



Solaris 8 のシステム管理 (追補)

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303
U.S.A. 650-960-1300

Part Number 816-0113-10
2001 年 5 月

Copyright 2001 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A. All rights reserved.

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

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

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

本製品に含まれる HG 明朝 L と HG ゴシック B は、株式会社リコーがリョーベイマジクス株式会社からライセンス供与されたタイプフェイスマスタをもとに作成されたものです。平成明朝体 W3 は、株式会社リコーが財団法人 日本規格協会 文字フォント開発・普及センターからライセンス供与されたタイプフェイスマスタをもとに作成されたものです。また、HG 明朝 L と HG ゴシック B の補助漢字部分は、平成明朝体 W3 の補助漢字を使用しています。なお、フォントとして無断複製することは禁止されています。

Sun、Sun Microsystems、docs.sun.com、AnswerBook、AnswerBook2、iPlanet、Solstice AdminSuite、Solaris Management Console、Sun Blade、Sun Ray、Sun StorEdge は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

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

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

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

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

Wnn6 は、オムロン株式会社で開発されたソフトウェアです。(Copyright OMRON Co., Ltd. 1999 All Rights Reserved.)

「ATOK」は、株式会社ジャストシステムの登録商標です。

「ATOK8」は株式会社ジャストシステムの著作物であり、「ATOK8」にかかる著作権その他の権利は、すべて株式会社ジャストシステムに帰属します。

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

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

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

Unicode は、Unicode, Inc. の商標です。

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

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

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

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、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ぬものとします。

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

原典: Solaris 8 System Administration Supplement

Part No: 806-7502-10

Revision A



目次

- はじめに 9
- 1. 新規機能の概要 13
- 2. システム管理ツールを使用した管理についてのトピック 21
- 3. **Solaris Management Console** の管理 23
 - Solaris Management Console の概要 23
 - Solaris Management Console の起動 25
 - ▼ コンソールをコマンド行から起動するには 25
 - ▼ SMC ツールボックスエディタを起動するには 26
 - ▼ SMC サーバーが動作しているか判断するには 26
 - ▼ SMC サーバーを起動するには 27
 - ▼ SMC サーバーを停止するには 27
- 4. **WBEM** でのデスクトップ、デバイス、ネットワークの管理 29
 - 『Solaris WBEM Services の管理』の更新 29
- 5. セキュリティの管理についてのトピック 31
- 6. **RBAC** でのセキュリティの管理 33
 - 役割によるアクセス制御 33
- 7. **GSS-API** でのデータの転送 37
 - 『GSS-API のプログラミング』 37

- 8. スマートカードでのセキュリティの管理 39
 - SPARC: 『Solaris スマートカードの管理』の更新 39
- 9. サーバーとクライアントの管理についてのトピック 41
- 10. ディスクレスクライアントの管理 43
 - ディスクレスクライアントの管理の概要 43
 - Diskless Client Management の使用 44
 - ユーザーの権利 45
 - 必要なディスクスペース 46
 - ▼ ディスクレスクライアント環境をセットアップするには 46
 - OS サービスを追加する準備 47
 - ▼ OS サービスを追加するには 49
 - ▼ ディスクレスクライアントを追加するには 50
 - OS サービスへのパッチ適用 51
 - トラブルシューティング 52
- 11. ファイルシステムの管理についてのトピック 57
- 12. UFS ソフトウェアでのファイルシステムの管理 59
 - UFS 直接入出力の並行処理の向上 59
- 13. fssnap でのファイルシステムのバックアップ 61
 - UFS スナップショットの概要 61
 - なぜ UFS スナップショットを使用するか 62
 - UFS スナップショットのパフォーマンス上の問題 63
 - UFS スナップショットの作成 63
 - ▼ UFS スナップショットを作成するには 63
 - ▼ UFS スナップショットの情報を表示するには 64
 - UFS スナップショットの削除 65
 - ▼ UFS スナップショットを削除するには 65
 - UFS スナップショットのバックアップ 66
 - ▼ UFS スナップショットのバックアップを行うには 67

- ▼ UFS スナップショットの増分ダンプを作成するには 68
 - UFS スナップショットのバックアップからのデータの復元 69
- 14. リムーバブルメディアの管理についてのトピック 71
- 15. リムーバブルメディアの管理 73
 - リムーバブルメディアの管理の概要 73
 - リムーバブルメディア上の情報へのアクセス 74
 - ▼ リムーバブルメディア上の情報にアクセスするには 75
 - Jaz ドライブおよび Zip ドライブへのアクセス 76
 - リムーバブルメディアのフォーマット (rmformat) 77
 - ▼ リムーバブルメディアをフォーマットするには (rmformat) 78
 - ▼ UFS または UDFS ファイルシステム用にリムーバブルメディアをフォーマットするには 79
 - ▼ PCFS ファイルシステム用にリムーバブルメディアをフォーマットするには 80
 - ▼ リムーバブルメディア上の PCFS ファイルシステムをチェックするには 81
 - ▼ リムーバブルメディア上の不良ブロックを修復するには 82
 - リムーバブルメディアへの読み取りまたは書き込み保護とパスワードによる保護の適用 83
 - ▼ リムーバブルメディアに書き込み保護を有効または無効にするには 83
 - ▼ Iomega 媒体上で読み取りまたは書き込み保護とパスワードを有効または無効にするには 83
- 16. デバイスの管理についてのトピック 87
- 17. **Reconfiguration Coordination Manager (RCM) スクリプト 89**
 - RCM スクリプトの概要 89
 - RCM スクリプトとは 90
 - RCM スクリプトで何ができるか 90
 - RCM スクリプトのプロセスはどのように機能するか 91
 - RCM スクリプトのタスク 92
 - アプリケーション開発者の RCM スクリプトのタスク 92
 - システム管理者の RCM スクリプトのタスク 93

RCM スクリプトの名付け方 94

RCM スクリプトのインストールまたは削除 94

▼ RCM スクリプトをインストールするには 95

▼ RCM スクリプトを削除するには 95

▼ RCM スクリプトをテストするには 96

テープのバックアップ用 RCM スクリプトの例 97

テープのバックアップ用 RCM スクリプトは何を行うか 97

テープバックアップ再構成のシナリオの結果 98

例 - テープのバックアップ用の RCM スクリプト 99

18. USB デバイスの管理 103

USB デバイスの概要 103

よく使用される USB 関連の略語 104

USB バスの説明 105

Solaris 環境における USB について 107

USB キーボードとマウス 107

USB ホストコントローラとルートハブ 109

USB ストレージデバイス 109

volld を実行している状態で USB 大容量ストレージデバイスを管理する 110

▼ volld を実行している状態で USB 大容量ストレージデバイスをマウントまたはマウント解除するには 111

▼ volld の実行中にホットプラグ可能な USB 大容量ストレージデバイスを取り外すには 112

▼ volld の実行中にホットプラグ可能な USB 大容量ストレージデバイスを追加するには 113

volld を使用せずに USB 大容量ストレージデバイスを管理する 113

▼ volld を使用せずに USB 大容量ストレージデバイスをマウントまたはマウント解除するには 114

▼ volld を使用せずにホットプラグ可能な USB 大容量ストレージデバイスを取り外すには 115

- ▼ vold を使用せずにホットプラグ可能な USB 大容量ストレージデバイスを追加するには 115
 - SPARC: USB CD 上にデータ作成するまたは USB CD からデータを取得するには 116
- ▼ USB CD 上へデータを作成するまたは USB CD からデータを取得するための準備を行うには 116
 - SPARC のみ: USB 電源管理 117
 - USB デバイスのホットプラグ 117
 - USB ケーブル 118
- USB プリンタのサポート 118
- 19. 動的再構成に関する問題のトラブルシューティング 121
 - 新しい動的再構成のエラーメッセージ 121
- 20. ネットワークの管理についてのトピック 125
- 21. メールサービス 127
 - sendmail に関する情報のその他の参照先 128
 - sendmail バージョン 8.9.3 からの変更点 128
 - 新しいコマンド行オプション 129
 - 新規および更新された構成ファイルオプションとその関連事項 129
 - 新たに定義された sendmail のマクロ 142
 - sendmail 構成ファイルの構築に使用される新しいマクロ 143
 - sendmail の新規および更新された m4 構成マクロとその関連事項 144
 - sendmail の新しいコンパイルフラグ 148
 - 新しい配信エージェントフラグ 149
 - 配信エージェントの新しい等式 150
 - 新しい待ち行列機能 151
 - sendmail における LDAP の新しい使い方 152
 - 新しい組み込み型のメールプログラム機能 153
 - 新しいルールセット機能 154
 - 新しいファイルのロケーション 155

mail.local における変更点	156
mailstats における変更点	157
makemap における変更点	157
関連するその他の変更点や機能	158
22. Berkeley Internet Name Domain (BIND) バージョン 8.1.2 から BIND バージョン 8.2.2 への移行 (パッチレベル 5)	161
BIND の更新	161
23. IP ネットワークマルチパス	163
ネットワークアダプタの切り離し	163
24. モバイル IP の管理	165
逆方向トンネリングと専用アドレス	165
25. システムのリソース管理についてのトピック	167
26. システムアカウントिंगでのリソースの管理	169
アカウントिंग機能の拡張	169
27. システムのパフォーマンス管理についてのトピック	171
28. DNLC でのシステムパフォーマンスの向上	173
DNLC 機能の向上	173
29. パフォーマンス向上のためのシステムのチューニング	177
『Solaris カーネルのチューンアップ・リファレンスマニュアル』の変更点	177

はじめに

『Solaris 8 のシステム管理 (追補)』では、Solaris™ 8 Update リリースで更新された機能について説明します。ここでの説明は、すでにリリースされている Solaris 8 のマニュアルセットの内容を補足または変更するものです。Solaris のマニュアルは、Solaris 8 の DOCUMENTATION CD に含まれています。

注 - Solaris オペレーティング環境は、2 種類のハードウェア (プラットフォーム) 上で動作します。つまり、SPARC™ と IA (Intel アーキテクチャ) です。Solaris オペレーティング環境は、64 ビットと 32 ビットの両方のアドレス空間で動作し、IA では 32 ビットのアドレス空間でのみ動作します。このマニュアルで説明する情報は、章、節、注、箇条書き、図、表、例、またはコード例において特に明記しない限り、両方のプラットフォームおよびアドレス空間に該当します。

Sun のマニュアルの注文方法

専門書を扱うインターネットの書店 Fatbrain.com から、米国 Sun Microsystems™, Inc. (以降、Sun™ とします) のマニュアルをご注文いただけます。

マニュアルのリストと注文方法については、<http://www1.fatbrain.com/documentation/sun> の Sun Documentation Center をご覧ください。

Sun のオンラインマニュアル

<http://docs.sun.com> では、Sun が提供しているオンラインマニュアルを参照することができます。マニュアルのタイトルや特定の主題などをキーワードとして、検索を行うこともできます。

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
<code>AaBbCc123</code>	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。	<code>.login</code> ファイルを編集します。 <code>ls -a</code> を使用してすべてのファイルを表示します。 <code>system%</code>
<code>AaBbCc123</code>	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。	<code>system% su</code> <code>password:</code>
<i><code>AaBbCc123</code></i>	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、 <code>rm filename</code> 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。

表 P-1 表記上の規則 続く

字体または記号	意味	例
[]	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`

ただし AnswerBook2™ では、ユーザーが入力する文字と画面上のコンピュータ出力は区別して表示されません。

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、「IA」という用語は、Intel 32 ビットのプロセッサアーキテクチャを意味します。これには、Pentium、Pentium Pro、Pentium II、Pentium II Xeon、Celeron、Pentium III、Pentium III Xeon の各プロセッサ、および AMD、Cyrix が提供する互換マイクロプロセッサチップが含まれます。

新規機能の概要

この章では、Solaris 8 Update リリースに追加された新機能について説明します。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「*Solaris 8 Reference Manual Collection*」には記載されていない新しい情報も提供されています。

説明	更新リリース
システム管理ツールによる管理	
<p>Solaris Management Console™ (SMC) 2.0 は GUI ベースの「傘型アプリケーション」で、各種の管理ツールの起動ポイントとして機能します。SMC は、以下のツールを含むデフォルトのツールボックスとともに使用することで、完全に機能するようになります。</p>	1/01
<ul style="list-style-type: none">■ プロセス - プロセスを中断、再開、監視、および制御します。■ ユーザー - ユーザーのアカウント、ユーザーのテンプレート、グループ、メンバーリスト、管理役割、および権利を、設定し維持します。ともに動作することのできる特定のアプリケーションを管理したり各ユーザーがどのタスクを実行できるかを管理するために、ユーザーや管理役割に対して権利を与えたり拒否したりします。■ スケジュールされたジョブ - ジョブをスケジュール、起動、および管理します。■ マウントと共有 - マウント情報、共有情報、利用情報を表示および管理します。■ ディスク - ディスクパーティションを作成および表示します。■ シリアルポート - 既存のシリアルポートを構成および管理します。■ ログビューア - アプリケーションやコマンド行メッセージを表示し、ログファイルを管理します。	
<p>ディスクレスクライアントも管理することができますが、GUI 形式ではなく、コマンド行からのみ可能です。</p>	
<p>SMC ツールボックスエディタの使用により、デフォルトのツールボックスにツールを追加したり削除したり、また別のツールのセットを管理するために新しいツールボックスを作成したりすることができます。</p>	
<p>コマンド行インタフェースの使用方法についての詳細は 23 ページの「Solaris Management Console の概要」を参照してください。SMC の起動方法についての詳細は 25 ページの「Solaris Management Console の起動」を参照してください。また、各ツールに関連したヘルプも参照してください。</p>	

説明	更新リリース
<p>WBEM (Web-based Enterprise Management) には、さまざまなプラットフォームにおけるシステム、ネットワーク、デバイスの Web ベースでの管理の標準規格が含まれています。この標準化により、システム管理者は、デスクトップ、デバイス、およびネットワークの管理を行うことができます。</p> <ul style="list-style-type: none"> ■ Solaris 8 10/00 Update リリースで、CIM Object Manager が使用するシステムプロパティの記述や、新しい Solaris_Printer およびその他の印刷定義のクラスについての記述などが追加されました。 ■ Solaris 8 1/01 Update リリースで、主に以下の事項が追加されました。 <ul style="list-style-type: none"> ■ <code>init.wbem</code> コマンドが CIM Object Manager だけでなく Solaris Management Console (SMC) サーバーも起動できるようになったことに伴い、<code>init.wbem</code> コマンドの記述が更新されました。 ■ CIM Object Manager のレポジトリをアップグレードする方法についての節が追加されました。 ■ 「セキュリティの管理」の章が更新され、役割によるアクセス制御 (RBAC) 実行のための Solaris Management Console (SMC) について記述されました。 ■ ログファイル情報を参照するためのアプリケーションとして <code>wbemlogviewer</code> が Solaris Management Console (SMC) ログビューアに置き換えられました。 ■ 新しい <code>Solaris_Network1.0.mof</code> ファイルおよび <code>Solaris_Users1.0.mof</code> ファイルの記述が追加されました。 <p>『Solaris WBEM Services の管理』の更新事項のリストは 29 ページの「『Solaris WBEM Services の管理』の更新」、詳細は『Solaris WBEM Services の管理』を参照してください。</p>	<p>10/00。 1/01 で更新</p>
<p>サーバーとクライアントの管理</p>	
<p>Diskless Client Management が Solaris Management Console のコマンド行から利用できるようになりました。ディスクレスクライアントの管理、ディスクレスクライアント用の OS サーバーのリスト表示、およびすべての既存のディスクレスクライアントについてのパッチの管理を行うことができます。</p> <p>ディスクレスクライアントの管理についての詳細は、43 ページの「ディスクレスクライアントの管理の概要」を参照してください。</p>	<p>1/01</p>
<p>システムのセキュリティの管理</p>	
<p>役割によるアクセス制御 (RBAC: role-based access control) データベースが、Solaris Management Console (SMC) グラフィカルインタフェースで管理できるようになりました。1 つの権利に他の権利を含めることもできるようになりました。権利は、デフォルトで <code>policy.conf</code> ファイルで割り当てることができます。</p> <p>詳細は 33 ページの「役割によるアクセス制御」を参照してください。</p>	<p>1/01</p>

説明	更新リリース
<p>Generic Security Service Application Programming Interface (GSS-API) は、セキュリティのフレームワークです。GSS-API を使用すると、アプリケーションは転送するデータを保護できます。GSS-API は認証、整合性、および機密性のサービスをアプリケーションに提供します。このインタフェースを使用すると、各アプリケーションはセキュリティに関して一般的に「汎用」になります。つまり、どのような実際のプラットフォーム (Solaris プラットフォームなど) やセキュリティ機構 (Kerberos など) が使用されるかを知る必要がありません。これは、GSS-API を使用するアプリケーションの移植性が高くなることを意味します。</p> <p>詳細は、『GSS-API のプログラミング』を参照してください。</p>	6/00
<p>SPARC: 『Solaris スマートカードの管理』が更新されました。Solaris 8 1/01 リリースで、内部カードリーダーのセットアップについての情報が追加されました。また、スマートカードのセットアップをより容易にするために、スマートカードサポートのためのセットアップ手順が効率化されています。</p> <p>Solaris 8 4/01 リリースで、以前のバージョンで技術的に不正確であった箇所が修正されました。また、スマートカードのセットアップに必要なタスクと、デフォルトのスマートカードプロパティではセキュリティ環境に充分でない場合に必要となる追加の構成タスクについて記述した、新しい章が追加されました。</p> <p>詳細は、『Solaris スマートカードの管理』を参照してください。</p>	1/01。 4/01 で更新。
ネットワークの管理	
<p>Berkeley Internet Name Domain (BIND) バージョン 8.2.2 の新機能には、以下が含まれます。</p> <ul style="list-style-type: none"> ■ In.named 構成オプション — conf (4) のマニュアルページを参照。 ■ マルチスレッド化されたアプリケーションで安全に使用できる、リゾルバ (3RESOLV) インタフェースへの拡張。 ■ 再構成 in.named の起動と停止に使用される ndc (1M) コマンド、および TSIG と DNSSEC キーの作成に使用される dnskeygen (1M) コマンドの追加。 <p>詳細は、『Solaris ネーミングの管理』を参照してください。</p>	4/01
<p>sendmail の機能が更新されました。新しいコマンド行オプション、新規および更新された構成ファイルのオプション、新しい定義されたマクロ、新規および更新された m4 構成マクロ、新規または更新されたコンパイルフラグ、新しい配信エージェントフラグ、配信エージェント用の新しい等式、新しい待ち行列機能、LDAP の新しい使用方法、新しいルールセットの機能、新しいファイルロケーション、新しい組み込み型のメールプログラム機能などです。第 21 章 ではまた、mail.local の変更点、mailstats の変更点、makemap の変更点も記載しています。</p> <p>詳細は、第 21 章を参照してください。</p>	4/01

説明	更新リリース
<p>IP ネットワークマルチパスでは、ネットワークアダプタにおけるシングルポイントの障害からの復旧機能や、トラフィックのスループットの向上をシステムに提供します。Solaris 8 10/00 リリースから、ネットワークアダプタにおいて障害が発生し、同じ IP リンクに代替アダプタが接続されている場合、システムがすべてのネットワークアクセスを障害の起きたアダプタから代替アダプタへ自動的に切り替えるようになりました。このプロセスにより、ネットワークへのアクセスの中断を防ぐことができます。また、同じ IP リンクに複数のネットワークアダプタが接続されている場合、トラフィックを複数のネットワークアダプタに分散させることにより、トラフィックのスループットが向上します。</p> <p>Solaris 8 4/01 リリースでは、動的再構成 (DR: Dynamic Reconfiguration) で IP ネットワークマルチパスを使用して、IP を使用中のユーザーに影響を及ぼすことなく特定のネットワークデバイスを切り離すことができます。</p> <p>詳細は、『IP ネットワークマルチパスの管理』を参照してください。</p>	<p>10/00。 4/01 で更新。</p>
<p>Mobile Internet Protocol (モバイル IP) を使用すると、モバイルコンピュータ (ラップトップ、無線通信など) 間で情報を転送できます。Solaris 8 6/00 リリースから、モバイルコンピュータは別のネットワークに場所を変更しても、モバイルコンピュータのホームネットワークを通じてアクセスおよび通信できるようになりました。モバイル IP の Solaris の実装では IPv4 だけがサポートされます。</p> <p>Solaris 8 4/01 リリースで、モバイル IP ではシステム管理者が逆方向トンネルのセットアップを行うことが可能になりました。モバイルノードの気付アドレスからホームエージェントへ逆方向トンネルを設定することで、IP データパケットについてトポロジカルに正しいソースアドレスを確保することができます。リバーストンネルの使用により、システム管理者はモバイルノードに専用アドレスを割り当てることもできます。</p> <p>詳細は、『モバイル IP の管理』を参照してください。</p>	<p>6/00。 4/01 で更新。</p>
<p>SPARC: Lightweight Directory Access Protocol (LDAP) が iPlanet™ Web Server ディレクトリサーバーでサポートされるようになりました。Solaris クライアントをサポートするよう iPlanet ディレクトリサーバーをセットアップする方法については、『LDAP の設定と構成』を参照してください。</p>	<p>1/01</p>
<p>ファイルシステムの管理</p>	
<p>UFS 機能の向上: 通常の UFS ファイルへの読み取りアクセスおよび書き込みアクセスの並行処理が可能になり、データベースアプリケーションがバッファ処理されていないファイルシステムデータにアクセスするために使用する直接入出力のパフォーマンスが向上しました。</p> <p>直接入出力の並行処理についての詳細は、59ページの「UFS 直接入出力の並行処理の向上」を参照してください。</p>	<p>1/01</p>

表 1-1 Solaris 8 の新規機能 続く

説明	更新リリース
<p>ファイルシステムのマウント中にファイルシステムのバックアップを行うために、新しい <code>fssnap</code> コマンドが UFS スナップショットで提供されるようになりました。スナップショットは、バックアップ操作のためのファイルシステムの一時的イメージです。これまでは、<code>ufsdump</code> コマンドを使用する際は、バックアップ中にファイルシステムをアクティブでない状態にするためにシステムをシングルユーザーモードにしておくことが推奨されていました。</p> <p>UFS スナップショットについての詳細は、63ページの「UFS スナップショットの作成」を参照してください。</p>	1/01
<p><code>mkfs</code> コマンドが更新され、ファイルシステムを作成する際のパフォーマンスが向上しました。これにより、<code>mkfs</code> のパフォーマンスは、以前の Solaris リリースの 10 倍の速さになることもあります。<code>mkfs</code> コマンドのパフォーマンスの向上は、大規模ファイルシステムと小規模ファイルシステムのどちらの作成時にも見られますが、特に大容量のシステムや高速ディスクで一番顕著に見られます。</p>	1/01
リムーバブルメディアの管理	
<p>リムーバブルメディア管理が、DVD-ROM、Zip ドライブと Jaz ドライブ、CD-ROM、フロッピーディスクなどのリムーバブルメディアを完全にサポートするようになりました。この機能の使い方については、第 15 章を参照してください。</p>	6/00。 10/00 で更新。
デバイスの管理	
<p>Reconfiguration Coordination Manager (RCM) スクリプトにより、Solaris のシステム管理者は、動的再構成の操作中にデバイスおよびアプリケーションをクリーンにシャットダウンするスクリプトを使用できます。</p> <p>詳細は、第 17 章を参照してください。</p>	4/01
<p>Solaris 印刷マネージャを使用して、USB (Universal Serial Bus) ポートを備えた SPARC システムに接続された USB プリンタを設定することができます。詳細は、118ページの「USB プリンタのサポート」を参照してください。</p> <p>USB の概要については、103ページの「USB デバイスの概要」を参照してください。</p>	10/00。 1/01、4/01 で更新。
<p>システム管理者が、構成したスワップ領域や専用ダンプデバイスなどのシステムリソースを削除する際の問題に対処できるように、動的再構成に関する新しいエラーメッセージが追加されました。</p> <p>動的再構成についての詳細は、121ページの「新しい動的再構成のエラーメッセージ」を参照してください。</p>	1/01
システムのリソースの管理	

表 1-1 Solaris 8 の新規機能 続く

説明	更新リリース
<p>アカウントの拡張によって、一般的なアカウントデータのグループを表す、新しい可変長の汎用アカウントファイル形式が導入されました。また、カーネルがさまざまなアカウントファイルに記録されたリソースの使用状況を構成する機能も導入されました。</p> <p>この機能の使い方については、169ページの「アカウント機能の拡張」を参照してください。</p>	6/00
システムのパフォーマンスの管理	
<p>機能拡張されたディレクトリ名検索キャッシュ (DNLC) により、大規模なディレクトリ内のファイルにアクセスするときのパフォーマンスが向上しました。</p> <p>この機能の使い方については、173ページの「DNLC 機能の向上」を参照してください。</p>	6/00
<p>『Solaris カーネルのチューンアップ・リファレンスマニュアル』が更新されました。semsys:seminfo_semmnu パラメータの情報が追加されています。</p> <p>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』を参照してください。</p>	1/01
アーリーアクセス	
<p>このリリースでは、アーリーアクセス (EA) ディレクトリにアーリーアクセスソフトウェアが含まれています。詳細は、Solaris 8 のリリースの SOFTWARE 2 of 2 CD に含まれる各アーリーアクセスソフトウェアの README を参照してください。</p>	—

システム管理ツールを使用した管理についてのトピック

このトピックでは、Solaris 環境においてシステム管理ツールを使用した管理について記述します。次の章で構成されています。

第 3 章

SMC ツールについての情報、および SMC の起動や停止の手順について記述します。

第 4 章

『Solaris WBEM Services の管理』の更新事項の概要を記述します。

Solaris Management Console の管理

Solaris 8 1/01 リリースで、Solaris Management Console が新規に追加されました。Solaris のシステム管理についての一般的な情報については、『Solaris のシステム管理 (第 1 巻)』を参照してください。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「Solaris 8 Reference Manual Collection」には記載されていない新しい情報も提供されています。

Solaris Management Console の概要

Solaris Management Console (SMC) 2.0 は GUI ベースの「傘型アプリケーション」で、各種の管理ツールの起動ポイントとして機能します。SMC は、以下のツールを含むデフォルトのツールボックスとともに使用することで、完全に機能するようになります。

- プロセス - プロセスを中断、再開、監視、および制御します。
- ユーザー - ユーザーのアカウント、ユーザーのテンプレート、グループ、メンバーリスト、管理役割、および権利を、設定し維持します。ともに動作することのできる特定のアプリケーションを管理したり各ユーザーがどのタスクを実行できるかを管理するために、ユーザーや管理役割に対して権利を与えたり拒否したりします。
- スケジュールされたジョブ - ジョブをスケジュール、起動、および管理します。

- マウントと共有 - マウント情報、共有情報、利用情報を表示および管理します。
- ディスク - ディスクパーティションを作成および表示します。
- シリアルポート - 既存のシリアルポートを構成および管理します。
- ログビューア - アプリケーションやコマンド行メッセージを表示し、ログファイルを管理します。

注 - ディスクレスクライアントも管理することができますが、GUI 形式ではなく、コマンド行からのみ可能です。「コマンド行インタフェースの使用」を参照してください。

SMC ツールボックスエディタの使用により、デフォルトのツールボックスにツールを追加したり削除したり、また別のツールのセットを管理するために新しいツールボックスを作成したりすることができます。

SMC の起動についての詳細は、25ページの「Solaris Management Console の起動」を参照してください。また、各ツールに関連するヘルプも参照してください。

コマンド行インタフェースの使用

GUI ベースでの SMC の使用に加えて、コマンド行インタフェースを使用して次のことを行うことができます。

- セキュリティ属性のデータベースをネームサービスに移植する - `smattrpop.1m`
- SMC を起動する - `smc.1m`
- SMC を構成する - `smcconf.1m`

また、コマンドを使用して以下を管理することができます。

- `crontab` データベース内のジョブ - `smcron.1m`
- ディスクレスクライアント:
 - GUI では実行できず、コマンド行からのみ可能 - `smdiskless.1m`
 - ディスクレスクライアントについての OS サービスをリスト表示し、すべての既存のディスクレスクライアントにおけるパッチを管理する - `smosservice.1m`

ディスクレスクライアントについての詳細は、第10章を参照してください。

- `exec_attr` データベース内のエントリ - `smexec.1m`

- グループエントリ - smgroup.1m
- 電子メール別名のエントリ - smmaillist.1m
- バッチのユーザー操作 - smmultiuser.1m
- OS サービス - smoservice.1m
- prof_attr および exec_attr データベースのプロファイル (権利) - smprofile.1m
- 役割アカウントにおける役割とユーザー - smrole.1m
- ユーザーのエントリ - smuser.1m

各コマンドについての詳細は、それぞれのコマンドのマニュアルページを参照してください。

Solaris Management Console の起動

Solaris Management Console (SMC) には、主に次の 3 つの要素があります。

- コンソール
- SMC ツールボックスエディタ
- SMC サーバー

コンソールは、コマンド行から (手順は以下を参照)、CDE フロントパネルの「ツール」メニューから、またはアプリケーション・マネージャかファイルマネージャの SMC アイコンのダブルクリックによって起動することができます。

▼ コンソールをコマンド行から起動するには

1. /usr/sadm/bin (デフォルトのディレクトリ) から、次のように実行します。

```
% smc
```

注 - SMC は一般ユーザーとして起動することができますが、一部のツールやアプリケーションは、スーパーユーザーとしてログインしない場合や SMC サーバーのログイン中に役割にならない場合はロードしないことがあります。

▼ SMC ツールボックスエディタを起動するには

1. /usr/sadm/bin (デフォルトのディレクトリ) から、次のように実行します。

```
% smc edit
```

注 - SMC ツールボックスエディタは一般ユーザーとして起動することができますが、スーパーユーザーとしてログインしない場合サーバーのツールボックスを保存することはできません。

▼ SMC サーバーが動作しているか判断するには

SMC の実行にトラブルがある場合、SMC サーバーが動作していないかまたは何らかのエラー状態にある可能性があります。SMC サーバーが動作しているかどうかを判断するには、以下の手順を実行します。

1. スーパーユーザーとして、以下を実行します。

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

SMC サーバーが動作している場合は、次のようなメッセージが返されます。

▼ SMC サーバーを起動するには

1. スーパーユーザーとして、以下を実行します。

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

SMC を起動すると、「SMC サーバーの準備が完了しました」というメッセージが返されます。

▼ SMC サーバーを停止するには

1. スーパーユーザーとして、以下を実行します。

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

「ポート 898 上の SMC サーバーを停止しています」というメッセージが返されます。

WBEM でのデスクトップ、デバイス、ネットワークの管理

Solaris 8 10/00 リリースおよび 1/01 リリースで、『Solaris WBEM Services の管理』の次の情報が更新されました。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「Solaris 8 Reference Manual Collection」には記載されていない新しい情報も提供されています。

『Solaris WBEM Services の管理』の更新

WBEM (Web-based Enterprise Management) には、さまざまなプラットフォームにおけるシステム、ネットワーク、デバイスの Web ベースでの管理の標準規格が含まれています。この標準化により、システム管理者は、デスクトップ、デバイス、およびネットワークの管理を行うことができます。

表 4-1 WBEM の変更点

<p>Solaris 8 10/00 Update リリースでの更新事項の概要</p>	<p>「Solaris スキーマ」の章が更新され、次の情報が含まれるようになりました。</p> <ul style="list-style-type: none"> ■ Solaris_CIMOM1.0.mof ファイルの記述。このファイルには、CIM Object Manager が使用するすべてのシステムプロパティの記述が含まれています。 ■ Solaris_Device1.0.mof ファイルが更新され、新しい Solaris_Printer とその他の印刷定義クラス、および Solaris_TimeZone クラスの記述が含まれるようになりました。 <p>詳細は、「Solaris スキーマ」の章を参照してください。</p>
<p>Solaris 8 1/01 Update リリースでの更新事項の概要</p>	<ul style="list-style-type: none"> ■ 「CIM Object Manager」の章が次のように更新されました。 <ul style="list-style-type: none"> ■ <code>init.wbem</code> コマンドの記述が更新されました。このコマンドは、CIM Object Manager だけでなく Solaris Management Console (SMC) サーバーも起動できるようにになりました。 ■ CIM Object Manager のレポジトリの更新方法を記述するセクションが追加されました。 ■ 「セキュリティの管理」の章が更新され、役割によるアクセス制御 (RBAC) 実行のための Solaris Management Console (SMC) について記述されました。 ■ 「システムのロギング」の章が次のように更新されました。 <ul style="list-style-type: none"> ■ <code>wbemlogviewer</code> アプリケーションが Solaris Management Console (SMC) ログビューアに置き換えられました。 ■ 「Solaris スキーマ」の章が次のように更新されました。 <ul style="list-style-type: none"> ■ <code>Solaris_SerialPortSetting</code> クラスが、<code>Solaris_Core1.0.mof</code> ファイルから <code>Solaris_Device1.0.mof</code> ファイルへ移動されました。 ■ <code>Solaris_Application1.0.mof</code> ファイルの記述が更新され、<code>Package Status</code> のパッケージ属性が削除されました。1 つのパッチ属性 <code>Packages</code> が追加されました。 ■ <code>Solaris_System1.0.mof</code> ファイルについての記述が更新され、このファイルで定義されたクラスの完全なリストが追加されました。 ■ 新しい <code>Solaris_Network1.0.mof</code> ファイルについての記述が追加されました。 ■ 新しい <code>Solaris_Users1.0.mof</code> ファイルについての記述が追加されました。

セキュリティの管理についてのトピック

このトピックでは、Solaris 環境におけるセキュリティの管理について記述します。次の章で構成されます。

第 6 章	用語の変更、権利プロファイルにおける変更、 <code>policy.conf(4)</code> における変更を含む、RBAC の変更点について記述します。
第 7 章	Generic Security Services Application Programming Interface (GSS-API) について記述します。
第 8 章	『Solaris スマートカードの管理』の更新事項の概要を記述します。

RBAC でのセキュリティの管理

Solaris 8 1/01 リリースで役割によるアクセス制御 (RBAC: role-based access control) のデータベースが拡張されました。次に記載する情報は、『Solaris のシステム管理 (第 2 巻)』の「役割によるアクセス制御」の章に記載されている RBAC データベースについての記述を補足するものです。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「Solaris 8 Reference Manual Collection」には記載されていない新しい情報も提供されています。

役割によるアクセス制御

役割によるアクセス制御 (RBAC: role-based access control) データベースを、Solaris Management Console (SMC) のグラフィカルインタフェースの「ユーザー」ツールから管理することができるようになりました。SMC についての詳細は、23ページの「Solaris Management Console の概要」を参照してください。RBAC では以下の機能が更新されています。

- *execution profiles* という用語が使用されなくなり、権利プロファイル (*rights profiles*) に置き換えられました。グラフィカルインタフェース上では「権利 (*rights*)」、コマンド行およびファイル内では *profiles* とも呼ばれます。
- 承認とセキュリティ属性を備えたコマンドに加え、権利プロファイルが他の権利プロファイルを含むことができるようになりました。複数の従属する権利プロ

ファイルに同じコマンドが現れる場合、ファイル内の最初の出現が優先されます。

- `policy.conf(4)` ファイルが、デフォルトで権利プロファイルを割り当てられるようにするキーワード `PROFS_GRANTED` を認識できるようになりました。

次の図は、拡張されたユーザー属性がどのようにユーザーに提供されるかを示します。

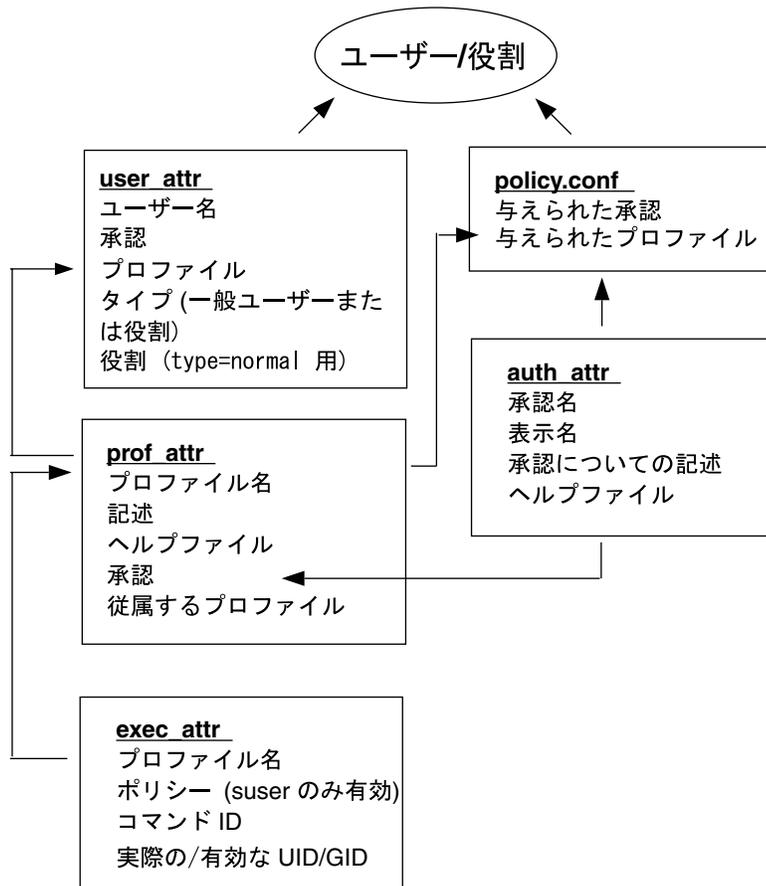


図 6-1 拡張された属性データベース

`user_attr` データベースは、表示されている属性を含み、またコンマ (,) で区分されたプロファイル名のリストを含んでいます。プロファイルの内容は、`prof_attr` ファイル (プロファイル識別情報、そのプロファイルに割り当てられた承認、従属プロファイルを含む) と `exec_attr` ファイル (ポリシーを識別し、関連するセキュリティ

ティ属性を備えたコマンドを含む)に分けられます。auth_attr ファイルは SMC ツールに承認情報を提供します。user_attr を使用してユーザーに直接承認を割り当てることはできますが、これは推奨されない方法であることに注意してください。policy.conf ファイルは、すべてのユーザーに割り当てられるデフォルトの属性を提供します。

たとえば、Printer Management 権利プロファイルが1つのユーザーまたは役割に割り当てられると、そのユーザーまたは役割の user_attr エントリはキーワードと値のペア (profiles=Printer Management) を含みます。prof_attr ファイルは以下の行でこのプロファイルを定義します。また、ヘルプファイルと承認も指定します。

```
Printer Management:::Manage printers, daemons,
spooling:help=RtPrntAdmin.html;auths=solaris.admin.printer, /
solaris.admin.printer.modify,solaris.admin.printer.delete
```

exec_attr ファイルでは、次の行が有効なユーザー ID = lp を Printer Management プロファイル内のコマンド /usr/sbin/accept に割り当てます。

```
Printer Management:suser:cmd:::/usr/sbin/accept:euid=lp
```

次の表は、承認を使用するコマンドのリストです。

表 6-1 コマンドとその承認

コマンド	関連する承認
at(1)	solaris.jobs.user
atq(1)	solaris.jobs.admin
crontab(1)	solaris.jobs.user, solaris.jobs.admin
allocate(1M)	solaris.device.allocate, solaris.device.revoke
deallocate(1M)	solaris.device.allocate, solaris.device.revoke
list_devices(1M)	solaris.device.revoke
rdate(1M)	solaris.system.date
smcron(1M)	solaris.jobs.admin, solaris.jobs.user

表 6-1 コマンドとその承認 続く

コマンド	関連する承認
<code>smdiskless (1M)</code>	<code>solaris.admin.dcmgr.clients</code> , <code>solaris.admin.dcmgr.read</code>
<code>smexec (1M)</code>	<code>solaris.profmgr.read</code> , <code>solaris.profmgr.write</code>
<code>smgroup (1M)</code>	<code>solaris.admin.usermgr.read</code> , <code>solaris.admin.usermgr.write</code>
<code>smmultiuser (1M)</code> , <code>smuser (1M)</code>	<code>solaris.admin.usermgr.pswd</code> , <code>solaris.admin.usermgr.read</code> , <code>solaris.admin.usermgr.write</code> , <code>solaris.profmgr.assign</code> , <code>solaris.profmgr.delegate</code> , <code>solaris.role.assign</code> , <code>solaris.role.delegate</code>
<code>mmaillist (1M)</code>	<code>solaris.admin.usermgr.read</code> , <code>solaris.admin.usermgr.write</code>
<code>smoservice (1M)</code>	<code>solaris.admin.dcmgr.admin</code> , <code>solaris.admin.dcmgr.read</code>
<code>smprofile (1M)</code>	<code>solaris.profmgr.read</code> , <code>solaris.profmgr.write</code>
<code>smrole (1M)</code>	<code>solaris.admin.usermgr.pswd</code> , <code>solaris.admin.usermgr.read</code> , <code>solaris.admin.usermgr.write</code> , <code>solaris.profmgr.assign</code> , <code>solaris.profmgr.delegate</code> , <code>solaris.role.assign</code> , <code>solaris.role.delegate</code>

GSS-API でのデータの転送

Solaris 8 6/00 リリースで『GSS-API のプログラミング』が新規に追加されました。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。
Solaris 8 Update リリースのマニュアルページには、「Solaris 8 Reference Manual Collection」には記載されていない新しい情報も提供されています。

『GSS-API のプログラミング』

Generic Security Service Application Programming Interface (GSS-API) は、セキュリティのフレームワークです。GSS-API を使用すると、アプリケーションは転送するデータを保護できます。GSS-API は認証、整合性、および機密性のサービスをアプリケーションに提供します。このインタフェースを使用すると、各アプリケーションはセキュリティに関して一般的に「汎用」になります。つまり、どのような実際のプラットフォーム (Solaris プラットフォームなど) やセキュリティ機構 (Kerberos など) が使用されるかを知る必要がありません。これは、GSS-API を使用するアプリケーションの移植性が高くなることを意味します。

詳細は、『GSS-API のプログラミング』を参照してください。

スマートカードでのセキュリティの管理

Solaris 8 1/01 リリースと 4/01 リリースで『*Solaris* スマートカードの管理』が更新され、次の新しい情報が追加されました。

注 - 最新のマニュアルページを参照するには、`man` コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「*Solaris 8 Reference Manual Collection*」には記載されていない新しい情報も提供されています。

SPARC: 『*Solaris* スマートカードの管理』の更新

『*Solaris* スマートカードの管理』には、Solaris 8 ソフトウェアリリースにおける新しいセキュリティ機能が記載されています。

- カードリーダーのセットアップに関する章に、内部カードリーダーのセットアップについての情報が追加されました。
- スマートカードのセットアップに必要なタスクと、デフォルトのスマートカードプロパティではセキュリティ環境に充分でない場合に必要となる追加の構成タスクについて記述した、新しい章が追加されました。Solaris 8 4/01 リリースではこれらの手順がより分かりやすくなり、スマートカードの設定プロセスが容易になりました。
- 以前のバージョンで技術的に不正確であった箇所が修正されました。

サーバーとクライアントの管理についてのトピック

このトピックでは、Solaris 環境におけるサーバーとクライアントの管理について記述します。次の章で構成されます。

第 10 章

ディスククライアント環境のセットアップおよび管理について記述します。

ディスクレスクライアントの管理

Solaris 8 1/01 リリースで Diskless Client Management が新規に追加され、Solstice AdminSuite™ 2.3 ディスクレスクライアントツールが更新されました。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「Solaris 8 Reference Manual Collection」には記載されていない新しい情報も提供されています。

ディスクレスクライアントの管理の概要

Diskless Client Management により Solstice AdminSuite 2.3 ディスクレスクライアントツールが更新されました。AdminSuite 2.3 ディスクレスクライアントツールは GUI をベースとしたもので、一方 Diskless Client Management はコマンド行インタフェースからのみ構成されています。

次のものがサポートされます。

- Solaris 8 1/01 オペレーティング環境または互換バージョンを実行する SPARC アーキテクチャまたは IA OS サーバー
- OS サーバーから Solaris 8 1/01、Solaris 2.7、または Solaris 2.6 オペレーティング環境のいずれかを実行する SPARC アーキテクチャの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、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 サーバーまたはあるホストに依存するワークステーションです。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

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。

Diskless Client Management の使用

Diskless Client Management ツールは、コマンド行インタフェースを使用して動作させます。独自のシェルスクリプトを書いて表 10-1 に示すコマンドを使用することにより、ディスクレスクライアント環境を簡単にセットアップし管理することができます。

表 10-1 Diskless Client Management のコマンド

コマンド	サブコマンド	タスク
/usr/sadm/bin/smosservice	add	OS サービスを追加する
	delete	OS サービスを削除する
	list	OS サービスをリスト表示する
	patch	OS サービスのパッチを管理する
/usr/sadm/bin/smdiskless	add	ディスクレスクライアントを OS サーバーに追加する
	delete	ディスクレスクライアントを OS サーバーから削除する
	list	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する
	modify	ディスクレスクライアントの属性を変更する

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。

- 「使用法」 - 「使用法」を参照するには、コマンド、サブコマンド、および必要なオプションの後に `-h` オプションを使用します。たとえば、`smdiskless add` の「使用法」を表示するには、次のように実行します。

```
% /usr/sadm/bin/smdiskless add -p my_password -u my_user_name -- -h
```

- マニュアルページ - マニュアルページを参照するには、`man` の後にコマンド名を入力します。たとえば `smdiskless` のマニュアルページを表示するには、次のように実行します。

```
% man smdiskless
```

ユーザーの権利

ユーザーは、割り当てられている権利に応じて、Diskless Client Management コマンドのサブセットまたはすべてのいずれかを利用することができます。表 10-2 は、Diskless Client Management コマンドの使用に必要な権利を示します。

表 10-2 必要な権利

権利	コマンド	タスク
基本的な Solaris ユーザー、 ネットワーク管理	<code>smosservice list</code>	OS サービスをリスト表示する
	<code>smosservice patch</code>	OS パッチをリスト表示する
	<code>smdiskless list</code>	ディスクレスクライアントをリスト表示する
ネットワーク管理	<code>smdiskless add</code>	ディスクレスクライアントを追加する
システム管理者	すべてのコマンド	すべてのタスク

必要なディスクスペース

ディスクレスクライアント環境をセットアップする前に、各ディスクレスクライアントのディレクトリ用に、必要なディスクスペースがあることを確認します。

表 10-3 必要なディスクスペース

パーティション	必要なスペース (単位: M バイト)
<code>/export/Solaris_version</code>	10
<code>/export/exec</code>	800
<code>/export/share</code>	5
<code>/export/swap/diskless_client</code>	32 (デフォルトのサイズ)
<code>/export/dump/diskless_client</code>	32 (デフォルトのサイズ)
<code>/export/root/templates/Solaris_version</code>	30
<code>/export/root/clone/Solaris_version/machine_class</code>	30 ~ 60。マシンのクラスによる
<code>/export/root/diskless_client</code> (上記のクローン)	30 ~ 60。マシンのクラスによる
<code>/tftpboot/inetboot.machine_class.Solaris_version</code>	<code>machine_class.Solaris_version</code> 1 つにつき、200K バイト

▼ ディスクレスクライアント環境をセットアップするには

- 次に示すどちらの手順から開始するかを選択します。
 - システムが、AdminSuite 2.3 Diskless Client ツールで作成されたディスクレスクライアントを現在サポートしている場合は、手順 2 へ進みます。

- システムが、AdminSuite 2.3 Diskless Client ツールで作成されたディスクレスクライアントを現在サポートしていない場合は、手順 4 へ進みます。
- 2. `admhostdel` コマンドを使用して、既存の **AdminSuite 2.3** ディスクレスクライアントを削除します。
- 3. `admhostmod` コマンドを使用して、既存の **AdminSuite 2.3 OS** サービスを削除します。
- 4. **OS** サーバーとして指定されたマシンを **Solaris 8 1/01** オペレーティング環境または互換バージョンへアップグレードします。
- 5. **SMC** のログビューアツールを使用してディスクレスクライアントのエラーメッセージを参照するには、**SMC** を起動します。

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

SMC のメインウィンドウから「ログビューア」を選択します。

- 6. 必要な **OS** サービスを追加します。
- 7. ディスクレスクライアントを追加します。
- 8. `boot net` コマンドを使用して、**PROM** レベルから各ディスクレスクライアントをブートします。このコマンドについての詳細は、『**Solaris** のシステム管理 (第 1 巻)』を参照してください。

OS サービスを追加する準備

`smosservice add` コマンドを使用して OS サービスを追加する場合は、サポートしたい各ディスクレスクライアントのプラットフォームの `platform`、`mediapath`、および `cluster` を入力する必要があります。そのため、各ディスクレスクライアントについて以下のことを決定するために最初にいくつかのハイレベルの作業を行う必要があります。

- `platform` (プラットフォーム) - `instruction_set.machine_class.Solaris_os_version` の形式でディスクレスクライアントのプラットフォームを指定します。たとえば、`sparc.sun4u.Solaris_8` などです。以下に、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_os_version</i>
sparc	sun4u	Solaris_8
	sun4m	Solaris_2.7
	sun4c	Solaris_2.6
	sun4d	
i386	i86pc	Solaris_8
		Solaris_2.7
		Solaris_2.6

- **media path** (メディアパス) - ディスクレスクライアント用にインストールしたいオペレーティングシステムを含む CD-ROM またはネットワークイメージへのフルパス。たとえば、`/net/install_files` など。

注 - SOLARIS 8 SOFTWARE CD から OS サービスをロードする場合 - Solaris 8 オペレーティング環境は、複数の CD で提供されます。しかし、Diskless Client Management ソフトウェアはこの複数 CD による提供をサポートしていません。ユーザーは、次のことを行うために、SOLARIS 8 SOFTWARE CD (およびオプションの LANGUAGES CD) にあるスクリプトを実行する必要があります。

1. サーバー上にインストールイメージを作成する。インストールサーバーのセットアップについての情報は、『Solaris 8 のインストール (上級編)』を参照してください。
2. イメージから必要な OS サービスをロードする。

スクリプトは次のとおりです。

- SOFTWARE 1 of 2 CD 用 -
`/cdrom/cdrom0/s0/Solaris_8/Tools/setup_install_server`
- SOFTWARE 2 of 2 CD 用 -
`/cdrom/cdrom0/s0/Solaris_8/Tools/add_to_install_server`
- LANGUAGES CD 用 -
`/cdrom/cdrom0/s0/Solaris_8/Tools/add_to_install_server`

- cluster (クラスタ) - ディスクレスクライアントの構成によって、Diskless Client の機能を含む次の4つのクラスタのうち1つを指定することができます。SUNWCXa11、SUNWCa11、SUNWCprog、またはSUNWCuserです。同じマシン (SPARC または IA) 上で同じオペレーティング環境を実行するディスクレスクライアントには、「同じクラスタ」を使用しなければなりません。

たとえば、次のディスクレスクライアントをセットアップするとします。

- sparc.sun4m.Solaris_8
- sparc.sun4u.Solaris_8
- sparc.sun4d.Solaris_8

sun4u を実行するマシンは SUNWCXa11 を必要とするため、また同じマシン上で同じオペレーティング環境 (この例の場合は Solaris_8) を実行するすべてのディスクレスクライアントは同じクラスタを使用する必要があるため、各ディスクレスクライアントには SUNWCXa11 クラスタを指定することになります。

注 - sun4u マシンを使用している場合、または高速化8ビットカラーメモリーフレームバッファ (cgsix) を備えたマシンを使用している場合は、クラスタとして必ず SUNWCXa11 を指定する必要があります。

▼ OS サービスを追加するには

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となります。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
```

1. 必要なメディアパス、プラットフォーム、およびクラスタのオプションとともに `smoservice add` コマンドを使用して、1 つ目の **OS** サービスを追加します。サーバーのスピードおよび選択した **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。
2. `smoservice add` コマンドを繰り返し使用して、他の **OS** サービスを追加します。
3. **OS** サービスの追加が完了したら、`smoservice list` コマンドを使用して **OS** サービスがインストールされたことを確認します。

▼ ディスクレスクライアントを追加するには

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
```

```
/export/dump/diskless_client (-x dump オプションを指定した場合)
```

```
/tftpboot/diskless_client_ipaddress_in_hex
```

注 - `-x` オプションを使用すると、`root`、`/swap`、および `/dump` ディレクトリのデフォルトのロケーションを変更することができます。ただし、`/export` 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。

1. 追加したい 1 つめのディスクレスクライアントについて、必要な IP アドレス、**Ethernet** アドレス (**MAC** アドレス)、名前、およびオペレーティングシステムのオプションとともに `smdiskless add` コマンドを使用します。オペレーティングシステムは、`instruction_set.machine_class.Solaris_os_version` のフォーマットで、**OS** サービスのセットアップのために `smossservice` コマンドを使用した時に指定した *platform* と同じにします。
2. `smdiskless add` コマンドを繰り返し使用して、各ディスクレスクライアントを追加します。
3. ディスクレスクライアントの追加が完了したら、`smdiskless list` コマンドを使用して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。

OS サービスへのパッチ適用

`smossservice patch` コマンドを使用して、次のことを行うことができます。

- OS サーバー上に `/export/diskless/Patches` パッチスプールディレクトリを確立する
- パッチスプールディレクトリにパッチを追加する。追加されるパッチによって既存のパッチが差し換えられる場合は、古いパッチは `/export/diskless/Patches/Archive` へ移動されます。
- パッチスプールディレクトリからパッチを削除する
- パッチスプールディレクトリ内のパッチをリスト表示する
- スプールされたパッチをクライアントに同期させる。クライアントがパッチの更新を認識できるように、同期させた各クライアントをリポートする必要があります。

注 - 推奨される OS のパッチを適時にインストールして、OS サーバーを常に最新の状態にしておいてください。

パッチの表示

ディスクレスクライアントのパッチは、次のようにパッチのタイプによって異なるディレクトリに保存されます。

- カーネルパッチはディスクレスクライアントの `/var/sadm/patch` ディレクトリに保存されます。ディスクレスクライアントからカーネルパッチを表示するには、次のように実行します。

```
% showrev -p
```

- `/usr` パッチは OS サーバーの `/export/Solaris_version/var/patch` ディレクトリに保存されます。各パッチ ID につき 1 つのディレクトリが作成されます。パッチをリスト表示するには、このディレクトリに移動して次のように実行します。

```
% ls -l
```

OS およびアーキテクチャによってスプール処理されたすべてのパッチをリスト表示するには、`smosservice` コマンドを `-P` オプションと共に使用します。

トラブルシューティング

この節では、Diskless Client Management でよく見られるいくつかの問題と、考えられる解決策を記載します。

問題

- OS サーバーがクライアントの RARP 要求に応答しない。
- OS サーバーがクライアントの `bootparam` 要求に応答しない。
- OS サーバーがディスクレスクライアントの `root` ファイルシステムをマウントできない。

解決策

files 環境において

- `files` が、OS サーバー上の `/etc/nsswitch.conf` 内の `hosts`、`ethers`、および `bootparams` の第一のソースとしてリスト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` 内にあることを確認します。
- クライアントの Ethernet アドレスが `/etc/ethers` 内にあることを確認します。

- /etc/bootparams ファイルに、クライアントの root および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。

```
diskless_client root=os_server:/export/  
root/diskless_client swap=os_server:/export/  
swap/diskless_client swapspace=24
```

スワップサイズは、ディスククライアントを追加する際に `-x swapspace` オプションを指定したかどうかによって異なります。ディスククライアントを追加する際に `-x dump` オプションを指定した場合は、次の行が存在します。

```
dump=os_server:/export/dump/diskless_client dumpsize=24
```

ダンプサイズは、ディスククライアントを追加する際に `-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/diskless_client/etc/inet/hosts` 内にあることを確認します。

name service 環境において

- OS サーバーの、およびクライアントの Ethernet アドレスと IP アドレスがすべて正しくマップされていることを確認します。
- /etc/bootparams に、次のようにクライアントの root 領域およびスワップ領域へのパスが含まれていることを確認します。

```
diskless_client root=os_server:/export/  
root/diskless_client swap=os_server:/export/  
swap/diskless_client swapspace=24
```

スワップサイズは、ディスククライアントを追加する際に `-x swapspace` オプションを指定したかどうかによって異なります。ディスククライアントを追加する際に `-x dump` オプションを指定した場合は、次の行が存在します。

```
dump=os_server:/export/dump/diskless_client dumpsize=24
```

ダンプサイズは、ディスククライアントを追加する際に `-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がパニックになる

解決策

- OS サーバーの Ethernet アドレスが対応する 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。あるネットワークから別のネットワークへマシンを物理的に移動した場合、そのマシンの新しい IP アドレスをリマップし忘れている可能性もあります。
- クライアントのホスト名、IP アドレス、および Ethernet アドレスが、クライアントの RARP、TFTP、または bootparam 要求に回答する同じサブネット上の他のサーバーのデータベース内に存在していないことを確認します。しばしば、テストマシンが、インストールサーバーから OS をインストールするようセットアップされることがあります。この場合、そのインストールサーバーがクライアントの RARP や bootparam 要求に回答して、正しくない IP アドレスを返します。この正しくない IP アドレスによって、間違ったアーキテクチャ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 root ファイルシステムのマウントに失敗してしまう可能性があります。
- ディスクレスクライアントの TFTP 要求が、正しくないブートプログラムを転送するインストールサーバー (または以前の OS サーバー) によって回答されていないことを確認してください。ブートプログラムが異なるアーキテクチャのものである場合は、クライアントはすぐにパニックになります。ブートプログラムが非 OS サーバーからロードする場合、クライアントは、その非 OS サーバーからの root パーティションと、OS サーバーからの /usr パーティションを取得する可能性があります。この場合、root と /usr パーティションが相反するアーキテクチャやバージョンのものであると、クライアントはパニックになります。
- 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、次のエントリが /etc/dfs/dfstab 内に存在することを確認します。

```
share -F nfs -o -ro /export/exec/Solaris_version_instruction_set.all/usr
```

version は 2.6、2.7、または 8 のいずれか、*instruction_set* は sparc または i386 です。

- ディスクレスクライアントの root、/swap、および /dump (指定した場合) パーティションに共有エントリがあることを確認します。

```
% share -F nfs -o rw=client_name,root=client_name /export/root/client_name
% share -F nfs -o rw=client_name,root=client_name /export/swap/client_name
% share -F nfs -o rw=client_name,root=client_name /export/dump/client_name
```

- OS サーバー上で次を実行して、どのファイルが共有されているかをチェックします。

```
% share
```

OS サーバーは、ディスクレスクライアントを追加した際に指定した、`/export/root/client_name` および `/export/swap/client_name` (デフォルト)、または `root`、`/swap`、および `/dump` パーティションを共有していなければなりません。

次のエントリが `/etc/dfs/dfstab` 内にあることを確認します。

```
% share -F nfs -o ro /export/exec/Solaris_version_instruction_set.all/usr
```

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。

解決策

クライアントの意図する OS サーバーから、クライアントの Ethernet アドレスを使用してスーパーユーザーとして `snoop` を実行します。

```
# snoop xx:xx:xx:xx:xx:xx
```

問題

ブートプログラムはダウンロードするが、プロセスの初期にパニックになる。

解決策

`snoop` を使用して、意図している OS サーバーがクライアントの TFTP および NFS 要求に応答していることを確認します。

問題

- ディスクレスクライアントがハングする。

- ディスクレスクライアントの RARP 要求に対して、正しくないサーバーが応答する。

解決策

OS サーバー上で、次のように再起動します。

```
% /usr/bin/rpc.bootparamd  
% /usr/sbin/in.rarpd -a
```

ファイルシステムの管理についてのトピック

このトピックでは、Solaris 環境におけるファイルシステムの管理について記述します。次の章で構成されます。

第 12 章

通常の UFS ファイルへの読み取りおよび書き込みアクセスの並行処理が可能になったことによる、UFS の直接入出力のパフォーマンスの向上について記述します。

第 13 章

新しい `fssnap` コマンドを使用してシステムの読み取り専用のスナップショットを作成することによる、ファイルシステムのバックアップの方法について記述します。

UFS ソフトウェアでのファイルシステムの管理

Solaris 8 1/01 リリースで UFS ソフトウェアが拡張されました。次に記載する情報は、『Solaris のシステム管理 (第 1 巻)』の「ファイルシステムの管理 (概要)」の章の直接入出力に関する情報を補足するものです。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「Solaris 8 Reference Manual Collection」には記載されていない新しい情報も提供されています。

UFS 直接入出力の並行処理の向上

Solaris 8 ソフトウェアリリースに、新しい UFS 機能が含まれるようになりました。バッファ処理されていないファイルシステムのデータにアクセスするためにデータベースアプリケーションが使用する直接入出力のパフォーマンスが改良され、通常の UFS ファイルへの読み取りおよび書き込みのアクセスの並行処理が可能になりました。これまでは、ファイルデータを更新する操作は、その更新操作が完了するまで、その他すべての読み取りアクセスまたは書き込みアクセスをロックアウトするようになっていました。

書き込みの並行処理は、ファイルのリライトという特別なケースのみに限られません。ファイルを拡張する場合は、書き込みは従来のようにシングルスレッドで行われます。一般に、データベースはファイルを事前に割り当て、その後はあまり拡張

することはありません。そのため、この拡張による効果は通常のデータベース操作時に見られます。

直接入出力の改良により、UFS ファイルシステムにおける入出力を長時間使用するデータベースのパフォーマンスは、raw パーティションのアクセススピードの約 90% に短縮されます。データベースが CPU やバス大域幅を長時間使用する場合は、このパフォーマンスの向上は見られないことになります。

データベーステーブルの保存にすでに UFS を使用している場合、入出力データベースのアプリケーションを直接入出力が可能な状態で実行することを想定してみてください。可能であれば、直接入出力の有効化にデータベースの管理手順を使用します。直接入出力をデータベース製品から可能にすることができない場合は、`mount -o forcedirectio` オプションを使用して各ファイルシステムについて直接入出力を有効にするか、または `directio(3C)` ライブラリ呼び出しを使用して直接入出力を有効にします。

詳細は、`mount_ufs(1M)` または `directio(3C)` のマニュアルページを参照してください。

fssnap でのファイルシステムのバックアップ

fssnap コマンドが Solaris 8 1/01 リリースで新規に追加されました。次に記載する情報は、『Solaris のシステム管理 (第 1 巻)』の「ファイルシステムのバックアップと復元 (概要)」に記載されているファイルシステムのバックアップについての情報を補足するものです。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「Solaris 8 Reference Manual Collection」には記載されていない新しい情報も提供されています。

UFS スナップショットの概要

Solaris 8 1/01 リリースには、ファイルシステムのマウント中にファイルシステムをバックアップするための、新しい fssnap コマンドが含まれるようになりました。

fssnap コマンドを使用して、ファイルシステムの読み取り専用のスナップショットを作成することができます。スナップショットは、バックアップ操作のためのファイルシステムの一時的イメージです。

fssnap コマンドを実行すると、1つの仮想デバイスと1つのバックアップストアファイルが作成されます。ユーザーは、既存の Solaris バックアップコマンドを使用して、実在のデバイスのように動作し実在のデバイスのように見えるこの仮想デバイスをバックアップすることができます。バックアップストアファイルは、スナッ

プショットがとられてから変更されたデータのコピーを含んだビットマップ化ファイルです。

なぜ UFS スナップショットを使用するか

UFS スナップショットにより、バックアップ時に、ファイルシステムをマウントされた状態にしシステムをマルチユーザーモードにしておくことができます。これまでは、バックアップを実行するために `ufsdump` コマンドを使用する時は、ファイルシステムをアクティブでない状態に保つためにシステムをシングルユーザーモードにすることが推奨されていました。より確実なバックアップのために、`tar` や `cpio` などの追加の Solaris のバックアップコマンドを使用して UFS スナップショットのバックアップを行うこともできます。

`fssnap` コマンドにより、企業レベルではないシステムの管理者が、大規模な記憶容量の必要なく、Sun StorEdge™ Instant Image のような企業レベルツールのパワーを得ることができます。

UFS スナップショットは、Instant Image 製品に似ています。Instant Image は、取り込まれるファイルシステム全体のサイズに等しいスペースを割り当てます。しかし、UFS スナップショットが作成するバックアップストアファイルは、必要なディスクスペースの容量しか占有せず、またバックアップストアファイルのサイズに上限を設定することもできます。

次の表は、UFS スナップショットと Instant Image との特徴的な違いを示します。

UFS スナップショット	Instant Image
バックアップストアファイルのサイズは、スナップショットがとられた後のデータの変更量による	バックアップストアファイルのサイズは、コピーされるファイルシステム全体のサイズに等しい
システムのリブート後は保持されない	システムのリブート後も保持される
UFS ファイルシステムで動作する	ルート (/) または /usr ファイルシステムでは使用できない
Solaris 8 1/01 リリースに組み込まれている	Enterprise Services Package に組み込まれている

UFS スナップショットは大規模なファイルシステムをコピーすることができますが、企業レベルのシステムには **Instant Image** の方が適しています。UFS スナップショットは、小さめのシステムに適しています。

UFS スナップショットのパフォーマンス上の問題

ファイルシステムのスナップショットが最初に作成される時、そのファイルシステムのユーザーは短い一時停止に気づくでしょう。一時停止の時間は、取り込まれるファイルシステムのサイズとともに増加します。ファイルシステムのスナップショットがアクティブな間、そのファイルシステムのユーザーは、ファイルシステムに書き込まれる際に若干のパフォーマンス上の影響に気づくでしょう。しかしファイルシステムが読まれる際には影響はありません。

UFS スナップショットの作成

`fssnap` コマンドを使用してファイルシステムのスナップショットを作成する際は、バックアップストアファイルがどれだけのディスクスペースを消費するかを監視してください。バックアップストアファイルは初めはスペースを全く使用せず、その後特によく使用されているシステムにおいて、急速に拡大します。バックアップストアファイルが拡大するのに十分なスペースを必ず確保しておくか、または `-o maxsize=n [k,m,g]` オプション (`n [k,m,g]` はバックアップストアファイルの最大限のサイズ) でそのサイズを制限してください。



注意 - バックアップストアファイルにスペースが不足する場合、スナップショットが削除されてしまうことがあり、バックアップが失敗します。スナップショットのエラーの可能性を調べるため、`/var/adm/messages` ファイルをチェックしてください。

▼ UFS スナップショットを作成するには

1. スーパーユーザーになります。
2. ファイルシステムにバックアップストアファイル用の十分なディスクスペースがあることを確認します。

```
# df -k
```

3. 同じロケーションに同じ名前の既存のバックストアファイルが存在していないことを確認します。

```
# ls /file-system/backing-store-file
```

4. ファイルシステムのスナップショットを作成します。

```
# fssnap -F ufs -o bs=/file-system/backing-store-file /file-system
```

例 - UFS スナップショットの作成

次の例では、`/usr` ファイルシステムのスナップショットを作成します。バックストアファイルは `/scratch/usr.back.file`、仮想デバイスは `/dev/fssnap/1` です。

```
# fssnap -F ufs -o bs=/scratch/usr.back.file /usr  
/dev/fssnap/1
```

次の例では、バックストアファイルを 500M バイトに制限します。

```
# fssnap -F ufs -o maxsize=500m,bs=/scratch/usr.back.file /export/home  
/dev/fssnap/1
```

▼ UFS スナップショットの情報を表示するには

`fssnap -i` オプションの使用により、システムの現在のスナップショットを表示することができます。1つのファイルシステムを指定する場合、そのスナップショットについての詳細な情報が表示されます。特定のファイルシステムを指定しない場合は、現在のすべてのファイルシステムのスナップショットとそれらに対応する仮想デバイスの情報が表示されます。

1. スーパーユーザーになります。
2. 現在のスナップショットをリスト表示します。

```
# fssnap -i
0 /
1 /usr
```

特定の1つのスナップショットについての詳細な情報を表示する場合は、次のように実行します。

```
# fssnap -i /usr
Snapshot number          : 1
Block Device             : /dev/fssnap/1
Raw Device               : /dev/rfssnap/1
Mount point              : /usr
Device state             : idle
Backing store path       : /scratch/usr.back.file
Backing store size       : 480 KB
Maximum backing store size : Unlimited
Snapshot create time     : Tue Aug 08 09:57:07 2000
Copy-on-write granularity : 32 KB
```

UFS スナップショットの削除

UFS スナップショットを作成する際、バックングストアファイルがリンクされないように指定することができます。これは、そのバックングストアファイルが、スナップショットが削除された後で削除されることを示します。UFS スナップショットを作成する際に `-o unlink` オプションを指定しない場合は、後で手動で削除する必要があります。

バックングストアファイルは、バックングストアファイルを削除するために `-o unlink` オプションを使用した場合はスナップショットが削除されるまで、そうでなければ手動で削除するまで、ディスクスペースを使用します。

▼ UFS スナップショットを削除するには

スナップショットは、システムをリブートするか、あるいは `fssnap -d` コマンドを使用してファイルシステムのスナップショットを含むファイルシステムのパスを指定することで、削除できます。

1. スーパーユーザーになります。
2. 削除するスナップショットを特定します。

```
# fssnap -i
```

3. そのスナップショットを削除します。

```
# fssnap -d /file-system  
Deleted snapshot 1.
```

4. (オプション) スナップショットの作成時に `-o unlink` オプションを使用しなかった場合は、そのバックアップストアファイルを手動で削除する必要があります。

```
# rm /file-system/backing-store-file
```

例 - UFS スナップショットの削除

以下は、`unlink` オプションを使用しなかった場合の、スナップショットを削除する例です。

```
# fssnap -i  
0 / 1 /usr  
# fssnap -d /usr  
Deleted snapshot 1.  
# rm /scratch/usr.back.file
```

UFS スナップショットのバックアップ

ファイルシステムのスナップショットを含む仮想デバイスは、標準の読み取り専用デバイスとして振舞います。これは、仮想デバイスを、ファイルシステムのデバイ

スをバックアップするかのようにバックアップすることができることを示します。

ufsdump コマンドを使用して UFS スナップショットをバックアップする場合、バックアップ時にスナップショットの名前を指定することができます。詳細は、次の節を参照してください。

tar コマンドを使用してスナップショットをバックアップする場合、次のように、バックアップを行う前にスナップショットをマウントします。

```
# mkdir /backups/home.bkup
# mount -F UFS -o ro /dev/fssnap/1 /backups/home.bkup
# cd /backups/home.bkup
# tar cvf /dev/rmt/0 .
```

ファイルシステムをバックアップする方法についての詳細は、『Solaris のシステム管理 (第 1 巻)』の「ファイルとファイルシステムのバックアップ(手順)」の章を参照してください。

▼ UFS スナップショットのバックアップを行うには

1. スーパーユーザーになります。
2. バックアップをとるファイルシステムのスナップショットを特定します。

```
# fssnap -i /file-system
```

例:

```
# fssnap -i /usr
Snapshot number           : 1
Block Device              : /dev/fssnap/1
Raw Device                : /dev/rfssnap/1
Mount point               : /usr
Device state              : idle
Backing store path        : /scratch/usr.back.file
Backing store size        : 480 KB
Maximum backing store size : Unlimited
Snapshot create time      : Tue Aug 08 09:57:07 2000
Copy-on-write granularity : 32 KB
```

3. ファイルシステムのスナップショットのバックアップを行います。

```
# ufsdump 0ucf /dev/rmt/0 /snapshot-name
```

例:

```
# ufsdump 0ucf /dev/rmt/0 /dev/rfssnap/1
```

4. スナップショットがバックアップされたことを確認します。

```
# ufsrestore ta /dev/rmt/0
```

▼ UFS スナップショットの増分ダンプを作成するには

ファイルシステムのスナップショットをインクリメンタル (増分的) に作成したい場合、つまり最後のスナップショット以降に変更のあったファイルだけをバックアップしたい場合は、`ufsdump` コマンドを新しい `N` オプションとともに使用します。このオプションは、増分ダンプをトラックするために `/etc/dumpdates` ファイルに挿入されるファイルシステムのデバイス名を指定します。

次の例では、`ufsdump` コマンド内で `fssnap` コマンドを組み込んでファイルシステムの増分ダンプを作成しています。

1. スーパーユーザーになります。
2. ファイルシステムスナップショットの増分ダンプを作成します。

```
# ufsdump lufN /dev/rmt/0 /dev/rdisk/c0t1d0s0 `fssnap -F ufs -o raw,bs=  
/export/scratch, unlink /dev/rdisk/c0t1d0s0`
```

上記の例では、ブロックデバイスではなく `raw` デバイスの名前を表示するために `-o raw` オプションが使用されています。このオプションの使用により、`fssnap` コマンドを `raw` デバイスを必要とするコマンド (`ufsdump` コマンドなど) に組み込むことが容易になります。

3. スナップショットがバックアップされたことを確認します。

```
# ufsrestore ta /dev/rmt/0
```

UFS スナップショットのバックアップからのデータの復元

仮想デバイスから作成されたバックアップは、基本的には、スナップショットがとられた時点でのオリジナルのファイルシステムの状態を表しています。バックアップから復元を行う場合は、オリジナルのファイルシステムから直接そのバックアップをとったかのように (たとえば `ufsrestore` コマンドを使用したものなどのように) 復元します。ファイルシステムの復元についての詳細は、『*Solaris* のシステム管理 (第 1 巻)』の「ファイルとファイルシステムの復元(手順)」の章を参照してください。

リムーバブルメディアの管理についてのトピック

このトピックでは、Solaris 環境におけるリムーバブルメディアの管理について記述します。次の章で構成されます。

第 15 章

Solaris のボリューム管理機能について記述します。共通デスクトップ環境 (CDE: Common Desktop Environment) でのボリューム管理と Solaris のコマンド行機能の両方が拡張され、リムーバブルメディアのサポートが向上しました。

リムーバブルメディアの管理

リムーバブルメディアの管理が Solaris 8 6/00 リリースで拡張され、この機能についてのマニュアルの記述が Solaris 8 10/00 リリースでも更新されました。次に記述する情報は、『Solaris のシステム管理 (第 1 巻)』の「CD とフロッピーディスクの使用方法 (概要)」の章のリムーバブルメディアの管理についての情報を補足するものです。ファイルマネージャを使用してこの機能を管理する方法については、『Solaris 8 デスクトップユーザズガイド (追補)』の「リムーバブルメディア」を参照してください。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「Solaris 8 Reference Manual Collection」には記載されていない新しい情報も提供されています。

リムーバブルメディアの管理の概要

Solaris 8 6/00 ソフトウェアリリースではボリューム管理機能が向上し、リムーバブルメディア (着脱式媒体) を完全にサポートするようになりました。この拡張により、DVD-ROM、Iomega と USB (Universal Serial Bus) の Zip ドライブと Jaz ドライブ、CD-ROM、およびフロッピーディスクは挿入時にマウントされ、読み取ることができるようになりました。

共通デスクトップ環境 (CDE) のボリューム管理と Solaris のコマンド行機能が Solaris 8 6/00 リリースで更新されました。

ボリューム管理の機能の向上によって、次のことが可能になりました。

- リムーバブルメディアのフォーマット、ラベル付け、および読み取りまたは書き込みソフトウェア保護の設定。新しい `rmformat` コマンドを使用します。このコマンドは以前にリムーバブルメディアのフォーマットに使用していた `fdformat` コマンドに代わって使用するものです。
- リムーバブルメディアへの PCFS ファイルシステムの作成と検証。 `mkfs_pcfs` コマンドと `fsck_pcfs` コマンドを使用します。
- SPARC システム上のリムーバブルメディアへの `fdisk` パーティションと PCFS ファイルシステムの作成。IA システムへのデータ転送を容易にします。

リムーバブルメディアを使用する場合、次のガイドラインがあります。

- DVD 媒体間でデータを転送するときは、UDFS と PCFS を使用します。
- 書き換え可能媒体 (UFS ファイルシステムを持つ PCMCIA メモリーカードやフロッピーディスクなど) 間でファイルを転送するときは、`tar` または `cpio` コマンドを使用します。SPARC システム上に作成された UFS ファイルシステムは、IA システム上に作成された PCMCIA またはフロッピーディスク上の UFS ファイルシステムとは異なります。
- Jaz ドライブや Zip ドライブ、またはフロッピーディスク上の重要なファイルを保護するには、書き込み保護を設定します。Iomega 媒体にはパスワードを適用します。

リムーバブルメディア上の情報へのアクセス

リムーバブルメディア上の情報にアクセスするには、ボリュームマネージャを使用する方法と使用しない方法があります。ファイルマネージャを使用してリムーバブルメディア上の情報にアクセスする方法については、『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』の「ファイル・マネージャでのリムーバブル・メディアの使い方」を参照してください。

Solaris 8 6/00 リリースから、ボリュームマネージャ (`vold`) ですべてのリムーバブルメディアデバイスを動的に管理できるようになりました。これに伴い、`/dev/rdisk/cntndnsn` や `/dev/dsk/cntndnsn` などのデバイス名でのリムーバブルメディアへのアクセスは、できなくなります。

ボリュームマネージャ `vold` が実行されている場合は、CDE のリムーバブルメディア・マネージャ、または `/cdrom0`、`/floppy`、`/rmdisk`、`/jaz0`、`/zip0` な

どのボリュームマネージャのパス名を使用して、デバイスにアクセスすることができます。

また、`/vol/dev` ディレクトリ内のエントリにより、リムーバブルメディアにアクセスすることもできます。たとえば、フロッピーディスクについては、次のように実行します。

```
/vol/dev/rdiskette0/volume-name
```

CD-ROM やリムーバブルハードディスクについては、次のように実行します。

```
/vol/dev/rdisk/cntndn/volume-name
```

リムーバブルメディアのデバイスにリムーバブルメディアが挿入されている場合は、`/vol/dev` ディレクトリ内のパスへのシンボリックリンクとして `/vol/dev/aliases` ディレクトリ内にその別名が現れます。たとえば、`test` とラベル付けされたフロッピーディスクがフロッピーディスクドライブ 0 に挿入されており、`test` とラベル付けされた CD が `/dev/rdisk/c2t1d0` の CD-ROM ドライブに挿入されている場合は、次の出力が表示されます。

```
$ ls -l /vol/dev/aliases
lrwxrwxrwx 1 root root 30 May 11 12:58 cdrom0 -> /vol/dev/rdisk/c2t1d0/test
lrwxrwxrwx 1 root root 30 May 11 12:58 floppy0 -> /vol/dev/rdiskette0/test
```

デバイス名が不明な場合は、`eject -n` コマンドを使用して、すべてのリムーバブルメディアのデバイスについてデバイス名を表示することができます。たとえば、`eject -n` 出力の右側のデバイス名の一覧をもとにして、`fsck`、`mkfs`、または `newfs` コマンドで使用するデバイス名を決定することができます。

▼ リムーバブルメディア上の情報にアクセスするには

コマンド行インタフェースを使用する場合は、適切なデバイス名を使用して情報にアクセスします。リムーバブルメディアにアクセスする前に `volcheck` コマンドを実行して、コマンド行からボリュームマネージャのニックネームを使用することもできます。デバイス名については、`rmformat(1)` のマニュアルページを参照してください。

例 - リムーバブルメディア上の情報にアクセスする

フロッピーディスク上の情報にアクセスするには、次のようにします。

```
$ volcheck
$ ls /floppy
myfile
```

Jaz ドライブ上の情報にアクセスするには、次のようにします。

```
$ volcheck
$ ls /rmdisk
jaz0/ jaz1/
```

CD-ROM 上の情報にアクセスするには、次のようにします。

```
$ volcheck
$ ls /cdrom
solaris_8_sparc/
```

Jaz ドライブおよび Zip ドライブへのアクセス

Solaris 8 6/00 リリースにアップグレードするか Solaris 8 6/00 リリースをインストールするかによって、Solaris 8 6/00 で更新された Jaz ドライブおよび Zip ドライブへのアクセス方法が異なります。

- 以前の Solaris リリースから Solaris 8 6/00 リリースへアップグレードする場合は、以前の Solaris リリースと同じ方法で Jaz ドライブおよび Zip ドライブにアクセスすることができます。
- 新規に Solaris 8 6/00 リリースをインストールする場合は、以前の Solaris リリースと同じ方法で Jaz ドライブおよび Zip ドライブにアクセスすることはできません。

以前の Solaris リリースと同じ方法で Jaz ドライブおよび Zip ドライブにアクセスしたい場合は、次の手順に従ってください。

1. スーパーユーザーになります。
2. /etc/vold.conf ファイル内の次の行でテキストの初めに # マークを挿入し、コメント扱いにします。

```
# use rmdisk drive /dev/rdisk/c*s2 dev_rmdisk.so rmdisk%d
```

3. システムをリブートします。

```
# init 6
```

リムーバブルメディアのフォーマット (rmformat)

リムーバブルメディアをフォーマットするには、`rmformat` コマンドを使用します。次の種類のフロッピーディスクもこのコマンドでフォーマットできます。

- 倍密度 — 720K バイト (3.5 インチ)
- 高密度 — 1.44M バイト (3.5 インチ)

`rmformat` コマンドは非スーパーユーザーのユーティリティであり、書き込み可能なリムーバブルメディアをフォーマットおよび保護できます。`rmformat` コマンドには3つのフォーマットオプションがあります。

- `quick` — このオプションは、トラックを検証せずに、あるいは、検証するトラックを制限して、リムーバブルメディアをフォーマットします。
- `long` — このオプションは、リムーバブルメディアを完全にフォーマットします。このオプションを使用するデバイスによっては、ドライブ自身による媒体全体の検証も含まれる場合があります。
- `force` — このオプションは、ユーザーへの確認なしに、リムーバブルメディアを完全にフォーマットします。パスワードによる保護機能を備えた媒体では、このオプションはフォーマットを行う前にパスワードをクリアします。この機能はパスワードを忘れてしまったときに便利です。パスワードによる保護機能を備えていない媒体では、`long` オプションのフォーマットが行われます。

▼ リムーバブルメディアをフォーマットするには (rmformat)

rmformat コマンドはリムーバブルメディアをフォーマットして、デフォルトで、パーティション 0 とパーティション 2 (媒体全体) の 2 つのパーティションを媒体上に作成します。

1. ボリュームマネージャが動作していることを確認します。つまり、デバイス名のニックネームを使用できます。

```
$ ps -ef | grep vold
root    212      1  0   Nov 03 ?           0:01 /usr/sbin/vold
```

リムーバブルメディアデバイス名を決定する方法とボリュームマネージャを (動作していない場合) 起動する方法については、『Solaris のシステム管理 (第 1 巻)』を参照してください。

2. リムーバブルメディアをフォーマットします。

```
$ rmformat -F [ quick | long | force ] device-name
```

rmformat のフォーマットオプションについては、前出の節を参照してください。

rmformat の出力が不良ブロックを示している場合、下記の手順を参照して、不良ブロックを修復してください。

3. (オプション) リムーバブルメディアに、Solaris 環境で使用する 8 文字のラベルを付けます。

```
$ rmformat -b label device-name
```

DOS ラベルを作成する方法については、mkfs_pcfs(1M) のマニュアルページを参照してください。

例 - リムーバブルメディアをフォーマットする

フロッピーディスクをフォーマットするには、次のようにします。

```
$ rmformat -F quick /dev/rdiskette
Formatting will erase all the data on disk.
Do you want to continue? (y/n) y
.....
```

Zip ドライブをフォーマットするには、次のようにします。

```
$ rmformat -F quick zip0
Formatting will erase all the data on disk.
Do you want to continue? (y/n) y
.....
```

▼ UFS または UDFS ファイルシステム用にリムーバブルメディアをフォーマットするには

1. リムーバブルメディアをフォーマットします。

```
$ rmformat -F quick device-name
```

2. (オプション) 代替の **Solaris** パーティションテーブルを作成します。

```
$ rmformat -s slice-file device-name
```

スライスファイルの例は次のようになります。

```
slices: 0 = 0, 30MB, "wm", "home" :
        1 = 30MB, 51MB :
        2 = 0, 94MB, "wm", "backup" :
        6 = 81MB, 13MB
```

代替の Solaris パーティションテーブルを作成する方法については、『Solaris のシステム管理 (第 1 巻)』を参照してください。

3. スーパーユーザーになります。
4. 適切なファイルシステムの種類を決定して、次のうちの 1 つを選択します。
 - a. **UFS** ファイルシステムを作成します。

```
# newfs device-name
```

- b. **UDFS** ファイルシステムを作成します。

```
# mkfs -F udfs device-name
```

例 - UFS ファイルシステム用にリムーバブルメディアをフォーマットする

次の例では、フロッピーディスクをフォーマットして、UFS ファイルシステムを作成します。

```
$ rmformat -F quick /dev/rdiskette
Formatting will erase all the data on disk.
Do you want to continue? (y/n)y
$ su
# newfs /dev/rdiskette
newfs: construct a new file system /dev/rdiskette: (y/n)? y
/dev/rdiskette: 2880 sectors in 80 cylinders of 2 tracks, 18 sectors
1.4MB in 5 cyl groups (16 c/g, 0.28MB/g, 128 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
32, 640, 1184, 1792, 2336,
#
```

▼ PCFS ファイルシステム用にリムーバブルメディアをフォーマットするには

1. リムーバブルメディアをフォーマットします。

```
$ rmformat -F quick device-name
```

2. スーパーユーザーになります。

3. (オプション) 代替の **Solaris** fdisk パーティションテーブルを作成します。

```
# fdisk device-name
```

fdisk パーティションを作成する方法については、『Solaris のシステム管理 (第 1 巻)』を参照してください。

4. PCFS ファイルシステムを作成します。

```
# mkfs -F pcfs device-name
```

例 - PCFS ファイルシステム用にリムーバブルメディアをフォーマットする

次の例では、代替 fdisk パーティションを作成して、PCFS ファイルシステムを作成します。

```
$ rmformat -F quick /dev/rdisk/c0t4d0s2:c
Formatting will erase all the data on disk.
Do you want to continue? (y/n)y
$ su
# fdisk /dev/rdisk/c0t4d0s2:c
# mkfs -F pcfs /dev/rdisk/c0t4d0s2:c
Construct a new FAT file system on /dev/rdisk/c0t4d0s2:c: (y/n)? y
#
```

次の例では、fdisk パーティションを作成せずに、PCFS ファイルシステムを作成します。

```
$ rmformat -F quick /dev/rdiskette
Formatting will erase all the data on disk.
Do you want to continue? (y/n)y
$ su
# mkfs -F pcfs -o nofdisk,size=2 /dev/rdiskette
Construct a new FAT file system on /dev/rdiskette: (y/n)? y
#
```

▼ リムーバブルメディア上の PCFS ファイルシステムをチェックするには

1. スーパーユーザーになります。
2. PCFS ファイルシステムをチェックします。

```
# fsck -F pcfs device-name
```

例 - リムーバブルメディア上の PCFS ファイルシステムを チェックする

```
# fsck -F pcfs /dev/rdisk/c0t4d0s2
** /dev/rdisk/c0t4d0s2
** Scanning file system meta-data
** Correcting any meta-data discrepancies
1457664 bytes.
0 bytes in bad sectors.
0 bytes in 0 directories.
0 bytes in 0 files.
1457664 bytes free.
512 bytes per allocation unit.
2847 total allocation units.
2847 available allocation units.
#
```

▼ リムーバブルメディア上の不良ブロックを修復する には

ドライブが不良ブロック管理をサポートしている場合にのみ、検証中に見つかった不良セクタを `rmformat` コマンドで検証、解析、および修復できます。ほとんどのフロッピーディスクや PCMCIA メモリーカードは不良ブロック管理をサポートしていません。

ドライブが不良ブロック管理をサポートしている場合、不良ブロックを修復するための最大の努力が行われます。それでも不良ブロックを修復できなかった場合、修復に失敗したことを示すメッセージが表示されます。

1. リムーバブルメディア上の不良ブロックを修復します。

```
$ rmformat -c block-numbers device-name
```

`block-numbers` には、前の `rmformat` セッションで獲得したブロック番号を 10 進数、8 進数、または 16 進数形式で指定します。

2. リムーバブルメディアを検証します。

```
$ rmformat -v read device-name
```

リムーバブルメディアへの読み取りまたは書き込み保護とパスワードによる保護の適用

Iomega 媒体 (Zip ドライブや Jaz ドライブなど) には、読み取り保護または書き込み保護を適用し、パスワードを設定することができます。その他の媒体では、パスワードの設定は無しで、書き込み保護を有効または無効にできます。

▼ リムーバブルメディアに書き込み保護を有効または無効にするには

1. 書き込み保護を有効または無効のどちらにするかを決定し、次のうちの 1 つを選択します。

- a. 書き込み保護を有効にします。

```
$ rmformat -w enable device-name
```

- b. 書き込み保護を無効にします。

```
$ rmformat -w disable device-name
```

2. リムーバブルメディアの書き込み保護が有効または無効であることを確認します。

```
$ rmformat -p device-name
```

▼ Iomega 媒体上で読み取りまたは書き込み保護とパスワードを有効または無効にするには

パスワードによる保護機能をサポートしている Iomega 媒体には、最大 32 文字のパスワードを適用できます。このような Iomega 媒体上で読み取り保護または書き込み保護を設定するときは、必ずパスワードを適用する必要があります。このとき、パスワードを提供するように促すプロンプトが表示されます。

パスワード機能をサポートしていないリムーバブルメディア上でパスワードを適用しようとする、警告メッセージが表示されます。

1. 読み取り保護または書き込み保護とパスワード保護を有効または無効のどちらにするかを決定します。

- a. 読み取り保護または書き込み保護を有効にします。

```
$ rmformat -W enable device-name  
Please enter password (32 chars maximum): xxx  
Please reenter password:
```

```
$ rmformat -R enable device-name  
Please enter password (32 chars maximum): xxx  
Please reenter password:
```

- b. 読み取り保護または書き込み保護を無効にし、パスワードを削除します。

```
$ rmformat -W disable device-name  
Please enter password (32 chars maximum): xxx
```

```
$ rmformat -R disable device-name  
Please enter password (32 chars maximum): xxx
```

2. リムーバブルメディアの読み取りまたは書き込み保護が有効または無効であることを確認します。

```
$ rmformat -p device-name
```

例 - 読み取りまたは書き込み保護を有効または無効にする

次の例では、Zip ドライブ上で書き込み保護を有効にし、パスワードを設定します。

```
$ rmformat -W enable /vol/dev/aliases/zip0
Please enter password (32 chars maximum): xxx
Please reenter password: xxx
```

次の例では、Zip ドライブ上で書き込み保護を無効にし、パスワードを削除します。

```
$ rmformat -W disable /vol/dev/aliases/zip0
Please enter password (32 chars maximum): xxx
```

次の例では、Zip ドライブ上で読み取り保護を有効にし、パスワードを設定します。

```
$ rmformat -R enable /vol/dev/aliases/zip0
Please enter password (32 chars maximum): xxx
Please reenter password: xxx
```

次の例では、Zip ドライブ上で読み取り保護を無効にし、パスワードを削除します。

```
$ rmformat -R disable /vol/dev/aliases/zip0
Please enter password (32 chars maximum): xxx
```


デバイスの管理についてのトピック

このトピックでは、Solaris 環境におけるデバイスの管理について記述します。次の章で構成されます。

第 17 章	動的再構成時にアプリケーションをシャットダウンするための独自のスクリプト、およびアプリケーションからデバイスをクリーンに解放するための独自のスクリプトを書くための、新しい RCM スクリプト機能の使用方法について記述します。
103ページの「USB デバイスの概要」	Universal Serial Bus (USB) デバイスの概要、およびそれらが Solaris オペレーティング環境でどのようにサポートされるかについて記述します。
118ページの「USB プリンタのサポート」	Solaris 印刷マネージャを使用した USB プリンタの設定手順について記述します。
第 19 章	動的再構成ソフトウェアに関するトラブルシューティング情報を記述します。

Reconfiguration Coordination Manager (RCM) スクリプト

Solaris 8 4/01 リリースで、Reconfiguration Coordination Manager (RCM) スクリプトの機能が新規に追加されました。Solaris のシステム管理についての全般的な情報は、『Solaris のシステム管理 (第 1 巻)』を参照してください。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「Solaris 8 Reference Manual Collection」には記載されていない新しい情報も提供されています。

RCM スクリプトの概要

システムリソースの動的再構成では、システムを動作させたままで、システムのコンポーネントを再構成することができます。この機能は、Solaris 8 リリースから `cfgadm` コマンドで使用できるようになりました。Reconfiguration Coordination Manager は、システムコンポーネントの動的削除を制御するフレームワークです。RCM の使用により、システムリソースを秩序立てて登録したり解放したりすることができます。

Solaris 8 4/01 リリースでは、動的再構成時にアプリケーションをシャットダウンしたり、アプリケーションからデバイスをクリーンに解放したりする独自のスクリプトを作成できる、新しい RCM スクリプト機能を使用することができるようになりました。再構成の要求がスクリプトによって登録されたリソースに影響を及ぼすも

のである場合、その再構成の要求に回答して RCM フレームワークは自動的にスクリプトを起動させます。

以前は、リソースを動的に削除するには、アプリケーションからリソースを手動で解放する必要がありました。または、`cfgadm` コマンドを `-f` オプションとともに使用して再構成の操作を強制的に実行することもできますが、このオプションでは、アプリケーションが認識されない状態のままになってしまいます。また、アプリケーションからリソースを手動で解放すると、エラーを引き起こすこともよくあります。

RCM スクリプト機能は、動的再構成のプロセスをシンプルにし、制御性を向上させます。RCM スクリプトの作成により、次のことが可能になります。

- デバイスを動的に削除する際に、自動的にデバイスを解放する。また、デバイスがアプリケーションによって開かれている場合、このプロセスがそのデバイスを閉じる。
- システムからデバイスを動的に削除する際に、サイト特有のタスクを行う。

RCM スクリプトとは

RCM スクリプトとは：

- RCM デーモンが実行する、実行可能シェルスクリプト (Perl、sh、csh、ksh)、またはバイナリプログラム。Perl は推奨言語です。
- スクリプトファイル所有者のユーザー ID を使用して、独自のアドレス領域内で実行されるスクリプト。
- システムリソースを動的に再構成するためにユーザーが `cfgadm` コマンドを使用する際、RCM デーモンによって実行されるスクリプト。

RCM スクリプトで何ができるか

デバイスを動的に削除する際、RCM スクリプトを使用してアプリケーションからデバイスを解放することができます。また、デバイスが現在開かれている状態の場合は、RCM スクリプトはそのデバイスを閉じます。

たとえば、テープのバックアップアプリケーション用の RCM スクリプトは、そのテープのバックアップアプリケーションに対して、テープドライブを閉じるよう通知したり、そのテープのバックアップアプリケーションをシャットダウンするよう通知したりすることができます。

RCM スクリプトのプロセスはどのように機能するか

次のようにして、スクリプトを起動できます。

```
$ script-name command [args ...]
```

1つのスクリプトは、次のような基本手順を実行します。

1. コマンド行の引数から RCM コマンドを取得します。
2. そのコマンドを実行します。
3. 結果を、名前 - 値のペアとして stdout (標準出力) に書き込みます。
4. 適切な終了ステータスで終了します。

RCM デーモンは、1度に1スクリプトの1つのインスタンスを実行します。たとえば、1つのスクリプトが実行されている時は、RCM デーモンは最初のスクリプトが終了するまで同じスクリプトを実行しません。

RCM スクリプトのコマンド

次の RCM コマンドは、RCM スクリプト内に含める必要があります。

- `scriptinfo` - スクリプトの情報を収集します。
- `register` - 対象となるものをリソースに登録します。
- `resourceinfo` - リソースの情報を収集します。

次の RCM コマンドは、必要に応じて一部またはすべてを含めることができます。

- `queryremove` - リソースが解放できるかどうかを照会します。
- `preremove` - リソースを解放します。
- `postremove` - リソースを削除した後の通知を発行します。
- `undoremove` - `preremove` で実行されたアクションを元に戻します。

これらの RCM コマンドの詳細な説明は、`rcmscript(4)` のマニュアルページを参照してください。

RCM スクリプトの処理の環境

ユーザーがデバイスを動的に削除する際、RCM デーモンは以下を実行します。

- スクリプト内で識別されるリソースのリスト (デバイス名) を収集するために、スクリプトの `register` コマンドを実行する。

- スクリプトに登録されたリソースが動的削除の操作の影響を受ける場合、リソースの削除を行う前に、スクリプトの `queryremove/preremove` コマンドを実行する。
- 削除の操作が成功した場合は、スクリプトの `postremove` コマンドを実行する。削除の操作が失敗した場合は、RCM デーモンはスクリプトの `undoremove` コマンドを実行する。

RCM スクリプトのタスク

次の節では、アプリケーション開発者およびシステム管理者のための RCM スクリプトのタスクについて記述します。

アプリケーション開発者の RCM スクリプトのタスク

次の表では、RCM スクリプトを作成するアプリケーション開発者のタスクを記載しています。

表 17-1 アプリケーション開発者のための RCM スクリプトのタスクマップ

タスク	説明	手順の参照先
1. アプリケーションが使用するリソースを明確にする。	動的に削除できる可能性がある、アプリケーションが使用するリソース (デバイス名) を明確にする。	<code>cfgadm(1m)</code> のマニュアルページ
2. リソースを解放するためのコマンドを明確にする。	アプリケーションからリソースをクリーンに解放するよう、アプリケーションに通知するためのコマンドを明確にする。	アプリケーションのマニュアル
3. リソースの削除後に実行するコマンドを明確にする。	リソースの削除をアプリケーションに通知するためのコマンドを含める。	<code>rcmscript(4)</code> のマニュアルページ
4. リソースの削除が失敗した場合のコマンドを明確にする。	利用可能なリソースをアプリケーションに通知するためのコマンドを含める。	<code>rcmscript(4)</code> のマニュアルページ

表 17-1 アプリケーション開発者のための RCM スクリプトのタスマップ 続く

タスク	説明	手順の参照先
5. RCM スクリプトを書く。		97ページの「テープのバックアップ用 RCM スクリプトの例」
6. RCM スクリプトをインストールする。	スクリプトを適切なスクリプトのディレクトリに追加する。	95ページの「RCM スクリプトをインストールするには」
7. RCM スクリプトをテストする。	スクリプトのコマンドを手動で実行し、動的再構成の操作を開始して、スクリプトをテストする。	96ページの「RCM スクリプトをテストするには」

システム管理者の RCM スクリプトのタスク

次の表では、サイトにおけるカスタマイズのために RCM スクリプトを作成するシステム管理者のタスクを記載しています。

表 17-2 システム管理者のための RCM スクリプトのタスマップ

タスク	説明	参照先
1. 動的に削除されるリソースを明確にする。	cfgadm -l コマンドの使用により、削除される可能性のあるリソース (デバイス名) を明確にします。	cfgadm(1m) のマニュアルページ
2. 停止されるアプリケーションを明確にする。	アプリケーションをクリーンに停止させるためのコマンドを明確にする。	アプリケーションのマニュアル
3. リソースの削除前および削除後に実行するコマンドを明確にする。	リソースが削除される前と後に実行するアクションを明確にします。	rcmscript(4) のマニュアルページ
4. RCM スクリプトを書く。		97ページの「テープのバックアップ用 RCM スクリプトの例」

表 17-3 RCM スクリプトのディレクトリ 続く

ディレクトリのロケーション	スクリプトのタイプ
<code>/usr/platform/`uname -m`/lib/rcm/scripts</code>	特定のハードウェアクラス用のスクリプト
<code>/usr/lib/rcm/scripts</code>	任意のハードウェア用のスクリプト

▼ RCM スクリプトをインストールするには

1. スーパーユーザーになります。
2. スクリプトを、表 17-3 に記載されているように適切なディレクトリにコピーします。
たとえば、次のようにします。

```
# cp SUNW,sample.pl /usr/lib/rcm/scripts
```

3. スクリプトのユーザー ID とグループ ID を希望する値に変更します。
たとえば、次のようにします。

```
# chown user:group /usr/lib/rcm/scripts/SUNW,sample.pl
```

4. SIGHUP を RCM デーモンに送信します。

```
# pkill -HUP -x -u root rcm_daemon
```

▼ RCM スクリプトを削除するには

1. スーパーユーザーになります。
2. RCM スクリプトのディレクトリから、スクリプトを削除します。
たとえば、次のようにします。

```
# rm /usr/lib/rcm/scripts/SUNW,sample.pl
```

3. SIGHUP を **RCM** デーモンに送信します。

```
# pkill -HUP -x -u root rcm_daemon
```

▼ RCM スクリプトをテストするには

1. スクリプトを実行する前に、コマンド行のシェルで RCM_ENV_FORCE などの環境変数を設定します。

たとえば、Korn シェルでは、以下を使用します。

```
$ export RCM_ENV_FORCE=TRUE
```

2. コマンド行からスクリプトのコマンドを手動で実行して、スクリプトをテストします。

たとえば、次のようにします。

```
$ script-name scriptinfo  
$ script-name register  
$ script-name preremove resource-name  
$ script-name postremove resource-name
```

3. スクリプト内の各 **RCM** スクリプトコマンドが stdout (標準出力) に適切な出力をプリントすることを確認します。
4. スクリプトを適切なスクリプトのディレクトリにインストールします。
詳細は、95ページの「RCM スクリプトをインストールするには」を参照してください。
5. 動的な削除の操作を開始して、スクリプトをテストします。

たとえば、スクリプトがデバイス /dev/dsk/c1t0d0s0 を登録していると仮定します。次のコマンドを実行してみます。

```
$ cfgadm -c unconfigure c1::dsk/c1t0d0
$ cfgadm -f -c unconfigure c1::dsk/c1t0d0
$ cfgadm -c configure c1::dsk/c1t0d0
```



注意 - これらのコマンドはシステムの状態を変えたり、システムに障害を引き起こしたりする可能性もあるため、これらのコマンドについて熟知しておく必要があります。

テープのバックアップ用 RCM スクリプトの例

次の例では、テープのバックアップ用に RCM スクリプトを使用する方法を示します。

テープのバックアップ用 RCM スクリプトは何を行うか

テープのバックアップ用 RCM スクリプトは、次の手順を実行します。

1. RCM コマンドのディスパッチテーブルをセットアップします。
2. 指定した RCM コマンドに対応するディスパッチルーチンを呼び出し、実装されていない RCM コマンドについてはステータス 2 で終了します。
3. `scriptinfo` セクションをセットアップします。

```
rcm_script_func_info=Tape backup appl script for DR
```

4. すべてのテープドライブデバイス名を `stdout` (標準出力) に出力して、システム内のすべてのテープドライブを登録します。

```
rcm_resource_name=/dev/rmt/$f
```

エラーが発生した場合は、エラー情報を stdout に出力します。

```
rcm_failure_reason=$errmsg
```

5. テープデバイスのリソース情報を設定します。

```
rcm_resource_usage_info=Backup Tape Unit Number $unit
```

6. バックアップアプリケーションがそのデバイスを使用しているかどうかをチェックして、preremove 情報をセットアップします。バックアップアプリケーションがそのデバイスを使用していない場合は、動的再構成の操作はそのまま続きます。バックアップアプリケーションがそのデバイスを使用している場合は、スクリプトは RCM_ENV_FORCE をチェックします。RCM_ENV_FORCE が FALSE に設定されている場合は、スクリプトは動的再構成の操作を否定し、次のようなメッセージを出力します。

```
rcm_failure_reason=tape backup in progress pid=...
```

RCM_ENV_FORCE が TRUE に設定されている場合は、バックアップアプリケーションが停止され、再構成の操作が進められます。

テープバックアップ再構成のシナリオの結果

次に、RCM スクリプトを使用せずに、cfgadm コマンドを使用してテープデバイスを削除する場合に起こる結果を示します。

- cfgadm コマンドを使用し、バックアップアプリケーションがテープデバイスを使用していない場合、操作は成功します。
- cfgadm コマンドを使用し、バックアップアプリケーションがテープデバイスを使用している場合、操作は失敗します。

次に、RCM スクリプトを使用して、cfgadm コマンドを使用してテープデバイスを削除する場合に起こる結果を示します。

- cfgadm コマンドを使用し、バックアップアプリケーションがテープデバイスを使用していない場合、操作は成功します。
- -f オプション無しで cfgadm コマンドを使用し、バックアップアプリケーションがテープデバイスを使用している場合、操作は失敗し、次のようなエラーメッセージが出力されます。

```
tape backup in progress pid=...
```

- cfgadm -f コマンドを使用し、バックアップアプリケーションがテープデバイスを使用している場合、スクリプトがバックアップアプリケーションを停止させ、cfgadm 操作は成功します。

例 - テープのバックアップ用の RCM スクリプト

```
#!/usr/bin/perl -w
#
# A sample site customization RCM script.
#
# When RCM_ENV_FORCE is FALSE this script indicates to RCM that it cannot
# release the tape drive when the tape drive is being used for backup.
#
# When RCM_ENV_FORCE is TRUE this script allows DR removing a tape drive
# when the tape drive is being used for backup by killing the tape
# backup application.
#

use strict;

my ($cmd, %dispatch);
$cmd = shift(@ARGV);
# dispatch table for RCM commands
%dispatch = (
    "scriptinfo"    =>    \&do_scriptinfo,
    "register"      =>    \&do_register,
    "resourceinfo" =>    \&do_resourceinfo,
    "queryremove"  =>    \&do_preremove,
    "preremove"    =>    \&do_preremove
);

if (defined($dispatch{$cmd})) {
    &{$dispatch{$cmd}};
} else {
    exit (2);
}

sub do_scriptinfo
{
    print "rcm_script_version=1\n";
    print "rcm_script_func_info=Tape backup appl script for DR\n";
    exit (0);
}

sub do_register
{
    my ($dir, $f, $errmsg);

    $dir = opendir(RMT, "/dev/rmt");
    if (!$dir) {
        $errmsg = "Unable to open /dev/rmt directory: $!";
        print "rcm_failure_reason=$errmsg\n";
        exit (1);
    }

    while ($f = readdir(RMT)) {
        # ignore hidden files and multiple names for the same device
        if (($f !~ /\./) && ($f =~ /^[0-9]+$/)) {
            print "rcm_resource_name=/dev/rmt/$f\n";
        }
    }
}
```

(続く)

```

    }

    }

    closedir(RMT);
    exit (0);
}
sub do_resourceinfo
{
    my ($rsrc, $unit);

    $rsrc = shift(@ARGV);
    if ($rsrc =~ /^\/dev\/rmt\/([0-9]+)$/) {
        $unit = $1;
        print "rcm_resource_usage_info=Backup Tape Unit Number $unit\n";
        exit (0);
    } else {
        print "rcm_failure_reason=Unknown tape device!\n";
        exit (1);
    }
}

sub do_preremove
{
    my ($rsrc);

    $rsrc = shift(@ARGV);

    # check if backup application is using this resource
    #if (the backup application is not running on $rsrc) {
        # allow the DR to continue
        #    exit (0);
    #}
    #
    # If RCM_ENV_FORCE is FALSE deny the operation.
    # If RCM_ENV_FORCE is TRUE kill the backup application in order
    # to allow the DR operation to proceed
    #
    if ($ENV{RCM_ENV_FORCE} eq 'TRUE') {
        if ($cmd eq 'preremove') {
            # kill the tape backup application
        }
        exit (0);
    } else {
        #
        # indicate that the tape drive can not be released
        # since the device is being used for backup by the
        # tape backup application
        #
        print "rcm_failure_reason=tape backup in progress pid=...\n"
;
        exit (3);
    }
}

```

(続く)

```
}
```


USB デバイスの管理

Solaris 8 4/01 ソフトウェアリリースで、USB デバイスの管理についての章が更新されました。詳細は、次の節を参照してください。

- 103ページの「USB デバイスの概要」
- 118ページの「USB プリンタのサポート」

Solaris のデバイス管理についての全般的な情報については、『Solaris のシステム管理 (第 1 巻)』の「デバイスの管理 (概要)」を参照してください。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「Solaris 8 Reference Manual Collection」には記載されていない新しい情報も提供されています。

USB デバイスの概要

Universal Serial Bus (USB) は PC 業界で開発された、周辺機器 (キーボード、マウス、プリンタなど) をシステムに接続するための低コストのソリューションです。

USB コネクタは 1 方向 1 種類のケーブルだけに適合するように設計されています。デバイスはハブデバイスに接続できます。ハブデバイスとは、複数のデバイス (他のハブデバイスも含む) を接続するためのものです。USB が設計された主な目的は、デバイスごとに異なる何種類ものコネクタを減らすことです。つまり、システムの背面パネルの混雑を軽減することです。他にも、USB デバイスを使用すると、次のような利点があります。

- USB デバイスはホットプラグ可能。詳細は、117ページの「USB デバイスのホットプラグ」を参照してください。
- 最大 126 台のデバイスを Solaris 環境でサポート
- 最大 12M ビット/秒のデータ転送速度をサポート
- ロースピードデバイス (1.5M ビット/秒) からフルスピードデバイス (12M ビット/秒) までをサポート
- 低コストの外部ハブを追加するだけで簡単にバスを拡張可能。また、ハブとハブを接続して、ツリートポロジを構築できます。

Sun Microsystems は次のようなシステムで USB デバイスをサポートします。

- Solaris 8 10/00 リリースが動作している Sun Blade™ 100 と Sun Blade 1000 システムは USB デバイスをサポートします。
- Sun Ray™ システムも USB デバイスをサポートします。
- Solaris 8 (Intel 版) が動作している IA システムは、キーボード、マウス、一部の大容量ストレージデバイス (Zip ドライブなど) について USB サポートを提供します。詳細は、scsa2usb (7D) のマニュアルページを参照してください。

次の表に、Solaris 環境でサポートされる USB デバイスを示します。

USB デバイス	サポートされるシステム
キーボードとマウス	ohci (7D) コントローラに基づいた、Sun の USB がサポートされている SPARC システム uhci (7D) コントローラに基づいた、USB バスを持つ IA システム オンボードの USB コントローラだけがサポートされます。プラグイン式のホストコントローラ PCI カードはサポートされません。
大容量ストレージ	SPARC と IA
プリンタ	SPARC と IA
ハブ	SPARC と IA

よく使用される USB 関連の略語

次の表に、Solaris 環境で使用される USB の略語について説明します。USB の構成要素と略語についての詳細は、<http://www.usb.org> を参照してください。

略語	説明
USB	Universal Serial Bus
USBA	Universal Serial Bus Architecture (Solaris)
USBAI	USBA Client Driver Interface (Solaris)
HCD	USB ホストコントローラドライバ

USB バスの説明

USB 仕様は、ライセンス料無しで誰でも入手できます。USB 仕様は、バスとコネクタの電気的および機械的なインタフェースを定義します。

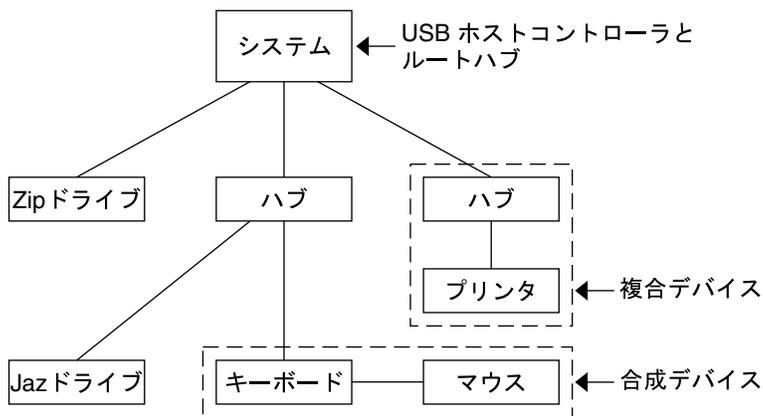


図 18-1 USB 物理デバイスの階層

USB が採用するトポロジでは、ハブが USB デバイスに接続点を提供します。ホストコントローラには、システム内のすべての USB ポートの原点となるルートハブが含まれます。ハブについての詳細は、109ページの「USB ホストコントローラとルートハブ」を参照してください。

上記の例では、1つのシステムが3つのアクティブな USB ポートを持っています。1番目の USB ポートは Zip ドライブに接続されています。この Zip ドライブにはハブが埋め込まれていないため、他のデバイスは接続できません。2番目の USB ポートはハブに接続されており、このハブには Jaz ドライブと合成デバイス (キーボード

とマウス)が接続されています。このキーボードにはハブが埋め込まれており、そのハブにマウスが接続されています。

次の表に、上記の例におけるデバイスのデバイスツリーパス名を示します。

Zip ドライブ	/pci@1f,4000/usb@5/storage@1
キーボード	/pci@1f,4000/usb@5/hub@2/keyboard@1
マウス	/pci@1f,4000/usb@5/hub@2/mouse@2
Jaz ドライブ	/pci@1f,4000/usb@5/hub@2/storage@3
プリンタ	/pci@1f,4000/usb@5/hub@3/printer@1

USB デバイスとドライバ

USB デバイスはいくつかのデバイスクラスに分類されます。各デバイスクラスは対応するドライバを持っています。1つのクラス内のデバイスは同じデバイスドライバで管理されます。しかし USB 仕様では、特定のクラスに属さない、ベンダー固有のデバイスも許可しています。同じような属性とサービスを持つデバイスはグループ化されます。

Human Interface Device (HID) クラスには、ユーザーが制御するデバイス (キーボード、マウス、ジョイスティックなど) が含まれます。Communication Device クラスには、電話に接続するデバイス (モデムや ISDN インタフェースなど) が含まれます。その他にも、Audio、Monitor、Printer、Storage Device などのデバイスクラスがあります。各 USB デバイスはデバイスのクラスを表す記述子を持っています。デバイスクラスは、そのメンバーが構成とデータ転送に関してどのように動作すべきかを指定します。クラス情報についての詳細は、<http://www.usb.org> サイトを参照してください。

Solaris USB Architecture (USBA)

USB デバイスは2つのレベルのデバイスツリーノードとして表現されます。1つのデバイスノードが USB デバイス全体を表し、1つまたは複数の子インタフェースノードがデバイス上にある個々の USB インタフェースを表します。特殊なケースとして、デバイスノードとインタフェースノードが1つのノードに結合される場合もあります。

ドライバのバインドは互換性のある名前属性の使用によって実現されます。詳細は、『IEEE 1275 USB binding』の 3.2.2.1 項と『Writing Device Drivers』を参照してください。ドライバは、デバイス全体にバインドしてすべてのインタフェースを制御することも、1つのインタフェース(キーボードやマウスなど)だけにバインドすることも可能です。デバイス全体にバインドするドライバがベンダーにもクラスにも存在しない場合、汎用 USB マルチインタフェースドライバがデバイスレベルのノードにバインドされます。このドライバは、互換性のある名前属性(『IEEE 1275 USB binding』の 3.2.2.1 項で定義されている)を使用して、各インタフェースにドライバをバインドしようとします。

図 18-1 には、複合デバイス(ハブとプリンタ)の例が示されています。ハブとプリンタは両方とも同じプラスチック製のケースに入っていますが、異なる USB バスアドレスを持ちます。また、図 18-1 には、合成デバイスの例も示されています。キーボードとコントローラは同じプラスチック製のケースに入っていますが、同じ USB バスアドレスを持ちます。この例では、1本のケーブルが USB マウスを合成デバイス(キーボードとコントローラ)に接続しています。

Solaris USB Architecture (USBA) は、USB 1.0 および 1.1 の仕様に加え、Solaris ドライバ条件に準拠しています。USBA モデルは Sun Common SCSI Architecture (SCSA) に似ています。USBA とは、汎用 USB トランスポート層という概念をクライアントドライバに提供する薄い層のことです。

SCSA と USBA の違いは、SCSA がバスを検査するときに .conf ファイルを使用するのに対して、USB ハブドライバは自己検査 nexus ドライバであることです。

Solaris 環境における USB について

次の節では、Solaris 環境における USB について知っておく必要のある情報を説明します。

USB キーボードとマウス

Solaris 環境では複数の USB キーボードとマウスをサポートしていないため、USB キーボードとマウスは常に 1つだけシステムに接続するようにしてください。次の項目の説明も参照してください。

- バス上のいずれかに接続されたキーボードおよびマウスは、コンソールキーボードおよびマウスとして構成されます。キーボードおよびマウスがルートハブ上がない場合、システムのブートにかかる時間が長くなります。

- コンソールキーボードおよびマウスは、システムのリブート後であればいつでも別のハブへ移動することができます。しかし、リブート中や ok プロンプトが出ている間は移動できません。キーボードおよびマウスは、プラグインした後は再び完全に機能します。
- **SPARC** のみ - USB キーボードの電源キーと Sun タイプ 5 キーボードの電源キーの動作は異なります。USB キーボードでは、「SUSPEND/SHUTDOWN」キーを使用してシステムを中断またはシャットダウンすることができますが、システムの電源を入れることはできません。
- Sun 社製以外の USB キーボードでは、キーパッドの左側にある機能は使用できません。
- 複数のキーボードはサポートされません。
 - キーボードは認識され、使用できますが、コンソールキーボードとしては認識されません。
 - ブート時に最初に認識されたキーボードがコンソールキーボードとなります。このため、ブート時に複数のキーボードがプラグインされていると混乱の原因となります。
 - コンソールキーボードを取り外した場合、次に利用可能な USB キーボードがコンソールキーボードにはなりません。次にホットプラグされるキーボードがコンソールキーボードになります。
- 複数のマウスはサポートされません。
 - マウスは認識され、使用できますが、コンソールマウスとしては認識されません。
 - ブート時に最初に認識されたマウスがコンソールマウスとなります。この認識のため、ブート時に複数のマウスがプラグインされていると混乱の原因となります。
 - コンソールマウスを取り外した場合、次に利用可能な USB マウスがコンソールマウスにはなりません。次にホットプラグされるマウスがコンソールマウスになります。
- Sun 社製以外の合成キーボードを PS/2 マウスと使用する場合、このキーボードがブート時に最初に認識されると、PS/2 マウスがプラグインされていなくても、このキーボードとマウスがコンソールキーボードとマウスになります。つまり、別の USB マウスがシステムにプラグインされていても、コンソールマウスとして構成されないので機能できません。

- 2 ボタンと 3 ボタンのマウスだけがサポートされます。ホイール付きマウスは 1 ボタンのマウスのように動作します。3 ボタンよりも多いマウスは 3 ボタンのマウスのように動作します。

USB ホストコントローラとルートハブ

USB ハブは次のことを行います。

- ポートにおけるデバイスの取り付けと取り外しのモニタ
- ポートにおける個々のデバイスの電源管理
- ポートへの電源の制御

USB ホストコントローラはルートハブという埋め込みハブを持っています。背面パネルに見えるポートはルートハブのポートです。USB ホストコントローラは次のことを行います。

- USB バスの指示。個々のデバイスはバスの調整はできません。
- デバイスによって決定されるポーリング間隔による、デバイスのポーリング。デバイスはポーリング間隔 (時間) の考慮に十分なバッファを持っていると想定されます。
- USB ホストコントローラとそれに接続されているデバイス間でのデータの送信。ピアツーピア通信はサポートされません。

USB ハブデバイス

- SPARC と IA のどちらのシステムにおいても、ハブを 4 段を超えて多段接続してはなりません。SPARC システムでは、Open Boot PROM (OBP) は 4 段を超えるデバイスを正確に認識できません。
- バス電源供給方式のハブは多段接続してはなりません。つまり、バス電源供給方式のハブを別のバス電源供給方式のハブに接続してはなりません。バス電源供給方式のハブは独自の電源を持っていません。USB フロッピーディスクデバイスはすべての電源をバスから取り入れるため、バス電源供給方式のハブ上では機能しません。

USB ストレージデバイス

Solaris 8 10/00 リリースから、USB の Zip、Jaz、Clik!、SmartMedia、CompactFlash、および ORB などのリムーバブル大容量ストレージデバイスがサ

ポートされるようになりました。Solaris 環境でサポートされるデバイスの完全なリストについては、`scsa2usb(7D)` のマニュアルページを参照してください。

これらのデバイスは、ボリューム管理を使用しても使用しなくても管理することができます。ボリューム管理を実行している状態でのデバイス管理についての情報は、`vold(1M)` のマニュアルページを参照してください。

vold を実行している状態で **USB** 大容量ストレージデバイスを管理する

Solaris 共通デスクトップ環境 (CDE) が実行されている場合は、USB リムーバブル大容量ストレージデバイスは、CDE ファイルマネージャのコンポーネントであるリムーバブルメディア・マネージャによって管理されます。CDE ファイルマネージャについての詳細は、`dtfile(1)` のマニュアルページを参照してください。

注 - この節に記載されているマニュアルページを表示するには、`MANPATH` 変数に `/usr/dt/man` を含める必要があります。これらのコマンドを使用するには、パスに `/usr/dt/bin` を含め、さらに CDE を実行している必要があります。また、これらのコマンドをリモートで使用する場合は、`DISPLAY` 変数を設定する必要があります。

次の表に、リムーバブルメディア・マネージャが CDE 環境からストレージデバイスを管理する際に使用するコマンドを示します。

コマンド	タスク
<code>sdtmedia_format(1)</code>	USB デバイスのフォーマットおよびラベル付けを行う
<code>sdtmedia_prop(1)</code>	デバイスのプロパティを表示する
<code>sdtmedia_prot(1)</code>	デバイスのアクセス権を変更する
<code>sdtmedia_slice(1)</code>	デバイス上のスライスを作成または変更する

USB デバイスのフォーマットが終了すると、通常は `/rmdisk/label` ディレクトリの下にマウントされます。リムーバブルストレージデバイスの構成についての詳細

は、`rmmount.conf(4)` または `vold.conf(4)` のマニュアルページを参照してください。

以下の手順は、ボリューム管理を実行している状態で USB 大容量ストレージデバイスを管理する方法を示しています。デバイスノードは `/vol/dev` ディレクトリの下に作成されます。詳細は、`scsa2usb(7D)` のマニュアルページを参照してください。以下の手順では、ホットプラグ可能な USB 大容量ストレージデバイスを追加および取り外す方法についても記載しています。デバイスのホットプラグとは、オペレーティングシステムをシャットダウンすることなくあるいはシステムの電源を切ることなく、デバイスを追加したり取り外したりすることを指します。

▼ vold を実行している状態で USB 大容量ストレージデバイスをマウントまたはマウント解除するには

1. すべてのリムーバブル大容量ストレージデバイス (**USB** 大容量ストレージデバイスを含む) について、デバイスの別名を表示します。

```
$ eject -n
.
.
.
rmdisk0 -> /vol/dev/rdisk/c4t0d0/clik40      (Generic USB storage)
cdrom0 -> /vol/dev/rdisk/c0t6d0/audio_cd    (Generic CD device)
zip1 -> /vol/dev/rdisk/c2t0d0/fat32        (USB Zip device)
zip0 -> /vol/dev/rdisk/c1t0d0/zip100       (USB Zip device)
jaz0 -> /vol/dev/rdisk/c3t0d0/jaz1gb       (USB Jaz device)
```

2. 上記のようにリストされたデバイス別名を使用して、**USB** 大容量ストレージデバイスをマウントします。

```
$ volrmmount -i device-alias
```

次の例では、USB Jaz ドライブを `/rmdisk/jaz0` の下にマウントします。

```
$ volrmmount -i jaz0
```

3. **USB** ストレージデバイスをマウント解除します。

```
$ volrmmount -e device-alias
```

次の例では、USB Zip ドライブを /rmdisk/zip0 からマウント解除します。

```
$ volrmmount -e zip0
```

4. **USB** デバイスを、汎用の **USB** ドライブから取り出します。

```
$ eject device-alias
```

次に例を示します。

```
$ eject rmdisk0
```

注 - eject コマンドは、デバイスがまだマウント解除されていない場合は、そのデバイスのマウント解除も行います。このコマンドはまた、そのデバイスにアクセスするアクティブなアプリケーションがある場合は、それらを終了させます。

▼ vold の実行中にホットプラグ可能な USB 大容量ストレージデバイスを取り外すには

以下の手順では、vold の実行中にホットプラグ可能な USB デバイスを取り外す例として、Zip ドライブを使用しています。

1. デバイスをマウント解除します。

```
$ volrmmount -e zip0
```

2. (オプション) そのデバイスを使用しているアクティブなアプリケーションがある場合は、それらを終了させます。

3. デバイスを取り出します。

```
$ eject zip0
```

4. スーパーユーザーになり、vold を停止します。

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

5. **USB** 大容量ストレージデバイスを取り外します。
6. vold を開始します。

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

▼ vold の実行中にホットプラグ可能な USB 大容量ストレージデバイスを追加するには

以下の手順は、vold の実行中にホットプラグ可能な USB デバイスを追加する方法を示します。

1. **USB** 大容量ストレージデバイスを追加します。
2. vold を再起動します。

```
# pkill -HUP vold
```

3. デバイスが追加されたことを確認します。

```
$ ls device-alias
```

vold を使用せずに **USB** 大容量ストレージデバイスを管理する

USB 大容量ストレージデバイスは、ボリュームマネージャ (vold) を使用せずに管理することもできます。ボリュームマネージャを使用しないようにする方法は、次の 2 とおあります。

- 次のコマンド実行して、vold を停止させます。

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

- `vold` は実行した状態にしておきますが、USB 大容量ストレージデバイスを `vold` に登録しないようにします。以下に示すように `/etc/vold.conf` ファイル内の次の行で `#` マークを挿入してコメント扱いにし、USB 大容量ストレージデバイスのボリュームマネージャの登録を削除します。

```
# use rmdisk drive /dev/rdisk/c*s2 dev_rmdisk.so rmdisk%d
```

上記のようにコメント扱いにした後、`vold` を再度開始します。

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



注意 - この行をコメント扱いにすると、そのシステム内に他の SCSI、ATAPI Zip、または Jaz リムーバブルデバイスがある場合、それらのデバイスの `vold` の登録も無効になります。

詳細は、`vold.conf(4)` のマニュアルページを参照してください。

以下の手順は、`vold(1M)` を使用せずに USB 大容量ストレージデバイスを管理する方法を示しています。デバイスノードは、文字型デバイスについては `/dev/rdisk` ディレクトリ、ブロック型デバイスについては `/dev/dsk` ディレクトリの下に作成されます。詳細は、`scsa2usb(7D)` のマニュアルページを参照してください。

▼ `vold` を使用せずに USB 大容量ストレージデバイスをマウントまたはマウント解除するには

1. スーパーユーザーになります。
2. USB 大容量ストレージデバイスをマウントします。

```
# mount -F fs-type /dev/dsk/cntndnsn /mount-point
```

デバイスが読み取り専用の場合は、このコマンドは失敗します。CD-ROM デバイスには次のコマンドを使用します。

```
# mount -F fs-type -o ro /dev/dsk/cntndnsn /mount-point
```

たとえば、次のようにします。

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /mnt
```

3. **USB** 大容量ストレージデバイスをマウント解除します。

```
# umount /mount-point
```

4. デバイスを取り出します。

```
# eject /dev/[r]dsk/cntndr1sn
```

▼ **vold** を使用せずにホットプラグ可能な **USB** 大容量ストレージデバイスを取り外すには

以下の手順は、**vold** を使用せずにホットプラグ可能な **USB** デバイスを取り外す方法を示しています。

1. スーパーユーザーになります。
2. ホットプラグ可能な **USB** デバイスを取り外します。
 - a. デバイスをマウント解除します。

```
# umount /mount-point
```

- b. (オプション) そのデバイスを使用しているアクティブなアプリケーションがある場合は、それらを停止します。
- c. デバイスを取り外します。

▼ **vold** を使用せずにホットプラグ可能な **USB** 大容量ストレージデバイスを追加するには

以下の手順は、**vold** を使用せずにホットプラグ可能な **USB** デバイスを追加する方法を示しています。

1. ホットプラグ可能な **USB** デバイスを **USB** ポートに追加します。
2. **USB** デバイスが追加されたことを確認します。

```
$ ls /dev/rdisk/cntndnsn
```

SPARC: USB CD 上にデータ作成するまたは USB CD からデータを取得するには

cdrw コマンドを使用して、オーディオ CD 上にデータを作成したりオーディオ CD からデータを取得したりできます。cdrw コマンドは、Software Supplement for the Solaris 8 1/01 Operating Environment CD (サブリメント CD) で入手できます。

- SCSI、ATAPI、USB の CD デバイスがサポートされます。現在のところ、Sun がサポートする CD-RW デバイスは、Sony Spres USB CD-RW のみです。
- CD-R や CD-RW ドライブは、MMC 準拠のものでなければなりません。

このコマンドの使用については、『Solaris on Sun Hardware Reference Manual Supplement』の cdrw のマニュアルページを参照してください。

▼ USB CD 上へデータを作成するまたは USB CD からデータを取得するための準備を行うには

cdrw コマンドは、vold が実行されている状態でも実行されていなくても機能することができます。詳細は、cdrw(1) および mkisofs(1M) のマニュアルページを参照してください。

1. **CD** を **CD-RW** デバイスに挿入します。

CD は、そのデバイスが読むことができるものであれば、どんな種類の CD でもかまいません。

2. デバイスのリストを表示して、その **CD-RW** ドライブが正しく接続されていることを確認します。

```
# cdwr -l
Node | Connected Device | Device type
-----+-----+-----
/dev/rdsk/c0t0d0s2 | SONY CD-RW CRX120E 1.0k | CD Reader/Writer
```

3. (オプション) リスト内にそのドライブが表示されていない場合は、システムがそのデバイスを認識するよう再構成ブートを行う必要があることもあります。

```
# touch /reconfigure
# init 6
```

SPARC のみ: USB 電源管理

システムが電源管理を有効にしている場合、USB のフレームワークはすべてのデバイスを電源管理しようと最大限に努力します。USB デバイスの電源管理には、デバイスが接続されているポートのハブドライバによる中断も含まれます。リモートウェイクアップ (呼び起こし) をサポートするかしらないかは、デバイスによって異なります。デバイスがリモートウェイクアップをサポートしている場合は、イベントの発生時 (たとえば、マウスが移動したときなど) に、接続されているハブをウェイクアップします。アプリケーションが入出力を送信した場合も、ホストシステムはデバイスをウェイクアップできます。

リモートウェイクアップ機能がサポートされている場合、すべての HID (キーボードやマウスなど)、ハブ、およびストレージデバイスは、デフォルトで電源管理されます。USB プリンタが電源管理されるのは、2つの印刷ジョブ間だけです。

電源消費を削減するために電源管理を行う場合は、まず、USB リーフデバイスの電源が切断され、しばらくしてから、親ハブの電源が切断されます。当該ハブのポートに接続されているすべてのデバイスの電源が切断されると、しばらくしてから、ハブの電源が切断されます。最も効率的な電源管理は、あまり多くのハブを多段接続しないことです。

USB デバイスのホットプラグ

USB デバイスは、プラグインするとすぐにシステムのデバイス階層に表示されます (`prtconf (1M)` コマンドで確認)。また、デバイスが使用中でない限り、USB デバイスを取り外すとシステムのデバイス階層から消えます。

使用中の USB デバイスを取り外した場合、ホットプラグの動作は少しかけ異なります。使用中の USB デバイスを取り外した場合、デバイスノードは残り、このデバイスを制御しているドライバはデバイス上のすべての動作を停止します。それ以降、このデバイスに発行される新しい入出力動作はエラーで戻されます。

このような場合、システムは元のデバイスを接続するようにユーザーにプロンプトを表示します。間違っ使用中の USB デバイスを取り外してしまった場合は、次のようにして回復します。

1. 元のデバイスを同じポートに接続します。
2. そのデバイスを使用しているアプリケーションを停止します。
3. デバイスを取り外します。

元のデバイスが再びプラグインされるまで、USB ポートは使用できません。デバイスが使用できない場合は、USB ポートは次にリブートするまで使用できません。

注 - アクティブな、つまり開いているデバイスを削除すると、データの整合性が損なわれる可能性があります。デバイスを取り外す前には、必ず、デバイスを閉じるようにしてください。ただし、コンソールキーボードとマウスは例外で、アクティブなときでも移動することができます。

USB ケーブル

市販されている USB ケーブルエクステンダは絶対に使用しないでください。デバイスを接続するときは、必ず、ハブと十分な長さのあるケーブルを使用してください。USB デバイスを接続するときは、必ず、フルレート (12M ビット/秒) の 20/28 AWG ケーブルを使用してください。

USB プリンタのサポート

Solaris 8 10/00 リリースから、Solaris の印刷マネージャを使用して USB ポート付きの SPARC システムに接続されている USB プリンタをセットアップできるようになりました。Solaris 8 4/01 リリースからは、IA システムにも USB プリンタをセットアップできるようになりました。

USB プリンタ用の新しい論理デバイス名は次のとおりです。

```
/dev/printers/[0...N]*
```

したがって、USB プリンタをプリンタサーバーに追加するときは、「新しいローカルプリンタを設定」画面の「プリンタポート」で、上記デバイスの 1 つを USB プリ

ンタ用に選択します。Solaris 印刷マネージャを使用してプリンタを設定する方法についての詳細は、『Solaris のシステム管理 (第 2 巻)』を参照してください。

新しい Solaris USB プリンタドライバは USB プリンタクラス準拠のプリンタをすべてサポートしますが、推奨される PostScript™ プリンタについては、usbprn(7D) のマニュアルページを参照してください。

usbprn ドライバは PostScript 以外のプリンタ (つまり、GhostScript などの Sun 社製以外の PostScript 変換パッケージを使用するプリンタ) にも準拠しています。変換パッケージは Solaris 8 Software Companion CD から入手できます。Solaris 8 Software Companion CD の入手方法については、<http://www.sun.com/software/solaris/binaries/package.html> を参照してください。

USB プリンタのホットプラグに関する情報と注意については、usbprn(7D) のマニュアルページの「NOTES」と「DIAGNOSTICS」の節を参照してください。

動的再構成に関する問題のトラブルシューティング

Solaris 8 1/01 リリースで動的再構成ソフトウェアが拡張されました。次に記述する情報は、『Solaris のシステム管理 (第 1 巻)』の「デバイスの構成」の章に記載されている動的再構成に関する問題への対処方法についての情報を補足するものです。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「Solaris 8 Reference Manual Collection」には記載されていない新しい情報も提供されています。

新しい動的再構成のエラーメッセージ

動的再構成ソフトウェアが拡張され、動的再構成に関する問題への対処方法 (トラブルシューティング) が改良されました。

cfgadm コマンドを使用してスワップデバイスや専用のダンプデバイスなどのシステムリソースを削除しようとした時に、そのシステムリソースがまだアクティブである場合は、エラーメッセージが表示されます。

以下は、エラーメッセージについての説明です。

エラーメッセージ

```
cfgadm: Component system is busy, try again: failed to
offline:
    device path
```

Resource	Information
----- /dev/dsk/device-name	----- swap area

原因

1 つまたは複数の構成されているスワップ領域を削除または置き換えようとした。

解決方法

指定されたデバイス上のスワップ領域の構成を解除してから、再度 `cfgadm` を実行します。

エラーメッセージ

```

cfgadm: Component
system is busy, try again: failed to offline:
  device path
    Resource          Information
    -----          -----
    /dev/dsk/device-name  dump device (swap)
    
```

原因

スワップ領域上に構成されているダンプデバイスを取り外そうとしたか、置き換えようとした。

解決方法

スワップ領域に構成されているダンプデバイスの構成を解除してから、再度 `cfgadm` を実行します。

エラーメッセージ

```
cfgadm: Component
system is busy, try again: failed to offline:
  device path
    Resource          Information
  -----
  /dev/dsk/device-name  dump device (dedicated)
```

原因

専用ダンプデバイスを取り外そうとしたか、置き換えようとした。

解決方法

専用となっているダンプデバイスの構成を解除し、再度 `cfgadm` を実行します。

詳細は、`cfgadm(1M)` のマニュアルページを参照してください。

ネットワークの管理についてのトピック

このトピックでは、Solaris 環境におけるネットワークの管理について記述します。次の章で構成されます。

第 21 章	Solaris 8 リリースに含まれていた <code>sendmail</code> のバージョン 8.9.3 に対する大幅な変更点について記述します。
161ページの「BIND の更新」	Solaris 8 4/01 リリースでの Berkeley Internet Name Domain (BIND) バージョン 8.2.2 の新機能の概要を記述します。
第 23 章	動的再構成において既存の IP ユーザーに影響を及ぼすことなく特定のネットワークデバイスを切り離すことを可能にする、IP ネットワークマルチパスの変更点について記述します。
第 24 章	システム管理者が逆方向トンネルをセットアップしたりモバイルノードに専用アドレスを割り当てることを可能にする、モバイル IP の変更点の概要を記述します。

メールサービス

Solaris 8 リリースに組み込まれていた `sendmail` バージョン 8.9.3 が、大幅に変更されました。この章では、Solaris 8 4/01 リリースに組み込まれている、`sendmail` の新しいバージョン 8.10.2+Sun での変更点について説明します。この章は次の節から構成されています。

- 128ページの「`sendmail` に関する情報のその他の参照先」
- 128ページの「`sendmail` バージョン 8.9.3 からの変更点」
- 156ページの「`mail.local` における変更点」
- 157ページの「`mailstats` における変更点」
- 157ページの「`makemap` における変更点」
- 158ページの「関連するその他の変更点や機能」

Solaris のメールサービスについての情報は、『*Solaris のシステム管理 (第 3 巻)*』の「メールサービスについてのトピック」を参照してください。

注 - 最新のマニュアルページを参照するには、`man` コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「*Solaris 8 Reference Manual Collection*」には記載されていない新しい情報も提供されています。

sendmail に関する情報のその他の参照先

sendmail に関する情報のその他の参照先には、次のものがあります。

- sendmail のホームページ - <http://www.sendmail.org>
- sendmail の FAQ - <http://www.sendmail.org/faq>
- 新しい sendmail 構成ファイルの README - <http://www.sendmail.org/m4/readme.html>
- Fatbrain.com の sendmail に関するマニュアル (特に、O'Reilly & Associates, Inc. の sendmail の second edition) - <http://www1.fatbrain.com/catalogs/computing/subjects.asp?SubjectCode=OML>

sendmail バージョン 8.9.3 からの変更点

この節には、次の情報が含まれています。

- 129ページの「新しいコマンド行オプション」
- 129ページの「新規および更新された構成ファイルオプションとその関連事項」
- 142ページの「新たに定義された sendmail のマクロ」
- 144ページの「sendmail の新規および更新された m4 構成マクロとその関連事項」
- 148ページの「sendmail の新しいコンパイルフラグ」
- 149ページの「新しい配信エージェントフラグ」
- 150ページの「配信エージェントの新しい等式」
- 151ページの「新しい待ち行列機能」
- 152ページの「sendmail における LDAP の新しい使い方」
- 153ページの「新しい組み込み型のメールプログラム機能」
- 154ページの「新しいルールセット機能」
- 155ページの「新しいファイルのロケーション」

新しいコマンド行オプション

次の表は、sendmail の新しいコマンド行オプションを示したものです。

表 21-1 sendmail の新しいコマンド行オプション

オプション	説明
-G	コマンド行から送られているメッセージは中継のためであって、最初の送信ではないことを表します。アドレスが完全指定でない場合、このメッセージは拒否されます。正規化は行われません。ftp://ftp.sendmail.org から入手できる sendmail 配布の一部である『RELEASE NOTES』に記述されているように、正しくない形式のメッセージは将来のリリースでは拒否される可能性があります。
-L tag	syslog メッセージで使用される識別子を指定した tag に設定します。
-U	ftp://ftp.sendmail.org から入手できる sendmail 配布の一部である『RELEASE NOTES』に記述されているように、このオプションは廃止になりました。メールユーザーエージェントは、これが中継のための送信であることを示す場合は、(-U ではなく) -G 引数を使用すべきです。

新規および更新された構成ファイルオプションとその関連事項

この節は、新規および更新された構成ファイルオプションの表と、次の関連事項の情報から構成されます。

- 135ページの「廃止された sendmail の構成ファイルオプション」
- 136ページの「新しい ClientPortOptions オプション」
- 137ページの「DaemonPortOptions オプションにおける変更点」
- 139ページの「PidFile と ProcessTitlePrefix オプションの追加の引数」
- 139ページの「PrivacyOptions オプションにおける変更点」
- 140ページの「Timeout オプションにおける変更点」

注 - 次の表に示す sendmail オプションは、通常、構成ファイルに宣言します。しかし、コマンド行から宣言することもできます。ただし、コマンド行を使用する際は、セキュリティ上のリスクを避けるために root のアクセス権を放棄します。

これらのオプションを宣言する際は、次の構文のいずれかを使用します。

<code>O OptionName=argument</code>	<code># 構成ファイル用</code>
<code>-OOptionName=argument</code>	<code># コマンド行用</code>
<code>define('m4Name', argument)</code>	<code># m4 の構成用</code>

次の表では、sendmail の新規のオプションと更新されたオプションを示します。

表 21-2 sendmail の新規のオプションと更新されたオプション

オプション	説明
ClientPortOption	詳細は、136ページの「新しい ClientPortOptions オプション」を参照してください。
ControlSocketName	m4 名: confCONTROL_SOCKET_NAME 引数: <i>filename</i> 。推奨ソケット名は /var/spool/mqueue/.smcontrol。セキュリティ保護のため、この UNIX ドメインソケットは、root でのみアクセス可能なディレクトリになければなりません。 この新しいオプションを設定すると、デーモン管理のための制御ソケットが作成されます。このオプションでは、名前付きソケットを使用して、外部プログラムから動作中の sendmail デーモンのステータスを制御したり照会したりすることを可能にします。このソケットは、INN ニュースサーバーへの ctlinnd インタフェースに類似しています。このオプションを設定しない場合、制御ソケットは一切利用できません。
DaemonPortOptions	詳細は、137ページの「DaemonPortOptions オプションにおける変更点」を参照してください。
DataFileBufferSize	m4 名: confDF_BUFFER_SIZE 引数: <i>number</i> この新しいオプションは、ディスクベースのファイルが使用される前の、メモリーバッファデータ (df) ファイルの最大サイズ (単位: バイト) を制御します。デフォルトは 4096 バイトです。Solaris オペレーティング環境では、変更の必要はありません。

表 21-2 sendmail の新規のオプションと更新されたオプション 続く

オプション	説明
DeadLetterDrop	<p>m4 名: confDEAD_LETTER_DROP</p> <p>引数: <i>filename</i></p> <p>この新しいオプションは、システム全体に渡る <code>dead.letter</code> ファイルのロケーションを定義します。以前は <code>/usr/tmp/dead.letter</code> にハードコード化されていました。このオプションを設定する必要はありません。</p>
DontBlameSendmail	<p>NonRootSafeAddr と呼ばれる新しい引数が追加されました。</p> <p><code>.forward</code> プログラムを実行したり、ファイルの所有者としてそのファイルへ配信するのに十分な特権が <code>sendmail</code> がない場合、アドレスは安全でない (<code>unsafe</code>) とマーク付けされます。さらに、<code>RunAsUser</code> が設定されている場合には、ユーザーは <code>.forward</code> プログラム内のプログラムを使用したり、ファイルへ配信したりすることができません。これらの問題を解決するために、新しい引数 <code>NonRootSafeAddr</code> を使用します。</p>
DontProbeInterfaces	<p>m4 名: confDONT_PROBE_INTERFACES</p> <p>引数: <code>true</code> または <code>false</code>。デフォルトは <code>false</code>。</p> <p>このオプションを <code>true</code> に設定すると、<code>sendmail</code> は、すべてのローカルインタフェースの名前およびアドレスをクラス <code>w</code> (<code>\$(=w)</code>) に挿入しません。したがって、これらのアドレスのサポートを同時に含める必要があります (たとえば、<code>mailertable</code> エントリに)。そうしないと、これらのインタフェースアドレスへのメールは構成エラーで戻されます。ただし、このオプションを設定すると、起動が速くなります。</p>
LDAPDefaultSpec	<p>m4 名: confLDAP_DEFAULT_SPEC</p> <p>引数: 適切な定義を持つクラススイッチ (たとえば、<code>-hhost</code>、<code>-pport</code>、<code>-dbind DN</code>)。</p> <p>この新しいオプションでは、LDAP マップのデフォルトのマップ指定が可能になります。K コマンドで個々のマップ指定を別途指定しない限り、指定したデフォルトの設定がすべての LDAP マップに使用されます。このオプションは、LDAP マップを定義する前に設定します。</p>

表 21-2 sendmail の新規のオプションと更新されたオプション 続く

オプション	説明
MaxAliasRecursion	<p>m4 名: confMAX_ALIAS_RECURSION</p> <p>引数: <i>number</i></p> <p>このオプションは、別名回帰の最大の深さを指定します。デフォルトは次の通りです。</p> <p>V1/Sun 構成ファイル (この構成ファイルの使用は推奨されません) の場合は、50。</p> <p>その他のバージョンの構成ファイルの場合は、10。</p>
MaxHeadersLength	<p>m4 名: confMAX_HEADERS_LENGTH</p> <p>引数: <i>number</i></p> <p>このオプションは、すべてのヘッダーを合計した長さの最大値を指定し、「サービス拒否」攻撃の防止に使用できます。デフォルトは 32768 です。16384 より小さい値を使用すると、警告が出されます。通常、Solaris オペレーティング環境では、デフォルト値を変更する必要はありません。</p>
MaxMimeHeaderLength	<p>m4 名: confMAX_MIME_HEADER_LENGTH</p> <p>引数: <i>number</i></p> <p>このオプションは、特定の MIME ヘッダーフィールド値の最大長として文字数 x を設定します。さらに、これらのヘッダー内のパラメータに対し最大長 y を指定できます。これらは x/y のように組み合わせて使用します。y を指定しない場合は、x の半分の値が使用されます。値を何も指定しない場合は、デフォルト値は 0 で、チェックが行われなかったことを表わします。このオプションは、メールユーザーエージェントを「バッファオーバーフロー」攻撃から防御するために意図されたものです。推奨値は 256/128 から 1024/256 です。128/40 より小さい値を使用すると、警告が出されます。</p>
MaxRecipientsPerMessage	<p>引数: <i>number</i></p> <p>このオプションでは、1つの SMTP 封筒内の受信数が指定した数以下となります。引数の最小値は 100 です。このオプションは今までどおりコマンド行からでも、構成ファイルからでも宣言できます。ただし、一般ユーザーは、コマンド行からこのオプションを設定することにより、<code>sendmail -bs</code> で送信されたメッセージをオーバーライドできるようにになりました。この場合は、<code>sendmail</code> は <code>root</code> 特権を放棄しません。</p>

表 21-2 sendmail の新規のオプションと更新されたオプション 続く

オプション	説明
PidFile	<p>m4 名: confPID_file</p> <p>引数: 139ページの「PidFile と ProcessTitlePrefix オプションの追加の引数」を参照してください。</p> <p>この新しいオプションは、pid ファイルのロケーションを定義します。ファイル名は、開かれる前にマクロ展開されます。デフォルトは /var/run/sendmail.pid です。</p>
PrivacyOptions	<p>詳細は、139ページの「PrivacyOptions オプションにおける変更点」を参照してください。</p>
ProcessTitlePrefix	<p>m4 名: confPROCESS_TITLE_PREFIX</p> <p>引数: 詳細は、139ページの「PidFile と ProcessTitlePrefix オプションの追加の引数」を参照してください。</p> <p>この新しいオプションは、/usr/ucb/ps auxww のリストに表示されるプロセスタイトルの接尾辞文字列を指定します。この文字列はマクロ処理されます。Solaris オペレーティング環境では、変更は必要ありません。</p>
QueueLA	<p>m4 名: confQUEUE_LA</p> <p>引数: <i>number</i></p> <p>デフォルト値が、8 から、システム起動時のオンラインプロセッサ数 × 8 に変更されました。プロセッサが1つだけのマシンの場合には、この変更の影響はありません。この値を変更すると、デフォルト値がオーバーライドされ、プロセッサ数が考慮されなくなります。したがって、値を変更する場合は、その影響を十分に考慮する必要があります。</p>
QueueSortOrder	<p>m4 名: confQUEUE_SORT_ORDER</p> <p>host 引数を指定すると、ソートの前にホスト名を前後逆に入れ替えるようになりました。これによってドメインでソートされるため、待ち行列が一括して実行されます。この改善により、接続キャッシュの使用機会が増えます (接続キャッシュがある場合)。</p> <p>新しい filename 引数では、待ち行列がファイル名でソートされます。これによって、待ち行列の実行を準備する際に、各待ち行列ファイルを開いて読み取る必要がなくなります。</p>

表 21-2 sendmail の新規のオプションと更新されたオプション 続く

オプション	説明
RefuseLA	<p>m4 名: confREFUSE_LA</p> <p>引数: <i>number</i></p> <p>デフォルト値が、12 から、システム起動時のオンラインプロセッサ数 × 12 に変更されました。プロセッサが 1 つだけのマシンの場合には、この変更の影響ありません。この値を変更すると、デフォルト値がオーバーライドされ、プロセッサ数が考慮されなくなります。したがって、値を変更する場合は、その影響を十分に考慮する必要があります。</p>
RrtImpliesDsn	<p>m4 名: confRRT_IMPLIES_DSN</p> <p>引数: true または false</p> <p>この新しいオプションを設定すると、Return-Receipt-To: ヘッダーによって、配信ステータス通知 (DSN: Delivery Status Notification) の要求が生成されます。この要求は、ヘッダーに指定されているアドレスではなく、封筒の送信側に送られます。</p>
SendMimeErrors	<p>m4 名: confMIME_FORMAT_ERRORS</p> <p>引数: true または false</p> <p>デフォルトは true になりました。</p>
Timeout	<p>詳細は、140ページの「Timeout オプションにおける変更点」を参照してください。</p>

表 21-2 sendmail の新規のオプションと更新されたオプション 続く

オプション	説明
TrustedUser	<p>m4 名: confTRUSTED_USER</p> <p>引数: <i>user name</i> または <i>user numeric ID</i></p> <p>この新しいオプションでは、root ではなくユーザー名を指定して重要なファイルを所有することが可能になります。このオプションを設定すると、生成された別名のデータベースおよび制御ソケット (構成されている場合は自動的にそのユーザーが所有者となります。このオプションには HASFCHOWN が必要です。HASFCHOWN の詳細は、148 ページの「sendmail の新しいコンパイルフラグ」を参照してください。</p> <p>別名マップを再構築できるのは、TrustedUser ユーザー、root、クラス t (\$=t) ユーザーに限られます。</p>
XscriptFileBufferSize	<p>m4 名: confXF_BUFFER_SIZE</p> <p>引数: <i>number</i></p> <p>この新しいオプションは、データベースのファイルが使用される前のメモリバッファトランスクリプト (xf) ファイルの最大サイズ (単位: バイト) を制御します。デフォルトは 4096 バイトです。Solaris オペレーティング環境では、変更の必要はありません。</p>

廃止された sendmail の構成ファイルオプション

次の表では、廃止された sendmail の構成ファイルオプションについて記述します。

表 21-3 廃止された sendmail の構成ファイルオプション

オプション	説明
AutoRebuildAliases	このオプションを設定すると「サービス拒否」攻撃が可能になるため、このオプションは廃止されました。ftp://ftp.sendmail.org から入手できる sendmail 配布の一部である『RELEASE NOTES』を参照してください。以前は、このオプションを設定することで別名ファイルを再構築している間に (そのファイルがまだ不安定な状態で)、sendmail プロセスを kill することができました。
MeToo	このオプションは、現在デフォルトで True になっており、廃止されました。ftp://ftp.sendmail.org から入手できる sendmail 配布の一部である『RELEASE NOTES』を参照してください。

新しい ClientPortOptions オプション

新しい ClientPortOptions オプションは、発信接続用で、DaemonPortOptions オプションに類似しています。このオプションは、一連の *key=value* ペアからなるクライアント SMTP オプションを設定します。このオプションを宣言するには、次のいずれかの構文を使用します。この例には構文を示すために 2 つのペアが示されていますが、実際には 1 つまたは複数のペアを指定することができます。

```
O ClientPortOptions=pair, pair          # 構成ファイル用
-OClientPortOptions=pair, pair         # コマンド行用
define('confCLIENT_OPTIONS', 'pair, pair') # m4 構成用
```

次の表では、このオプションの新しいキーを示します。

表 21-4 ClientPortOptions の新しいキー

キー	説明
Addr	アドレスマスクを指定します。この値は、ドット表記の数値アドレス、またはネットワーク名を指定できます。ペアを省略すると、デフォルトは INADDR_ANY になり、どのネットワークからの接続も許可されます。
Family	アドレスファミリを指定します。このキーのデフォルト値は AF_INET の inet です。その他の値には、AF_INET6 用の inet6、AF_ISO 用の iso、AF_NS 用の ns、AF_CCITT 用の x.25 があります。

表 21-4 ClientPortOptions の新しいキー 続く

キー	説明
Listen	待機待ち行列のサイズを指定します。このキーのデフォルトは 10 です。Solaris オペレーティング環境では、変更の必要はありません。
Port	待機ポートの名前と数を指定します。このキーのデフォルト値は smtp です。
RcvBufSize	TCP/IP 送信バッファのサイズを指定します。このキーにはデフォルト値はなく、サイズが自動的に指定されることはありません。オプションでゼロより大きい値を設定すると、その値が使用されます。Solaris オペレーティング環境では、このバッファのサイズを制限する必要はありません。
Modifier	sendmail のフラグを指定します。フラグ h は、接続パラメータで選択されたかデフォルトで選択されたかにかかわらず、HELO または EHLO コマンドに、発信インタフェースアドレスに対応する名前を使用します。

DaemonPortOptions オプションにおける変更点

下記に示す 2 つの表は、このオプションの 2 つの新しいキー、および新しいキーの 1 つである Modifier キーのいくつかの特定の値を示したものです。このオプションを宣言するには、次のいずれかの構文を使用します。この例では、*pair* は *key=value* を表しています。この例では構文を示すために 2 つのペアが示されていますが、実際には 1 つまたは複数のペアを指定することができます。

```
O DaemonPortOptions=pair, pair           # 構成ファイル用
-ODaemonPortOptions=pair, pair         # コマンド行用
define('confDAEMON_OPTIONS', 'pair, pair') # m4 構成用
```

注 - セキュリティ上のリスクを避けるため、コマンド行からこのオプションを設定すると、sendmail は root アクセス権を放棄します。

次の表では、DaemonPortOptions オプションの 2 つの新しいキーを示します。

表 21-5 DaemonPortOptions の新しいキー

キー	説明
Name	sendmail に対するユーザー定義可能な名前を指定します。この名前は、エラーメッセージやロギングに使用されます。デフォルトは MTA です。
Modifier	sendmail に対する値を指定します。値は、区切り文字で区切らずに連続してリストすることができます。指定可能な値は、表 21-6 を参照してください。

次の表では、新しい Modifier キーの値を示します。

表 21-6 DaemonPortOptions の Modifier キーの値

値	説明
C	ホスト名の正規化を行いません。
E	ETRN コマンドを許可しません。
a	認証を要求します。
b	このメールを受け取るインタフェースにバインドします。
c	ホスト名の正規化を行います。この値は、構成ファイルでの宣言でのみ使用します。
f	完全指定であるホスト名を要求します。この値は、構成ファイルでの宣言でのみ使用します。
h	発信 HELO コマンドにインタフェースの名前を使用します。
u	完全指定でないアドレスを許可します。この値は、構成ファイルでの宣言でのみ使用します。

PidFile と ProcessTitlePrefix オプションの追加の引数

次の表では、PidFile と ProcessTitlePrefix オプションの、追加のマクロ処理引数を示します。これらのオプションの詳細は、表 21-2 を参照してください。

表 21-7 PidFile と ProcessTitlePrefix オプションの引数

マクロ	説明
<code>#{daemon_addr}</code>	デーモンアドレスを指定します (たとえば、0.0.0.0)
<code>#{daemon_family}</code>	デーモンファミリを指定します (たとえば、inet、inet6 など)
<code>#{daemon_info}</code>	デーモン情報を指定します (たとえば、SMTP+queueing@00:30:00)
<code>#{daemon_name}</code>	デーモン名を指定します (たとえば、MSA)
<code>#{daemon_port}</code>	デーモンポートを指定します (たとえば、25)
<code>#{queue_interval}</code>	待ち行列の実行間隔を指定します (たとえば、00:30:00)

PrivacyOptions オプションにおける変更点

下記の表では、PrivacyOptions (popt) の新しい引数と更新された引数を示します。このオプションは、sendmail が root 特権を放棄することなく、コマンド行から宣言することができます。この sendmail オプションを宣言するには、次のいずれかの構文を使用します。

```
O PrivacyOptions=argument           # 構成ファイル用
-OPrivacyOptions=argument         # コマンド行用
define('confPRIVACY_FLAGS', 'argument') # m4 構成用
```

次の表では、PrivacyOptions オプションの新しい引数と更新された引数を示します。

表 21-8 PrivacyOptions の新しい引数と更新された引数

引数	説明
goaway	noetrn フラグと noreceipts フラグは許可されなくなりました。
nobodyreturn	この引数は、オリジナルのメッセージの本文を配信ステータス通知に含めないことを sendmail に知らせます。
noreceipts	この引数を設定すると、配信ステータス通知 (DSN: Delivery Status Notification) が発行されません。

Timeout オプションにおける変更点

次の表では、Timeout オプションにおける変更点についての情報を示します。特に、この sendmail オプションには、いくつかの新しいキーワードと、ident の新しい値が追加されています。Solaris オペレーティング環境では、この表に示されているキーワードのデフォルトの値を変更する必要はありません。しかし、変更を行う場合には、*keyword=value* 構文を使用します。*value* は時間の間隔です。次の例を参照してください。

```
O Timeout.keyword=value # 構成ファイル用
-OTimeout.keyword=value # コマンド行用
define(`m4_name', value) # m4 構成用
```

注 - セキュリティ上のリスクを避けるため、コマンド行からこのオプションを設定すると、sendmail は root アクセス権を放棄します。

表 21-9 Timeout の新規の設定値および更新された設定値

キーワード	デフォルト 値	説明
<code>control</code>	2m	m4 名: <code>confTO_CONTROL</code> 制御ソケット要求を満たすためだけに使用される時間の合計を制限します。
<code>ident</code>	5s	m4 名: <code>confTO_IDENT</code> IDENT パケットをドロップするサイトへのメーリングに関連する一般的な遅延を防ぐため、デフォルトは 30 秒ではなく、5 秒になっています。
<code>queuereturn</code>	5d	m4 名: <code>confTO_QUEUERETURN</code> 値 <code>now</code> を含めると、エントリは、配信を試行することなく、待ち行列からただちに返されます。
<code>resolver.retrans</code>	<i>varies</i>	m4 名: <code>confTO_RESOLVER_RETRANS</code> リゾルバの再伝送時間の間隔を秒数で設定します。この値は、 <code>resolver.retrans.first</code> と <code>resolver.retrans.normal</code> に適用されます。
<code>resolver.retrans.first</code>	<i>varies</i>	m4 名: <code>confTO_RESOLVER_RETRANS_FIRST</code> メッセージ配信の最初の試みに対する、リゾルバの再伝送時間の間隔を秒数で設定します。
<code>resolver.retrans.normal</code>	<i>varies</i>	m4 名: <code>confTO_RESOLVER_RETRANS_NORMAL</code> 最初の配信の試みを除くすべてのリゾルバのルックアップに対する、リゾルバの再伝送時間の間隔を秒数で設定します。
<code>resolver.retry</code>	<i>varies</i>	m4 名: <code>confTO_RESOLVER_RETRY</code> リゾルバの照会を再伝送する回数を設定します。この値は <code>Timeout.resolver.retry.first</code> と <code>Timeout.resolver.retry.normal</code> に適用されます。

表 21-9 Timeout の新規の設定値および更新された設定値 続く

キーワード	デフォルト 値	説明
<code>resolver.retry.first</code>	<i>varies</i>	m4 名: <code>confTO_RESOLVER_RETRY_FIRST</code> メッセージ配信の最初の試みに対する、リゾルバの照会を再伝送する回数を設定します。
<code>resolver.retry.normal</code>	<i>varies</i>	m4 名: <code>confTO_RESOLVER_RETRY_NORMAL</code> 最初の配信の試みを除くすべてのリゾルバのルックアップに対する、リゾルバの照会を再伝送する回数を設定します。

新たに定義された sendmail のマクロ

次の表では、sendmail プログラムでの使用のために予約されている新しいマクロを示します。これらの値は内部的に割り当てられます。

表 21-10 新たに定義された sendmail のマクロ

マクロ	説明
<code>\${auth_authen}</code> , <code>\${auth_type}</code> , <code>\${auth_author}</code>	クライアントの認証資格 (認証に使用される機構) と承認 ID (AUTH=パラメータ。指定されている場合) を保持します。
<code>\${client_resolve}</code>	<code>\${client_name}</code> に対する <code>resolve</code> 呼び出しの結果 (OK、FAIL、FORGED、または TEMP) を保持します。
<code>\${deliveryMode}</code>	sendmail が使用している現在の配信モードを指定します (DeliveryMode オプションの代わりです)。
<code>\${dsn_notify}</code> , <code>\${dsn_envid}</code> , <code>\${dsn_ret}</code>	対応する DSN パラメータの値を保持します。
<code>\${if_addr}</code>	インターネットがループバックネットに属さない場合、着信接続にインタフェースのアドレスを指定します。これは仮想ホスティングに特に便利です。

表 21-10 新たに定義された sendmail のマクロ 続く

マクロ	説明
<code>\${if_name}</code>	着信接続にインタフェースのホスト名を指定します。これは仮想ホスティングに特に便利です。
<code>\${load_avg}</code>	実行待ち行列にあるジョブの現在の平均数をチェックし、報告します。
<code>\${msg_size}</code>	メッセージが収集される前の、ESMTP ダイアログにおけるメッセージサイズの値 (<code>SIZE=parameter</code>) を保持します。その後、マクロは <code>sendmail</code> によって計算されたメッセージサイズを保持し、これは <code>check_compat</code> で使用されます。
<code>\${ntries}</code>	配信を試みる回数を保持します。
<code>\${rcpt_mailer}, \${rcpt_host}, \${rcpt_addr}, \${mail_mailer}, \${mail_host}, \${mail_addr}</code>	RCPT 引数と MAIL 引数を解析した結果、つまり、メール配信エージェント (<code>mailer</code>)、ホスト (<code>host</code>)、ユーザー (<code>addr</code>) の 3 つからなる解決済みの RHS を保持します。

sendmail 構成ファイルの構築に使用される新しいマクロ

次の表では、sendmail 構成ファイルの構築に使用される新しいマクロを示します。

表 21-11 sendmail 構成ファイルの構築に使用される新しいマクロ

フラグ	説明
<code>LOCAL_MAILER_EOL</code>	ローカルメールプログラムのデフォルトの行末の文字列をオーバーライドします。
<code>LOCAL_MAILER_FLAGS</code>	デフォルトで <code>Return-Path:</code> ヘッダーを追加します。
<code>MAIL_SETTINGS_DIR</code>	メール設定ディレクトリへのパス (後続のスラッシュを含む) を保持します。

表 21-11 sendmail 構成ファイルの構築に使用される新しいマクロ 続く

フラグ	説明
MODIFY_MAILER_FLAGS	* MAILER_FLAGS を改善します。このマクロはフラグを設定、追加、または削除します。
RELAY_MAILER_FLAGS	中継メールプログラムの追加フラグを定義します。
USENET_MAILER_FLAGS	ローカルメールプログラムではないため、1フラグは削除されました。

sendmail の新規および更新された m4 構成マクロとその関連事項

この節では、sendmail の新規および更新された m4 構成マクロと、次の関連事項を記述します。

- 145ページの「新規および更新された FEATURE() 宣言」
- 148ページの「m4 構成の更新された MAILER() 宣言」

これ以降の表に示すマクロの宣言には次の構文を使用します。

`symbolic_name('value')`

表 21-12 sendmail の新規および更新された m4 構成マクロ

m4 マクロ	説明
FEATURE() 宣言	詳細は、145ページの「新規および更新された FEATURE() 宣言」を参照してください。
LOCAL_DOMAIN()	このマクロはクラス w (\$=w) にエントリを追加します。

表 21-12 sendmail の新規および更新された m4 構成マクロ 続く

m4 マクロ	説明
MASQUERADE_EXCEPTION()	偽装できないホストまたはサブドメインを定義する新しいマクロ。
SMART_HOST()	user@[host] などの、括弧付きのアドレスにこのマクロを使用できるようになりました。
TRUST_AUTH_MECH()	SMTP AUTH が使用されている場合、「信頼される」機構として認証されているすべてのユーザーに対し、中継が許可されます。これは、この機構が TRUST_AUTH_MECH('list_of_mechanisms') 宣言で定義されていることを意味します。
VIRTUSER_DOMAIN() または VIRTUSER_DOMAIN_FILE()	これらのマクロを使用する場合は、 $\$=\{\text{VirtHost}\}$ を $\$=R$ に含めます。 $\$=R$ は、中継を許可されているホスト名のセットです。

新規および更新された FEATURE() 宣言

次の表では、m4 FEATURE() 宣言の新規のキーワードと更新されたキーワードを示します。.mc ファイルに feature を宣言するには、次の例に示す構文を使用します。

```
FEATURE('key_word', 'argument')
```

次の表では、どのキーワードに arguments が必要であることを示します。

表 21-13 FEATURE() 宣言の新規および更新されたキーワード

キーワード	説明
delay_checks	<p>引数: friend (spam-friend テストを有効にする)、または hater (spam-hater テストを有効にする)</p> <p>すべてのチェックを遅らせる新しいキーワード。FEATURE('delay_checks') を使用すると、クライアントが接続したり MAIL コマンドを発行した時にルールセット check_mail や check_relay は呼び出されません。その代わりに、これらのルールセットは check_rcpt ルールセットから呼び出されます。詳細は、/usr/lib/mail/README ファイルを参照してください。</p>
dnsbl	<p>引数を最大 2 つまで指定できる新しいキーワード。 DNS サーバー名 拒否メッセージ</p> <p>このキーワードは、複数回含めることができます。</p>
generics_entire_domain	<p>引数: なし</p> <p>genericstable を \$=G のサブドメインに適用するために使用することもできる新しいキーワード。</p>
ldap_routing	<p>引数: 詳細は、/usr/lib/mail/README の LDAP ROUTING の節を参照してください。</p> <p>LDAP アドレスルーティングを実装する新しいキーワード。</p>
local_lmtp	<p>引数: LMTP 対応メールプログラムのパス名。デフォルトは mail.local で、これは Solaris のリリースでは LMTP 対応です。</p> <p>ローカルメールプログラムの配信ステータス通知 (DSN) 診断コードタイプを SMTP の適切な値に設定するキーワード。</p>

表 21-13 FEATURE() 宣言の新規および更新されたキーワード 続く

キーワード	説明
nocanonicalize	<p>引数: <code>canonicalize_hosts</code> または、なし</p> <p>このキーワードには、次の機能が含まれるようになりました。</p> <p><code>CANONIFY_DOMAIN</code> または <code>CANONIFY_DOMAIN_FILE</code> によって指定されたドメインのリストを <code>\$(および \$)</code> 演算子に渡し、正規化を依頼することができます。</p> <p><code><user@host></code> のような 1 つのホスト名だけを持つアドレスの正規化を依頼できる。ただし、そのパラメータとして <code>canonicalize_hosts</code> が指定されている場合に限りです。</p> <p>複数のコンポーネントからなるアドレスに後続のドットを付加します。</p>
no_default_msa	<p>引数: なし</p> <p>m4 によって生成された構成ファイルにある <code>sendmail</code> のデフォルト設定をオフにして、いくつかの別のポートで待機する、新しいキーワード。RFC 2476 の実装です。</p>
nouucp	<p>引数: <code>reject</code> (! トークンを許可しない) または <code>nospecial</code> (! トークンを許可する)。</p> <p>アドレスのローカル部分に ! トークンを許可するかどうかを指定するキーワード</p>
nullclient	<p>引数: なし</p> <p>通常構成のすべてのルールセットを提供するキーワード。これにより、<code>anti-spam</code> チェックを行うことができます。</p>

表 21-13 FEATURE() 宣言の新規および更新されたキーワード 続く

キーワード	説明
relay_mail_from	<p>引数: <i>domain</i> (オプションの引数)</p> <p>メール送信側がアクセスマップに RELAY としてリストされており、From: ヘッダー行が付加されている場合、中継を許可する、新しいキーワード。オプションの <i>domain</i> 引数が指定されている場合は、メール送信側のドメイン部分もチェックされます。</p>
virtuser_entire_domain	<p>引数: なし</p> <p>VIRTUSER_DOMAIN または VIRTUSER_DOMAIN_FILE によって移植できる virtusertable エントリのマッチングのための新しいクラス $\\${VirtHost}$ を適用するために使用できるようになったキーワード。</p> <p>FEATURE('virtuser_entire_domain') では、クラス $\\${VirtHost}$ をサブドメイン全体に適用することもできます。</p>

m4 構成の更新された MAILER() 宣言

MAILER() 宣言は、配信エージェントのサポートを指定します。配信エージェントを宣言するには、次の構文を使用します。

```
MAILER('symbolic_name')
```

sendmail のこの新しいバージョンでは、MAILER('smtp') 宣言に追加のメールプログラム、dsmtplib を含むようになりました。dsmtplib は、F=% メールプログラムフラグを使用することによって要求に応じて配信を行います。dsmtplib メールプログラム定義では、新しい DSMTP_MAILER_ARGS が使用されます。このデフォルトは IPC \$h です。

sendmail の新しいコンパイルフラグ

次の表では、sendmail のコンパイルに使用される新しいフラグを示します。構成により他のフラグが必要な場合は、ユーザーがそのソースをダウンロードし、バイナリを再コンパイルする必要があります。このプロセスについては、<http://www.sendmail.org> を参照してください。

表 21-14 sendmail のコンパイルに使用される新しいフラグ

フラグ	説明
HASFCHOWN	fchown(2) の使用をサポートします。
HASRANDOM	rand(3C) の使用をサポートします (random(3C) ではなく)。
MAXINTERFACES	sendmail がクラス w (\$=w) のホスト名と IP アドレスを検索するときに読み取るインタフェースの数を指定します。デフォルト値は 512 です。

新しい配信エージェントフラグ

次の表は、新しい配信エージェントフラグについて記述したものです。デフォルトでは、これらのフラグは設定されていません。これらの単一文字フラグはブール値です。フラグの設定や設定解除は、構成ファイルの `F=statement` にフラグを含めるか、含めないかで決まります。次はその例です。

Mlocal,	P=/usr/lib/mail.local, F=lsDFMAw5:/ @qSXfmnz9, S=10/30, R=20/40,
Mprog,	P=/bin/sh, F=lsDFMoqeu9, S=10/30, R=20/40, D=\$z:/,
Msmtp,	P=[IPC], F=mDFMuX, S=11/31, R=21, E=\r\n, L=990,
Mesmtpt,	P=[IPC], F=mDFMuXa, S=11/31, R=21, E=\r\n, L=990,
Msmtp8,	P=[IPC], F=mDFMuX8, S=11/31, R=21, E=\r\n, L=990,
Mrelay,	P=[IPC], F=mDFMuXa8, S=11/31, R=61, E=\r\n, L=2040,

表 21-15 新しいメールプログラムのフラグ

フラグ	説明
%	このフラグを使用するメールプログラムは、待ち行列のメッセージが ETRN 要求か、待ち行列オプション <code>-qI</code> 、 <code>-qR</code> 、 <code>-qS</code> のどれかで選択されない限り、メッセージの最初の受信人や、待ち行列実行にメッセージを配信しません。
6	このフラグにより、メールプログラムは、ヘッダーを 7 ビットでカットします。

配信エージェントの新しい等式

下記の表では、M 配信エージェント定義コマンドで使用できる新しい等式を示します。次の構文は、構成ファイルにすでに存在する等式や引数に新しい等式や引数を付加する方法を示しています。

```
Magent_name, equate, equate, ...
```

次の例は、新しい W= の等式を含んでおり、すべてのデータの送信後にメールプログラムが戻るまでに待つ最大の時間を指定しています。

```
Msmtp, P=[IPC], F=mDFMuX, S=11/31, R=21, E=\r\n, L=990, W=2m
```

m4 構成に対する値の定義を変更するには、次の例に示す構文を使用します。

```
define('SMTP_MAILER_MAXMSGS', '1000')
```

上記の例では、smtp メールプログラムで配信される 1 接続当たりのメッセージの数を 1000 に限定しています。

注 - 通常、mailer ディレクトリの等式の定義を変更するのは、細かい調整が必要なきだけです。

表 21-16 配信エージェントの新しい等式

等式	説明
/=	引数: ディレクトリへのパス メールプログラムを実行する前に、 <code>chroot()</code> を使って入るディレクトリを指定します。
m=	引数: <code>define()</code> ルーチンですすでに定義されている次の m4 値のいずれか SMTP_MAILER_MAXMSGS (smtp メールプログラムの場合) LOCAL_MAILER_MAXMSGS (local メールプログラムの場合) RELAY_MAILER_MAXMSGS (relay メールプログラムの場合) smtp、local、または relay メールプログラムで配信される 1 接続当たりのメッセージの数を制限します。
w=	引数: 時間の増分 すべてのデータの送信後にメールプログラムが戻るまでの最大の時間を指定します。

新しい待ち行列機能

以下は、新しい待ち行列機能の詳細を示したものです。

- 複数の待ち行列ディレクトリがサポートされるようになりました。複数の待ち行列を使用するには、構成ファイルに `QueueDirectory` オプションの値を指定し、次の例に示すようにその終りにアスタリスク (*) を付加します。

```
O QueueDirectory=/var/spool/mqueue/q*
```

このオプション値 `/var/spool/mqueue/q*` は、待ち行列ディレクトリとして `q` で始まるすべてのディレクトリ (またはディレクトリへのシンボリックリンク) を使用することを示しています。sendmail が動作している間に待ち行列ディレクトリ構造を変更しないでください。待ち行列実行は、冗長フラグ (`-v`) が非デーモン待ち行列実行に使用されていない限り、各待ち行列を実行する別個のプロセスを作成します。新しい項目は、待ち行列にランダムに割り当てられます。

- 新しい待ち行列ファイル命名システムでは、60 年間必ず一意になるファイル名が使用されます。このシステムでは、複雑なファイルシステムロックを行わずに待

ち行列 ID を割り当てることができるだけでなく、待ち行列化された項目を待ち行列間で簡単に移動できます。

sendmail における LDAP の新しい使い方

以下では、sendmail における Lightweight Directory Access Protocol (LDAP) の使い方の変更点を記述します。

- ftp://ftp.sendmail.org から入手できる sendmail 配布の一部である『RELEASE NOTES』に記述されているように、LDAPX マップの名前が LDAP に変更されました。LDAP には次の構文を使用します。

```
Kldap ldap options
```

- 単一の LDAP ルックアップからの複数の値の返りをサポートするようになりました。それには、`-v` オプションで、次に示すように返される値をコマで区切った文字列として指定します。

```
Kldap ldap -v"mail,more_mail"
```

- LDAP マップ宣言に LDAP 属性が指定されていない場合、一致したものの中に存在するすべての属性が返されます。
- このバージョンでは、引用符でくくったキーの中のコンマや、LDAP の別名ファイル指定の中の値文字列によって、単一のエントリが複数のエントリに分割されることはありません。
- `%s` トークンを使って LDAP のフィルタ指定を解析する代わりに、新しいトークン `%0` を使ってキーバッファを符号化することができます。`%0` トークンは、LDAP の特殊文字に文字そのものの意味を割り当てます。

次の例では、“*”を使ったルックアップでこれらのトークンがどのように異なるかを示します。

表 21-17 トークンの比較

LDAP マップ指定	同等な指定	結果
-k"uid=%s"	-k"uid=*"	ユーザー属性を持つすべてのレコードと一致する。
-k"uid=%0"	-k"uid=\2A"	名前として"*"を持つユーザーと一致する。

次の表では、新しい LDAP マップのフラグを示します。

表 21-18 新しい LDAP マップフラグ

フラグ	説明
-1	単一の一致を返す必要がある。複数の一致が返された場合、結果は、レコードが 1 つも見つからなかったのと同じことになる。
-r <i>never</i> <i>always</i> <i>search</i> <i>find</i>	LDAP 別名展開オプションを設定する。
-Z <i>size</i>	返す一致の数を制限する。

新しい組み込み型のメールプログラム機能

古い [TCP] 組み込み型のメールプログラムは廃止されました。その代わりに P=[IPC] (プロセッサ間通信) 組み込み型のメールプログラムを使用します。[IPC] 組み込み型のメールプログラムでは、システム上の UNIX ドメインソケットへの配信が可能です。ただし、そのシステムは UNIX ドメインソケットをサポートしていなければなりません。このメールプログラムは、指定されたソケットで待機する LMTP 配信エージェントとともに使用できます。次はメールプログラムの例です。

```
Mexecmail, P=[IPC], F=lsDFMmnqSXzA5@/:|, E=\r\n,
S=10, R=20/40, T=DNS/RFC822/X-Unix, A=FILE /var/run/lmtpd
```

[IPC] メールプログラムの最初のメールプログラム引数は、値が有効かどうかをチェックします。次の表では、最初のメールプログラム引数として指定できる値を示します。

表 21-19 最初のメールプログラム引数として指定できる値

値	説明
A=FILE	UNIX ドメインソケットへの配信に使用される。
A=TCP	TCP/IP 接続に使用される。
A=IPC	将来のバージョンでは廃止される予定。

新しいルールセット機能

次の表では、新しいルールセットとその実行する動作を記述します。

表 21-20 新しいルールセット

セット	説明
check_eoh	収集された情報をヘッダー間で関連比較し、足りないヘッダーがないかチェックする。このルールセットはマクロストレージマップとともに使用され、すべてのヘッダーが収集された後で呼び出される。
check_etrn	ETRN コマンドを使用する (check_rcpt が RCPT を使用する場合など)。
check_expn	EXPN コマンドを使用する (check_rcpt が RCPT を使用する場合など)。
check_vrfy	VERFY コマンドを使用する (check_rcpt が RCPT を使用する場合など)。
trust_auth	MAIL コマンドの指定された AUTH= パラメータが信頼すべきものかどうかを判定する。

次の各項では、新しいルールセット機能について述べます。

- 番号付きルールセットには名前も付けられますが、番号でアクセスすることができます。
- H ヘッダー構成ファイルコマンドでは、ヘッダーチェックにデフォルトのルールセットを指定できます。このルールセットは、個々のヘッダーに独自のルールセットが割り当てられていない場合だけ呼び出されます。
- 構成ファイルバージョンが9 以上の場合、ルールセット内のコメント（つまり、括弧内のテキスト）は削除されません。たとえば、次のルールは `token (1)` という入力には一致しますが、`token` という入力には一致しません。

```
R$+ (1)  $@ 1
```

- `sendmail` は、TCP ラッパーや `check_relay` ルールセットに基づいてコマンドを拒否する場合でも、SMTP RSET コマンドは受け付けます。
- `OperatorChars` オプションを複数回設定すると、警告が出されます。ルールセットを定義した後で `OperatorChars` を設定しないでください。
- 有効でないルールセットを宣言すると、ルールセットの名前およびその行は無視されます。そのルールセットの行は `s0` に追加されません。

新しいファイルのロケーション

次のファイルのロケーションが変更されていますので注意してください。

- `helpfile` は `/etc/mail/helpfile` に移されました。古い名前 (`/etc/mail/sendmail.hf`) には、新しい名前へのシンボリックリンクが指定されています。
- `trusted-users` ファイルは `/etc/mail/trusted-users` に移されました。アップグレードの際に古い名前 (`/etc/mail/sendmail.ct`) が検出され、新しい名前が検出されない場合、古い名前から新しい名前へのハードリンクが作成されます。新しい名前が検出された場合は、何も行われません。デフォルトの内容は `root` です。
- `local-host-names` ファイルは `/etc/mail/local-host-names` に移されました。アップグレードの際に古い名前 (`/etc/mail/sendmail.cw`) が検出され、新しい名前が検出されない場合、古い名前から新しい名前へのハードリンクが作成されます。新しい名前が検出された場合は、何も行われません。デフォルトの内容は長さがゼロです。

mail.local における変更点

次の表は、sendmail がローカルメールの配信エージェントとして使用する mail.local プログラムの新しいコマンド行オプションを示したものです。

表 21-21 mail.local の新しいコマンド行オプション

オプション	説明
-7	ローカルメール転送プロトコル (LMTP) モードが、LHLO 応答の中で 8BITMIME サポートを通知するのを防止する。
-b	メールボックスが quota の設定値を超過した場合、一時エラーではなく、永続エラーを起こす。

LMTP モードのデフォルトは mail.local です。ただし、このリリースでは、LMTP モードにならずにローカル配信エージェントとして mail.local を使用することを選択する場合には、次のどちらかを行って s フラグを設定する必要があります。

構成ファイルには、次の構文を使用します。

```
MODIFY_MAILER_FLAGS('LOCAL', '+S')      # 構成ファイル用
```

m4 構成には、次の 2 つのステップを行います。

```
define('MODIFY_MAILER_FLAGS', 'S')dnl    # 1 つ目のステップ  
MAILER(local)dnl                         # 2 つ目のステップ
```

注 - MODIFY_MAILER_FLAGS は、構成ファイルを構築するために使用する新しいマクロです。詳細は、143ページの「sendmail 構成ファイルの構築に使用される新しいマクロ」を参照してください。

mailstats における変更点

メールプログラムの使用状況の統計を示す mailstats プログラムは、sendmail プログラムと共に配布されます。次の表は、mailstats の新しいオプションを示したものです。

表 21-22 新しい mailstats のオプション

オプション	説明
<code>-C filename</code>	sendmail 構成ファイルを指定する。
<code>-p</code>	プログラムで読み取り可能なモードで、統計情報を出力する。

makemap における変更点

makemap コマンドでは、sendmail のデータベースファイルを作成します。次の表は、新しい makemap オプションを示したものです。オプションの宣言には、次の構文を使用します。

```
makemap options class filename
```

この構文を使用する際には、次の点に注意してください。

- `options` の前にはダッシュが必要です (たとえば、`-dN`)。
- `class` にはデータベースのタイプを指定します (たとえば、`btree`、`dbm`、`hash` など)。
- `filename` には、データベースファイルへのフルパス (または相対名) を指定します。

表 21-23 新しい makemap オプション

オプション	説明
-C	TrustedUser オプションの検索に指定された sendmail 構成ファイルを使用する。
-c	指定された hash と btree キャッシュサイズを使用する。
-e	RHS (right-hand side) の空値を許可する。
-l	サポートされるマップタイプをリスト表示する。
-u	データベースの内容を標準出力にダンプ (マップ解除) する。

注 - makemap がスーパーユーザーとして動作している場合、生成されるマップの所有者は、sendmail 構成ファイルの指定に従って、自動的に TrustedUser に変更されます。TrustedUser オプションについては、表 21-2 を参照してください。

関連するその他の変更点や機能

次の各項は、関連するその他の変更点や機能を示したものです。

- ftp://ftp.sendmail.org から入手できる sendmail 配布の一部である『RELEASE NOTES』に記述されているように、XUSR SMTP コマンドは廃止されました。メールユーザーエージェントは、最初のユーザーメッセージ送信に RFC 2476 Message Submission を使用すべきです。
- Content-Length: ヘッダーは、Sun 構成ファイルのどのバージョンでも、プログラムにパイプ接続されるメッセージで提供されなくなりました。ただし、このヘッダーは、Sun の構成ファイルのどのバージョンを使用する付加メッセージおよび通常のメールボックス配信でもまだ提供されています。
- sendmail は、ディスク容量が少ない場合でも接続を受け入れるようになりました。そのような場合には ETRN コマンドしか許可しません。

- 別名ファイルのエントリを次行に続けるには、新しい行の直前にバックスラッシュを入れます。
- SMTP によるメッセージの送信に対するタイムアウトが、配信の進捗を5分ごとにチェックするように変更されました。この変更により、情報をそれより速く送信できないことが事前にわかるため、タイムアウトになるのを待っているプロセスの数が減ります。
- 次の例に示す構文を使用して、クラスの内容を別のクラスにコピーできるようになりました。

```
c{Dest} $={Source}
```

上記の例では、クラス $\$={Source}$ 内のすべての項目がクラス $\$={Dest}$ にコピーされます。

- マップはデフォルトでオプションでしたが、そうではなくなりました。さらに、マップに問題がある場合には、エラーメッセージが出力されるようになりました。
- クラス P ($\$=P$) のホストやドメインに対する正規化は行われなくなりました。
- オプションに関連する値が全くない場合、= 等式はオプションの展開に組み込まれません。
- ルーティング情報を表わすアドレスは削除されます。たとえば、 $\langle@a,@b,@c:user@d\rangle$ は $\langle user@d\rangle$ にカットされます。

Berkeley Internet Name Domain (BIND) バージョン 8.1.2 から BIND バージョン 8.2.2 への移行 (パッチレベル 5)

Solaris 8 4/01 リリースで、Berkeley Internet Name Domain (BIND) がバージョン 8.1.2 から 8.2.2 へ移行しました。次に記載する情報は、『Solaris ネーミングの管理』の BIND に関する情報を補足するものです。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「Solaris 8 Reference Manual Collection」には記載されていない新しい情報も提供されています。

BIND の更新

BIND バージョン 8.2.2 には次の新機能が含まれます。

- In.named 構成オプション。conf(4) のマニュアルページを参照。
- マルチスレッド化されたアプリケーションで安全に使用できる、リゾルバ (3RESOLV) インタフェースへの拡張。
- 再構成 in.named の起動と停止に使用される ndc(1M) コマンド、および TSIG と DNSSEC キーの作成に使用される dnskeygen(1M) コマンドの追加。

IP ネットワークマルチパス

Solaris 8 4/01 リリースで IP ネットワークマルチパスが機能拡張されました。次の情報は、『IP ネットワークマルチパスの管理』に記載されている IP ネットワークマルチパスについての情報を補足するものです。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「Solaris 8 Reference Manual Collection」には記載されていない新しい情報も提供されています。

ネットワークアダプタの切り離し

IP ネットワークマルチパスは、ネットワークアダプタの単一点障害を回避し、単位時間当たりのデータの流量を向上させます。ネットワークアダプタで障害が発生し、同じ IP リンクに別のアダプタが接続されている場合、すべてのネットワークアクセスは障害が発生したアダプタからこのアダプタに自動的に切り替えられます。この処理によりネットワークへのアクセスは中断されません。また、複数のネットワークアダプタが同じ IP リンクへ接続されている場合、複数のネットワークアダプタにデータの流れが分散されるため、単位時間当たりのデータの流量が向上します。

Solaris 8 4/01 リリースでは、動的再構成 (DR: Dynamic Reconfiguration) で IP ネットワークマルチパスを使用して、IP を使用中のユーザーに影響を及ぼすことなく特定のネットワークデバイスを切り離すことができます。

動的再構成および IP ネットワークマルチパスについての詳細は、『IP ネットワークマルチパスの管理』の「ネットワークアダプタの切り離し」を参照してください。第 2 章に、関連の手順が記載されています。

モバイル IP の管理

Solaris 8 4/01 リリースでモバイル IP の管理が拡張されました。次の情報は、『モバイル IP の管理』に記載されているモバイル IP の管理についての情報を補足するものです。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「Solaris 8 Reference Manual Collection」には記載されていない新しい情報も提供されています。

逆方向トンネリングと専用アドレス

モバイルインターネットプロトコル (モバイル IP) は、ラップトップや無線通信機器などのモバイルコンピュータとの情報の送受信を可能にします。モバイルコンピュータは外部のネットワークに移動しても、元のネットワークにアクセスし、通信することができます。モバイル IP の Solaris による実装では、IPv4 のみをサポートしています。

Solaris 8 4/01 リリースで、モバイル IP では逆方向トンネルのセットアップが可能になりました。モバイルノードの気付アドレスからホームエージェントへ逆方向トンネルを設定することで、IP データパケットについてトポロジカルに正しいソースアドレスを確保することができます。リバーストンネルの使用により、モバイルノードに専用アドレスを割り当てることもできます。

逆方向トンネリングを使用したモバイル IP の導入、および専用アドレスの使用については、『モバイル IP の管理』の「モバイル IP について」を参照してくだ

さい。第 2 章で、これらの新機能の Solaris モバイル IP での実装について記述しています。

システムのリソース管理についてのトピック

このトピックでは、Solaris 環境におけるデバイスドライバの作成について記述します。次の章で構成されます。

第 26 章

新しい機能拡張されたアカウントファイルフォーマットおよび新しい構成オプションを含む、Solaris アカウントソフトウェアにおける更新事項の概要について。

システムアカウントिंगでのリソースの管理

Solaris 8 6/00 リリースで、Solaris のアカウントングソフトウェアが機能拡張されました。ここに記載する情報は、『Solaris のシステム管理 (第 2 巻)』の「アカウントングの設定と管理作業」に記載されたシステムアカウントングの使用によるリソースの管理についての情報を補足するものです。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「Solaris 8 Reference Manual Collection」には記載されていない新しい情報も提供されています。

アカウントング機能の拡張

アカウントングの拡張によって、一般的なアカウントングデータのグループを表す、新しい可変長の汎用アカウントングファイル形式が導入されました。また、カーネルがさまざまアカウントングファイルに記録したリソースの使用状況を構成する機能も導入されました。次に、拡張されたアカウントング機能について説明します。

- タスク — リソースの使用状況を追跡するための新しいプロセス集合
- プロジェクト — リソースの使用状況を請求するための新しい管理データベース。タスクによるリソースの使用をプロジェクトに請求できます。

- `acctadm` — 拡張アカウント機能の様々な属性を構成するための新しいツール。たとえば、アカウントシステムが追跡するリソースをシステム全体に対して構成できます。

新しいデフォルトのアカウント構成は管理が不要で簡単です。しかし、拡張アカウント機能を使用する場合、`/etc/project` ファイルを削除してはなりません。このファイルには、拡張アカウントの構成についての重要な情報が入っています。

このリリースで拡張されたアカウント機能についての詳細な情報は、次の表を参照してください。

情報	参照先
拡張されたアカウント機能の停止および起動について	<code>acctadm(1M)</code>
プロジェクトデータベースの記述について	<code>projects(4)</code>
拡張されたアカウント機能のデータの処理について	<code>libexacct(3LIB)</code> 、 <code>getacct(2)</code> 、 <code>putacct(2)</code> 、 <code>wracct(2)</code>

システムのパフォーマンス管理についてのトピック

このトピックでは、Solaris 環境におけるシステムのパフォーマンスの管理について記述します。次の章で構成されています。

第 28 章

ディレクトリ名検索キャッシュ (DNLC) の機能が拡張され、1000 以上のファイルを持つ大規模なディレクトリ内のファイルにアクセスする時の性能が向上しました。

第 29 章

Solaris 8 1/01 リリースでの『Solaris カーネルのチューンアップ・リファレンスマニュアル』の更新事項の概要を記述します。

DNLC でのシステムパフォーマンスの向上

Solaris 8 6/00 ソフトウェアリリースで、ディレクトリ名検索キャッシュ (DNLC) の機能が拡張されました。ここに記載する情報は、『Solaris のシステム管理 (第 2 巻)』の「システム性能の概要」に記載されているシステムのパフォーマンスの管理についての情報を補足するものです。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「Solaris 8 Reference Manual Collection」には記載されていない新しい情報も提供されています。

DNLC 機能の向上

ディレクトリ名検索キャッシュ (DNLC) の機能が拡張され、1000 以上のファイルを持つ大規模なディレクトリ内のファイルにアクセスする時の性能が向上しました。

DNLC は一般的なファイルシステムサービスであり、最近参照されたディレクトリ名とそれに関連する v ノードをキャッシュに書き込みます。UFS ディレクトリエントリはディスクに直線的に格納されます。つまり、エントリを見つけるには、各エントリを名前で検索する必要があります。新しいエントリを追加する際は、ディレクトリ全体を検索して、その名前が存在していないことを確認する必要があります。この性能における問題を解決するため、DNLC はディレクトリ全体をメモリー (キャッシュ) に書き込みます。

このリリースにおける DNLC のもう一つの新機能は、検索したが存在しなかったファイルオブジェクトをキャッシュに書き込むことです。これは「ネガティブ

キャッシング」と呼びます。ファイルが存在するかどうかを繰り返しテストするアプリケーションに便利です。

次の節では、調整可能な新しい DNLC パラメータについて説明します。このようなパラメータは最適に設定されており、通常は変更すべきではありません。

注 - MAXUINT は符号なし整数の最大値です。

`dnlc_dir_enable`

説明	大規模ディレクトリキャッシングを有効にする
データ型	符号なし整数
デフォルト値	1 (有効)
範囲	0 (無効)、1 (有効)
変更するとき	ディレクトリキャッシングに現在既存の障害はありません。しかし、問題が発生した場合は、 <code>dnlc_dir_enable</code> を 0 に設定し、キャッシングを無効にしてください。

`dnlc_dir_min_size`

説明	1つのディレクトリに対してキャッシュできるエントリの最小数
データ型	符号なし整数
デフォルト値	40
範囲	0 から MAXUINT まで
変更するとき	小規模ディレクトリのキャッシングで性能の問題が発生した場合、 <code>dnlc_dir_min_size</code> の値を増やしてください。個々のファイルシステムがディレクトリのキャッシングについて独自の制限を持つ場合もあるので注意してください。たとえば、UFS は各エントリが 16 バイトであると仮定し、ディレクトリの最小数を <code>ufs_min_dir_cache</code> バイト (約 1024 エントリ) に制限します。

`dnlc_dir_max_size`

説明	キャッシュされるまでのディレクトリエントリの最大数
データ型	符号なし整数
デフォルト値	MAXUINT
範囲	0 から MAXUINT まで
変更するとき	大規模ディレクトリで性能の問題が発生した場合、 <code>dnlc_dir_max_size</code> の値を減らしてください

パフォーマンス向上のためのシステムの チューニング

Solaris 8 1/01 リリースで『Solaris カーネルのチューンアップ・リファレンスマニュアル』が拡張されました。

注 - 最新のマニュアルページを参照するには、man コマンドを使用してください。Solaris 8 Update リリースのマニュアルページには、「Solaris 8 Reference Manual Collection」には記載されていない新しい情報も提供されています。

『Solaris カーネルのチューンアップ・リファレンスマニュアル』の変更点

Solaris 8 1/01 リリースで『Solaris カーネルのチューンアップ・リファレンスマニュアル』が更新されました。

semsys:seminfo_semmnu パラメータの情報がマニュアルに追加されています。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』を参照してください。