001193b20: forced crash dump initiated at user request

000002a1008f7860 genunix:kadmin+438 (b4, 0, 0, 0, 5, 0)
  %l0-3: 0000000000000000 0000000000000000 0000000000000004 0000000000000004
  %l4-7: 000000000000003cc 0000000000000010 0000000000000004 0000000000000004
000002a1008f7920 genunix:uadmin+110 (5, 0, 0, 6d7000, ff00, 4)
  %l0-3: 0000030002216938 0000000000000000 0000000000000001 0000004237922872
  %l4-7: 000000423791e770 0000000000004102 0000030000449308 0000000000000005

syncing file systems... 1 1 done
dumping to /dev/dsk/c0t0d0s1, offset 107413504, content: kernel
100% done: 5339 pages dumped, compression ratio 2.68, dump succeeded
Program terminated
ok boot
Resetting ...

Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #10933339.
Ethernet address 8:0:20:a6:d4:5b, Host ID: 80a6d45b.

Rebooting with command: boot
Boot device: /pci@1f,0/pci@1,1/ide@3/disk@0,0:a
File and args: kernel/sparcv9/unix
SunOS Release 5.10 Version s10_60 64-bit
Copyright 1983-2004 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
configuring IPv4 interfaces: hme0.
add net default: gateway 172.20.27.248
Hostname: jupiter
The system is coming up. Please wait.
NIS domain name is example.com
.
.
.
System dump time: Wed Jul 21 14:13:41 2004
Jul 21 14:15:23 jupiter savecore: saving system crash dump
in /var/crash/jupiter/*.0
Constructing namelist /var/crash/jupiter/unix.0
Constructing corefile /var/crash/jupiter/vmcore.0
100% done: 5339 of 5339 pages saved

Starting Sun(TM) Web Console Version 2.1-dev...
.
.
.
```

▼ SPARC: 復旧を目的としてシステムをブートする方法

次の手順は、`/etc/passwd` などの重要なファイルに、無効なエントリがあり正常にブートできない場合に使用します。

`root` パスワードがわからない場合やシステムにログインできない場合は、この手順で示すアポートキーシーケンスを使用します。詳細は、[296 ページの「SPARC: 復旧を目的としてシステムを停止する方法」](#)を参照してください。

次の手順で使用されている変数 `device-name` は、修復するファイルシステムのデバイス名に置き換えてください。システムのデバイス名を調べる場合は、『[Solaris のシステム管理 \(デバイスとファイルシステム\)](#)』の「[デバイス構成情報の表示](#)」を参照してください。

- 1 システムのアポートキーシーケンスを使用して、システムを停止します。
- 2 システムをシングルユーザーモードでブートします。
 - Solaris SOFTWARE - 1 CD または DVD からシステムをブートします。
 - Solaris SOFTWARE - 1 CD または DVD をドライブに挿入します。
 - Solaris SOFTWARE - 1 CD または DVD から、シングルユーザーモードでブートします。

```
ok boot cdrom -s
```

- ネットワーク (インストールサーバーまたはリモート CD/DVD ドライブが使用できない場合) からシステムをブートします。

```
ok boot net -s
```

- 3 ファイル内に無効なエントリがあるファイルシステムをマウントします。

```
# mount /dev/dsk/device-name /a
```

- 4 新しくマウントしたファイルシステムに移動します。

```
# cd /a/file-system
```

- 5 端末タイプを設定します。

```
# TERM=sun
```

```
# export TERM
```

- 6 エディタを使って、ファイルから無効なエントリを削除します。

```
# vi filename
```

- 7 ルート (/) ディレクトリに変更します。

```
# cd /
```

- 8 /a ディレクトリのマウントを解除します。

```
# umount /a
```

- 9 システムを再起動します。

```
# init 6
```

- 10 システムが実行レベル 3 になっていることを確認します。

ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示されます。

```
hostname console login:
```

例 13-3 SPARC: 復旧を目的としてシステムをブートする (パスワードファイルが損傷した場合)

次の例は、ローカルの CD-ROM からブートしたあとで重要なシステムファイル (この場合は /etc/passwd) を修復する方法を示しています。

```
ok boot cdrom -s
# mount /dev/dsk/c0t3d0s0 /a
# cd /a/etc
# TERM=vt100
# export TERM
# vi passwd
    (Remove invalid entry)
# cd /
# umount /a
# init 6
```

例 13-4 SPARC: root パスワードを忘れた場合にシステムをブートする

次の例は、root パスワードを忘れた場合にネットワークからシステムをブートする方法を示しています。この例では、ネットワークのブートサーバーが利用可能になっていることを前提とします。システムをリブートしたあとで必ず新しい root パスワードを適用してください。

```
ok boot net -s
# mount /dev/dsk/c0t3d0s0 /a
# cd /a/etc
# TERM=vt100
# export TERM
# vi shadow
```

(Remove root's encrypted password string)

```
# cd /
# umount /a
# init 6
```

▼ SPARC: カーネルデバッグ (kmdb) を使ってシステムをブートする方法

この手順では、カーネルデバッグ (kmdb) を読み込むための基本的な操作を示します。詳細は、『Solaris モジュールデバッグ』を参照してください。

注- システムを対話式でデバッグする時間がない場合は、`-d` オプションを指定して `reboot` コマンドと `halt` コマンドを使用します。`-d` オプションを指定して `halt` コマンドを実行するには、実行後にシステムを手動でリブートする必要があります。これに対し、`reboot` コマンドを使用した場合、システムは自動的にブートします。詳細は、[reboot\(1m\)](#) のマニュアルページを参照してください。

- 1 システムを停止します。これにより、`ok` プロンプトが表示されます。
システムを正常に停止するには、`/usr/sbin/halt` コマンドを使用します。
- 2 `boot kmdb` または `boot -k` のどちらかを入力して、カーネルデバッグの読み込みを要求します。**Return** キーを押します。
- 3 カーネルデバッグを入力します。
デバッグを入力するための方法は、システムのアクセスに使用するコンソールのタイプによって異なります。
 - ローカル接続されたキーボードを使用している場合は、キーボードのタイプに応じて `Stop + A` または `L1 + A` キーを押してください。
 - シリアルコンソールを使用している場合は、使用中のシリアルコンソールのタイプに適した方法でブレークを送信します。

カーネルデバッグを初めて入力したときに、ウェルカムメッセージが表示されます。

```
Rebooting with command: kadb
Boot device: /iommu/sbus/espdma@4,800000/esp@4,8800000/sd@3,0
.
.
.
```

例 13-5 SPARC: システムをカーネルデバッガ (kmdb) でブートする

```
ok boot kmdb
Resetting...

Executing last command: boot kmdb -d
Boot device: /pci@1f,0/ide@d/disk@0,0:a File and args: kmdb -d
Loading kmdb...
```

x86 プラットフォームのブートのトラブルシューティング (作業マップ)

作業	説明	参照先
復旧を目的としてシステムを停止します。	破損したファイルによってシステムのブートが正常に行えない場合は、最初にシステムを停止して復旧を試みます。	302 ページの「x86: 復旧を目的としてシステムを停止する方法」
強制的にシステムのクラッシュダンプを実行し、システムをリブートします。	トラブルシューティング手段としてシステムのクラッシュダンプとリブートを強制的に実行できます。	303 ページの「x86: クラッシュダンプを強制してシステムをリブートする方法」
カーネルデバッガを使ってシステムをブートします。	カーネルデバッガを使ってシステムをブートして、ブート問題のトラブルシューティングを実行できます。システムのブートには kmdb コマンドを使用します。	304 ページの「x86: GRUB ブート環境でカーネルデバッガ (kmdb) を使ってシステムをブートする方法」

▼ x86: 復旧を目的としてシステムを停止する方法

- 1 可能であれば、次のいずれかのコマンドを使用してシステムを停止します。
- キーボードとマウスを使用できる場合は、スーパーユーザーになります。次に、init 0 と入力してシステムを停止します。Press any key to reboot プロンプトが表示されたら、任意のキーを押してシステムをリブートします。

■ キーボードとマウスを使用できる場合は、スーパーユーザーになります。次に、init 6 と入力してシステムをリブートします。
- 2 マウスまたはキーボードからの入力にシステムが応答しない場合、リセットキーがあればそのキーを押してシステムをリブートします。
- あるいは、電源スイッチを使用してシステムをリブートします。

x86: クラッシュダンプを強制してシステムをリブートする

場合によっては、トラブルシューティングの目的でクラッシュダンプを強制実行し、システムをリブートする必要があります。デフォルトでは、`savecore` 機能を使用できます。

システムのクラッシュダンプについては、『[Solaris のシステム管理 \(上級編\)](#)』の第 17 章「[システムクラッシュ情報の管理 \(手順\)](#)」を参照してください。

▼ x86: クラッシュダンプを強制してシステムをリブートする方法

`reboot -d` コマンドまたは `halt -d` コマンドを使用できない場合は、カーネルデバッガ `kldb` を使用してクラッシュダンプを強制的に実行できます。次の手順を実行するには、ブート時または `mdb -k` コマンド経由でカーネルデバッガが読み込まれている必要があります。

注-カーネルデバッガ (`kldb`) にアクセスするにはテキストモードでなければなりません。したがって、まずウィンドウシステムを終了してください。

1 カーネルデバッガにアクセスします。

デバッガにアクセスするための方法は、システムへのアクセスに使用するコンソールのタイプによって異なります。

- ローカル接続されているキーボードを使用している場合は、F1-A を押します。
- シリアルコンソールを使用している場合は、そのシリアルコンソールタイプに適した方法でブレークを送信します。

`kldb` プロンプトが表示されます。

2 クラッシュを強制実行するために、`systemdump` マクロを使用します。

```
[0]> $<systemdump
```

パニックメッセージが表示され、クラッシュダンプが保存され、システムがリブートします。

3 コンソールログインプロンプトでログインして、システムがリブートされていることを確認します。

例 13-6 x86:halt -d コマンドを使用してシステムのクラッシュダンプとリブートを強制実行する

この例は、halt -d コマンドと boot コマンドを使って、x86 システム neptune のクラッシュダンプとリブートを強制実行する方法を示しています。システムのクラッシュダンプを強制するには、この方法を使用してください。このあとに、システムを手動でリブートします。

```
# halt -d
4ay 30 15:35:15 wacked.Central.Sun.COM halt: halted by user

panic[cpu0]/thread=ffffffff83246ec0: forced crash dump initiated at user request

fffffe80006bbd60 genunix:kadmin+4c1 ()
fffffe80006bbec0 genunix:uadmin+93 ()
fffffe80006bbf10 unix:sys_syscall32+101 ()

syncing file systems... done
dumping to /dev/dsk/clt0d0s1, offset 107675648, content: kernel
NOTICE: adpu320: bus reset
100% done: 38438 pages dumped, compression ratio 4.29, dump succeeded

Welcome to kmdb
Loaded modules: [ audiosup crypto ufs unix krtld s1394 sPPP nca uhci lofs
genunix ip usba specfs nfs md random sctp ]
[0]>
kmdb: Do you really want to reboot? (y/n) y
```

▼ x86: GRUB ブート環境でカーネルデバッグ (kmdb) を使ってシステムをブートする方法

この手順では、カーネルデバッグ (kmdb) を読み込むための基本的な操作を示します。デフォルトでは、savecore 機能を使用できます。カーネルデバッグの使用方法の詳細については、『[Solaris モジュールデバッグ](#)』を参照してください。

- 1 システムをブートします。
システムがブートすると、GRUB メニューが表示されます。
- 2 GRUB メニューが表示されたら、e と入力して **GRUB** 編集メニューにアクセスします。
- 3 矢印キーを使用して、kernel\$ 行を選択します。
矢印キーを使用できない場合は、^キーを使って上方へスクロールしたり、v キーを使って下方へスクロールしたりします。

- 4 e と入力して、その行を編集します。
ブートエントリメニューが表示されます。このメニューで、`kernel$` 行の終わりに追加のブート引数を指定して、Solaris ブート動作を変更できます。
- 5 行の終わりに `-k` と入力します。
- 6 **Enter** キーを押して、**GRUB** メインメニューに戻ります。
- 7 カーネルデバッグが使用可能な状態でシステムをブートするために、`b` と入力します。
- 8 カーネルデバッグにアクセスします。
デバッグにアクセスするための方法は、システムへのアクセスに使用するコンソールのタイプによって異なります。
 - ローカル接続されているキーボードを使用している場合は、`F1-A` を押します。
 - シリアルコンソールを使用している場合は、そのシリアルコンソールタイプに適した方法でブレイクを送信します。
 カーネルデバッグに初めてアクセスした場合、ウェルカムメッセージが表示されます。

例 13-7 x86: カーネルデバッグ (GRUB マルチブート実装) を使用してシステムをブートする

次の例は、64 ビット対応 x86 システムをカーネルデバッグが使用可能な状態で手でブートする方法を示しています。

```
kernel$ /platform/i86pc/multiboot kernel/amd64/unix -k -B $ZFS-BOOTFS
```

次の例は、64 ビット対応 x86 システムをカーネルデバッグ使用可能な状態で 32 ビットモードでブートする方法を示しています。

```
kernel$ /platform/i86pc/multiboot kernel/unix -k -B $ZFS-BOOTFS
```


Solaris ブートアーカイブの管理 (手順)

この章では、Solaris OS でのブートアーカイブの管理について説明します。bootadm コマンドの詳細な使用手順も説明します。

この章の内容は次のとおりです。

- 307 ページの「Solaris ブートアーカイブの管理 (作業マップ)」
- 308 ページの「Solaris ブートアーカイブの説明」
- 309 ページの「boot-archive サービスの管理」
- 313 ページの「bootadm コマンドを使用してブートアーカイブを管理する」

ブートプロセスの概要については、第 9 章「システムのシャットダウンとブート (概要)」を参照してください。システムのブート手順については、第 12 章「Solaris システムのブート (手順)」を参照してください。

Solaris ブートアーカイブの管理 (作業マップ)

表 14-1 Solaris ブートアーカイブの管理: 作業マップ

作業	説明	参照先
boot-archive サービスを管理します。	boot-archive サービスは、サービス管理機能 (SMF) によって制御されます。サービスを有効/無効にするには、svcadm コマンドを使用します。boot-archive サービスが実行されているかどうかを確認するには、svcs コマンドを使用します。	309 ページの「boot-archive サービスの管理」

表 14-1 Solaris ブートアーカイブの管理:作業マップ (続き)

作業	説明	参照先
boot-archive サービスをクリアーします。	この手順は、フェイルセーフアーカイブのブートに代わる方法として使用します。boot-archive サービスがクリアーされたあとで、bootadm コマンドが暗黙のうちに実行されてブートアーカイブが更新されます。	310 ページの「boot-archive サービスをクリアーして矛盾したブートアーカイブを更新する方法」
bootadm コマンドを使用してブートアーカイブを更新します。	ブートアーカイブを手動で更新する場合は、bootadm update-archive コマンドを使用します。	313 ページの「ブートアーカイブを手動で更新する方法」
ミラー化ルート (/) パーティションのブートアーカイブを手動で更新します。	ルート (/) パーティションにメタデバイスマirrorを使用するシステムでは、フェイルセーフアーカイブをブートし、bootadm update-archive コマンドを実行してブートアーカイブを手動更新すると失敗します。この問題は、ミラーがメタデバイスであるために発生します。そのため、ブートアーカイブは手動で更新してください。	314 ページの「RAID-1 (ミラー) ボリューム上のブートアーカイブを手動で更新する方法」
bootadm コマンドを使用してブートアーカイブの内容を一覧表示します。	ブートアーカイブの内容を一覧表示する場合は、bootadm list-archive コマンドを使用します。	320 ページの「ブートアーカイブの内容を一覧表示する方法」
x86 のみ: bootadm コマンドを使用してアクティブな GRUB メニューを検出します。	アクティブな GRUB メニューの場所を調べる場合は、bootadm list-menu コマンドを使用します。	320 ページの「x86: アクティブな GRUB メニューを検出し、現在のメニューエントリを一覧表示する方法」
x86 のみ: bootadm コマンドを使用して GRUB メニューのデフォルトのブートエントリを設定します。	GRUB メニューのデフォルトのブートエントリを設定する場合は、bootadm set-menu コマンドを使用します。	321 ページの「x86: アクティブな GRUB メニューのデフォルトのブートエントリを設定する方法」

Solaris ブートアーカイブの説明

Solaris OS をシステムにインストールすると、bootadm コマンドによって一次ブートアーカイブとフェイルセーフアーカイブが1つずつ作成されます。

「一次ブートアーカイブ」は、ルート (/) ファイルシステムのサブセットです。このブートアーカイブには、すべてのカーネルモジュール、driver.conf ファイル、およびいくつかの構成ファイルが含まれています。これらのファイルは、/etc ディレクトリにあります。ブートアーカイブ内のファイルは、ルート (/) ファイルシステムがマウントされる前に、カーネルによって読み取られます。ルート (/) ファイルシステ

ムがマウントされると、ブートアーカイブはカーネルによってメモリーから破棄されます。次に、ファイル I/O がルートデバイスに実行されます。

SPARC ブートアーカイブを構成するファイルは、`/platform` ディレクトリにあります。

このディレクトリの内容は、3つのグループのファイルに分けられます。

- `sun4u` ブートアーカイブに必要なファイル
- `sun4v` ブートアーカイブに必要なファイル
- `sun4us` ブートアーカイブに必要なファイル

x86 ブートアーカイブを構成するファイルは、`/platform/i86pc` ディレクトリにあります。

ブートアーカイブに含まれるファイルとディレクトリを一覧表示するには、`bootadm list-archive` コマンドを使用します。

アーカイブ内のファイルが更新された場合は、ブートアーカイブを再構築する必要があります。変更を有効にするには、次にシステムがリブートする前にアーカイブを再構築する必要があります。

「フェイルセーフ」ブートアーカイブは、Solaris OS のインストール時に作成される 2 番目の種類のアーカイブです。

フェイルセーフブートアーカイブには、次のような利点と特徴があります。

- 独立したブートアーカイブである
- ブートアーカイブ自体でブートできる
- デフォルトでは OS のインストール中に作成される
- 保守の必要がない

フェイルセーフモードでのシステムのブートの詳細については、[264 ページ](#)の「SPARC システムでフェイルセーフアーカイブをブートする」および [284 ページ](#)の「x86 システムでフェイルセーフアーカイブをブートする」を参照してください。

boot-archive サービスの管理

boot-archive サービスは、サービス管理機能 (SMF) によって制御されます。boot-archive サービスインスタンスは、`svc:/system/boot-archive:default` です。サービスを有効/無効にするには、`svcadm` コマンドを使用します。

boot-archive サービスが実行されているかどうかを確認するには、`svcs` コマンドを使用します。

詳細は、[svcadm\(1M\)](#) および [svcs\(1\)](#) のマニュアルページを参照してください。

▼ boot-archive サービスを有効または無効にする方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
- 2 boot-archive サービスを有効または無効にするには、次のように入力します。

```
# svcadm enable | disable system/boot-archive
```

- 3 boot-archive サービスの状態を確認するには、次のように入力します。

```
% svcs boot-archive
```

サービスが実行されている場合は、オンラインサービスの状態が表示されます。

```
STATE          STIME      FMRI
online          9:02:38    svc:/system/boot-archive:default
```

サービスが実行されていない場合は、サービスがオフラインになっていることが表示されます。

注意事項 boot-archive サービスをクリアーしてブートアーカイブを更新する方法については、[310 ページの「boot-archive サービスをクリアーして矛盾したブートアーカイブを更新する方法」](#)を参照してください。

▼ boot-archive サービスをクリアーして矛盾したブートアーカイブを更新する方法

boot-archive サービス svc:/system/boot-archive は SMF によって管理されます。次の手順では、ブートプロセス中に矛盾したアーカイブが検出されたときにブートアーカイブを更新する方法を示します。このサービスをクリアーすることは、boot-F failsafe コマンドを実行するのと同じ効果があります。この方法でブートアーカイブを更新すると、フェイルセーフアーカイブをブートしたり、bootadm update-archive コマンドを実行したりする必要がありません。このコマンドは、boot-archive サービスがクリアーされたあとで暗黙のうちに実行されます。



注意 - 矛盾したブートアーカイブを修正する場合は、フェイルセーフモードでシステムをブートすることをお勧めします。フェイルセーフアーカイブをブートする方法については、次の情報を参照してください。

SPARC システムの場合は、[269 ページ](#)の「[SPARC システムをネットワークからブートする](#)」を参照してください。

x86 システムの場合は、[284 ページ](#)の「[x86 システムでフェイルセーフアーカイブをブートする](#)」を参照してください。

- 1 システムのブートプロセス中に、次のような警告が表示された場合は、その警告を無視します。

WARNING: The following files in / differ from the boot archive:

changed file-name

システムがシステム保守モードに入ります。

- 2 次のコマンドを入力して、**boot-archive** サービスをクリアーします。

```
# svcadm clear system/boot-archive
```

このコマンドが実行されたあとで、`bootadm update-archive` コマンドが暗黙のうちに実行されます。ブートアーカイブが正常に更新されると、システムがリブートします。

- 3 サービスが実行されていることを確認します。

```
# svcs boot-archive
STATE          STIME          FMRI
online          9:02:38      svc:/system/boot-archive:default
```

例 14-1 SPARC: boot-archive サービスをクリアーして矛盾したブートアーカイブを更新する

```
screen not found.
Can't open input device.
Keyboard not present. Using ttys for input and output.
```

```
Sun Enterprise 220R (2 X UltraSPARC-II 450MHz), No Keyboard
OpenBoot 3.23, 1024 MB memory installed, Serial #13116682.
Ethernet address 8:0:20:c8:25:a, Host ID: 80c8250a.
```

```
Rebooting with command: boot
Boot device: /pci@1f,4000/scsi@3/disk@1,0:a File and args:
```

```
SunOS Release 5.10 64-bit
Copyright 1983-2007 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
DEBUG enabled
misc/forthdebug (507204 bytes) loaded
Hostname: marnie
```

WARNING: The following files in / differ from the boot archive:

```
changed /kernel/drv/sd.conf
```

The recommended action is to reboot to the failsafe archive to correct the above inconsistency. To accomplish this, on a GRUB-based platform, reboot and select the "Solaris failsafe" option from the boot menu. On an OBP-based platform, reboot then type "boot -F failsafe". Then follow the prompts to update the boot archive. Alternately, to continue booting at your own risk, you may clear the service by running: "svcadm clear system/boot-archive"

```
Nov 21 15:47:20 svc.startd[100004]: svc:/system/boot-archive:default: Method
"/lib/svc/method/boot-archive" failed with exit status 95.
Nov 21 15:47:20 svc.startd[100004]: system/boot-archive:default failed fatally:
transitioned to maintenance (see 'svcs -xv' for details)
Requesting System Maintenance Mode
(See /lib/svc/share/README for more information.)
Console login service(s) cannot run
```

```
Root password for system maintenance (control-d to bypass):
single-user privilege assigned to /dev/console.
Entering System Maintenance Mode
```

```
Nov 21 15:48:36 su: 'su root' succeeded for root on /dev/console
Sun Microsystems Inc. SunOS 5.10,
2007
```

```
.
.
.#
#
# svcadm clear system/boot-archive
#
# NIS domain name is mpklab.sfbay.sun.com
/dev/rdisk/c0t1d0s5 is clean
Reading ZFS config: done.
#
# bootadm update-archive
# svcs boot-archive
STATE          STIME      FMRI
online         9:02:38  svc:/system/boot-archive:default
```


bootadm コマンドを使用してブートアーカイブを管理する

/sbin/bootadm コマンドを使用すると、次の作業を行えます。

- システム上の現在のブートアーカイブを手動で更新する。
- システム上のブートアーカイブに含まれるファイルとディレクトリを一覧表示する。
- **x86** のみ: GRUB メニューを保守する。
- **x86** のみ: アクティブな GRUB メニューと現在の GRUB メニューエントリを検出する。

このコマンドの構文は次のとおりです。

```
/sbin/bootadm [subcommand] [-option] [-R altroot]
```

bootadm コマンドの詳細は、[bootadm\(1M\)](#) のマニュアルページを参照してください。

▼ ブートアーカイブを手動で更新する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 現在のブートアーカイブを更新するには、次のように入力します。

```
# bootadm update-archive
```

bootadm システム上のブートアーカイブを管理します。

update-archive 必要に応じて、現在のブートアーカイブを更新します。SPARC と x86 の両方のシステムに適用されます。

- 代替ルートのブートアーカイブを更新するには、次のように入力します。

```
# bootadm update-archive -R /a
```

-R altroot update-archive サブコマンドに適用する代替ルートパスを指定します。

注 - `-R` オプションを使用して非大域ゾーンのルート (`/`) ファイルシステムを参照してはいけません。そうすると、大域ゾーンのファイルシステムが損傷したり、大域ゾーンのセキュリティが低下したり、非大域ゾーンのファイルシステムが損傷したりする場合があります。[zones\(5\)](#) のマニュアルページを参照してください。

▼ RAID-1 (ミラー) ボリューム上のブートアーカイブを手動で更新する方法

注 - ここで示す手順は、Solaris ボリュームマネージャー (SVM) を使用して作成および管理している RAID-1 (ミラー) ボリューム上での、ブートアーカイブの更新に適用されます。

ブートアーカイブとルート (`/`) ファイルシステムが矛盾した状態になると、システムのブート時にエラーメッセージが表示されます。通常は、システムをフェイルセーフモードでブートしてから、`bootadm update-archive` コマンドを実行して、ブートアーカイブを更新するという操作をお勧めします。ただし、ルート (`/`) ファイルシステムがミラー化メタデバイス (RAID-1 ボリューム) の場合は、この方法でブートアーカイブを正常に更新することはできません。

システムをフェイルセーフモードでブートすると、次のようなメッセージが表示されます。

```
Searching for installed OS instances...
/dev/dsk/c0t0d0s0 is under md control, skipping.
/dev/dsk/c1t3d0s0 is under md control, skipping.
No installed OS instance found.
```

このメッセージは、メタデバイスがスキップされたことを示します。ブートアーカイブを手動で更新するには、以下に説明している手順に従います。

- 1 矛盾するブートアーカイブのあるシステム上で、スーパーユーザーになるか、同等の役割を引き受けます。
- 2 フェイルセーフアーカイブをブートします。
 - SPARC システムでは、次のように入力します。

```
# reboot -- "-F failsafe"
```

ok プロンプトからフェイルセーフアーカイブをブートするには、次のように入力します。

```
ok boot -F failsafe
```

詳細については、[265 ページの「SPARC システムでフェイルセーフアーカイブをブートする方法」](#)を参照してください。

- **x86 システムでは、システムをブートし、GRUB メニューでフェイルセーフブートエントリを選択します。**

詳細は、[284 ページの「GRUB を使用して x86 システムでフェイルセーフアーカイブをブートする方法」](#)を参照してください。

システムがフェイルセーフモードでブートし、インストールされている OS のインスタンスを検索したあと、前に説明した“No installed OS instance found”というメッセージが返されます。ブートシーケンスが完了すると、コマンドプロンプトが表示されます。

3 一次サブミラーをマウントします。

次に例を示します。

```
# mount /dev/dsk/c0t0d0s0 /a
```

注-システムがフェイルセーフモードでブートされる場合は、`metastat -p` コマンドを使用して一次サブミラーを判断できません。どのスライスが一次サブミラーかについての情報は、フェイルセーフブートプロセス時にコンソールに出力されます。代わりにこの方法を使用して、一次サブミラーを判断できます。

4 単一のルート (/) パーティションを使用するように、/etc/vfstab ファイルを一時的に更新します。

- 元の vfstab ファイルのコピーを作成します。

```
# cp /a/etc/vfstab /a/etc/vfstab.orig
```

- テキストエディタを使用して、vfstab ファイルを次のように編集します。

- ルート (/) ミラーメタデバイスの行をコメントアウトします。

#device	device	mount	FS	fsck	mount	mount
#to mount	to fsck	point	type	pass	at boot	options
#	.	.	.	.	.	.
#/dev/md/dsk/d10	/dev/md/rdisk/d10	/	ufs	1	no	-

前の例では、`/dev/md/dsk/d10` の行がコメントアウトされました。

- ii. 一次サブミラーのディスクデバイス用の新しい行を追加します。

```
#device      device      mount      FS      fsck      mount      mount
#to mount    to fsck      point      type      pass      at boot    options
#
.
.
.
#/dev/md/dsk/d10 /dev/md/rdisk/d10 /      ufs      1      no      -
/dev/dsk/c0t0d0s0 /dev/rdisk/c0t0d0s0 /      ufs      1      no      -
.
.
.
```

前の例では、一次サブミラーのディスクデバイス用の新しい行として `/dev/dsk/c0t0d0s0` が追加されました。

- c. 変更を保存します。

- 5 システムがメタデバイスからのブートを試行しないように、`/etc/system` ファイルを次のように一時的に更新します。

- a. 元の `/etc/system` ファイルのコピーを作成します。

```
# cp /a/etc/system /a/etc/system.orig
```

- b. テキストエディタを使用して `/etc/system` ファイルを編集し、`rootdev` 行をコメントアウトします。この行は、`Begin MDD root` 行と `End MDD root` 行の間にあります。

```
* Begin MDD root info (do not edit)
# rootdev:/pseudo/md@0:0,0,blk
* End MDD root info (do not edit)
```

- c. 変更を保存します。

- 6 次のコマンドを実行して、ブートアーカイブを更新します。

```
# bootadm update-archive -R /a
```

- 7 一次サブミラーのマウントを解除してから、システムをリブートします。

```
# umount /a
```

- それでもシステムが正常にブートしない場合は、フェイルセーフアーカイブをリブートし、`/etc/vfstab` ファイルと `/etc/system` ファイルの情報が正しいことを確認してください。

- 8 システムのリブートに成功したら、メタデバイスを再構築します。
- vfstab ファイルからルート (/) ミラーメタデバイスの名前を取得します。
メタデバイスの名前は、手順 5 でコメントアウトした行です。

- metastat コマンドを使用して、ミラーのコンポーネントを表示します。
次に例を示します。

```
# metastat -p
d10 -m d0 d1 1
d0 1 1 c0t0d0s0
d1 1 1 c1t3d0s0
```

- 障害の発生したサブミラーを切り離します。

```
# metadetach mirror submirror
```

次に例を示します。

```
# metadetach d10 d1
```

- 既存の /etc/vfstab ファイルのコピーを元のファイルで置き換えます。

```
# cp /etc/vfstab.orig /etc/vfstab
```

- 既存の /etc/system ファイルのコピーを元のファイルで置き換えます。

```
# cp /etc/system.orig /etc/system
```

- システムを再起動します。

```
# shutdown -i 6
```

システムのリブート後、ミラー化ルート (/) パーティションがメタデバイス上に復元されます。

- 9 前の手順で切り離したサブミラーを再接続します。

```
# metattach mirror submirror
```

次に例を示します。

```
# metattach d10 d1
```

ミラーの再同期が始まります。

- 10 再同期処理の状態を確認するには、metastat コマンドを使用します。

```
# metastat | grep 'Resync in progress'
```

何の出力も返されない場合、プロセスは完了しています。

例 14-2 SPARC:RAID-1(ミラー) ボリューム上のブートアーカイブの手動更新

この例では、SVM ルート (/) のミラー化メタデバイスのあるシステム上のブートアーカイブを手動で更新する手順を示します。この例でを使用したシステムは、Solaris 10 10/08 リリースが稼動している SPARC システムです。

```
SunOS Release 5.10 Version Generic_137137-09 64-bit
Copyright 1983-2008 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
WARNING: Unexpected EOF on line 14 of /kernel/drv/md.conf
Hostname: pilgrim1
```

```
WARNING: The following files in / differ from the boot archive:
```

```
changed /kernel/drv/md.conf
```

The recommended action is to reboot to the failsafe archive to correct the above inconsistency. To accomplish this, on a GRUB-based platform, reboot and select the "Solaris failsafe" option from the boot menu. On an OBP-based platform, reboot then type "boot -F failsafe". Then follow the prompts to update the boot archive. Alternately, to continue booting at your own risk, you may clear the service by running: "svcadm clear system/boot-archive"

```
Sep 18 15:22:06 svc.startd[7]: svc:/system/boot-archive:default:
Method "/lib/svc/method/boot-archive" failed with exit status 95.
Sep 18 15:22:06 svc.startd[7]: system/boot-archive:default
failed fatally: transitioned to maintenance (see 'svcs -xv' for details)
Requesting System Maintenance Mode
(See /lib/svc/share/README for more information.)
Console login service(s) cannot run
```

```
Root password for system maintenance (control-d to bypass):
single-user privilege assigned to /dev/console.
Entering System Maintenance Mode
```

```
Sep 18 15:22:18 su: 'su root' succeeded for root on /dev/console
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
# reboot -- "-F failsafe"
syncing file systems... done
rebooting...
Resetting ...
Rebooting with command: boot -F failsafe
Boot device: /pci@1f,4000/scsi@3/disk@0,0:a File and args: -F failsafe
SunOS Release 5.10 Version Generic_137137-08 64-bit
Copyright 1983-2008 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
```

```

Configuring devices.
Searching for installed OS instances...
/dev/dsk/c0t0d0s0 is under md control, skipping.
/dev/dsk/clt3d0s0 is under md control, skipping.
No installed OS instance found.

Starting shell.

# mount /dev/dsk/c0t0d0s0 /a
# cp /a/etc/vfstab /a/etc/vfstab.orig
# vi /a/etc/vfstab

<< input changes to vfstab file, then save changes >>

# cp /a/etc/system /a/etc/system.orig
# vi /a/etc/system

<< input changes to /etc/system file, then save changes >>

# bootadm update-archive -R /a
Creating boot_archive for /a
updating /a/platform/sun4u/boot_archive
15+0 records in
15+0 records out
# umount /a
# shutdown -i 6

<< reboot the system >>

Rebooting with command: boot
Boot device: /pci@1f,4000/scsi@3/disk@0,0:a File and args:
SunOS Release 5.10 Version Generic_137137-08 64-bit
Copyright 1983-2008 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
[...]
# metastat -p d10
# metadetach d10 d1
# cp /etc/vfstab.orig /etc/vfstab
# cp //etc/system.orig /etc/system
# shutdown -i 6

<< reboot the system >>

Rebooting with command: boot
Boot device: /pci@1f,4000/scsi@3/disk@0,0:a File and args:
SunOS Release 5.10 Version Generic_137137-08 64-bit
Copyright 1983-2008 Sun Microsystems, Inc. All rights reserved.

```

```
Use is subject to license terms.  
[...]  
# metattach d10 d1  
# metastat | grep 'Resync in progress'  
    Resync in progress: 4 % done  
# metastat | grep 'Resync in progress'
```

▼ ブートアーカイブの内容を一覧表示する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 ブートアーカイブに含まれるファイルとディレクトリを一覧表示するには、次のように入力します。

```
# bootadm list-archive
```

list-archive ブートアーカイブに含まれるファイルとディレクトリを一覧表示します。SPARC と x86 の両方のシステムに適用されます。

▼ x86: アクティブな GRUB メニューを検出し、現在のメニューエントリを一覧表示する方法

次の手順に従って、アクティブな GRUB メニューの場所を調べ、現在の GRUB メニューエントリを一覧表示します。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 アクティブな GRUB メニューの場所と現在の GRUB メニューエントリを一覧表示するには、次のように入力します。

```
# bootadm list-menu
```

list-menu アクティブな GRUB メニューの場所と現在の GRUB メニューエントリを一覧表示します。autoboot-timeout、デフォルトのエントリ番号、および各エントリのタイトルに関する情報がこのリストに表示されます。x86 システムだけに適用されます。

例 14-3 アクティブな GRUB メニューの場所と現在の GRUB メニューエントリを一覧表示する

```
# bootadm list-menu
The location for the active GRUB menu is: /stubboot/boot/grub/menu.lst
default=0
timeout=10
(0) Solaris10
(1) Solaris10 Failsafe
(2) Linux
```

▼ x86: アクティブな GRUB メニューのデフォルトのブートエントリを設定する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、[『Solaris のシステム管理 \(セキュリティサービス\)』の「RBAC の構成 \(作業マップ\)」](#)を参照してください。
- 2 アクティブな GRUB メニューのデフォルトのブートエントリを設定するには、次のように入力します。

```
# bootadm set-menu menu-entry

set-menu      GRUB メニューを保守します。アクティブな GRUB メニューの場所は
               boot/grub/menu.lst です。x86 システムだけに適用されます。

menu-entry    デフォルトとして設定する GRUB メニューエントリを指定します。
```

- 3 デフォルトのメニューエントリが変更されていることを確認するには、次のように入力します。

```
# bootadm list-menu

新しいデフォルトのメニューエントリが表示されるはずです。
```

例 14-4 GRUB のデフォルトのメニューエントリを切り替える

次の例は、デフォルトの GRUB メニューを、前の例に表示されているメニューエントリのいずれかに切り替える方法を示しています。選択したメニューエントリは、Linux (メニューエントリ 2) です。

```
# bootadm set-menu default=2
```

参照 それぞれの GRUB 実装における menu.lst ファイルについては、[329 ページの「x86: サポートされている GRUB 実装」](#)を参照してください。

x86: GRUB ベースのブート (参照情報)

この章には、GRUB 実装の詳細や GRUB の追加の参照情報など、x86 ブートプロセスに関する情報が記載されています。

概要については、[第9章「システムのシャットダウンとブート \(概要\)」](#)を参照してください。

システムのブート手順については、[第12章「Solaris システムのブート \(手順\)」](#)を参照してください。

x86: ブートプロセス

この節では、x86 システムに固有のブートプロセスについて説明します。

x86: システム BIOS

電源が入っているときには、システムは読み取り専用メモリー (ROM) の BIOS (Basic Input/Output System) によって制御されます。BIOS は、64 ビットおよび 32 ビットの x86 をサポートする、Solaris オペレーティングシステムのファームウェアインタフェースです。

ハードウェアアダプタには通常、そのデバイスの物理特性を表示するオンボード BIOS が搭載されています。この BIOS は、デバイスにアクセスするときに使用されます。システムの BIOS は、起動プロセス中に、アダプタの BIOS が存在するかどうかを調べます。アダプタが見つかった場合は、そのアダプタの BIOS を読み込んで実行します。アダプタの BIOS はそれぞれセルフテスト診断を実行してから、デバイス情報を表示します。

ほとんどのシステムの BIOS には、ブートデバイスの優先順位を選択するためのユーザーインタフェースが用意されており、次の項目が表示されます。

- フロッピーディスク
- CD または DVD
- ハードディスク
- ネットワーク

BIOS は、ブート可能なプログラムがある有効なデバイスが見つかるまで、このリストに従って各デバイスからブートを試みます。

x86: カーネル初期化プロセス

/platform/i86pc/multiboot プログラムは、マルチブート仕様に定義されているヘッダーを含む、ELF32 実行可能ファイルです。

マルチブートプログラムは、次のタスクを実行します。

- ブートアーカイブの内容を解釈する
- 64 ビット対応のシステムを自動検出する
- システムのブートに最適なカーネルモードを選択する
- コアカーネルモジュールをメモリー内でアセンブルする
- システムの制御を Solaris カーネルに渡す

システムの制御を渡されたカーネルは、CPU、メモリー、およびデバイスサブシステムを初期化します。カーネルは次に、ルートデバイスをマウントします。/boot/solaris/bootenv.rc ファイルに指定されている bootpath および fstype プロパティに対応するデバイスがマウントされます。このファイルはブートアーカイブに含まれます。これらのプロパティが bootenv.rc ファイルまたは GRUB コマンド行に指定されていない場合には、ルートファイルシステムはデフォルトで /devices/ramdisk:a 上の UFS に設定されます。インストールミニルートをブートする場合には、ルートファイルシステムはデフォルトで UFS に設定されます。ルートデバイスがマウントされたあと、カーネルは sched コマンドと init コマンドを初期化します。これらのコマンドは、サービス管理機能 (SMF) サービスを開始します。

x86: GRUB ブートローダーの Solaris サポート

この節には、Solaris OS で GRUB を管理するための追加の参照情報が記載されています。

x86: GRUB の用語

GRUB の概念を十分に把握するには、次の用語を理解することが不可欠です。

注- このリストに記載されている用語の中には、GRUB ベースのブートだけに限定されないものもあります。

ブートアーカイブ

Solaris OS のブートに使用されるクリティカルなファイルの集まり。これらのファイルは、ルートファイルシステムがマウントされる前、システムの起動中に必要です。システムは、複数のブートアーカイブを維持管理しています。

- 「一次ブートアーカイブ」は、x86 システムで Solaris OS をブートするために使用されます。
- 「フェイルセーフブートアーカイブ」は、一次ブートアーカイブが損傷を受けたとき、回復のために使用されます。このブートアーカイブは、ルートファイルシステムをマウントすることなくシステムを起動します。GRUB メニューでは、このブートアーカイブは「フェイルセーフ」と呼ばれます。このアーカイブの主な目的は一次ブートアーカイブを再生成することであり、通常は一次ブートアーカイブがシステムのブートに使用されます。

ブートローダー

システムの電源を入れたあとに最初に行われるソフトウェアプログラム。このプログラムがブートプロセスを開始します。

フェイルセーフアーカイブ

ブートアーカイブを参照してください。

GRUB

GRUB (GNU GRand Unified Bootloader) は、メニューインタフェースを備えたオープンソースのブートローダーです。このメニューには、システムにインストールされているオペレーティングシステムの一覧が表示されます。GRUB を使用すると、Solaris OS、Linux、Windows などのさまざまなオペレーティングシステムを簡単にブートすることができます。

GRUB メインメニュー

システムにインストールされているオペレーティングシステムがリストされたブートメニュー。このメニューから、BIOS または fdisk パーティションの設定を変更することなく、簡単にオペレーティングシステムをブートできます。

GRUB 編集メニュー	GRUB メインメニューのサブメニュー。このサブメニューには、GRUB コマンドが表示されます。これらのコマンドを編集して、ブート動作を変更できます。
menu.lst ファイル	システムにインストールされているすべてのオペレーティングシステムがリストされた構成ファイル。このファイルの内容は、GRUB メニューに表示されるオペレーティングシステムの一覧を記述したものです。GRUB のメニューから、BIOS または fdisk パーティションの設定を変更することなく、簡単にオペレーティングシステムをブートできます。
ミニルート	ブート可能な最小ルート (/) ファイルシステム。Solaris インストールメディアに収録されています。ミニルートは、システムのインストールおよびアップグレードに必要な Solaris ソフトウェアで構成されます。x86 システムでは、ミニルートはシステムにコピーされて、フェイルセーフブートアーカイブとして使用されます。フェイルセーフブートアーカイブの詳細は、ブートアーカイブを参照してください。
一次ブートアーカイブ	ブートアーカイブを参照してください。

x86: GRUB の機能コンポーネント

GRUB は、次の機能コンポーネントで構成されています。

- stage1 – Solaris fdisk パーティションの最初のセクターにインストールされるイメージです。installgrub コマンドに -m オプションを指定して、stage1 をマスターブートセクターにインストールすることもできます。詳細は、[installgrub\(1M\)](#) のマニュアルページと『[Solaris のシステム管理 \(デバイスとファイルシステム\)](#)』の「[GRUB ブート環境でのディスク管理](#)」を参照してください。
- stage2 – Solaris fdisk パーティションの予約済み領域にインストールされるイメージです。stage2 イメージは GRUB のコアイメージです。
- menu.lst ファイル – 通常は UFS ルートを含むシステムの /boot/grub ディレクトリおよび ZFS ルートを含むシステムの /pool-name/boot/grub ディレクトリにあります。このファイルは、GRUB stage2 ファイルによって読み取られます。詳細は、[241 ページ](#)の「[x86: menu.lst ファイルを編集してブート動作を変更する](#)」の節を参照してください。

dd コマンドを使用して、stage1 および stage2 のイメージをディスクに書き込むことはできません。stage1 イメージは、ディスク上にある stage2 イメージの場所に関す

る情報を受け取れる必要があります。installgrub コマンドを使用してください。このコマンドは、GRUB ブートブロックをインストールするためにサポートされている方法です。

GRUB の構成に使用される命名規則

GRUB が使用するデバイス命名規則は、以前の Solaris リリースの場合と多少異なっています。GRUB デバイス命名規則を理解すると、使用しているシステムで GRUB を構成するときに、ドライブとパーティションの情報を正しく指定できます。

次の表に、この Solaris リリースの GRUB デバイス命名規則を示します。

表 15-1 GRUB デバイスの規則

デバイス名	説明
(fd0)	最初のフロッピーディスク
(fd1)	2 番目のフロッピーディスク
(nd)	ネットワークデバイス
(hd0,0)	最初のハードディスク上の最初の fdisk パーティション
(hd0,1)	最初のハードディスク上の 2 番目の fdisk パーティション
(hd0,0,a),	最初のハードディスク上の最初の fdisk パーティションのスライス a
(hd0,0,b)	最初のハードディスク上の最初の fdisk パーティションのスライス b

注-GRUB デバイス名はすべて括弧で囲む必要があります。

fdisk パーティションの詳細については、『Solaris のシステム管理 (デバイスとファイルシステム)』の「fdisk パーティションの作成上のガイドライン」を参照してください。

findroot コマンドで使用される命名規則

Solaris 10 10/08 リリース以降では、GRUB で以前使用されていた root コマンドが findroot コマンドに置き換わりました。findroot コマンドは、ブートデバイスに関係なく対象ディスクを見つけるための拡張機能を提供します。findroot コマンドは、ZFS ルートファイルシステムからのブートもサポートします。

次に、さまざまな GRUB 実装で `findroot` コマンドが使用するデバイス命名規則についての説明を示します。

- Solaris Live Upgrade の場合:

```
findroot (BE_x,0,a)
```

x 変数はブート環境の名前です。

- ZFS をサポートしているシステムの標準アップグレードおよび新規インストール:

```
findroot(pool_p,0,a)
```

p 変数はルートプールの名前です。

- UFS をサポートしているシステムの標準アップグレードおよび新規インストール:

```
findroot (rootfsN,0,a)
```

N 変数は 0 から始まる整数です。

GRUB で複数のオペレーティングシステムをサポートする方法

この節では、同じディスク上にある複数のオペレーティングシステムを GRUB でサポートする方法について説明します。次の例は、Solaris 10 10/08 OS、Solaris 9 OS、Linux、および Windows が同じディスク上にインストールされている x86 システムを示しています。

表 15-2 GRUB メニュー設定の例

オペレーティングシステム	ディスク上の場所
Windows	fdisk パーティション 0
Linux	fdisk パーティション 1
Solaris	fdisk パーティション 2
Solaris 9 OS	スライス 0
Solaris 10 10/08 OS	スライス 3

この情報に基づいて、GRUB メニューが次のように表示されます。

```
title Solaris 10
    findroot (pool_rpool,0,a)
    kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
```



```

        module /platform/i86pc/boot_archive
title Solaris 9 OS (pre-GRUB)
    root (hd0,2,a)
    chainloader +1
    makeactive
title Linux
    root (hd0,1)
    kernel <from Linux GRUB menu...>
    initrd <from Linux GRUB menu...>
title Windows
    root (hd0,0)
    chainloader +1

```

注 - Solaris スライスを実アクティブなパーティションにする必要があります。また、Windows メニューの下に、`makeactive` を指定しないでください。そのようにすると、毎回 Windows がブートされることになります。Linux が GRUB をマスターブートブロック上にインストールした場合には、Solaris ブートオプションにアクセスできません。Solaris パーティションを実アクティブなパーティションにしたかどうかにかかわらず、Solaris ブートオプションにアクセスできなくなります。

このような場合には、次のいずれかの操作を実行できます。

- Linux 上でメニューを変更して、Linux GRUB からチェーンロードする。
「チェーンロード」は、別のブートローダーを使用して、サポートされていないオペレーティングシステムを読み込むためのメカニズムです。
- `m` オプションを指定した `-installgrub` コマンドを実行して、マスターブートブロックを Solaris GRUB に置き換える。

```
# installgrub -m /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/root-slice
```

詳細は、[installgrub\(1M\)](#) のマニュアルページを参照してください。

Solaris Live Upgrade ブート環境の詳細については、『[Solaris 10 インストールガイド \(Solaris Live Upgrade とアップグレードの計画\)](#)』を参照してください。

x86: サポートされている GRUB 実装

Solaris 10 OS では、GRUB はマルチブート実装を使用します。Solaris Express リリースでは、`menu.lst` ファイルの内容は、実行している Solaris リリース、使用するインストール方法、およびシステムを ZFS ルートまたは UFS ルートのどちらからブートしているかに応じて異なります。

- **GRUB ZFS** ブートのサポート
`menu.lst` ファイルと例については、[330 ページの「menu.lst ファイルの説明 \(ZFS サポート\)」](#)を参照してください。

■ GRUB UFS ブートのサポート

menu.lst ファイルと例については、[330 ページの「menu.lst ファイルの説明 \(UFS サポート\)」](#)を参照してください。

menu.lst ファイルの説明 (ZFS サポート)

次に、ZFS ブートローダーを含むブート環境の menu.lst ファイルのさまざまな例を示します。

注- ミニルートは実際のルートファイルシステムとしてマウントされるため、フェイルセーフアーカイブが ZFS データセットから読み取られても、menu.lst ファイル内のフェイルセーフブート用のエントリが ZFS bootfs プロパティーに変わることはありません。ブートローダーがミニルートを読み取ったあとは、ZFS データセットはアクセスされません。

例 15-1 デフォルトの menu.lst ファイル (新規インストールまたは標準アップグレード)

```
title Solaris 10 5/08 s10x_nbu6wos_nightly X86
findroot (pool_rpool,0,a)
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive

title Solaris failsafe
findroot (pool_rpool,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttyb
module /boot/x86.miniroot-safe
```

例 15-2 デフォルトの menu.lst ファイル (Solaris Live Upgrade)

```
title bel
findroot (BE_bel,0,a)
bootfs rpool/ROOT/szboot_0508
kernel$ /platform/i86pc/multiboot -B $ZFS-BOOTFS
module /platform/i86pc/boot_archive

title bel failsafe
findroot (BE_bel,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttyb
module /boot/x86.miniroot-safe
```

menu.lst ファイルの説明 (UFS サポート)

次に、UFS からのブートをサポートするシステムの menu.lst ファイルの例を示します。

例 15-3 デフォルトの GRUB menu.lst ファイル (新規インストールまたは標準アップグレード)

```
title Solaris 10 5/08 s10x_nbu6wos_nightly X86
findroot (pool_rpool,0,a)
kernel /platform/i86pc/multiboot
module /platform/i86pc/boot_archive

title Solaris failsafe
findroot (rootfs0,0,a)
kernel /boot/multiboot kernel/unix -s -B console-ttyb
module /boot/x86.miniroot-safe
```

例 15-4 デフォルトの GRUB menu.lst ファイル (Solaris Live Upgrade)

```
title bel
findroot (BE_bel,0,a)
kernel /platform/i86pc/multiboot
module /platform/i86pc/boot_archive

title bel failsafe
findroot (BE_bel,0,a)
kernel /boot/multiboot kernel/unix -s -B console=ttyb
module /boot/x86.miniroot-safe
```


x86: GRUB を実装しないシステムのブート (手順)

この章では、GRUB ベースのブートを実装しないリリースの Solaris 10 OS で x86 システムをブートする手順について説明します。

注 - Solaris 10 1/06 リリース以降では、オープンソースの GRand Unified Bootloader (GRUB) が x86 システム上で実装されています。GRUB は、カーネルモジュールおよび構成ファイルが含まれているブートアーカイブをシステムのメモリーに読み込む役割を果たします。GRUB ベースのブートについての詳細は、[271 ページの「GRUB を使用して x86 システムをブートする \(作業マップ\)」](#)を参照してください。

ブートプロセスの概要については、[第 9 章「システムのシャットダウンとブート \(概要\)」](#)を参照してください。

SPARC システムのブート手順については、[第 12 章「Solaris システムのブート \(手順\)」](#)を参照してください。

x86: システムのブート (作業マップ)

作業	説明	参照先
x86 システムを実行レベル 3 でブートします。	実行レベル 3 でブートする - システムをシャットダウンするか、なんらかのシステムハードウェアの保守作業を行なった後で使用します。	336 ページの「x86: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法」
x86 システムをシングルユーザーモードでブートします。	実行レベル S でブートする - システム保守作業 (ファイルシステムのバックアップなど) を行なったあとで使用します。	339 ページの「x86: システムを実行レベル S (シングルユーザーレベル) でブートする方法」

作業	説明	参照先
x86 システムを対話式でブートします。	対話式でブートする - テストのために、システムファイルまたはカーネルを一時的に変更したあとで使用します。	340 ページの「x86: システムを対話式でブートする方法」
x86 システムをネットワークからブートします。	PXE または非 PXE デバイスをデフォルトのネットワーク構成方法でネットワークからブートするときに使用します。この方法は、ディスクレスクライアントをブートする場合に使用します。	343 ページの「x86: システムをネットワークからブートする方法」
Solaris 10: Solaris オペレーティングシステム (x86 システム) においてデバイス構成用補助を使用します。 注 - Solaris 10 1/06 リリース以降では、Device Configuration Assistant (デバイス構成用補助) が GRUB メニューで置き換えられました。	システムのハードウェア構成を変更したあとで使用します。このユーティリティを使用すると、異なるブートデバイスから Solaris システムをブートする、新しいハードウェアを構成する、間違って構成したハードウェアを構成し直すなど、デバイス関連やブート関連の作業を行うことができます。	346 ページの「x86: Device Configuration Assistant (デバイス構成用補助) を表示する方法」

作業	説明	参照先
復旧を目的としてシステムをブートします。	復旧を目的としてブートする - 損傷したファイルによってシステムのブートが行えないときにシステムをブートするために使用します。復旧を目的としてブートするには、必要に応じて次のどちらかまたは両方の作業を行います。 1. まず、システムを停止して復旧を試みます。 2. クラッシュダンプを強制実行し、システムをリブートする - トラブルシューティングの目的でクラッシュダンプを強制実行するときに使用します。 3. システムをブートして、システムの正常なブートを妨げている重要なシステムファイルを修復します。	346 ページの「x86: 復旧を目的としてシステムを停止する方法」 352 ページの「x86: クラッシュダンプを強制してシステムをリブートする」 347 ページの「x86: 復旧を目的としてシステムをブートする方法」
	kmdb をブートする - システムに関する問題のトラブルシューティングを行うときに使用します。	349 ページの「x86: カーネルデバッグ (kmdb) を使ってシステムをブートする方法」 システムを対話式でデバッグする時間がない場合は、-d オプションを指定して reboot コマンドと halt コマンドを使用します。-d オプションを指定して halt コマンドを実行した場合、実行後にシステムを手動でリブートする必要があります。これに対し、reboot コマンドを使用した場合、システムは自動的にブートします。
64 ビットコンピューティングに対応するシステムにおけるブート問題のトラブルシューティングを行います。	64 ビットモードに対応していないデバイスドライバを読み込むシステムを要求するハードウェアがある場合、システムを 64 ビットモードでブートできない可能性があります。この場合、システムを 32 ビットモードでブートする必要があります。	354 ページの「x64: 64 ビットブートの失敗のトラブルシューティング」

x86: GRUB を実装しないシステムをブートする

次に、リセットボタンを使ってシステムを再起動する手順を示します。システムにリセットボタンがない場合は、電源スイッチを使ってシステムを再起動します。システムの状態によっては Ctrl + Alt + Del キーを使って、システムの動作に割り込むことができます。

▼ x86: システムを実行レベル 3 (マルチユーザーレベル) でブートする方法

次の手順に従って、現時点で実行レベル 0 になっているシステムを実行レベル 3 でブートします。

- 1 Press any key to reboot プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。
あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。
「Current Boot Parameters」メニューが、数分後に表示されます。
- 2 システムを実行レベル 3 でブートするには、b と入力します。Enter キーを押します。5 秒以内に選択しない場合、システムは自動的に実行レベル 3 でブートします。
- 3 システムが実行レベル 3 になっていることを確認します。
ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示されます。

hostname console login:

例 16-1 x86: システムを実行レベル 3 (マルチユーザーレベル) でブートする

Solaris OS を新しくインストールした場合、ブートプロンプトで b と入力すると、64 ビット対応の x86 システムは自動的に 64 ビットモードでブートします。Solaris OS にアップグレードした場合も、eeprom の boot-file パラメータが kernel/unix 以外の値にあらかじめ設定されていないかぎり、ブートプロンプトで b と入力すると、64 ビット対応の x86 システムは 64 ビットモードでブートします。

次の例は、64 ビットコンピューティング対応の x86 システムを実行レベル 3 でブートする方法を示しています。

Press any key to reboot

·
·
·

<<< Current Boot Parameters >>>

Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a

Boot args:

Type	b [file-name] [boot-flags] <ENTER>	to boot with options
or	i <ENTER>	to enter boot interpreter
or	<ENTER>	to boot with defaults

<<< timeout in 5 seconds >>>


```

Select (b)oot or (i)nterpreter: b
SunOS Release 5.10 Version amd64-gate-2004-09-27 64-bit
Copyright 1983-2004 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
DEBUG enabled
Hostname: venus
NIS domain name is example.com
checking ufs filesystems
/dev/rdisk/c1d0s7: is logging.
venus console login:

```

例 16-2 x64: 64 ビットコンピューティング対応のシステムを、手動で、64 ビットモードかつ実行レベル 3 (マルチユーザーレベル) でブートする

Solaris OS を新しくインストールした場合、ブートプロンプトで **b** と入力すると、64 ビット対応の x86 システムは自動的に 64 ビットモードでブートします。Solaris OS にアップグレードした場合も、`eeeprom` の `boot-file` パラメータが `kernel/unix` 以外の値にあらかじめ設定されていないかぎり、ブートプロンプトで **b** と入力すると、64 ビット対応の x86 システムは 64 ビットモードでブートします。

次の例は、このタイプのシステムを「手動で」64 ビットモードかつ実行レベル 3 でブートする方法を示しています。

```

# init 0
# svc.startd: The system is coming down. Please wait.
svc.startd: 68 system services are now being stopped.
umount: /etc/svc/volatile busy
svc.startd: The system is down.
syncing file systems... done
Press any key to reboot.

Initializing system
Please wait...

<<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:

Type   b [file-name] [boot-flags] <ENTER>   to boot with options
or     i <ENTER>                           to enter boot interpreter
or     <ENTER>                             to boot with defaults

<<< timeout in 5 seconds >>>

```

```

Select (b)oot or (i)nterpreter: b kernel/amd64/unix

```

```

SunOS Release 5.10 Version amd64-gate-2004-09-27 64-bit
Copyright 1983-2004 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
DEBUG enabled
Hostname: venus
NIS domain name is example.com
checking ufs filesystems
/dev/rdisk/c1d0s7: is logging.
venus console login:

```

例 16-3 32 ビット x64:64 ビットコンピューティング対応のシステムを、手動で、32 ビットモードかつ実行レベル3(マルチユーザーレベル)でブートする

Solaris OS を新しくインストールした場合、ブートプロンプトで **b** と入力すると、64 ビット対応の x86 システムは自動的に64 ビットモードでブートします。Solaris OS にアップグレードした場合も、eeprom の `boot-file` パラメータが `kernel/unix` 以外の値にあらかじめ設定されていないかぎり、ブートプロンプトで **b** と入力すると、64 ビット対応の x86 システムは64 ビットモードでブートします。

次の例は、このタイプのシステムを「手動で」32 ビットモードかつ実行レベル3でブートする方法を示しています。

```

# init 0
# svc.startd: The system is coming down. Please wait.
svc.startd: 68 system services are now being stopped.
umount: /etc/svc/volatile busy
svc.startd: The system is down.
syncing file systems... done
Press any key to reboot.
Resetting...
If the system hardware has changed, or to boot from a different
device, interrupt the autoboot process by pressing ESC.

Initializing system
Please wait...

<<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:

Type   b [file-name] [boot-flags] <ENTER>    to boot with options
or     i <ENTER>                          to enter boot interpreter
or     <ENTER>                            to boot with defaults

```

```
<<< timeout in 5 seconds >>>
```

```
Select (b)oot or (i)nterpreter: b kernel/unix
SunOS Release 5.10 Version amd64-gate-2004-09-30 32-bit
Copyright 1983-2004 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
DEBUG enabled
Hostname: venus
NIS domain name is example.com
checking ufs filesystems
/dev/rdisk/c1d0s7: is logging.
venus console login:
```

▼ x86: システムを実行レベル S (シングルユーザーレベル) でブートする方法

次の手順に従って、現時点で実行レベル 0 になっているシステムを実行レベル S でブートします。

- 1 Press any key to reboot プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。
あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。

「Current Boot Parameters」メニューが、数分後に表示されます。

- 2 システムを実行レベル S でブートするには、**b -s** と入力して、**Enter** キーを押します。
5 秒以内に選択しない場合、システムは自動的に実行レベル 3 でブートします。
- 3 プロンプトが表示されたら、スーパーユーザーのパスワードを入力します。
- 4 システムが実行レベル S になっていることを確認します。

```
# who -r
.          run-level S  Jul 19 14:37      S      0  3
```

- 5 実行レベルを S に移行して行う必要がある保守作業を行います。
- 6 システム保守作業が完了したら、**Ctrl+D** キーを押してシステムをマルチユーザー状態にします。

例 16-4 x86: システムを実行レベル S (シングルユーザーレベル) でブートする

```

Press any key to reboot.
Resetting...
.
.
.
Initializing system
Please wait...

<<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:

Type    b [file-name] [boot-flags] <ENTER>    to boot with options
or      i <ENTER>                             to enter boot interpreter
or      <ENTER>                             to boot with defaults

<<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b -s
SunOS Release 5.10 Version amd64-gate-2004-09-30 32-bit
Copyright 1983-2004 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
DEBUG enabled
Booting to milestone "milestone/single-user:default".
Hostname: venus
NIS domain name is example.com
Requesting System Maintenance Mode
SINGLE USER MODE

Root password for system maintenance (control-d to bypass): xxxxxx
Entering System Maintenance Mode
.
.
.
# who -r
.      run-level S  Jul 19 14:37      S      0  3
      (Perform some maintenance task)
# ^D

```

▼ x86: システムを対話式でブートする方法

代替のカーネルまたは /etc/system ファイルを指定する必要がある場合、次の手順に従ってシステムをブートします。

- 1 Press any key to reboot プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。
あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。
「Primary Boot Subsystem」メニューが、数分後に表示されます。
- 2 **Solaris** パーティションにアクティブを示すマークが付いていない場合、その **Solaris** パーティションをリストから選択します。Enter キーを押します。
5 秒以内に選択しない場合、アクティブなブートパーティションが自動的に選択されます。
「Current Boot Parameters」メニューが、数分後に表示されます。
- 3 b -a と入力して、システムを対話式でブートします。Enter キーを押します。
5 秒以内に選択しない場合、システムは自動的に実行レベル 3 でブートします。
- 4 システムプロンプトに次のように教えてください。
 - a. プロンプトが表示されたら、ブートに使用するカーネルの名前を入力します。
デフォルトのカーネルファイル名を使用する場合は、Enter キーを押します。そうでない場合は、代替カーネルの名前を入力して、Enter キーを押します。
 - b. プロンプトが表示されたら、モジュールディレクトリの代替パスを入力します。
デフォルトのモジュールディレクトリを使用する場合は、Enter キーを押します。そうでない場合は、モジュールディレクトリへの代替パスを入力して、Enter キーを押します。
 - c. プロンプトが表示されたら、代替システムファイルの名前を入力します。
/etc/system ファイルが破損している場合、/dev/null を入力します。
 - d. プロンプトが表示されたら、root ファイルシステムのタイプを入力します。
UPS によるローカルディスクブート (デフォルト) を選択する場合は Enter キーを押し、ネットワークブートを選択する場合は NFS と入力します。
 - e. プロンプトが表示されたら、ルートデバイスの物理名を入力します。
代替デバイス名を入力します。デフォルトを使用する場合は Return キーを押します。
- 5 これらの質問に応答するためのプロンプトが表示されない場合は、boot -a コマンドを正しく入力しているかどうか確認してください。

例 16-5 x86: システムを対話式でブートする

次の例では、利用できるデフォルトの選択例 ([] で囲まれた部分) を示します。

```

Press any key to reboot.
Resetting...
.
.
.
Autobooting from bootpath: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a

If the system hardware has changed, or to boot from a different
device, interrupt the autoboot process by pressing ESC.

Initializing system
Please wait...

<<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:

Type    b [file-name] [boot-flags] <ENTER>    to boot with options
or      i <ENTER>                               to enter boot interpreter
or      <ENTER>                               to boot with defaults
Running Configuration Assistant...
<<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b -a
Enter default directory for modules [/platform/i86pc/kernel /kernel /usr/kernel]:
    Press Enter
Name of system file [etc/system]:      Press Enter
SunOS Release 5.10 Version amd64-gate-2004-09-30 32-bit
Copyright 1983-2004 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
DEBUG enabled
root filesystem type [ufs]:      Press Enter
Enter physical name of root device[/pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a]:
                                                    Press
                                                    Enter

Hostname: venus
NIS domain name is example.com
checking ufs filesystems
/dev/rdisk/cld0s7: is logging.
venus console login:

```

x86: ネットワークからのブート

ブートサーバーが利用できれば、どのようなシステムもネットワークからブートできます。スタンドアロンのシステムがローカルディスクからブートできない場合、そのシステムを復旧目的でネットワークからブートできます。

PXE (Preboot Execution Environment) ネットワークブートプロトコルをサポートしている x86 システムでは、Solaris ブートフロッピーディスクを使用しなくても、ネットワークから直接 Solaris x86 システムをブートできます。PXE ネットワークブートは、Intel の PXE 仕様を実装しているデバイスでのみ動作します。システムが PXE ネットワークブートに対応している場合は、Device Configuration Assistant (デバイス構成用補助) ブートフロッピーディスクや Solaris SOFTWARE - 1 CD または DVD を使用しなくても、ネットワークから直接システムをブートできます。

▼ x86: システムをネットワークからブートする方法

ここに示す手順には、Solaris Device Configuration Assistant (デバイス構成用補助) を使用してネットワークから x86 システムをブートする方法が含まれています。Solaris 10 以降のリリースでは、Device Configuration Assistant (デバイス構成用補助) の動作は変更されています。

Solaris 1/06 10/ リリース以降では、Solaris OS が動作する x86 システム上で GRUB ベースのブートが実装されています。Device Configuration Assistant (デバイス構成用補助) は GRUB メニューに置き換えられています。GRUB を使用したネットワークからの x86 システムのブートについては、[289 ページの「x86 システムをネットワークからブートする」](#)を参照してください。

ネットワークの構成方法には、RARP (Reverse Address Resolution Protocol)、DHCP (Dynamic Host Configuration Protocol) の 2 種類があります。PXE ネットワークブートのデフォルトのネットワークブート方法は、DHCP です。非 PXE デバイスのデフォルトのネットワークブート方法は、RARP です。非 PXE デバイスの場合、ネットワーク上で RARP ブートサーバー、DHCP ブートサーバーのいずれが利用可能であるかに応じて、いずれかの方法を使用できます。

注 - PXE ネットワークブートに DHCP サーバーを使用する場合は、DHCP の設定を追加する必要があります。DHCP の設定の概要については、『[Solaris のシステム管理 \(IP サービス\)](#)』のパート III 「[DHCP](#)」を参照してください。インストールに対応するように DHCP サーバーを設定する場合は、『[Solaris 10 インストールガイド \(ネットワークインストール\)](#)』を参照してください。

Solaris 10 リリースでは、PXE ネットワークブートを実行する場合、あるいは Solaris SOFTWARE - 1 CD または DVD からシステムをブートする場合、システムは自動的にブートします。Device Configuration Assistant (デバイス構成用補助) のメニューはデ

フォルトで表示されなくなりました。非 PXE デバイスをブートする場合、次の手順内で、ネットワーク構成を変更するために Device Configuration Assistant (デバイス構成用補助) のメニューを表示する方法を記述した部分の手順に従う必要があります。

- 1 **Device Configuration Assistant** (デバイス構成用補助) ブートフロッピーディスクまたは、**Solaris SOFTWARE - 1 CD** または **DVD** を適切なドライブに挿入します。あるいは、システムまたはネットワークアダプタの **BIOS** 構成プログラムを使用して、**PXE** ネットワークブートを使用可能にします。

- ブートフロッピーディスクを使用する場合、Device Configuration Assistant (デバイス構成用補助) の最初のメニューが表示されます。
- Solaris SOFTWARE - 1 CD または DVD を使用するか、ネットワークから PXE デバイスをブートする場合、システムは自動的にブートします。

ネットワーク構成を変更し、Device Configuration Assistant (デバイス構成用補助) のメニューを表示させたい場合、次のメッセージが表示されたときに Esc キーを押します。

If the system hardware has changed, or to boot from a different device, interrupt the autoboot process by pressing ESC.

Press ESCape to interrupt autoboot in 5 seconds.

「Device Configuration Assistant」画面が表示されます。

- 2 Press any key to reboot プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。

あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。

- 3 「F2_Continue」(F2 キー)を押して、デバイスを走査します。
デバイスの識別処理が実行されます。続いて、「Identified Devices」画面が表示されます。

- 4 「F2_Continue」(F2 キー)を押して、ドライバを読み込みます。
ブート可能なドライバが読み込まれます。続いて、「Boot Solaris」メニューが表示されます。

- 5 **Device Configuration Assistant** (デバイス構成用補助) を使用してネットワーク構成を変更します。

- a. 「F4_Boot Tasks」(F4 キー)を押します。

- b. 「Set Network Configuration Strategy」を選択します。「F2_Continue」(F2 キー)を押します。

- c. 「RARP」または「DHCP」を選択して、「F2_Continue」(F2 キー)を押します。

注 - 上記手順は、非 PXE デバイスをネットワークからブートする場合にのみ適用されます。PXE ネットワークブートの場合は、デフォルトのネットワークブート方法である DHCP を使用する必要があります。

新しいネットワークブート方法を確認する画面が表示されます。ここで選択したネットワークブート方法は、以降、このフロッピーディスクでブートしたときのデフォルトのネットワークブート方法として保存されます。

- d. 「F3_Back」を押して、「Boot Solaris」メニューに戻ります。

- 6 「NET」をブートデバイスとして選択します。そして、「F2_Continue」を押して、ネットワークデバイスをブートします。
「Solaris boot option」画面が表示されます。

x86: Device Configuration Assistant (デバイス構成用補助) の使用

注 - この Solaris リリースでは、Device Configuration Assistant (デバイス構成用補助) は GRUB メニューに置き換えられています。この機能についての詳細は、[271 ページの「GRUB を使用して x86 システムをブートする \(作業マップ\)」](#)を参照してください。

Solaris 10: Device Configuration Assistant (デバイス構成用補助) (Solaris オペレーティングシステム x86 版) は、さまざまなハードウェア構成作業やブート作業を実行するためのプログラムです。

Device Configuration Assistant (デバイス構成用補助) のメニューにアクセスするには、次のいずれかの方法を使用します。

- Solaris ブートフロッピーディスク
- Solaris SOFTWARE - 1 CD または DVD
- PXE ネットワークブート
- Solaris OS がインストールされたハードディスク

この章で説明する手順では、Device Configuration Assistant (デバイス構成用補助) ブートフロッピーディスクを挿入して Device Configuration Assistant (デバイス構成用補助) をブートするように要求されることがあります。システムの BIOS が CD または DVD からのブートをサポートしている場合は、Solaris SOFTWARE - 1 CD または DVD を挿入して Device Configuration Assistant (デバイス構成用補助) をブートすることも可能です。

▼ x86: Device Configuration Assistant (デバイス構成用補助) を表示する方法

Solaris 10: この手順では、ブートプロセスに割り込んで Device Configuration Assistant (デバイス構成用補助) を表示する方法を示します。現在の Solaris リリースでは、Device Configuration Assistant (デバイス構成用補助) は GRUB メニューに置き換えられています。

1 システムをブートします。

- Device Configuration Assistant (デバイス構成用補助) ブートフロッピーディスクからブートした場合、数分後に Device Configuration Assistant (デバイス構成用補助) の最初のメニューが表示されます。
- Solaris SOFTWARE - 1 CD、DVD、またはハードディスクからブートしたか、PXE ネットワークブートを実行した場合、次のメッセージが表示されます。

If the system hardware has changed, or to boot from a different device, interrupt the autoboot process by pressing ESC.

Press ESCape to interrupt autoboot in 5 seconds.

Device Configuration Assistant (デバイス構成用補助) のメニューを表示させたい場合、Esc キーを押して自動ブートプロセスに割り込みをかけます。

「Device Configuration Assistant」メニューが表示されます。

2 Press any key to reboot プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。

あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。

▼ x86: 復旧を目的としてシステムを停止する方法

1 可能であれば、次のいずれかのコマンドを使用してシステムを停止します。

- システムが動作している場合、スーパーユーザーになって `init 0` と入力してシステムを停止します。Press any key to reboot プロンプトが表示されたら、任意のキーを押してシステムをリブートします。
- システムが動作している場合、スーパーユーザーになって `init 6` と入力してシステムをリブートします。

- 2 マウスまたはキーボードからの入力にシステムが応答しない場合、リセットキーがあればそのキーを押してシステムをリブートします。あるいは、電源スイッチを使用してシステムをリブートします。

▼ x86: 復旧を目的としてシステムをブートする方法

重要なシステムリソースを修復するには、次の手順に従ってシステムをブートします。この例は、Solaris SOFTWARE - 1 CD またはネットワークからブートし、ルート (/) ファイルシステムをディスクにマウントし、/etc/passwd ファイルを修復する方法を示しています。

次の手順で使用されている *device-name* 変数は、修復するファイルシステムのデバイス名に置き換えてください。システムのデバイス名を調べる場合は、『Solaris のシステム管理 (デバイスとファイルシステム)』の「デバイス構成情報の表示」を参照してください。

- 1 システムのアボートキーシーケンスを使用して、システムを停止します。
システムのアボートキーシーケンスは、root のパスワードがわからない場合やシステムにログインできない場合に使用します。詳細は、346 ページの「x86: 復旧を目的としてシステムを停止する方法」を参照してください。
- 2 Solaris SOFTWARE - 1 CD または DVD から、あるいはネットワークから、システムをシングルユーザーモードでブートします。
 - a. **Device Configuration Assistant** (デバイス構成用補助) ブートフロッピーディスクまたは、Solaris SOFTWARE - 1 CD または DVD を適切なドライブに挿入します。

注- ブートフロッピーディスクを使用する場合、Device Configuration Assistant (デバイス構成用補助) のメニューが表示されます。Solaris SOFTWARE - 1 CD または DVD を使用する場合、システムは自動的にブートします。Device Configuration Assistant (デバイス構成用補助) のメニューを表示するには、システムによるプロンプトが表示されたときに Esc キーを押してブートプロセスに割り込みをかけます。

- b. Press any key to reboot プロンプトが表示されている場合は、任意のキーを押してシステムをリブートします。
あるいは、リセットボタンを使用することもできます。システムが停止している場合は、電源スイッチを押してシステムを起動します。
- 3 「Current Boot Parameters」メニューが、数分後に表示されます。
- 4 プロンプトで b -s と入力して、Enter キーを押します。
数分後に、シングルユーザーモードの # プロンプトが表示されます。

- 5 無効な `passwd` ファイルがあるルート (/) ファイルシステムをマウントします。
- 6 新しくマウントした `etc` ディレクトリに移動します。
- 7 エディタを使用して `passwd` ファイルに必要な変更を加えます。
- 8 ルート (/) ディレクトリに変更します。
- 9 /a ディレクトリのマウントを解除します。
- 10 システムを再起動します。システムが実行レベル 3 になっていることを確認します。
ブートプロセスが正常に終了すると、ログイン画面かログインプロンプトが表示されます。

host-name console login:

例 16-6 x86: Solaris 10: 復旧を目的としてシステムをブートする

次の例は、Solaris 10 OS で、ローカルの CD-ROM からシステムが自動的にブートしたあとに、`/etc/passwd` ファイルを修復する方法を示しています。GRUB ベースのブートは、Solaris 10 1/06 リリースで導入されました。GRUB ベースのブート環境での復旧を目的としたシステムのブートについては、[284 ページの「GRUB を使用して x86 システムでフェイルセーフアーカイブをブートする方法」](#)を参照してください。

SunOS Secondary Boot version 3.00

Solaris Booting System

Running Configuration Assistant...

If the system hardware has changed, or to boot from a different device, interrupt the autoboot process by pressing ESC.

Press ESCape to interrupt autoboot in 5 seconds.

```

Initializing system
Please wait...

```

```
<<< Current Boot Parameters >>>
```

```

Boot path: /pci@0,0/pci-ide@7,1/ide@1/sd@0,0:a
Boot args:

```

Select the type of installation you want to perform:

- 1 Solaris Interactive
- 2 Custom JumpStart
- 3 Solaris Interactive Text (Desktop session)
- 4 Solaris Interactive Text (Console session)

Enter the number of your choice followed by the <ENTER> key.
Alternatively, enter custom boot arguments directly.

If you wait for 30 seconds without typing anything,
an interactive installation will be started.

Select type of installation: **b -s**

```

.
.
.
# mount /dev/dsk/c0t0d0s0 /a
.
.
.
# cd /a/etc
# vi passwd
    (Remove invalid entry)
# cd /
# umount /a
# init 6

```

▼ x86: カーネルデバッガ (kldb) を使ってシステムをブートする方法

この手順では、Solaris 10 OS においてカーネルデバッガ (kldb) を読み込むための基本的な操作を示します。デフォルトでは、`savecore` 機能を使用できます。カーネルデバッガの使用法の詳細については、『[Solaris モジュールデバッガ](#)』を参照してください。

現在の Solaris リリースにおいて、カーネルデバッグを使用してシステムをブートする手順については、[304 ページの「x86: GRUB ブート環境でカーネルデバッグ \(kmdb\) を使ってシステムをブートする方法」](#)を参照してください。

- 1 システムをブートします。
- 2 Select (b)oot or (i)nterpreter プロンプトで **b -k** と入力します。**Enter** キーを押します。
- 3 カーネルデバッグにアクセスします。
デバッグを入力するための方法は、システムのアクセスに使用するコンソールのタイプによって異なります。
 - ローカル接続されたキーボードを使用している場合、F1 + A キーを押します。
 - シリアルコンソールを使用している場合、そのシリアルコンソールの種類に合った方法を使って改行を送信します

カーネルデバッグに初めてアクセスした場合、ウェルカムメッセージが表示されます。

例 16-7 x86: システムをカーネルデバッグ (kmdb) でブートする

Select (b)oot or (i)nterpreter ブートプロンプトで **-b k** と入力すると、システムはデフォルトのモードでブートして **kmdb** を読み込みます。次の例は、32 ビットコンピューティング対応の x86 システムを 32 ビットモードでブートして **kmdb** を読み込む方法を示しています。

```
Press any key to reboot.
.
.
.
<<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:

Type    b [file-name] [boot-flags] <ENTER>    to boot with options
or      i <ENTER>                             to enter boot interpreter
or      <ENTER>                             to boot with defaults
Running Configuration Assistant...
          <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b -k
Loading kmdb...
SunOS Release 5.10 Version gate:2004-10-21 32-bit
Copyright 1983-2004 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
```

```

.
.
.

```

例 16-8 x64: 64 ビットコンピューティング対応のシステムを、カーネルデバッグ (kmdb) を使って 64 ビットモードで手動ブートする

次の例は、64 ビットコンピューティング対応の x86 システムを、kmdb を使って 64 ビットモードで手動ブートする方法を示しています。

```

Press any key to reboot

.
.
.

<<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
Type      b [file-name] [boot-flags] <ENTER>    to boot with options
or        i <ENTER>                               to enter boot interpreter
or        <ENTER>                                to boot with defaults

<<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b kernel/amd64/unix -k
Loading kmdb...

```

例 16-9 32 ビット x64: 64 ビットコンピューティング対応のシステムを、カーネルデバッグ (kmdb) を使って 32 ビットモードで手動ブートする

次の例は、64 ビットコンピューティング対応の x86 システムを、kmdb を使って 32 ビットモードで手動ブートする方法を示しています。

```

Press any key to reboot

.
.
.

<<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
Type      b [file-name] [boot-flags] <ENTER>    to boot with options
or        i <ENTER>                               to enter boot interpreter
or        <ENTER>                                to boot with defaults

<<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b kernel/unix -k
Loading kmdb...

```

x86: クラッシュダンプを強制してシステムをリブートする

場合によっては、トラブルシューティングの目的でクラッシュダンプを強制実行し、システムをリブートする必要があります。デフォルトでは、`savecore` 機能を使用できます。

システムのクラッシュダンプについては、『[Solaris のシステム管理 \(上級編\)](#)』の第 17 章「[システムクラッシュ情報の管理 \(手順\)](#)」を参照してください。

▼ x86: クラッシュダンプを強制してシステムをリブートする方法

`reboot -d` コマンドまたは `halt -d` コマンドを使用できない場合は、カーネルデバッガ `kldb` を使用してクラッシュダンプを強制的に実行できます。次の手順を実行するには、ブート時または `mdb -k` コマンド経由でカーネルデバッガが読み込まれている必要があります。

注 - カーネルデバッガ (`kldb`) に入るにはテキストモードでなければなりません。したがって、まずウィンドウシステムを終了してください。

- 1 ローカル接続されたキーボードをシステムコンソールとして使用している場合、そのキーボード上で **F1+A** キーを押します。リモート (シリアル) コンソールを使用するようにシステムが設定されている場合、そのコンソールに合った機構を使って改行文字を送信します。
`kldb` プロンプトが表示されます。
- 2 `systemdump` マクロを使ってクラッシュを誘発します。

```
[0]> $<systemdump
```


パニックメッセージが表示され、クラッシュダンプが保存され、システムがリブートします。
- 3 コンソールログインプロンプトでログインして、システムがリブートされていることを確認します。

例 16-10 x86: `halt -d` コマンドを使用してシステムのクラッシュダンプとリブートを強制実行する

この例は、`halt -d` コマンドと `boot` コマンドを使って、x86 システム `neptune` のクラッシュダンプとリブートを強制実行する方法を示しています。システムのクラッシュダンプを強制するには、この方法を使用してください。-d オプションを指定して `halt` コマンドを実行したあと、手動でリブートする必要があります。


```
# halt -d
Aug 11 12:51:27 neptune halt:
halted by <user> panic[cpu45]/thread=d3971a00: forced crash dump initiated at user request

d363ae58 genunix:kadmin+bd (5, 0, 0, d3fefac0)
d363af88 genunix:uadmin+88 (5, 0, 0, 0, 0, d363afb4)

syncing file systems... done
dumping to /dev/dsk/c0t0d0s1, offset 107806720, content: kernel
100% done: 40223 pages dumped, compression ratio 4.11, dump succeeded
Press any key to reboot.
Resetting...
.
.
.
SunOS Secondary Boot version 3.00
Autobooting from bootpath: /pci@0,0/pci1028,10a@3/sd@0,0:a
Running Configuration Assistant...
If the system hardware has changed, or to boot from a different
device, interrupt the autoboot process by pressing ESC.

Initializing system
Please wait...

<<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci1028,10a@3/sd@0,0:a
Boot args:

Type      b [file-name] [boot-flags] <ENTER>    to boot with options
or        i <ENTER>                             to enter boot interpreter
or        <ENTER>                             to boot with defaults

                <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter:
Loading kmdb...
SunOS Release 5.10 Version s10_62 32-bit
Copyright 1983-2004 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
configuring IPv4 interfaces: iprb0.
add net default: gateway 172.20.26.248
Hostname: neptune
The system is coming up. Please wait.
checking ufs filesystems
/dev/rdisk/c0t0d0s7: is logging.
NIS domain name is example.com
starting rpc services: rpcbind keyserv ypbind done.
```

```
Setting netmask of iprb0 to 255.255.255.0
Setting default IPv4 interface for multicast: add net 224.0/4: gateway venus
syslog service starting.
System dump time: Wed Aug 11 12:51:29 2004
Aug 11 13:13:26 venus savecore: saving system crash dump in /var/crash/venus/*.1
Constructing namelist /var/crash/venus/unix.1
Constructing corefile /var/crash/venus/vmcore.1
100% done: 42157 of 42157 pages saved
volume management starting.
The system is ready.
.
.
.
```

x64: 64 ビットブートの失敗のトラブルシューティング

場合によって、64 ビット対応の x86 システムが 64 ビットモードでのブートに失敗することがあります。この失敗によって、次のようなエラーが生成されます。

```
Select (b)oot or (i)nterpreter: b kernel/amd64/unix
.
.
.
pci: cannot load driver
Cannot load drivers for /pci@0,0/pci1022,7450@a/pci17c2,10@4/sd@0,0:a
(Can't load the root filesystem)
Press any key to reboot.
.
.
.
```

このような失敗が起きる場合は、Select (b)oot or (i)nterpreter ブートプロンプトで次のコマンドを入力して、システムを 32 ビットモードでブートしてください。

```
Select (b)oot or (i)nterpreter: b kernel/unix
```

詳細は、[例 16-3](#) を参照してください。

x86: ブートプロセス (参照情報)

次の節には、GRUB ベースのブートを実装しない Solaris x86 システムのブートに関連する参照情報が記載されています。

注 - この Solaris リリースでは、Solaris Device Configuration Assistant (デバイス構成用補助) が GRUB メニューに置き換えられました。この Solaris リリースでの x86 システムのブートの詳細は、[271 ページの「GRUB を使用して x86 システムをブートする \(作業マップ\)」](#)を参照してください。

x86: ブートサブシステム

ブートプロセス中は、ブートサブシステムメニューは異なるデバイスとブートオプションを表示します。システムは何回かのタイムアウトのあとで応答を受け付けなくなった場合、デフォルトの設定値を使って自動的にブートを継続します。ブートサブシステムメニューが表示されるたびに、ブートプロセスを停止することができます。また、自動的に継続させることもできます。

Solaris のブート時に、ブートシステムについて次の選択を行えます。

- 一次ブートサブシステム (「Partition Boot」メニュー) - この最初のメニューは、複数のオペレーティングシステムがディスク上にある場合に表示されます。このメニューで、インストールされたどのオペレーティングシステムからブートするかを指定できます。デフォルトでは、アクティブなオペレーティングシステムからブートします。
Solaris オペレーティングシステム以外のシステムからブートする場合、次の2つのメニューは表示されません。
- 自動ブートプロセスへの割り込み - 自動ブートプロセスに割り込むと、Device Configuration Assistant (デバイス構成用補助) のメニューにアクセスできます。
Solaris Device Configuration Assistant (デバイス構成用補助) を使用すると、異なるブートデバイスから Solaris システムをブートする、新しいハードウェアを構成する、間違って構成したハードウェアを構成し直すなど、デバイス関連やブート関連の作業を実行できます。
- 「Current Boot Parameters」メニュー - このメニューには2つの形式があります。1つは、通常の Solaris ブート用のメニューで、もう1つは Solaris インストールブート用のメニューです。
 - 通常の「Current Boot Parameters」メニューでは、オプションを付けて Solaris システムをブートしたり、ブートインタプリタを実行したりできます。
 - インストール用の「Current Boot Parameters」メニューでは、インストールの種類を選択したり、ブートプロセスをカスタマイズしたりできます。

次の表に、主な x86 システムのブートインタフェースの目的を示します。以降の節では、各ブートインタフェースを詳細に説明し、例を示します。

表 16-1 x86: ブートサブシステム

ブートサブシステム	目的
一次ブートサブシステム (「Partition Boot」メニュー)	このメニューは、現在ブート中のディスクに複数のオペレーティングシステム (Solaris オペレーティングシステム (Solaris OS) を含む) が含まれている場合に表示されます。
二次ブートサブシステム	Solaris リリースをブートするたびに表示されます。自動ブートプロセスに割り込んで、Solaris Device Configuration Assistant (デバイス構成用補助) を実行することを選択していない限り、Solaris リリースが自動的にブートされます。
Solaris Device Configuration Assistant (デバイス構成用補助) ブートフロッピーディスク	「Device Configuration Assistant (デバイス構成用補助)」メニューを表示するには次の 2 つの方法があります。 ■ Device Configuration Assistant (デバイス構成用補助) ブートフロッピーディスクまたは Solaris SOFTWARE - 1 CD (CD-ROM ドライブからブートできるシステムのみ) を使用してシステムをブートします。 ■ インストールしたディスクから Solaris ソフトウェアをブートするときに自動ブートプロセスに割り込みます。
「Current Boot Parameters」メニュー	このメニューは、Solaris をディスク、CD-ROM、またはネットワークからブートするときに使用します。このメニューにはブートオプションが表示されます。

注 - Solaris Device Configuration Assistant (デバイス構成用補助) ブートフロッピーディスクを作成する必要がある場合は、http://www.sun.com/bigadmin/hcl/drivers/dca_diskettes/ を参照してください。

x86: Solaris リリースのブート

このリリースでは、Solaris SOFTWARE - 1 CD または DVD を使って x86 システムをブートする場合や PXE ネットワークブートを実行する場合には、システムが自動的にブートされます。Device Configuration Assistant (デバイス構成用補助) を使用するには、システムプロンプトが表示された際に、Esc キーを押してブートプロセスに割り込む必要があります。

デバイスを識別する段階では、Device Configuration Assistant (デバイス構成用補助) は次の処理を実行します。

- システムにインストールされているデバイスを走査します。
- 識別されたデバイスを表示します。

- キーボードの選択やデバイスとそのリソースの編集などのオプションの作業を可能にします。

ブートの段階では、Device Configuration Assistant (デバイス構成用補助) は次の処理を実行します。

- ブートするデバイスのリストを表示します。アスタリスク (*) が付いたデバイスがデフォルトのブートデバイスです。
- 自動ブート設定やプロパティ設定の編集、ネットワーク構成方法の選択などのオプションの作業を可能にします。

次の節では、デバイスを識別する段階で表示されるメニューの例を示します。デバイス出力は、各システム構成によって異なります。

x86: デバイス識別段階で表示される画面

Device Configuration Assistant (デバイス構成用補助) がシステム上のデバイスを識別するときに、いくつかの画面が表示されます。

この節では、次のブートサブシステム画面の例を示します。

- Device Configuration Assistant 画面
- Bus Enumeration 画面
- Scanning Devices 画面
- Identified Devices 画面

x86: Device Configuration Assistant 画面

注 - 現在の Solaris リリースでは、x86 システムにおいて「Device Configuration Assistant」画面が GRUB メニューに置き換えられています。詳細は、[271 ページ](#)の「GRUB を使用して x86 システムをブートする (作業マップ)」を参照してください。

Solaris 10 の初期のリリースでは、ブート段階でシステムプロンプトが表示された際にユーザーが Esc キーを押さない限り、自動ブートプロセスで Device Configuration Assistant (デバイス構成用補助) メニューが省略されます。Device Configuration Assistant (デバイス構成用補助) を使用する選択を行なった場合、次の画面が表示されます。

Solaris Device Configuration Assistant

```
The Solaris(TM)Device Configuration Assistant
scans to identify system hardware, lists identified devices, and can
boot the Solaris software from a specified device. This program must be
used to install the Solaris operating environment, add a driver,
```

or change the hardware on the system.

> To perform a full scan to identify all system hardware, choose Continue.
> To diagnose possible full scan failures, choose Specific Scan.
> To add new or updated device drivers, choose Add Driver.

About navigation...

- The mouse cannot be used.
- If the keyboard does not have function keys or they do not respond, press ESC. The legend at the bottom of the screen will change to show the ESC keys to use for navigation.
- The F2 key performs the default action.

F2_Continue

F3_Specific Scan

F4_Add Driver

F6_Help

x86: Bus Enumeration 画面

この画面は、Device Configuration Assistant (デバイス構成用補助) が自動検出できるデバイスのハードウェア構成データを集めているときに一時的に表示されます。

Bus Enumeration

Determining bus types and gathering hardware configuration data ...

Please wait ...

x86: Scanning Devices 画面

この画面は、特別なドライバだけで検出できるデバイスを Device Configuration Assistant (デバイス構成用補助) が手動で走査しているときに表示されます。

Scanning Devices

The system is being scanned to identify system hardware.

If the scanning stalls, press the system's reset button. When the system reboots, choose Specific Scan or Help.

Scanning: Floppy disk controller

#####

0		20	40	60	80	100

Please wait ...

x86: Identified Devices 画面

この画面は、システムで識別されたデバイスを表示します。この画面からは、「Boot Solaris」メニューに移動できます。

また、次のオプションのデバイス作業も実行できます。

- キーボード構成の設定
- デバイスの表示と編集
- シリアルコンソールの設定
- 構成の保存と削除

Identified Devices

The following devices have been identified on this system. To identify devices not on this list or to modify device characteristics, such as keyboard configuration, choose Device Tasks. Platform types may be included in this list.

ISA: Floppy disk controller

ISA: Motherboard

ISA: PnP bios: 16550-compatible serial controller

ISA: PnP bios: 16550-compatible serial controller

ISA: PnP bios: Mouse controller

ISA: PnP bios: Parallel port

ISA: System keyboard (US-English)

PCI: Bus Mastering IDE controller

PCI: Universal Serial Bus

PCI: VGA compatible display adapter

F2_Continue F3_Back F4_Device Tasks F6_Help

x86: ブート段階で表示されるメニュー

注 - Solaris 10 1/06 リリース以降では、システムのブート時に GRUB が表示されます。GRUB ベースのブートについての詳細は、[271 ページの「GRUB を使用して x86 システムをブートする \(作業マップ\)」](#)を参照してください。

この段階では、システムをブートする方法を指定できます。

ブート段階で表示されるメニューは次のとおりです。

- 「Boot Solaris」メニュー
- 「Current Boot Parameters」メニュー

x86: Boot Solaris メニュー

「Boot Solaris」メニューからは、Solaris リリースをブートするデバイスを選択できます。また、autoboot やプロパティ設定の表示や編集など、オプションの作業も実行できます。ブートデバイスを選択し、「Continue」を選択すると、Solaris カーネルがブートを開始します。

Boot Solaris

Select one of the identified devices to boot the Solaris kernel and choose Continue.

To perform optional features, such as modifying the autoboot and property settings, choose Boot Tasks.

An asterisk (*) indicates the current default boot device.

> To make a selection use the arrow keys, and press Enter to mark it [X].

```
[X] DISK: (*) Target 0:QUANTUM FIREBALL1280A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ] DISK: Target 1:ST5660A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ] DISK: Target 0:Maxtor 9 0680D4
    on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ] CD : Target 1:TOSHIBA CD-ROM XM-5602B 1546
    on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
```

F2_Continue F3_Back F4_Boot Tasks F6_Help

x86: 「Current Boot Parameters」メニュー

このメニューは、ローカルディスクから Solaris リリースをブートするたびに表示されます。デフォルトの Solaris カーネルからブートする場合は、5 秒間のタイムアウトが経過するのを待ちます。別のオプションでブートする場合は、タイムアウト期間が経過する前に、適切なオプションを選択します。

```
<<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
Type      b [file-name] [boot-flags] <ENTER>      to boot with options
or        i <ENTER>                                to enter boot interpreter
or        <ENTER>                                to boot with defaults
```

<<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter:

x86: ブートプロセス

次の表では、x86 システムのブートプロセスについて説明します。

表 16-2 x86: ブートプロセスの説明

ブート段階	説明
BIOS	1. システムの電源をオンにすると、BIOS がセルフテスト診断を実行してシステムのハードウェアとメモリーを検査します。エラーが検出されなければ、システムは自動的にブートを開始します。エラーが検出されると、復元オプションを示すエラーメッセージが表示されます。 追加のハードウェアデバイスの BIOS がここで実行されます。 2. BIOS ブートプログラムが、ブートデバイスの最初のディスクセクターを読もうとします。ブートデバイス上のこの最初のディスクセクターにはマスターブートレコード <code>mboot</code> が格納されており、このレコードが読み込まれて実行されます。 <code>mboot</code> ファイルが見つからなかった場合は、エラーメッセージが表示されます。
ブートプログラム	3. マスターブートレコード <code>mboot</code> には、アクティブなパーティションと、Solaris ブートプログラム <code>pboot</code> の位置を見つけるのに必要なディスク情報が格納されており、<code>mboot</code> は、<code>pboot</code> を読み込んで実行します。 4. Solaris ブートプログラム <code>pboot</code> が一次ブートプログラム <code>bootblk</code> を読み込みます。 <code>bootblk</code> の目的は、UFS ファイルシステム内に格納された二次ブートプログラムを読み込むことにあります。 5. ブート可能なパーティションが複数ある場合、<code>bootblk</code> は <code>fdisk</code> テーブルを読んでデフォルトのブートパーティションを探し、利用可能なパーティションのメニューを作成して表示します。ブートするパーティションを選択する際に、30 秒経過するとタイムアウトになります。これは、ブート可能なパーティションが複数ある場合にのみ発生します。 6. <code>bootblk</code> はルート (<code>/</code>) ファイルシステムで二次ブートプログラム <code>boot.bin</code> または <code>ufsboot</code> を探し、実行します。Solaris Device Configuration Assistant (デバイス構成用補助) を起動するときは、5 秒以内に自動ブートに割り込みをかけます。 7. 二次ブートプログラム <code>boot.bin</code> または <code>ufsboot</code> が <code>/etc/bootrc</code> スクリプトを実行するコマンドインタプリタを起動します。このスクリプトによってシステムのブート方法を選択するメニューが表示されます。デフォルトでは、カーネルが読み込まれ、実行されます。ブートオプションを指定したり、ブートインタプリタを起動したりするときは、5 秒以内に行います。

表 16-2 x86: ブートプロセスの説明 (続き)

ブート段階	説明
カーネル初期設定	8. カーネルは二次ブートプログラム <code>boot.bin</code> または <code>ufsboot</code> を使用してファイルを読み込み、カーネル自身を初期化して、モジュールの読み込みを開始します。カーネルはルート (<code>/</code>) ファイルシステムをマウントするのに十分なモジュールを読み込むと、二次ブートプログラムの対応づけを解除し、自身のリソースを使って動作を続けます。 9.カーネルはユーザープロセスを作成し、 <code>/sbin/init</code> プロセスを起動します。このプロセスは <code>/etc/inittab</code> ファイルを読み込んで他のプロセスを起動します。
init	10. この Solaris リリースでは、 <code>/sbin/init</code> プロセスが <code>/lib/svc/bin/svc.startd</code> を開始し、さらにその <code>svc.startd</code> が次の各処理を行うシステムサービスを開始します。 ■ ファイルシステムの検査とマウント ■ ネットワークとデバイスの設定 ■ 各種プロセスの起動とシステム保守作業の実行 さらに、 <code>svc.startd</code> は互換性を維持する目的で実行制御 (<code>rc</code>) スクリプトを実行します。

x86: ブートファイル

実行制御スクリプトおよびブートファイル以外にも、x86 システムのブートに関する追加ブートファイルがあります。

表 16-3 x86: ブートファイル

ファイル	説明
<code>/etc/bootrc</code>	Solaris リリースのブートに使用するメニューとオプションを含みます。
<code>/boot</code>	システムをブートするために必要なファイルとディレクトリを含みます。
<code>/boot/mdboot</code>	第 1 レベルのブートストラッププログラム (<code>strap.com</code>) をディスクからメモリーにロードする DOS 実行可能ファイル。
<code>/boot/mdbootbp</code>	第 1 レベルのブートストラッププログラム (<code>strap.com</code>) をフロッピーディスクからメモリーにロードする DOS 実行可能ファイル。
<code>/boot/rc.d</code>	インストールスクリプトを含むディレクトリ。内容の変更はできません。
<code>/boot/solaris</code>	ブートサブシステムの項目を含むディレクトリ。

表 16-3 x86: ブートファイル (続き)

ファイル	説明
/boot/solaris/boot.bin	Solaris カーネルまたはスタンドアロンの kmdb をロードします。この実行可能ファイルは、ブートファームウェアサービスも提供します。
/boot/solaris/boot.rc	x86 システム上の Solaris オペレーティングシステムを出力し、DOS エミュレーションモードで Device Configuration Assistant (デバイス構成用補助) を実行します。
/boot/solaris/bootconf.exe	Device Configuration Assistant (デバイス構成用補助) の DOS 実行可能ファイル。
/boot/solaris/bootconf.txt	Device Configuration Assistant (デバイス構成用補助) の国際化されたメッセージを含むテキストファイル (bootconf.exe)。
/boot/solaris/bootenv.rc	ブート環境の設定用の eeprom 変数を格納します。
/boot/solaris/devicedb	master ファイル (リアルモードドライバでサポートされるすべての有効なデバイスのデータベース) を含むディレクトリ。
/boot/solaris/drivers	リアルモードドライバを含むディレクトリ。
/boot/solaris/itup2.exe	インストール時更新 (ITU) プロセスで実行される DOS 実行可能ファイル。
/boot/solaris/machines	以前は使用されていたが、現在はもう使用されていないディレクトリ。
/boot/solaris/nbp	ネットワークブート関連ファイル。
/boot/solaris/strap.rc	どのロードモジュールをロードするか、メモリー内のどこにロードするかの指示を含むファイル。
/boot/strap.com	第2レベルのブートストラッププログラムをメモリーにロードする DOS 実行可能ファイル。

サービスの管理 (概要)

この章では、サービス管理機能 (SMF) の概要について説明します。さらに、実行レベルについても説明します。

この章の内容は以下のとおりです。

- 365 ページの「SMF について」
- 367 ページの「SMF の概念」
- 372 ページの「SMF 管理およびプログラミングインタフェース」
- 373 ページの「SMF コンポーネント」
- 374 ページの「SMF 互換性」
- 375 ページの「実行レベル」
- 377 ページの「/etc/inittab ファイル」

SMF に関連する手順については、379 ページの「サービスの管理 (作業マップ)」を参照してください。実行レベルに関連する手順については、396 ページの「実行制御スクリプトの使用 (作業マップ)」を参照してください。

SMF について

SMF は、従来の UNIX の起動スクリプト、init 実行レベル、および構成ファイルを補強するインフラストラクチャーを提供します。SMF には、次の機能が備わっています。

- 管理エラー、ソフトウェアのバグ、修復不可能なハードウェアエラーなどの原因によって失敗したサービスを、依存関係の順番に自動的に再起動します。依存関係の順番は依存記述で定義されます。
- 表示可能なサービスオブジェクトを新しいコマンド `svcs` を使って作成したり、管理可能なサービスオブジェクトをコマンド `svcadm` と `svccfg` を使って作成したりします。SMF サービスおよび従来の `init.d` スクリプトのどちらにおいても、`svcs -p` を使用すると、サービスとプロセスの関係を表示することができます。

- サービス構成のスナップショットを自動的に取ることで、バックアップ、復元、およびサービスへの変更の取り消しを容易にします。
- サービスが実行できない原因を `svcs -x` で明らかにし、デバッグおよびサービスに関する質問を容易にします。また、この処理は、各サービスの個別の永続的なログファイルを使用するとより楽に行えます。
- `svcadm` を使用してサービスを有効化および無効化します。この変更は、更新後やリブート後も維持できます。-t オプションを使用した場合、変更は一時的です。
- ルート以外のユーザーに対して管理者がタスクを安全に委任する機能を拡張します。この機能には、プロパティの変更と、システムのサービスの有効化、無効化、および再起動とが含まれます。
- 大規模システムにおけるブートを高速化します。これは、サービス間の依存関係に従って各サービスを並列的に起動することで実現しています。シャットダウン時にはその逆の処理が実行されます。
- ブートコンソールの出力をカスタマイズして、表示を最小限にする (デフォルト) か、表示を多くする (`boot -m verbose` を使用) かを選択できます。
- 可能な場合には既存の管理業務との互換性を維持します。たとえば、顧客および ISV から提供される rc スクリプトの大部分は、通常どおり動作します。

「依存記述」では、サービス間の関係を定義します。これらの関係を使用すると、すべてのサービスを再起動するのではなく、障害の影響を直接受けているサービスのみを再起動することにより、障害を的確に封じ込めることができます。依存記述のもう 1 つの利点は、スケーラブルで再現可能な初期化プロセスを実現できることです。また、依存性をすべて定義することにより、独立したすべてのサービスを並列に起動できるため、今日の高並列マシンをうまく利用することができます。

SMF では、管理者がサービスに対して呼び出すことのできる一連のアクションを定義します。これらのアクションには、有効化、無効化、再表示、再起動、維持などがあります。各サービスは、管理アクションを実行するサービスリスタートによって管理されます。通常、アクションを実行する場合、リスタートはサービスに対していくつかのメソッドを実行します。各サービスのメソッドは、サービス構成リポジトリで定義されます。リスタートは、これらのメソッドを使って、サービスのある状態から別の状態へ移行できます。

サービス構成リポジトリでは、フォールバックができるように、各サービスが正常に起動されたときにサービスごとのスナップショットを取ります。また、リポジトリを使用すると、一貫した永続的な方法でサービスを有効または無効にしたり、サービスの状態を一貫して表示したりできます。この機能は、サービスの構成に関する問題を修正するのに役立ちます。

SMF 使用時の動作に関する変更

SMF が提供する機能のほとんどが、ユーザーの目に触れることなく実行されます。それ以外の機能には新しいコマンドでアクセスします。動作に関する可視的な変更は次のとおりです。

- ブートプロセスで生成されるメッセージが少なくなりました。デフォルトでは、サービスの起動時にメッセージは表示されません。ブートメッセージによって提供されていた情報は、`/var/svc/log` にある各サービス用のログファイルで提供されるようになりました。ブートの問題の診断には `svcs` コマンドが役立ちます。なお、`boot` コマンドで `-v` オプションを使用すれば、ブートプロセス中に各サービスが起動されるたびにメッセージが生成されます。
- サービスは可能なかぎり自動的に再起動されるため、プロセスを終了させないように見えます。サービスに障害があれば保守モードに切り替わりますが、通常、対応するプロセスを強制終了してもサービスは再起動されます。SMF サービスが実行されないようにするには、`svcadm` コマンドを使用してそのプロセスを停止する必要があります。
- `/etc/init.d` および `/etc/rc*.d` のスクリプトが多数削除されました。サービスの有効化および無効化に、これらのスクリプトはもう必要ありません。`/etc/inittab` のエントリも削除され、サービスの管理に SMF が使用できるようになりました。ISV によって提供されるスクリプトおよび `inittab` エントリ、あるいはローカルで開発されたそれらは、従来どおり機能します。ブートプロセス中で、これらのサービスが起動される時点が以前と異なる可能性があります。必ず SMF サービスのあとに起動されるため、サービス依存関係に問題は起きません。

SMF の概念

この節では、SMF フレームワーク内で使われる用語とその定義をいくつか紹介します。これらの用語は、マニュアル全体で使用されます。SMF の概念を理解するには、これらの用語を理解する必要があります。

SMF サービス

SMF フレームワークでの基本的な管理単位は「サービスインスタンス」です。それぞれの SMF サービスでは、構成された複数のバージョンを保持することができます。さらに、1 つの Solaris システム上で同じバージョンの複数のインスタンスが動作できます。「インスタンス」とは、サービスの特定の構成のことです。Web サーバーはサービスです。ポート 80 で待機するよう構成された Web サーバーデーモンはインスタンスです。Web サーバーサービスの各インスタンスには、異なる構成要件を設定できます。サービスにはシステム全体の構成要件が設定されています。

が、各インスタンスでは必要に応じて特定の要件を無効にできます。1つのサービスの複数のインスタンスは、サービスオブジェクトの子オブジェクトとして管理されます。

サービスは、`in.dhcpd` や `nfsd` などの標準の長年続いているシステムサービスだけでなく、Oracle ソフトウェアなどの ISV アプリケーションを含むさまざまなシステムエンティティーも意味します。また、次のような今まであまり使われなかったエンティティーを組み込むこともできます。

- 物理ネットワークデバイス
- IP アドレス構成
- カーネル構成情報
- マルチユーザー実行レベルなど、システムの `init` 状態に対応するマイルストーン

一般に、サービスはアプリケーションやほかのサービス(ローカルやリモート)に機能リストを提供するエンティティーです。サービスは、暗黙のうちに宣言されたローカルサービスのリストに依存しています。

「マイルストーン」とは、特殊なタイプのサービスです。マイルストーンサービスは、システムの高レベルの属性を表します。たとえば、実行レベル `S`、`2`、および `3` を構成するサービスはマイルストーンサービスによってそれぞれ表されます。

サービス識別子

各サービスインスタンスの名前は、障害管理リソース識別子 (FMRI) によって付けられます。FMRI には、サービス名とインスタンス名が含まれます。たとえば、`rlogin` サービスの FMRI は `svc:/network/login:rlogin` となり、ここでの `network/login` はサービスを、`rlogin` はサービスインスタンスをそれぞれ示します。

次の FMRI の形式はどれも同じです。

- `svc://localhost/system/system-log:default`
- `svc:/system/system-log:default`
- `system/system-log:default`

また、SMF コマンドの中には、`svc:/system/system-log` という FMRI 形式を使用できるものや、あいまいさがまったくない場合に、使用するインスタンスを推測するものもあります。適切な FMRI 形式を判断する方法については、[`svcadm\(1M\)`](#) や [`svcs\(1\)`](#) などの SMF コマンドのマニュアルページを参照してください。

通常、サービス名には一般的な機能カテゴリが含まれます。カテゴリには、次のものがあります。

- `application`
- `デバイス`
- `milestone`

- network
- platform
- site
- system

また、従来の `init.d` スクリプトは、`svc` ではなく `lrc` で始まる FMRI (たとえば、`lrc:/etc/rcS_d/S35cacheos_sh`) で表現されます。従来のサービスは、SMF を使用して監視できますが、管理することはできません。

SMF を使用してシステムを初めてブートしたときに、`/etc/inetd.conf` に示されたサービスが自動的に SMF サービスに変換されます。これらのサービスの FMRI は多少異なります。変換された `inetd` サービスの構文は次のとおりです。

```
network/<service-name>/<protocol>
```

また、RPC プロトコルを使用するサービスの構文は次のとおりです。

```
network/rpc-<service-name>/rpc_<protocol>
```

ここで、`<service-name>` は `/etc/inetd.conf` に定義されている名前であり、`<protocol>` はそのサービスに使用されるプロトコルです。たとえば、`rpc.cmsd` サービスの FMRI は `network/rpc-100068_2-5/rpc_udp` となります。

サービスの状態

`svcs` コマンドは、サービスインスタンスの状態、開始時刻、および FMRI を表示します。各サービスの状態は次のいずれかになります。

- `degraded` – サービスインスタンスは有効ですが、限られた能力で実行されています。
- `disabled` – サービスインスタンスは無効で、実行されていません。
- `legacy_run` – 従来のサービスは SMF によって管理されませんが、監視することはできます。この状態は従来のサービスでのみ使用されます。
- `maintenance` – サービスインスタンスに、管理者が解決しなければならないエラーが発生しました。
- `offline` – サービスインスタンスは有効ですが、サービスが実行されていないか、利用できる状態にありません。
- `online` – サービスインスタンスは有効で、正常に起動されました。
- `uninitialized` – この状態は、すべてのサービスの構成が読み込まれる前の初期状態です。

SMF 目録

SMF の「目録」とは、サービスまたはサービスインスタンスに関連付けられた完全なプロパティースセットを含む XML ファイルのことです。これらのファイルは `/var/svc/manifest` に格納されます。目録は、サービスプロパティの変更には使用しないでください。サービス構成リポジトリは、信頼できる構成情報ソースです。情報を目録からリポジトリに取り込むには、`svccfg import` を実行するか、システムのブート時に、サービスが情報をインポートできるようにする必要があります。

SMF 目録の詳しい内容については、[service_bundle\(4\)](#) のマニュアルページを参照してください。サービスのプロパティを変更する必要がある場合には、[svccfg\(1M\)](#) または [inetadm\(1m\)](#) のマニュアルページを参照してください。

SMF プロファイル

SMF の「プロファイル」とは、サービスインスタンスの一覧とそれぞれを有効にするかどうかを示す XML ファイルのことです。この Solaris リリースで配布されるプロファイルには、次のようなものがあります。

- `/var/svc/profile/generic_open.xml` – このプロファイルは、以前の Solaris リリースではデフォルトで起動されていた標準のサービスを有効化します。
- `/var/svc/profile/generic_limited_net.xml` – このプロファイルは、以前の Solaris リリースではデフォルトで起動されていた標準のインターネットサービスの多くを無効化します。ネットワーク接続を提供するために `network/ssh` サービスは有効化されます。
- `/var/svc/profile/ns_*.xml` – これらのプロファイルは、システムで実行されるよう設定されているネームサービスに関連するサービスを有効化します。
- `/var/svc/profile/platform_*.xml` – これらのプロファイルは、特定のハードウェアプラットフォームに関連するサービスを有効化します。

Solaris 10 リリースを新規にインストールした場合やそのリリースにアップグレードした場合は、そのあとの最初のブート時に、いくつかの Solaris プロファイルが自動的に適用されます。具体的には、`/var/svc/profile/generic.xml` プロファイルが適用されます。通常、このファイルには、`generic_open.xml` または `generic_limited_net.xml` へのシンボリックリンクが設定されています。また、最初のブート時に `site.xml` というプロファイルが `/var/svc/profile` に存在している場合や、次のブートまでに追加された場合は、このプロファイルの内容が適用されます。管理者は `site.xml` プロファイルを使用して、有効にするサービスの初期セットをカスタマイズすることができます。

プロファイルの使用方法については、[389 ページの「SMF プロファイルを適用する方法」](#)を参照してください。

サービス構成リポジトリ

「サービス構成リポジトリ」には、永続的な構成情報と SMF 実行時サービスデータが格納されます。リポジトリは、ローカルメモリーとローカルファイルの間で割り当てられます。SMF は、最終的にサービスデータをネットワークディレクトリサービスで表現できるように設計されています。ネットワークディレクトリサービスはまだ利用できません。サービス構成リポジトリ内のデータを使用すると、多数の Solaris インスタンス間で構成情報の共有や管理の簡素化ができます。サービス構成リポジトリは、SMF インタフェースを使ってのみ操作または照会できます。リポジトリの操作やアクセスの方法については、[svccfg\(1M\)](#) および [svcprop\(1\)](#) のマニュアルページを参照してください。サービス構成リポジトリデーモンについては、[svc.configd\(1M\)](#) のマニュアルページを参照してください。サービス構成ライブラリについては、[libscf\(3LIB\)](#) のマニュアルページを参照してください。

SMF リポジトリのバックアップ

SMF では、次に示すリポジトリのバックアップを自動的行います。

- ブートバックアップは、システムを起動するたびに、リポジトリに対する最初の変更が行われる直前に行われます。
- `manifest_import` バックアップは、`svc:/system/manifest-import:default` によって新しい目録がインポートされたりアップグレードスクリプトが実行されたりした場合、このサービスの完了後に行われます。

タイプごとに 4 つのバックアップがシステムによって管理されます。必要に応じて、もっとも古いバックアップから削除されます。バックアップは `/etc/svc/repository -type-YYYYMMDD_HHMMSSWS` という名前で格納されます。ここでの `YYYYMMDD` (年、月、日) と `HHMMSS` (時、分、秒) は、バックアップが行われた日時です。時間は 24 時間形式で表されます。

エラーが発生した場合は、これらのバックアップからリポジトリを復元できます。そのためには、`/lib/svc/bin/restore_repository` コマンドを使用します。詳細は、[400 ページの「破壊されたりリポジトリを修復する方法」](#)を参照してください。

SMF スナップショット

サービス構成リポジトリ内のデータには、編集可能な構成情報のほかに「スナップショット」もあります。各サービスインスタンスに関するデータは、スナップショットに格納されます。標準のスナップショットは、次のとおりです。

- `initial` – 目録の最初のインポート時に取られる
- `running` – サービスメソッドが実行されるときに使用される
- `start` – 最後に起動が成功したときに取られる

SMF サービスは常に `running` スナップショットを使って実行します。このスナップショットが存在しない場合は、自動的に作成されます。

`svcadm refresh` コマンド (ときどきこのあとに `svcadm restart` コマンドが実行される) によってスナップショットがアクティブになります。以前のスナップショットに含まれるインスタンス構成を表示したり、そこに戻ったりするには、`svccfg` コマンドを使用します。詳細については、[387 ページの「別の SMF スナップショットに戻す方法」](#)を参照してください。

SMF 管理およびプログラミングインタフェース

この節では、SMF の使用時に利用できるインタフェースについて簡単に説明します。

SMF コマンド行管理ユーティリティー

SMF には、SMF とやりとりしたり、標準の管理タスクを実行したりする 1 連のコマンド行ユーティリティーが用意されています。SMF の管理には、次のユーティリティーを使用できます。

表 17-1 サービス管理機能のユーティリティー

コマンド名	機能
<code>inetadm</code>	<code>inetd</code> によって制御されるサービスを監視または構成できます
<code>svcadm</code>	サービスインスタンスの有効化、無効化、再起動などの一般的なサービス管理タスクを実行できます
<code>svccfg</code>	サービス構成リポジトリの内容を表示および操作できます
<code>svccprop</code>	シェルスクリプトでの使用に適した出力形式で、サービス構成リポジトリからプロパティ値を取り出します
<code>svcs</code>	サービス構成リポジトリ内のすべてのサービスインスタンスのサービス状態を詳細に表示できます

サービス管理構成ライブラリインタフェース

SMF には、`svc.configd` デーモンを介してサービス構成リポジトリとのやりとりを行うための一連のプログラミングインタフェースが用意されています。このデーモンは、ローカルのリポジトリデータストアに対するすべての要求を判定します。サービス構成リポジトリ内のサービスとの最低レベルのやりとりの手段として、1 連の基本的なインタフェースが定義されています。これらのインタフェースを使うと、トランザクションやスナップショットなどのサービス構成リポジトリのすべての機能にアクセスできます。

多くの開発者は、SMF とやりとりするための一般的なタスクのみを必要としています。これらのタスクは、便利な機能として基本サービスの上位で実装され、実装にかかる負荷を軽減しています。

SMF コンポーネント

SMF には、マスターリスタータデーモンと委任リスタータがあります。

SMF マスターリスタータデーモン

`svc.startd` デーモンは、Solaris OS のマスタープロセススタータおよびリスタータです。このデーモンは、システム全体のサービス依存性を管理する役割を担っています。適切な実行レベルで適切な `/etc/rc*.d` スクリプトを起動することは、以前は `init` の役割でしたが、現在はこのデーモンの役割です。まず、`svc.startd` はサービス構成リポジトリに格納されている情報を取り出します。次に、サービス依存性が満たされたときにそのサービスを起動します。また、失敗したサービスの再起動や、依存性が満たされなくなったサービスの停止も行います。このデーモンは、プロセスの消滅などのイベントを通してオペレーティングシステムの可用性の点からサービス状態を追跡します。

SMF 委任リスタータ

一部のサービスは、起動時に共通の動きが見られます。これらのサービス間に共通性を持たせるために、委任リスタータがこれらのサービスに対する責任を負うことがあります。また、より複雑な再起動やアプリケーション固有の再起動を行えるようにする場合にも委任リスタータを使用できます。委任リスタータは、別のメソッド群をサポートできますが、マスターリスタータと同じサービス状態をエクスポートします。リスタータの名前は、サービスとともに格納されます。委任リスタータの例には、インターネットサービスを常に実行しておくのではなく、要求に応じて起動できる `inetd` があります。

SMF とブート

SMF には、システムをブートするための新しい方法が用意されています。次に例を示します。

- all マイルストーンに関連付けられるシステム状態が追加されています。all マイルストーンでは、multi-user-server マイルストーン上で定義済みの依存関係を持つすべてのサービスと、定義済みの依存関係を持たないすべてのサービスが起動されます。Sun 以外の製品など、サービスを追加しても、次のコマンドを実行するまでそれらのサービスが自動的に起動されないことがあります。

```
ok boot -m milestone=all
```

- システムをブートするときに、冗長オプションを使用して詳細なメッセージを表示することができます。デフォルトでは、これらのメッセージは表示されません。冗長モードでブートするには、次のコマンドを使用します。

```
ok boot -mverbose
```

- none マイルストーンに関連付けられた新しいシステム状態が存在します。このマイルストーンを使ってシステムをブートすると、init、svc.startd、および svc.configd だけが起動されます。この状態は、ブートに関する問題のデバッグを行う際、非常に役立つ場合があります。特に、サービスが一切起動されないため、SMF サービスの設定に関する問題のデバッグがより単純になります。none マイルストーンの使用手順については、[402 ページの「どのサービスも起動しないでブートする方法」](#)を参照してください。

SMF 互換性

標準の Solaris サービスの多くは SMF によって管理されていますが、実行レベルの移行に対しては /etc/rc*.d 内にあるスクリプトが引き続き実行されます。以前の Solaris リリースに含まれていた /etc/rc*.d スクリプトの大半は、SMF の一環として削除されました。残りのスクリプトを引き続き実行できることにより、SMF を使用するようにサービスを変換しなくても Sun 以外のアプリケーションを追加できます。

また、/etc/inittab と /etc/inetd.conf は、インストール後のスクリプトで修正するパッケージで利用可能でした。これらは、従来の実行サービスと呼ばれます。inetconv コマンドは、これらの従来の実行サービスをサービス構成リポジトリに追加する場合に実行されます。従来の実行サービスの状態は表示できますが、ほかの変更は一切 SMF でサポートされていません。この機能を使用するアプリケーションは、SMF が提供する高精度の障害の封じ込めによるメリットを受けられません。

SMF を利用するように変換されたアプリケーションは、/etc/inittab と /etc/inetd.conf の各ファイルに対して変更を行えなくなります。変換されたアプリ

ケーションは、`/etc/rc*.d` スクリプトを使用しません。また、新しいバージョンの `inetd` は `/etc/inetd.conf` のエントリを検索しません。

実行レベル

システムの「実行レベル」(「init 状態」とも呼ばれる)は、ユーザーが使用できるサービスとリソースを定義します。システムが一度に持つことのできる実行レベルは1つだけです。

Solaris OS には8つの実行レベルがあります(次の表を参照)。デフォルトの実行レベル3は、`/etc/inittab` ファイルに指定されています。

表 17-2 Solaris 実行レベル

実行レベル	init 状態	種類	目的
0	電源切断状態	電源切断	オペレーティングシステムをシャットダウンして、システムの電源を安全に落とせるようにします。
s または S	シングルユーザー状態	シングルユーザー	一部のファイルシステムがマウントされ使用可能な状態で、シングルユーザーとして動作します。
1	システム管理状態	シングルユーザー	すべての使用可能なファイルシステムにアクセスします。ユーザーログインは使用できません。
2	マルチユーザー状態	マルチユーザー	通常の運用に使用します。複数のユーザーがシステムとすべてのファイルシステムにアクセスできます。NFS サーバーデーモンを除く、すべてのデーモンが動作します。
3	NFS リソースを共有したマルチユーザーレベル	マルチユーザー	NFS リソースを共有する通常の運用に使用します。Solaris OS におけるデフォルトの実行レベルです。
4	マルチユーザー状態(予備)		デフォルトでは設定されていませんが、顧客用に使用できます。
5	電源切断状態	電源切断	オペレーティングシステムをシャットダウンしてシステムの電源を安全に落とせるようにします。可能であれば、この機能をサポートしているシステムでは電源を自動的に切断します。

表 17-2 Solaris 実行レベル (続き)

実行レベル	init 状態	種類	目的
6	リブート状態	リブート	システムをシャットダウンして実行レベル 0 にしたあと、NFS リソースを共有するマルチユーザーレベル(または <code>inittab</code> ファイルに指定されたデフォルトのレベル)でリブートします。

また、`svcadm` コマンドを使用してシステムの実行レベルを変更することもできます。その場合は、実行するときのマイルストーンを選択してください。次の表に、各マイルストーンに対応する実行レベルを示します。

表 17-3 Solaris 実行レベルと SMF マイルストーン

実行レベル	SMF マイルストンの FMRI
S	milestone/single-user:default
2	milestone/multi-user:default
3	milestone/multi-user-server:default

実行レベルとマイルストンの使い分け

ほとんどの状況下では、`init` コマンドといずれかの実行レベルを使用してシステムの状態を変更するだけで十分です。マイルストーンを使用したシステム状態の変更は複雑であり、予期しない動作につながる可能性があります。さらに、`init` コマンドではシステムのシャットダウンも行えます。したがって、`init` が、システムの状態を変更するための最適なコマンドであると言えます。

ただし、`none` マイルストーンによるシステムのブートは、起動時の問題のデバッグを行う際に非常に役立つ可能性があります。`none` マイルストーンと同等の実行レベルはありません。具体的な手順については、[402 ページの「どのサービスも起動しないでブートする方法」](#)を参照してください。

システムの実行レベルを確認する

`who -r` コマンドを使用すると、実行レベルに関する情報が表示されます。

```
$ who -r
```

システムの現在の実行レベルを調べるには、`who -r` コマンドを使用します。

例 17-1 システムの実行レベルを確認する

次の例では、システムの現在の実行レベルと以前の実行レベルに関する情報を表示します。

```
$ who -r
.      run-level 3   Dec 13 10:10   3   0 S
$
```

who -r コマンドの出力	説明
run-level 3	現在の実行レベル
Dec 13 10:10	実行レベルが最後に変更された日時
3	現在の実行レベル
0	最後にリブートしてからシステムがこの実行レベルになった回数
S	以前の実行レベル

/etc/inittab ファイル

init または shutdown コマンドを使用してシステムをブートしたり、実行レベルを変更したりすると、init デーモンは /etc/inittab ファイルから情報を読み取ってプロセスを起動します。/etc/inittab ファイルには、init プロセスにとって重要な次の情報が定義されています。

- init プロセスが再起動すること
- 起動、監視するプロセス、および停止時に再起動するプロセス
- システムが新しい実行レベルに移行したとき行う処理

/etc/inittab ファイル内の各エントリは、次のフィールドからなります。

id:*rstate* :*action*:*process*

次の表に、inittab エントリの各フィールドを要約します。

表 17-4 inittab ファイルのフィールドの説明

フィールド	説明
<i>id</i>	エントリに固有の (一意の) 識別子。
<i>rstate</i>	このエントリが適用される実行レベルのリスト。

表 17-4 inittab ファイルのフィールドの説明 (続き)

フィールド	説明
アクション	プロセスフィールドに指定されたプロセスの実行方法。指定できる値は、sysinit、boot、bootwait、wait、および respawn です。 ほかの action キーワードについては、 inittab(4) のマニュアルページを参照してください。
<i>process</i>	実行するコマンドまたはスクリプトを定義します。

例 17-2 デフォルトの inittab ファイル

次の例では、Solaris リリースでインストールされるデフォルトの inittab ファイルを示します。そのあとに、この例の出力の各行についての説明も示します。

```
ap::sysinit:/sbin/autopush -f /etc/iu.ap (1)
sp::sysinit:/sbin/soconfig -f /etc/sock2path (2)
smf::sysinit:/lib/svc/bin/svc.startd >/dev/msglog 2<>/dev/msglog (3)
p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog 2<>/dev/... (4)
```

- 1. STREAMS モジュールを初期化します
- 2. ソケット転送プロバイダを構成します
- 3. SMF 用のマスターリスタータを初期化します
- 4. 電源障害の場合のシャットダウンを指定します

システムが実行レベル 3 になると実行される処理

- 1. init プロセスが起動されます。init プロセスは、/etc/default/init ファイルを読み取って環境変数を設定します。デフォルトでは、TIMEZONE 変数だけが設定されます。
- 2. init は inittab ファイルを読み取り、次の処理を行います。
 - a. action フィールドが sysinit になっているすべてのプロセスエントリを実行して、ユーザーがログインする前に特別な初期設定処理がすべて行われるようにします。
 - b. 起動アクティビティを svc.startd に渡します。

init プロセスが inittab ファイルを使用する方法については、[init\(1M\)](#) のマニュアルページを参照してください。

サービスの管理 (手順)

この章では、サービス管理機能 (Service Management Facility: SMF) の管理と監視に必要なとなる作業について説明します。さらに、実行レベルスクリプトの管理方法に関する情報も提供します。次の項目について説明します。

- [379 ページの「サービスの管理 \(作業マップ\)」](#)
- [380 ページの「SMF サービスの監視」](#)
- [384 ページの「SMF サービスの管理」](#)
- [391 ページの「SMF サービスの設定」](#)
- [396 ページの「実行制御スクリプトの使用」](#)
- [399 ページの「サービス管理機能のトラブルシューティング」](#)

サービスの管理 (作業マップ)

次の作業マップに、SMF を使用するために必要な手順を示します。

作業	説明	参照先
サービスインスタンスの状態を表示します。	実行中のすべてのサービスインスタンスの状態を表示します。	380 ページの「サービスの状態を一覧表示する方法」
特定のサービスに依存するサービスを表示します。	指定されたサービスに依存するサービスを表示します。	382 ページの「特定のサービスに依存するサービスを表示する方法」
特定のサービスが依存するサービスを表示します。	指定されたサービスが依存するサービスを表示します。この情報があれば、特定のサービスが起動できない原因を特定しやすくなります。	382 ページの「特定のサービスが依存するサービスを表示する方法」

作業	説明	参照先
サービスインスタンスを無効にします。	正常に動作していないサービスや、セキュリティを高めるためにオフにする必要があるサービスを無効にします。	384 ページの「サービスインスタンスを無効にする方法」
サービスインスタンスを有効にします。	特定のサービスを起動します。	385 ページの「サービスインスタンスを有効にする方法」
サービスインスタンスを再起動します。	無効化と有効化に各コマンドを使用することなしにサービスを再起動します。	386 ページの「サービスを再起動する方法」
サービスインスタンスを変更します。	指定されたサービスインスタンスの設定パラメータを変更します。 <code>inetd</code> によって制御されるサービスの設定プロパティを変更します。 <code>inetd</code> によって制御されるサービスの起動オプションを変更します。	391 ページの「サービスを変更する方法」 392 ページの「<code>inetd</code> 制御サービスのプロパティを変更する方法」 394 ページの「<code>inetd</code> 制御サービスのコマンド行引数を変更する方法」
<code>inetd.conf</code> エントリを変換します。	<code>inetd</code> サービスを、SMF で監視可能な従来の実行サービスに変換します。	395 ページの「<code>inetd.conf</code> エントリを変換する方法」
破壊されたサービス設定リポジトリを修復します。	破壊されたりリポジトリをデフォルトのバージョンと交換します。	400 ページの「破壊されたりリポジトリを修復する方法」
どのサービスも起動しないでシステムをブートします。	ブートを妨げている設定の問題を解決できるように、どのサービスも起動せずにシステムをブートします。	402 ページの「どのサービスも起動しないでブートする方法」

SMF サービスの監視

次の各作業は、SMF サービスの監視方法を示したものです。

▼ サービスの状態を一覧表示する方法

この手順は、実行中のサービスを表示する場合に使用できます。

- `svcs` コマンドを実行します。

オプションを指定しないでこのコマンドを実行した場合、FMRI で指定されたサービスの状態レポートが表示されます。

```
% svcs -l FMRI
```

例 18-1 rlogin サービスの状態を表示する

この例では、多くの契約を含むサービスの状態を表示しています。

```
% svcs -l network/login:rlogin
fmri          svc:/network/login:rlogin
enabled       true
state         online
next_state    none
restarter     svc:/network/inetd:/default
contract_id   42325 41441 40776 40348 40282 40197 39025 38381 38053\
              33697 28625 24652 23689 15352 9889 7194 6576 6360 5387 1475 3015\
              6545 6612 9302 9662 10484 16254 19850 22512 23394 25876 26113 27326\
              34284 37939 38405 38972 39200 40503 40579 41129 41194
```

例 18-2 sendmail サービスの状態を表示する

この例では、依存するサービスを持つサービスの状態を表示しています。

```
% svcs -l network/smtp:sendmail
fmri          svc:/network/smtp:sendmail
enabled       true
state         online
next_state    none
restarter     svc:/system/svc/restarter:default
contract_id   29462
dependency    require_all/refresh file://localhost/etc/nsswitch.conf (-)
dependency    require_all/refresh file://localhost/etc/mail/sendmail.cf (-)
dependency    optional_all/none svc:/system/system-log (online)
dependency    require_all/refresh svc:/system/identity:domain (online)
dependency    require_all/refresh svc:/milestone/name-services (online)
dependency    require_all/none svc:/network/service (online)
dependency    require_all/none svc:/system/filesystem/local (online)
```

例 18-3 すべてのサービスの状態を表示する

次のコマンドを実行すると、システムにインストールされているすべてのサービスが一覧表示されます。その際、各サービスの状態も表示されます。このコマンドでは、有効になっているサービスだけでなく、無効になっているサービスも表示されます。

```
% svcs -a
```

例 18-4 inetd によって制御されるサービスの状態を表示する

次のコマンドを実行すると、inetd によって制御されるサービスが一覧表示されます。サービスごとに、FMRI のほかに、実行状態と有効になっているかどうかが表示されます。

```
% inetadm
```

▼ 特定のサービスに依存するサービスを表示する方法

この手順は、指定されたサービスに依存するサービスインスタンスを確認する方法を示しています。

- 特定のサービスに依存するサービスを表示します。

```
% svcs -D FMRI
```

例 18-5 マルチユーザーマイルストーンに依存するサービスインスタンスを表示する

次の例は、マルチユーザーマイルストーンに依存するサービスインスタンスを確認する方法を示しています。

```
% svcs -D milestone/multi-user
STATE          STIME    FMRI
online         Apr_08   svc:/milestone/multi-user-server:default
```

▼ 特定のサービスが依存するサービスを表示する方法

この手順は、指定されたサービスインスタンスが依存するサービスを確認する方法を示しています。

- 特定のサービスが依存するサービスを表示します。

```
% svcs -d FMRI
```

例 18-6 マルチユーザーマイルストーンが依存するサービスインスタンスを表示する

次の例では、マルチユーザーマイルストーンが依存するサービスインスタンスを表示しています。

```
% svcs -d milestone/multi-user:default
```

```
STATE      STIME      FMRI
disabled   Aug_24     svc:/platform/sun4u/sf880drd:default
online     Aug_24     svc:/milestone/single-user:default
online     Aug_24     svc:/system/utmp:default
online     Aug_24     svc:/system/system-log:default
online     Aug_24     svc:/system/system-log:default
online     Aug_24     svc:/system/rmtmpfiles:default
online     Aug_24     svc:/network/rpc/bind:default
online     Aug_24     svc:/milestone/name-services:default
online     Aug_24     svc:/system/filesystem/local:default
online     Aug_24     svc:/system/mdmonitor:default
```

SMF サービス (作業マップ)

作業	説明	参照先
サービスインスタンスを無効にします。	実行中のサービスを停止し、サービスが再び起動しないようにします。	384 ページの「サービスインスタンスを無効にする方法」
サービスインスタンスを有効にします。	特定のサービスを起動します。また、以降のリポート時にサービスが再起動されます。	385 ページの「サービスインスタンスを有効にする方法」
特定のサービスを再起動します。	サービスの停止と起動を 1 つのコマンドで行います。	386 ページの「サービスを再起動する方法」
保守状態のサービスを復元します。	保守状態のサービスをクリーンアップして再起動する方法を示します。	386 ページの「保守状態のサービスを復元する方法」
スナップショットに戻します。	以前のスナップショットを使用して、サービスの問題を修正します。	387 ページの「別の SMF スナップショットに戻す方法」
プロファイルを作成します。	必要に応じてサービスを無効または有効にするために、プロファイルを作成します。	388 ページの「SMF プロファイルを作成する方法」
プロファイルを適用します。	プロファイルの情報を使用して、必要に応じてサービスを無効または有効にします。	389 ページの「SMF プロファイルを適用する方法」
netserives コマンドを使用して、サービスとその設定を変更します。	generic_limited.xml プロファイルまたは generic_open.xml プロファイルの情報を使用して、サービスを無効または有効にし、サービスの設定の変更も行います。	390 ページの「ネットワークに提供するサービスを generic*.xml で変更する」

SMF サービスの管理

この節では、SMF サービスの管理について説明します。

RBAC 権利プロファイルを使って SMF を管理する

RBAC 権利プロファイルを使用して root アクセス権を付与していないユーザーに、一部の SMF サービスの管理を許可することができます。権利プロファイルには、そのユーザーが実行できるコマンドを定義します。SMF のために次のプロファイルが作成されています。

- **Service Management:** サービスを追加、削除、または変更できます。
- **Service Operator:** 再起動や再表示など、サービスインスタンスの状態の変更を要求できます。

承認については、[smf_security\(5\)](#) のマニュアルページを参照してください。権利プロファイルを割り当てる手順については、『[Solaris のシステム管理\(セキュリティサービス\)](#)』の「[ユーザーの RBAC プロパティーを変更する方法](#)」を参照してください。

▼ サービスインスタンスを無効にする方法

サービスを無効にするには、次の手順に従います。サービスの状態変化はサービス設定リポジトリ内に記録されます。サービスがいったん無効になると、その無効な状態はリブート後も維持されます。サービスを実行中の状態に戻す唯一の方法は、そのサービスを有効にすることです。

- 1 スーパーユーザーになるか、Service Management 権利プロファイルを含む役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理\(セキュリティサービス\)](#)』の「[RBAC の構成](#)」を参照してください。

- 2 無効にするサービスに依存するサービスの有無を確認します。

必要とするサービスがこのサービスに依存していた場合、このサービスを無効にできません。

```
# svcs -D FMRI
```

- 3 サービスを無効にします。

```
# svcadm disable FMRI
```


例 18-7 rlogin サービスを無効にする

最初のコマンドの出力結果から、rlogin サービスに依存するサービスは存在しないことがわかります。この例の 2 番目のコマンドは、rlogin サービスを無効にしています。3 番目のコマンドの結果から、rlogin サービスインスタンスの状態が無効になっていることがわかります。

```
# svcs -D network/login:rlogin
# svcadm disable network/login:rlogin
STATE          STIME      FMRI
# svcs network/login:rlogin
STATE          STIME      FMRI
disabled       11:17:24  svc:/network/login:rlogin
```

▼ サービスインスタンスを有効にする方法

サービスを有効にするには、次の手順に従います。サービスの状態変化はサービス設定リポジトリ内に記録されます。サービスがいったん有効になると、そのサービスの依存関係に問題がない限り、その有効な状態はシステムのリブート後も維持されます。

- 1 スーパーユーザーになるか、Service Management 権利プロファイルを含む役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「RBAC の構成」を参照してください。

- 2 サービスの依存関係に問題がないか確認します。

依存するサービスが有効になっている場合、サービスの依存関係に問題はありません。そうでない場合は、`svcadm enable -r FMRI` を使用して、すべての依存関係を再帰的に有効にします。

```
# svcs -l FMRI|grep enabled
```

- 3 サービスを有効にします。

```
# svcadm enable FMRI
```

例 18-8 rlogin サービスを有効にする

この例の 2 番目のコマンドは、rlogin サービスを有効にしています。3 番目のコマンドの結果から、rlogin サービスインスタンスの状態がオンラインになっていることがわかります。

```
# svcs -l network/login:rlogin|grep enabled
enabled      false
# svcadm enable network/login:rlogin
```

```
# svcs network/login:login
STATE          STIME    FMRI
online         12:09:16 svc:/network/login:login
```

例 18-9 シングルユーザーモードでサービスを有効にする

次のコマンドは `rpcbind` を有効にします。 `-t` オプションは、サービスリポジトリを変更しないモードでサービスを一時的に起動します。このリポジトリにシングルユーザーモードで書き込むことはできません。 `-r` オプションは、指定されたサービスのすべての依存関係を再帰的に有効にします。

```
# svcadm enable -rt rpc/bind
```

▼ サービスを再起動する方法

設定が変更されたなどの理由により現在実行中のサービスを再起動する必要がある場合、サービスの停止時と起動時に個々のコマンドを入力することなく、サービスの再起動を行えます。サービスの無効化と有効化を別々に行う必要があるのは、サービスの無効化後かつ有効化前に変更を反映する必要がある場合だけです。

- 1 スーパーユーザーになるか、Service Management 権利プロファイルを含む役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成](#)」を参照してください。

- 2 特定のサービスを再起動します。

```
# svcadm restart FMRI
```

▼ 保守状態のサービスを復元する方法

- 1 スーパーユーザーになるか、Service Management 権利プロファイルを含む役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成](#)」を参照してください。

- 2 そのサービスに依存するプロセスのうち、停止していないものが存在しないかどうかを確認します。

通常、サービスインスタンスが保守状態になっている場合、そのインスタンスに関連付けられたすべてのプロセスが停止しています。しかしながら、処理を進める前

にそのことを確認する必要があります。次のコマンドを実行すると、特定のサービスインスタンスに関連付けられたすべてのプロセスが一覧表示されます。その際、各プロセスの PID も表示されます。

```
# svcs -p FMRI
```

- 3 (省略可能)残っているプロセスをすべて強制終了します。

svcs コマンドで表示されたすべてのプロセスに対して、この手順を繰り返します。

```
# pkill -9 PID
```

- 4 必要であれば、サービスの設定を修復します。

/var/svc/log 内の対応するサービスログファイルを開き、エラーが発生していないか確認します。

- 5 サービスを復元します。

```
# svcadm clear FMRI
```

▼ 別の SMF スナップショットに戻す方法

サービスの設定が間違っている場合、正常に起動した最後のスナップショットに戻すことで問題を解決できます。この手順では、console-login サービスの以前のスナップショットを使用します。

- 1 スーパーユーザーになるか、Service Management 権利プロファイルを含む役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成](#)」を参照してください。

- 2 svccfg コマンドを実行します。

```
# svccfg
svc:>
```

- a. 修正するサービスインスタンスを選択します。

注-インスタンスを完全に定義した FMRI を使用する必要があります。短縮形は使えません。

```
svc:> select system/console-login:default
svc:/system/console-login:default>
```

- b. 利用可能なスナップショットの一覧を生成します。

```
svc:/system/console-login:default> listsnap
initial
running
start
svc:/system/console-login:default>
```

- c. 戻し先として start スナップショットを選択します。

start スナップショットとは、サービスが正常に起動した最後のスナップショットのことです。

```
svc:/system/console-login:default> revert start
svc:/system/console-login:default>
```

- d. svccfg を終了します。

```
svc:/system/console-login:default> quit
#
```

- 3 サービス設定リポジトリ内の情報を更新します。

この手順では、start スナップショットの設定情報に基づいてリポジトリを更新します。

```
# svcadm refresh system/console-login
```

- 4 サービスインスタンスを再起動します。

```
# svcadm restart system/console-login
```

▼ SMF プロファイルを作成する方法

プロファイルとは、SMF サービスの一覧とそれぞれを有効にするかどうかを示す XML ファイルのことです。プロファイルは、多くのサービスを一度に有効または無効にするために使用されます。すべてのサービスをプロファイルに含める必要はありません。効果的なプロファイルを作成するには、各プロファイルには有効または無効にする必要のあるサービスだけを含めるようにします。

- 1 プロファイルを作成します。

この例では、svccfg コマンドを使用して、現在のシステムで有効になっているサービスと無効になっているサービスを反映するプロファイルを作成します。代わりに、既存のプロファイルのコピーを作成し、それを編集することもできます。

```
# svccfg extract> profile.xml
```

JumpStart を使用する場合、同一のシステムが多数ある場合、またはあとで復元できるようにシステムの設定のアーカイブを作成する場合は、この手順を使用して単一の SMF プロファイルを作成することができます。

2 profile.xml ファイルを編集して、必要な変更を加えます。

a. service_bundle 宣言内のプロファイルの名前を変更します。

この例では、profile という名前に変更しています。

```
# cat profile.xml
...
<service_bundle type='profile' name='profile'
  xmlns:xi='http://www.w3.org/2003/XInclude'
  ...
```

b. このプロファイルで管理すべきでないサービスをすべて削除します。

各サービスについて、サービスを説明する 3 行を削除します。各サービスの説明は、<service で始まり、</service で終わります。この例は、LDAP クライアントサービスの行を示しています。

```
# cat profile.xml
...
<service name='network/ldap/client' version='1' type='service'>
  <instance name='default' enabled='true' />
</service>
```

c. このプロファイルで管理すべきサービスをすべて追加します。

上記の 3 行の構文を使用して各サービスを定義する必要があります。

d. 必要に応じて、選択したサービスの enabled フラグを変更します。

この例では、sendmail サービスを無効にしています。

```
# cat profile.xml
...
<service name='network/smtp' version='1' type='service'>
  <instance name='sendmail' enabled='false' />
</service>
...
```

3 必要に応じて、この新しいプロファイルを適用します。

手順については、[389 ページの「SMF プロファイルを適用する方法」](#)を参照してください。

▼ SMF プロファイルを適用する方法

1 スーパーユーザーになるか、Service Management 権利プロファイルを含む役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「RBAC の構成」を参照してください。

2 プロファイルを適用します。

この例では、`profile.xml` プロファイルを使用しています。

```
# svccfg apply profile.xml
```

注 - `generic_limited_net.xml` と `generic_open.xml` を切り替えるための具体的な手順、およびこの切り替えを行うときに適用する必要のあるプロパティについては、[390 ページの「ネットワークに提供するサービスを `generic\*.xml` で変更する」](#)を参照してください。

▼ ネットワークに提供するサービスを `generic*.xml` で変更する

`netservices` コマンドを使用すると、ネットワークに最小限のシステムサービスを公開するか、従来の (以前の Solaris リリースと同じ) サービスを公開するかを切り替えることができます。この切り替えは、`generic_limited.xml` プロファイルと `generic_open.xml` プロファイルによって行われます。また、このコマンドによって一部のサービスのプロパティが変更され、それによってサービスがローカル専用モードや従来モードに制限されます。

1 スーパーユーザーになるか、Service Management 権利プロファイルを含む役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成](#)」を参照してください。

2 `netservices` コマンドを実行します。

この例では、`open`、つまり従来のネットワークサービスの公開を選択しています。

```
# /usr/sbin/netservices open
```

例 18-10 ネットワークサービスの公開を制限する

このコマンドは、一部のサービスをローカルモードで実行するようにプロパティを変更し、有効にするサービスを `generic_limited_net` プロファイルによって制限します。このコマンドは、`generic_open.xml` プロファイルが適用されている場合のみ使用するようにしてください。

```
# /usr/sbin/netservices limited
```

SMF サービスの設定

▼ サービスを変更する方法

次の手順は、inetd サービスによって管理されていないサービスの設定を変更する方法を示します。

- 1 スーパーユーザーになるか、Service Management 権利プロファイルを含む役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成」を参照してください。
- 2 必要に応じて設定ファイルを変更します。
多くのサービスは、起動方法などの設定情報を定義するための設定ファイルを1つ以上持っています。それらのファイルはサービス実行中でも変更可能です。ファイルの内容がチェックされるのはサービス起動時です。
- 3 サービスを再起動します。

```
# svcadm restart FMRI
```

例 18-11 NFS ファイルシステムを共有する

NFS サービスを使って特定のファイルシステムを共有するには、そのファイルシステムを /etc/dfs/dfstab ファイル内に定義したあと、NFS サービスを再起動する必要があります。この例は、dfstab ファイルの内容とサービスの起動方法を示しています。

```
# cat /etc/dfs/dfstab
.
.
share -F nfs -o rw /export/home
# svcadm restart svc:/network/nfs/server
```

▼ サービスの環境変数を変更する方法

この手順は、デバッグに役立つよう cron 環境変数を変更する方法を示します。

- 1 スーパーユーザーになるか、Service Management 権利プロファイルを含む役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成」を参照してください。

- 2 サービスが実行されていることを確認します。

```
# svcs system/cron
STATE          STIME      FMRI
online         Dec_04    svc:/system/cron:default
```

- 3 環境変数を設定します。

次の例では、環境変数 UMEM_DEBUG および LD_PRELOAD を設定しています。setenv サブコマンドについては、[svccfg\(1M\)](#) のマニュアルページを参照してください。

```
# svccfg -s system/cron:default setenv UMEM_DEBUG default
# svccfg -s system/cron:default setenv LD_PRELOAD libumem.so
```

- 4 サービスの再表示と再起動を行います。

```
# svcadm refresh system/cron
# svcadm restart system/cron
```

- 5 正しく変更されたことを確認します。

```
# pargs -e 'pgrep -f /usr/sbin/cron'
100657: /usr/sbin/cron
envp[0]: LOGNAME=root
envp[1]: LD_PRELOAD=libumem.so
envp[2]: PATH=/usr/sbin:/usr/bin
envp[3]: SMF_FMRI=svc:/system/cron:default
envp[4]: SMF_METHOD=/lib/svc/method/svc-cron
envp[5]: SMF_RESTARTER=svc:/system/svc/restarter:default
envp[6]: TZ=GB
envp[7]: UMEM_DEBUG=default
#
```

▼ inetd 制御サービスのプロパティーを変更する方法

- 1 スーパーユーザーになるか、Service Management 権利プロファイルを含む役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理\(セキュリティサービス\)](#)』の「RBAC の構成」を参照してください。

- 2 特定のサービスのプロパティーを一覧表示します。

このコマンドを実行すると、FMRI で指定されたサービスのすべてのプロパティーが表示されます。

```
# inetadm -l FMRI
```


3 サービスのプロパティを変更します。

inetd 制御サービスの各プロパティは、プロパティ名とその値によって定義されます。値を指定せずにプロパティ名を指定した場合、そのプロパティはデフォルト値にリセットされます。特定のサービスのプロパティの詳細については、そのサービスのマニュアルページを参照してください。

```
# inetadm -m FMRI property-name=value
```

4 プロパティが変更されたことを確認します。

プロパティを再度一覧表示し、適切に変更されていることを確認します。

```
# inetadm -l FMRI
```

5 変更が適用されていることを確認します。

プロパティに対する変更が期待どおりに適用されていることを確認します。

例 18-12 telnet の tcp_trace プロパティを変更する

次の例は、telnet の tcp_trace プロパティを true に設定する方法を示しています。telnet コマンドの実行後の syslog 出力の結果から、変更が適用されていることがわかります。

```
# inetadm -l svc:/network/telnet:default
SCOPE    NAME=VALUE
         name="telnet"
.
.
default  inherit_env=TRUE
default  tcp_trace=FALSE
default  tcp_wrappers=FALSE
# inetadm -m svc:/network/telnet:default tcp_trace=TRUE
# inetadm -l svc:/network/telnet:default
SCOPE    NAME=VALUE
         name="telnet"
.
.
default  inherit_env=TRUE
         tcp_trace=TRUE
default  tcp_wrappers=FALSE
# telnet localhost
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
login: root
Password:
Last login: Mon Jun 21 05:55:45 on console
Sun Microsystems Inc.    SunOS 5.10      s10_57   May 2004
```

```
# ^D
Connection to localhost closed by foreign host.
# tail -1 /var/adm/messages
Jun 21 06:04:57 yellow-19 inetd[100308]: [ID 317013 daemon.notice] telnet[100625]
    from 127.0.0.1 32802
```

▼ inetd 制御サービスのコマンド行引数を変更する方法

- 1 スーパーユーザーになるか、Service Management 権利プロファイルを含む役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成」を参照してください。

- 2 特定のサービスの exec プロパティを表示します。

このコマンドを実行すると、FMRI で指定されたサービスのすべてのプロパティが表示されます。grep コマンドを追加することで、サービスの exec プロパティに出力を制限しています。

```
# inetadm -l FMRI|grep exec
```

- 3 サービスの exec プロパティを変更します。

exec プロパティに設定される *command-syntax* は、サービス起動時に実行されるコマンド文字列を定義します。

```
# inetadm -m FMRI exec="command-syntax"
"
```

- 4 プロパティが変更されたことを確認します。

プロパティを再度一覧表示し、適切に変更されていることを確認します。

```
# inetadm -l FMRI
```

例 18-13 ftp コマンドに接続ロギング (-l) オプションを追加する

この例では、ftp デーモンの起動オプションに -l オプションを追加しています。この変更が適用されたことを確かめるには、ftp ログインセッション完了後に syslog 出力を調べます。

```
# inetadm -l svc:/network/ftp:default | grep exec
    exec="/usr/sbin/in.ftpd -a"
# inetadm -m svc:/network/ftp:default exec="/usr/sbin/in.ftpd -a -l"
# inetadm -l svc:/network/ftp:default
SCOPE      NAME=VALUE
```

```

name="ftp"
endpoint_type="stream"
proto="tcp6"
isrpc=FALSE
wait=FALSE
exec="/usr/sbin/in.ftpd -a -l"
.
.
# ftp localhost
Connected to localhost.
220 yellow-19 FTP server ready.
Name (localhost:root): mylogin
331 Password required for mylogin.
Password:
230 User mylogin logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> quit
221-You have transferred 0 bytes in 0 files.
221-Total traffic for this session was 236 bytes in 0 transfers.
221-Thank you for using the FTP service on yellow-19.
221 Goodbye.
# tail -2 /var/adm/messages
Jun 21 06:54:33 yellow-19 ftpd[100773]: [ID 124999 daemon.info] FTP LOGIN FROM localhost
[127.0.0.1], mylogin
Jun 21 06:54:38 yellow-19 ftpd[100773]: [ID 528697 daemon.info] FTP session closed

```

▼ inetd.conf エントリを変換する方法

次の手順では、inetd.conf エントリを SMF サービス目録に変換します。inetd に依存する Sun 以外のアプリケーションをシステムに追加するたびに、この手順を実行する必要があります。また、/etc/inetd.conf 内のエントリに設定変更を行う必要がある場合にも、この手順を実行してください。

- 1 スーパーユーザーになるか、Service Management 権利プロファイルを含む役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成](#)」を参照してください。

- 2 inetd.conf エントリを変換します。

inetconv コマンドは、選択されたファイル内の各エントリをサービス目録に変換します。

```
# inetconv -i filename
```

例 18-14 /etc/inet/inetd.conf エントリを SMF サービス目録に変換する

```
# inetconv -i /etc/inet/inetd.conf
```

実行制御スクリプトの使用 (作業マップ)

作業	説明	参照先
サービスを起動または停止します。	実行制御スクリプトを使用してサービスを起動または停止します。	396 ページの「実行制御スクリプトを使用して従来のサービスを起動または停止する方法」
実行制御スクリプトを追加します。	実行制御スクリプトを作成し、それを /etc/init.d ディレクトリに追加します。	397 ページの「実行制御スクリプトを追加する方法」
実行制御スクリプトを無効にします。	ファイル名を変更することで実行制御スクリプトを無効にします。	398 ページの「実行制御スクリプトを無効にする方法」

実行制御スクリプトの使用

▼ 実行制御スクリプトを使用して従来のサービスを起動または停止する方法

実行レベルごとに対応するスクリプトを持つことの利点は、/etc/init.d ディレクトリ内の個々のスクリプトを実行することによって、システムの実行レベルを変更しないで (現在の実行レベルの) システムサービスを停止できる点です。

- 1 スーパーユーザーになるか、Service Management 権利プロファイルを含む役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成」を参照してください。

- 2 システムサービスを停止します。

```
# /etc/init.d/filename
stop
```

- 3 システムサービスを再開します。

```
# /etc/init.d/filename
start
```

- 4 サービスが停止または起動されたことを確認します。

```
# pgrep -f service
```

例 18-15 実行制御スクリプトを使用してサービスを起動または停止する

NFS サーバーデーモンを停止するには、次のように入力します。

```
# /etc/init.d/nfs.server stop
# pgrep -f nfs
```

そして、NFS サーバーデーモンを再開するには、次のように入力します。

```
# /etc/init.d/nfs.server start
# pgrep -f nfs
101773
101750
102053
101748
101793
102114
# pgrep -f nfs -d, | xargs ps -fp
      UID      PID  PPID  C   STIME TTY      TIME CMD
daemon 101748    1    0   Sep 01 ?    0:06 /usr/lib/nfs/nfsmapid
daemon 101750    1    0   Sep 01 ?    26:27 /usr/lib/nfs/lockd
daemon 101773    1    0   Sep 01 ?    5:27 /usr/lib/nfs/statd
root   101793    1    0   Sep 01 ?    19:42 /usr/lib/nfs/mountd
daemon 102053    1    0   Sep 01 ?   2270:37 /usr/lib/nfs/nfsd
daemon 102114    1    0   Sep 01 ?    0:35 /usr/lib/nfs/nfs4cbd
```

▼ 実行制御スクリプトを追加する方法

サービスを起動または停止するための実行制御スクリプトを追加するには、そのスクリプトを/etc/init.dディレクトリにコピーします。次に、サービスを起動または停止するrcn.dディレクトリ内にリンクを作成します。

実行制御スクリプトの命名法の詳細については、/etc/rcn.dディレクトリ内のREADME ファイルを参照してください。次に、実行制御スクリプトを追加する方法について説明します。

- 1 スーパーユーザーになるか、Service Management 権利プロファイルを含む役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成」を参照してください。

- 2 スクリプトを /etc/init.d ディレクトリに追加します。

```
# cp filename
/etc/init.d
# chmod 0744 /etc/init.d/filename
# chown root:sys /etc/init.d/filename
```

- 3 適切な rcn.d ディレクトリへのリンクを作成します。

```
# cd /etc/init.d
# ln filename /etc/rc2.d/
Snnfilename
# ln filename /etc/rc
n.d/Knnfilename
```

- 4 スクリプトのリンクが指定のディレクトリ内にあることを確認します。

```
# ls /etc/init.d/*filename
/etc/rc2.d/*filename /etc/rc
n.d/*filename
```

例 18-16 実行制御スクリプトを追加する

次の例は、xyz サービスの実行制御スクリプトを追加する方法を示しています。

```
# cp xyz /etc/init.d
# chmod 0744 /etc/init.d/xyz
# chown root:sys /etc/init.d/xyz
# cd /etc/init.d
# ln xyz /etc/rc2.d/S99xyz
# ln xyz /etc/rc0.d/K99xyz
# ls /etc/init.d/*xyz /etc/rc2.d/*xyz /etc/rc0.d/*xyz
```

▼ 実行制御スクリプトを無効にする方法

実行制御スクリプトを無効にするには、ファイル名の先頭に下線(_)を付けてファイル名を変更します。下線またはドット(.)で始まるファイルは実行されません。接尾辞を追加してファイルをコピーすると、両方のファイルが実行されます。

- 1 スーパーユーザーになるか、Service Management 権利プロファイルを含む役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成」を参照してください。

- 2 新しいファイルの先頭に下線(_)を付けて、スクリプト名を変更します。

```
# cd /etc/rcn
.d
# mv filename _filename
```

- 3 スクリプトの名前が変更されたことを確認します。

```
# ls _*
_filename
```

例 18-17 実行制御スクリプトを無効にする

次の例は、S99datainit スクリプトの名前を変更する方法を示しています。

```
# cd /etc/rc2.d
# mv S99datainit _S99datainit
# ls _*
_S99datainit
```

サービス管理機能のトラブルシューティング

▼ 起動していないサービスをデバッグする

この手順では、印刷サービスが無効になっています。

- 1 スーパーユーザーになるか、Service Management 権利プロファイルを含む役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成](#)」を参照してください。

- 2 ハングしたサービスに関する情報を要求します。

```
# svcs -xv
svc:/application/print/server:default (LP Print Service)
  State: disabled since Wed 13 Oct 2004 02:20:37 PM PDT
Reason: Disabled by an administrator.
  See: http://sun.com/msg/SMF-8000-05
  See: man -M /usr/share/man -s 1M lpsched
Impact: 2 services are not running:
      svc:/application/print/rfc1179:default
      svc:/application/print/ipp-listener:default
```

-x オプションを指定すると、影響を受けたサービスインスタンスに関する情報も表示されます。

- 3 サービスを有効にします。

```
# svcadm enable application/print/server
```

▼ 破壊されたりポジトリを修復する方法

この手順は、破壊されたりポジトリをデフォルトのリポジトリと交換する方法を示しています。リポジトリデーモン `svc.configd` が起動すると、設定リポジトリの整合性チェックが行われます。このリポジトリは、`/etc/svc/repository.db` に格納されています。このリポジトリは、次のいずれかが原因で破壊されることがあります。

- ディスク障害
- ハードウェアのバグ
- ソフトウェアのバグ
- 過失によるファイルの上書き

整合性チェックに失敗すると、`svc.configd` デーモンは次のようなメッセージをコンソールに書き出します。

```
svc.configd: smf(5) database integrity check of:
```

```
    /etc/svc/repository.db
```

```
failed. The database might be damaged or a media error might have
prevented it from being verified. Additional information useful to
your service provider is in:
```

```
    /etc/svc/volatile/db_errors
```

```
The system will not be able to boot until you have restored a working
database.  svc.startd(1M) will provide a sulogin(1M) prompt for recovery
purposes.  The command:
```

```
    /lib/svc/bin/restore_repository
```

```
can be run to restore a backup version of your repository. See
http://sun.com/msg/SMF-8000-MY for more information.
```

その後 `svc.startd` デーモンが終了し、`sulogin` が起動して保守作業を行えるようになります。

- 1 `sulogin` プロンプトで `root` パスワードを入力します。`sulogin` では、`root` ユーザーはシステム保守モードに入ってシステムを修復できます。

- 2 次のコマンドを実行します。

```
# /lib/svc/bin/restore_repository
```


このコマンドを実行すると、破壊されていないバックアップの復元に必要な手順が示されます。SMF では、システムの重要な局面でリポジトリのバックアップが自動的に作成されます。詳細は、[371 ページの「SMF リポジトリのバックアップ」](#)を参照してください。

`/lib/svc/bin/restore_repository` コマンドが起動すると、次のようなメッセージが表示されます。

Repository Restore utility

See <http://sun.com/msg/SMF-8000-MY> for more information on the use of this script to restore backup copies of the smf(5) repository.

If there are any problems which need human intervention, this script will give instructions and then exit back to your shell.

Note that upon full completion of this script, the system will be rebooted using `reboot(1M)`, which will interrupt any active services.

復元するシステムがローカルゾーンでない場合は、読み取り権と書き込み権を付けて `/` ファイルシステムおよび `/usr` ファイルシステムをマウントし直しデータベースを復元する方法が表示されます。これらの説明を表示したあと、スクリプトは終了します。エラーの発生に特に注意しながら、これらの説明に従います。

書き込み権を付けてルート (`/`) ファイルシステムをマウントしたあと、またはシステムがローカルゾーンである場合は、復元するリポジトリのバックアップを選択するよう求められます。

The following backups of `/etc/svc/repository.db` exists, from oldest to newest:

... *list of backups* ...

バックアップには、バックアップのタイプとバックアップが作成された時間に基づいて名前が付けられています。`boot` で始まっているのは、システムのブート後、リポジトリに対して最初の変更が行われる前に作成されたバックアップです。`manifest_import` で始まっているのは、`svc:/system/manifest-import:default` のプロセス終了後に作成されたバックアップです。バックアップ時間は、`YYYYMMDD_HHMMSS` 形式で記録されます。

3 適切な応答を入力します。

通常は、最新のバックアップオプションを選択します。

Please enter one of:

- 1) `boot`, for the most recent post-boot backup
- 2) `manifest_import`, for the most recent `manifest_import` backup.
- 3) a specific backup repository from the above list
- 4) `-seed-`, the initial starting repository. (All customizations will be lost.)

5) `-quit-`, to cancel.

Enter response [boot]:

復元するバックアップを指定しないでEnterキーを押した場合は、[]で囲まれたデフォルトの応答が選択されます。`-quit-`を選択すると、`restore_repository` スクリプトが終了して、シェルスクリプトに戻ります。

注--`-seed-`を選択すると、`seed` リポジトリが復元されます。このリポジトリは、初期インストールとアップグレード時に使用する目的で作成されたものです。`seed` リポジトリを回復の目的で使用するの、最後の手段にしてください。

復元するバックアップを選択すると、妥当性検査が行われ、その整合性がチェックされます。なんらかの問題があると、`restore_repository` コマンドによってエラーメッセージが表示され、別のバックアップを選択するよう促されます。有効なバックアップを選択すると、次の情報が表示され、最終確認を入力するよう促されます。

After confirmation, the following steps will be taken:

```
svc.startd(1M) and svc.configd(1M) will be quiesced, if running.
/etc/svc/repository.db
-- renamed --> /etc/svc/repository.db_old_YYYYMMDD_HHMMSS
/etc/svc/volatile/db_errors
-- copied --> /etc/svc/repository.db_old_YYYYMMDD_HHMMSS_errors
repository_to_restore
-- copied --> /etc/svc/repository.db
and the system will be rebooted with reboot(1M).
```

Proceed [yes/no]?

4 yes と入力して障害を修復します。

`restore_repository` コマンドが表示されたアクションをすべて実行すると、システムがリブートします。

▼ どのサービスも起動しないでブートする方法

サービスの起動に関する問題が発生した場合、ブート時にシステムがハングアップする可能性があります。この手順は、そうした問題のトラブルシューティングを行う方法を示しています。

- 1 どのサービスも起動しないでブートします。
このコマンドを実行すると、`svc.startd` デーモンはすべてのサービスを一時的に無効にし、コンソール上で `sulogin` を起動します。
`ok boot -m milestone=none`
- 2 システムに `root` としてログインします。
- 3 すべてのサービスを有効にします。
`# svcadm milestone all`
- 4 ブートプロセスがどこでハングアップするのかを確認します。
ブートプロセスがハングアップしたら、動作していないサービスを確認するために、`svcs -a` を実行します。`/var/svc/log` のログファイル内でエラーメッセージの有無を確認します。
- 5 問題が解決したら、すべてのサービスが起動していることを確認します。
 - a. 必要なサービスがすべてオンラインになっていることを確認します。
`# svcs -x`
 - b. `console-login` サービスの依存関係に問題がないことを確認します。
このコマンドを使えば、コンソール上の `login` プロセスが実行されるかどうかを確認できます。
`# svcs -l system/console-login:default`
- 6 通常のブートプロセスを継続します。

▼ ブート中に `system/filesystem/local:default` サービスで障害が発生した場合に `sulogin` プロンプトを強制的に表示する方法

Solaris OS のブートに必要なでないローカルファイルシステム

は、`svc:/system/filesystem/local:default` サービスによってマウントされます。これらのファイルシステムのいずれかがマウントできない場合、このサービスは保守状態になります。システムの起動は続行され、`filesystem/local` に依存しないサービスは起動されます。依存関係を介して起動されるサービスで、オンラインになるために `filesystem/local` を必要とするものは、起動されません。

サービスで障害が発生した場合に、システムの起動の続行を許可する代わりに、ただちに `sulogin` プロンプトを表示するようシステムの設定を変更するには、次の手順に従ってください。

- 1 system/console-login サービスを変更します。

```
# svccfg -s svc:/system/console-login
svc:/system/console-login> addpg site,filesystem-local dependency
svc:/system/console-login> setprop site,filesystem-local/entities = fmri: svc:/system/filesystem/local

svc:/system/console-login> setprop site,filesystem-local/grouping = astring: require_all

svc:/system/console-login> setprop site,filesystem-local/restart_on = astring: none

svc:/system/console-login> setprop site,filesystem-local/type = astring: service

svc:/system/console-login> end
```

- 2 サービスを再表示します。

```
# svcadm refresh console-login
```

例 18-18 JumpStart を使用して sulogin プロンプトを強制的に表示する

次に示すコマンドをスクリプトに保存し、/etc/rcS.d/S01site-customfs として保存します。

```
#!/bin/sh
#
# This script adds a dependency from console-login -> filesystem/local
# This forces the system to stop the boot process and drop to an sulogin prompt
# if any file system in filesystem/local fails to mount.

PATH=/usr/sbin:/usr/bin
export PATH

svccfg -s svc:/system/console-login << EOF
addpg site,filesystem-local dependency
setprop site,filesystem-local/entities = fmri: svc:/system/filesystem/local
setprop site,filesystem-local/grouping = astring: require_all
setprop site,filesystem-local/restart_on = astring: none
setprop site,filesystem-local/type = astring: service
EOF

svcadm refresh svc:/system/console-login

[ -f /etc/rcS.d/S01site-customfs ] &&
rm -f /etc/rcS.d/S01site-customfs
```

注意事項 system/filesystem/local:default サービスで障害が発生したときは、svcs -vx コマンドを使用して障害を特定するようにしてください。障害が修正されたら、次のコマンドでエラー状態をクリアし、システムのブートを続行できるようにします。

```
svcadm clear filesystem/local。
```

ソフトウェアの管理 (概要)

ソフトウェアの管理には、スタンドアロンシステム、サーバー、およびそのクライアントへのソフトウェアの追加や削除が含まれます。この章では、ソフトウェアのインストールや管理に使用できる各種ツールについて説明します。

この章では、新しいシステムでの Solaris オペレーティングシステム (Solaris OS) のインストールについては説明しません。また、新バージョンの Solaris OS のインストールやアップグレードについても説明しません。Solaris OS のインストールやアップグレードについては、『[Solaris 10 インストールガイド \(基本編\)](#)』を参照してください。

この章の内容は以下のとおりです。

- 406 ページの「Solaris オペレーティングシステムでのソフトウェア管理における新機能」
- 409 ページの「ソフトウェア管理作業についての参照先」
- 410 ページの「ソフトウェアパッケージの概要」
- 414 ページの「ソフトウェアパッケージを管理するためのツール」
- 415 ページの「ソフトウェアパッケージの追加または削除 (pkgadd)」
- 416 ページの「ソフトウェアパッケージの追加にあたっての重要な注意点 (pkgadd)」
- 416 ページの「パッケージの削除に関するガイドライン (pkgrm)」
- 417 ページの「ゾーンに対応していない Solaris リリースでソフトウェアパッケージおよびパッチを追加および削除する際の制限」
- 418 ページの「パッケージ追加時のユーザーの対話操作を省略する (pkgadd)」

Solaris OS でのパッケージとパッチの管理についての最新情報は、<http://www.sun.com/bigadmin/patches/> を参照してください。

ソフトウェアを管理する手順については、第 20 章「Solaris システム管理ツールによるソフトウェアの管理 (手順)」および第 21 章「パッケージコマンドによるソフトウェアの管理 (手順)」を参照してください。

ゾーンがインストールされている Solaris システムでソフトウェアを管理する方法については、『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』の第 25 章「ゾーンがインストールされている Solaris システムでのパッケージとパッチの追加および削除 (手順)」を参照してください。

注 - **Solaris 10 5/08:** この情報は Solaris 10 5/08 リリースで追加されたものですが、すべての Solaris 10 OS に当てはまります。使用している Solaris システムを登録するには、<https://inventory.sun.com/inventory/> にアクセスしてください。Sun Inventory を使用してハードウェア、ソフトウェア、およびオペレーティングシステムを登録する方法については、[Sun Inventory Information Center \(http://wikis.sun.com/display/SunInventory/Sun+Inventory\)](http://wikis.sun.com/display/SunInventory/Sun+Inventory) を参照してください。

Sun Ops Center を使ってデータセンターのシステムのプロビジョニング、更新、および管理を行う場合は、Sun Ops Center のドキュメントを参照し、ソフトウェアを登録する方法についての情報を確認してください。

Solaris オペレーティングシステムでのソフトウェア管理における新機能

この節では、この Solaris リリースの新しいソフトウェア管理機能について説明します。

Solaris の新機能の全一覧や各 Solaris リリースの説明については、『Solaris 10 の概要』を参照してください。

ゾーンをサポートするためのパッケージおよびパッチツールの拡張

Solaris 10 10/09: このリリースから、複数の非大域ゾーンがインストールされているシステムをサポートするために、パッケージおよびパッチツールに対して次の拡張が行われています。

- ゾーンの並列パッチ

ゾーンの並列パッチ機能は、標準の Solaris 10 パッチ適用ツールを強化して、非大域ゾーンに並列してパッチを適用できるようにします。この機能拡張により、システムのパッチにかかる時間が短縮され、複数の非大域ゾーンがシステムにインストールされている場合のパッチ処理のパフォーマンスが向上します。ゾーンは分離された環境のため、複数のゾーンに対する並列パッチを安全に行うことができます。この機能は、疎ルートゾーンおよび完全ルートゾーンの両方に適用されます。

これらの機能強化は、主に `patchadd` および `patchrm` コマンドに対する変更によるものです。新しい構成ファイル `/etc/patch/pdo.conf` は、並列してパッチするゾーンの数に定義する数値を指定するように変更することができます。

注 - Solaris 10 10/09 リリース以上を実行していない場合は、この機能はパッチ 119254-66 以降のリビジョン (SPARC) および 119255-66 以降のリビジョン (x86) で実装されます。これまでどおり、大域ゾーンには、非大域ゾーンよりも前にパッチが適用されます。

詳細は、[patchadd\(1M\)](#) と [patchrm\(1M\)](#) のマニュアルページを参照してください。

- **SVr4** パッケージコマンドのターボチャージ

この機能は、SVr4 パッケージコマンドを使用して、Solaris OS のインストール、Solaris Live Upgrade の使用、または非大域ゾーンのインストールにかかる時間を大幅に短縮します。この機能は無効にできません。

ゾーン環境での Solaris パッケージおよびパッチツールの使用をサポートするこれらの機能強化の詳細については、次のドキュメントを参照してください。

- 『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』の第 25 章「ゾーンがインストールされている Solaris システムでのパッケージとパッチの追加および削除 (手順)」
- 『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』の第 23 章「非大域ゾーンの移動と移行 (手順)」
- 『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』の第 28 章「非大域ゾーンにインストールされている Solaris 10 システムのアップグレード」

遅延起動パッチ

パッチツールが大規模なパッチを処理できるように変更されました。パッチ 119254-42 および 119255-42 以降、パッチインストールコマンド `patchadd` および `patchrm` が変更され、新しい機能を提供する特定のパッチの処理方法が変わりました。この変更は、これらのパッチをどの Solaris 10 リリースにインストールする場合にも影響を与えます。これらの「遅延起動」パッチによって、機能パッチで提供される大規模な変更の処理能力が向上します。

詳細は、<http://www.sun.com/bigadmin/sundocs/articles/patch-wn.jsp> を参照してください。

Solaris OS に組み込まれた Common Agent Container

Common Agent Container は、Java 管理アプリケーションのコンテナを実装する、スタンドアロンの Java プログラムです。このプログラムが提供する管理インフラストラクチャーは、Java Management Extensions (JMX) および Java Dynamic Management Kit (Java DMK) ベースの管理機能向けに設計されています。このソフトウェアは SUNWcacaort パッケージによってインストールされ、`/usr/lib/cacao` ディレクトリに置かれます。

通常、このコンテナは表示されません。

ただし、次の 2 つの場合は、コンテナデーモンとの対話が必要になることがあります。

- Common Agent Container 用に予約されているネットワークポートを、別のアプリケーションが使用しようとしている場合。
- 証明書ストアが危殆化した場合。この衝突が発生した場合は、Common Agent Container 証明書鍵の再生成が必要になることがあります。

これらの問題のトラブルシューティングの方法については、『[Solaris のシステム管理 \(上級編\)](#)』の「[Solaris OS での共通エージェントコンテナの問題解決](#)」を参照してください。

patchadd -M コマンドが複数のパッチを処理する方法の改善

Solaris 10: Solaris 10 リリース以降では、`patchadd -M` コマンドの機能が改善され、複数のパッチおよびパッチ間の依存関係をより効果的かつ効率的に処理できるようになりました。このため、このコマンドを使用するときに、パッチ ID を番号順に指定する必要がなくなりました。

パッチ ID を指定しないで `patchadd -M` コマンドを使用した場合でも、ディレクトリ内のすべてのパッチが自動的にシステムにインストールされます。特定のパッチをインストールする場合は、`patchadd -M` コマンドを使用するときにそのパッチ ID を指定する必要があります。

詳細は、[patchadd\(1M\)](#) のマニュアルページを参照してください。

パッケージおよびパッチツールの拡張

Solaris 10: 初期の Solaris 10 3/05 リリースでは、Solaris パッケージツールとパッチツールが拡張されており、パフォーマンスの向上と機能の拡張が図られています。

そうした拡張の一部として、ファイルをパッケージに対応付けするための新しいオプションが、`pkgchk` コマンドに追加されました。ファイルとパッケージ間のマッピングを行うには、`grep -pattern/var/sadm/install/contents` の代わりに `pkgchk P` オプションを使用します。`-P` オプションでは部分パスが使えます。このオプションを `-l` オプションと組み合わせると、指定した部分パスを含むファイルの情報が一覧表示されます。詳細は、[467 ページの「インストール済みオブジェクトの整合性を検査する方法 \(pkgchk -p、pkgchk -P\)」](#) と、[pkgchk\(1M\)](#) のマニュアルページを参照してください。

ソフトウェア管理作業についての参照先

ソフトウェアを管理する手順については、次の表を参照してください。

ソフトウェア管理作業	詳細
Solaris ソフトウェアのインストール。	『Solaris 10 インストールガイド (基本編)』
インストール後の Solaris ソフトウェアパッケージの追加または削除。	第 20 章「Solaris システム管理ツールによるソフトウェアの管理 (手順)」および第 21 章「パッケージコマンドによるソフトウェアの管理 (手順)」
インストール後の Solaris パッチの追加または削除。	476 ページの「Solaris オペレーティングシステムでのパッチ管理」
ソフトウェアパッケージ問題のトラブルシューティング。	『Solaris のシステム管理 (上級編)』の第 21 章「ソフトウェアパッケージで発生する問題の解決 (手順)」

ソフトウェアパッケージの概要

ソフトウェア管理には、ソフトウェア製品のインストールと削除が含まれます。Sun および Sun 以外の ISV は、ソフトウェアを 1 つ以上の「パッケージ」の集まりとして提供します。

「パッケージング」という用語は一般に、ソフトウェア製品が使用されるシステムにその製品を配布してインストールする方法を指します。パッケージは、定義済みフォーマットによるファイルとディレクトリの集まりです。このフォーマットは、アプリケーションバイナリインタフェース (ABI) に準拠します。ABI は、System V インタフェース定義を補足するものです。Solaris OS には、このフォーマットを解釈し、パッケージをインストールまたは削除したり、パッケージインストールを検査したりする方法を提供する 1 組のユーティリティがあります。

「パッチ」とは、Solaris OS 内またはその他のサポート対象ソフトウェア内の既知または潜在的な問題に対する修正をまとめたものです。また、パッチは、特定のソフトウェアリリースに対する新機能や機能拡張の提供も行います。パッチは、既存のファイルやディレクトリを置換または更新するファイルやディレクトリから構成されます。ほとんどの Solaris パッチは、一連の疎パッケージとして提供されます。

「疎パッケージ」には、そのパッケージが Solaris ディストリビューションの一部として初めて提供されたあとで変更されたオブジェクトのみが含まれます。疎パッケージには、ソフトウェア更新を提供するために完全なパッケージとして再配布した場合よりも小さなパッチが収められています。また、疎パッケージとして提供すると、顧客の環境に対する変更も最小限に抑えられます。パッチの詳細については、[476 ページ](#)の「[Solaris オペレーティングシステムでのパッチ管理](#)」を参照してください。

署名付きのパッケージ、パッチ、およびソフトウェア更新

パッケージには、デジタル署名を含めることができます。有効なデジタル署名付きのパッケージは、署名が適用された以降はパッケージの変更が行われていないことを保証します。署名付きパッケージを使用すると、パッケージをシステムに追加する前にデジタル署名を検証できるため、パッケージを安全にダウンロードまたは追加できます。

同じことが、署名付きパッチにも言えます。有効なデジタル署名付きのパッチは、署名が適用された以降はパッチの変更が行われていないことを保証します。署名付きパッチを使用すると、パッチをシステムに適用する前にデジタル署名を検証できるため、パッチを安全にダウンロードまたは適用できます。

署名付きパッチをシステムに「適用」する方法の詳細については、[479 ページ](#)の「[patchadd コマンドによる Solaris パッチの管理 \(作業マップ\)](#)」を参照してください。

署名付きパッケージの作成方法については、『[アプリケーションパッケージ開発者ガイド](#)』を参照してください。

署名付きパッケージは、デジタル署名が含まれる点以外は、署名なしパッケージとまったく同一です。このパッケージのインストール、照会、または削除は、既存の Solaris パッケージツールを使って行うことができます。また、署名付きパッケージと署名なしパッケージは、バイナリレベルで互換性があります。

pkgadd と patchadd を使ってデジタル署名付きのパッケージまたはパッチをシステムに追加するには、信頼される証明書を使ってパッケージキーストアを設定しておく必要があります。これらの証明書は、パッケージやパッチ上のデジタル署名の有効性を確認する際に使用されます。

次では、署名付きのパッケージおよびパッチに関する一般的な用語を説明します。

キーストア 証明書と鍵を格納するためのリポジトリ。それらの情報が必要になると、このリポジトリが検索されます。

- Java キーストア – Solaris リリースでデフォルトでインストールされる証明書のリポジトリ。Java キーストアは通常、`/usr/j2se/jre/lib/security` ディレクトリに格納されます。
- パッケージキーストア – 署名付きパッケージおよびパッチをシステムに追加する際にインポートする証明書用のリポジトリ。

パッケージキーストアは、デフォルトで `/var/sadm/security` ディレクトリに格納されます。

信頼される証明書 別のエンティティに属する公開鍵を備えた証明書。「信頼される証明書」という呼び名は、証明書に含まれている公開鍵が、その証明書のサブジェクトまたは所有者によって示された本人のものであることを、キーストアの所有者が信頼することに由来しています。この信頼を表明するために、証明書の発行者はその証明書に署名します。

信頼される証明書は、署名を検証する際やセキュリティー保護されたサーバー (SSL サーバー) への接続を確立する際に使用されます。

ユーザー鍵 暗号鍵に関する機密情報を保持します。この情報は、不正なアクセスを防ぐために、セキュリティーが施された形式で格納されます。ユーザー鍵は、ユーザーの非公開鍵と対応する公開鍵証明書から構成されます。

pkgadd コマンドまたは patchadd コマンドを使って署名付きのパッケージまたはパッチをシステムに追加する手順は、基本的に次の3つから構成されます。

1. pkgadm コマンドを使ってシステムのパッケージキーストアに証明書を追加する
2. (省略可能) pkgadm コマンドを使って証明書を一覧表示する
3. pkgadd コマンドを使ってパッケージを追加するか、patchadd コマンドを使ってパッチを適用する

署名付きパッケージをシステムに追加する手順については、[451 ページの「pkgadd コマンドによる署名付きパッケージの追加および削除 \(作業マップ\)」](#)を参照してください。

patchadd コマンドを使って署名付きパッチをシステムに適用するステップ単位の手順については、[479 ページの「patchadd コマンドによる Solaris パッチの管理 \(作業マップ\)」](#)を参照してください。

Sun の証明書による署名付きパッケージおよびパッチの検証

Sun の証明書をインポートするときに特別なパスワードを使用することで、パッケージキーストアへのアクセスを保護できます。

pkgadm listcert コマンドを使えば、ローカルのパッケージキーストア内に格納されている証明書の情報を表示できます。次に例を示します。

```
# pkgadm listcert -P pass:store-pass
Keystore Alias: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O
Common Name: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O
Certificate Type: Trusted Certificate
Issuer Common Name: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O
Validity Dates: <May 18 00:00:00 1998 GMT> - <Aug 1 23:59:59 2028 GMT>
MD5 Fingerprint: 2D:BB:E5:25:D3:D1:65:82:3A:B7:0E:FA:E6:EB:E2:E1
SHA1 Fingerprint: B3:EA:C4:47:76:C9:C8:1C:EA:F2:9D:95:B6:CC:A0:08:1B:67:EC:9D
```

次では、pkgadm listcert コマンドの出力について説明します。

キーストア別名	印刷、署名、削除のいずれかの目的で証明書を取得する場合、この名前を証明書の参照名として使用する必要があります。
共通名	証明書の共通の名前。信頼される証明書の場合、この名前はキーストア別名と同じになります。
証明書の種類	次の2つの種類があります。 ■ Trusted Certificate (信頼される証明書) – ほかの証明書を検証する際の信頼基盤として使用可能な証明書。信頼される証明書には非公開鍵は関連付けられません。

	■ Signing Certificate (署名証明書) – パッケージまたはパッチへの署名用として使用可能な証明書。署名証明書には非公開鍵が関連付けられます。
発行者の共通名	この証明書の署名、発行をしたエンティティの名前。信頼される証明書発行局 (CA、Certificate Authority) が発行した証明書の場合、発行者の共通名は、共通名と同じになります。
有効な日付	証明書の有効な日付。
MD5 指紋	証明書の MD5 ダイジェスト。このダイジェストを使えば、その証明書が証明元からの転送中に改変されなかったかどうかを検証できます。
SHA1 指紋	MD5 指紋に似ていますが、異なるアルゴリズムを使って計算されます。

証明書を認証するには、各証明書の MD5 ハッシュおよび SHA1 ハッシュ(「指紋」とも呼ばれる)を、発行者によって公開されている既知の正しい指紋と比較します。

Sun の信頼される証明書のインポート

署名付きパッケージおよびパッチを追加するための Sun の信頼される証明書を取得するには、次の方法を使います。

- **Java キーストア** – Sun のルート CA 証明書をインポートします。この証明書は、この Solaris リリースをインストールする際に Java キーストア内にデフォルトで格納されます。
- **Sun の公開鍵インフラストラクチャー (PKI) サイト** – システム上に Java キーストアが存在していない場合は、次のサイトから証明書をインポートできます。

パッケージキーストアのセットアップ

証明書の格納された Java キーストアがシステム上にすでに存在している場合、Sun Microsystems のルート CA 証明書を Java キーストアから `keytool` コマンドを使ってエクスポートできます。続いて、その証明書を、`pkgadm` コマンドを使ってパッケージキーストア内にインポートします。

ルート CA 証明書のパッケージキーストアへのインポートが完了すると、`pkgadd` コマンドや `patchadd` コマンドを使って署名付きパッケージおよびパッチをシステムに追加できるようになります。

注 – なお、Sun Microsystems のルート証明書が必要となるのは、Sun が署名したパッチやパッケージを追加する場合のみです。

証明書をパッケージキーストアにインポートする手順については、[452 ページの「信頼される証明書を Java キーストアからインポートする方法 \(pkgadm addcert\)」](#)を参照してください。

pkgadd コマンドを使って署名付きパッケージを追加する完全な手順については、[457 ページの「署名付きパッケージを追加する方法 \(pkgadd\)」](#)を参照してください。

ソフトウェアパッケージを管理するためのツール

次の表では、Solaris リリースがインストールされたシステム上でソフトウェアパッケージを追加および削除するためのツールについて説明します。

表 19-1 ソフトウェアパッケージを管理するためのツールまたはコマンド

ツールまたはコマンド	説明	マニュアルページ
installer	Solaris インストール GUI などのインストーラを起動して、Solaris メディアからソフトウェアを追加します。インストーラがローカル、リモートのいずれかで利用可能になっている必要があります。	installer(1M)
prodreg (GUI)	インストーラを起動して、ソフトウェア製品情報を追加、削除、または表示します。Solaris インストール GUI プログラムや Solaris pkgadd コマンドを使って初めからインストールされているソフトウェア製品に関する情報を削除または表示する場合は、Solaris Product Registry を使用します。	prodreg(1M)
Solaris Product Registry prodreg ビューア (CLI)	prodreg コマンドでは、Solaris インストール GUI や Solaris pkgadd コマンドを使ってインストールされているソフトウェア製品に関する情報を削除または表示できます。	prodreg(1M)
pkgadd	署名付きまたは署名なしのソフトウェアパッケージをインストールします。	pkgadd(1M)

表 19-1 ソフトウェアパッケージを管理するためのツールまたはコマンド (続き)

ツールまたはコマンド	説明	マニュアルページ
pkgadm	署名付きパッケージおよびパッチの管理に使用する鍵と証明書を保守します。	pkgadm(1M)
pkgchk	ソフトウェアパッケージのインストールを検査します。	pkgchk(1M)
pkginfo	ソフトウェアパッケージ情報を表示します。	pkginfo(1)
pkgparam	ソフトウェアパッケージのパラメータ値を表示します。	pkgparam(1)
pkgrm	ソフトウェアパッケージを削除します。	pkgrm(1M)
pkgtrans	インストール可能なパッケージをある形式から別の形式に変換します。-g オプションを指定して <code>pkgtrans</code> コマンドを実行すると、署名が生成され、該当するデータストリーム内に格納されます。	pkgtrans(1)

これらのコマンドの詳細は、第 20 章「Solaris システム管理ツールによるソフトウェアの管理 (手順)」および第 21 章「パッケージコマンドによるソフトウェアの管理 (手順)」を参照してください。

ソフトウェアパッケージの追加または削除 (pkgadd)

表 19-1 に記載されているソフトウェア管理ツールはすべて、ソフトウェアを追加したり、インストール済みソフトウェアを削除したり、それらのソフトウェアに関する情報を照会したりするために使用されます。Solaris Product Registry の `prodreg` ビューアと Solaris インストール GUI はどちらも、Solaris Product Registry に格納されているインストールデータにアクセスします。`pkgadd` コマンドや `pkgrm` コマンドといったパッケージツールも、インストールデータにアクセスしたりインストールデータを変更したりします。

パッケージを追加する際、`pkgadd` コマンドは、ファイルを解凍して、インストール用メディアからローカルシステムのディスクにコピーします。パッケージを削除する際、`pkgrm` コマンドは、そのパッケージに関連するファイルがほかのパッケージと共有されている場合を除いて、それらをすべて削除します。

パッケージファイルはパッケージ形式で配布され、配布されたままの状態では使用できません。pkgadd コマンドは、ソフトウェアパッケージの制御ファイルを解釈してから、製品ファイルを解凍して、システムのローカルディスクにインストールします。

pkgadd と pkgrm の各コマンドは、標準の場所にそのログ出力を記録しませんが、インストールまたは削除されるパッケージを常時追跡しています。pkgadd と pkgrm の各コマンドは、インストールまたは削除されたパッケージに関する情報をソフトウェア製品用データベースに格納します。

このデータベースを更新することによって、pkgadd および pkgrm の各コマンドは、システムにインストールされたすべてのソフトウェア製品の記録を追跡します。

ソフトウェアパッケージの追加にあたっての重要な注意点 (pkgadd)

システム上でパッケージをインストールまたは削除する場合には、次の点に注意する必要があります。

- パッケージの命名規則 – Sun パッケージは、SUNWacctr、SUNWadmap、SUNWcsu などのように、必ず接頭辞 SUNW で始まります。Sun 以外のパッケージは、通常、その会社を表す接頭辞で始まります。
- インストール済みのソフトウェア – Solaris インストール GUI、Solaris Product Registry の prodreg ビューア (GUI または CLI)、pkginfo コマンドのいずれかを使えば、システム上にすでにインストールされているソフトウェアを特定できます。
- サーバーとクライアントによるソフトウェアの共有の状態 – クライアントのソフトウェアは、一部がサーバーに、一部がクライアントに置かれる場合があります。このような場合、クライアントにソフトウェアを追加するには、サーバーとクライアントの両方にパッケージを追加する必要があります。

パッケージの削除に関するガイドライン (pkgrm)

パッケージを削除する場合、rm コマンドを使用するかもしれませんが、代わりに表 19-1 に記載されているツールのいずれかを使用するようにしてください。たとえば、rm コマンドを使用すると、バイナリ実行可能ファイルを削除することができます。しかしながら、そのような操作は、pkgrm コマンドを使ってそのバイナリ実行可能ファイルを含むソフトウェアパッケージを削除する操作と等価ではありません。rm コマンドを使用してパッケージのファイルを削除すると、ソフトウェア製品用データベースが破壊されます。1 つのファイルだけを削除する場合は、removef コマンドを使用することができます。このコマンドを使用した場合、該当ファイルがパッケージから削除されるように、ソフトウェア製品データベースが正しく更新されます。詳細は、[removef\(1M\)](#) のマニュアルページを参照してください。

複数のバージョンのパッケージをインストールしておきたい場合は、`pkgadd` コマンドを使ってインストール済みパッケージとは別のディレクトリに新しいバージョンをインストールしてください。たとえば、複数のバージョンの文書処理アプリケーションをインストールしておきたい場合などが考えられます。パッケージがインストールされているディレクトリは、ベースディレクトリと呼ばれます。ベースディレクトリは、管理ファイルと呼ばれる特殊ファイルに `basedir` キーワードを設定することによって操作できます。管理ファイルの使用方法やベースディレクトリの設定方法の詳細は、[418 ページの「パッケージ追加時のユーザーの対話操作を省略する \(pkgadd\)」](#) と、[admin\(4\)](#) のマニュアルページを参照してください。

注-Solaris ソフトウェアをインストールするときにアップグレードオプションを使用すると、Solaris インストール用ソフトウェアは、ソフトウェア製品用データベースを調べて、すでにシステムにインストールされている製品があるかどうかを確認します。

ゾーンに対応していない **Solaris** リリースでソフトウェアパッケージおよびパッチを追加および削除する際の制限

ゾーンに対応していない Solaris リリースが稼働するシステム上では、`-R` オプションを指定できるコマンドを使って非大域ゾーンがインストールされた大域ゾーンの代替ルートパスを指定しても、そのコマンドは正しく動作しません。

それらのコマンドは次のとおりです。

- `pkgadd`
- `pkgrm`
- `patchadd`
- `patchrm`

[pkgadd\(1M\)](#)、[pkgrm\(1M\)](#)、[patchadd\(1M\)](#)、および [patchrm\(1M\)](#) のマニュアルページを参照してください。

追加情報については、[483 ページの「patchadd -R を使って代替 root パスを作成する際の制限」](#) を参照してください。

パッケージ追加時のユーザーの対話操作を省略する (pkgadd)

この節では、pkgadd コマンドを使用してパッケージを追加する際にユーザーの対話操作を省略する方法について説明します。

管理ファイルの使用

pkgadd -a コマンドを使用した場合、特殊な管理ファイルからインストールの処理方法に関する情報が取得されます。通常、pkgadd コマンドはいくつかのチェックを行い、指定されたパッケージを実際に追加する前に、プロンプトを表示してユーザーに確認します。ただし、管理ファイルを作成すれば、このようなチェックを省略して、ユーザーの確認なしでパッケージをインストールするように pkgadd コマンドに指示できます。

デフォルトでは、pkgadd コマンドは現在の作業用ディレクトリに管理ファイルがないか調べます。現在の作業用ディレクトリの中に管理ファイルを見つけることができなかった場合、pkgadd コマンドは /var/sadm/install/admin ディレクトリに指定の管理ファイルがないか調べます。また、pkgadd コマンドには管理ファイルへの絶対パスも使用できます。

注- 管理ファイルは注意して使用してください。管理ファイルを使用して、pkgadd が通常実行するチェックとプロンプトを省略する場合は、事前にパッケージのファイルがインストールされている場所や、パッケージのインストール用スクリプトがどのように実行されるのかを知っておく必要があります。

次の例は、pkgadd コマンドがパッケージのインストール前にユーザーに確認のプロンプトを表示しないようにするための管理ファイルを示しています。

```
mail=
instance=overwrite
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
networktimeout=60
networkretries=3
authentication=quit
```

```
keystore=/var/sadm/security  
proxy=  
basedir=default
```

管理ファイルは、パッケージ追加時にユーザーの対話操作を省略する以外にも、さまざまな用途で使用できます。たとえば、管理ファイルを使用すれば、エラーが発生した場合に(ユーザーの対話操作なしに)パッケージのインストールを終了したり、`pkgrm` コマンドでパッケージを削除する場合に対話操作を省略できます。

また、特別なインストールディレクトリをパッケージに割り当てることもできます。この方法は、1つのシステム上で複数のバージョンのパッケージを管理する場合に役に立ちます。これを行うには、`basedir` キーワードを使って管理ファイル内に代替ベースディレクトリを設定します。このキーワードは、パッケージがインストールされる場所を示します。詳細は、[admin\(4\)](#) のマニュアルページを参照してください。

応答ファイルの使用 (pkgadd)

応答ファイルには、「対話型パッケージ」で尋ねられる特定の質問に対するユーザーの応答が格納されます。対話型パッケージには、パッケージをインストールする前にいくつかの質問(たとえば、パッケージのオプションをインストールするかどうかなど)をユーザーに尋ねる `request` スクリプトが格納されています。

インストールするパッケージが対話型パッケージであることがインストール前にわかっていて、今後インストールする際にユーザーの対話操作を省略できるように応答を格納しておきたい場合は、`pkgask` コマンドを使ってユーザーの応答を保存します。このコマンドの詳細は、[pkgask\(1M\)](#) のマニュアルページを参照してください。

`request` スクリプトが尋ねる質問への応答を格納した後は、`pkgadd -r` コマンドを使用して、ユーザーの対話操作なしにパッケージをインストールすることができます。

Solaris システム管理ツールによるソフトウェアの管理 (手順)

この章では、Solaris インストール GUI (Graphical User Interface) と Solaris Product Registry を使ってソフトウェアパッケージを追加、確認、および削除する方法について説明します。

このリリースで新しく追加されたソフトウェア管理機能についての詳細は、[406 ページの「Solaris オペレーティングシステムでのソフトウェア管理における新機能」](#)を参照してください。

ソフトウェアの管理作業に関連する手順については、次の節を参照してください。

- [422 ページの「Solaris インストール GUI によるソフトウェアの追加」](#)
- [423 ページの「Solaris Product Registry の GUI によるソフトウェアの管理 \(作業マップ\)」](#)
- [429 ページの「Solaris Product Registry のコマンド行インタフェースによるソフトウェアの管理 \(作業マップ\)」](#)

Solaris Product Registry および Solaris インストール GUI ツール(ソフトウェア管理用)

次の表は、ソフトウェアパッケージインストールを追加、削除、および確認する際に使用するコマンド (Solaris インストール GUI および Solaris Package Registry ツール) の一覧です。

表 20-1 ソフトウェアパッケージ管理用のシステム管理ツール

ツール	説明	マニュアルページ
installer	インストーラを使用してソフトウェアパッケージをインストールまたは削除します	installer(1M)

表 20-1 ソフトウェアパッケージ管理用のシステム管理ツール (続き)

ツール	説明	マニュアルページ
prodreg	Solaris Product Registry 内のソフトウェアのブラウズ、登録解除、およびアンインストールを可能にします	prodreg(1M)

Solaris インストール GUI によるソフトウェアの追加

この節では、Solaris オペレーティングシステム (Solaris OS) がすでにインストールされているシステムに、Solaris インストール GUI を使ってソフトウェアを追加する方法を説明します。Solaris インストール GUI では、Solaris OS を最初にインストールしたときにインストール対象から外したソフトウェアグループのコンポーネントだけをインストールします。OS のインストールまたはアップグレード後に別のソフトウェアグループにアップグレードすることはできません。

▼ Solaris インストール GUI プログラムを使ってソフトウェアをインストールする方法

注- この手順では、システムがボリューム管理 (vold) を実行しているものとし
ます。システムがボリューム管理を実行していない場合は、『Solaris のシステム管理 (デバイスとファイルシステム)』の第 3 章「リムーバブルメディアへのアクセス (手順)」を参照してください。この章では、ボリューム管理を使用しないでリムーバブルメディアにアクセスする方法について説明しています。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 2 CD、DVD、ネットワークのいずれからソフトウェアをインストールするかを決めます。
次のいずれかの手順に従います。
 - CD からインストールする場合は、CD-ROM ドライブに CD を挿入します。
Solaris 10 LANGUAGES CD を挿入すると、Solaris インストール GUI が自動的に起動します。手順 5 に進みます。
 - DVD からインストールする場合は、DVD-ROM ドライブに DVD を挿入します。
 - ネットワークからインストールする場合は、インストールするソフトウェアのネットイメージを見つけます。

- 3 **Solaris インストール GUI** インストーラが見つかるまで、ディレクトリを変更します。
Solaris インストール GUI インストーラは、CD や DVD 上のさまざまなディレクトリにあります。
 - Solaris 10 SOFTWARE CD または DVD。
 - Solaris 10 Documentation DVD。
 - Solaris 10 LANGUAGES CD。この CD を挿入すると、Solaris インストール GUI が自動的に起動されます。
- 4 指示に従って、ソフトウェアをインストールします。
 - コマンド行から次のコマンドを入力します。

```
% ./installer [options]
```

 - nodisplay GUI を使用しないでインストーラを実行します。
 - noconsole 対話式のテキストコンソールデバイスを使用しないで実行します。ソフトウェアをインストールするための UNIX スクリプトに `installer` コマンドを組み込んだときは、このオプションを `-nodisplay` オプションと合わせて使用します。
 - ファイルマネージャから、「インストール (Installer)」をダブルクリックします。
「インストール (Installer)」ウィンドウが表示され、続いて「Solaris installation GUI」ダイアログボックスが表示されます。
- 5 画面の指示に従ってソフトウェアをインストールします。
- 6 ソフトウェアの追加が終わったら、「終了 (Exit)」をクリックします。
Solaris インストール GUI が終了します。

Solaris Product Registry の GUI によるソフトウェアの管理 (作業マップ)

次の作業マップに、Solaris Product Registry を使って行うことができるソフトウェアの管理作業を示します。

作業	説明	参照先
インストールまたはアンインストールしたソフトウェアを Solaris Product Registry を使って表示します。	インストールまたはアンインストールしたソフトウェアについて調べたい場合に使用します。	426 ページの「インストールまたはアンインストールしたソフトウェアの情報を Solaris Product Registry の GUI を使って表示する方法」
Solaris Product Registry を使ってソフトウェアをインストールします。	Solaris Product Registry を使ってソフトウェアを検索し、Solaris インストール GUI を起動できます。このプログラムの指示に従ってソフトウェアのインストールを行います。	427 ページの「Solaris Product Registry の GUI を使ってソフトウェアをインストールする方法」
Solaris Product Registry を使ってソフトウェアをアンインストールします。	Solaris Product Registry を使ってソフトウェアをアンインストールする場合に使用します。	428 ページの「Solaris Product Registry の GUI を使ってソフトウェアをアンインストールする方法」

Solaris Product Registry は、インストールされているソフトウェアをユーザーが管理するためのツールです。ソフトウェアのインストールが終わると、Solaris Product Registry は、Solaris インストール GUI または Solaris pkgadd コマンドを使用して、インストールされたすべてのソフトウェアの一覧を表示します。

Solaris Product Registry は、GUI またはコマンド行インタフェース (CLI) で使用できます。Solaris Product Registry CLI の使用方法の詳細については、[429 ページの「Solaris Product Registry のコマンド行インタフェースによるソフトウェアの管理 \(作業マップ\)」](#)を参照してください。

Solaris Product Registry の GUI インタフェースでは、次の作業を実行できます。

- インストールおよび登録されているソフトウェアとソフトウェア属性の一覧表示
- System Software Localizations ディレクトリにインストールされている地域対応版のすべての Solaris システム製品の表示。
- インストーラの選択と起動
- 追加のソフトウェア製品のインストール
- ソフトウェアや個々のソフトウェアパッケージのアンインストール。

Solaris Product Registry の GUI のメインウィンドウは、次の 3 つの情報区画から構成されています。

- インストール、登録、削除されたソフトウェア
- 選択されているソフトウェアの標準属性
- カスタマイズされた属性と登録されているソフトウェアの内部属性

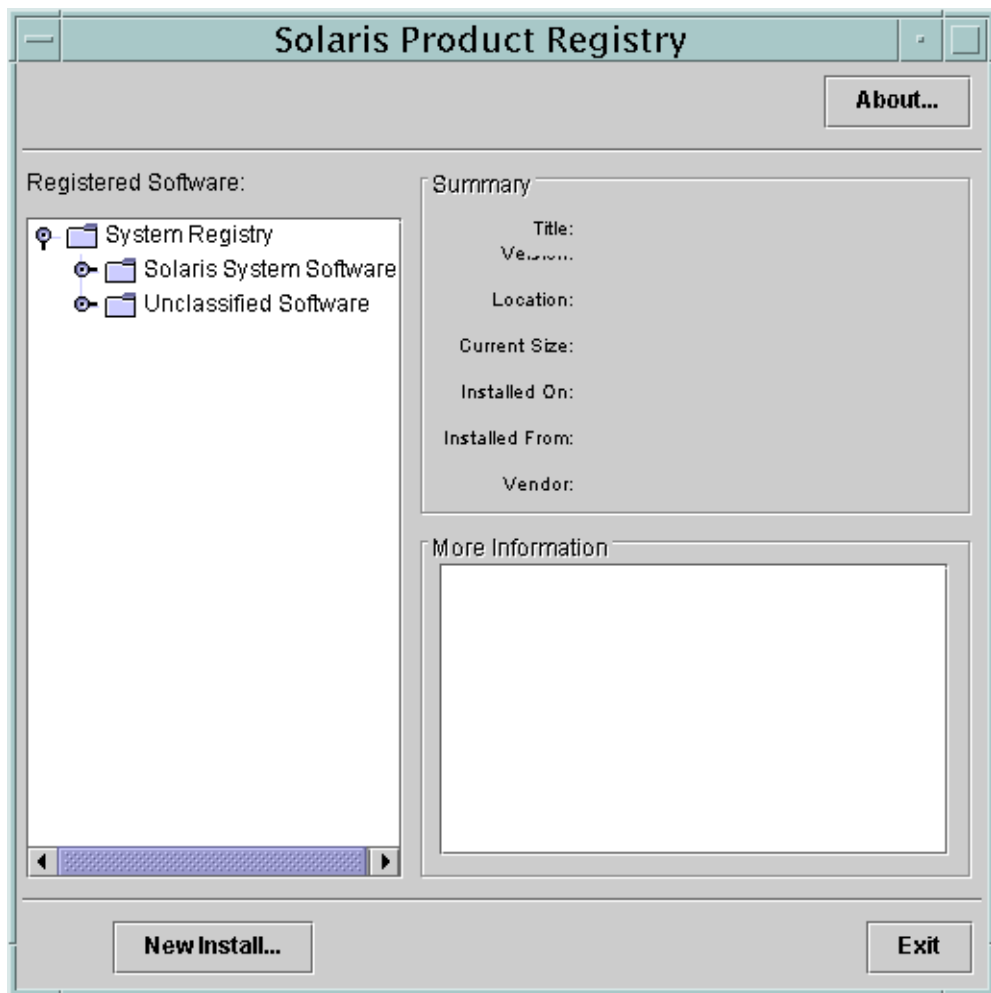


図 20-1 「Solaris Product Registry」メインウィンドウ

▼ インストールまたはアンインストールしたソフトウェアの情報を **Solaris Product Registry** の **GUI** を使って表示する方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。
- 2 **Solaris Product Registry** ツールを起動します。

```
# prodreg &
```

「Solaris Product Registry」メインウィンドウが表示されます。
- 3 「登録されているソフトウェア (Registered Software)」ボックスの「システムレジストリ (System Registry)」ディレクトリの左側にあるアイコンをクリックします。
右側を指していたアイコンが下側が変わります。レジストリの項目は、左側にテキストファイルアイコンがあるものを除き、どの項目も展開または縮小できます。
「登録されているソフトウェア (Registered Software)」ボックスにインストールされているソフトウェアには、必ず次のコンポーネントが含まれます。
 - Solaris のインストール時に選択した構成ソフトウェアグループ。ソフトウェアグループは、「簡易ネットワークサポート (Reduced Network Support)」、「コアシステムサポート (Core System Support)」、「エンドユーザーシステムサポート (End User System Support)」、「開発者システムサポート (Developer System Support)」、「全体ディストリビューション (Entire Distribution)」、「全体ディストリビューションと OEM サポート (Entire Distribution plus OEM Support)」のいずれかです。
 - 追加のシステムソフトウェア (選択したソフトウェアグループに含まれていない Solaris 製品を含む)。
 - 分類されていないソフトウェア (非 Solaris 製品またはソフトウェアグループに含まれていない製品)。このソフトウェアには、`pkgadd` コマンドを使ってインストールしたすべてのパッケージが含まれます。
- 4 表示するソフトウェアアプリケーションが見つかるまでディレクトリを選択します。
ディレクトリを開くと、リストが展開されます。
- 5 属性を表示する場合は、ディレクトリまたはファイルを選択します。
属性情報が「一覧 (Summary)」ボックスと「詳細情報 (more information)」ボックスに表示されます。

- Solaris インストール GUI でインストールしたソフトウェア製品の場合、Solaris Product Registry には次の値が必ず表示されます。「タイトル (Title)」、「バージョン (Version)」、「インストール場所 (Location)」、および「インストール日時 (Installed on)」。製品またはソフトウェアグループの下で展開されているリスト内の項目の場合、その製品のバージョン情報が表示されます。
- pkgrm コマンドを使って製品のすべてまたは一部を削除した場合は、ソフトウェア製品の名前の横にそれを警告するアイコンが表示されます。

▼ Solaris Product Registry の GUI を使ってソフトウェアをインストールする方法

Solaris Product Registry を使用すると、ソフトウェアを検索し、Solaris インストール GUI を起動できます。このプログラムの指示に従ってソフトウェアのインストールを行います。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、[『Solaris のシステム管理 \(セキュリティサービス\)』の「RBAC の構成 \(作業マップ\)」](#)を参照してください。
- 2 **Solaris Product Registry** ツールを起動します。

```
# prodreg
```


「Solaris Product Registry」メインウィンドウが表示されます。
- 3 インストールを **CD**、**DVD**、またはネットワークのいずれから行うかを決めます。次のいずれかの手順に従います。
 - **CD** からインストールする場合は、**CD-ROM** ドライブに **CD** を挿入します。
 - **DVD** からインストールする場合は、**DVD-ROM** ドライブに **DVD** を挿入します。
 - ネットワークからインストールする場合は、インストールするソフトウェアのネットイメージを見つけます。
- 4 インストールおよび登録されているソフトウェアの一覧を表示するには、その左側にあるアイコンをクリックします。

- 5 「**Solaris Product Registry**」ウィンドウの下にある「新規インストール(**New Install**)」ボタンをクリックします。
「Select Installer」ダイアログボックスが表示され、最初は /cdrom ディレクトリまたは現在使用中のディレクトリが選択されています。
- 6 **Solaris** インストール GUI インストーラが見つかるまで、ディレクトリを選択します。
Solaris インストール GUI インストーラは、CD や DVD 上のさまざまなディレクトリにあります。
 - Solaris 10 SOFTWARE CD または DVD。
 - Solaris 10 Documentation DVD。
 - Solaris 10 LANGUAGES CD。この CD を挿入すると、Solaris インストール GUI が自動的に起動されます。
- 7 起動するインストーラを見つけたら、「ファイル(**Files**)」ボックスでそのインストーラの名前を選択します。
- 8 「了解(**OK**)」をクリックします。
選択したインストーラが起動されます。
- 9 インストーラによって表示される手順に従って、ソフトウェアをインストールします。

▼ **Solaris Product Registry** の GUI を使ってソフトウェアをアンインストールする方法

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。
- 2 **Solaris Product Registry** ツールを起動します。

```
# prodreg
```


「Solaris Product Registry」メインウィンドウが表示されます。
- 3 インストールおよび登録されているソフトウェアの一覧を表示するには、その左側にあるアイコンをクリックします。
- 4 アンインストールするソフトウェアの名前が見つかるまでディレクトリを選択します。

- 5 ソフトウェア属性を読んで、このソフトウェアがアンインストールするソフトウェアであることを確認します。
- 6 「Solaris Product Registry」ウィンドウの下にある「アンインストール(Uninstall)」ボタンをクリックします。
選択したソフトウェアがアンインストールされます。

Solaris Product Registry のコマンド行インタフェースによるソフトウェアの管理 (作業マップ)

次の作業マップに、Solaris Product Registry のコマンド行インタフェースを使って行うことができるソフトウェアの管理作業を示します。

作業	説明	参照先
インストールまたはアンインストールしたソフトウェアを表示します。	<code>browse</code> サブコマンドを使ってソフトウェア情報を表示できます。	430 ページの「インストールまたはアンインストールしたソフトウェアの情報を表示する方法 (prodreg)」
ソフトウェア属性を表示します。	<code>info</code> サブコマンドを使って特定のソフトウェア属性を表示できます。	433 ページの「ソフトウェア属性を表示する方法 (prodreg)」
ソフトウェアコンポーネント間の依存関係を確認します。	<code>info</code> サブコマンドを使って特定のソフトウェアコンポーネントに依存するコンポーネントを表示できます。	436 ページの「ソフトウェアの依存関係を確認する方法 (prodreg)」
損傷を受けたソフトウェア製品を特定します。	適切なアンインストーラを使用しないでインストール済みのソフトウェアファイルやパッケージを削除すると、システム上のソフトウェアが損傷を受ける可能性があります。	437 ページの「損傷を受けたソフトウェア製品を特定する方法 (prodreg)」
ソフトウェアをアンインストールします。	<code>uninstall</code> サブコマンドを使ってソフトウェアをシステムから削除できます。	440 ページの「ソフトウェアをアンインストールする方法 (prodreg)」
損傷を受けたソフトウェアをアンインストールします。	ソフトウェアコンポーネントのアンインストーラをシステムから削除した場合は、損傷を受けたコンポーネントのアンインストールが失敗することがあります。	444 ページの「損傷を受けたソフトウェアをアンインストールする方法 (prodreg)」

作業	説明	参照先
損傷を受けたソフトウェアコンポーネントを再インストールします。	損傷を受けたソフトウェアコンポーネントに依存するソフトウェアがある場合は、損傷を受けたコンポーネントと依存ソフトウェアをアンインストールするのではなく、損傷を受けたコンポーネントを再インストールできます。	448 ページの「損傷を受けたソフトウェアコンポーネントを再インストールする方法 (prodreg)」

Solaris Product Registry のコマンド行インタフェースによるソフトウェアの管理

prodreg コマンドは、Solaris Product Registry のコマンド行インタフェース (CLI) です。prodreg コマンドは、システム上のソフトウェアを管理するための複数のサブコマンドをサポートしています。

端末ウィンドウで prodreg コマンドを実行して、次の作業を行うことができます。

- インストールおよび登録されているソフトウェアとソフトウェア属性の一覧表示。
- System Software Localizations ディレクトリにインストールされている地域対応版のすべての Solaris システム製品の表示。
- 損傷を受けたソフトウェアの特定。
- Solaris Product Registry のソフトウェアエントリの削除。
- ソフトウェアや個々のソフトウェアパッケージのアンインストール。

コマンド行インタフェース経由で Solaris Product Registry を管理する方法については、[prodreg\(1M\)](#) のマニュアルページを参照してください。

▼ インストールまたはアンインストールしたソフトウェアの情報を表示する方法 (prodreg)

Solaris Product Registry 内のソフトウェアに関する情報を確認するには、端末ウィンドウで prodreg コマンドのサブコマンド browse を実行します。

- 1 端末ウィンドウを開きます。
- 2 **Solaris Product Registry** をブラウズします。

% **prodreg browse**

BROWSE # +/-/. UUID

NAME

=====	=====	=====	=	=====
1	-	root	1	System Registry
2	+	a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b	1	Solaris 10 System Software
3	+	8f64eabf-1dd2-11b2-a3f1-0800209a5b6b	1	Unclassified Software

prodreg のサブコマンド browse では、登録済みソフトウェアに関する次の情報を表示できます。

BROWSE # prodreg browse コマンドを実行すると、Solaris Product Registry により、登録済みの各ソフトウェアコンポーネントの「ブラウズ番号」が生成されます。この番号を prodreg browse コマンドや prodreg info コマンドの引数に指定することにより、登録済みの特定のコンポーネントの下位コンポーネントを表示できます。

注-ブラウズ番号 (BROWSE#) は、システムのリブートや再インストールによって変更される可能性があります。ブラウズ番号をスクリプト内に記述したり、その他のログインセッションで再利用したりすることはできません。

+/-/. このフィールドは、ソフトウェアコンポーネントが Solaris Product Registry に登録された子コンポーネントを持っているかどうかを示します。

このフィールドには、次の文字が表示されます。

- +-ソフトウェアコンポーネントは、現在表示されていないが子コンポーネントを持っていることを示します。
- --ソフトウェアコンポーネントは、現在表示されている子コンポーネントを持っていることを示します。
- .-ソフトウェアコンポーネントが子コンポーネントを持っていないことを示します。

UUID このフィールドには、Solaris Product Registry に格納されているソフトウェア固有の識別子が表示されます。

このフィールドには、システム上のソフトウェアコンポーネントのインスタンス番号が表示されます。システム上に複数のソフトウェアコンポーネントインスタンスがある場合、Solaris Product Registry はそれぞれのインスタンスに別々のインスタンス番号を割り当てます。

NAME このフィールドには、ソフトウェアの地域対応化された名前が表示されます。この出力例では、Solaris OS の名前は「Solaris 10 System

3 Solaris Product Registry 内の任意のソフトウェアコンポーネント 1 個の情報をブラウズします。

-m “name” コマンドは、*name* を名前を持つソフトウェアコンポーネントの情報を表示します。

```
% prodreg browse -u name-UUID -i
instance -n number
```

一意の識別子 *name-UUID* で指定された *name* ソフトウェアコンポーネントの情報を表示します。

インスタンス番号 *instance* で指定された *name* ソフトウェアコンポーネントの情報を表示します。

コンポーネントのブラウザ番号 *number* を参照することでソフトウェアの情報を表示します。

例 20-1 コンポーネント名によるソフトウェア情報の表示 (prodreg)

BROWSE #	+/-/.	UUID	#	NAME
1	-	root	1	System Registry
2	+	a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b	1	Solaris 10 System Software
3	+	8f64eabf-1dd2-11b2-a3f1-0800209a5b6b	1	Unclassified Software

```
% prodreg browse -m "Solaris 10 System Software"
```


例 20-2 コンポーネントのブラウザ番号によるソフトウェア情報の表示 (prodreg)

次の例では、`prodreg browse` コマンドの `-n` オプションでコンポーネントのブラウザ番号を指定して、ソフトウェアの情報を表示します。

```
% prodreg browse
  BROWSE # +/-/.  UUID                                     #  NAME
  =====
  1      -      root                                     1  System
                                           Registry
  2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                           System
                                           Software
  3      +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                           Software

% prodreg browse -n 2
```

例 20-3 コンポーネントの UUID によるソフトウェア情報の表示 (prodreg)

次の例では、`prodreg browse` コマンドの `-u` オプションでコンポーネントの UUID を指定して、ソフトウェアの情報を表示します。UUID は、Solaris Product Registry におけるソフトウェアの一意の識別子です。

```
% prodreg browse
  BROWSE # +/-/.  UUID                                     #  NAME
  =====
  1      -      root                                     1  System
                                           Registry
  2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                           System
                                           Software
  3      +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                           Software

% prodreg browse -u a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b
```

▼ ソフトウェア属性を表示する方法 (prodreg)

`prodreg info` コマンドを使って、特定のソフトウェア属性を表示できます。

prodreg info コマンドは、登録済みソフトウェアに関するさまざまな情報を表示します。次に例を挙げます。

- ソフトウェアコンポーネントの名前
- ソフトウェアコンポーネントの説明
- ソフトウェアの必須コンポーネント
- このソフトウェアを必要とするその他のコンポーネント
- ソフトウェアのベースディレクトリ
- ソフトウェアコンポーネントのパス

- 1 端末ウィンドウを開きます。
- 2 Solaris Product Registry をブラウズします。

```
% prodreg browse
BROWSE # +/-/.  UUID                                     #  NAME
=====
1      -      root                                     1  System
                                           Registry
2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                           System
                                           Software
3      +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                           Software
```

- 3 表示されたソフトウェアコンポーネントのいずれかの属性を表示します。

```
% prodreg info -m "name"

-m "name" コマンドは、name を名前に持つソフトウェアコンポーネントの属性を表示します。
```

- 4 表示するソフトウェアコンポーネントごとに、手順3を繰り返します。

例 20-4 コンポーネント名によるソフトウェア属性の表示 (prodreg)

次の例では、コンポーネント名を指定してソフトウェア属性を表示します。

```
% prodreg browse
BROWSE # +/-/.  UUID                                     #  NAME
=====
1      -      root                                     1  System
                                           Registry
2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                           System
                                           Software
3      +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                           Software
```

```
% prodreg info -m "Solaris 10 System Software"
```

例 20-5 コンポーネントのブラウズ番号によるソフトウェア属性の表示 (prodreg)

次の例では、`prodreg info` コマンドの `-n` オプションでコンポーネントのブラウズ番号を指定して、ソフトウェアの属性を表示します。

```
% prodreg browse
  BROWSE # +/-/.  UUID                                     #  NAME
  =====
  1      -      root                                     1  System
                                           Registry
  2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                           System
                                           Software
  3      +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                           Software

% prodreg info -n 2
```

例 20-6 コンポーネントの UUID によるソフトウェア属性の表示 (prodreg)

次の例では、`prodreg info` コマンドの `-u` オプションでコンポーネントの UUID を指定して、ソフトウェア属性を表示します。UUID は、Solaris Product Registry におけるソフトウェアの一意の識別子です。

```
% prodreg browse
  BROWSE # +/-/.  UUID                                     #  NAME
  =====
  1      -      root                                     1  System
                                           Registry
  2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                           System
                                           Software
  3      +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                           Software

% prodreg info -u a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b
```

▼ ソフトウェアの依存関係を確認する方法 (prodreg)

特定のソフトウェアコンポーネントに依存するコンポーネントを表示するには、`prodreg info` コマンドを使用します。特定のコンポーネントをアンインストールする前に、ソフトウェア製品間の依存関係を確認できます。

- 1 端末ウィンドウを開きます。
- 2 **Solaris Product Registry** をブラウズします。

```
% prodreg browse
BROWSE # +/-/.  UUID                                     #  NAME
=====
1          -      root                                     1  System
                                           Registry
2          +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                           System
                                           Software
3          +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                           Software
```

確認するソフトウェアコンポーネントが表示されるまで、`prodreg browse` コマンドを繰り返します。`prodreg browse` コマンドを使って Solaris Product Registry をブラウズする方法の詳細については、[430 ページの「インストールまたはアンインストールしたソフトウェアの情報を表示する方法 \(prodreg\)」](#)を参照してください。

- 3 特定のソフトウェアコンポーネントの依存関係を表示します。

```
% prodreg info -m "name" -a "Dependent Components"

-m "name"                                name で名前を指定されたソフトウェアコン
                                           ポーネントの属性を表示します。

-a "Dependent Components"                Dependent Components 属性の値を表示する
                                           ことにより、name ソフトウェアに依存する
                                           コンポーネントを表示します。
```

このコマンドの出力には、*name* ソフトウェアに依存するソフトウェアコンポーネントの一覧が表示されます。

例 20-7 その他のソフトウェア製品に依存するコンポーネントの表示 (prodreg)

次の例では、ExampleSoft というソフトウェア製品に依存するコンポーネントを一覧表示します。

```
% prodreg -m "ExampleSoft" -a "Dependent Components"
Dependent Components:
Name                                     UUID                                     #
```

ExampleSoftA7f49ecvb-1ii2-11b2-a3f1-0800119u7e8e1

▼ 損傷を受けたソフトウェア製品を特定する方法 (prodreg)

適切なアンインストーラを使用しないでインストール済みのソフトウェアファイルやパッケージを削除すると、システム上のソフトウェアが損傷を受ける可能性があります。損傷を受けたソフトウェアは、正しく機能しない場合があります。prodreg info コマンドを使って、ソフトウェア製品が損傷を受けていないかどうかを確認できます。

1 確認するソフトウェアの Solaris Product Registry 情報を表示します。

```
% prodreg browse -m name
BROWSE # +/-/. UUID # NAME
=====
1 - root 1 System Registry
2 + a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b 1 Solaris 10 System Software
3 + 8f64eabf-1dd2-11b2-a3f1-0800209a5b6b 1 Unclassified Software
4 - name-UUID 1
name
233 . component-a-pkg 1
component-a
234 . component-b-pkg 1
-m "name"
```

name で名前を指定されたソフトウェアコンポーネントの情報を表示します。

name-UUID *name* ソフトウェアコンポーネントの UUID を指定します。

component-a-pkg *name* ソフトウェアに依存する *component-a* コンポーネントのパッケージ名を指定します。

component-a *name* ソフトウェアに依存するコンポーネントの名前を指定します。

component-b-pkg *name* ソフトウェアに依存する *component-b* コンポーネントのパッケージ名を指定します。

前出の出力例では、Name フィールドに *component-b-pkg* エントリの名前が表示されていません。Solaris Product Registry にソフトウェアコンポーネント名が表示されない場合、そのコンポーネントは損傷を受けている可能性があります。

- 2 ソフトウェアコンポーネントが損傷を受けていないかどうかを検証します。

```
% prodreg info -u name-UUID -i 1 -d
```

```
isDamaged=TRUE
```

```
-u name-UUID
```

name ソフトウェアコンポーネントの情報を表示します。

```
-i 1
```

name ソフトウェアコンポーネントの 1 番目のインスタンスの情報を表示します。

```
-d
```

name ソフトウェアコンポーネントの *isDamaged* 属性の値を表示します。

isDamaged=TRUE と出力された場合、*name* ソフトウェアコンポーネントは損傷を受けています。

- 3 *name-UUID* ソフトウェアコンポーネントを構成するパッケージを特定します。

```
% prodreg info -u name-UUID -i 1 -a PKGS
```

```
pkgs:
```

```
component-a-pkg component-b-pkg
```

- 4 これらのパッケージがシステムにインストールされていることを確認します。

```
% pkginfo component-a-pkg
```

```
application component-a-pkg component-a
```

```
% pkginfo component-b-pkg
```

```
ERROR: information on "component-b-pkg" was not found
```

pkginfo component-b-pkg コマンドのエラーメッセージは、*component-b-pkg* パッケージがシステムから削除されたことを示しています。*name* ソフトウェアコンポーネントは、*component-b-pkg* パッケージがないと正しく機能しない可能性があります。

例 20-8 損傷を受けたソフトウェアコンポーネントの特定 (prodreg)

次の例では、ExampleSoft ソフトウェアが損傷を受けていないかどうかを確認します。

```
% prodreg browse -m Examplesoft
```

```
BROWSE # +/-. UUID
```

```
# NAME
```

=====	=====	=====	=	=====
1	-	root	1	System Registry
2	+	a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b	1	Solaris 10 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	.	EXS0zzt	1	
235	.	EXS0blob	1	Example Data

NAME フィールドに、ExampleSoft の子コンポーネント EXS0zzt のエントリがありません。ExampleSoft ソフトウェアは損傷を受けている可能性があります。prodreg info コマンドに -u、-i、および -d オプションを指定して、ExampleSoft ソフトウェアが損傷を受けているかどうかを確認します。

```
% prodreg info -u 95842091-725a-8501-ef29-0472985982be -i 1 -d
```

```
isDamaged=TRUE
```

isDamaged=TRUE という出力結果から、ExampleSoft ソフトウェアが損傷を受けていることがわかります。prodreg info コマンドの -a PKGS オプションを実行して、ExampleSoft ソフトウェアパッケージを特定します。

```
% prodreg info
  -u 95842091-725a-8501-ef29-0472985982be
  -i 1 -a PKGS
```

```
pkgs:
EXS0zzt EXS0blob
```

パッケージ EXS0zzt と EXS0blob がシステムにインストールされていることを確認するために、pkginfo コマンドを使用します。

```
% pkginfo EXS0zzt
ERROR: information for "EXS0zzt" was not found
```

```
% pkginfo EXS0blob
application EXS0blob      Example Data
```

pkginfo コマンドの出力結果から、EXS0zzt パッケージがシステムにインストールされていないことがわかります。したがって、ExampleSoft ソフトウェアは損傷を受けています。

▼ ソフトウェアをアンインストールする方法 (prodreg)

prodreg uninstall コマンドを使って、システムからソフトウェアを削除できます。prodreg uninstall コマンドを使ってソフトウェアをアンインストールすると、指定のソフトウェアとそのソフトウェアに関連付けられている子コンポーネントがすべて削除されます。アンインストールするソフトウェアに依存するソフトウェアがないことを確認してから、ソフトウェアを削除してください。[436 ページの「ソフトウェアの依存関係を確認する方法 \(prodreg\)」](#)を参照してください。

ソフトウェアのアンインストールが完了したら、prodreg unregister -r コマンドを使って、ソフトウェアそのものと、その子コンポーネント全部を Solaris Product Registry から削除できます。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、[『Solaris のシステム管理 \(セキュリティサービス\)』の「RBAC の構成 \(作業マップ\)」](#)を参照してください。
- 2 アンインストールするソフトウェアの情報を表示します。

```
# prodreg browse -u name-UUID
BROWSE #  +/-./.  UUID                                     #  NAME
=====  =====
1          -      root                                     1  System
Registry
2          +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
System
Software
3          +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
Software
1423       -      name-UUID                                1
name
1436       .      component-a-UUID                          1
component-a
1437       -      component-b-UUID                          1
component-b
1462       .      component-c-UUID                          1
component-c
-u name-UUID
```

一意の識別子 *name-UUID* で指定されたソフトウェアコンポーネントの情報を表示します。

<i>name</i>	一意の識別子 <i>name-UUID</i> でアンインストールするソフトウェアコンポーネントの名前を指定します。
<i>.component-a-UUID</i>	<i>name</i> ソフトウェアによって必要とされているソフトウェアコンポーネント <i>component-a</i> の一意の識別子を指定します。
<i>component-a</i>	<i>name</i> ソフトウェアによって必要とされているコンポーネントの名前を指定します。
<i>- component-b-UUID</i>	<i>name</i> ソフトウェアによって必要とされているソフトウェアコンポーネント <i>component-b</i> の一意の識別子を指定します。ハイフン(-)は、 <i>component-b</i> がその他のソフトウェアコンポーネントを必要としていることを表します。
<i>component-b</i>	<i>name</i> ソフトウェアによって必要とされているソフトウェアコンポーネントの名前を指定します。
<i>.component-c-UUID</i>	<i>component-b</i> ソフトウェアによって必要とされているソフトウェアコンポーネント <i>component-c</i> の一意の識別子を指定します。
<i>component-c</i>	<i>component-b</i> ソフトウェアによって必要とされているソフトウェアコンポーネントの名前を指定します。

- 3 ソフトウェアをアンインストールします。
- 4 アンインストールするソフトウェアの依存関係を確認します。

```
# prodreg uninstall -u name-UUID

# prodreg info -u name-UUID
Title: name
.
.
.
Child Components:
Name                               UUID                               #
-----
component-a                        component-a-UUID
1
component-b                        component-b-UUID
1
```

Required Components:		
Name	UUID	#

<i>component-a</i>	<i>component-a-UUID</i>	
	1	
<i>component-b</i>	<i>component-b-UUID</i>	
	1	

prodreg info コマンドによって出力された次の情報を確認します。

- 「Child Components」 - *name* ソフトウェアコンポーネントに関連付けられたソフトウェアコンポーネントを一覧表示します。 *name* ソフトウェアの登録を解除すると、その子コンポーネントの登録も解除されます。前出の `prodreg info` コマンドでは、すべての子コンポーネントが一覧表示されます。これらの子コンポーネントの登録を解除してもよいかどうか確認してください。
- 「Required Components」 - *name* ソフトウェアによって必要とされているソフトウェアコンポーネントを一覧表示します。ソフトウェアコンポーネントは、子コンポーネント以外のコンポーネントを必要とする場合があります。コンポーネントとともにアンインストールされ、登録を解除されるのは、子コンポーネントだけです。
- 「Dependent Components」 - *name* ソフトウェアがないと実行できないコンポーネントを一覧表示します。 *name* ソフトウェアの登録を解除すると、その依存コンポーネントの登録も解除されます。 `prodreg info` コマンドの出力結果に依存コンポーネントが含まれていた場合、それらの依存コンポーネントの登録を解除してもよいかどうか確認してください。

前出の出力例では、 *name* ソフトウェアに依存コンポーネントはありません。

5 *name* ソフトウェアの子コンポーネントの依存関係を確認します。

`prodreg info -u component-a-UUID -i 1 -a "Dependent Components"`

Dependent Components:		
Name	UUID	#

<i>name</i>	<i>name-UUID</i>	1

`prodreg info -u component-b-UUID -i 1 -a "Dependent Components"`

Dependent Components:		
Name	UUID	#

<i>name</i>	<i>name-UUID</i>	1

`prodreg info -u component-c-UUID -i 1 -a "Dependent Components"`

Dependent Components:

Name	UUID	#
-----	-----	-
component-b	component-b-UUID	
	1	

この出力例では、*name* ソフトウェアの子コンポーネントに依存するソフトウェアはありません。

6 ソフトウェアコンポーネントとその子コンポーネントの登録を解除します。

```
# prodreg unregister -r -u name-UUID -i 1
```

-r	一意の識別子 <i>name-UUID</i> を指定して、ソフトウェアとその子コンポーネント全部の登録を再帰的に解除します。
-u name-UUID	登録を解除するソフトウェアを表す一意の識別子を指定します。
-i 1	登録を解除するソフトウェアのインスタンスを指定します。

例 20-9 ソフトウェアコンポーネントのアンインストール (prodreg)

次の例では、ExampleSoft ソフトウェアとその子コンポーネントすべてをアンインストールします。

```
# prodreg browse -m "ExampleSoft"
```

BROWSE #	+/-/.	UUID	#	NAME
=====	=====	=====	=	=====
1	-	root	1	System Registry
2	+	a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b	1	Solaris 10 System Software
3	+	8f64eabf-1dd2-11b2-a3f1-0800209a5b6b	1	Unclassified Software
1423	-	95842091-725a-8501-ef29-0472985982be	1	ExampleSoft
1436	.	90209809-9785-b89e-c821-0472985982be	1	Example Doc
1437	-	EXS0zzt	1	Example Data
1462	.	EXS0blob	1	Example Data

```
# prodreg uninstall -u 95842091-725a-8501-ef29-0472985982be -i 1
```

```
# prodreg info -u 95842091-725a-8501-ef29-0472985982be
```

Title: ExampleSoft Software

.

.

```
.
Child Components:
Name                               UUID                               #
-----
Example Doc                        90209809-9785-b89e-c821-0472985982be 1
Example Data                       EXS0zzt                             1

Required Components:
Name                               UUID                               #
-----
Example Doc                        90209809-9785-b89e-c821-0472985982be 1
Example Data                       EXS0zzt                             1

# prodreg info -u 90209809-9785-b89e-c821-0472985982be -i 1
-a "Dependent Components"
Dependent Components:
Name                               UUID                               #
-----
ExampleSoft                        95842091-725a-8501-ef29-0472985982be 1

# prodreg info -u EXS0zzt -i 1 -a "Dependent Components"
Dependent Components:
Name                               UUID                               #
-----
ExampleSoft                        95842091-725a-8501-ef29-0472985982be 1

# prodreg info -u EXS0blob -i 1 -a "Dependent Components"
Dependent Components:
Name                               UUID                               #
-----
Example Data                       EXS0zzt                             1

# prodreg unregister -r -u 95842091-725a-8501-ef29-0472985982be -i 1
```

▼ 損傷を受けたソフトウェアをアンインストールする方法 (prodreg)

損傷を受けたソフトウェアコンポーネントを `prodreg uninstall` コマンドでアンインストールしようとする、失敗します。これは、ソフトウェアコンポーネントのアンインストールプログラムがシステムから削除されている場合に発生する現象です。

システム上にソフトウェアコンポーネントのアンインストールプログラムがない場合は、次の手順に従ってアンインストールを行ってください。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
- 役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 アンインストールするソフトウェアの情報を表示します。

```
# prodreg browse -m "name"
BROWSE # +/-/.  UUID                                     # NAME
===== =====
1      -      root                                     1 System Registry
2      +      a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1 Solaris 10 System Software
3      +      8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1 Unclassified Software
4      -      UUID                                     1
name
1436      .      component-a-UUID                       1
component-a
1437      .      component-b-UUID                       1
```

-m "name"

UUID

.component-a-UUID

component-a

.component-b-UUID

アンインストールする *name* ソフトウェアコンポーネントの 情報を表示します。

アンインストールするソフトウェアコンポーネントの UUID を指定します。

component-a ソフトウェアコンポーネントの UUID を指定します。

name ソフトウェアの子ソフトウェアコンポーネントの名前を指定します。

name ソフトウェアの子ソフトウェアコンポーネントの UUID を指定します。

前出の出力例では、*component-b-UUID* エントリにコンポーネント名が関連付けられていません。このように名前の値がない場合、このコンポーネントは損傷を受けている可能性があります。

- 3 ソフトウェアをアンインストールします。

```
# prodreg uninstall -u UUID -i 1
The install program requested could not be found
```

-u UUID

アンインストールするソフトウェアコンポーネントの UUID を指定します。

-i 1 アンインストールするソフトウェアのインスタンスを指定します。

前出のエラーメッセージは、システム上にアンインストールプログラムが存在しないことを示しています。

4 ソフトウェアコンポーネントのアンインストールプログラムを特定します。

```
# prodreg info -m "name" -a uninstallprogram
```

```
uninstallprogram: /usr/bin/java -mx64m -classpath
uninstaller-location uninstall_name
```

-m “*name*” *name* ソフトウェアコンポーネントの情報を表示します。

<code>-a uninstallprogram</code>	<i>name</i> ソフトウェアコンポーネントに関連付けられたアンインストールプログラムの情報を表示します。

<i>uninstaller-location</i>	<i>name</i> ソフトウェアコンポーネントのアンインストールプログラムの登録場所を指定します。

5 登録場所にアンインストーラが存在するかどうか確認します。

```
# ls uninstaller-location
```

uninstaller-location:

No such file or directory

ls(1) コマンドの出力結果から、登録場所にアンインストールプログラムが存在しないことがわかります。

6 次のいずれかの方法でシステムからソフトウェアを削除します。

- システムのバックアップを使用できる場合は、次の手順に従います。
 - a. バックアップからアンインストールプログラムをロードします。
 - b. 端末ウィンドウなどのシェルコマンド行インタフェースからアンインストールプログラムを実行します。
- バックアップのアンインストールプログラムにアクセスできない場合は、次の手順に従います。
 - a. ソフトウェアコンポーネントの登録を解除します。

```
# prodreg unregister -u UUID -i 1
```

- b. 削除するソフトウェアによって必要とされている登録済みコンポーネントをすべて削除します。

```
# pkgrm component-a-UUID
```

例 20-10 損傷を受けたソフトウェアのアンインストール (prodreg)

次の例では、損傷を受けた ExampleSoft ソフトウェアをアンインストールします。この例では、システムバックアップ上のアンインストールプログラムにはアクセスできないものとします。

```
# prodreg browse -m Examplesoft
BROWSE #  +/-.  UUID                                     #  NAME
=====  =====
1          -    root                                     1  System
                                           Registry
2          +    a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                           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         .    EXS0zzt                                1
235         .    EXS0blob                                1  Example Data

# prodreg uninstall -u 95842091-725a-8501-ef29-0472985982be -i 1
The install program requested could not be found

# prodreg info -m "ExampleSoft" -a uninstallprogram
uninstallprogram: /usr/bin/java -mx64m -classpath
/var/sadm/prod/org.example.ExampleSoft/987573587 uninstall_ExampleSoft

# ls /var/sadm/prod/org.example.ExampleSoft/987573587
/var/sadm/prod/org.example.ExampleSoft/987573587:
No such file or directory

# prodreg unregister -u 95842091-725a-8501-ef29-0472985982be -i 1

# pkgrm EXS0blob
```

損傷を受けたソフトウェアコンポーネントに依存するソフトウェアがある場合は、損傷を受けたコンポーネントと依存ソフトウェアをアンインストールするのではなく、損傷を受けたコンポーネントを再インストールできます。-f オプションを指定して `prodreg unregister` コマンドを実行すれば、損傷を受けたコンポーネントの登録を強制的に解除できます。その後、そのコンポーネントを再インストールできます。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
- 役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。

- 2** 再インストールするソフトウェアの情報を表示します。

#	prodreg	browse	-m "name"	#	NAME
BROWSE	#	+/-/.	UUID		
=====	=====	=====	=====	=	=====
1	-	root		1	System Registry
2	+	a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b		1	Solaris 10 System Software
3	+	8f64eabf-1dd2-11b2-a3f1-0800209a5b6b		1	Unclassified Software
4	.	UUID		1	
name					

-m “name”

再インストールする *name* ソフトウェアコンポーネントの情報を表示します。

UUID

再インストールするソフトウェアコンポーネントの UUID を指定します。

- 3 再インストールするソフトウェアに依存するソフトウェアを特定します。**

```
# prodreg info -m "name" -a "Dependent Components"
```

Dependent Components:

Name	UUID	#
<i>component-a</i>	<i>component-a-UUID</i>	
1		

<code>-m "name"</code>	再インストールするソフトウェアコンポーネントの名前を指定します。
<code>-a "Dependent Components"</code>	<i>name</i> ソフトウェアに依存するコンポーネントを表示します。
<code>component-a</code>	<i>name</i> ソフトウェアに依存するソフトウェアコンポーネントの名前を指定します。
<code>component-a-UUID</code>	<i>component-a</i> ソフトウェアコンポーネントのUUIDを指定します。

component-a ソフトウェアコンポーネントは、再インストールするソフトウェアに依存しています。*component-a* の登録を解除しないで *name* ソフトウェアを再インストールする場合は、*name* ソフトウェアの登録を強制的に解除したあと、*name* ソフトウェアを再インストールします。

4 再インストールするソフトウェアコンポーネントの登録を解除します。

```
# prodreg unregister -f -u UUID
```

5 ソフトウェアコンポーネントを再インストールします。

```
# /usr/bin/java -cp /usr/installers/installer
```

installer オプションには、*name* ソフトウェアのインストーラプログラム名を指定します。

例 20-11 損傷を受けたソフトウェアコンポーネントの再インストール (prodreg)

次の例では、損傷を受けたソフトウェアコンポーネント ComponentSoft を、依存コンポーネントである ExampleSoft の登録解除やアンインストールを行うことなく再インストールします。

```
# prodreg browse -m "ComponentSoft"
BROWSE # +/-/. UUID # NAME
=====
1 - root 1 System Registry
2 + a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b 1 Solaris 10 System Software
3 + 8f64eabf-1dd2-11b2-a3f1-0800209a5b6b 1 Unclassified Software
4 . 86758449-554a-6531-fe90-4352678362fe 1 ComponentSoft

# prodreg info -m "ComponentSoft" -a "Dependent Components"
Dependent Components:
```

Name	UUID	#
-----	-----	-
ExampleSoft	95842091-725a-8501-ef29-0472985982be	1


```
# prodreg unregister -f -u 86758449-554a-6531-fe90-4352678362fe -i 1
```



```
# /usr/bin/java -cp /usr/installers/org.example.componentsoft
```

パッケージコマンドによるソフトウェアの管理 (手順)

この章では、パッケージコマンドを使ってソフトウェアパッケージを追加、確認、および削除する方法について説明します。

これらの作業に関連する手順については、次の節を参照してください。

- 451 ページの「[pkgadd コマンドによる署名付きパッケージの追加および削除 \(作業マップ\)](#)」
- 458 ページの「[パッケージコマンドによるソフトウェアパッケージの管理 \(作業マップ\)](#)」

pkgadd コマンドによる署名付きパッケージの追加および削除 (作業マップ)

次の作業マップは、署名付きパッケージコマンドを使って実行可能なソフトウェア管理作業を示したものです。

作業	説明	参照先
証明書をインポートします。	<code>pkgadm addcert</code> コマンドを使って信頼される証明書をインポートできます。	452 ページの「信頼される証明書を Java キーストアからインポートする方法 (pkgadm addcert)」
1 つ以上の証明書の詳細情報を表示します。	<code>pkgadm listcert</code> コマンドを使って証明書の詳細情報を表示できます。	454 ページの「証明書の情報を表示する方法 (pkgadm listcert)」
証明書を削除します。	<code>pkgadm removecert</code> コマンドを使って証明書を削除できます。	455 ページの「証明書を削除する方法 (pkgadm removecert)」

作業	説明	参照先
プロキシサーバーをセットアップします。	この手順は、プロキシを備えたファイアウォールの背後にシステムが存在している場合に使用します。	455 ページの「プロキシサーバーをセットアップする方法 (pkgadd)」
署名付きパッケージを追加します。	ルート証明書がインポートされると、pkgadd コマンドを使って署名付きパッケージを追加できます。	457 ページの「署名付きパッケージを追加する方法 (pkgadd)」

pkgadd コマンドによる署名付きパッケージの追加および削除

次の手順では、pkgadd コマンドを使って署名付きパッケージを追加および削除する方法について説明します。

▼ 信頼される証明書を **Java** キーストアからインポートする方法 (pkgadm addcert)

- 1
- スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2
- Java™ キーストア内にルート CA (証明書発行局) 証明書が存在することを確認します。

```
# keytool -storepass storepass
-list -keystore certfile
```

keytool

非公開鍵とそれらに対応する公開鍵を認証するための X.509 証明書チェーンの Java キーストア (データベース) を管理します。また、信頼されるエンティティが発行した証明書も管理します。keytool ユーティリティの詳細は、<http://java.sun.com/j2se/1.4.2/docs/tooldocs/solaris/keytool.html> を参照してください。

-storepass storepass

キーストアの完全性を維持するためのパスワードを指定します。

- list デフォルトで、証明書の MD5 指紋を出力します。
- keystore *certfile* キーストア永続ファイルの名前と場所を指定します。

3 Java キーストア内のルート CA 証明書を一時ファイルへエクスポートします。

```
# keytool -export -storepass
storepass -alias verisignclass2g2ca -keystore
/usr/java/jre/lib/security/cacerts certfile -file
filename
```

- export 信頼される証明書をエクスポートします。
- storepass *storepass* Java キーストアの完全性を維持するためのパスワードを指定します。
- alias *verisignclass2g2ca* 信頼される証明書の別名を指定します。
- keystore *certfile* キーストアファイルの名前と場所を指定します。
- file *filename* エクスポートされた証明書を格納するファイルを指定します。

4 信頼される証明書をパッケージキーストアにインポートします。

```
# pkgadm addcert -t -f format
certfile
```

- t 証明書が信頼される CA 証明書であることを示します。このオプションを指定すると、出力に証明書の詳細情報が含まれ、ユーザーはその情報の確認を求められます。
- f *format* 証明書および非公開鍵の形式を指定します。インポートする証明書は、PEM、バイナリ DER のいずれかの形式でコード化されている必要があります。
- certfile* 証明書を含むファイルを指定します。

5 一時ファイルを削除します。

```
# rm /tmp/file-name
```

詳細は、[pkgadm\(1M\)](#) のマニュアルページを参照してください。

例 21-1 Java キーストアから信頼される証明書をインポートする

次の例は、信頼される証明書をインポートする方法を示しています。この例では、keytool コマンドを使って、Java キーストアからパッケージキーストアへ Sun のルート CA 証明書をインポートしています。

```
# keytool -export -storepass changeit -alias verisignclass2g2ca \  
-keystore /usr/java/jre/lib/security/cacerts -file /tmp/root.crt  
Certificate stored in file </tmp/root.crt>
```

```
# pkgadm addcert -t -f der /tmp/root.crt
```

```
Keystore Alias: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O  
Common Name: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O  
Certificate Type: Trusted Certificate  
Issuer Common Name: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O  
Validity Dates: <May 18 00:00:00 1998 GMT> - <Aug 1 23:59:59 2028 GMT>  
MD5 Fingerprint: 2D:BB:E5:25:D3:D1:65:82:3A:B7:0E:FA:E6:EB:E2:E1  
SHA1 Fingerprint: B3:EA:C4:47:76:C9:C8:1C:EA:F2:9D:95:B6:CC:A0:08:1B:67:EC:9D
```

Are you sure you want to trust this certificate? **yes**

Trusting certificate </C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O>

Type a Keystore protection Password. **xxxxxx**

Press ENTER for no protection password (not recommended):

For Verification: Type a Keystore protection Password.

Press ENTER for no protection password (not recommended):

Certificate(s) from </tmp/root.crt> are now trusted

▼ 証明書を表示する方法 (pkgadm listcert)

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 パッケージキーストアの内容を表示します。

```
# pkgadm listcert -p passarg
```

例 21-2 証明書を表示する

次の例は、ローカルに格納されている証明書の詳細情報を表示する方法を示しています。

```
# pkgadm listcert -P pass:test123
```

```
Keystore Alias: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O  
Common Name: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O  
Certificate Type: Trusted Certificate  
Issuer Common Name: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O  
Validity Dates: <May 18 00:00:00 1998 GMT> - <Aug 1 23:59:59 2028 GMT>  
MD5 Fingerprint: 2D:BB:E5:25:D3:D1:65:82:3A:B7:0E:FA:E6:EB:E2:E1  
SHA1 Fingerprint: B3:EA:C4:47:76:C9:C8:1C:EA:F2:9D:95:B6:CC:A0:08:1B:67:EC:9D
```

▼ 証明書を削除する方法 (pkgadm removecert)

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 信頼される証明書をパッケージキーストアから削除します。

```
# pkgadm removecert -n "certfile"
```

```
"
```

removecert の -n "certfile" オプションには、ユーザー証明書/鍵ペアの別名、信頼される証明書の別名のいずれかを指定します。

注 - 証明書の別名を表示するには、pkgadm listcert コマンドを使用します。

例 21-3 証明書を削除する

次の例は、証明書を削除する方法を示しています。

```
# pkgadm listcert
Keystore Alias: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O
Common Name: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O
Certificate Type: Trusted Certificate
Issuer Common Name: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O
Validity Dates: <May 18 00:00:00 1998 GMT> - <Aug 1 23:59:59 2028 GMT>
MD5 Fingerprint: 2D:BB:E5:25:D3:D1:65:82:3A:B7:0E:FA:E6:EB:E2:E1
SHA1 Fingerprint: B3:EA:C4:47:76:C9:C8:1C:EA:F2:9D:95:B6:CC:A0:08:1B:67:EC:9D
# pkgadm removecert -n "/C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O"

Enter Keystore Password: storepass
Successfully removed Certificate(s) with alias \
</C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O>
```

▼ プロキシサーバーをセットアップする方法 (pkgadd)

プロキシを備えたファイアウォールの背後にシステムが存在している場合、先にプロキシサーバーをセットアップします。そのあとに、pkgadd コマンドを使用して HTTP サーバーからパッケージを追加します。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。
- 2 次のいずれかの方法を使ってプロキシサーバーを指定します。
 - a. 環境変数 `http_proxy`、`HTTPPROXY`、`HTTPPROXYPORT` のいずれかを使ってプロキシサーバーを指定します。
次に例を示します。

```
# setenv http_proxy http://mycache.domain:8080
```


または、次のいずれかを指定します。

```
# setenv HTTPPROXY mycache.domain  
# setenv HTTPPROXYPORT 8080
```
 - b. `pkgadd` コマンド行でプロキシサーバーを指定します。
次に例を示します。

```
# pkgadd -x mycache.domain:8080  
-d http://myserver.com/pkg SUNWpkg
```
 - c. プロキシサーバーの情報を含む管理ファイルを作成します。
次に例を示します。

```
# cat /tmp/admin  
mail=  
instance=unique  
partial=ask  
runlevel=ask  
idepend=ask  
rdepend=ask  
space=ask  
setuid=ask  
conflict=ask  
action=ask  
networktimeout=60  
networkretries=3  
authentication=quit  
keystore=/var/sadm/security  
basedir=default  
proxy=mycache.domain:8080
```


続いて、`pkgadd -a` コマンドでこの管理ファイルを指定します。次に例を示します。

```
# pkgadd -a /tmp/admin
-d http://myserver.com/pkg SUNWpkg
```

▼ 署名付きパッケージを追加する方法 (pkgadd)

この手順では、Sun のルート CA 証明書が、すでにインポートされているものと仮定しています。詳細については、[452 ページの「信頼される証明書を Java キーストアからインポートする方法 \(pkgadm addcert\)」](#)を参照してください。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。

- 2 署名付きパッケージを追加します。

```
# pkgadd -d /pathname/device-name
```

`-d device-name` オプションには、パッケージのインストール元デバイスを指定します。デバイスとして指定できるのは、ディレクトリ、テープ、フロッピーディスク、リムーバブルディスクのいずれかです。また、`pkgtrans` コマンドによって作成されたデータストリームをデバイスとして指定することもできます。

例 21-4 署名付きパッケージを追加する

次の例は、システム上に格納された署名付きパッケージを追加する方法を示しています。

```
# # pkgadd -d /tmp/signed_pppd
The following packages are available:
  1  SUNWpppd      Solaris PPP Device Drivers
                        (sparc) 11.10.0,REV=2003.05.08.12.24

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: all
Enter keystore password:
## Verifying signature for signer <User Cert 0>

.
.
.
```

次の例は、デバイス名として HTTP URL を指定して署名付きパッケージをインストールする方法を示しています。指定した URL は、ストリーム形式のパッケージを指している必要があります。

```
# pkgadd -d http://install/signed-video.pkg

## Downloading...
.....25%.....50%.....75%.....100%
## Download Complete
.
.
.
```

パッケージコマンドによるソフトウェアパッケージの管理 (作業マップ)

次の作業マップは、署名付きパッケージと署名なしパッケージの両方に対して、パッケージコマンドで行えるソフトウェア管理作業を示したものです。

作業	説明	参照先
ソフトウェアパッケージをローカルシステムに追加します。	pkgadd コマンドを使ってソフトウェアパッケージをローカルシステムに追加できます。	459 ページの「ソフトウェアパッケージを追加する方法 (pkgadd)」
ソフトウェアパッケージをスプールディレクトリに追加します。	ソフトウェアを実際にインストールしないで、ソフトウェアパッケージをスプールディレクトリに追加できます。	462 ページの「ソフトウェアパッケージをスプールディレクトリに追加する」
インストールされたすべてのソフトウェアパッケージに関する情報を表示します。	pkginfo コマンドを使用して、インストール済みのパッケージに関する情報を表示できます。	465 ページの「インストール済みのすべてのパッケージに関する情報を表示する方法 (pkginfo)」
インストールされたソフトウェアパッケージの整合性を検査します。	pkgchk コマンドを使って、インストールされたソフトウェアパッケージの整合性を検査できます。	466 ページの「インストール済みのソフトウェアパッケージの整合性を検査する方法 (pkgchk)」
インストール済みオブジェクトの整合性を検査します。	pkchk コマンドでオプション -p または -P を指定すると、インストールされたオブジェクトの整合性を検証できます。-p オプションには絶対パス名を指定します。新しい -P オプションには部分パス名を指定します。	467 ページの「インストール済みオブジェクトの整合性を検査する方法 (pkgchk -p, pkgchk -P)」

作業	説明	参照先
ソフトウェアパッケージを削除します。	<code>pkgrm</code> コマンドを使って不要になったソフトウェアパッケージを削除できます。	470 ページの「ソフトウェアパッケージを削除する方法 (pkgrm)」

パッケージコマンドによるソフトウェアパッケージの管理

次の手順では、パッケージコマンドを使ってソフトウェアパッケージを管理する方法を説明します。

▼ ソフトウェアパッケージを追加する方法 (pkgadd)

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、[『Solaris のシステム管理 \(セキュリティサービス\)』の「RBAC の構成 \(作業マップ\)」](#)を参照してください。
- 2 インストール済みのパッケージの中で、追加しようとしているものと同じ名前のものがあれば、すべて削除します。
この手順によって、追加および削除されたソフトウェアの正しい記録が取れます。同じアプリケーションの複数のバージョンをシステムで管理する場合もあります。ソフトウェアの複数のコピーを管理する方法については、[416 ページの「パッケージの削除に関するガイドライン \(pkgrm\)」](#)を参照してください。この作業の手順については、[470 ページの「ソフトウェアパッケージを削除する方法 \(pkgrm\)」](#)を参照してください。

- 3 ソフトウェアパッケージをシステムに追加します。

```
# pkgadd -a admin-file
-d device-name pkgid ...
```

- a *admin-file* (省略可能) インストール時に `pkgadd` コマンドが確認する管理ファイルを示します。管理ファイルの使用方法については、[418 ページの「管理ファイルの使用」](#)を参照してください。
- d *device-name* ソフトウェアパッケージのあるディレクトリの絶対パスを指定します。*device-name* には、デバイス、ディレクトリ、またはスプールディレクトリへのパスを指定できます。パッケージのあるパスを指定しないと、`pkgadd` コマンドはデフォルトのスプールディレクトリ (`/var/spool/pkg`) をチェックします。パッケージがそこにある場合、パッケージのインストールは失敗します。

pkgid (省略可能) インストールする1つまたは複数のパッケージの名前を空白で区切って指定します。省略した場合、**pkgadd** コマンドは、指定されたデバイス、ディレクトリ、またはスプールディレクトリで利用可能なすべてのパッケージをインストールします。

パッケージのインストール中に **pkgadd** コマンドによって問題が検出されると、その問題に関連するメッセージに続いて、次のプロンプトが表示されます。

```
Do you want to continue with this installation?
```

このプロンプトには、**yes**、**no**、または **quit** のいずれかで応答します。複数のパッケージを指定した場合は、**no** と入力して、インストール中のパッケージのインストールを中止してください。ほかのパッケージのインストールが続行されます。インストールを中止するには、**quit** と入力してください。

- 4 指定したパッケージが正常にインストールされていることを確認します。

```
# pkgchk -v pkgid
```

エラーがない場合は、インストールされたファイルのリストが返されます。エラーがある場合は、**pkgchk** コマンドによってエラーが報告されます。

例 21-5 マウント済みの CD からソフトウェアパッケージを追加する

次の例は、マウント済みの Solaris 10 CD から **SUNWpl5u** パッケージをインストールする方法を示しています。また、パッケージファイルが正しくインストールされているかどうかを確認する方法も示します。

```
# pkgadd -d /cdrom/cdrom0/Solaris_10/Product SUNWpl5u
```

```
.
.
.
Installation of <SUNWpl5u> was successful.
# pkgchk -v SUNWpl5u
/usr
/usr/bin
/usr/bin/perl
/usr/perl5
/usr/perl5/5.8.4
.
.
.
```

この例は、Solaris 10 10/08 リリース以上を実行していない場合に使用するパスを示しています。

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_10/Product SUNWpl5u
.
.
.
Installation of <SUNWpl5u> was successful.
# pkgchk -v SUNWpl5u
/usr
/usr/bin
/usr/bin/perl
/usr/perl5
/usr/perl5/5.8.4
.
.
.
```

例 21-6 リモートパッケージサーバーからソフトウェアパッケージをインストールする

インストールするパッケージがリモートシステムから利用できる場合は、そのパッケージが(パッケージ形式で)格納されているディレクトリを手動でマウントして、ローカルシステムにパッケージをインストールすることができます。

次の例は、リモートシステムからソフトウェアパッケージをインストールする方法を示しています。この例では、`package-server` というリモートシステムの `/latest-packages` ディレクトリにソフトウェアパッケージが格納されているものとします。`mount` コマンドはパッケージをローカルの `/mnt` 上にマウントします。`pkgadd` コマンドは `SUNWpl5u` パッケージをインストールします。

```
# mount -F nfs -o ro package-server:/latest-packages /mnt
# pkgadd -d /mnt SUNWpl5u
.
.
.
Installation of <SUNWpl5u> was successful.
```

オートマウントがサイトで実行されている場合は、リモートパッケージサーバーを手動でマウントする必要はありません。代わりに、オートマウントパス(この場合は `/net/package-server/latest-packages`)を、`-d` オプションの引数として使用します。

```
# pkgadd -d /net/package-server/latest-packages SUNWpl5u
.
.
.
Installation of <SUNWpl5u> was successful.
```

例 21-7 管理ファイルを指定してリモートパッケージサーバーからソフトウェアパッケージをインストールする

この例は、前の例に似ていますが、`-a` オプションを使って `noask-pkgadd` という管理ファイルを指定している点が異なります。この管理ファイルについては、[418 ページ](#) の「[パッケージ追加時のユーザーの対話操作を省略する \(pkgadd\)](#)」を参照してください。この例では、`noask-pkgadd` 管理ファイルがデフォルトの格納場所である `/var/sadm/install/admin` にあるものとします。

```
# pkgadd -a noask-pkgadd -d /net/package-server/latest-packages SUNWpl5u
```

```
.
.
.
Installation of <SUNWpl5u> was successful.
```

例 21-8 ソフトウェアパッケージを特定の HTTP URL からインストールする

次の例は、特定の HTTP URL をデバイス名として使ってパッケージをインストールする方法を示しています。指定した URL は、ストリーム形式のパッケージを指している必要があります。

```
# pkgadd -d http://install/xf86-4.3.0-video.pkg
```

```
## Downloading...
.....25%.....50%.....75%.....100%
## Download Complete
```

The following packages are available:

- 1 SUNWxf86r XFree86 Driver Porting Kit (Root)
(i386) 4.3.0,REV=0.2003.02.28
- 2 SUNWxf86u XFree86 Driver Porting Kit (User)
(i386) 4.3.0,REV=0.2003.02.28

```
.
.
.
```

ソフトウェアパッケージをスプールディレクトリに追加する

頻繁にインストールするパッケージをスプールディレクトリにコピーしておくと便利です。パッケージをデフォルトのスプールディレクトリ `/var/spool/pkg` にコ

ピーする場合、`pkgadd` コマンドを使用するときに、コピー元のパッケージの位置 (`-d device-name` 引数) を指定する必要はありません。`pkgadd` コマンドは、デフォルトで `/var/spool/pkg` ディレクトリを調べて、コマンド行に指定されたパッケージをすべて見つけます。パッケージをスプールディレクトリにコピーすることと、パッケージをシステム上にインストールすることとは異なることに注意してください。

▼ ソフトウェアパッケージをスプールディレクトリに追加する方法 (`pkgadd`)

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。
- 2 スプール済みのパッケージの中で、追加しようとしているものと同じ名前のものがあれば、すべて削除します。
スプール済みのパッケージを削除する方法については、[例 21-20](#) を参照してください。

- 3 ソフトウェアパッケージをスプールディレクトリに追加します。

```
# pkgadd -d device-name
-s spooldir pkgid ...
```

`-d device-name` ソフトウェアパッケージのあるディレクトリの絶対パスを指定します。`device-name` には、デバイス、ディレクトリ、またはスプールディレクトリへのパスを指定できます。

`-s spooldir` パッケージがスプールされるスプールディレクトリの名前を指定します。`spooldir` を指定する必要があります。

`pkgid` (省略可能) スプールディレクトリに追加する 1 つまたは複数のパッケージの名前を空白で区切って指定します。この引数を省略すると、使用可能なすべてのパッケージがコピーされます。

- 4 指定したパッケージがスプールディレクトリに正常にコピーされていることを確認します。

```
$ pkginfo -d spooldir
| grep pkgid
```

`pkgid` が正常にコピーされた場合、`pkginfo` コマンドによって、`pkgid` に関する 1 行の情報が返されます。正常にコピーされなかった場合は、`pkginfo` コマンドによってシステムプロンプトが返されます。

例 21-9 マウント済みのCDからスプールディレクトリを設定する

次の例は、SUNWman パッケージをマウント済みの SPARC Solaris 10 CD からデフォルトのスプールディレクトリ (/var/spool/pkg) に転送する方法を示しています。

```
# pkgadd -d /cdrom/cdrom0/Solaris_10/Product -s /var/spool/pkg SUNWman
```

```
Transferring <SUNWman> package instance
```

例 21-10 リモートソフトウェアパッケージサーバーからスプールディレクトリを設定する

コピーするパッケージがリモートシステムから利用できる場合は、そのパッケージが (パッケージ形式で) 格納されているディレクトリを手動でマウントして、ローカルスプールディレクトリにパッケージをコピーすることができます。

次の例は、そのためのコマンドを示したものです。この例では、package-server というリモートシステムの /latest-packages ディレクトリにソフトウェアパッケージが格納されているものとします。mount コマンドはパッケージディレクトリをローカルの /mnt 上にマウントします。pkgadd コマンドは、SUNWpl5p パッケージを /mnt からデフォルトのスプールディレクトリ (/var/spool/pkg) にコピーします。

```
# mount -F nfs -o ro package-server:/latest-packages /mnt
```

```
# pkgadd -d /mnt -s /var/spool/pkg SUNWpl5p
```

```
Transferring <SUNWpl5p> package instance
```

オートマウントが実行されている場合は、リモートパッケージサーバーを手動でマウントする必要はありません。代わりに、オートマウントパス (この場合は、/net/package-server/latest-packages) を、-d オプションの引数として使用してください。

```
# pkgadd -d /net/package-server/latest-packages -s /var/spool/pkg SUNWpl5p
```

```
Transferring <SUNWpl5p> package instance
```

例 21-11 デフォルトのスプールディレクトリからソフトウェアパッケージをインストールする

次の例は、デフォルトのスプールディレクトリから SUNWpl5p パッケージをインストールする方法を示しています。オプションを何も指定しないと、pkgadd コマンドは /var/spool/pkg ディレクトリを検索して名前付きのパッケージを見つけます。

```
# pkgadd SUNWpl5p
```

```
.\n.
```



```
Installation of <SUNWpl5p> was successful.
```

▼ インストール済みのすべてのパッケージに関する情報を表示する方法 (pkginfo)

- インストール済みのパッケージに関する情報を表示するには、pkginfo コマンドを使用します。

```
$ pkginfo
```

例 21-12 インストール済みのパッケージを表示する

この例は、システムがスタンドアロンシステム、サーバーのいずれであるかにかかわらず、そのローカルシステム上にインストールされているすべてのパッケージを表示する方法を示しています。基本カテゴリ、パッケージ名、およびパッケージの説明が表示されています。

```
$ pkginfo
system      SUNWaccr      System Accounting, (Root)
system      SUNWaccu      System Accounting, (Usr)
system      SUNWadmap     System administration applications
system      SUNWadmc      System administration core libraries
.
.
.
```

例 21-13 ソフトウェアパッケージに関する詳細情報を表示する

この例は、システム上にインストールされているすべてのパッケージを長形式で表示する方法を示しています。この表示には、指定されたパッケージで利用可能なすべての情報が含まれます。

```
$ pkginfo -l SUNWcar
PKGINST:  SUNWcar
NAME:      Core Architecture, (Root)
CATEGORY:  system
ARCH:      sparc.sun4u
VERSION:   11.9.0,REV=2002.04.06.15.27
BASEDIR:   /
VENDOR:    Sun Microsystems, Inc.
DESC:      core software for a specific hardware platform group
PSTAMP:    leo20031003183400
INSTDATE:  Feb 20 2004 16:57
```

```
HOTLINE: Please contact your local service provider
STATUS:  completely installed
FILES:    114 installed pathnames
          36 shared pathnames
          40 directories
          57 executables
          21469 blocks used (approx)
```

▼ インストール済みのソフトウェアパッケージの整合性を検査する方法 (pkgchk)

- 1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「RBAC の構成 (作業マップ)」を参照してください。

- 2 インストール済みのパッケージの状態を調べます。

- ファイルの属性または内容を検査するには、次のように入力します。

```
# pkgchk -a| -c -v
pkgid ...
```

- スプールディレクトリの絶対パスを指定するには、次のように入力します。

```
# pkgchk -d spooldir pkgid ...
```

-a ファイルの属性と内容(デフォルト)ではなく、ファイルの属性(アクセス権)だけを検査することを指定します。

-c ファイルの内容と属性(デフォルト)ではなく、ファイルの内容だけを検査することを指定します。

-v 冗長モードを指定します。このモードでは、処理されるたびにファイル名が表示されます。

-d *spooldir* スプールディレクトリへの絶対パスを指定します。

pkgid (省略可能) 1 つまたは複数のパッケージの名前を空白で区切って指定します。*pkgid* を指定しない場合は、システムにインストールされているすべてのソフトウェアパッケージが検査されます。

例 21-14 インストール済みのソフトウェアパッケージの内容を検査する

次の例は、パッケージの内容を検査する方法を示しています。

```
# pkgchk -c SUNWbash
```

エラーがない場合は、システムプロンプトが返されます。エラーがある場合は、pkgchk コマンドによってエラーが報告されます。

例 21-15 インストール済みのソフトウェアパッケージのファイル属性を検査する

次の例は、パッケージのファイル属性を検査する方法を示しています。

```
# pkgchk -a SUNWbash
```

エラーがない場合は、システムプロンプトが返されます。エラーがある場合は、pkgchk コマンドによってエラーが報告されます。

例 21-16 スプールディレクトリにインストールされているソフトウェアパッケージを検査する

次の例は、スプールディレクトリ (/export/install/packages) にコピーされたソフトウェアパッケージを検査する方法を示しています。

```
# pkgchk -d /export/install/packages
## checking spooled package <SUNWadmap>
## checking spooled package <SUNWadmfw>
## checking spooled package <SUNWadmc>
## checking spooled package <SUNWsadml>
```

スプールされたパッケージの検査には限界があります。これは、パッケージがインストールされないすべての情報を検査できないからです。

▼ インストール済みオブジェクトの整合性を検査する方法 (pkgchk -p、**pkgchk -P**)

この手順は、pkgchk コマンドを使ってインストール済みオブジェクトの整合性を検査する方法を示しています。新しい -p オプションを使用すると、部分パスを指定できます。このオプションは、ファイルとパッケージ間のマッピングを支援する目的で追加されました。このオプションを -l オプションと組み合わせると、指定した部分パスを含むファイルの情報が一覧表示されます。インストール済みオブジェクト

の整合性を絶対パスを指定して検査するには、`-p` オプションを使用します。詳細は、[pkgchk\(1M\)](#) のマニュアルページを参照してください。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。
- 2 インストール済みオブジェクトの整合性を検査します。
 - 1つまたは複数の絶対パス名に対するインストール済みオブジェクトの整合性を検査するには、次のように入力します。

```
# pkgchk -lp path-name
```

- 1つまたは複数の部分パス名に対するインストール済みオブジェクトの整合性を検査するには、次のように入力します。

```
# pkgchk -lP partial-path-name
```

<code>-p path</code>	指定された1つまたは複数のパス名に対する整合性のみを検査します。パスは、1つまたは複数のパス名をコンマで区切って指定します。ファイルの属性と内容(デフォルト)ではなく、ファイルの属性(アクセス権)だけを検査することを指定します。
<code>-P partial-path</code>	指定された1つまたは複数の部分パス名に対する整合性のみを検査します。部分パスは、1つまたは複数の部分パス名をコンマで区切って指定します。部分パスに含まれる文字列を含むすべてのパス名に一致します。ファイルの内容と属性(デフォルト)ではなく、ファイルの内容だけを検査することを指定します。
<code>-l</code>	同一パッケージに属する選択されたファイルに関する情報を表示します。このオプションを、 <code>-a</code> 、 <code>-c</code> 、 <code>-f</code> 、 <code>-g</code> 、 <code>-v</code> の各オプションと同時に使用することはできません。冗長モードを指定します。このモードでは、処理されるたびにファイル名が表示されます。

例 21-17 絶対パス名を指定してインストール済みオブジェクトの整合性を検査する

この例では、`pkgchk -lp` コマンドで絶対パス名を指定してファイルシステム上の特定のオブジェクトの内容/属性を検査する方法を示しています。`-l` オプションは、同一パッケージに属する選択されたファイルに関する情報を表示します。

```
# pkgchk -lp /usr/sbin/pkgadd
Pathname: /usr/sbin/pkgadd
Type: regular file
```

```

Expected mode: 0555
Expected owner: root
Expected group: sys
Expected file size (bytes): 867152
Expected sum(1) of contents: 45580
Expected last modification: Jul 02 02:20:34 2004
Referenced by the following packages:
    SUNWpkgcmdsu
Current status: installed

```

例 21-18 部分パス名を指定してインストール済みオブジェクトの整合性を検査する

この例では、`pkgchk -lP` コマンドでファイル名やディレクトリ名などの部分パス名を指定してファイルシステム上の特定のオブジェクトの内容/属性を検査する方法を示しています。`-l` オプションは、同一パッケージに属する選択されたファイルに関する情報を表示します。

```

# pkgchk -lP /sbin/pkgadd
Pathname: /usr/sbin/pkgadd
Type: regular file
Expected mode: 0555
Expected owner: root
Expected group: sys
Expected file size (bytes): 867152
Expected sum(1) of contents: 45580
Expected last modification: Jul 02 02:20:34 2004
Referenced by the following packages:
    SUNWpkgcmdsu
Current status: installed

Pathname: /usr/sbin/pkgask
Type: linked file
Source of link: ../../usr/sbin/pkgadd
Referenced by the following packages:
    SUNWpkgcmdsu
Current status: installed

```

ソフトウェアパッケージを削除する

ソフトウェアパッケージを削除 (アンインストール) するには、そのパッケージの追加 (インストール) 時に使用した関連ツールを使用します。たとえば、Solaris インストール GUI を使ってインストールしたソフトウェアは、Solaris インストール GUI を使ってアンインストールします。



注意-ソフトウェアパッケージを削除するのに `rm` コマンドを使用しないでください。それを行なった場合、システム上にインストールされたすべてのパッケージを追跡しているデータベースの整合性がとれなくなります。

▼ ソフトウェアパッケージを削除する方法 (pkgrm)

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『[Solaris のシステム管理 \(セキュリティサービス\)](#)』の「[RBAC の構成 \(作業マップ\)](#)」を参照してください。
- 2 インストール済みのパッケージを削除します。

```
# pkgrm pkgid ...
```

`pkgid` は、削除する 1 つまたは複数のパッケージの名前 (空白で区切られている) を識別します。この引数を省略すると、使用可能なパッケージがすべて削除されます。

例 21-19 ソフトウェアパッケージを削除する

次の例は、パッケージの削除方法を示しています。

```
# pkgrm SUNWctu
```

```
The following package is currently installed:
```

```
SUNWctu          Netra ct usr/platform links (64-bit)
                  (sparc.sun4u) 11.9.0,REV=2001.07.24.15.53
```

```
Do you want to remove this package? y
```

```
## Removing installed package instance <SUNWctu>
## Verifying package dependencies.
## Processing package information.
## Removing pathnames in class <none>
.
.
.
```

例 21-20 スプールされたソフトウェアパッケージを削除する

次の例は、スプールされたパッケージの削除方法を示しています。

```
# pkgrm -s /export/pkg SUNWaudh
The following package is currently spooled:
  SUNWaudh      Audio Header Files
                (sparc) 11.10.0,REV=2003.08.08.00.03
Do you want to remove this package? y
Removing spooled package instance <SUNWaudh>
```


patchadd コマンドによる Solaris パッチの管理 (手順)

パッチの管理には、Solaris パッチとソフトウェア更新のシステムへの「適用」が含まれます。また、不要なパッチや障害の発生したパッチの削除が含まれる場合があります。パッチの削除は、パッチの「バックアウト」とも呼ばれます。

この章では、patchadd コマンドを使って Solaris パッチを管理する手順について説明します。追加情報については、[patchadd\(1M\)](#)のマニュアルページを参照してください。

この章の内容は次のとおりです。

- 474 ページの「パッチの種類」
- 475 ページの「Solaris パッチへのアクセス」
- 476 ページの「Solaris オペレーティングシステムでのパッチ管理」
- 477 ページの「Solaris パッチ管理の用語と定義」
- 479 ページの「patchadd コマンドによる Solaris パッチの管理 (作業マップ)」

注 - **Solaris 10 5/08:** この情報は Solaris 10 5/08 リリースで追加されたものですが、すべての Solaris 10 OS に当てはまります。使用している Solaris システムを登録するには、<https://inventory.sun.com/inventory/> にアクセスしてください。Sun Inventory を使用してハードウェア、ソフトウェア、およびオペレーティングシステムを登録する方法については、[Sun Inventory Information Center \(http://wikis.sun.com/display/SunInventory/Sun+Inventory\)](http://wikis.sun.com/display/SunInventory/Sun+Inventory) を参照してください。

Sun Ops Center を使用してデータセンターでシステムをプロビジョニング、更新、および管理する場合は、ソフトウェアを Sun Ops Center に登録する方法を [Sun xVM Information Center \(http://wikis.sun.com/display/xVM/Sun+xVM+Ops+Center\)](http://wikis.sun.com/display/xVM/Sun+xVM+Ops+Center) で参照してください。

ディスクレスクライアントシステムにパッチを適用する方法については、[179 ページ](#)の「ディスクレスクライアント OS サービスにパッチを適用する」を参照してください。

Solaris パッチ使用時に推奨される戦略やプラクティスについては、『[Solaris Patch Management: Recommended Strategies](#)』を参照してください。

パッチの種類

「パッチ」とは、Solaris OS 内またはその他のサポート対象ソフトウェア内の既知または潜在的な問題に対する修正をまとめたものです。また、パッチは、特定のソフトウェアリリースに対する新機能や機能拡張の提供も行います。パッチは、既存のファイルやディレクトリを置換または更新するファイルやディレクトリから構成されます。ほとんどの Solaris パッチは、一連の疎パッケージとして提供されます。パッケージの詳細は、[第 19 章「ソフトウェアの管理 \(概要\)」](#)を参照してください。

ソフトウェアの「更新」とは、既存の問題を訂正したり機能を導入したりするために、ソフトウェアに適用する変更のことです。また、更新は、ソフトウェア更新をシステムに適用するプロセスでもあります。

patchadd コマンドを使用すれば、Solaris システム上でパッチを管理できます。

署名付きパッチと署名なしパッチ

「[署名付き](#)」とは、「[デジタル署名](#)」が適用されたパッチのことです。パッチのデジタル署名が検証されると、その署名の適用後にそのパッチが変更されていないことが保証されます。パッチがユーザーのシステムに「[{3598}ダウンロードされた](#)」後、署名付きパッチのデジタル署名は検証されます。

Solaris 2.6 以降のリリースの Solaris OS 用パッチは、署名付きパッチおよび「[署名なしパッチ](#)」として利用可能です。署名なしパッチにはデジタル署名は含まれていません。

署名付きパッチは Java アーカイブ形式 (JAR) ファイルに格納され、SunSolve OnlineSM から入手できます。署名なしパッチはディレクトリ形式で格納され、同じく SunSolve Online から .zip ファイルとして入手できます。

patchadd コマンドを使ってシステムにパッチを適用する方法については、[479 ページの「patchadd コマンドによる Solaris パッチの管理 \(作業マップ\)」](#)を参照してください。

署名付きパッチに関するその他の概要情報については、[410 ページの「署名付きのパッケージ、パッチ、およびソフトウェア更新」](#)を参照してください。

Solaris パッチへのアクセス

Sun の顧客は、SunSolve Patch Portal Web サイトからパッチにアクセスできます。ただし、一部のパッチは、SunSpectrumSM や Solaris Service Plan の顧客など、サービスプランに加入している顧客にしかアクセスできないことがあります。いずれの場合も、SunSolve Patch Portal に入るには、Sun に登録して Sun オンライン ID を取得する必要があります。これらのパッチは、夜間に更新されます。

Solaris パッチは、<http://sunsolve.sun.com> の Web サイトから取得できます。SunSolve Patch Portal Web サイトからパッチにアクセスするには、システムがインターネットに接続されており、かつ Mozilla ブラウザなどの Web ブラウザを実行できる必要があります。

個々のパッチにアクセスしたり、パッチクラスタから一連のパッチにアクセスしたり、パッチレポートを参照したりできます。

各パッチには、そのパッチに関する情報を含む README ファイルが関連付けられています。

Solaris パッチの番号付け

パッチは、一意の「**パッチ ID**」によって識別されます。パッチ ID とは、パッチのベースコードとパッチのバージョン番号を表す数字とをハイフンでつなぎ合わせた英数字文字列のことです。たとえば、パッチ 118833-10 は、SunOS 5.10 カーネル更新用パッチ、第 10 リビジョンのパッチ ID です。

Solaris パッチの管理

この節では、利用可能な Solaris パッチツールを使って Solaris パッチを管理する方法について説明します。

パッチツールは次のことを行います。

- 管理ホストとターゲットホストの Solaris バージョン番号を確認する。
- パッチパッケージの pkginfo ファイル内に含まれる次の情報を更新する。
 - 適用するパッチによって「**廃棄**」されたパッチ
 - このパッチが必要とするその他のパッチ
 - このパッチと「**互換性のない**」パッチ

パッチを適用している間は、patchadd コマンドによって `/var/sadm/patch/patch-id` /log ファイル内に情報がロギングされます。

注 - この Solaris リリースでは、`patchadd -M` コマンドが改善されています。このコマンドを使用してシステムにパッチを適用するときに、パッチ ID を番号順に指定する必要がなくなりました。パッチ ID を指定しないで `patchadd -M` コマンドを使用すれば、ディレクトリ内のすべてのパッチがシステムにインストールされます。これらの変更の詳細は、[patchadd\(1M\)](#) のマニュアルページを参照してください。

この `patchadd` コマンドは、次の場合にはパッチまたはソフトウェア更新を適用できません。

- パッケージが、システムに完全にインストールされていない。
- パッチパッケージのアーキテクチャーが、システムのアーキテクチャーと異なる。
- パッチパッケージのバージョンが、インストール済みパッケージのバージョンと一致しない。
- 同じベースコードと上位のバージョン番号を持つパッチが、すでに適用されている。
- このパッチを廃棄するパッチが、すでに適用されている。
- このパッチと互換性のないパッチが、すでにシステムに適用されている。この情報は、適用済みの各パッチの `pkginfo` ファイル内に格納されている。
- 適用するパッチが依存するパッチのいずれかが、まだ適用されていない。

Solaris オペレーティングシステムでのパッチ管理

次の情報に基づいて Solaris パッチの管理作業を特定してください。各作業は、署名付きパッチや署名なしパッチの管理をはじめとする追加作業を示しています。

作業	説明	参照先
署名付きパッチと署名なしパッチのどちらを適用するかを決定します。	現在の環境にとって署名付きパッチと署名のないパッチのどちらが適しているかを判断します。	477 ページの「システムに署名付きパッチ、署名なしパッチのいずれを適用するか決定」
システムにパッチを適用します。	Solaris 2.6、Solaris 7、Solaris 8、Solaris 9、または Solaris 10 システムで、 <code>patchadd</code> コマンドを使って署名なしの Solaris パッチを適用します。	479 ページの「<code>patchadd</code> コマンドによる Solaris パッチの管理 (作業マップ)」

システムに署名付きパッチ、署名なしパッチのいずれを適用するか の決定

システムに署名付きパッチ、署名なしパッチのいずれを適用するかを決定する際に重要となるのは、パッチの供給元を信頼するかどうかです。

パッチの供給元(既知の販売店から入手したパッチ CD や信頼できる Web サイトとの HTTPS 接続など)を信頼する場合、署名なしパッチを使用できます。一方、供給元を信頼しない場合は署名付きパッチを使用します。

パッチの供給元を信頼できるかどうかわからない場合は、署名付きパッチを使用してください。

Solaris パッチ管理の用語と定義

次の用語は、パッチ管理に関する各章で使用されているものです。

適用	パッチをシステムにインストールすること。
バックアウト	パッチをシステムから削除すること。
バックアウトデータ	パッチを適用したときに作成されるデータ。そのパッチを削除(バックアウト)した場合に、システムを元の状態に戻せるようにするためのデータ。
バックアウトディレクトリ	バックアウトデータが格納されるディレクトリ。デフォルトでは、パッチによってインストールされた各パッケージの <code>save</code> ディレクトリです。
依存関係	下記の パッチの依存関係 を参照してください。
デジタル署名	電子署名の 1 つ。署名が適用されたあとに、ドキュメントが変更されていないことを保証するために使用されます。
ダウンロード	パッチソース(Sun パッチサーバーなど)の 1 つ以上のパッチを、それらの適用先のシステムにコピーすること。
ダウンロードディレクトリ	パッチソースからダウンロードされたパッチが格納されるディレクトリ。このディレクトリに格納されたパッチがシステムに適用されます。デフォルトの場所は <code>/var/sadm/spool</code> 。
キーストア	署名付きパッチを適用しようとするときに照会される証明書と鍵のレポジトリ。
非標準パッチ	非標準パッチは、 <code>patchadd</code> コマンドを使ってインストールすることができません。非標準パッチは通常、パッケージ形式で提供されないファームウェアやソフトウェアアプリケーションの修正を提供するために使用されますが、こうしたパッチはその README ファイル内で指定された手順に従ってインストールする必要があります。
順序付け	一連のパッチを適用しやすいようにソートすること。

パッケージ	システムに配布してインストールするためのソフトウェア製品の形式。定義済みフォーマットによるファイルとディレクトリの集まり。
patch	既存の問題を訂正したり特定の機能を導入するためのソフトウェア更新。
パッチ解析	システムに適したパッチを判断するためにシステムを検査する方式。
パッチの依存関係	あるパッチが、システム上に存在するほかのパッチと依存関係を持っていること。1つ以上のパッチと依存関係を持っているパッチを適用するには、依存関係を持つパッチがシステムにすでに適用されている必要があります。
パッチ ID	一意の英数字の文字列。パッチのベース番号、ハイフン (-)、パッチの改訂バージョン番号を表す数字で構成されます。
パッチの非互換性	まれに、2つのパッチが同じシステム上で共存できないこと。この問題は、パッチの相互関係で、各パッチが互いに互換性を持っていないために発生します。適用しようとするパッチがシステムに適用済みのパッチと互換性を持っていない場合には、適用済みのパッチを最初に削除する必要があります。適用済みのパッチを削除すれば、新しいパッチを適用できます。
パッチ一覧	パッチの一覧が含まれるファイル。各行にパッチ ID が付いています。パッチ一覧は、パッチ操作を実行するときに使用されます。パッチ一覧は、システムの解析またはユーザー入力に基づいて生成されます。 パッチ一覧の各行は2つの列で構成されています。第1列はパッチ ID、第2列はパッチの機能説明です。
パッチの廃棄	あるパッチが別のパッチを置き換える際、古いパッチを廃棄すること。古いパッチがシステムに適用されていない場合も含まれます。あるパッチによって1つ以上のパッチが廃棄されると、それらの古いパッチは完全に置き換えられます。新しいパッチを適用する前に、古いパッチが適用されている必要はありません。
パッチサーバー	パッチ解析の結果をもとに、システムによって使用される適切な Solaris パッチソースを取得するためのサーバー。
署名付きパッチ	有効なデジタル署名によって署名されたパッチ。署名付きパッチでは、署名のないパッチに比べて高いセキュリティが保証されます。パッチがシステムに適用される前に、パッチのデジタル署名が検証されます。有効なデジタル署名は、署名が適用された以降にパッチの変更が行われていないことを保証します。署名付きパッチは、Java Archive (JAR) 形式のファイルに格納されます。
ソフトウェア更新	既存の問題を訂正したり特定の機能を導入するために適用する、ソフトウェアへの変更。
特殊処理	シングルユーザーモードでインストールする必要があることを示すプロパティーを含むパッチ。また、適用後にシステムを再起動する必要があるパッチは、「特殊処理要件」を持つパッチ、と呼ばれます。
標準パッチ	標準パッチとは Solaris パッチ仕様に準拠したパッチのことであり、 <code>patchadd</code> コマンドを使ってインストールできます。非標準パッチは、 <code>patchadd</code> コマンドを使ってインストールすることができません

Sun アラート

顧客のコンピュータ環境または生産性に否定的な影響を与える可能性がある既知の製品の問題について、顧客に通知すること。ある問題が Sun アラートの通知対象として認められるには、可用性、セキュリティ、データ損失の少なくとも 1 つにかかわる問題を伴っている必要があります。

SunSolve Online

パッチ、パッチ情報、およびパッチクラスタへのアクセスを提供する、Sun Microsystems のパッチポータル Web サイト。詳細は、<http://sunsolve.sun.com> を参照してください。

署名なしパッチ

デジタル署名を使用して署名されていないパッチ。

Web プロキシ

システムをインターネットに接続するために使用されるサーバー。システムからインターネットに直接接続することはできず、必ず Web プロキシを使用して接続を確立します。

patchadd コマンドによる **Solaris** パッチの管理 (作業マップ)

作業	説明	参照先
1. (オプション) パッケージキーストアをセットアップします。	システムに署名付きパッチを適用する予定がある場合、Sun ルート CA 証明書をパッケージキーストアに事前にインポートしておく必要があります。	480 ページの「信頼される証明書をパッケージキーストアにインポートする方法」
2. (オプション) Web プロキシを指定します。	Web プロキシを備えたファイアウォールの背後にシステムが存在している場合、Sun パッチサーバーからパッチを取得するには、その Web プロキシを指定する必要があります。	482 ページの「Web プロキシを指定する方法」
3. パッチをダウンロードおよび適用します。	patchadd コマンドを使ってパッチをシステムにダウンロードおよび適用します。	483 ページの「Solaris パッチをダウンロードおよび適用する方法」
4. (オプション) システムに適用済みのパッチに関する情報を表示します。	すでにシステムに適用されているパッチに関する情報を表示するには、patchadd、showrev、pkgparam のいずれかのコマンドを使用します。	485 ページの「Solaris パッチの情報を表示する方法」
5. (オプション) システムからパッチを削除します。	必要に応じて、patchrm コマンドを使ってシステムからパッチを削除します。	486 ページの「patchrm コマンドを使用して Solaris パッチを削除する方法」

▼ 信頼される証明書をパッケージキーストアにインポートする方法

patchadd コマンドを使って「署名付きパッチ」をシステムに適用するには、その署名付きパッチの署名を検証できるように、Sun ルート CA 証明書を最低限追加しておく必要があります。この証明書は、Java 「キーストア」からパッケージキーストアへインポートできます。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
- 2 patchadd コマンドを使って署名付きパッチをインストールする場合は、信頼できる新しい Verisign 証明書をキーストアに追加します。
 - a. Web ブラウザを開き、<http://www.sun.com/pki/certs/ca/> Web サイトに移動します。
 - b. 「VeriSign Public Primary Certification Authorities」セクションを探し、**Class 2 Public Primary Certification Authority - G2** 証明書をダウンロードします。
証明書の識別名は次のとおりです。
C=US, O=VeriSign, Inc., OU=Class 2 Public Primary Certification Authority - G2, OU=(c) 1998 VeriSign, Inc. - For authorized use only, OU=VeriSign Trust Network
 - c. バイナリ形式(DER エンコード)を選択します。
 - d. 証明書をファイル /tmp/root.crt にコピーします。

注 - Class 2 Public Primary Certification Authority - G2 証明書をダウンロードできない場合は、[481 ページ](#)の「Java キーストアからのルート CA 証明書のエクスポート」を参照し、代替の手順に従ってください。

- 3 一時ファイル内のルート CA 証明書をパッケージキーストアにインポートします。
システム管理者が変更しないかぎり、Java キーストアのデフォルトのパスワードは changeit です。
次に例を示します。

```
# pkgadm addcert -t -f der /tmp/root.crt
```

```
Keystore Alias: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O
Common Name: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O
Certificate Type: Trusted Certificate
Issuer Common Name: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O
Validity Dates: <May 18 00:00:00 1998 GMT> - <Aug 1 23:59:59 2028 GMT>
```



```
MD5 Fingerprint: 2D:BB:E5:25:D3:D1:65:82:3A:B7:0E:FA:E6:EB:E2:E1
SHA1 Fingerprint: B3:EA:C4:47:76:C9:C8:1C:EA:F2:9D:95:B6:CC:A0:08:1B:67:EC:9D
```

```
Are you sure you want to trust this certificate? yes
Trusting certificate </C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O>
Type a Keystore protection Password. changeit
Press ENTER for no protection password (not recommended):
For Verification: Type a Keystore protection Password.
Press ENTER for no protection password (not recommended):
Certificate(s) from </tmp/root.crt> are now trusted
```

- t 証明書が信頼される CA 証明書であることを示します。コマンド出力に証明書の詳細情報が含まれ、ユーザーはその内容の確認を求められます。
- f *format* 証明書または非公開鍵の形式を指定します。インポートする証明書は、PEM (pem)、バイナリ DER (der) のいずれかの形式を使ってコード化されている必要があります。
- certfile* 証明書を含むファイルを指定します。

4 証明書の情報を表示します。

```
# pkgadm listcert
Enter Keystore Password: storepass
Keystore Alias: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O
Common Name: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O
Certificate Type: Trusted Certificate
Issuer Common Name: /C=US/O=VeriSign, Inc./OU=Class 2 Public Primary Certification Authority - G2/O
Validity Dates: <May 18 00:00:00 1998 GMT> - <Aug 1 23:59:59 2028 GMT>
MD5 Fingerprint: 2D:BB:E5:25:D3:D1:65:82:3A:B7:0E:FA:E6:EB:E2:E1
SHA1 Fingerprint: B3:EA:C4:47:76:C9:C8:1C:EA:F2:9D:95:B6:CC:A0:08:1B:67:EC:9D
```

5 一時ファイルを削除します。

```
# rm /tmp/root.crt
```

Java キーストアからのルート CA 証明書のエクスポート

480 ページの「信頼される証明書をパッケージキーストアにインポートする方法」の手順 2 の説明に従って、信頼できる Class 2 Public Primary Certification Authority - VeriSign 社の G2 証明書を <http://www.sun.com/pki/certs/ca/> からダウンロードできない場合は、ルート CA 証明書を Java キーストアから一時ファイルにエクスポートできます。

次に例を示します。

```
# keytool -export -storepass changeit -alias verisignclass2g2ca \
-keystore /usr/java/jre/lib/security/cacerts -file /tmp/root.crt
Certificate stored in file </tmp/root.crt>
```

-export	信頼される証明書をエクスポートします。
-storepass <i>storepass</i>	Java キーストアの完全性を維持するためのパスワードを指定します。
-alias <i>verisignclass2g2ca</i>	信頼される証明書の別名を指定します。
-keystore <i>certfile</i>	キーストアファイルの名前と場所を指定します。
-file <i>filename</i>	エクスポートされた証明書を格納するファイルを指定します。

これで、ルート CA 証明書を一時ファイルからパッケージキーストアにインポートする準備が整いました。手順については、[480 ページの「信頼される証明書をパッケージキーストアにインポートする方法」](#)の節に記載されている残りの手順を参照してください。

▼ Web プロキシを指定する方法

Web プロキシを備えたファイアウォールの背後にシステムが存在している場合、patchadd を使ってパッチを「適用」するには、その Web プロキシを指定する必要があります。

- 1 スーパーユーザーになるか、同等の役割を引き受けます。
- 2 次のいずれかの方法を使って Web プロキシを指定します。
 - 環境変数 `http_proxy`、`HTTPPROXY`、`HTTPPROXYPORT` のいずれかを使って Web プロキシを指定します。
次に例を示します。

```
# setenv http_proxy http://mycache.domain:8080
```

または、次のいずれかを指定します。

```
# setenv HTTPPROXY mycache.domain
# setenv HTTPPROXYPORT 8080
```

- patchadd コマンド行で Web プロキシを指定します。
次に例を示します。

```
# patchadd -x mycache.domain:8080 \
-M http://www.sun.com/solaris/patches/latest 101223-02 102323-02
```

patchadd -R を使って代替 root パスを作成する際の制限

ゾーンに対応していない Solaris リリースが稼働するシステム上では、patchadd コマンドまたは -R オプションを指定できる任意のコマンドを使って、非大域ゾーンがインストールされた大域ゾーンの代替 root パスを指定しても、そのコマンドは正しく動作しません。

代替ブート環境で非大域ゾーンが構成済みであるがインストール済みではない場合、-R オプションを使ってソフトウェアパッケージとパッチを追加および削除できます。

潜在的な問題を回避するには、-R オプションを使って代替ルートパスを作成しないようにしてください。

Solaris 10 OS を実行している場合には、次のいずれかの方法を選択することもできます。

- Solaris 10 1/06 以降の OS を実行していないシステムのすべてを、Solaris 10 1/06 リリースにアップグレードします。
- Solaris 10 初期 3/05 リリースを実行している場合、次のパッチをインストールすれば、-R オプションを指定できるコマンドを使って代替ルートパスを作成できるようになります。
 - **SPARC** システムの場合 - パッチ 119254 のリビジョン 19 以降をインストールします。
 - **x86** システムの場合 - パッチ 119255 のリビジョン 19 以降をインストールします。
- Solaris 10 リリースなどの代替ルートを、アクティブな OS としてブートします。これで、-R オプションを使用しないでパッケージおよびパッチをインストールおよびアンインストールできるようになります。

詳細は、[patchadd\(1M\)](#)、[patchrm\(1M\)](#)、[pkgadd\(1M\)](#)、および [pkgrm\(1M\)](#) のマニュアルページを参照してください。

▼ Solaris パッチをダウンロードおよび適用する方法

署名付き Solaris パッチまたは「署名なしパッチ」を「[ダウンロードする](#)」し、それをシステムに適用するには、次の手順を実行します。

署名付きパッチを適用するには、パッケージキーストアを事前にセットアップしておく必要があります。

- 1 次のいずれかの方法でシステムにアクセスします。
 - パッチを適用するシステムにログインします。
 - パッチをダウンロードし、それを ftp コマンド経由でターゲットシステムにコピーします。
- 2 Web ブラウザを起動し、<http://sunsolve.Sun.COM> の SunSolve Online Patch Portal にアクセスします。
- 3 特定のパッチをダウンロードするのか、パッチクラスタをダウンロードするのかを決定したあと、次のいずれかを実行します。
 - 「パッチ検索」検索フィールドにパッチ番号 (*patch-id*) を入力し、「パッチ検索」をクリックします。

patch-id を入力すると、最新バージョンのパッチがダウンロードされます。

自由に使用できるパッチの場合、そのパッチの README が表示されます。このパッチが自由に使用できない場合は、ACCESS DENIED メッセージが表示されます。

SPARC システムと x86 システムのパッチ番号が異なることに注意してください。「[パッチ ID](#)」は、パッチの README 内に記載されています。システムのアーキテクチャーに適合したパッチを適用してください。
 - パッチを適用するシステム上で動作している Solaris リリースに適合した推奨パッチクラスタを選択します。
- 4 次の手順に従ってパッチをダウンロードします。
 - 署名付きパッチをダウンロードするには、「Download Signed Patch (*n bytes*)」ボタンをクリックします。
 - 署名なしパッチをダウンロードするには、「Download Patch (*n bytes*)」ボタンをクリックします。

1 つまたは複数のパッチのダウンロードに成功したら、Web ブラウザを閉じます。
- 5 ダウンロードされたパッチが格納されているディレクトリに移動します。
- 6 スーパーユーザーになるか、同等の役割を引き受けます。
- 7 (署名なしパッチ) 署名なしパッチをダウンロードした場合、パッチを解凍します。

```
# unzip patch-id
```
- 8 署名付きまたは署名なしのパッチを適用します。

- 署名付きパッチをダウンロードした場合、それを適用します。
次に例を示します。

```
# patchadd /tmp/111879-01.jar
```

- 署名なしパッチをダウンロードした場合、それを適用します。
次に例を示します。

```
# patchadd /tmp/111879-01
```

- 9 パッチの適用に成功したことを確認します。
次に例を示します。

```
# patchadd -p | grep 111879
```

```
Patch: 111879-01 Obsoletes: Requires: Incompatibles: Packages: SUNWwsr
```

▼ Solaris パッチの情報を表示する方法

パッチの適用に先立ち、以前に適用されたパッチの詳細を確認することをお勧めします。

次のコマンドは、すでにシステムに適用されているパッチに関する有用な情報を提供します。

- `patchadd -p or showrev -p`
システムに適用されたすべてのパッチを表示します。
- `pkgparam pkgid PATCHLIST`
pkgid (SUNWadmap など) によって識別されるパッケージに適用されたすべてのパッチを表示します。
- `patchadd -S Solaris-OS - p`
特定の OS サーバーに適用されたすべての `/usr` パッチを表示します。

- システムに適用されたパッチに関する情報を表示するには、次のいずれかの `patchadd` コマンド行を使用します。

- システムに適用されたすべてのパッチに関する情報を取得するには、次のように入力します。

```
$ patchadd -p
```

- 特定のパッチがシステムに適用されているかどうかを確認するには、たとえば次のように入力します。

```
$ patchadd -p | grep 111879
```

▼ patchrm コマンドを使用して **Solaris** パッチを削除する方法

1 スーパーユーザーになります。

2 パッチを削除します。

```
# patchrm 111879-01
Checking installed patches...

Backing out patch 111879-01...

Patch 111879-01 has been backed out.
```

3 パッチが削除されたことを確認します。

```
# patchadd -p | grep 111879
#
```



付 録 A

SMF サービス

次の表に、SMFを使用するように変換されたサービスをいくつか示します。サービスごとに、デーモンまたはサービスの名前、サービスのFMRI、サービスを開始するための実行スクリプト、およびサービスが `inetd` によって開始されるかどうかを示します。

表 A-1 SMF サービス

サービス名	FMRI	実行スクリプト	inetd サービス
automount	svc:/system/filesystem/autofs:default	autofs	該当しない
consadm	svc:/system/consadm:default	rootusr	該当しない
coreadm	svc:/system/coreadm:default	coreadm	該当しない
cron	svc:/system/cron:default	cron	該当しない
cryptoadm	svc:/system/cryptosvc:default	なし	該当しない
cvcd	svc:/system/cvc:default	cvcd	該当しない
dcs	svc:/platform/<arch>/dcs:default	なし	該当する
dtlogin	svc:/application/graphical-login/cde-login:default	dtlogin	該当しない
dtprintinfo	svc:/application/cde-printinfo:default	dtlogin	該当しない
dtspcd	svc:/network/cde-spc:default	なし	該当する
dumpadm	svc:/system/dumpadm:default	savecore	該当しない
efdaemon	svc:/platform/<arch>/efdaemon:default	efcode	該当しない
fmd	svc:/system/fmd:default	なし	該当しない
gssd	svc:/network/rpc/gss:default	なし	該当する

表 A-1 SMF サービス (続き)

サービス名	FMRI	実行スクリプト	inetd サービス
imapd	svc:/network/imap/tcp:default svc:/network/imapnew/tcp:default	なし	該当する
in.chargend	svc:/network/chargen:dgram svc:/network/chargen:stream	なし	該当する
in.comsat	svc:/network/comsat:default	なし	該当する
in.daytimed	svc:/network/daytime:dgram svc:/network/daytime:stream	なし	該当する
in.dhcpd	svc:/network/dhcp-server:default	dhcp	該当しない
in.discardd	svc:/network/discard:dgram svc:/network/discard:stream	なし	該当する
in.echod	svc:/network/echo:dgram svc:/network/echo:stream	なし	該当する
in.fingerd	svc:/network/finger:default	なし	該当する
in.ftpd	svc:/network/ftp:default	なし	該当する
in.named	svc:/network/dns/server:default	inetsvc	該当しない
in.rarpd	svc:/network/rarp:default	boot.server	該当しない
in.rdisc	svc:/network/initial:default	inetinit	該当しない
in.rexecd	svc:/network/rexec:default	なし	該当する
in.rlogind	svc:/network/login:rlogin svc:/network/login:eklogin svc:/network/login:klogin	なし	該当する
in.routed	svc:/network/initial:default	inetinit	該当しない
in.rshd	svc:/network/shell:default svc:/network/kshell	なし	該当する
in.talkd	svc:/network/talk:default	なし	該当する
in.telnetd	svc:/network/telnet:default	なし	該当する
in.tftpd	svc:/network/tftp/udp6:default	なし	該当する

表 A-1 SMF サービス (続き)

サービス名	FMRI	実行スクリプト	inetd サービス
in.timed	svc:/network/time:dgram svc:/network/time:stream	なし	該当する
in.tnamed	svc:/network/tname:default	なし	該当する
in.uucpd	svc:/network/uucp:default	なし	該当する
inetd-upgrade	svc:/network/inetd-upgrade:default	なし	該当しない
inetd	svc:/network/inetd:default	inetsvc	該当しない
intrd	svc:/system/intrd:default	なし	該当しない
ipop3d	svc:/network/pop3/tcp:default	なし	該当する
kadmind	svc:/network/security/kadmin:default	kdc.master	該当しない
kbd	svc:/system/keymap:default	keymap	該当しない
keyserv	svc:/network/rpc/keyserv:default	rpc	該当しない
kpropd	svc:/network/security/krb5_prop:default	なし	該当する
krb5kdc	svc:/network/security/krb5kdc:default	kdc	該当しない
ktkt_warnd	svc:/network/security/ktkt_warn:default	なし	該当する
ldap_cachemgr	svc:/network/ldap/client:default	ldap.client	該当しない
loadkeys	svc:/system/keymap:default	keymap	該当しない
lockd	svc:/network/nfs/client:default svc:/network/nfs/server:default	nfs.server	該当しない
lp sched および lpshut	svc:/application/print/server:default	lp	該当しない
mdmonitord	svc:/system/mdmonitor:default	svm.sync	該当しない
metainit	svc:/system/metainit:default	svm.init	該当しない
metadevadm	svc:/platform/<arch>/mpxio-upgrade:default	なし	該当しない
mount	svc:/system/filesystem/local:default svc:/system/filesystem/minimal:default svc:/system/filesystem/root:default svc:/system/filesystem/usr:default	nfs.client、rootusr、 standardmounts	該当しない
mountd	svc:/network/nfs/server:default	nfs.server	該当しない

表 A-1 SMF サービス (続き)

サービス名	FMRI	実行スクリプト	inetd サービス
nfsd	svc:/network/nfs/server:default	nfs.server	該当しない
nfsmapid	svc:/network/nfs/client:default svc:/network/nfs/server:default	nfs.server	該当しない
nis_cachemgr	svc:/network/rpc/nisplus:default	rpc	該当しない
nscd	svc:/system/name-service-cache:default	nscd	該当しない
ntpdate	svc:/network/ntp:default	xntpd	該当しない
ocfserv	svc:/network/rpc/ocfserv:default	ocfserv	該当しない
picld	svc:/system/picl:default	picld	該当しない
pmconfig	svc:/system/power:default	power	該当しない
printd	svc:/application/print/cleanup:default	spc	該当しない
quotaon	svc:/system/filesystem/local:default	ufs_quota	該当しない
rcapd	svc:/system/rcap:default	rcapd	該当しない
rpcbind	svc:/network/rpc/bind:default	rpc	該当しない
rpc.bootparamd	svc:/network/rpc/bootparams:default	boot.server	該当しない
rpc.mdcomm	svc:/network/rpc/mdcomm:default	なし	該当する
rpc.metad	svc:/network/rpc/meta:default	なし	該当する
rpc.metamedd	svc:/network/rpc/metamed:default	なし	該当する
rpc.metamhd	svc:/network/rpc/metamh:default	なし	該当する
rpc.nisd	svc:/network/rpc/nisplus:default	rpc	該当しない
rpc.nispasswd	svc:/network/rpc/nisplus:default	rpc	該当しない
rpc.rexd	svc:/network/rpc/rex:default	なし	該当する
rpc.rstatd	svc:/network/rpc/rstat:default	なし	該当する
rpc.rusersd	svc:/network/rpc/rusers:default	なし	該当する
rpc.smserverd	svc:/network/rpc/smserver:default	なし	該当する
rpc.sprayd	svc:/network/rpc/spray:default	なし	該当する
rpc.ttdbserverd	svc:/network/rpc/ttdbserver:tcp	なし	該当する
rpc.walld	svc:/network/rpc/wall:default	なし	該当する

表 A-1 SMF サービス (続き)

サービス名	FMRI	実行スクリプト	inetd サービス
rpc.yppasswdd およ び rpc.yupdated	svc:/network/nis/server:default	rpc	該当しない
rquotad	svc:/network/nfs/rquota:default	なし	該当する
sadc	svc:/system/sar:default	perf	該当しない
savecore	svc:/system/dumpadm:default	savecore	該当しない
sendmail	svc:/network/smtp:sendmail	sendmail	該当しない
sf880drd	svc:/platform/<arch>/sf880drd:default	sf880dr	該当しない
slpd	svc:/network/slp:default	slpd	該当しない
sshd	svc:/network/ssh:default	sshd	該当しない
statd	svc:/network/nfs/client:default svc:/network/nfs/server:default	nfs.server	該当しない
svc.startd	svc:/system/svc/restarter:default	なし	該当しない
syseventd	svc:/system/sysevent:default	devfsadm	該当しない
sysidpm、sysidns、 sysidroot、sysidsys	svc:/system/sysidtool:system	sysid.sys	該当しない
sysidnet	svc:/system/sysidtool:net	sysid.net	該当しない
syslogd	svc:/system/system-log:default	syslog	該当しない
ttymon	svc:/system/console-login:default	inittab	該当しない
utmpd	svc:/system/utmp:default	utmpd	該当しない
vold	svc:/system/filesystem/volfs:default	volmgt	該当しない
xntpd	svc:/network/ntp:default	xntpd	該当しない
ypbind	svc:/network/nis/client:default	rpc	該当しない
ypserv	svc:/network/nis/server:default	rpc	該当しない
ypxfrd	svc:/network/nis/server:default	rpc	該当しない
zoneadm	svc:/system/zones:default	なし	該当しない
なし	svc:/network/loopback:default	network	該当しない
なし	svc:/network/physical:default	network	該当しない

索引

数字・記号

\$ZFS-BOOTFS, ZFS ブートオプション, 213-214

A

ARCH=all パッケージ、失われたものの追加方法、
ディスクレスクライアントのトラブル
シューティング, 183-192
authTypes タグ, Java Web Console, 85

B

banner コマンド (PROM), 229
basedir キーワード (管理ファイル), 417
bin グループ, 94
BIOS
 システム BIOS
 GRUB ブート環境の, 323-324
boot-archive サービスの管理, 309-312
boot-file プロパティ, 変更, 234
bootadm update-archive, ルート (/) ミラー上の
 ブートアーカイブの更新, 314-320
bootadm コマンド, ブートアーカイブの管理に使用,
 309-312
bootfs プールプロパティ, 212
Bourne シェル
 「ユーザー初期設定ファイル」も参照
 基本機能, 112, 113
Break キー, 296

C

CD-ROM デバイス
 マウント済みの CD からソフトウェアを追加する
 例, 460
CDPATH 環境変数, 114
.cshrc ファイル
 カスタマイズ, 112, 119
 説明, 110
C シェル
 環境変数, 113, 114, 117
 基本機能, 112, 113
 シェル (ローカル) 変数, 113, 114
 ユーザー初期設定ファイル, 110, 119
 「ユーザー初期設定ファイル」を参照
 サイト初期設定ファイルを参照する, 112
 作成, 112

D

daemon グループ, 94
dfstab ファイル, ユーザーのホームディレクトリ
 の共有, 132
DHCP, GRUB ベースの PXE ブートの設定, 290
DHCP の設定, GRUB を使用したネットワークから
 のブート, 290-292
DHCP マクロ, GRUB での使用, 290-292

E

eeeprom コマンド
 ブート動作の変更, 236-237
 ブートパラメータの設定に使用する方法
 GRUB, 236-237
eeeprom コマンドを使用したブート動作の設定,
 GRUB ベースのブート, 236-237
env コマンド, 113
/etc/dfs/dfstab ファイル, ユーザーのホーム
 ディレクトリの共有, 132
/etc/init.d ディレクトリ, 398
/etc/inittab ファイル, 377
 エントリの説明, 377
 デフォルトの例, 378
/etc/passwd ファイル, 101
 説明, 101
 フィールド, 102
 復元 (例)
 x86, 288
 復旧
 SPARC, 300
 復旧 (例)
 x86, 348
 ユーザー ID 番号の割り当て, 94
 ユーザーアカウントの削除, 136
/etc/passwd ファイルの修復, x86, 288
/etc/shadow ファイル, 説明, 101
/etc/skel ディレクトリ, 110
/etc/vfstab ファイル, 133
/etc ファイル
 ユーザーアカウント情報, 99
/export/home ファイルシステム, 99

F

findroot コマンド
 GRUB メニューエントリの追加, 248-249
 menu.lst エントリ, 246-248
FMRI, 説明, 368-369

G

GID, 94

GID (続き)

 大きな数値, 95
 定義, 96
 割り当て, 97
groups コマンド, 97
group ファイル
 説明, 102
 フィールド, 104
 ユーザーアカウントの削除, 136
GRUB
 ブート動作の変更
 menu.lst ファイルの編集, 242-245
 複数のオペレーティングシステムのサ
 ポート, 328-329
GRUBClient, GRUB ベースのネットワーク
 ブート, 289-293
GRUB 機能コンポーネント, 326-328
GRUB デバイスの命名規則, 327
GRUB によるブート, 参照情報, 209-210
GRUB の管理, 参照情報, 209-210
GRUB の機能コンポーネント, 326-328
GRUB のコンポーネント, 326-328
GRUB の通常アーカイブ
 ブートアーカイブ
 参照情報, 208-209
GRUB の用語, 324-326
GRUB ブート環境でのカーネル初期化, 324
GRUB ブート環境での複数のオペレーティングシ
 ステム, 328-329
GRUB ブート環境のシステム BIOS, 323-324
GRUB ベースのネットワークブート, 289-293
GRUB ベースのブート
 DHCP マクロについて, 290-292
 参照情報, 209-210
 システムの対話式のブート, 277-279
 システムを実行レベル S でブートする方
 法, 274-276
 破壊されたブートアーカイブを再構築する方
 法, 286-289
 ブート時に GRUB カーネルの動作を変
 更, 240-241, 280-283
 ブート時に GRUB メニューでカーネルの動作
 を変更, 237-238
 フェイルセーフアーカイブのブート, 284-286

GRUB メニュー

- GRUB カーネルの動作を変更, 237-238
- 説明, 241-242
- GRUB メニューエントリの追加, findroot コマンド, 248-249
- GRUB メニューでカーネルの使用法を変更, 240-241, 280-283
- GRUB を使用した x86 システムの対話式のブート, 277-279
- GRUB を使用したシステムのブート、概要, 329-331
- GRUB を使用したネットワークからのブート, 289-293
- DHCP 設定, 290
- GRUB を使用してシステムを実行レベル s でブートする方法, 274-276

H

- halt コマンド, 217
- history 環境変数, 114
- HOME 環境変数, 114
- /home ファイルシステム, ユーザーのホームディレクトリ, 99

I

- ID 番号
 - グループ, 94, 96, 97
 - ユーザー, 94, 135
- inetadm コマンド, 説明, 372
- init コマンド
 - スタンドアロンシステムをシャットダウンする, 222
 - 説明, 217
- init 状態, 「実行レベル」を参照
- IP マクロ, DHCP の設定, 291

J

- Java Web Console
 - authorizing users of applications, 85

Java Web Console (続き)

- authTypes タグの使用, 85
- noaccess ユーザー ID, 70
- アプリケーションの起動, 62
- アプリケーションの登録, 80-81, 81-82
- アプリケーションの登録解除, 81, 82
- アプリケーションの特権, 84
- アプリケーションへのアクセス, 84
- (概要), 58
- コンソールサービスの管理, 64-66
- コンソールサービスの起動, 64
- コンソールサービスの停止, 65
- コンソールサービスの無効化, 66
- コンソールサービスの有効化, 65
- コンソールへのアクセス, 83
- コンソールを実行するユーザー ID の変更, 70
- 状態, 74-77
- セキュリティに関する注意事項, 83
- 設定, 66
- デフォルトロギングとデバッグロギングの違い, 67
- トラブルシューティング, 74
- 内部パスワード, 85
- 配備済みアプリケーションの一覧表示, 78-79
- プロパティ, 74-77
- プロパティの設定, 69-70
- プロパティの変更
 - 監査実装, 68
 - コンソールのセッションタイムアウト, 67
 - ロギングレベル, 67
- ほかのアプリケーションとの互換性, 59
- リファレンス情報, 83-89
- リモートシステムへのアプリケーションのアクセス, 85
- レガシーアプリケーション, 78

K

- Korn シェル
 - 基本機能, 112
 - ユーザー初期設定ファイル, 110

L

L1 + Aキー, 296
LANG 環境変数, 114, 117
LC 環境変数, 117
LK パスワード, 135
local.cshrc ファイル, 110
local.login ファイル, 110
local.profile ファイル, 110
locale 環境変数, 114
.login ファイル
 カスタマイズ, 112, 119
 説明, 110
LOGNAME 環境変数, 114
LPDEST 環境変数, 115
-L オプション
 ZFS ブートオプション
 使用可能な BE の表示, 212-213
-L ブートオプション, SPARC プラットフォームで
 の ZFS ルートファイルシステムの
 ブート, 259-264

M

MAIL 環境変数, 114, 115
MANPATH 環境変数, 115
menu.lst, GRUB コンポーネント, 326-328
menu.lst ファイル
 findroot コマンドを使用するエントリの追
 加, 248-249
 場所, 242
 ブート時の対話処理
 説明, 241-242
 ブート動作の変更, 242-245
 マルチブート実装, 330-331
menu.lst ファイルの編集, ブート動作の変
 更, 242-245

N

newgrp コマンド, 97
NIS
 ユーザーアカウント, 99, 101

NIS+
 グループ, 97
 ユーザーアカウント, 136
noaccess ユーザー/グループ, 94
 Java Web Console, 70
noask_pkgadd 管理ファイル, 418, 462
nobody ユーザー/グループ, 94

O

OS サーバー, 説明, 155

P

passwd ファイル, 101
passwd ファイル, フィールド, 102
passwd ファイル
 復元 (例)
 x86, 288
 復旧
 SPARC, 300
 復旧 (例)
 x86, 348
 ユーザー ID 番号の割り当て, 94
 ユーザーアカウントの削除, 136
patchadd コマンド, による作業, 479-486
PATH 環境変数
 設定, 116
 説明, 115, 116
path シェル変数, 113
pkgadd コマンド
 -a オプション (管理ファイル), 418, 419, 459, 462
 -d オプション (デバイス名), 459, 463, 464
 -s オプション (スプールディレクトリ), 463,
 464
 概要, 414
 使用時の要件, 416
 スプールディレクトリ, 463
 スプールディレクトリ (例), 464
 代替ベースディレクトリ, 419
 パッケージの追加 (手順), 459
 HTTP URL の使用, 462
 ユーザーの対話操作の省略, 418, 419

pkgadm コマンド, 概要, 414
 pkgchk コマンド
 概要, 414
 使用(例), 467
 pkginfo コマンド
 概要, 414, 416
 使用方法, 465
 すべてのインストール済みパッケージの表示
 (例), 465
 pkgparam コマンド, 概要, 414
 pkgrm コマンド
 rm コマンド(比較), 416
 概要, 414
 使用時の要件, 416
 注意, 416
 pkgtrans コマンド, 概要, 414
 /pkg ディレクトリ, 464
 poweroff コマンド, 217
 prodreg コマンド, 概要, 414
 .profile ファイル
 カスタマイズ, 112, 119
 説明, 110
 PROM, PROM リビジョンの確認, 229
 prompt シェル変数, 115
 PS1 環境変数, 115
 PXEclient, GRUB ベースのネットワーク
 ブート, 289-293

R

reboot コマンド, 217
 removef コマンド, 416
 reset コマンド, 234
 root パスワードの復元(手順), SPARC, 300
 root パスワード、忘れる, SPARC, 300
 root パスワードを忘れる, SPARC, 300

S

setenv コマンド, 113, 114
 set コマンド, 113
 shadow ファイル, 説明, 101
 shadow ファイル, フィールド, 103

SHELL 環境変数, 115
 shutdown コマンド
 サーバーのシャットダウン, 198
 サーバーのシャットダウン(方法), 219
 説明, 217
 ユーザーに通知する, 217
 /skel ディレクトリ, 110
 smcwebserver コマンド, Java Web Console, 60
 SMF
 委任リスタータ, 373
 概要, 365
 コマンド, 372
 ライブラリインタフェース, 372-373
 smreg コマンド
 Java Web Console, 60, 81
 SMV ミラー化ルート(/) メタデバイス, ブート
 アーカイブの更新, 314-320
 Solaris Device Configuration Assistant (デバイス構成
 用補助), 概要, 345-346
 Solaris OS での GRUB の実装, 329-331
 Solaris 管理コンソール
 RBAC と組み合わせて使用, 40
 起動(方法), 45
 使用する理由, 34
 説明, 31
 ツールの説明, 32
 Solaris 管理コンソールのナビゲーション区画,
 ノード, 35
 Solaris ブートアーカイブ
 フェイルセーフと通常
 参照情報, 208-209
 Solaris ブート動作, 管理方法, 227-249
 SPARC ブートオプション, ZFS ルートファイルシ
 ステムからのブート, 212-213
 SPARC プラットフォームでの ZFS ブート, 使用さ
 れるブートオプション, 259-264
 staff グループ, 97
 stage2, GRUB コンポーネント, 326-328
 Stop + A キー, 296
 stty コマンド, 117
 Sun Java Web Console, 57
 Java Web Console からのアプリケーションの登録
 解除, 81
 Java Web Console サービスの管理, 64-66

Java Web Console のコマンド

smcwebserver, 60

smreg, 60

wcadmin, 60

Java Web Console の設定, 66

Java Web Console のプロパティの変更, 監査実装
の選択, 68

SUNW 接頭辞, 416

Sun ソフトウェアパッケージ

インストール, 462

追加(例), 460

svc.startd デーモン, 説明, 373

svcadm コマンド, 説明, 372

svccfg コマンド, 説明, 372

svccprop コマンド, 説明, 372

svcs コマンド, 説明, 372

sync コマンド, 297

T

TERMINFO 環境変数, 115

TERM 環境変数, 115

ttys (擬似), 94

ttytype 擬似ユーザーログイン, 94

TZ 環境変数, 115

U

UID, 135

大きな数値, 95

定義, 94

割り当て, 94

umask コマンド, 118

UNIX グループ, 96

uucp グループ, 94

V

/var/sadm/install/admin ディレクトリ, 418

/var/sadm/patch ディレクトリ, 475

/var/spool/pkg ディレクトリ, 462, 464

W

wcadmin コマンド, Java Web Console, 60

Web ベースのシステム管理アプリケーション, Java
Web Console, 58

who コマンド, 218, 376

X

x86 ブートオプション, ZFS ルートファイルシステ
ムからのブート, 213-214

Z

ZFS ルートファイルシステム, SPARC プラット
フォームでのブート, 259-264

ZFS ルートファイルシステムからのブート

SPARC ブートオプション, 212-213

SPARC プラットフォーム, 259-264

x86 ブートオプション, 213-214

-Z オプション, ZFS ブートオプション, 212-213

-Z ブートオプション, SPARC プラットフォームで
の ZFS ルートファイルシステムの
ブート, 259-264

あ

アーカイブ

Solaris のフェイルセーフと一次

説明, 208-209

フェイルセーフアーカイブのブート

GRUB, 284-286

アクセス権, 118

アクティブな menu.lst ファイル, 場所, 242

アクティブな menu.lst ファイルの場所, 242

アプライアンス, 定義, 150

アプリケーションの起動, Java Web Console の起動
ページ, 62

アプリケーションの特権, Java Web Console, 84

アプリケーションへのアクセス, Java Web
Console, 84

暗号化, 101

い

依存記述 (SMF), 説明, 366
一次グループ, 97
委任リスタータ (SMF), 373

う

失われた ARCH=all パッケージの追加
(例), 186-187

か

カーネルの動作, GRUB メニューで変更, 237-238
鍵、ユーザー、「ユーザー鍵」を参照
確認

pkginfo コマンドによるソフトウェア
パッケージのインストール, 463
システムの実行レベル (方法), 376
ソフトウェアパッケージのインストール
pkginfo コマンド, 463

環境変数

LOGNAME, 114
LPDEST, 115
PATH, 115, 116
SHELL, 115
TZ, 115
説明, 113, 117

監査イベント, Java Web Console, 68
監査実装, Java Web Console, 68
管理ファイル, キーワード, 417

き

キーストア, 410
擬似 ttys, 94
擬似ユーザーログイン, 94
共有, ユーザーのホームディレクトリ (方法), 131

く

クライアントマクロ, DHCP の設定, 291

クラスマクロ, DHCP の設定, 291

グループ

ID 番号, 94, 96, 97
UNIX, 96
一次, 97
一次グループの変更, 97
管理用のガイドライン, 96, 97
情報の格納, 102, 104
説明, 96
デフォルト, 97
名前
説明, 96
名前の説明, 96
二次, 97
ネームサービス, 97
ユーザーが所属しているグループの表示, 97
グループ ID 番号, 94, 96, 97

け

検査

インストール済みのパッケージ (例), 467
ソフトウェアインストール (例), 467

こ

構成リポジトリ (SMF), 「リポジトリ」を参照
コマンド (SMF), 一覧, 372
コンソールのセッションタイムアウト, Java Web
Console のプロパティの変更, 67
コンソールへのアクセス, Java Web Console, 83

さ

サーバー, OS サーバー, 155
サービス (SMF), 説明, 367
サービス管理機能
「SMF」を参照
サービス構成リポジトリ, 「リポジトリ」を参照
サービスの起動と停止, 397
サービスの状態, 説明, 369

最小

- ユーザーのパスワード長, 98
- ユーザーのログイン名の長さ, 100

最大

- ユーザーのパスワード長, 98
- ユーザーのログイン名の長さ, 100

最大数, ユーザーが所属できる二次グループ, 97

最大値, ユーザー ID 番号, 94

サイト初期設定ファイル, 111

削除

- 管理ファイルによるパッケージの削除, 419
- ソフトウェアパッケージ
 - ガイドライン, 416
- ディスクレスクライアントの OS サービス (方法), 179
- ディスクレスクライアントの OS サービス (例), 179
- パッチ
 - patchrm の使用, 486
- ユーザーのホームディレクトリ, 136
- ユーザーのメールボックス, 136

参照情報, GRUB の管理, 209-210

し

シェル

- 環境, 113
- 環境変数, 113, 114, 117
- 基本機能, 112, 113
- ユーザー初期設定ファイル, 119
- ローカル変数, 113, 114

シェル変数, 114

時間帯の環境変数, 115

システムアカウント, 94

システムシャットダウンコマンド, 217

システム初期設定ファイル, 100

システムタイプ

- アプライアンス, 150
- 概要, 148
- スタンドアロンシステム, 149
- 選択のガイドライン, 150
- ディスクレスクライアント, 150

システムのダウン時間をユーザーに通知する, 218

実行制御スクリプト

- サービスの起動と停止, 397
- 追加 (方法), 398
- 無効にする (方法), 399

実行レベル

- 0 (電源切断レベル), 375
- 1 (シングルユーザーレベル), 375
- 2 (マルチユーザーレベル), 375
- 3 (NFS を使用したマルチユーザー)
 - システムが実行レベル 3 になると実行される処理, 378
 - ブート, 253, 273, 336
- 3 (NFS を持つマルチユーザー), 375
- 3 (マルチユーザーと NFS)
 - ブート, 177
- 6 (リブートレベル), 376
- s または S (シングルユーザー状態)
 - ブート, 254
- s または S (シングルユーザーレベル), 375
 - ブート, 339
- 確認 (方法), 376
- 定義, 375
- デフォルトの実行レベル, 375

実行レベル S でのシステムのブート

GRUB ベースのブート

方法, 274-276

自動マウント, ユーザーのホームディレクトリ, 100

シャットダウン

- shutdown と init の各コマンドでシステムを正常にシャットダウンする, 216
- システム、ガイドライン, 198

修復/etc/passwd ファイル

SPARC, 300

x86, 348

障害管理リソース識別子, 「FMRI」を参照

使用可能な BE のリストの表示

ZFS ルートのブート

boot -L オプション, 212-213

証明書、信頼される、定義, 410

初期設定ファイル、システム, 100

署名付きパッチ, 474

「パッチ」も参照

使用する場合, 477

署名なしパッチ, 474
 使用する場合, 477
 新機能, SMF, 365
 シングルサインオン、安全な https ポート, Java
 Web Console, 59
 シングルユーザーレベル、「実行レベル s または
 S」を参照

す

スーパーユーザー (root) パスワード、忘れる、
 SPARC, 300
 スケルトンディレクトリ (/etc/skel), 110
 スタンドアロンシステム, 定義, 149
 スナップショット (SMF), 説明, 371-372
 スプールディレクトリ
 ソフトウェアパッケージのインストール (方
 法), 463
 ソフトウェアパッケージのインストール
 (例), 464, 467

せ

制限, パッケージおよびパッチの追加および削
 除, 417
 正常なシャットダウン, 216
 セキュリティー, ユーザー ID 番号の再利用, 94
 セキュリティーに関する注意事項, Java Web
 Console, 83
 セッションタイムアウト時間, Java Web Console の
 プロパティーの変更, 70
 戦略, Solaris パッチ使用時, 474

そ

ゾーン, パッケージおよびパッチの追加および削
 除に関する制限, 417
 ソフトウェア管理
 ツール, 414
 パッケージ, 410
 パッケージの命名規則, 416

ソフトウェアパッケージ
 インストール, 464
 スプールディレクトリからのインストール
 (例), 464

た

対話式のブート, GRUB を使用した x86 システムの
 ブート, 277-279
 ダウンロード
 パッチ
 patchadd の使用, 484

つ

追加
 管理ファイルによるパッケージの追加, 418
 実行制御スクリプト (方法), 398
 スプールディレクトリからのパッケージの追加
 (例), 464
 スプールディレクトリへのパッケージの追加
 (例), 467
 ディスクレスクライアントの OS サービス (方
 法), 168
 パッケージ (要件), 416
 パッケージ, 例, 461
 複数のバージョンのパッケージ, 417
 マウント済みの CD からパッケージを追加する
 (例), 460
 ユーザー初期設定ファイル, 110
 リモートパッケージサーバーからパッケージを
 追加する (例), 462

て

停止
 復旧目的のシステム停止 (手順)
 x86, 302, 346
 復旧を目的としたシステムの停止
 SPARC, 296
 ディスクレスクライアント
 OS サービスの削除 (方法), 179

ディスクレスクライアント (続き)

OS サービスの削除 (例), 179

OS サービスの追加 (方法), 168

定義, 150

パッチの適用, 473

ブート (方法), 177

ディスクレスクライアントのインストールの問題

のトラブルシューティング, 失われた ARCH=all

パッケージの追加方法, 183-192

ディスクレスクライアントの管理コマンド

smosservice

OS サービスの追加, 155

ディスクレスクライアントのトラブル

シューティング, 失われた ARCH=all

パッケージの追加方法, 183-192

ディレクトリ

PATH 環境変数, 115, 116

アクセス権の制御, 118

スケルトン, 110

ベースディレクトリ (basedir), 417

ホーム, 99

ユーザーアカウントの所有権の変更, 135

デジタル署名, 署名付きパッチ, 474

デバイスの命名規則

GRUB, 327

と

トラブルシューティング

64 ビットブートの失敗, 354

Java Web Console, 74

な

名前

SUNW 接頭辞, 416

グループ

説明, 96

ソフトウェアパッケージの命名規則, 416

ユーザーログイン

説明, 93

変更, 135

に

二次グループ, 97

ね

ネームサービス

グループ, 97

ユーザーアカウント, 99, 101

ネットワークブート, GRUB を使用, 289-293

ネットワークマクロ, DHCP の設定, 291

の

ノード, Solaris 管理コンソールのナビゲーション
区画, 35

は

破壊されたブートアーカイブ, 再構築方
法, 286-289破壊されたブートアーカイブの再構築 (方
法), 286-289

パスワード (ユーザー)

LK パスワード, 135

暗号化, 101

設定, 98, 138

説明, 98, 138

選択, 98

注意事項, 98

変更

頻度, 98

ユーザーツール, 138

ユーザーによる, 98

有効期限, 101, 138

ユーザーアカウントの無効化/ロック, 135

ユーザーツール, 138

パッケージ

概要, 410

署名付き

「パッケージ、署名付き」を参照

追加

「pkgadd コマンド」も参照

パッケージ (続き)

定義, 410

パッケージおよびパッチの追加および削除
制限

ゾーン, 417

パッケージ、署名付き, 概要, 410

パッチ

Solaris パッチへのアクセス, 475-476

管理, 476

使用される用語, 477-479

情報の表示, 485

署名付き, 474

適用, 410

署名なし, 474

ダウンロード

patchadd の使用, 484

定義, 474

パッチの README ファイル, 475

番号付けの方式, 475

パッチ管理ツール, 作業マップ, 476

パッチ関連の用語集, 477-479

パッチ関連用語の定義, 477-479

パッチの一覧

表示

patchadd による, 485

パッチの適用

patchadd の使用, 483-485

ディスクリスクライアント, 473

ひ

表示

インストール済みソフトウェアの情報, 465

環境変数, 113

パッケージの情報 (例), 465

パッチの一覧

patchadd による, 485

ユーザーマスク, 118

ふ

ファイル

アクセス権の制御, 118

ファイル (続き)

新しくインストールされたパッケージの属性の
検査, 467

ユーザーアカウントの所有権の変更, 135

ファイルシステムと sync コマンドとの同期
化, 297ファイルとディレクトリのアクセス権の制
御, 118

ブート

64 ビット対応の x86 システムを 32 ビット
モードでブートする (例), 351Solaris Device Configuration Assistant (デバイス構
成用補助) (手順)

x86, 346

システム、ガイドライン, 199

実行レベル S

SPARC, 254

対話式 (方法)

SPARC, 256

ディスクリスクライアント (方法), 177

ブートアーカイブ

管理, 307-322

種類, 208-209

破壊されたブートアーカイブを再構築する方
法, 286-289ブートアーカイブ、管理、フェイルセーフアーカ
イブのブート, 307-308

ブートアーカイブの管理, 手順, 307-322

ブートアーカイブの更新, ミラー化ルート

パーティション, 314-320

ブートアーカイブの手動更新, ミラー化ルート (/)

パーティションのあるシステム, 314-320

ブートオプション

-L

ZFS ルートファイルシステム, 259-264

-Z

ZFS ルートファイルシステム, 259-264

ブート時の対話処理, GRUB メニュー, 241-242

ブート動作

GRUB menu.lst ファイルの編集

方法, 242-245

GRUB メニューでの変更方法, 240-241, 280-283

管理, 227-249

ブート動作、x86 システムでの変更, 235-249

- ブート動作の管理, 227-249
- ブート動作の変更
 - GRUB menu.lst ファイルの編集
 - 方法, 242-245
 - ブート時に GRUB メニューで, 236-237
- ブート動作の変更 (作業マップ), 235-249
- ブートプロセス, x86, 361
- ブートプロパティーの変更, 234
- フェイルセーフアーカイブ
 - GRUB の参照情報
 - 説明, 208-209
 - GRUB ベースのブート
 - 復元, 284-286
 - SPARC システムでのブート, 265-269
 - ブート, 307-308
- フェイルセーフアーカイブのブート
 - GRUB ベースのブート, 284-286
 - SPARC システム, 265-269
 - 破壊されたブートアーカイブを再構築するため, 286-289
- 復元
 - フェイルセーフアーカイブのブート
 - GRUB ベースのブート, 284-286
- 複数のバージョンのソフトウェア
 - パッケージ, 417, 419
- プライマリ管理者の役割
 - 権利の付与, 42
 - 作成 (概要), 42
- プロパティー, boot-file プロパティーの変更, 234
- プロファイル (SMF), 説明, 370
- へ
- ベースディレクトリ (basedir), 417, 419
- 別名, ユーザーログイン名との違い, 93
- 変更
 - Java Web Console のプロパティー
 - セッションタイムアウト時間, 70
 - ユーザー ID 番号, 135
 - ユーザーアカウントのディレクトリ所有者, 135
 - ユーザーアカウントのファイル所有権, 135
 - ユーザーのパスワード
 - 頻度, 98
 - 変更, ユーザーのパスワード (続き)
 - ユーザーによる, 98
 - ユーザーパスワード
 - ユーザーツール, 138
 - ユーザーログイン名, 135
- 変数
 - 環境, 113, 117
 - シェル (ローカル), 113
- ほ
- ほかのアプリケーションとの互換性, Java Web Console, 59
- ま
- マウント
 - ユーザーのホームディレクトリ
 - 自動マウント, 100
 - ユーザーのホームディレクトリ (方法), 133
- マクロ, DHCP, 290-292
- マルチブート実装, menu.lst ファイルの説明, 330-331
- マルチユーザーレベル, 「実行レベル 3」を参照
- み
- ミラー化ルート (/) パーティション, ブートアーカイブの更新, 314-320
- む
- 無効化
 - ユーザーアカウント
 - パスワード, 135
 - ユーザーツール, 135
- 無効にする, 実行制御スクリプト (方法), 399

め

メール別名 (エイリアス), ユーザーログイン名との違い, 93

メニュー

GRUB

説明, 241-242

も

目録 (SMF), 説明, 370

ゆ

ユーザー ID 番号, 94, 135

ユーザーアカウント, 92

ID 番号, 94, 135

ガイドライン, 100

情報の格納, 99

設定

情報シート, 124

説明, 92, 93

ネームサービス, 99, 101

無効化/ロック

パスワード, 135

ユーザーツール, 135

ログイン名, 93, 135

ユーザー鍵, 410

ユーザー初期設定ファイル

カスタマイズ, 110, 119

概要, 111

カスタマイズされたファイルの追加, 110

環境変数, 114, 117

サイト初期設定ファイル, 111

シェル変数, 114, 115

ユーザーマスクの設定, 118

ローカルシステムへの参照を避ける, 112

シェル, 112, 119

説明, 99, 100, 110

デフォルト, 110

例, 119

ユーザーツール

アカウントの無効化, 135

パスワード管理, 138

ユーザーのパスワードの有効期限, 138

ユーザーのホームディレクトリ

カスタマイズされた初期設定ファイル, 110

共有 (方法), 131

削除, 136

所有者の変更, 135

説明, 99

非ローカル参照 (\$HOME), 99, 112

マウント

自動マウント, 100

マウント (方法), 133

ユーザーパスワードの有効期限, 101

ユーザーマスク, 118

ユーザーログイン (擬似), 94

ユーザーログイン名

説明, 93

変更, 135

よ

用語, GRUB, 324-326

ら

ライブラリインタフェース, SMF, 372-373

り

リスタータ (SMF), 373

説明, 366

リセット, SPARC システム, 234

リポジトリ (SMF)

説明, 366, 371

リモートシステムへのアプリケーションのアクセス, Java Web Console, 85

リモートパッケージサーバー

スプールディレクトリへのパッケージの追加 (例), 464

ソフトウェアのインストール, 462

ソフトウェアのインストール (例), 461

れ

レガシーアプリケーション, Java Web Console, 78

ろ

ログインレベルの選択, Java Web Console のプロパ
ティーの変更, 67

ログイン名 (ユーザー)

説明, 93

変更, 135