

**Oracle® Solaris 11.3 でのネットワークファイ
ルシステムの管理**

ORACLE®

Part No: E62641
2016 年 11 月

Part No: E62641

Copyright © 2002, 2016, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

OracleおよびJavaはオラクル およびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。

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

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。適用されるお客様とOracle Corporationとの間の契約に別段の定めがある場合を除いて、Oracle Corporationおよびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。適用されるお客様とOracle Corporationとの間の契約に定めがある場合を除いて、Oracle Corporationおよびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

ドキュメントのアクセシビリティについて

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWeb サイト(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>)を参照してください。

Oracle Supportへのアクセス

サポートをご契約のお客様には、My Oracle Supportを通して電子支援サービスを提供しています。詳細情報は(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>)か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>)を参照してください。

目次

このドキュメントの使用法	11
1 ネットワークファイルシステムについて	13
NFS サービスについて	13
autofs について	14
NFS の用語	15
NFS サーバーとクライアント	15
NFS ファイルシステム	15
NFS サービスの機能	16
NFS Version 2 プロトコル	16
NFS Version 3 プロトコル	16
NFS Version 4 プロトコル	17
NFS バージョンの制御	18
NFS ACL サポート	18
TCP 経由の NFS	19
UDP 経由の NFS	19
RDMA 経由の NFS の概要	19
ネットワークロックマネージャーと NFS	20
NFS 大規模ファイルのサポート	20
NFS クライアントフェイルオーバー	20
NFS サービスのための Kerberos のサポート	20
WebNFS サポート	21
RPCSEC_GSS セキュリティー方式	21
NFS マウントの拡張機能	21
WebNFS サービスのセキュリティーネゴシエーション	22
NFS サーバーロギング	22
autofs の機能	22
Oracle Solaris 11 リリースでの重要な変更	23

2 ネットワークファイルシステムの機能	25
NFS サービスのしくみ	25
RDMA 経由の NFS	26
NFS におけるバージョンのネゴシエーション	28
NFS Version 4 における機能	29
UDP と TCP のネゴシエーション	39
ファイル転送サイズのネゴシエーション	39
NFS Version 3 でのファイルシステムのマウント方法	40
マウント時の <code>-public</code> オプションと NFS URL の意味	41
クライアント側フェイルオーバー	41
NFS サーバーログ機能のしくみ	44
WebNFS サービスのしくみ	44
WebNFS セキュリティーネゴシエーション機能のしくみ	45
Web ブラウザの使用と比較した場合の WebNFS の制約	46
Secure NFS システム	47
ミラーマウントのしくみ	50
ミラーマウントを使用してファイルシステムをマウントする	50
ミラーマウントを使用してファイルシステムをアンマウントする	51
NFS リフェラルのしくみ	52
どのような場合に NFS リフェラルを使用するか	52
NFS リフェラルの作成	52
NFS リフェラルの削除	52
autofs のしくみ	52
autofs のネットワークナビゲート	54
autofs マップ	55
autofs のナビゲーションプロセス開始法	60
autofs マウントプロセス	61
autofs がクライアント用のもっとも近い読み取り専用ファイルを選択する 方法	63
autofs と重み付け	67
autofs マップエントリ内の変数	67
他のマップを参照するマップ	68
実行可能な autofs マップ	69
ネームサービスに対する autofs のデフォルトの動作	70
autofs リファレンス	72
autofs とメタキャラクタ	72
autofs と特殊文字	73

3 ネットワークファイルシステムの管理	75
ネットワークファイルシステムの管理について	75
ファイルシステムの自動共有	76
ファイルシステム共有のタスクマップ	76
▼ ファイルシステム自動共有を設定する方法	76
▼ NFS サーバーログを有効にする方法	77
ファイルシステムのマウント	78
ファイルシステムマウントのタスクマップ	78
▼ ブート時にファイルシステムにマウントする方法	79
▼ コマンド行からファイルシステムをマウントする方法	80
オートマウンタによるマウント	80
▼ サーバーからすべてのファイルシステムをマウントする方法	81
▼ クライアント側フェイルオーバーを使用する方法	81
▼ 1つのクライアントに対するマウントアクセスを無効にする方法	82
▼ ファイアウォールを越えて NFS ファイルシステムをマウントする方 法	82
NFS URL を使用した NFS ファイルシステムのマウント	83
マウントに使用できるファイルシステムに関する情報の表示	84
NFS サービスの設定	85
NFS サービスの起動と停止	85
オートマウンタの起動と停止	85
異なるバージョンの NFS の選択	86
Secure NFS システムの管理	89
▼ DH 認証を使用して Secure NFS 環境を設定する方法	89
WebNFS の管理	91
WebNFS アクセスの計画	91
▼ WebNFS アクセスを有効にする方法	92
ブラウザを使用した NFS URL へのアクセス	93
ファイアウォール経由 WebNFS アクセスの有効化	94
NFS リフェラルの管理	94
▼ NFS リフェラルの作成とアクセスの方法	94
▼ NFS リフェラルを削除する方法	95
FedFS の管理	95
FedFS サーバー用の DNS レコードの設定	95
▼ 名前空間データベースの作成方法	96
▼ NSDB へのセキュアな接続を使用する方法	97
▼ FedFS リフェラルを作成する方法	97

4 autofs の管理	99
autofs 管理	99
SMF パラメータを使用して autofs 環境を構成する	100
▼ SMF パラメータを使用して autofs 環境を構成する方法	101
マップの管理タスク	101
マップの変更	102
▼ マスターマップを修正する方法	102
▼ 間接マップを修正する方法	102
▼ 直接マップを修正する方法	103
マウントポイント競合の回避	103
非 NFS ファイルシステムへのアクセス	104
オートマウンタのカスタマイズ	104
/home の共通ビューの設定	104
▼ 複数のホームディレクトリファイルシステムで /home を設定する方 法	105
▼ 共通ディレクトリにプロジェクト関連ファイルを統合する方法	106
▼ 共有名前空間にアクセスするために異なるアーキテクチャーを設定 する方法	108
▼ 非互換のクライアントオペレーティングシステムのバージョンをサ ポートする方法	109
▼ 複数のサーバーを通じて共用ファイルを複製する方法	109
autofs セキュリティー制限	110
▼ autofs で公開ファイルハンドルを使用する方法	110
▼ autofs で NFS URL を使用する方法	111
autofs ブラウザビリティの無効化	111
5 ネットワークファイルシステムを管理するためのコマンド	115
NFS コマンド	115
automount コマンド	116
clear_locks コマンド	117
fsstat コマンド	117
mount コマンド	118
umount コマンド	124
mountall コマンド	125
umountall コマンド	125
sharectl コマンド	126
share コマンド	128
unshare コマンド	134
shareall コマンド	134

unshareall コマンド	134
showmount コマンド	135
nfsref コマンド	136
FedFS コマンド	136
6 ネットワークファイルシステムのトラブルシューティング	139
NFS のトラブルシューティングの方法	139
NFS のトラブルシューティング用のコマンド	140
nfsstat コマンド	140
pstack コマンド	142
rpcinfo コマンド	143
snoop コマンド	145
truss コマンド	145
NFS トラブルシューティング手順	146
▼ NFS クライアントの接続性を確認する方法	147
▼ NFS サーバーをリモートで確認する方法	148
▼ サーバーで NFS サービスを確認する方法	149
▼ NFS サービスを再起動する方法	150
NFS サービスを提供するホストの識別	151
▼ mount コマンドで使用されたオプションを確認する方法	151
autofs のトラブルシューティング	152
automount -v で生成されるエラーメッセージ	152
その他のエラーメッセージ	153
autofs のその他のエラー	156
NFS エラーメッセージ	156
7 ネットワークファイルシステムへのアクセス	163
NFS ファイル	163
/etc/default/nfslogd ファイル	164
/etc/nfs/nfslog.conf ファイル	165
NFS デーモン	167
automountd デーモン	167
lockd デーモン	168
mountd デーモン	169
nfs4cbd デーモン	170
nfsd デーモン	170
nfslogd デーモン	171
nfsmapid デーモン	172

rearsed デーモン	178
statd デーモン	178
索引	181

このドキュメントの使用方法

- **概要** - ネットワークファイルシステムの管理およびアクセス方法について説明します。
- **対象読者** - 技術者、システム管理者、および認定サービスプロバイダ。
- **前提知識** - 基本および高度なネットワーク管理技術。

製品ドキュメントライブラリ

この製品および関連製品のドキュメントとリソースは <http://www.oracle.com/pls/topic/lookup?ctx=E62101-01> で入手可能です。

フィードバック

このドキュメントに関するフィードバックを <http://www.oracle.com/goto/docfeedback> からお聞かせください。

ネットワークファイルシステムについて

この章では、ネットワーク経由でファイルシステムにアクセスするために使用できる、ネットワークファイルシステム (NFS) サービスの概要を説明します。NFS サービスによって、任意のシステムからほかのシステムのファイルシステムへのアクセスが可能になります。システムはネットワーク上で任意の特定の時点で、クライアント、サーバー、またはその両方の役割を担うことができます。autofs は、NFS サービスで共有されるファイルシステムをマウントするために使用される、クライアント側サービスです。autofs は、自動マウントを提供するファイルシステム構造です。また、NFS サービスを理解するために必要な概念、および NFS と autofs の最新の機能についても説明します。

この章の内容は、次のとおりです。

- 15 ページの「NFS の用語」
- 13 ページの「NFS サービスについて」
- 14 ページの「autofs について」
- 16 ページの「NFS サービスの機能」
- 23 ページの「Oracle Solaris 11 リリースでの重要な変更」

注記 - システムでゾーンが有効になっていて、非大域ゾーンで NFS を使用する必要がある場合は、『Oracle Solaris 10 ゾーンの作成と使用』の「NFS クライアントとして機能する非大域ゾーン」を参照してください。

NFS サービスについて

NFS サービスにより、さまざまなオペレーティングシステムを実行するさまざまなアーキテクチャーのシステムが、ネットワーク経由でファイルシステムを共有できます。

NFS 環境は、NFS はアーキテクチャー仕様ではなくファイルシステムの抽象モデルを定義するものなので、さまざまなオペレーティングシステムに実装できます。各 OS はそのファイルシステムセマンティクスに NFS モデルを適用します。このモデルは、

読み取りや書き込みなどのファイルシステム操作がローカルファイルにアクセスしているかのように機能することを意味します。

NFS サービスには次の利点があります。

- 複数のシステムが同じファイルを使用できるため、ネットワーク上のだれもが同じデータにアクセスできる
- ユーザーアプリケーションごとのローカルディスク領域を必要とする代わりに、システム共有アプリケーションを持つことでストレージコストが少なくなる
- すべてのユーザーが同一セットのファイルを読み取るので、データの整合性と信頼性が向上する
- ファイルシステムをユーザーに透過的な形でマウントできる
- リモートファイルに透過的にアクセスできる
- 異機種混在環境をサポートする
- システム管理の手間を省ける

NFS サービスによって、ファイルシステムの物理場所がユーザーに関係なくなります。よく使用されるファイルを各システムに配置する代わりに、NFS サーバーのファイルシステムからオリジナルファイルを共有できます。他のシステムからネットワークを通じてアクセスできるようにします。NFS オペレーションでは、リモートファイルシステムとローカルファイルシステムの区別がありません。

autofs について

NFS サービスで共有されるファイルシステムは、自動マウントを使用してマウントできます。クライアント側サービス **autofs** は、自動マウントを提供するファイルシステム構造です。**autofs** のファイルシステムは、**automount** で作成されます。**automount** は、システムをブートすると自動的に実行されます。自動マウントデーモン **automountd** は、必要に応じてリモートファイルシステムをマウントおよびアンマウントしながら、継続的に動作しています。

automountd を実行しているクライアントシステムがリモートファイルシステムにアクセスしようとするときは、常にデーモンがリモートファイルシステムをマウントします。このリモートファイルシステムは、必要な間はマウントされたままです。リモートファイルシステムが一定期間アクセスされない場合は、自動的にアンマウントされます。

ブート時にマウントが不要なので、ユーザーはディレクトリをマウントするためにスーパーユーザーパスワードを知っている必要がなくなりました。ユーザーは **mount** および **umount** コマンドを使用する必要がありません。**autofs** は、ユーザーの介入なしに、必要に応じてファイルシステムをマウントまたはアンマウントします。

automountd で一部のファイルシステムをマウントすることで、ほかのシステムを **mount** でマウントできなくなることはありません。ディスクレスコンピュータ

は、`mount` コマンドと `/etc/vfstab` ファイルを使用して `/` (ルート)、`/usr`、および `/usr/kvm` をマウントする必要があります。

autofs サービスの詳細は、次を参照してください。

- [表6](#)
- [52 ページの「autofs のしくみ」](#)

NFS の用語

このセクションには、NFS サービスを操作するために理解する必要がある基本用語が含まれています。NFS サービスに関する詳細は、[第5章「ネットワークファイルシステムを管理するためのコマンド」](#)を参照してください。

NFS サーバーとクライアント

用語「クライアント」と「サーバー」は、システムがファイルシステムを共有するときに担う役割を説明しています。サーバーは、ネットワークを介してそのファイルシステムを共有するシステムです。これらのファイルシステムにアクセスするシステムがクライアントです。

クライアントは、サーバーの共有ファイルシステムをマウントすることによってサーバーのファイルにアクセスします。クライアントがリモートファイルシステムをマウントしたとき、ファイルシステムがコピーされるわけではありません。マウントプロセスでは一連のリモートプロシージャ呼び出しによって、クライアントからサーバーの共有ファイルシステムに透過的にアクセスできるようになります。マウントはローカルマウントのように行われます。ファイルシステムがローカルであるかのようにコマンドを入力できます。詳細は、[78 ページの「ファイルシステムのマウント」](#)を参照してください。

サーバーのファイルシステムは、NFS オペレーションによって共有すると、クライアントからアクセスできるようになります。NFS ファイルシステムは、`autofs` を使用すると自動的にマウントできます。`autofs` については、[14 ページの「autofs について」](#)を参照してください。`share` コマンドと `autofs` に関連するタスクについては、[76 ページの「ファイルシステムの自動共有」](#) および [表6](#)を参照してください。

NFS ファイルシステム

NFS サービスで共有できるオブジェクトには、単一ファイルを含めて、ディレクトリツリー (ファイル階層) の全体またはその一部が含まれます。すでに共有されている

ファイル階層を覆うファイル階層は共有できません。モデムやプリンタなどの周辺機器も共有できません。

ほとんどの UNIX システム環境では、共有可能なファイル階層は、ファイルシステムまたはファイルシステムの一部に対応します。ただし、NFS はオペレーティングシステムをまたがって機能し、ファイルシステム概念は非 UNIX 環境で意味がない可能性があるため、用語「ファイルシステム」は、NFS で共有およびマウントできるファイルまたはファイル階層を指します。

NFS サービスの機能

このセクションでは、NFS サービスの重要な機能について説明します。

NFS Version 2 プロトコル

NFS プロトコルの最初のバージョンである NFS Version 2 は広く使用されています。すべての Oracle Solaris リリースが NFS Version 2 プロトコルをサポートします。

NFS Version 3 プロトコル

NFS Version 2 プロトコルとは異なり、NFS Version 3 プロトコルは 2G バイトより大きいファイルを扱えます。NFS での大きなファイルの扱いについては、[20 ページの「NFS 大規模ファイルのサポート」](#)を参照してください。

NFS Version 3 プロトコルではサーバー上で安全な非同期書き込みが可能なので、サーバーがクライアント書き込み要求をメモリー内にキャッシュすることを許可することでパフォーマンスが向上します。サーバーが変更をディスクにコミットするのをクライアントが待機する必要がなくなるため、応答時間が速くなります。サーバーは要求をバッチ処理することもできるので、サーバー上の応答時間も短縮されました。

多くの Solaris NFS Version 3 操作は、ローカルキャッシュに保存されている、ファイル属性を返します。キャッシュがより頻繁に更新されるため、このデータを更新するために別個の操作の実行が必要になる頻度が少なくなります。したがって、サーバーへのリモートプロシージャ呼び出し (RPC) の回数が減少し、パフォーマンスが向上します。

ファイルアクセス権の検証処理が改善されました。version 2 では、ユーザーが適切なアクセス権を持っていないリモートファイルをコピーしようとする、「書き込みエラー」や「読み取りエラー」というメッセージが出力されました。Version 3 では、

ファイルが開かれる前にアクセス権がチェックされるため、エラーが「オープンエラー」として報告されます。

NFS Version 3 プロトコルでは、8K バイト転送サイズ制限が解除されました。クライアントとサーバーは、Version 2 で適用された 8K バイト制限に準拠する代わりに、クライアントとサーバーがサポートする転送サイズにかかわらずネゴシエートできます。以前の Solaris の実装では、デフォルトで、プロトコルが 32K バイトの転送サイズに設定されていることに注意してください。Oracle Solaris 10 リリースから、書き込み転送サイズの制限が緩和されました。使用するトランスポートプロトコルに基づいて転送サイズが決定されるようになりました。

NFS Version 4 プロトコル

NFS Version 4 プロトコルでは、ユーザー ID とグループ ID を文字列として表現します。nfsmapid デーモンは、NFS Version 4 クライアントおよびサーバーで次のマッピングに使用されます。

- ユーザー ID およびグループ ID 文字列をローカル値 ID にマップする
- ローカル数値 ID をユーザー ID およびグループ ID 文字列にマップする

nfsmapid デーモンの詳細は、[167 ページの「NFS デーモン」](#) を参照してください。

NFS Version 4 では、サーバー上のアクセス制御リスト (ACL) エントリ内のユーザー ID またはグループ ID をクライアント上の ACL エントリ内のユーザー ID またはグループ ID にマッピングするために、nfsmapid デーモンが使用されます。逆も同じです。ユーザー ID とグループ ID のマッピングの詳細は、[37 ページの「NFS Version 4 での ACL と nfsmapid」](#) および[18 ページの「NFS ACL サポート」](#) を参照してください。

NFS Version 4 では、ファイルシステムを共有解除すると、そのファイルシステム内のオープンファイルまたはファイルロックのすべての状態情報が破棄されます。NFS Version 3 では、ファイルシステムが共有解除される前にクライアントが取得したロックをサーバーが保守します。NFS Version 4 でのファイルシステム共有解除の詳細は、[29 ページの「NFS Version 4 におけるファイルシステムの共有解除と再共有」](#) を参照してください。

NFS Version 4 サーバーは擬似ファイルシステムを使用して、サーバー上でエクスポートされたオブジェクトへのアクセスをクライアントに提供します。擬似ファイルシステムの詳細は、[30 ページの「NFS Version 4 でのファイルシステム名前空間」](#) を参照してください。NFS Version 4 は揮発性ファイルハンドルをサポートします。詳細は、[32 ページの「NFS Version 4 における揮発性ファイルハンドル」](#) を参照してください。

委託とは、サーバーがファイルの管理をクライアントに委託するテクニックです。委託は、クライアントとサーバーの両方でサポートされます。たとえば、サーバーは、

読み取り委託または書き込み委託のいずれかをクライアントに付与できます。委託の詳細は、[35 ページの「NFS Version 4 における委託」](#)を参照してください。

NFS Version 4 は LIPKEY/SPKM セキュリティーをサポートしません。

また、NFS Version 4 は次のデーモンを使用しません。

- lockd
- nfslogd
- statd

NFS Version 4 での機能の完全なリストについては、[29 ページの「NFS Version 4 における機能」](#)を参照してください。

NFS サービスの設定については、[85 ページの「NFS サービスの設定」](#)を参照してください。

NFS バージョンの制御

SMF リポジトリには、クライアントとサーバーの両方によって使用される NFS プロトコルを制御するためのパラメータが含まれています。たとえば、パラメータを使用して、バージョンネゴシエーションを管理できます。クライアントおよびサーバーパラメータの詳細は、[167 ページの「NFS デーモン」](#)を参照してください。NFS デーモンのパラメータ値の詳細は、[nfs\(4\)](#)のマニュアルページを参照してください。

NFS ACL サポート

アクセス制御リスト (ACL) は、UNIX 標準ファイルアクセス権を使用する代わりに、ファイルアクセス権を設定するメカニズムを提供します。NFS ACL サポートは、Oracle Solaris NFS クライアントから Oracle Solaris NFS サーバーへ ACL エントリを変更および表示する方法を提供します。

NFS Version 2 および Version 3 実装は、古い POSIX ドラフトスタイル ACL をサポートします。POSIX ドラフト ACL は、UFS によりネイティブでサポートされます。POSIX ドラフト ACL の詳細は、『[Oracle Solaris 11.3 でのファイルのセキュリティ保護とファイル整合性の検証](#)』の「[アクセス制御リストによる UFS ファイルの保護](#)」を参照してください。

NFS Version 4 プロトコルは NFS Version 4 スタイル ACL をサポートします。NFS Version 4 ACL は Oracle Solaris ZFS でネイティブサポートされます。NFS Version 4 ACL のあらゆる機能を利用するには、NFS Version 4 サーバーで ZFS を基盤となるファイルシステムとして使用する必要があります。NFS Version 4 ACL は、豊富な継

承プロパティセットと、標準の読み取り、書き込み、および実行より多くのアクセス権ビットセットを備えています。ZFS ファイルを保護するために ACL を使用する方法の詳細については、『[Oracle Solaris 11.3 での ZFS ファイルシステムの管理](#)』の第 9 章、「[ACL および属性を使用した Oracle Solaris ZFS ファイルの保護](#)」を参照してください。NFS Version 4 での ACL のサポートの詳細は、[37 ページの「NFS Version 4 での ACL と nfsmapid」](#)を参照してください。

TCP 経由の NFS

NFS プロトコルのデフォルトトランスポートプロトコルは、TCP (Transport Control Protocol) です。TCP は、低速ネットワークとワイドエリアネットワークのパフォーマンスの向上に役立ちます。TCP には、トラフィック抑制機能とエラー回復機能もあります。TCP 経由の NFS は、NFS Version 2、NFS Version 3、および NFS Version 4 プロトコルで動作します。

注記 - InfiniBand ハードウェアがシステムで使用可能な場合は、デフォルトトランスポートプロトコルが TCP から Remote Direct Memory Access (RDMA) プロトコルに変わります。詳細は、[19 ページの「RDMA 経由の NFS の概要」](#)および [26 ページの「RDMA 経由の NFS」](#)を参照してください。proto=tcp マウントオプションを使用する場合、NFS マウントは TCP のみを使用するように強制されます。

UDP 経由の NFS

Oracle Solaris 11 リリースから、NFS クライアントは 1 つの UDP (User Datagram Protocol) 予約済みポート (構成可能) のみを使用します。システムパフォーマンスを向上するために、複数のポートを使用するようにシステムを構成できます。この機能は、最初からこのように構成可能であった TCP 経由 NFS を、忠実に反映しています。NFS 環境の調整の詳細は、『[Oracle Solaris 11.3 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。

RDMA 経由の NFS の概要

InfiniBand ハードウェアがシステムで使用可能な場合は、デフォルトトランスポートプロトコルが TCP から RDMA プロトコルに変わります。RDMA プロトコルは、高速ネットワーク経由でデータをメモリー間転送するためのテクノロジーです。特に、RDMA により、CPU の介入なしでメモリーにリモートデータ転送を直接行えます。この機能を提供するために、RDMA は InfiniBand のインターコネクト I/O テクノロジーと Oracle Solaris OS を結合します。ただし、proto=tcp マウントオプションを使用する

場合、NFS マウントは TCP のみを使用するように強制されます。NFS で RDMA プロトコルを使用する方法の詳細は、[26 ページの「RDMA 経由の NFS」](#)のマニュアルページを参照してください。

ネットワークロックマネージャーと NFS

ネットワークロックマネージャーは、NFS 経由で共有されるファイルに UNIX レコードロックを提供します。このロックメカニズムによって、クライアントはその I/O 要求をほかのクライアントと同期できるため、データ整合性が確保されます。

注記 - ネットワークロックマネージャーは、NFS Version 2 と NFS Version 3 マウントでのみ使用されます。ファイルロックは NFS Version 4 プロトコルに組み込まれていません。

NFS 大規模ファイルのサポート

NFS Version 3 プロトコルでは 2G バイトより大きなファイルを扱えますが、NFS Version 2 プロトコルではできません。

NFS クライアントフェイルオーバー

読み取り専用ファイルシステムの動的フェイルオーバーは、マニュアルページ、その他のドキュメント、共有バイナリなどのすでに複製された読み取り専用リソースに高いレベルの可用性を提供します。フェイルオーバーは、ファイルシステムマウント後はいつでも発生できます。手動マウントでは、今までのリリースのオートマウントのように複数の複製を一覧表示できるようになりました。オートマウントは、ファイルシステムが再マウントされるまでフェイルオーバーが待機することがなくなった点以外は、変更されていません。詳細は、[81 ページの「クライアント側フェイルオーバーを使用する方法」](#)と [41 ページの「クライアント側フェイルオーバー」](#)を参照してください。

NFS サービスのための Kerberos のサポート

NFS サービスは、Kerberos をサポートするように NFS クライアントおよびサーバーを構成するときに、Kerberos Version 5 認証、整合性、およびプライバシーをサポートします。安全な認証のために Kerberos を使用するとき、`mount` および `share` コマンド行オプションを使用できます。Kerberos Version 5 認証については、『[Oracle Solaris](#)

11.3 での [Kerberos およびその他の認証サービスの管理](#)』の「[Kerberos NFS サーバーの構成](#)」を参照してください。

WebNFS サポート

WebNFS は、ファイアウォール経由でアクセス可能なファイルシステムをインターネット上に作成する機能を提供します。この機能は NFS プロトコルの拡張機能を使用します。インターネットアクセスに WebNFS™ プロトコルを使用する利点の 1 つは、その信頼性です。このサービスは、NFS Version 3 および Version 2 プロトコルの拡張として構築されます。さらに、WebNFS では匿名 ftp サイトの管理オーバーヘッドなしでファイルを共有できます。WebNFS の詳細は、[22 ページの「WebNFS サービスのセキュリティーネゴシエーション](#)」 および [91 ページの「WebNFS の管理](#)」を参照してください。

注記 - NFS Version 4 プロトコルは、WebNFS サービスに優先します。NFS Version 4 は、MOUNT プロトコルと WebNFS サービスに追加されたすべてのセキュリティーネゴシエーションを完全に統合します。

RPCSEC_GSS セキュリティー方式

Solaris 7 リリースでは、RPCSEC_GSS と呼ばれるセキュリティーフレーバーがサポートされます。この方式では、標準的な GSS-API インタフェースを使用して、認証、一貫性、機密性を実現し、複数のセキュリティーメカニズムをサポートしています。Kerberos V5 認証のサポートについての詳細は、[20 ページの「NFS サービスのための Kerberos のサポート](#)」を参照してください。GSS-API の詳細は、『[Oracle Solaris 11 セキュリティー開発者ガイド](#)』を参照してください。

NFS マウントの拡張機能

NFS サービスは、Oracle Solaris の mount および automountd コマンドへの拡張機能を提供します。これらの拡張機能によって、マウント要求は MOUNT プロトコルの代わりに公開ファイルハンドルを使用できます。WebNFS サービスは MOUNT プロトコルをアクセスとして使用します。公開ファイルハンドルを使用することで、ファイル経由でマウントが発生できます。サーバーとクライアント間のトランザクションが減少するため、マウント発生が速くなります。

この拡張機能によって、標準パス名の代わりに NFS URL を使用することもできます。また、mount コマンドとオートマウントマップで public オプションを使用して、公開ファイルハンドルの使用を強制できます。WebNFS サービスの詳細は、[21 ページの「WebNFS サポート](#)」を参照してください。

WebNFS サービスのセキュリティーネゴシエーション

NFS サービスにより、WebNFS クライアントは NFS サーバーとセキュリティーメカニズムをネゴシエートできます。WebNFS クライアントは NFS サーバーとセキュリティーメカニズムをネゴシエートするためのプロトコルを使用します。このプロトコルによって、WebNFS サービスで安全なトランザクションを使用できます。WebNFS のセキュリティーネゴシエーションの詳細は、[45 ページの「WebNFS セキュリティーネゴシエーション機能のしくみ」](#) を参照してください。

NFS サーバーロギング

注記 - NFS Version 4 はサーバーロギング機能をサポートしません。

NFS サーバーロギングによって、NFS サーバーはファイルシステムで実行されたファイル操作の記録を提供できます。この記録には、どのファイルが、いつ、だれによってアクセスされたかという情報が含まれています。一連の構成オプションを使用して、これらの情報を含むログの場所を指定することができます。これらのオプションを使用して、ログに記録する操作を選択することもできます。NFS サーバーロギング機能は、NFS および WebNFS クライアントに対して匿名 FTP アーカイブを利用可能にするサイトで特に便利です。詳細は、[77 ページの「NFS サーバーログを有効にする方法」](#) を参照してください。

autofs の機能

autofs は、ローカル名前空間に指定されているファイルシステムで機能します。この情報は、LDAP、NIS (ネットワーク情報サービス)、またはローカルファイルで保守できます。Autofs は次の機能をサポートします。

- 完全にマルチスレッド化されたバージョンの automountd 機能により、autofs の信頼性が向上します。この機能により、複数マウントを並行して提供できるため、サーバーが使用できなくなった場合でもサービスのハングアップを回避できます。
- automountd 機能はオンデマンドマウントも提供します。最上位ファイルシステムのみがマウントされます。そのマウントポイントに関係する他のファイルシステムは、必要に応じてマウントされます。
- autofs サービスは間接マップの「ブラウザビリティ」をサポートします。このサポートにより、ユーザーは各ファイルシステムを実際にマウントすることなく、どのディレクトリをマウントできるかを確認できます。-nobrowse オプションにより、/net や /home などの大きなファイルシステムが自動的にブラウザ可能にならないことが保証されます。-automount コマンドで n オプションを使用するこ

とによって、autofs ブラウザビリティをクライアントごとにオフにすることもできます。autofs ブラウザビリティを無効にするためのさまざまな方法の詳細は、[111 ページの「autofs ブラウザビリティの無効化」](#)を参照してください。

Oracle Solaris 11 リリースでの重要な変更

Oracle Solaris 11 リリースには、次の拡張機能が含まれています。

- 新しいプロパティ `nfs_props/showmount_info` が `/network/nfs/server:default` サービスに追加されました。このプロパティは、`showmount` コマンドがリモートクライアントに表示する情報を制御します。`nfs_props/showmount_info` プロパティの詳細は、[showmount\(1M\)](#) のマニュアルページを参照してください。
- フェデレーテッドファイルシステム (FedFS) リフェラルのサポートが追加されました。この機能により、いくつかのサーバーのリフェラル情報を LDAP で集中管理できます。FedFS リフェラルの詳細は、[95 ページの「FedFS の管理」](#)を参照してください。
- `/etc/default/autofs` と `/etc/default/nfs` を編集することで設定するために使用される構成プロパティが、サービス管理機能 (SMF) リポジトリ内で設定できるようになりました。新しい SMF プロパティと、新しい SMF プロパティを使用するデーモンについての詳細は、[167 ページの「NFS デーモン」](#)を参照してください。
- NFS サービスはミラーマウントのサポートを提供します。ミラーマウントにより、NFS Version 4 クライアントはサーバー名前空間内で共有ファイルシステムのマウントポイントをトラバースできます。NFS Version 4 マウントの場合、オートマウンタはサーバー名前空間ルートのマウントを実行し、ミラーマウントを利用してそのファイルシステムにアクセスします。オートマウンタに対してミラーマウントが提供する主な利点は、ミラーマウントを使用してファイルシステムをマウントすると、自動マウントマップ管理に関連するオーバーヘッドが必要なくなることで、ミラーマウントは次の機能を提供します。
 - 名前空間の変更はすべてのクライアントでただちに認識されます。
 - 新しい共有ファイルシステムは即時に発見され、自動的にマウントされます。
 - ファイルシステムは、一定期間非アクティブである場合、自動的にアンマウントされます。

ミラーマウントの詳細は、次を参照してください。

- [81 ページの「サーバーからすべてのファイルシステムをマウントする方法」](#)
- [50 ページの「ミラーマウントのしくみ」](#)
- NFS サービスに NFS リフェラルが追加されました。リフェラルは、NFS Version 4 クライアントがファイルシステムを見つけるためにたどることができる、サーバーベースリダイレクションです。NFS サーバーは `nfsref` コマンドによって作成され

たリフェラルをサポートします。NFS Version 4 クライアントはこれらのリフェラルにたどって、実際の場所からファイルシステムをマウントします。オートマウンタマップの編集はリフェラルの作成に置き換わります。NFS リフェラルは次の機能を提供します。

- ミラーマウントのすべての機能
- オートマウンタに似た機能 (オートマウンタへの依存なし)
- クライアントとサーバーの両方で設定を必要としない

NFS リフェラルの詳細については、次を参照してください。

- [94 ページの「NFS リフェラルの管理」](#)
- [52 ページの「NFS リフェラルのしくみ」](#)
- [nfsref\(1M\)](#) のマニュアルページ
- DNS ドメインルート単位で FedFS 名前空間をマウントできる機能が追加されました。このマウントポイントを NFS リフェラルと一緒に使用して、あるファイルサーバーから別のファイルサーバーにブリッジすることで、任意の大きな名前空間を構築できます。FedFS ドメインルートの詳細は、次を参照してください。
 - [95 ページの「FedFS サーバー用の DNS レコードの設定」](#)
 - [57 ページの「/nfs4 マウントポイント」](#)
- `sharect1` ユーティリティーにより、NFS などのファイル共有プロトコルを構成および管理できます。たとえば、このユーティリティーを使用して、クライアントおよびサーバー操作プロパティーを設定したり、特定のプロトコルのプロパティー値を表示したり、プロトコルのステータスを取得したりできます。詳細は、[sharect1\(1M\)](#) のマニュアルページを参照してください。
- NFS サービスを、予約済みポート (1024 未満のポート番号) を使用するクライアントに限定する機能が追加されました。これは `AUTH_SYS` で共有されているファイルシステムに影響を与えます。これは、`sharect1` コマンドで `resvport` 設定を使用してすべての共有に対して、または `resvport` 共有オプションを使用して個々のファイルシステムに対して設定できます。

◆◆◆ 第 2 章

ネットワークファイルシステムの機能

この章では、Remote Direct Memory Access (RDMA) とその他のトランスポートプロトコルの関係について説明します。RDMA は NFS のデフォルトトランスポートです。この章では、バージョンネゴシエーションや NFS Version 4 で導入されたファイル共有機能を含めて、NFS サービスがどのように動作するかについて説明します。

この章の内容は、次のとおりです。

- 26 ページの「RDMA 経由の NFS」
- 25 ページの「NFS サービスのしくみ」
- 50 ページの「ミラーマウントのしくみ」
- 52 ページの「NFS リフェラルのしくみ」
- 55 ページの「autofs マップ」
- 52 ページの「autofs のしくみ」
- 72 ページの「autofs リファレンス」

注記 - システムでゾーンが有効なときに非大域ゾーンでこの機能を使用する場合は、『Oracle Solaris ゾーンの紹介』を参照してください。

NFS サービスのしくみ

次のセクションでは、NFS の複雑な機能をいくつか紹介します。このセクションで紹介する機能のいくつかは、NFS Version 4 専用です。

- 28 ページの「NFS におけるバージョンのネゴシエーション」
- 29 ページの「NFS Version 4 における機能」
- 39 ページの「UDP と TCP のネゴシエーション」
- 39 ページの「ファイル転送サイズのネゴシエーション」
- 40 ページの「NFS Version 3 でのファイルシステムのマウント方法」
- 41 ページの「マウント時の `-public` オプションと NFS URL の意味」

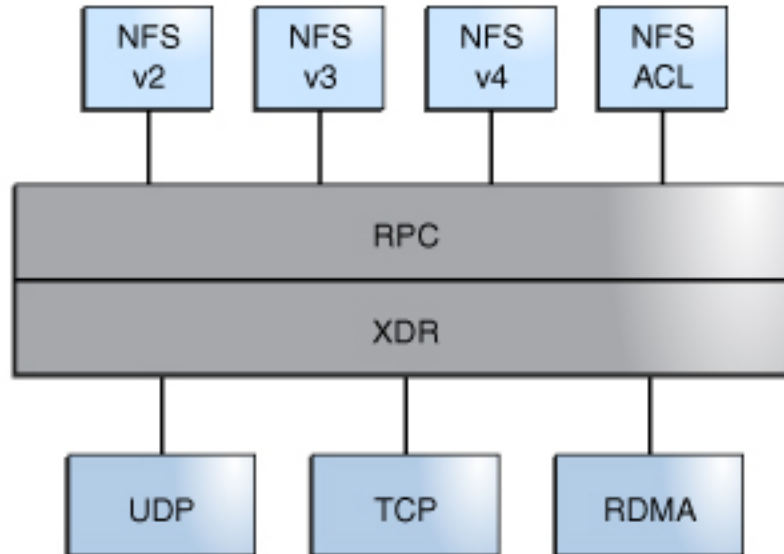
- 41 ページの「クライアント側フェイルオーバー」
- 44 ページの「NFS サーバーログ機能のしくみ」
- 44 ページの「WebNFS サービスのしくみ」
- 46 ページの「Web ブラウザの使用と比較した場合の WebNFS の制約」
- 47 ページの「Secure NFS システム」
- 47 ページの「Secure RPC」

注記 - システムでゾーンが有効なときに非大域ゾーンでこの機能を使用する場合は、『[Oracle Solaris ゾーンを紹介](#)』を参照してください。

RDMA 経由の NFS

Oracle Solaris 11.1 リリース以降、NFS のデフォルトトランスポートは Remote Direct Memory Access (RDMA) プロトコルです。このプロトコルは、高速ネットワークを介したメモリー間データ転送を提供します。特に、RDMA により、CPU の介入なしでメモリーにリモートデータ転送を直接行えます。RDMA は直接データ配置も提供しており、これによってデータコピーがなくなり、さらに CPU 介入がなくなります。このように、RDMA はホストの CPU を解放するだけでなく、ホストメモリーと入出力バスの競合を少なくします。この機能を提供するために、RDMA は、InfiniBand のインターコネクト入出力テクノロジー (SPARC および x86 プラットフォームの両方で使用可能) と Oracle Solaris オペレーティングシステムを結合します。次の図は、UDP や TCP など、その他のプロトコルとの RDMA の関係を示します。

図 1 その他のプロトコルとの RDMA の関係



RDMA は NFS のデフォルトのトランスポートプロトコルなので、クライアントまたはサーバーで RDMA を使用するために特別な share オプションや mount オプションは必要ありません。既存のオートマウントマップ、`vfstab` とファイルシステム共有は、RDMA トランスポートで機能します。クライアントとサーバーの間に InfiniBand 接続が存在するときは、RDMA トランスポート経由の NFS マウントが透過的に実行されます。InfiniBand 接続機能は SPARC と x86 の両方のプラットフォームで動作します。RDMA トランスポートをクライアントとサーバーで使用できない場合、TCP トランスポートが初期フォールバックになります。TCP が使用できない場合は UDP がフォールバックになります。ただし、`proto=rdma` マウントオプションを使用する場合、NFS マウントは RDMA だけを使用するように強制されます。

TCP と UDP のみを使用するように指定するときは、`proto=tcp/udp` mount オプションを使用できます。このオプションは、NFS クライアントの RDMA を無効にします。NFS マウントオプションの詳細は、[mount_nfs\(1M\)](#) および [mount\(1M\)](#) のマニュアルページを参照してください。

注記 - InfiniBand の RDMA は、IP アドレス指定形式および IP ルックアップインフラストラクチャーを使用して、ピアを指定します。ただし、RDMA は、独立したプロトコルスタックであるため、すべての IP のセマンティクスを完全には実装しません。たとえば、RDMA はピアと通信するための IP アドレス指定を使用しません。したがって、RDMA は、IP アドレスに基づいたさまざまなセキュリティーポリシーの構成を省略することがあります。ただし、mount 制限や Secure RPC などの NFS と RPC の管理ポリシーは省略されません。

NFS におけるバージョンのネゴシエーション

NFS 起動プロセスには、サーバーとクライアントのためのプロトコルバージョンレベルのネゴシエーションが含まれています。バージョンのレベルを指定しない場合、デフォルトにより最適なレベルが選択されます。たとえば、クライアントとサーバーの両方が NFS Version 3 をサポートできる場合は、それが使用されます。クライアントまたはサーバーが NFS Version 2 のみをサポートできる場合、それが使用されます。

sharectl コマンドを使用すること

で、`client_versmin`、`client_versmax`、`server_versmin`、および `server_versmax` パラメータを設定できます。これらのパラメータのデフォルト値は、クライアントとサーバーに指定した最小値と最大値によって置き換えられます。クライアントとサーバーの両方で、デフォルトの最小値は 2、デフォルトの最大値は 4 です。サーバーによってサポートされるバージョンを見つけるために、NFS クライアントは `client_versmax` の値から始めて、`client_versmin` のバージョン値に到達するまで各バージョンの試行を続けます。サポートされるバージョンが検出されるとすぐに、この処理は終了します。たとえば、`client_versmax=4` で `client_versmin=2` の場合、クライアントは最初に NFS Version 4、次に NFS Version 3、最後に NFS Version 2 を試行します。`client_versmin` と `client_versmax` が同じ値に設定されている場合は、クライアントは常にこのバージョンを使用し、ほかのバージョンは試行しません。サーバーがこのバージョンをサポートしていない場合、マウントは失敗します。

注記 - NFS のバージョンネゴシエーションによって決まった値をオーバーライドするには、`mount` コマンドで `vers` オプションを使用します。`mount` コマンドで使用可能なコマンドオプションの詳細は、[mount_nfs\(1M\)](#) のマニュアルページを参照してください。

NFS サービスの設定については、85 ページの「[NFS サービスの設定](#)」を参照してください。

NFS Version 4 における機能

このセクションでは、NFS Version 4 で導入された機能について説明します。

- [29 ページの「NFS Version 4 におけるファイルシステムの共有解除と再共有」](#)
- [30 ページの「NFS Version 4 でのファイルシステム名前空間」](#)
- [32 ページの「NFS Version 4 における揮発性ファイルハンドル」](#)
- [33 ページの「NFS Version 4 におけるクライアント回復」](#)
- [35 ページの「NFS Version 4 における OPEN 共有サポート」](#)
- [35 ページの「NFS Version 4 における委託」](#)
- [37 ページの「NFS Version 4 での ACL と nfsmapid」](#)

注記 - Oracle Solaris 10 以降のリリースでは、NFS Version 4 は LIPKEY/SPKM セキュリティーフレーバをサポートしません。また、NFS Version 4 は mountd、nfslogd、および statd デーモンを使用しません。

NFS サービスの設定については、[85 ページの「NFS サービスの設定」](#)を参照してください。

NFS Version 4 におけるファイルシステムの共有解除と再共有

NFS Version 3 と Version 4 の両方で、クライアントが共有解除されたファイルシステムにアクセスしようとする、サーバーはエラーコードを返します。ただし、NFS Version 3 では、サーバーはファイルシステムが共有解除される前にクライアントが取得したロックを保持します。したがって、ファイルシステムが再共有されると、NFS Version 3 クライアントは、そのファイルシステムが共有解除されなかったかのようにファイルシステムにアクセスできます。

NFS Version 4 では、ファイルシステムが共有解除されると、そのファイルシステム内のオープンファイルまたはファイルロックに関するすべての状態情報が破棄されます。クライアントは、これらのファイルにアクセスしようとしたりロックしようとしたらすると、エラーを受け取ります。このエラーは通常、アプリケーションに対する入出力エラーとして報告されます。ただし、オプションを変更するために現在共有されているファイルシステムを再共有しても、サーバーの状態情報は破棄されません。

NFS Version 4 のクライアント回復については、[33 ページの「NFS Version 4 におけるクライアント回復」](#)を参照してください。unshare コマンドで使用可能なコマンドオプションについては、[unshare_nfs\(1M\)](#)のマニュアルページを参照してください。

NFS Version 4 でのファイルシステム名前空間

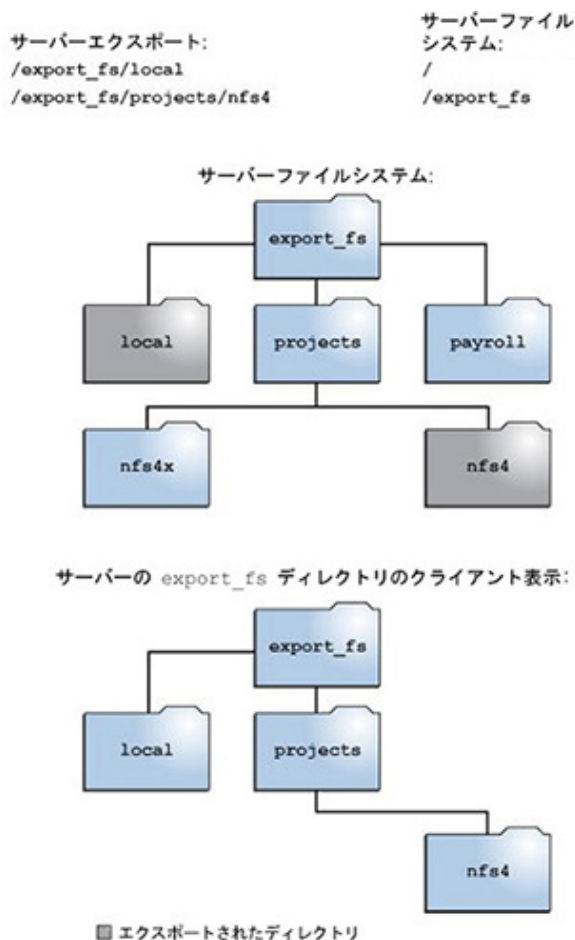
NFS Version 4 サーバーは、サーバー上のすべてのエクスポート済みファイルへのシームレスアクセスを NFS クライアントに提供する、擬似ファイルシステムを作成および保守します。NFS Version 4 以前は、擬似ファイルシステムは存在しませんでした。NFS クライアントは、アクセスする各共有サーバーのファイルシステムに強制的にマウントされます。

擬似ファイルシステムは、ディレクトリだけを含む構造で、サーバーによって作成されます。擬似ファイルシステムを介して、クライアントはエクスポート済みファイルシステムの階層を参照します。つまり、クライアントから見る擬似ファイルシステムは、エクスポート済みファイルシステムに関連付けられたパスに制限されます。

以前のバージョンの NFS では、クライアントは、サーバーのファイルシステムを検索するには、各ファイルシステムをマウントする必要がありました。ただし、NFS Version 4 ではサーバー名前空間が次のことを行います。

- クライアントから見えるファイルシステムをサーバーエクスポートに関連付けられたディレクトリに制限します。
- サーバーエクスポートへのシームレスアクセスをクライアントに提供します (クライアントは配下の各ファイルシステムをマウントする必要がない)。ただし、オペレーティングシステムが異なるときは、クライアントが各サーバーファイルシステムをマウントしなければいけない場合があります。

図 2 NFS Version 4 でのサーバーファイルシステムおよびクライアントファイルシステムの見え方



図に示されている例では、クライアントは payroll ディレクトリと nfs4x ディレクトリを見ることができません (これらのディレクトリはエクスポート済みではなく、エクスポート済みディレクトリに関連付けられていないため)。ただし、local ディレクトリはエクスポート済みディレクトリであるため、local ディレクトリはクライアントが見ることができます。projects ディレクトリはエクスポート済みディレクトリ、nfs4 に関連付けられているため、projects ディレクトリはクライアントが見ることができます。つまり、サーバー名前空間のうち明示的にエクスポート済みでない部分は、擬似ファイルシステムでブリッジされ、これ経由でエクスポート済みディレクト

りおよびサーバーエクスポートに関連付けられたディレクトリだけを見ることになります。

NFS Version 4 における揮発性ファイルハンドル

ファイルハンドルは、NFS サーバー上で作成され、ファイルとディレクトリを一意に識別する情報を持ちます。NFS Version 2 と NFS Version 3 では、NFS サーバーが永続ファイルハンドルを返しました。したがって、NFS クライアントは、NFS サーバーが常に同じファイルを参照するファイルハンドルを生成することを保証できました。次に例を示します。

- ファイルが削除され同じ名前のファイルに置き換えられた場合、サーバーは必ず新しいファイルの新しいファイルハンドルを生成する。クライアントが古いファイルハンドルを使用していた場合、サーバーはファイルハンドルが無効であることを示すエラーを返す。
- ファイル名が変更されている場合、ファイルハンドルは変更されない。
- サーバーがリブートされた場合、ファイルハンドルは変更されない。

このように、サーバーがファイルハンドルを含むクライアントからの要求を受け取った場合、解決策は単純であり、ファイルハンドルは常に正しいファイルを参照します。

NFS 操作のために永続ファイルハンドルを使用してファイルとディレクトリを識別する方法は、ほとんどの UNIX ベースサーバーで機能しました。ただし、この方法は、ファイルのパス名などほかの識別方法を使用するサーバー上では実装できません。この問題を解決するために、サーバーは NFS Version 4 プロトコル経由でそのファイルハンドルが揮発性であることを宣言できます。ファイルハンドルが変更された場合、クライアントは新しいファイルハンドルを検出する必要があります。

NFS Versions 2 と NFS Version 3 サーバーと同様に、Oracle Solaris NFS Version 4 サーバーは常に永続ファイルハンドルを提供します。ただし、Oracle Solaris NFS Version 4 以外のサーバーにアクセスする Oracle Solaris NFS Version 4 クライアントは、サーバーが揮発性ファイルハンドルを使用する場合はそれらをサポートする必要があります。具体的には、サーバーがクライアントにファイルハンドルが揮発性であることを通知すると、クライアントはパス名とファイルハンドル間のマッピングをキャッシュする必要があります。クライアントは、期限切れになるまで、揮発性ファイルハンドルを使用します。期限が切れたとき、クライアントは次を実行します。

- そのファイルハンドルを参照するキャッシュされた情報をフラッシュする
- そのファイルの新しいファイルハンドルを検索する
- 操作を再試行する

注記 - サーバーは常にクライアントに対して、どのファイルハンドルが永続で、どのファイルハンドルが揮発性かを通知します。

揮発性ファイルハンドルはこれらの状況で期限切れになる可能性があります。

- ファイルを閉じたとき
- ファイルハンドルのファイルシステムが移行するとき
- クライアントがファイル名を変更するとき
- サーバーがリブートするとき

クライアントが新しいファイルハンドルを見つけられない場合、エラーメッセージログが `syslog` ファイルに記録されます。このファイルにさらにアクセスしようとする、入出力エラーで失敗します。

NFS Version 4 におけるクライアント回復

NFS Version 4 プロトコルはステートフルプロトコルです。NFS クライアントと NFS サーバーの両方が、オープンファイルとファイルロックに関する現在の情報を保守します。

サーバーがクラッシュしてリブートしたとき、サーバーの状態は消失します。クライアントは、サーバーがリブートしたことを検出すると、障害の前に存在していたオープンおよびロック状態をサーバーが再確立するのを支援するプロセスを開始します。このプロセスは、クライアントがプロセスを指示するため、クライアント回復と呼ばれています。

クライアントは、サーバーがリブートしたことを検出すると、ただちに現在の動作を停止して、クライアント回復のプロセスを開始します。回復プロセスが開始されると、次のようなメッセージが、システムエラーログ `/var/adm/messages` に表示されます。

```
NOTICE: Starting recovery server server-name
```

回復プロセスの間、クライアントは、クライアントの以前の状態に関するサーバー情報を送信します。ただし、この間、クライアントはサーバーに新しい要求を送信しません。ファイルを開いたりファイルロックを設定したりするための新しい要求は、サーバーが回復プロセスを完了するのを待ってから続行する必要があります。

クライアント回復プロセスが完了すると、次のメッセージがシステムエラーログ `/var/adm/messages` に表示されます。

```
NOTICE: Recovery done for server server-name
```

この時点で、クライアントはサーバーに状態情報を送信するのを正常に完了しました。ただし、クライアントがこのプロセスを完了しても、ほかのクライアントがそうでない可能性があります。つまり、猶予期間と呼ばれる期間は、サーバーはすべてのクライアントがそれぞれの回復を完了できるように、いかなるオープンおよびロック要求も受け入れません。

猶予期間中に、クライアントが新しいファイルを開こうとしたり、新しいロックを確立しようとしたりとすると、サーバーは **GRACE** エラーコードで要求を拒否します。このエラーを受け取ったクライアントは、猶予期間が終わるのを待ってから、要求をサーバーに再送信します。猶予期間中は、次のメッセージが表示されます。

```
NFS server recovering
```

猶予期間中も、ファイルを開いたりファイルロックを設定したりしないコマンドは処理できます。たとえば、コマンド `ls` と `cd` はファイルを開いたりファイルロックを設定したりせず、これらのコマンドは一時停止しません。ただし、ファイルを開く `cat` などのコマンドは、猶予期間が終わるまで中断されます。

猶予期間が終了すると、次のメッセージが表示されます。

```
NFS server recovery ok.
```

クライアントは、サーバーに新しいオープン要求またはロック要求を送信できるようになります。

クライアント回復は、さまざまな理由により失敗することがあります。たとえば、サーバーのリブート後にネットワークパーティションが存在する場合、クライアントは、猶予期間が終了する前にサーバーとの状態を再確立できない場合があります。猶予期間が終了すると、新しい状態操作により競合が発生する可能性があるため、サーバーはクライアントに状態の再確立を許可しません。たとえば、新しいファイルロックは、クライアントが回復しようとしている古いファイルロックと競合します。このような状況が発生すると、サーバーは **NO_GRACE** エラーコードをクライアントに返します。

ファイルに対するオープン操作の回復が失敗すると、クライアントはファイルを使用不可能としてマークし、次のメッセージが表示されます。

```
WARNING: The following NFS file could not be recovered and was marked dead  
(can't reopen: NFS status n): file : filename
```

回復に失敗している間にファイルロックを再確立すると、次のエラーメッセージが表示されます。

```
NOTICE: nfs4_send_siglost: pid process-ID lost  
lock on server server-name
```

この場合、**SIGLOST** シグナルがプロセスに送信されます。**SIGLOST** シグナルのデフォルトの動作は、プロセスを中断することです。

この状態から回復するには、障害発生時にファイルを開いていたアプリケーションを再起動する必要があります。ファイルを再度開くことができない一部のプロセスは入出力エラーを受け取る可能性があります。ファイルを再度開いたり、回復の失敗後にオープン操作を実行したりしたほかのプロセスは、問題なくファイルにアクセスできます。

このように、特定のファイルにアクセスできるプロセスとアクセスできないプロセスがあります。

NFS Version 4 における OPEN 共有サポート

NFS Version 4 プロトコルにはファイル共有モードがいくつか用意されていて、クライアントはほかのクライアントによるファイルアクセスを制御するために使用できます。クライアントは、次のように指定できます。

- DENY_NONE モードを指定すると、ほかのクライアントはファイルへの読み取りと書き込みアクセスを許可されます。
- DENY_READ モードを指定すると、ほかのクライアントはファイルへの読み取りアクセスを拒否されます。
- DENY_WRITE モードを指定すると、ほかのクライアントはファイルへの書き込みアクセスを拒否されます。
- DENY_BOTH モードを指定すると、ほかのクライアントはファイルへの読み取りと書き込みアクセスを拒否されます。

Oracle Solaris NFS Version 4 サーバーは、これらのファイル共有モードを完全に実装します。したがって、クライアントが現在の共有モードと矛盾する方法でファイルを開こうとすると、サーバーは操作を失敗させて、その試行を拒否します。このような試行がファイルのオープンまたは作成操作の開始で失敗すると、NFS Version 4 クライアントはプロトコルエラーを受け取ります。このエラーは、アプリケーションエラー EACCES にマップされます。

プロトコルはいくつかの共有モードを提供していますが、Oracle Solaris でのオープン操作は複数の共有モードを提供していません。ファイルを開くとき、Oracle Solaris NFS Version 4 クライアントは、DENY_NONE モードだけを使用できます。

注記 - `fcntl` システムコールには、ファイルの共有を制御する `F_SHARE` コマンドがありますが、`fcntl` コマンドは NFS Version 4 で正しく実装できません。NFS Version 4 クライアントでこれらの `fcntl` コマンドを使用すると、クライアントはアプリケーションに `EAGAIN` エラーを返します。

NFS Version 4 における委託

NFS Version 4 は、委託のためにクライアントサポートとサーバーサポートを提供します。委託は、サーバーがファイルの管理を NFS クライアントに委託するためのテクニックです。たとえば、サーバーは、読み取り委託または書き込み委託のいずれかをクライアントに付与できます。読み取り委託は互いに競合しないため、複数のクライアントに同時に付与できます。書き込み委託はほかのクライアントによるファイルアクセスと競合するため、1つのクライアントにだけ付与できます。書き込み委託を保持している間、クライアントは、ファイルへの排他的アクセスを保証されているために、さまざまな操作をサーバーに送信しません。同様に、読み取り委託を保持してい

る間、クライアントはさまざまな操作をサーバーに送信しません。クライアントが書き込みモードでファイルを開けないことをサーバーが保証するためです。

委託により、委託されたファイルに対するサーバーとクライアントの相互作用を大幅に減少することができます。したがって、ネットワークトラフィックが減少し、クライアントとサーバーのパフォーマンスが向上します。ただし、パフォーマンス向上の度合いは、アプリケーションが使用するファイル相互作用の種類およびネットワークとサーバー輻輳の量によって異なります。

クライアントは、委託を要求しません。委託を付与するかどうかの決定は、ファイルのアクセスパターンに基づいてサーバーがすべて行います。ファイルが最近複数の異なるクライアントから書き込みモードでアクセスされた場合、このアクセスパターンが将来競合する可能性を示しているためサーバーは委託を付与しないことがあります。

競合は、ファイルに付与されている委託と一致しない方法でクライアントがそのファイルにアクセスするときに発生します。たとえば、あるクライアントがファイルの書き込み委託を保持しており、2番目のクライアントが読み取りまたは書き込みアクセス用にそのファイルを開くとサーバーは最初のクライアントの書き込み委託を再呼び出します。同様に、あるクライアントが読み取り委託を保持しており、別のクライアントが書き込み用に同じファイルを開くと、サーバーは読み取り委託を再呼び出します。どちらの場合も、競合が現在存在しているため、2番目のクライアントは委託を付与されません。

競合が発生すると、NFS サーバーはコールバックメカニズムを使用して、委託を保持しているクライアントと連絡をとります。このコールバックを受信すると、クライアントはファイルの更新された状態をサーバーに送信し、委託を返します。クライアントが再呼び出しに対する応答に失敗すると、サーバーは委託を取り消します。こうした場合、サーバーはこのファイルに対するクライアントの操作をすべて拒否し、クライアントは要求された操作を失敗として報告します。一般的に、これらの失敗は入出力エラーとしてアプリケーションに報告されます。これらのエラーから回復するには、ファイルを閉じてから再度開く必要があります。取り消された委託による失敗は、クライアントが委託を保持している間にクライアントとサーバー間にネットワークパーティションが存在しているときに発生します。

サーバーは、別のサーバーに格納されているファイルに対するアクセス競合を解決できません。つまり、NFS サーバーは、格納しているファイルに対する競合だけを解決します。さらに、さまざまなバージョンの NFS を実行しているクライアントによって発生する競合に対して、NFS サーバーは NFS Version 4 を実行しているクライアントにだけ再呼び出しを開始できます。以前のバージョンの NFS を実行しているクライアントに再呼び出しを開始できません。

競合を検出するプロセスはさまざまです。たとえば、NFS Version 4 とは異なり、NFS Version 2 と NFS Version 3 にはオープン手順がないため、クライアントがファイルの読み取り、書き込み、またはロックを試行したあとでのみ、競合が検出されます。これらの競合に対するサーバーの応答もさまざまです。次に例を示します。

- NFS Version 3 では、サーバーは JUKEBOX エラーを返し、クライアントはアクセス要求を停止してあとで再試行します。クライアントは、File unavailable というメッセージを表示します。
- NFS Version 2 では、JUKEBOX エラーと同等のエラーが存在しないためサーバーは応答せず、クライアントは待機してから再試行します。クライアントは、NFS server not responding というメッセージを表示します。

エラーメッセージは、委託の競合が解決されたときにクリアされます。

デフォルトでは、NFS サーバー委託は有効になっています。server_delegation パラメータを off に設定することで、委託を無効にできます。

```
# sharectl set -p server_delegation=off nfs
```

NFS クライアントの委託にキーワードは必要ありません。NFS Version 4 コールバックデーモン nfs4cbdd により、クライアント上のコールバックサービスが提供されます。このデーモンは、NFS Version 4 のマウントが有効になると自動的に起動されます。デフォルトで、クライアントは、/etc/netconfig システムファイルに一覧表示されているすべてのインターネット転送に必要なコールバック情報を提供します。クライアントで IPv6 が有効であり、クライアントの名前の IPv6 アドレスを決定できる場合、コールバックデーモンは IPv6 接続を受け入れます。

コールバックデーモンは、一時的なプログラム番号と動的に割り当てられたポート番号を使用します。この情報は、サーバーに提供され、サーバーは委託を付与する前にコールバックパスをテストします。コールバックパスが正常にテストされない場合、サーバーは委託を付与しません。外部から見ることでできる動作だけになります。

コールバック情報は NFS Version 4 要求に埋め込まれているため、サーバーは、NAT (Network Address Translation) を使用するネットワークデバイスを通してクライアントに連絡できません。また、コールバックデーモンは、動的ポート番号も使用します。したがって、ファイアウォールがポート 2049 上で通常の NFS トラフィックを有効にしている場合でも、サーバーがファイアウォールを検索できない場合があります。この場合、サーバーは委託を付与しません。

NFS Version 4 での ACL と nfsmapid

アクセス制御リスト (ACL) は、ファイルの所有者が、ファイル所有者、グループ、ほかの固有のユーザーおよびグループに関するファイルアクセス権を定義できるようにすることで、ファイルのセキュリティを提供します。ZFS ファイルシステムでは、chmod コマンドを使用することで、ACL を NFS サーバーおよび NFS クライアント上で設定できます。UFS ファイルシステムの場合は、setfacl コマンドを使用できます。詳細は、[chmod\(1\)](#) および [setfacl\(1\)](#) のマニュアルページを参照してください。NFS Version 4 では、ID マッパー nfsmapid を使用して、サーバー上の ACL エントリ内のユーザー ID またはグループ ID を、クライアント上の ACL エントリ内のユーザー ID またはグループ ID にマッピングします。逆も同じです。ACL エントリの

ユーザーおよびグループ ID は、クライアントとサーバーの両方に存在する必要があります。

ACL および `nfsmapid` の詳細は、次を参照してください。

- 『Oracle Solaris 11.3 での ZFS ファイルシステムの管理』の第 9 章、「ACL および属性を使用した Oracle Solaris ZFS ファイルの保護」
- 167 ページの「NFS デーモン」

ID マッピングの問題

次の状態は、ID マッピングが失敗する原因になる可能性があります。

- サーバー上の ACL エントリ内に存在するユーザーまたはグループをクライアント上の有効なユーザーまたはグループにマッピングできない場合、ユーザーは ACL を読み取ることはできますが、ユーザーやグループの一部が `unknown` と表示されます。
たとえば、この状況で `ls -lv` や `ls -lV` コマンドを発行した場合、一部の ACL エントリでグループやユーザーが `unknown` と表示されます。
- クライアント上で設定されている ACL エントリ内のユーザーまたはグループ ID をサーバー上の有効なユーザーまたはグループ ID にマッピングできない場合、`setfacl` や `chmod` コマンドが失敗し、`Permission denied` エラーメッセージを返す可能性があります。
- クライアントとサーバーで `nfsmapid_domain` の値が一致しない場合、ID マッピングは失敗します。詳細は、167 ページの「NFS デーモン」を参照してください。

ID マッピングの問題を回避するには、次を行います。

- `nfsmapid_domain` の値が正しく設定されていることを確認します。現在選択されている NFSv4 ドメインは、`/var/run/nfs4_domain` ファイル内で入手できます。
- ACL エントリ内のすべてのユーザー ID およびグループ ID が NFS Version 4 のクライアントとサーバーの両方に存在することを確認します。

マッピングされていないユーザー ID またはグループ ID を確認する

NFS サーバーまたはクライアント上でユーザーまたはグループをマッピングできないかどうかを判別するには、次のスクリプトを使用します。

```
#!/usr/sbin/dtrace -Fs
sdt:::nfs4-acl-nobody
{
    printf("validate_idmapping: (%s) in the ACL could not be mapped!",
```

```
stringof(arg0));  
}
```

注記 - このスクリプトで使用されているプローブ名は、将来変更される可能性があるインタフェースです。詳細については、『[Oracle Solaris 11.3 DTrace \(Dynamic Tracing\) Guide](#)』の「[Stability Levels](#)」を参照してください。

UDP と TCP のネゴシエーション

NFS Version 2 と Version 3 では、トランスポートプロトコルのネゴシエーションはマウント時に実行されます。開始時には、トランスポートプロトコルもネゴシエートされます。デフォルトでは、NFS クライアントと NFS サーバーの両方がサポートしているコネクション型トランスポートの中で最初に見つかったものが選択されます。それが見つからない場合は、コネクションレス型トランスポートプロトコルの中で最初に見つかったものが使用されます。システムでサポートされているトランスポートプロトコルは、`/etc/netconfig` ファイルにリストされています。TCP はコネクション型トランスポートプロトコルで、Solaris 2.6 からサポートされています。UDP はコネクションレス型トランスポートプロトコルです。

NFS プロトコルのバージョンとトランスポートプロトコルが両方ともネゴシエーションによって決まった場合は、NFS プロトコルのバージョンがトランスポートプロトコルよりも優先されます。UDP を使用する NFS Version 3 プロトコルの方が、TCP を使用する NFS Version 2 プロトコルよりも優先されます。`mount` コマンドでは NFS プロトコルのバージョンもトランスポートプロトコルも手動で選択できます。`mount` コマンドの NFS 固有オプションについては、[mount_nfs\(1M\)](#) のマニュアルページを参照してください。ほとんどの場合、ネゴシエーションによって選択されるオプションの方が適切です。

ファイル転送サイズのネゴシエーション

ファイル転送サイズは、NFS クライアントと NFS サーバーの間でデータを転送するときに使用されるバッファのサイズを確立します。原則として、ファイル転送サイズが大きいほどパフォーマンスが向上します。NFS Version 3 プロトコルには転送サイズに上限はありません。クライアントはマウント時に小さい転送サイズを指定できますが、ほとんどの場合この指定は必要ありません。

転送サイズは、NFS Version 2 を使用しているシステムとはネゴシエートされません。ファイル転送サイズの上限は 8K バイトに設定されます。

`mount` コマンドに対して `-rsize` オプションと `-wsize` オプションを使用すると、転送サイズを手動で設定できます。システムクライアントの一部では転送サイズを小さく

しなければいけない場合があります。また、NFS サーバーが大きなファイル転送サイズに構成されている場合は、転送サイズを大きくすることができます。

注記 - Oracle Solaris 10 以降のリリースでは、書き込み転送サイズの制限が緩和されました。使用するトランスポートプロトコルに基づいて転送サイズが決定されるようになりました。たとえば、UDP 使用時の NFS 転送の上限は、以前と同じく 32K バイトです。これに対し、TCP は UDP のようなデータグラム制限を持たないストリーミングプロトコルであるため、TCP 使用時の最大転送サイズは、1M バイトまで拡張されています。

NFS Version 3 でのファイルシステムのマウント方法

このセクションの情報は NFS Version 3 マウントだけに適用されます。NFS Version 4 マウントプロセスは、ポートマップサービスおよび MOUNT プロトコルを取り込みません。

NFS クライアントが NFS サーバーからファイルシステムのマウントを試行するとき、クライアントはサーバーからファイルハンドルを取得する必要があります。ファイルハンドルは、そのファイルシステムに対応していなければなりません。そのためには、クライアントとサーバーの間でいくつかのトランザクションが発生します。この例では、NFS クライアントは NFS サーバーから /home/user をマウントしようとしています。このトランザクションの snoop トレースは次のとおりです。

```
client -> server PORTMAP C GETPORT prog=100005 (MOUNT) vers=3 proto=UDP
server -> client PORTMAP R GETPORT port=33482
client -> server MOUNT3 C Null
server -> client MOUNT3 R Null
client -> server MOUNT3 C Mount /export/home9/user
server -> client MOUNT3 R Mount OK FH=9000 Auth=unix
client -> server PORTMAP C GETPORT prog=100003 (NFS) vers=3 proto=TCP
server -> client PORTMAP R GETPORT port=2049
client -> server NFS C NULL3
server -> client NFS R NULL3
client -> server NFS C FSINFO3 FH=9000
server -> client NFS R FSINFO3 OK
client -> server NFS C GETATTR3 FH=9000
server -> client NFS R GETATTR3 OK
```

この追跡結果では、NFS クライアントがまずマウントポート番号を NFS サーバーの portmap サービスに要求します。クライアントが取得したマウントポート番号 (33492) は、サーバーでサービスが使用可能かどうかをテストするために使用されます。このポート番号でサービスが実行中であることが確認できると、クライアントはマウントを要求します。サーバーはこの要求に応答するときに、マウントするファイルシステムのファイルハンドル (9000) を取り込みます。これに対してクライアントは、NFS ポート番号を要求します。クライアントはサーバーからこの番号を受け取ると、NFS サービス (nfsd) が使用可能かどうかをテストします。また、そのファイルハンドルを使うファイルシステムに関する NFS 情報を要求します。

次のトレースでは、クライアントは `public` オプション付きファイルシステムをマウントしています。

```
client -> server NFS C LOOKUP3 FH=0000 /export/home9/user
server -> client NFS R LOOKUP3 OK FH=9000
client -> server NFS C FSINFO3 FH=9000
server -> client NFS R FSINFO3 OK
client -> server NFS C GETATTR3 FH=9000
server -> client NFS R GETATTR3 OK
```

デフォルトの公開ファイルハンドル (`0000`) を使用しているために、すべてのトランザクションにポートマップサービスから情報が与えられ、NFS ポート番号を決定するためのトランザクションはありません。

注記 - NFS Version 4 は、揮発性ファイルハンドルをサポートします。詳細は、[32 ページの「NFS Version 4 における揮発性ファイルハンドル」](#)を参照してください。

マウント時の `-public` オプションと NFS URL の意味

`-public` オプションを使用すると、マウントが失敗することがあります。NFS URL を追加することも失敗の原因になる可能性があります。次のリストで、これらのオプションを使用した場合にファイルシステムがどのようにマウントされるかを説明します。

- **public オプションと NFS URL** – 公開ファイルハンドルを使用します。公開ファイルハンドルがサポートされていないと、マウントは失敗します。
- **public オプションと通常のパス** – 公開ファイルハンドルを使用します。公開ファイルハンドルがサポートされていないと、マウントは失敗します。
- **NFS URL のみ** – NFS サーバーでサポートされていれば、公開ファイルハンドルを使用します。public ファイルハンドルを使用したときにマウントが失敗する場合は、MOUNT プロトコルを使ってマウントしてみてください。
- **通常のパスのみ** – 公開ファイルハンドルは使用しないでください。MOUNT プロトコルが使用されます。

クライアント側フェイルオーバー

フェイルオーバーは、複製されたファイルシステムをサポートしているサーバーのリストから、1つのサーバーを選択するプロセスです。通常は、ソートされたリスト内の次のサーバーが使用されます (応答に失敗しないかぎり)。クライアント側フェイルオーバーを使用することで、NFS クライアントは複数のサーバーが同じデータを利用できるようにしているときを検出でき、現在のサーバーが使用不能になったときに代替サーバーに切り替えることができます。この切り替えは再マッピングと呼ばれます。

す。正常使用のときは、クライアントはリモートファイルシステム上に各アクティブファイルのパス名を格納します。再マッピング時には、これらのパス名が評価されて新しいサーバー上のファイルが検出されます。

次のいずれかの状態が発生すると、ファイルシステムが使用不能になる可能性があります。

- ファイルシステムが、クラッシュしているサーバーに接続している
- サーバーの過負荷
- ネットワーク障害

これらの状況でのフェイルオーバーは通常、ユーザーに透過的です。クライアント上で実行中のプロセスを中断することなく、任意のときに実行される可能性があります。

フェイルオーバーを実行するには、ファイルシステムが読み取り専用でマウントされている必要があります。また、ファイルシステムが完全に同じでないとフェイルオーバー機能は成功しません。ファイルシステムが同じであることについては、[42 ページの「複製されたファイルシステムとは」](#)を参照してください。フェイルオーバー機能の候補としては、静的なファイルシステム、または変更の少ないファイルシステムが適しています。

同じ NFS マウント上では、CacheFS 機能とクライアント側のフェイルオーバー機能の両方は使用できません。CacheFS ファイルシステムは、それぞれについて追加情報が格納されています。この情報はフェイルオーバーの際に更新できないため、ファイルシステムをマウントするときにはこれら 2 つの機能の一方のみを使用できます。

各ファイルシステムで確立する必要がある複製の数は、さまざまな要素によって変わります。理想的には、2 つ以上のサーバーを持つことをお勧めします。それぞれのサーバーが複数のサブネットをサポートする必要があります。これは、各サブネットに一意のサーバーを設置するよりもよい方法です。このプロセスではリストされているすべてのサーバーを確認する必要があります。そのため、サーバーの台数を増やすと、それぞれのマウント処理が遅くなります。

複製されたファイルシステムとは

クライアント側フェイルオーバーが目的の場合は、元のファイルシステムとサイズが同じで、ファイルサイズまたはファイルタイプが同じときは、そのファイルシステムを「複製」と呼ぶことができます。アクセス権、作成日付などのファイル属性は関係ありません。ファイルサイズまたはファイルタイプが異なる場合は再マッピングは失敗し、古いサーバーが使用可能になるまでプロセスはハングアップします。NFS Version 4 では、この動作は異なります。クライアント側フェイルオーバーの詳細については、[43 ページの「NFS Version 4 におけるクライアント側フェイルオーバー機能」](#)を参照してください。

rsync、cpio、または別のファイル転送メカニズムを使用することで、複製されたファイルシステムを保守できます。複製されたファイルシステムを更新すると不一致が発生するため、最良の結果を得るには次の予防策を考慮してください。

- 新しいバージョンのファイルをインストールする前に、古いバージョンのファイル名を変更する。
- クライアント使用率が低い夜間に更新を実行する。
- 更新は小規模にとどめる。
- ファイルのコピーの数を最小限に抑える。

フェイルオーバー機能と NFS ロック

ソフトウェアパッケージの一部は、ファイルに読み取りロックをかける必要があります。そのようなソフトウェアが正常に動作できるようにするため、読み取り専用ファイルシステムに対しても読み取りロックがかけられるようになっています。ただし、これはクライアント側でしか認識されません。サーバーがロックを検出できないため、再マッピング時もロックは永続します。ファイルはもともと変更が許されないのので、サーバー側でファイルをロックする必要はありません。

NFS Version 4 におけるクライアント側フェイルオーバー機能

NFS Version 4 では、ファイルサイズが違うまたはファイルタイプが同じでないために複製を確立できない場合、次のことが起こります。

1. ファイルが使用不能とマークされる。
2. 警告が表示される。
3. 複製されたマウントのファイルを使用するアプリケーションがシステムコール失敗を受け取る。

注記 - アプリケーションを再起動して、ファイルに再度アクセスすると、正常にアクセスできます。

NFS Version 4 では、ディレクトリのサイズが異なってもレプリケーションエラーを受け取ることはありません。以前のバージョンの NFS では、この状態はエラーとして扱われ、再マッピングプロセスを妨げました。

さらに、NFS Version 4 では、ディレクトリ読み取り操作が不成功の場合、次にリストされたサーバーによって操作が行われます。以前のバージョンの NFS では、読み取り操作が不成功のときは、再マッピングが失敗し、プロセスは元のサーバーが使用可能になるまでハングアップします。

NFS サーバーログ機能のしくみ

注記 - サーバーロギングは NFS Version 4 ではサポートされません。

NFS サーバーロギングは、NFS の読み取りと書き込み、およびこのファイルシステムを変更する操作の記録を提供します。これらのレコードは情報へのアクセスをトレースするのに利用できます。さらに、この記録は、情報へのアクセスを測定する定量的な方法を提供します。

ログ機能が有効になっているファイルシステムにアクセスすると、カーネルが raw データをバッファファイルに書き込みます。このデータには、次の内容が含まれています。

- タイムスタンプ
- クライアント IP アドレス
- 要求者の UID
- アクセスされているファイルまたはディレクトリオブジェクトのファイルハンドル
- 発生した操作のタイプ

nfslogd デモンはこの raw データを、ログファイルに保存される ASCII レコードに変換します。使用可能なネームサービス機能が一致しているものを見つけると、その変換中に IP アドレスはホスト名に変更され、UID はログインに変更されます。ファイルハンドルはパス名にも変換されます。デモンはファイルハンドルを追跡し、情報を別のファイルハンドルパステブルに保存して、変換を完了します。このようにすると、ファイルハンドルにアクセスされるたびに、パスを識別し直す必要がなくなります。nfslogd が無効になっている場合はファイルハンドル - パステブル内のマッピングへの変更が行われなため、このデモンは実行したままにする必要があります。

WebNFS サービスのしくみ

WebNFS サービスとは、あるディレクトリに置かれたファイルを、公開ファイルハンドルを使ってクライアントからアクセスできるようにするものです。ファイルハンドルは、NFS クライアントがファイルを識別できるようにカーネルが生成するアドレスです。公開ファイルハンドルの値はあらかじめ定義されているため、サーバーがクライアントに対してファイルハンドルを生成する必要はありません。定義済みのファイルハンドルを使用するというこの機能によって、MOUNT プロトコルが不要になってネットワークトラフィックが減り、クライアントにとってはプロセスが高速化します。

デフォルトでは、NFS サーバーの公開ファイルハンドルはルートファイルシステムに対して設定されます。このデフォルトのため、サーバーに対してマウント権限を持つ

ているすべてのクライアントに対して WebNFS アクセス権が与えられます。公開ファイルハンドルは、share コマンドによって任意のファイルシステムに切り替えることができます。

ファイルシステムに対するファイルハンドルを NFS クライアントが持っているとき、アクセスするファイルに対応するファイルハンドルを判断するために LOOKUP が実行されます。NFS プロトコルでは、パス名のコンポーネントを 1 度に 1 つしか評価できません。したがって、ディレクトリ階層のレベルが 1 つ増えるたびに 1 回ずつ LOOKUP を実行します。LOOKUP が公開ファイルハンドルへの相対パスのときは、WebNFS サーバーは単一マルチコンポーネントルックアップトランザクションでパス名全体を評価できます。WebNFS サーバーはマルチコンポーネントルックアップにより、パス名内でディレクトリレベルごとにファイルハンドルを交換せずに、目的のファイルにファイルハンドルを配布できます。

また、NFS クライアントは、単一の TCP 接続を介して、複数のファイルを同時にダウンロードすることができます。このようにして接続すると、サーバーに複数の接続を設定することによる負荷をかけることなく、すばやくアクセスすることができます。Web ブラウザアプリケーションも複数ファイルを同時にダウンロードできますが、それぞれのファイルに独自の接続が確立されます。WebNFS ソフトウェアは接続を 1 つしか使用しないため、サーバーに対するオーバーヘッドを軽減できます。

パス名の中の最後のコンポーネントが他のファイルシステムに対するシンボリックリンクである場合、通常の NFS アクティビティによってあらかじめそのファイルへのアクセス権を持っている場合、クライアントはそのファイルにアクセスできます。

通常、NFS URL は公開ファイルハンドルからの相対位置として評価されます。評価をサーバーのルートファイルシステムへの相対パスに変更するには、パスの先頭にスラッシュを追加してください。次の 2 つの NFS URL は、公開ファイルハンドルが /export/ftp ファイルシステムで確立されている場合は同等です。

```
nfs://server/junk
nfs://server//export/ftp/junk
```

注記 - NFS Version 4 プロトコルは、WebNFS サービスに優先します。NFS Version 4 は、MOUNT プロトコルと WebNFS サービスに追加されたすべてのセキュリティーネゴシエーションを完全に統合します。

WebNFS セキュリティーネゴシエーション機能のしくみ

NFS サービスには、WebNFS クライアントが WebNFS サーバーと、選択されたセキュリティーメカニズムについてネゴシエーションできるようにするプロトコルが含まれています。この新しいプロトコルは、以前のバージョンの WebNFS プロトコルで使用されていたマルチコンポーネントルックアップの拡張である、セキュリティーネゴシエーションマルチコンポーネントルックアップを使用します。

WebNFS クライアントは、公開ファイルハンドルを使って通常のマルチコンポーネントルックアップ要求を行うことにより、このプロセスを開始します。このクライアントには、サーバーがどのようにしてこのパスを保護しているかについての知識がないため、デフォルトのセキュリティーメカニズムが使用されます。デフォルトセキュリティーメカニズムでは不十分な場合は、サーバーは `AUTH_TOOWEAK` エラーを返します。クライアントはより強力なメカニズムを使用する必要があることを意味しています。

クライアントは、`AUTH_TOOWEAK` エラーを受信すると、サーバーに対してどのセキュリティーメカニズムが必要か決定するように要求します。この要求が成功すると、サーバーは、指定されたパスに必要なセキュリティーメカニズムの配列を返します。このセキュリティーメカニズムの配列のサイズによっては、クライアントは完全な配列を得るためにさらに要求を出さなければならない場合があります。サーバーが WebNFS セキュリティーネゴシエーションをサポートしていない場合は、この要求は失敗します。

要求が成功すると、WebNFS クライアントは、クライアントがサポートしている最初のセキュリティーメカニズムを配列から選択します。クライアントはその後、選択したセキュリティーメカニズムを使用して、通常のマルチコンポーネントルックアップ要求を発行し、ファイルハンドルを獲得します。この後に続くすべての NFS 要求は、選択されたセキュリティーメカニズムとファイルハンドルを使って出されます。

Web ブラウザの使用と比較した場合の WebNFS の制約

HTTP を使用する Web サイトが提供できる機能のいくつかは、WebNFS ソフトウェアではサポートされていません。この違いは NFS サーバーがファイルを送信するだけであるという事実由来のため、クライアント側で特別な処理を行う必要があります。ある Web サイトを WebNFS と HTTP 両方のアクセスに対応させるには、次を考慮してください。

- NFS によるブラウズでは CGI スクリプトは実行されません。したがって、CGI スクリプトを多用している Web サイトを含むファイルシステムは、NFS によるブラウズに適していない可能性があります。
- 新しい NFS URL から異なるファイル形式のこれらのファイルにアクセスすると、ファイル名からファイルタイプが判別可能な場合には外部のビューアが起動されます。WebNFS ソフトウェアはファイル内部をチェックしてファイルタイプを判別することはしないため、ファイルタイプを判別する唯一の方法はファイル名拡張子です。ブラウザは、標準 MIME タイプについてはファイル名拡張子を認識するはずですが。
- NFS ブラウズではサーバー側イメージマップを使用できませんが、クライアント側イメージマップは使用できます (URL が場所で定義されているため)。ドキュメントサーバーからの応答は不要です。

Secure NFS システム

NFS 環境は、アーキテクチャーやオペレーティングシステムの異なるコンピュータから構成されるネットワーク上でファイルシステムを共有するための強力で使いやすい手段です。しかし、NFS の操作によるファイルシステムの共有を便利にする機能が、一方ではセキュリティ上の問題につながっています。今まで、NFS はほとんどのバージョンで UNIX (AUTH_SYS) 認証を使用してきましたが、現在では AUTH_DH のようなより強力な認証方式も使用可能です。UNIX 認証を使用している場合、NFS サーバーはユーザーではなくファイル要求を行ったコンピュータを認証することで要求を認証します。したがって、NFS クライアントユーザーはスーパーユーザーになるために `su` を実行し、ファイルの所有者を偽装できます。DH 認証では、NFS サーバーはユーザーを認証するため、このような操作が困難になります。

ルートアクセス権とネットワークプログラミングについての知識があれば、だれでも任意のデータをネットワークに取り入れたり、任意のデータをネットワークから取り出したりできます。もっとも危険な攻撃にはデータの持ち込みが伴います。たとえば、有効なパケットを生成したり、または「対話」を記録し後で再生することによってユーザーを装うなどの手段があります。これらはデータの整合性に影響を与えません。ユーザーを装わず、単にネットワークトラフィックを傍受するための受動的な盗聴が行われる攻撃であれば、データの整合性が損なわれることはないため、それほど危険ではありません。ネットワーク上でやりとりされるデータを暗号化すると、機密情報のプライバシーを保護できます。

ネットワークのセキュリティ問題に対する共通の対処方法は、解決策を各アプリケーションにゆだねることです。さらに優れた手法としては、すべてのアプリケーションを対象として、標準の認証システムを導入することです。

Oracle Solaris オペレーティングシステムには、NFS の操作が構築されるメカニズムである RPC のレベルで、認証システムが組み込まれています。このシステムは Secure RPC と呼ばれ、ネットワーク環境のセキュリティを大幅に向上させるとともに、NFS のセキュリティを強化します。Secure RPC が提供する機能を利用した NFS システムを Secure NFS システムといいます。

Secure RPC

Secure RPC は Secure NFS システムの基盤です。Secure RPC の目標は、少なくともタイムシェアリングシステム程度に安全なシステムを構築することです。タイムシェアリングシステムでは、すべてのユーザーが1つのコンピュータを共有し、ユーザーはログインパスワードによって認証されます。データ暗号化規格 (DES) 認証でも、同じ認証処理が実行されます。ユーザーは、ローカル端末の場合と同じように、任意のリモートコンピュータにログインできます。ユーザーのログインパスワードは、ネットワークセキュリティへの保証です。タイムシェアリングでは、システム管理者は信頼のおける人で、パスワードを変更してだれかを装うようなことはしないという道徳

上の義務を負います。Secure RPC では、ネットワーク管理者は「公開鍵」を格納するデータベースのエントリを変更しないという前提で信頼されています。

RPC 認証システムは、資格とベリファイアを使用します。ID バッジを例にとれば、資格とは、名前、住所、誕生日など個人を識別するものです。ベリファイアとはバッジに添付された写真です。バッジの写真をその所持者と照合することによって、そのバッジが盗まれたものではないことを確認できます。RPC では、クライアントプロセスは RPC 要求のたびに資格とベリファイアの両方をサーバーに送信します。クライアントはサーバーの資格をすでに知っているため、サーバーはベリファイアだけを送り返します。

RPC の認証は拡張が可能で、UNIX、DH、および KERB などのさまざまな認証システムを組み込むことができます。

ネットワークサービスで UNIX 認証を使用する場合、資格にはクライアントのホスト名、UID、GID、グループアクセスリストが含まれます。ただし、ベリファイアが存在しないため、スーパーユーザーは su などのコマンドを使用して、適切な資格を偽ることができます。もう 1 つの問題は、ネットワーク上のすべてのコンピュータを UNIX コンピュータと想定していることです。UNIX 認証を異機種ネットワーク内の他のオペレーティングシステムに適用した場合、これは正常に動作しません。

UNIX 認証の問題を克服するために、Secure RPC では DH 認証を使用します。

注記 - Kerberos 認証システムのサポートは Secure RPC の一部としてはもう提供されていませんが、このリリースにはサーバー側とクライアント側の実装が含まれています。Kerberos 認証の実装に関する詳細は、『[Oracle Solaris 11.3 での Kerberos およびその他の認証サービスの管理](#)』の第 2 章、「[Kerberos サービスについて](#)」を参照してください。

DH 認証

DH 認証は、Data Encryption Standard (DES) と Diffie-Hellman 公開鍵暗号手法を使ってネットワーク上のユーザーとコンピュータの両方を認証します。DES は、標準の暗号化メカニズムです。Diffie-Hellman 公開鍵暗号手法は、2 つの鍵、つまり公開鍵と秘密鍵を持つ暗号方式です。公開鍵と秘密鍵は名前空間に格納されます。NIS では、これらの鍵は public-key マップに保存されています。これらのマップにはすべての認証の候補ユーザーの公開鍵と秘密鍵が入っています。マップの設定方法に関する詳細は、『[Oracle Solaris 11.3 ディレクトリサービスとネームサービスでの作業: DNS と NIS](#)』を参照してください。

DH 認証のセキュリティは、送信側が現在時間を暗号化する機能に基づいていて、受信側はこれを復号化して、自分の時間と照合します。タイムスタンプは DES を使用して暗号化されます。2 つのエージェントは現在の時間で同意し、送信側と受信側が同じ暗号化鍵を使用する必要があります。

ネットワークが時間同期プログラムを実行する場合、NFS クライアントと NFS サーバー上の時間は自動的に同期がとられます。時間同期プログラムを使用できない場合、ネットワーク時間ではなく、サーバーの時間を使ってタイムスタンプを計算できます。クライアントは、RPC セッションを開始する前にサーバーに時間を要求し、自分のクロックとサーバーのクロックとの時間差を計算します。タイムスタンプを計算するときには、この差を使ってクライアントのクロックを補正します。クライアントとサーバーのクロックが同期しなくなると、サーバーはクライアントの要求を拒否し始めます。その場合、クライアントの DH 認証システムはサーバーとの間で再び同期をとります。

クライアントとサーバーは、ランダムな対話鍵(セッション鍵とも呼ばれる)を生成したあと公開鍵暗号方式を使って共通鍵を類推することによって、同一の暗号化鍵に到達します。この共通鍵は、クライアントとサーバーだけが推理できる鍵です。対話鍵は、クライアントのタイムスタンプを暗号化および復号化するために使用されます。共通鍵は、この対話鍵を暗号化および復号化するために使用されます。

NFS での Secure RPC の使用

Secure RPC を使用する予定の場合は、次の点を考慮してください。

- NFS サーバーがクラッシュしたときにシステム管理者がいない場合(停電のあとなど)には、システムに格納されていた秘密鍵がすべて削除されます。どのプロセスからも、セキュアなネットワークサービスにアクセスしたり NFS ファイルシステムをマウントしたりできません。リブート中の重要な処理は、通常 root として実行されます。そのため、root の秘密鍵を別に保存していればこれらのプロセスを実行できますが、その秘密鍵を復号化するパスワードを入力することはできません。keylogin -r を使用すると root の秘密鍵がそのまま /etc/.rootkey に格納され、keyserv がそれを読み取ります。
- システムによっては、シングルユーザーモードでブートし、コンソールには root のログインシェルが表示されてパスワードの入力が要求されないことがあります。このような場合は、物理的なセキュリティが不可欠です。
- ディスクレスコンピュータのブートは、完全に安全とはいえません。ブートサーバーになりすましてリモートコンピュータで秘密鍵を記録するような、不正なカーネルをだれかがブートする可能性があります。Secure NFS システムによって保護されているのはカーネルと鍵サーバーが起動した後だけです。そうでないと、ブートサーバーからの応答を認証することができません。この制限は重大な問題につながる可能性があります。この制限を攻撃するにはカーネルのソースコードを使用した高度な技術が必要です。また、不法行為の痕跡が残ります。つまり、ネットワークを通じてブートサーバーにポーリングすれば、不正なブートサーバーの場所がわかります。
- 多くの setuid プログラムは root が所有者です。root の秘密鍵が /etc/.rootkey に格納されていれば、これらのプログラムは正常に動作します。しかし、ユーザーが所有者である setuid プログラムは動作しない可能性があります。たとえば、ある setuid プログラムの所有者が dave であり、ブート後 dave が 1 度もログインしてい

ないとします。このプログラムはセキュアなネットワークサービスにはアクセスできません。

- リモートコンピュータに (`login`、`rlogin`、または `telnet` を使用して) ログインし、`keylogin` を使ってアクセスすると、自分のアカウントへのアクセスを提供することになります。秘密鍵がそのコンピュータの鍵サーバーに渡され、その秘密鍵が格納されます。このプロセスが問題になるのは、相手側のリモートコンピュータを信用できない場合だけです。しかし、疑いがある場合は、パスワードを要求するリモートコンピュータにはログインしないでください。代わりに NFS 環境を使用して、そのリモートコンピュータから共有されているファイルシステムをマウントします。または、`keylogout` を使って鍵サーバーから秘密鍵を消去します。
- ホームディレクトリが `-o sec=dh` オプションにより共有されていると、リモートログインによって問題が生じる可能性があります。`/etc/hosts.equiv` ファイルまたは `~/.rhosts` ファイルに、パスワードを要求するように設定されていない場合は、ログインが成功します。ただし、ローカルで認証されていないため、ユーザーは自分のホームディレクトリにアクセスできません。ユーザーがパスワードを要求される場合には、パスワードがネットワークパスワードと一致すれば自分のホームディレクトリにアクセスできます。

ミラーマウントのしくみ

Oracle Solaris 11 リリースには、ミラーマウントと呼ばれる新しいマウント機能が含まれています。ファイルシステムが NFS Version 4 サーバーで共有されると、ミラーマウントによって NFS Version 4 クライアントは、ただちにそのファイルシステムのファイルにアクセスできます。`mount` コマンドを使用したり `autofs` マップを更新したりするオーバーヘッドなしで、ファイルにアクセスできます。実際、NFS Version 4 ファイルシステムがクライアントにマウントされたあとは、そのサーバーからほかのファイルシステムもマウントできます。

一般に、次のことを行う必要がある場合を除き、NFS Version 4 クライアントではミラーマウント機能を使用することをお勧めします。

- サーバー上に存在する階層とは異なる階層をクライアントで使用する
- 親ファイルシステムとは異なるマウントオプションを使用する

ミラーマウントを使用してファイルシステムをマウントする

手動マウントまたは `autofs` を使用して NFS Version 4 クライアントにファイルシステムをマウントする場合、マウントされるファイルシステムに追加される追加ファイルシ

システムをミラーマウント機能でクライアントにマウントできます。クライアントは、親ディレクトリに使用したものと同一マウントオプションを使用して、新しいファイルシステムへのアクセスを要求します。なんらかの理由でマウントが失敗すると、サーバーとクライアントの間で通常の NFS Version 4 セキュリティーネゴシエーションが実行され、マウント要求が成功するようにマウントオプションが調整されます。

特定のサーバーファイルシステムのための自動マウントトリガーが存在する場合は、ミラーマウントよりも自動マウントトリガーが優先されるため、そのファイルシステムに対してミラーマウントは実行されません。この場合にミラーマウントを使用するには、自動マウントエントリを削除する必要があります。

Oracle Solaris 11 リリースでは、`/net` または `/home` 自動マウントポイントにアクセスすると、サーバー名前空間 `/net` または `/home` がマウントされます。これらのディレクトリの下にあるディレクトリやファイルに対するアクセスは、ミラーマウント機能によって提供されます。

ミラーマウントの使用方法の詳細については、[81 ページの「サーバーからすべてのファイルシステムをマウントする方法」](#)を参照してください。

ミラーマウントを使用してファイルシステムをアンマウントする

ミラーマウントされたファイルシステムは、それらが一定期間非アクティブ後のアイドル状態である場合、自動的にアンマウントされます。この期間は `timeout` パラメータ (オートマウントによって同じ目的に使用される) を使用することで設定されます。

NFS ファイルシステムを手動でアンマウントする場合、ミラーマウントされたファイルシステムが含まれているときは、アイドル状態であればそれらもアンマウントされます。ミラーマウントされたファイルシステムがアクティブな場合は、元のファイルシステムがビジー状態であるかのように、手動アンマウントは失敗します。ただし、強制アンマウントは、そこに含まれているミラーマウントされたファイルシステムすべてに伝播されます。

自動マウントされるファイルシステムの中にファイルシステム境界が見つかった場合、ミラーマウントが実行されます。オートマウントが親ファイルシステムをアンマウントする場合、ミラーマウントされたファイルシステムが含まれているときは、アイドル状態であればそれらも自動的にアンマウントされます。ミラーマウントされたアクティブなファイルシステムがある場合、自動アンマウントは実行されず、現在の自動マウント動作が維持されます。

NFS リフェラルのしくみ

Oracle Solaris 11.1 リリース以降では、NFS リフェラルによって、NFS Version 4 サーバーは、複数の NFS Version 4 サーバーを均一な名前空間に接続する手段として、ほかの NFS Version 4 サーバーにあるファイルシステムを参照できます。

NFS Version 2、NFS Version 3、およびその他のタイプのクライアントは、リフェラルをたどることができます (シンボリックリンクのように見える)。

どのような場合に NFS リフェラルを使用するか

NFS リフェラルは、複数のサーバーにわたって 1 組のファイル名に見えるものを作成し、この目的に `autofs` を使用しない場合に役立ちます。NFS Version 4 サーバーのみを使用でき、サーバーはリフェラルをホストするために Oracle Solaris 11.1 リリース以降を実行している必要があります。

NFS リフェラルの作成

NFS リフェラルの作成には `nfsref` コマンドを使用します。リフェラルが作成されマウント点がまだ存在しない場合、シンボリックリンクが生成されます。このシンボリックリンクは、再解析ポイントとしてオブジェクトを識別する特別なフラグを含んでいます。再解析ポイントは、特殊な処理が必要であることを示すために使用される特別なマーカーです。再解析ポイントがすでに存在している場合は、状況に応じて、NFS サービスデータが追加されるか、既存の NFS サービスデータと置き換えられません。

NFS リフェラルの削除

NFS リフェラルの削除には `nfsref` コマンドを使用します。このコマンドは指定された再解析ポイントから NFS サービスデータを削除します。ほかのタイプのサービスデータが存在しない場合、再解析ポイントも削除します。

autofs のしくみ

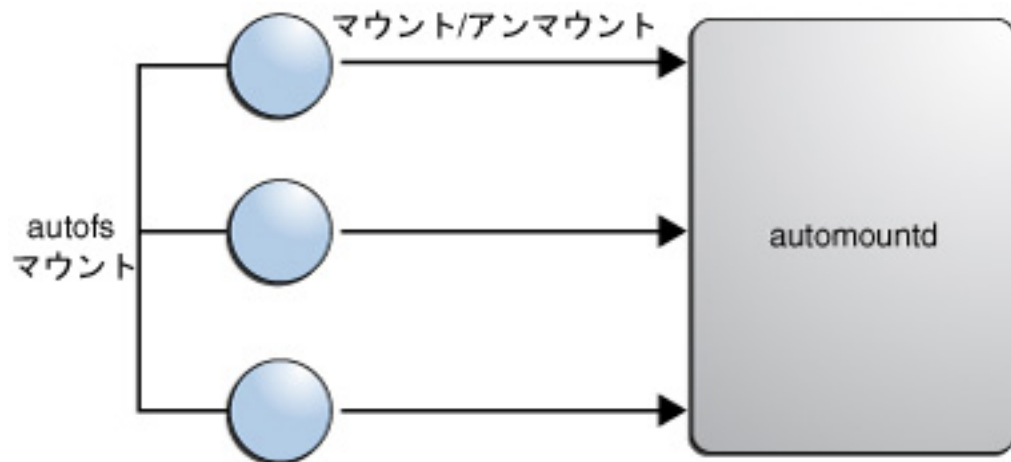
`autofs` は、自動マウント操作とアンマウント操作をサポートするカーネルファイルシステムの 1 つです。自動マウントを行うために、次のコンポーネントが連動します。

- automount コマンド
- autofs ファイルシステム
- automountd デモン

autofs は自動的に適切なファイルシステムをマウントするクライアント側サービスです。自動マウントサービス (svc:/system/filesystem/autofs) は、システムが起動するときに呼び出され、マスターマップファイル `auto_master` を読み取って autofs マウントの初期セットを作成します。これらの autofs マウントは起動時ではなく、あとからファイルシステムがマウントされる時点で自動的にマウントされます。このようなポイントをトリガーノードと呼ぶこともあります。ナビゲーションプロセスの起動の詳細は、60 ページの「[autofs のナビゲーションプロセス開始法](#)」を参照してください。

次の図は、autofs サービスが automount コマンドを起動する方法を示しています。

図 3 svc:/system/filesystem/autofs サービスによる automount の起動



autofs マウントが設定されると、これらのマウントはそれらに基づいたファイルシステムのマウントをトリガーできます。たとえば、autofs が、現在マウントされていないファイルシステムをアクセスする要求を受け取ると、automountd を呼び出して要求されたファイルシステムを実際にマウントさせます。

autofs マウントポイントで、ファイルシステムへのアクセスが要求された場合は、次の動作が行われます。

1. autofs がその要求に介入します。
2. autofs は要求されたファイルシステムをマウントするよう、automountd デーモンにメッセージを送信します。
3. automountd デーモンがマップからファイルシステム情報を見つけ、トリガーノードを作成し、マウントを実行します。
4. autofs は、介入した要求の実行を続行させます。
5. そのファイルシステムが一定期間非アクティブである場合、autofs はそのファイルシステムをアンマウントします。

最初に autofs マウントをマウントしたあとは、必要に応じて automount コマンドを使用し、autofs マウントを更新します。このコマンドは、auto_master マップにあるマウントのリストと、マウントテーブルファイル /etc/mnttab (前のバージョンでは /etc/mtab) にあるマウントされたファイルシステムのリストを比較します。その後、automount によって、適切な変更が加えられます。このプロセスにより、システム管理者は auto_master 内のマウント情報を変更し、autofs デーモンを停止したり再起動したりすることなく、それらの変更を autofs プロセスに使用させることができます。ファイルシステムがマウントされれば、以後のアクセスに automountd は不要になります。次に automountd が必要になるのは、ファイルシステムが自動的にアンマウントされたときです。

mount とは異なり、automount はマウントすべきファイルシステムのリストを調べるために /etc/vfstab ファイル (コンピュータごとに異なる) を参照しません。automount コマンドは、ドメイン内とコンピュータ上で名前空間とローカルファイルを通して制御されます。

注記 - autofs サービスによって管理されるマウントは、手動でマウントまたはアンマウントは行わないでください。たとえこの操作がうまくいったとしても、autofs サービスはオブジェクトがアンマウントされたことを認識しないので、一貫性が損なわれる恐れがあります。リブートによって、autofs のマウントポイントがすべてクリアされます。

autofs のネットワークナビゲート

autofs は一連のマップを検索することによって、ネットワークをナビゲートします。マップは、ネットワーク上の全ユーザーのパスワードエントリや、ネットワーク上の全ホストコンピュータの名前などの情報を含むファイルです。マップには UNIX の管理ファイルに相当するネットワーク規模の管理ファイルも含まれています。マップはローカルに使用するか、あるいは NIS や LDAP のようなネットワークネームサービスを通じて使用できます。

autofs マップ

autofs は 3 種類のマップを使用します。

- マスターマップ
- 直接マップ
- 間接マップ

autofs マスターマップ

auto_master マップは、ディレクトリをマップに関連付けます。このマップは、すべてのマップを指定するマスターリストであり、autofs が参照します。次の例は auto_master ファイルに含まれている可能性のある情報のタイプを示しています。

例 1 /etc/auto_master ファイルの例

```
# Master map for automounter
#
+auto_master
/net          -hosts          -nosuid,nobrowse
/home        auto_home      -nobrowse
/nfs4        -fedfs           -ro,nosuid,nobrowse
/-           auto_direct     -ro
```

この例では、汎用の auto_master ファイルに auto_direct マップのための追加が行われています。マスターマップ /etc/auto_master の各行は、次の構文に従っています。

mount-point map-name [mount-options]

mount-point ディレクトリのフル (絶対) パス名です。このディレクトリが存在しない場合、可能ならば autofs はこのディレクトリを作成します。このディレクトリが存在し、しかも空ではない場合、マウントすることによってその内容が隠されます。この場合、autofs は警告を出します。

マウントポイントとして /- を指定すると、この特定のマップが直接マップであり、マップに関連付けられている特定のマウントポイントがないことを表します。

map-name 位置に対する指示またはマウント情報を検出するために autofs が使用するマップの名前です。この名前がスラッシュ (/) で始まる場合、autofs はこの名前をローカルファイルとして解釈します。それ以外の場合、autofs はネームサービススイッチ構成ファイル (/etc/nsswitch.conf) で指定される検索を使用してマウント情報を検索します。また、/net には、特別なマップを使用します。

詳細は、56 ページの「[/net マウントポイント](#)」を参照してください。

mount-options (オプション) *map-name* で指定されたエントリのマウントに適用されるオプションのコンマ区切りリスト (*map-name* のエントリがほかのオプションをリストしている場合を除く)。特定タイプのファイルシステムのオプションについては、そのファイルシステムの *mount* のマニュアルページを参照してください。NFS 固有のマウントオプションについては、[mount_nfs\(1M\)](#) のマニュアルページを参照してください。NFS 固有のマウントポイントの場合、*bg* (バックグラウンド) オプションと *fg* (フォアグラウンド) オプションは適用されません。

で始まる行はコメントです。その行のテキストの最後まですべて無視されます。

長い行を短い行に分割するには、行末にバックslash (\) を入力します。入力できる文字数の上限は 1024 です。

注記 - 2 つのエントリで同じマウントポイントが使用されるときは、1 番目のエントリは *automount* コマンドが使用します。2 番目のエントリは無視されます。

/home マウントポイント

/home マウントポイントは、*/etc/auto_home* (間接マップ) にリストされたエントリがマウントされるディレクトリです。

注記 - *autofs* はすべてのコンピュータで動作し、デフォルトでは */net* と */home* (自動マウントされるホームディレクトリ) をサポートします。これらのデフォルトエントリは、NIS *auto.master* マップ内または */etc/auto_master* ファイルをローカル編集することで、オーバーライドできます。

/net マウントポイント

autofs は、ホストデータベースのみを使用する特殊な組み込みマップ *-hosts* 内の全エントリを */net* ディレクトリの下にマウントします。たとえば、*hosts* データベースにあるコンピュータ *system1* が、そのファイルシステムをエクスポートするとします。次のコマンドを入力すると、現在のディレクトリがコンピュータ *gumbo* のルートディレクトリに変更されます。

```
# cd /net/gumbo
```

autofs はホスト *system1* のエクスポート済みファイルシステム (つまり、ローカルディスク上のそれらのファイルシステムではなく、ネットワークユーザーが使用できるサーバー上のそれらのファイルシステム) だけをマウントできます。したがっ

て、`system1` 上のすべてのファイルおよびディレクトリを `/net/system1` から使用できるとは限らないこともあります。

`/net` を使用したアクセスでは、サーバー名はパスの中に指定されるため、位置に依存します。したがって、エクスポートされるファイルシステムを別のサーバーに移動すると、そのパスは使用できなくなります。代わりに、`/net` を使用するのではなく、マップ内でそのファイルシステムに対応するエントリを設定することをお勧めします。

注記 - NFS Version 3 およびそれより前のプロトコルを使用すると、autofs はマウント時のみサーバーのエクスポートリストを調べます。サーバーのファイルシステムが一度マウントされると、そのファイルシステムがアンマウントされ、次にマウントされるまで autofs はそのサーバーをチェックしません。したがって、新たにエクスポートされたファイルシステムは、それがサーバーからアンマウントされ、再度マウントされるまでは見えません。NFS Version 4 を使用するシステムでは、ミラーマウントによって、エクスポート済みファイルシステムのリストに加えられた動的変更がサーバーに反映されます。

/nfs4 マウントポイント

`/nfs4` マウントポイントは、擬似マップを使用して FedFS ドメインルートをマウントします。`/nfs4/example.net` を参照すると、ドメインルートで DNS ドメイン `example.net` を検索し、その場所でマウントしようとします。`/nfs4` の下でパスをマウントするには、[95 ページの「FedFS サーバー用の DNS レコードの設定」](#) で説明されているとおり、DNS サーバーがレコードを返す必要があります。

直接 autofs マップ

直接マップは自動マウントポイントです。つまり、直接マップによって、クライアント上のマウントポイントとサーバー上のディレクトリが直接対応付けられます。直接マップにはフルパス名があり、明示的に関係を示します。次の例に典型的な `/etc/auto_direct` マップを示します。

```
/usr/local      -ro \
  /bin          system1:/export/local/sun4 \
  /share        system1:/export/local/share \
  /src          system1:/export/local/src
/usr/man        -ro system2:/usr/man \
               system3:/usr/man \
               system4:/usr/man
/usr/games      -ro system5:/usr/games
/usr/spool/news -ro system6:/usr/spool/news \
               system4:/var/spool/news
```

直接マップ内の行は、次の構文に従います。

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

<i>key</i>	直接マップ内のマウントポイントのパス名。
<i>mount-options</i>	この特定のマウントに適用するオプション。これらのオプションが必要なのは、マップのデフォルトと異なる場合だけです。特定タイプのファイルシステムのオプションについては、そのファイルシステムの <code>mount</code> のマニュアルページを参照してください。NFS 固有のマウントオプションについては、 mount_nfs(1M) のマニュアルページを参照してください。
<i>location</i>	ファイルシステムの場所。NFS ファイルシステムの場合、1 つまたは複数のファイルシステムが <code>server:pathname</code> として指定されます。

注記 - パス名に自動マウントされたマウントポイントを含めるべきではありません。パス名は、ファイルシステムへの実際の絶対パスにするようにしてください。たとえば、ホームディレクトリの位置は、`server:/home/username` ではなく、`server:/export/home/username` としてリストするようにしてください。

マスターマップと同様、# で始まる行はコメントです。その行のテキストの最後まですべて無視されます。長い行を短い行に分割するには、行の最後にバックスラッシュを入力します。

すべてのマップにおいて、直接マップ内のエントリは、`/etc/vfstab` 内の対応するエントリにもっともよく似ています。`/etc/vfstab` のエントリは、次のようになるとします。

```
dancer:/usr/local - /usr/local/tmp nfs - yes ro
```

直接マップ内では、同じエントリが次のようになります。

```
/usr/local/tmp -ro dancer:/usr/local
```

注記 - オートマウントマップの間では、オプションの連結はされません。オートマウントマップに追加されたどのオプションも、前に検索されたマップに表示されているすべてのオプションをオーバーライドします。たとえば、`auto_master` マップに指定されているオプションは、他のマップの中の対応するエントリによってオーバーライドされます。

直接 autofs マップの機能については、[63 ページの「autofs がクライアント用のもっとも近い読み取り専用ファイルを選択する方法」](#)を参照してください。

/- マウントポイント

例1 「[/etc/auto_master ファイルの例](#)」のマウントポイント `/-` は、`auto_direct` 内のエントリを特定のマウントポイントに関連付けないように autofs に指示します。間接マップは、`auto_master` ファイルに定義されたマウントポイントを使います。直接

マップの場合は、名前付きマップ内で指定したマウントポイントを使用します。直接マップ内では、鍵、つまりマウントポイントはフルパス名です。

NIS の `auto_master` ファイルには、直接マップのエントリは1つしか存在できません。マウントポイントは1つの名前空間の中で一意の値にする必要があるためです。`auto_master` がローカルファイルならば、重複しないかぎり直接マップのエントリがいくつあってもかまいません。

間接 autofs マップ

間接マップは、鍵の置換値を使って、クライアント上のマウントポイントとサーバー上のディレクトリとの間の関連付けを確立します。間接マップは、ホームディレクトリなどの特定のファイルシステムをアクセスするのに便利です。`auto_home` マップは間接マップの一例です。

間接マップ内の行は次の汎用構文に従います。

`key [mount-options] location`

<code>key</code>	間接マップでのスラッシュ (/) のない名前。
<code>mount-options</code>	この特定のマウントに適用するオプション。これらのオプションが必要なのは、マップのデフォルトと異なる場合だけです。特定タイプのファイルシステムのオプションについては、そのファイルシステムの <code>mount</code> のマニュアルページを参照してください。たとえば、NFS 固有のマウントオプションについては、 <code>mount_nfs(1M)</code> のマニュアルページを参照してください。
<code>location</code>	ファイルシステムの場所。1つまたは複数のファイルシステムを <code>server: pathname</code> として指定します。

注記 - パス名に自動マウントされたマウントポイントを含めるべきではありません。パス名は、ファイルシステムへの実際の絶対パスにするようにしてください。たとえば、ディレクトリの位置は、`server:/usr/local`, not as `server:/net/server/usr/local` として指定するようにしてください。

マスターマップと同様、# で始まる行はコメントです。その行のテキストの最後まですべて無視されます。長い行を短い行に分割するには、行の最後にバックスラッシュ (\) を入力します。例1「[/etc/auto_master ファイルの例](#)」に、次のエントリを含む `auto_master` マップを示します。

```
/home      auto_home      -nobrowse
```

`auto_home` は、`/home` の下にマウントされるエントリを含む間接マップの名前です。通常、`auto_home` マップには、次のパスが含まれています。

```

user1          server1:/export/home/user1
user2          server2:/export/home/user2
user3          server3:/export/home/user3
user4          server4:/export/home/user4
user5          server5:/export/home/user5
user6          server6:/export/home/user6
user7  -rw,nosuid server7:/export/home/user7

```

例として、前のマップがホスト `master-server` にあると想定します。パスワードデータベースに、ユーザー `user7` のホームディレクトリを `/home/user7` として指定するエントリがあるとします。`user7` がコンピュータ `master-server` にログインするたびに、`autofs` は、コンピュータ `server7` に常駐するディレクトリ `/export/home/user7` をマウントします。彼女のホームディレクトリは、読み書き可能な `nosuid` にマウントされます。

次の状況が発生したと想定してください。ユーザー `user7` のホームディレクトリがパスワードデータベースに、`/home/user7` としてリストされます。`user7` も含むけれども、`auto_home` マップを参照するマスターマップで設定された任意のコンピュータからこのパスにアクセスできます。

これらの状況では、ユーザー `user7` はこれらのコンピュータのいずれかで `login` や `rlogin` を実行し、自分のホームディレクトリを所定の場所にマウントさせることができます。

さらに、`user7` はここで次のコマンドを入力することもできます。

```
# cd ~user1
```

`autofs` は `user1` のために `user7` のホームディレクトリをマウントします (すべてのアクセス権が許可する場合)。

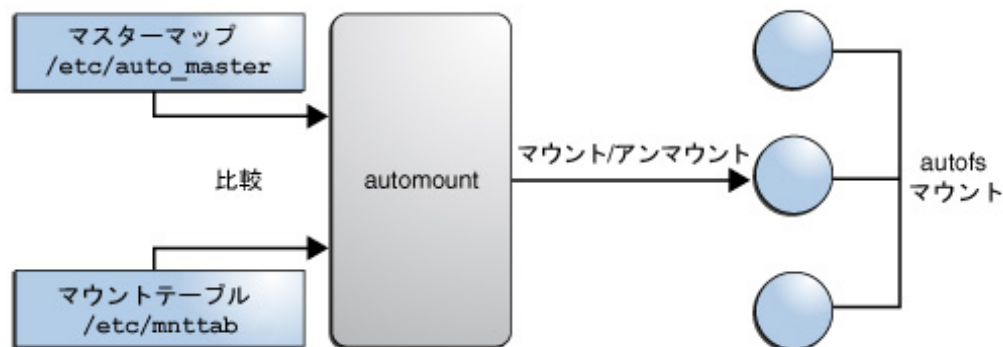
注記 - オートマウントマップの間では、オプションの連結はされません。オートマウントマップに追加されたどのオプションも、前に検索されたマップに表示されているすべてのオプションをオーバーライドします。たとえば、`auto_master` マップに含まれているオプションは、他のいずれかのマップの対応するエントリによってオーバーライドされます。

ネームサービスのないネットワークで、`Linda` が自分のファイルにアクセスすることを許可するには、ネットワーク上のすべてのシステムで、すべての関連ファイル (`/etc/passwd` など) を変更する必要があります。NIS では、NIS マスターサーバーで変更を行い、関連するデータベースをスレーブのデータベースに伝達します。

autofs のナビゲーションプロセス開始法

`automount` コマンドはシステムの起動時にマスターマップを読み取ります。マスターマップ内の各エントリは、直接マップ名または間接マップ名、そのパス、およびそのマウントオプションです。エントリの具体的な順序は重要ではありません。

図 4 マスターマップによるナビゲーション



この図は、`automount` がマスターマップ内のエン트리とマウントテーブル内のエントリとを比較して、現在のリストを生成することを示しています。

autofs マウントプロセス

マウント要求がトリガーされたときに `autofs` サービスが何をするかは、オートマウントマップの構成によって異なります。マウントプロセスの基本はすべてのマウントで同じですが、指定されているマウントポイントとマップの複雑さによって結果が変わります。マウントプロセスにはトリガーノードの作成が含まれます。

単純な `autofs` マウント

`autofs` マウントプロセスの説明のために、次のファイルがインストールされていると仮定します。

```
$ cat /etc/auto_master
# Master map for automounter
#
+auto_master
/net      -hosts      -nosuid,nobrowse
/home     auto_home   -nobrowse
/share    auto_share
$ cat /etc/auto_share
# share directory map for automounter
#
ws        gumbo:/export/share/ws
```

/share ディレクトリがアクセスされると、autofs サービスは /share/ws に対するトリガーノードを作成します。これは、/etc/mnttab の中では次のようなエントリになります。

```
-hosts /share/ws    autofs  nosuid,nobrowse,ignore,nest,dev=###
```

/share/ws ディレクトリがアクセスされると、autofs サービスは次ような処理を完了します。

1. サーバーのマウントサービスが使用可能かどうかを確認します。
2. 要求されたファイルシステムを、/share の下にマウントします。これで、/etc/mnttab ファイルには次のエントリが追加されます。

```
-hosts /share/ws    autofs  nosuid,nobrowse,ignore,nest,dev=###
gumbo:/export/share/ws /share/ws  nfs    nosuid,dev=####  #####
```

階層型マウント

オートマウンタファイルに複数の層が定義されているときは、マウントプロセスは複雑になります。前の例の /etc/auto_shared ファイルを拡張して、次の行を追加したとします。

```
# share directory map for automounter
#
ws      /      gumbo:/export/share/ws
        /usr  gumbo:/export/share/ws/usr
```

/share/ws マウントプロセスは基本的に、マウントポイントがアクセスされる前の例と同じです。さらに、/share/ws ファイルシステム内に次のレベル (/usr) へのトリガーノードが作成されるので、次のレベルがアクセスされたときにマウントできます。この例でトリガーノードが作成されるためには、NFS に /export/share/ws/usr が存在している必要があります。



注意 - 階層的にマウントを指定する場合は、-soft オプションは使用しないでください。詳細は、[62 ページの「autofs アンマウント」](#) を参照してください。

autofs アンマウント

一定のアイドル時間後に発生するアンマウントは、下位から上位方向です (マウントと逆の順序)。あるディレクトリより上位のディレクトリが使用中であれば、それより下のディレクトリだけがアンマウントされます。アンマウントすると、トリガーノードがすべて削除され、ファイルシステムがアンマウントされます。ファイルシステムが使用中であれば、アンマウントは失敗してトリガーノードは再インストールされません。



注意 - 階層的にマウントを指定する場合は、`-soft` オプションは使用しないでください。`-soft` オプションを使用すると、トリガーノードを再インストールする要求がタイムアウトすることがあります。トリガーノードを再インストールできないと、マウントの次の階層にアクセスできません。この問題を解決するには、オートマウンタを使用して、階層にあるすべてのコンポーネントのマウントを解除します。オートマウンタでアンマウントするには、ファイルシステムが自動的にアンマウントされるのを待つか、システムをリブートします。

autofs がクライアント用のもっとも近い読み取り専用ファイルを選択する方法

このセクションでは、autofs がどのようにもっとも近い読み取り専用ファイルをクライアントに選択するかを説明するために、次の直接マップの例を使用します。

```
/usr/local      -ro \
  /bin          ivy:/export/local/sun4\
  /share       ivy:/export/local/share\
  /src         ivy:/export/local/src
/usr/man        -ro oak:/usr/man \
               rose:/usr/man \
               willow:/usr/man
/usr/games      -ro peach:/usr/games
/usr/spool/news -ro pine:/usr/spool/news \
               willow:/var/spool/news
```

マウントポイント `/usr/man` および `/usr/spool/news` には 1 つ以上の場所 (最初のマウントポイントには 3 つの場所、2 番目のマウントポイントには 2 つの場所) がリストされています。複製された場所のどこからマウントしてもユーザーは同じサービスを受けられます。ユーザーの書き込みまたは変更が可能ならば、その変更をロケーション全体で管理しなければならないので、この手順は、読み取り専用のファイルシステムをマウントするときだけに意味があります。あるときに、あるサーバー上のファイルを変更し、そのすぐあとに別のサーバー上で「同じ」ファイルを変更するといった作業は避けたいものです。この利点は、最善の利用可能なサーバーがユーザーの作業なしで自動的に使用されることです。

ファイルシステムを複製として構成してあると (42 ページの「複製されたファイルシステムとは」を参照)、NFS クライアントはフェイルオーバー機能を使用できます。最適な NFS サーバーが自動的に決定されるだけでなく、そのサーバーが使用できなくなるとクライアントは自動的に 2 番目に適したサーバーを使います。

複製として構成するのに適しているファイルシステムの例は、マニュアルページです。大規模なネットワークでは、複数のサーバーがマニュアルページをエクスポートできます。どのサーバーからマニュアルページをマウントするかは、サーバーが実行中でそのファイルシステムをエクスポートしているかぎり、重要ではありません。直接マップの例では、複数のマウント場所は、マップエントリ内のマウント場所のリストとして表現されています。

```
/usr/man -ro oak:/usr/man rose:/usr/man willow:/usr/man
```

この例では、サーバー oak、rose、または willow からマニュアルページをマウントできます。どのサーバーが最適であるかは、次のいくつかの要素によって決まります。

- 特定の NFS プロトコルレベルをサポートするサーバーの数
- サーバーの近接性
- 重み付け

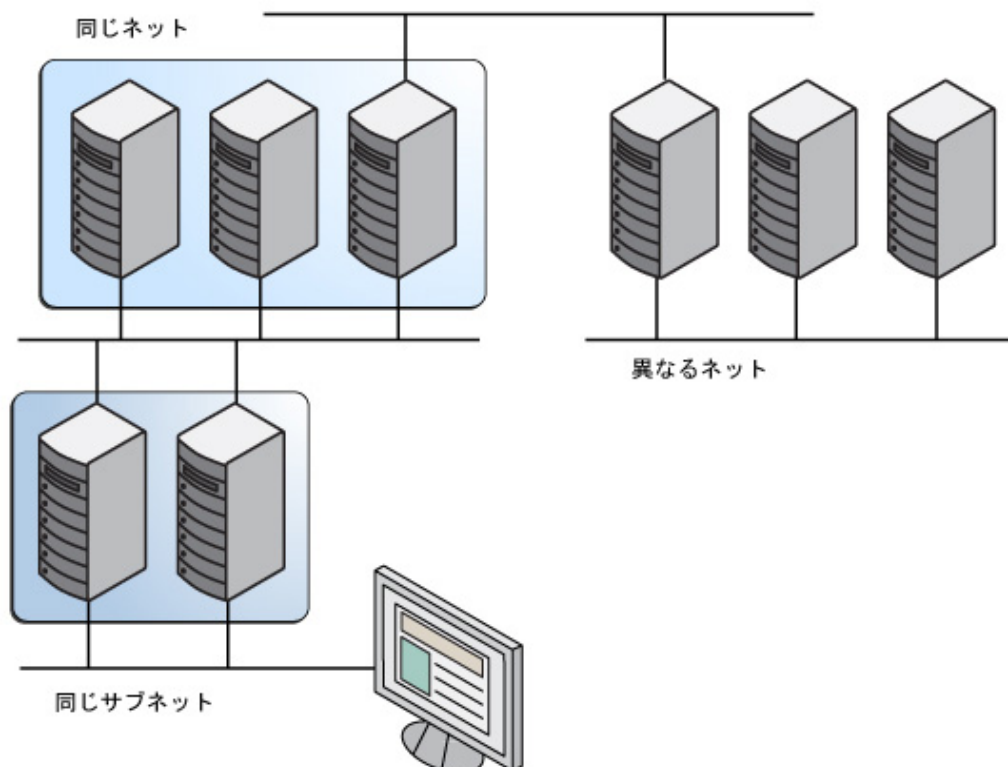
順位を決定するときには、各バージョンの NFS プロトコルをサポートしているサーバーの数が数えられます。サポートしているサーバーの数が多いうプロトコルがデフォルトになります。これによって、クライアントにとっては利用できるサーバーの数が最大になります。

プロトコルが同じバージョンのサーバーの組の中で数がもっとも多いものがわかると、サーバーのリストが距離によってソートされます。近接性を判定するために、IPv4 アドレスが調査されてどのサーバーが各サブネットにあるかが判別されます。ローカルサブネット上のサーバーには、リモートサブネット上のサーバーよりも高い優先順位が付けられます。もっとも近いサーバーが優先されることにより、待ち時間とネットワークトラフィックが軽減されます。

注記 - IPv6 アドレスを使用している複製に対しては、距離を判定できません。

図5は、サーバー近接性を示しています。

図 5 サーバー近接性



ローカルサブネット上に同じプロトコルをサポートしているサーバーが複数あるときは、それぞれのサーバーに接続する時間が計測され、速いものが使用されます。重み付けを使用することで選別が影響されることもあります。重み付けの詳細は、67 ページの「[autofs と重み付け](#)」を参照してください。

たとえば、ローカルサブネット上で NFS Version 4 サーバーの方が多く場合には、NFS Version 4 がデフォルトで使用されるプロトコルになります。ただし、ローカルサブネット上のサーバーが異なるプロトコルをサポートするときは、選別プロセスが複雑になります。次に、優先順位の決定の例をいくつか示します。

- ローカルサブネット上のサーバーには、リモートサブネット上のサーバーよりも高い優先順位が付けられます。ローカルサブネット上に NFS Version 3 サーバーがあり、もっとも近い NFS Version 4 サーバーがリモートサブネット上にある場合は、NFS Version 3 サーバーが優先されます。同様に、ローカルサブネットが NFS

Version 2 サーバーで構成される場合は、それらが NFS Version 3 と NFS Version 4 サーバーで構成されるリモートサブネットより優先されます。

- ローカルサブネットが NFS Version 2、NFS Version 3、および NFS Version 4 サーバーで構成されていて、それぞれ数が異なる場合は、より複雑な選別が必要になります。オートマウンタは、ローカルサブネット上でもっとも高いバージョンを優先します。この場合、NFS Version 4 がもっとも高いバージョンです。ただし、ローカルサブネット上で NFS Version 4 サーバーよりも NFS Version 3 または NFS Version 2 サーバーが多い場合には、オートマウンタはローカルサブネット上のもっとも高いバージョンから 1 バージョン「競り下げ」ます。たとえば、ローカルサブネットが NFS Version 4 で 3 サーバー、NFS Version 3 で 3 サーバー、NFS Version 2 で 10 サーバーで構成される場合は、NFS Version 3 サーバーが選択されません。
- 同じように、ローカルサブネットが NFS Version 2 と NFS Version 3 サーバーで構成され、それぞれ数が異なる場合は、オートマウンタは最初にどのバージョンがローカルサブネット上でもっとも高いバージョンを示しているかを見つけてみます。次に、オートマウンタは各バージョンを実行するサーバーの数を数えます。ローカルサブネット上でもっとも高いバージョンが、同時にもっとも多いサーバーの場合、もっとも高いバージョンが選択されます。低いバージョンのサーバーの数が多く、オートマウンタはローカルサブネットのもっとも高いバージョンから 1 つ下のバージョンを選択します。たとえば、ローカルサブネット上で NFS Version 2 サーバーが NFS Version 3 サーバーよりも多い場合、NFS Version 2 サーバーが選択されます。

注記 - 重み付けには、SMF リポジトリに保存されているパラメータも影響します。具体的には、`server_versmin`、`client_versmin`、`server_versmax`、および `client_versmax` の値によって、いくつかのバージョンを選別プロセスから除外できます。これらのパラメータの詳細は、[167 ページの「NFS デーモン」](#) を参照してください。

フェイルオーバー機能を指定していると、この優先順位はサーバーが選択されるマウント時に確認されます。複数の場所を指定しておく、個々のサーバーが一時的にファイルシステムをエクスポートできないときに便利です。

多くのサブネットで構成される大きなネットワークでは、フェイルオーバーが特に便利です。autofs は適切なサーバーを選択して、ネットワークトラフィックをローカルネットワークのセグメントに限定することができます。サーバーが複数のネットワークインタフェースを持つ場合は、それぞれのインタフェースが別々のサーバーであるとみなして、各ネットワークインタフェースに対応付けられているホスト名を指定します。autofs はそのクライアントにいちばん近いインタフェースを選択します。

注記 - 手動によるマウントでは、重み付けと距離の確認は行われません。mount コマンドは、左から右へ一覧表示されるサーバーの優先順位を付けます。

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

autofs と重み付け

距離のレベルが同じサーバーから 1 つを選択するために、autofs マップに重み付けの値を追加することができます。次に例を示します。

```
/usr/man -ro oak,rose(1),willow(2):/usr/man
```

括弧内の数値が重み付けを示します。重み付けのないサーバーの値はゼロであり、選択される可能性が最高になります。重み付けの値が大きいほど、そのサーバーが選択される可能性は低くなります。

注記 - 重み付けは、サーバーの選択に関係する要素の中でもっとも小さい影響力しかありません。ネットワーク上の距離が同じサーバーの間で選択を行う場合に考慮されるだけです。

autofs マップエントリ内の変数

名前の前にドル記号 (\$) を付けることによって、クライアント固有の変数を作成できます。変数は、ファイルシステム内の同じ場所に異なるタイプのアーキテクチャーがアクセスできるようにするのに役立ちます。変数名を括弧でくくることで、その後続く文字や数字と変数とを区切ることができます。次の表は、定義済みマップ変数を示しています。

表 1 定義済みのマップ変数

変数	意味	派生元	例
ARCH	アーキテクチャータイプ	uname -m	sun4
CPU	プロセッサタイプ	uname -p	sparc
HOST	ホスト名	uname -n	system1
OSNAME	オペレーティングシステム名	uname -s	SunOS
OSREL	オペレーティングシステムリリース	uname -r	5.10
OSVERS	オペレーティングシステムのバージョン (リリースのバージョン)	uname -v	GENERIC

鍵として使用する場合を除いて、変数はエントリ行内のどこにでも使用できます。たとえば、`/usr/local/bin/sparc` から SPARC アーキテクチャー、`/usr/local/bin/x86` から x86 アーキテクチャーのバイナリをそれぞれエクスポートするファイルサーバーがあるとします。クライアントは、次のようなマップエントリを使ってマウントすることができます。

```
/usr/local/bin -ro server:/usr/local/bin/$CPU
```

これですべてのクライアントの同じエントリがすべてのアーキテクチャーに適用されます。

注記 - どの sun4 アーキテクチャー向けに書かれたアプリケーションでも、ほとんどはすべての sun4 プラットフォームで実行できます。-ARCH 変数は、sun4 にハードコードされています。

他のマップを参照するマップ

ファイルマップ内のマップエントリ内のマップ名で使用される特殊文字は、マップ名の処理方法に影響します。

- マップエントリ `+mapname` がファイルマップで使用されている場合、`automount` は指定されたマップが現在のファイルに含まれているかのように読み取ります。
- `mapname` の前にスラッシュがない場合、`autofs` はそのマップ名を文字列として扱い、ネームサービススイッチ方式を使用してマップ名を検出します。パス名が絶対パス名の場合、`automount` はその名前のローカルマップを検索します。
- マップ名がダッシュ (-) で始まる場合、`automount` は `hosts` などの適切な組み込みマップを参照します。

`svc:system/name-service/switch` サービスはネームサービスの検索順序を保持しています。`config` プロパティグループの `automount` プロパティは、自動マウントエントリを探すときのネームサービスデータベースの検索順序を指定します。特定の `config/automount` プロパティが指定されていない場合は、`config/default` プロパティで定義された順序が使用されます。

例 2 automount コマンドでマップの検索順序を表示する

```
# svcprop -p config svc:/system/name-service/switch
config/value_authorization astring solaris.smf.value.name-service.switch
config/printer astring user\ files
config/default astring files\ nis
config/automount astring files\ nis
```

この例は、ローカルファイル内のマップが NIS マップよりも先に検索されることを示しています。`config/automount` プロパティが指定されていなかったとしても、`config/default` エントリが使用されるため、結果は同じになります。そのため、ローカルマップ `/etc/auto_home` に、もっとも頻繁にアクセスするホームディレクトリ用のエントリを含めることができます。他のエントリについては、スイッチを使用して NIS マップにフォールバックすることができます。

```
bill          cs.csc.edu:/export/home/bill
bonny         cs.csc.edu:/export/home/bonny
```

取り込まれたマップを参照したあと、一致するものがなければ、`automount` は現在のマップの走査を続けます。そのため、+エントリの後にさらにエントリを追加できます。

```
bill          cs.csc.edu:/export/home/bill
bonny        cs.csc.edu:/export/home/bonny
+auto_home
```

組み込まれたマップは、ローカルファイルまたは組み込みマップとすることができます。ローカルファイルだけが+エントリを含めることができます。

```
+/etc/auto_mystuff    # local map
+auto_home            # NIS map
+-hosts               # built-in hosts map
```

注記 - NIS マップでは「+」エントリを使用できません。

実行可能な autofs マップ

autofs マウントポイントを生成するコマンドを実行する autofs マップを作成できます。データベースやフラットファイルから autofs 構造を作成できる必要がある場合は、実行可能 autofs マップが役立ちます。短所は、マップをすべてのホストにインストールしなければならないことです。実行可能なマップは、NIS および LDAP ネームサービスに含めることができません。

実行可能マップは、`auto_master` ファイルにエントリが必要です。

```
/execute    auto_execute
```

次の例は、サンプル実行可能マップを示しています。

```
#!/bin/ksh
#
# executable map for autofs
#

case $1 in
    src) echo '-nosuid,hard bee:/export1' ;;
esac
```

この例が機能するためには、ファイルが `/etc/auto_execute` としてインストールされ、実行可能ビットが設定されている必要があります。アクセス権は 744 に設定します。この場合、次のコマンドを実行すると、`bee` のファイルシステム `/export1` がマウントされます。

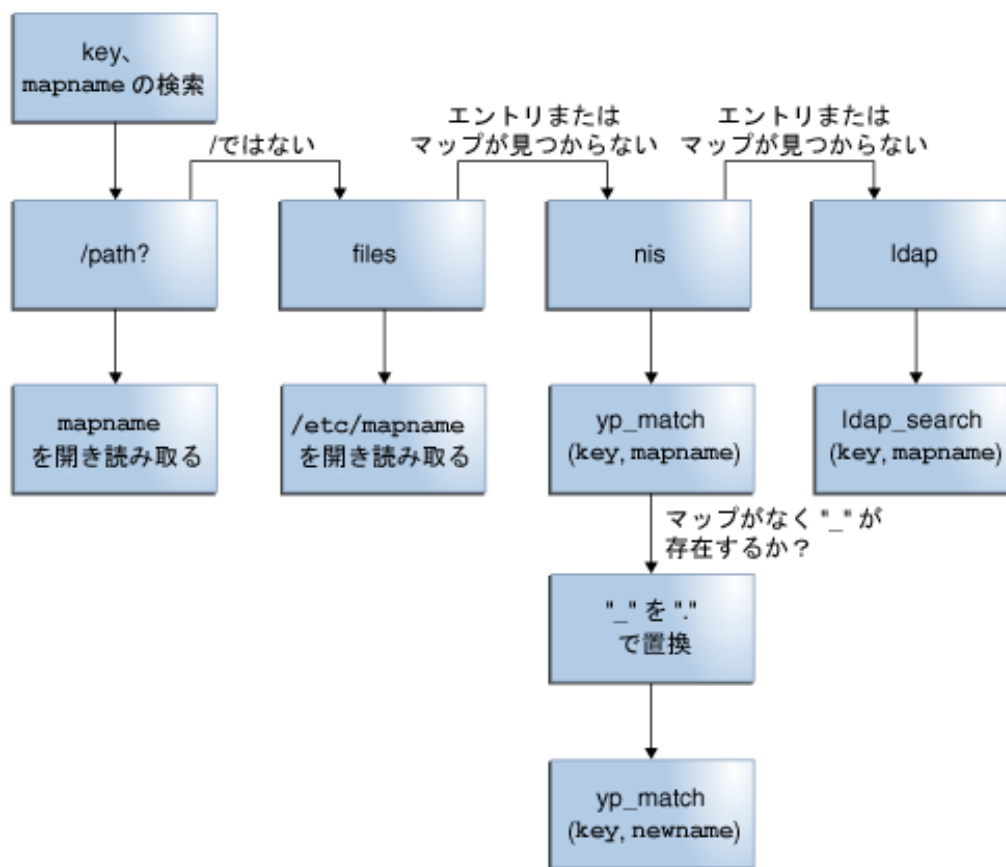
```
# ls /execute/src
```

ネームサービスに対する autofs のデフォルトの動作

ブート時に、autofs がサービス `svc:/system/filesystem/autofs` によって起動され、autofs がマスター `auto_master` マップを確認します。

autofs は、`svc:/system/name-service/switch` サービスの `config/automount` プロパティで指定されたネームサービス順序を使用します。`config/automount` プロパティが定義されていない場合は、`config/default` プロパティが使用されません。NIS が選択されていて、autofs が使用できるマップを autofs が検出できないけれども、1 つまたは複数の下線を含むマップ名を検出した場合は、従来の NIS ファイル名が機能するように下線がドットに変更されます。次に autofs はもう一度マップを調べます。`automount: files nis ldap` というネームサービススイッチ設定の場合、マップは次の図のように検索されます。

図 6 autofs によるネームサービスの使用



このセッションでは、画面は次の例のようになります。

```
$ grep /home /etc/auto_master
/home          auto_home
```

```
$ ypmatch brent auto_home
Can't match key brent in map auto_home. Reason: no such map in
server's domain.
```

```
$ ypmatch brent auto.home
diskus:/export/home/diskus1/&
```

ネームサービスとして「ファイル」が選択された場合、すべてのマップは `/etc` ディレクトリ内のローカルファイルとみなされます。autofs は、使用するネームサービスとは無関係に、スラッシュ (/) で始まるマップ名をローカルとして解釈します。

autofs リファレンス

このセクションでは autofs の高度な機能とトピックについて説明します。

autofs とメタキャラクタ

autofs は一部の文字を、特別な意味を持つものとして認識します。たとえば、ある文字が置き換えに使用されたり、ある文字が autofs マップ構文解析機能からほかの文字を保護するために使用されたりします。

アンパサンド (&)

たとえば、次の例のように、多くのサブディレクトリが指定されているマップがある場合は、文字列置換を使用することを検討してください。

```
john      willow:/home/john
mary      willow:/home/mary
joe       willow:/home/joe
able     pine:/export/able
baker    peach:/export/baker
```

この場合、アンパサンド文字 (&) を使用して、任意の位置に記述されたこのキーを置換することができます。アンパサンドを使用すると、前述のマップは次のテキストに変わります。

```
john      willow:/home/&
mary      willow:/home/&
joe       willow:/home/&
able     pine:/export/&
baker    peach:/export/&
```

キー置換を、直接マップ内で次の例のような置換で使用することもできます。

```
/usr/man      willow,cedar,poplar:/usr/man
```

また、このエントリは、次のようにさらに簡単にすることができます。

```
/usr/man      willow,cedar,poplar:&
```

アンパサンド文字による置換では、鍵文字列全体を使用していることに注意してください。そのため、直接マップ内の鍵の最初の文字が / である場合は、そのスラッシュ

が置換に含まれます。したがって、たとえば次のエントリを含めることはできません。

```
/progs    &1,&2,&3:/export/src/progs
```

autofs はこの例を次のように解釈します。

```
/progs    /progs1,/progs2,/progs3:/export/src/progs
```

アスタリスク (*)

任意のキーを照合するために、汎用置換文字、アスタリスク (*) を使用できます。たとえば、このマップエントリによってすべてのホストから /export ファイルシステムをマウントできます。

```
*        &:/export
```

ここでは、各アンパサンドは特定の鍵の値によって置換されています。autofs はこのアスタリスクをファイルの終わりとして解釈します。

autofs と特殊文字

特殊文字で構成されるマップエントリがある場合は、autofs マップ構文解析機能が適切に処理できない名前を持つディレクトリをマウントしなければならないことがあります。autofs の構文解析機能は、名前に含まれるコロン、コンマ、スペースなどを認識します。これらの名前は次の例のように二重引用符で囲むようにしてください。

```
/vms      -ro    vmserver: - - - "rc0:dk1 - "  
/mac      -ro    gator:/ - "Mr Disk - "
```


◆◆◆ 第 3 章

ネットワークファイルシステムの管理

この章では、NFS サービスの設定、共有する新規ファイルシステムの追加、ファイルシステムのマウントなど、NFS 管理タスクの実行方法について説明します。この章では NFS および FedFS リフェラルを構成および保守するために使用する手順についても説明します。

この章の内容は、次のとおりです。

- 76 ページの「ファイルシステムの自動共有」
- 78 ページの「ファイルシステムのマウント」
- 85 ページの「NFS サービスの設定」
- 89 ページの「Secure NFS システムの管理」
- 91 ページの「WebNFS の管理」
- 94 ページの「NFS リフェラルの管理」
- 95 ページの「FedFS の管理」

注記 - システムでゾーンが有効になっていて、非大域ゾーンでこの機能を使用する場合は、『Oracle Solaris ゾーンの紹介』を参照してください。

ネットワークファイルシステムの管理について

NFS 管理者の責任は、サイトの要求やネットワーク管理者としての役割によって変わります。ローカルネットワークのシステム全部を担当する場合は、次の決定の責任を持つ可能性があります。

- 専用 NFS サーバーとして提供できるシステム
- NFS サーバーおよび NFS クライアントの両方として提供できるシステム
- NFS クライアントのみを提供するシステム

設定が完了したサーバーの保守には、次のタスクが必要です。

- ファイルシステムの共有開始と共有解除

- 管理ファイルを修正し、システムが自動的にマウントするファイルシステムのリストを更新する
- ネットワークのステータスのチェック
- NFS に関連した問題の診断と解決
- autofs のマップの設定

システムは NFS サーバーにも NFS クライアントにもなれます。つまり、システムを使用して、ローカルファイルシステムをリモートシステムと共有したり、リモートファイルシステムをマウントしたりできます。

ファイルシステムの自動共有

Oracle Solaris 11 リリースでは、share コマンドが永続的な共有を作成し、この共有はシステムの起動時に自動的に共有されます。以前のリリースとは異なり、次回以降のリブートのために /etc/dfs/dfstab ファイルを編集して共有に関する情報を記録する必要はありません。このファイルは使用されなくなりました。

ファイルシステム共有のタスクマップ

次のタスクマップは、NFS サービスを使用したファイルシステム共有を説明する手順にリンクします。

表 2 ファイルシステムの共有 (タスクマップ)

タスク	説明	手順
ファイルシステムの自動共有を確立します	サーバーのリブート時、ファイルシステムが自動的に共有されるようにサーバーを構成します。	76 ページの「ファイルシステム自動共有を設定する方法」
NFS サーバーログを有効にします	NFS ロギングが選択したファイルシステム上で動作するようにサーバーを構成します。	77 ページの「NFS サーバーログを有効にする方法」

▼ ファイルシステム自動共有を設定する方法

1. 管理者になります。
詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「[割り当てられている管理権利の使用](#)」を参照してください。
2. 共有するファイルシステムを定義します。

注記 - NFS サーバーのプロセスはデフォルトでは起動しません。

ZFS ファイルシステムを共有するように `share.nfs` プロパティを設定できます。

```
# zfs set share.nfs=on | off filesystem
```

次に例を示します。

```
# zfs set share.nfs=on pond/amy
```

使用可能なコマンドオプションについては、[zfs_share\(1M\)](#) のマニュアルページを参照してください。

3. ファイルシステムが共有されていることを確認します。

`share` コマンドを使用して、すべての共有ファイルシステムのリストを取得できます。

次に例を示します。

```
# share
pond_amy                /pond/amy                nfs sec=sys,rw
```

次の手順 次の手順では、サーバー上で共有したファイルシステムにクライアントがアクセスできるように `autofs` マップを設定します。 `autofs` マップを設定する方法の詳細は、[表6](#)を参照してください。

▼ NFS サーバーログを有効にする方法

1. 管理者になります。

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. ファイルシステム構成値を変更します。

次のいずれかの方法で、`/etc/nfs/nfslog.conf` ファイルの設定を変更できます。

- すべてのファイルシステムのデフォルト設定を編集するために、`global` タグに関連付けられたデータを変更します。
- ファイルシステムに新しいタグを追加します。

`/etc/nfs/nfslog.conf` ファイルの形式については、[nfslog.conf\(4\)](#) のマニュアルページを参照してください。

3. NFS サーバーログを使用するファイルシステムを定義します。

`share` コマンドを使用して、各ファイルシステムを定義します。 `log=tag` オプションで使用するタグは、`/etc/nfs/nfslog.conf` ファイルに指定する必要があります。

次の例では、`global` タグのデフォルト設定を使用します。

```
# share -F nfs -ro,log=global /export/ftp
```

4. 指定したオプションがリストされていることを確認します。

次に例を示します。

```
# share -F nfs
export_share_man      /export/share/man    sec=sys,ro
usr_share_src         /usr/src             sec=sys,rw=eng
export_ftp            /export/ftp          public,log=global,sec=sys,ro
```

5. NFS ログデーモン `nfslogd` が動作していることを確認します。

```
# ps -ef | grep nfslogd
```

6. `nfslogd` デーモンの状態を確認します。

```
# svcadm restart network/nfs/server:default
```

ファイルシステムのマウント

ファイルシステムは、システムのブート時、コマンド行からの要求時、またはオートマウンタによって、自動的にマウントできます。オートマウンタには、ブート時やコマンド行からのマウントと比較していくつもの利点があります。状況によってこの3つの方法を組み合わせる必要があります。また、ファイルシステムのマウント時に使用するオプションに応じて、プロセスを有効または無効にする方法がいくつかあります。

ファイルシステムマウントのタスクマップ

次の表に、ファイルシステムマウントに関連付けられたタスクをリストします。

表 3 ファイルシステムのマウント (タスクマップ)

タスク	説明	手順
ブート時にファイルシステムをマウントします	システムがリブートされるときに必ずファイルシステムがマウントされるようにします。	79 ページの「ブート時にファイルシステムにマウントする方法」
コマンドを使用してファイルシステムをマウントします	システムが動作しているときにファイルシステムをマウントします。この手順はテストに役立ちます。	80 ページの「コマンド行からファイルシステムをマウントする方法」
オートマウンタでファイルシステムをマウントします	コマンド行を使用することなく、オンデマンドでファイルシ	80 ページの「オートマウンタによるマウント」

タスク	説明	手順
	ステムにアクセスできるようにします。	
ミラーマウントを使用してすべてのファイルシステムをマウントします	1つのサーバーからすべてのファイルシステムをマウントします。	81 ページの「サーバーからすべてのファイルシステムをマウントする方法」
クライアント側フェイルオーバーを開始します	サーバーで失敗した場合にアクティブなファイルシステムへの自動フェイルオーバーを有効にします。	81 ページの「クライアント側フェイルオーバーを使用する方法」
クライアントに対するマウントアクセスを無効にします	あるクライアントがリモートシステムにアクセスする機能を無効にします。	82 ページの「1つのクライアントに対するマウントアクセスを無効にする方法」
ファイアウォールを越えてファイルシステムにアクセスを提供します	WebNFS プロトコルを使用することでファイアウォール経由でファイルシステムにアクセスできるようにします。	82 ページの「ファイアウォールを越えて NFS ファイルシステムをマウントする方法」
NFS URL を使ってファイルシステムをマウントします	NFS URL を使用することでファイルシステムにアクセスできるようにします。このプロセスによって、MOUNT プロトコルを使用しないでファイルシステムにアクセスできます。	83 ページの「NFS URL を使用して NFS ファイルシステムをマウントする方法」

▼ ブート時にファイルシステムにマウントする方法

この手順では、autofs マップを使用するのではなく、ブート時にファイルシステムをマウントする方法を示します。この手順は、リモートファイルシステムにアクセスする必要があるすべての NFS クライアントで完了する必要があります。

1. 管理者になります。

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

2. ファイルシステムに関するエントリを /etc/vfstab ファイルに追加します。

/etc/vfstab ファイルのエントリの構文は次のとおりです。

```
#device      device      mount      FS      fsck      mount      mount
#to mount    to fsck     point      type    pass     at boot   options
```

/etc/vfstab ファイルエントリについては、[vfstab\(4\)](#)のマニュアルページを参照してください。



注意 - NFS クライアント vfstab エントリを持つ NFS サーバーには、リブート時のシステムハングアップを避けるために、常に `bg` オプションも指定する必要があります。詳細は、[mount\(1M\)](#) のマニュアルページを参照してください。

3. NFS クライアントサービスを有効にします。

```
# svcadm enable -r nfs/client
```

例 3 クライアントの /etc/vfstab ファイルのエントリ

クライアントシステムにサーバー wasp から /var/mail ディレクトリをマウントするとします。ファイルシステムをクライアントに /var/mail としてマウントし、そのクライアントに読み取り - 書き込みアクセスを付与します。次のエントリをクライアントの vfstab ファイルに追加することになります。

```
wasp:/var/mail - /var/mail nfs - yes rw
```

▼ コマンド行からファイルシステムをマウントする方法

新しいマウントポイントをテストするために、コマンド行からファイルシステムをマウントすることがよく実行されます。このタイプのマウントでは、オートマウンタでアクセスできないファイルシステムに一時的にアクセスできます。ファイルシステムは、`umount` でまたはローカルホストをリブートすることでアンマウントできます。

1. 管理者になります。

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. ファイルシステムをマウントします。

```
mount -F nfs -o specific-options resource mount-point
```

例:

```
# mount -F nfs -o ro bee:/export/share/local /mnt
```

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

この例では、サーバー bee からの /export/share/local ファイルシステムが、ローカルシステムの /mnt に読み取り専用でマウントされます。



注意 -mount コマンドは無効なオプションについての警告を提供しません。解釈できないオプションがあると無視されるだけです。予期しない動作を避けるために、使用するオプションはすべて確認してください。

オートマウンタによるマウント

NFS クライアントは、汎用システムを変更することなく、/net マウントポイント経由でリモートファイルシステムにアクセスできます。オートマウンタによるマウント

の確立およびサポートについては、表6を参照してください。/export/share/local ファイルシステムをマウントするには次のコマンドを入力します。

```
# cd /net/bee/export/share/local
```

すべてのユーザーがオートマウンタでファイルシステムをマウントできるので、root アクセスは必要ありません。オートマウンタはファイルシステムを自動的にアンマウントするので、アクセスする必要がなくなったあとにファイルシステムを手動でアンマウントする必要はありません。

▼ サーバーからすべてのファイルシステムをマウントする方法

自動ミラーマウント機能によって、NFS クライアントはサーバーから1つのマウントを正常に完了したあと、そのNFSサーバーからNFSを使用して共有されているすべての使用可能なファイルシステムにアクセスできます。ミラーマウントが自動的に実行されるので、必要なのはそのファイルシステムへのアクセスだけです。詳細は、50ページの「ミラーマウントのしくみ」を参照してください。

1. 管理者になります。

詳細は、『Oracle Solaris 11.3でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. サーバーのエクスポートされた名前空間のルートをマウントします。

このコマンドは、サーバーからファイルシステム階層をクライアント上にミラー化します。この例では、/mnt/export/share/local ディレクトリ構造が作成されます。

```
# mount bee:/ /mnt
```

3. ファイルシステムにアクセスします。

このコマンド、またはそのファイルシステムにアクセスするほかのコマンドによって、ファイルシステムがマウントされます。

```
# cd /mnt/export/share/local
```

▼ クライアント側フェイルオーバーを使用する方法

1. 管理者になります。

詳細は、『Oracle Solaris 11.3でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. **NFS クライアント上で、ro オプションを使用してファイルシステムをマウントします。**

コマンド行から、オートマウンタで、または /etc/vfstab ファイルに次のようなエントリを追加することでマウントできます。

```
bee,wasp:/export/share/local - /usr/local nfs - no ro
```

注記 - 異なるバージョンの NFS プロトコルを実行しているサーバーを、コマンド行または vfstab エントリで発行されるコマンドで混在させることはできません。NFS Version 2、NFS Version 3、または NFS Version 4 プロトコルをサポートするサーバーの混在は、autofs でのみ実行できます。autofs では、NFS Version 2、NFS Version 3、または NFS Version 4 サーバーのもっとも良いサブセットが使用されます。

▼ 1つのクライアントに対するマウントアクセスを無効にする方法

1. **管理者になります。**

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. **1つのクライアントに対するマウントアクセスを無効にします。**

```
# share -F nfs -o specific_options pathname
```

次に例を示します。

```
# share -F nfs ro=-rose:eng /export/share/man
```

```
ro=-rose:eng      eng ネットワークグループ内のすべてのクライアントへの読み取り専用マウントアクセスを許可するアクセスリスト (rose という名前のホストを除く)
```

```
/export/share/   共有するファイルシステム  
man
```

▼ ファイアウォールを越えて NFS ファイルシステムをマウントする方法

始める前に この手順では、NFS サーバーのファイルシステムを public オプションで共有する必要があります。また、NFS クライアントと NFS サーバー間のファイアウォールでは、

ポート 2049 で TCP 接続できるようにする必要があります。共有しているすべてのファイルシステムに、公開ファイルハンドルでアクセスできます。そのため、デフォルトでは、`public` オプションが適用されています。

1. 管理者になります。

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

2. 次のコマンドを使用して、ファイルシステムを手動でマウントします。

```
# mount -F nfs host:pathname mount-point
```

次に例を示します。

```
# mount -F nfs bee:/export/share/local /mnt
```

この例では、ファイルシステム `/export/share/local` は、公開ファイルハンドルを使用することでローカルクライアントにマウントされます。標準のパス名の代わりに、NFS URL を使用することができます。ただし `bee` サーバーで公開ファイルハンドルがサポートされていないと、マウント操作は失敗します。

NFS URL を使用した NFS ファイルシステムのマウント

NFS URL に、`public` オプションを含めるかどうかを選択できます。`public` オプションを指定しない場合、サーバーが公開ファイルハンドルをサポートしていなければ、MOUNT プロトコルが使用されます。`public` オプションを指定すると、必ず公開ファイルハンドルを使用するように指定され、公開ファイルハンドルがサポートされていないとマウントは失敗します。

注記 - ファイルシステムのマウント時に使用される NFS プロトコルバージョンは、NFS クライアントと NFS サーバーの両方によってサポートされる最上位バージョンです。ただし、`vers=#` オプションを使用して特定の NFS プロトコルバージョンを選択できます。

▼ NFS URL を使用して NFS ファイルシステムをマウントする方法

1. 管理者になります。

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

2. NFS URL を使用したファイルシステムの手動マウント。

```
# mount -F nfs nfs://host[:port]/pathname mount-point
```

例 4 NFS URL を使用した NFS ファイルシステムのマウント

```
# mount -F nfs nfs://bee:3000/export/share/local /mnt
```

この例では、NFS ポート番号 3000 を使って、サーバー bee から /export/share/local ファイルシステムがマウントされています。ポート番号は必要なく、デフォルトで NFS 標準ポート番号 2049 が使用されます。

マウントに使用できるファイルシステムに関する情報の表示

showmount コマンドは、リモートでマウントされているファイルシステムまたはマウントに使用できるファイルシステムに関する情報を表示します。共有ファイルシステムをリストするには、-e オプションを使用してください。次に例を示します。

```
# /usr/sbin/showmount -e bee
export list for bee:
/export/share/local (everyone)
/export/home tulip,lilac
/export/home2 rose
```

ほかのオプションについては、[showmount\(1M\)](#) のマニュアルページを参照してください。

一部の環境では、共有ファイルシステムおよびそれらをマウントしたシステムに関する情報は表示されるべきではありません。sharectl コマンドの showmount_info プロパティを none に設定することで、次のファイルシステム情報をクライアントが表示できないことが保証されます。

- クライアントがアクセスできないファイルシステムに関する情報
- すべての共有ファイルシステムに関する情報
- ファイルシステムをマウントしたほかのシステムに関する情報

例 5 クライアントに表示されるファイルシステム情報を制限する

```
bee# sharectl set -p showmount_info=none nfs
```

次の情報がクライアント rose に表示されます。

```
# /usr/sbin/showmount -e bee
export list for bee:
/export/share/local (everyone)
/export/home2 rose
```

/export/home ファイルシステムに関する情報が表示されなくなります。

NFS サービスの設定

このセクションでは、NFS サービスの設定に必要ないくつかのタスクについて説明します。

注記 - NFS Version 4 は、Oracle Solaris 11.2 でサポートされているデフォルトバージョンの NFS です。

表 4 NFS サービスの設定

タスク	説明	手順
NFS サーバーを起動および停止する	NFS サービスが自動的に開始されなかった場合は、開始してください。NFS サービスを停止します。通常は、サービスを停止する必要はありません。	85 ページの「 NFS サービスの起動と停止 」
オートマウンタを起動および停止する	オートマウンタを起動および停止します。オートマウンタマップが変更された場合、この手順が必要です。	85 ページの「 オートマウンタの起動と停止 」
異なるバージョンの NFS を選択する	サーバーとクライアントで NFS Version 4 以外の NFS バージョンを選択します。	86 ページの「 異なるバージョンの NFS の選択 」

NFS サービスの起動と停止

管理者として、svcadm コマンドを使用してサーバーの NFS サービスを有効化および無効化してください。

- サーバーで NFS サービスを有効にするには:

```
# svcadm enable network/nfs/server
```

- サーバーで NFS サービスを無効にするには:

```
# svcadm disable network/nfs/server
```

オートマウンタの起動と停止

管理者として、svcadm コマンドを使用して autofs デーモンを有効化および無効化してください。

- autofs デーモンを有効にするには:

- ```
svcadm enable system/filesystem/autofs
```
- autofs デーモンを無効にするには:

```
svcadm disable system/filesystem/autofs
```

## 異なるバージョンの NFS の選択

NFS Version 4 以外のバージョンの NFS を使用する必要がある場合、別のバージョンを選択できます。

- NFS サーバーで異なるバージョンの NFS を選択する場合は、[86 ページの「サーバー上で異なるバージョンの NFS を選択する方法」](#)を参照してください。
- NFS クライアントで異なるバージョンの NFS を選択する場合は、[87 ページの「クライアント上で異なるバージョンの NFS を選択する方法」](#)を参照してください。
- コマンド行を使用して NFS クライアントで異なるバージョンの NFS を選択する場合は、[88 ページの「mount コマンドを使用してクライアント上で異なるバージョンの NFS を選択する方法」](#)を参照してください。

### ▼ サーバー上で異なるバージョンの NFS を選択する方法

デフォルトに設定されている NFS Version 4 を使用しないことを選択する場合は、別のバージョンの NFS を選択できます。

**1. 管理者になります。**

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

**2. SMF パラメータを変更して NFS バージョン番号を設定します。**

たとえば、サーバーから NFS Version 3 のみを提供する場合は、`server_versmax` および `server_versmin` パラメータの両方の値を 3 に設定します。

```
sharectl set -p server_versmax=3 nfs
sharectl set -p server_versmin=3 nfs
```

**3. サーバー委託を無効にする場合、`server_delegation` プロパティを変更します。**

```
sharectl set -p server_delegation=off nfs
```

NFS サーバー委託によって NFS クライアントはファイルをキャッシュできます (別の NFS クライアントが同じファイルへのアクセスを必要とするまで)。NFS Version 4 で

は、サーバー委託はデフォルトで有効になっています。詳細は、[35 ページの「NFS Version 4 における委託」](#)を参照してください。

4. クライアントとサーバーに共通のドメインを設定する場合は、`nfsmapid_domain` プロパティを変更します。

クライアントとサーバー間のユーザー ID またはグループ ID マッピングを有効にするために、クライアントとサーバーに共通ドメインを設定できます。

```
sharectl set -p nfsmapid_domain=my.example.com nfs
```

`my.example.com` は共通ドメイン名を提供します。

`nfsmapid` デーモンの詳細は、[167 ページの「NFS デーモン」](#)を参照してください。

5. NFS サービスがサーバー上で動作しているかどうかを確認します。

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

6. 必要に応じて、NFS サービスを有効にします。

NFS サービスがオフラインの場合、次のコマンドを入力してサービスを有効にします。

```
svcadm enable network/nfs/server
```

NFS サービスの構成については、[76 ページの「ファイルシステム自動共有を設定する方法」](#)を参照してください。

参照 [28 ページの「NFS におけるバージョンのネゴシエーション」](#)

## ▼ クライアント上で異なるバージョンの NFS を選択する方法

次の手順では、クライアント上で使用される NFS バージョンを制御する方法を説明します。デフォルトで設定される NFS バージョンは NFS Version 4 です。

1. 管理者になります。

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. SMF パラメータを変更して NFS バージョン番号を設定します。

たとえば、すべてのファイルシステムを NFS Version 3 プロトコルを使用してマウントする必要がある場合は、`client_versmax` と `client_versmin` パラメータの両方の値を 3 に設定します。

```
sharectl set -p client_versmax=3 nfs
```

```
sharectl set -p client_versmin=3 nfs
```

### 3. クライアント上で NFS をマウントします。

```
mount server-name:/share-point /local-dir
```

|                     |                 |
|---------------------|-----------------|
| <i>server-name</i>  | サーバーの名前。        |
| <i>/share-point</i> | リモートディレクトリのパス   |
| <i>/local-dir</i>   | ローカルマウントポイントのパス |

参照 [28 ページの「NFS におけるバージョンのネゴシエーション」](#)

## ▼ mount コマンドを使用してクライアント上で異なるバージョンの NFS を選択する方法

次の手順では、mount コマンドを使用して、クライアントで特定のマウントに使用される NFS バージョンを制御する方法を説明します。クライアントによってマウントされるすべてのファイルシステムで NFS バージョンを変更する方法を確認するには、[87 ページの「クライアント上で異なるバージョンの NFS を選択する方法」](#)を参照してください。

### 1. 管理者になります。

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

### 2. クライアント上で、目的のバージョンの NFS をマウントします。

```
mount -o vers=value server-name:/share-point /local-dir
```

|                     |                 |
|---------------------|-----------------|
| <i>value</i>        | NFS バージョン番号     |
| <i>server-name</i>  | サーバーの名前         |
| <i>/share-point</i> | リモートディレクトリのパス   |
| <i>/local-dir</i>   | ローカルマウントポイントのパス |

---

注記 - このコマンドは、SMF リポジトリ内の NFS クライアント設定をオーバーライドします。

---

参照 [28 ページの「NFS におけるバージョンのネゴシエーション」](#)



## Secure NFS システムの管理

Secure NFS システムを使用するには、責任を持っているすべてのシステムにドメイン名を付与する必要があります。一般的に、ドメインはいくつかのシステムの管理エンティティで、大きなネットワークの一部になっています。ネームサービスを実行している場合、そのドメインに対してネームサービスを設定するようにしてください。ネームサービスについては、『[Oracle Solaris 11.3 ディレクトリサービスとネームサービスでの作業: DNS と NIS](#)』を参照してください。

NFS サービスでは、Kerberos version 5 認証もサポートされています。詳細は、『[Oracle Solaris 11.3 での Kerberos およびその他の認証サービスの管理](#)』の第 2 章、「[Kerberos サービスについて](#)」を参照してください。

Secure NFS 環境は、Diffie-Hellman 認証を使用するように構成することもできます。Diffie-Hellman 認証の詳細は、『[Oracle Solaris 11.3 での Kerberos およびその他の認証サービスの管理](#)』の第 10 章、「[ネットワークサービスの認証の構成](#)」を参照してください。

### ▼ DH 認証を使用して Secure NFS 環境を設定する方法

#### 1. ドメイン名を割り当てます。

ドメイン名をドメイン内の各システムに認識させます。マシンの NIS ドメイン名の設定については、『[Oracle Solaris 11.3 ディレクトリサービスとネームサービスでの作業: DNS と NIS](#)』の「[マシンの NIS ドメイン名を設定する方法](#)」を参照してください。

```
domainname domain-name
```

#### 2. newkey コマンドを使用することで、クライアントのユーザーの公開鍵と秘密鍵を設定します。

```
newkey -u username -s name-service
```

ユーザーは chkey コマンドを使用することでパーソナル Secure RPC パスワードを確立できます。

```
chkey -p -s name-service -m mechanism
```

公開鍵と秘密鍵が生成されると、公開鍵と暗号化された秘密鍵が publickey データベースに格納されます。

これらのコマンドについては、[newkey\(1M\)](#) および [chkey\(1\)](#) のマニュアルページを参照してください。

#### 3. ネームサービスが応答していることを確認します。

次に例を示します。

- NIS を実行している場合は、ypbind デーモンが動作していることを確認してください。詳細は、『Oracle Solaris 11.3 ディレクトリサービスとネームサービスでの作業: DNS と NIS』の「ypbind が NIS クライアントで実行されていない」を参照してください。
- LDAP を実行している場合は、ldap\_cachemgr デーモンが動作していることを確認してください。詳細は、『Oracle Solaris 11.3 ディレクトリサービスとネームサービスでの作業: LDAP』の「LDAP クライアントステータスのモニタリング」を参照してください。

#### 4. 鍵サーバーの keyerv デーモンが動作していることを確認します。

```
ps -ef | grep keyerv
root 100 1 16 Apr 11 ? 0:00 /usr/sbin/keyerv
root 2215 2211 5 09:57:28 pts/0 0:00 grep keyerv
```

デーモンが動作していない場合は、次のように入力して鍵サーバーを起動します。

```
svcadm enable network/rpc/keyerv
```

#### 5. 秘密鍵を復号化し、保存します。

通常、ログインパスワードはネットワークパスワードと同じです。この場合、keylogin は不要です。パスワードが異なる場合、ユーザーはログインしてから keylogin を実行する必要があります。さらに、keylogin -r コマンドを root として使用して、復号化した秘密鍵を /etc/.rootkey に保存する必要があります。

---

注記 - root 秘密鍵が変化している場合、または /etc/.rootkey ファイルが失われている場合は、keylogin -r を実行する必要があります。

---

#### 6. 共有されるファイルシステムのセキュリティーモードを設定します。

Diffie-Hellman 認証の場合は、コマンド行に sec=dh オプションを追加します。

```
share -F nfs -o sec=dh /export/home
```

セキュリティーモードの詳細については、[nfssec\(5\)](#) のマニュアルページを参照してください。

#### 7. ファイルシステムに対するオートマウントマップを更新します。

Diffie-Hellman 認証を使用している場合は、auto\_master データを編集して、適切なエントリ内にマウントオプションとして sec=dh を含めます。

```
/home auto_home -nosuid,sec=dh
```

システムを再インストール、移動、またはアップグレードするときに、root の新しい鍵を確立しない場合またはそれらを変更しない場合でも、忘れずに /etc/.rootkey ファイルを保存してください。/etc/.rootkey ファイルを削除する場合は、次のコマンドを入力します。

```
keylogin -r
```

## WebNFS の管理

このセクションでは、WebNFS システムを管理する手順について説明します。

表 5 WebNFS の管理 (タスクマップ)

| タスク                             | 説明                                                                                     | 手順                                           |
|---------------------------------|----------------------------------------------------------------------------------------|----------------------------------------------|
| WebNFS を計画します                   | WebNFS サービスを有効にする前に考慮する項目。                                                             | 91 ページの「WebNFS アクセスの計画」                      |
| WebNFS を有効にします                  | WebNFS プロトコルを使用することで NFS ファイルシステムのマウントを有効にします。                                         | 92 ページの「WebNFS アクセスを有効にする方法」                 |
| ファイアウォール経由で WebNFS を有効にします      | WebNFS プロトコルを使用することでファイアウォール経由でファイルにアクセスできるようにします。                                     | 94 ページの「ファイアウォール経由 WebNFS アクセスの有効化」          |
| NFS URL を使ってブラウズします             | Web ブラウザ内で NFS URL を使用します。                                                             | 93 ページの「ブラウザを使用した NFS URL へのアクセス」            |
| autofs で公開ファイルハンドルを使用します        | オートマウントでファイルシステムをマウントするときに公開ファイルハンドルを使用します。                                            | 110 ページの「autofs で公開ファイルハンドルを使用する方法」          |
| autofs で NFS URL を使用します         | オートマウントマップに NFS URL を追加します。                                                            | 111 ページの「autofs で NFS URL を使用する方法」           |
| ファイアウォールを越えてファイルシステムにアクセスを提供します | WebNFS プロトコルを使用することでファイアウォール経由でファイルシステムにアクセスできるようにします。                                 | 82 ページの「ファイアウォールを越えて NFS ファイルシステムをマウントする方法」  |
| NFS URL を使ってファイルシステムをマウントします    | NFS URL を使用することでファイルシステムにアクセスできるようにします。このプロセスによって、MOUNT プロトコルを使用しないでファイルシステムにアクセスできます。 | 83 ページの「NFS URL を使用して NFS ファイルシステムをマウントする方法」 |

## WebNFS アクセスの計画

WebNFS を使用するには、まず NFS URL (たとえば、`nfs://server/path`) を実行およびロードできるアプリケーションが必要です。次に、WebNFS アクセスのためにエクスポートするファイルシステムを選択します。アプリケーションが Web ブラウザの場合は、Web サーバーのドキュメントのルートがよく使用されます。WebNFS アクセスのためにエクスポートするファイルシステムを選択するときは、次の事項を検討する必要があります。

- 各サーバーには 1 つの公開ファイルハンドルが割り当てられています (デフォルトではサーバーのルートファイルシステムに関連付けられている)。NFS URL に示されたパスは、この公開ファイルハンドルに関連付けられているディレクトリからの

相対パスとして評価されます。その結果としてパスが示す先のファイルまたはディレクトリが、エクスポートされたファイルシステムの中にあると、サーバーによってアクセスが実現されます。share コマンドの **public** オプションを使用すると、エクスポートされる特定のディレクトリにこの公開ファイルハンドルを関連付けることができます。このオプションを使用して、URL をサーバーのルートファイルシステムではなく公開ファイルシステムからの相対パスにできます。ルートファイルシステムを共有しないと、ルートファイルシステムへの Web アクセスはできません。

- WebNFS 環境では、すでにマウント権限を持っているユーザーは、ブラウザからファイルにアクセスできます。この機能は、ファイルシステムが **public** オプションを使ってエクスポートされたかどうかに関係なく有効になっています。ユーザーは NFS の設定によってファイルへのアクセス権を持っているため、ブラウザからのアクセスを許すことによって新たにセキュリティが損なわれる恐れはありません。ファイルシステムをマウントできないユーザーは、**public** オプションを使ってファイルシステムを共有するだけで、WebNFS アクセスを使用できるようになります。
- すでに公開されているファイルシステムには、**public** オプションを使用することをお勧めします。たとえば、ftp アーカイブの最上位のディレクトリや Web サイトのメイン URL ディレクトリなどです。
- share コマンドで **index** オプションを使用することで、HTML ファイルを強制的にロードできます。そうしない場合は、NFS URL がアクセスされたときにディレクトリが一覧表示されます。

ファイルシステムを選択したあとで、ファイルを確認し、必要に応じてファイルやディレクトリの表示を制限するようにアクセス権を設定してください。アクセス権は、共有される NFS ファイルシステムに合わせて設定します。多くのサイトでは、ディレクトリには 755、ファイルには 644 が適切なアクセスレベルです。

また、NFS と HTTP URL の両方を使用して 1 つの Web サイトにアクセスする場合は、その他の事項も検討する必要があります。WebNFS 制限の詳細は、[46 ページの「Web ブラウザの使用と比較した場合の WebNFS の制約」](#)を参照してください。

## ▼ WebNFS アクセスを有効にする方法

**始める前に** デフォルトでは、NFS マウントで利用可能なすべてのファイルシステムは、自動的に WebNFS アクセスで利用可能になります。この手順は次のいずれかの理由で使用しません。

- NFS マウントが現時点で利用可能になっていないサーバーで NFS マウントができるようにする場合
- share コマンドの **public** オプションを使用することで、公開ファイルハンドルを再設定して NFS URL を短くする場合

- `share` コマンドの `index` オプションを使用することで、HTML ファイルが強制的にロードされるようにする場合

`sharectl` ユーティリティーを使用して NFS などのファイル共有プロトコルを構成することもできます。ファイル共有プロトコルの構成の詳細は、[sharectl\(1M\)](#) のマニュアルページを参照してください。

WebNFS サービスを起動する前に考慮する問題については、[91 ページの「WebNFS アクセスの計画」](#)を参照してください。

### 1. 管理者になります。

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

### 2. WebNFS サービスによって共有されるファイルシステムを定義します。

`share` コマンドを使用して各ファイルシステムを定義します。

```
share -F nfs -o specific-options pathname
```

`share_nfs` コマンドで使用可能なオプションについては、[share\\_nfs\(1M\)](#) のマニュアルページを参照してください。

### 3. 指定したオプションがリストされていることを確認します。

```
share -F nfs
```

次に例を示します。

```
share -F nfs
export_share_man /export/share/man sec=sys,ro
usr_share_src /usr/src sec=sys,rw=eng
export_ftp /export/ftp sec=sys,ro,public,index=index.html
```

## ブラウザを使用した NFS URL へのアクセス

ブラウザが WebNFS サービスをサポートしている場合は、次のような NFS URL にアクセスできます。

```
nfs://server[:port]/path
```

`server`                   ファイルサーバー名

`port`                    使用するポート番号 (デフォルト値は 2049)

`path`                    公開ファイルハンドルまたはルートファイルシステムに関連するファイルへのパス

---

**注記** - ほとんどのブラウザでは、URL サービスタイプ (nfs や http など) はトランザクション間で記憶されます。ただし、異なるタイプのサービスを含む URL を読み込んだ場合に例外があります。たとえば、NFS URL を使用したあとに HTTP URL への参照がロードされた場合、後続のページは NFS プロトコルではなく HTTP プロトコルを使ってロードされます。

---

## ファイアウォール経由 WebNFS アクセスの有効化

ローカルサブネットに属していないクライアントに WebNFS アクセスを有効にするには、ポート 2049 での TCP 接続を許可するようにファイアウォールを構成します。httpd にアクセスを許可するだけでは、NFS URL は使用できません。

## NFS リフェラルの管理

NFS Version 4 サーバーは NFS リフェラルによって、複数の NFS Version 4 サーバーを均一名前空間に接続する手段として、ほかの NFS Version 4 サーバーにあるファイルシステムを参照できます。

### ▼ NFS リフェラルの作成とアクセスの方法

**1. NFS サーバー上で、リフェラルを作成します。**

NFS 共有ファイルシステムにリフェラルを追加し、1 つ以上の既存の NFS 共有ファイルシステムを指します。次に例を示します。

```
server1 nfsref add /share/docs server2:/usr/local/docs server3:/tank/docs
Created reparse point /share/docs
```

**2. リフェラルが作成されたことを確認します。**

```
server1# nfsref lookup /share/docs
/share/docs points to:
server2:/usr/local/docs
server3:/tank/docs
```

**3. クライアント上で、リフェラルをマウントするマウントポイントにアクセスします。**

```
client1# ls /share/docs
```

マウントに失敗した場合は、NFS クライアントで接続を確認し、NFS サーバーで共有ファイルシステムを確認します。NFS のトラブルシューティングの詳細は、[146 ページの「NFS トラブルシューティング手順」](#)を参照してください。

**例 6 既存の NFS リフェラルの変更**

server4:/tank/docs などの別のファイルシステムをこの手順で作成された既存のリフェラルに追加するには、新しいファイルシステムで手順 2 のコマンドを入力してください。

```
server1# nfsref add /share/docs server2:/usr/local/docs \
server3:/tank/docs server4:/tank/docs
```

add サブコマンドによって、現在のリフェラル内の情報がコマンドからの新しい情報で置き換えられます。

**▼ NFS リフェラルを削除する方法**

- NFS リフェラルを削除するには、次のコマンドを入力します。

```
server1# nfsref remove /share/docs
Removed svc_type 'nfs-basic' from /share/docs
```

これによって、/share/docs で作成された単一リフェラルが削除されます。

**FedFS の管理**

フェデレーテッドファイルシステムを構築および保守するには、FedFS プロトコルを使用します。このファイルシステムでは、多数の異なるファイルサーバーを取り込んで、マルチベンダーグローバル名前空間を作成できます。

**FedFS サーバー用の DNS レコードの設定**

適切な DNS レコードが作成されたあとにマウントポイントがアクセスされると、オートマウンタによって FedFS を使用するファイルシステムのマウントが完了されます。サーバーの DNS レコードが次のように表示されます。

```
nslookup -q=srv _nfs-domainroot._tcp.example.com bee.example.com
Server: bee.example.com
Address: 192.168.1.1

_nfs-domainroot._tcp.example.com service = 1 0 2049 bee.example.com.
```

DNS レコードを設定したあと、アプリケーションが /nfs4/example.com マウントポイントにアクセスすると、Oracle Solaris が自動的に FedFS ファイルシステムをマウントします。

## ▼ 名前空間データベースの作成方法

名前空間データベース (NSDB) は、単一 FedFS 名前空間に結合されているさまざまなタイプのサーバーから、ファイルセットに関する情報を提供するために使用されます。この手順は LDAP サーバーで実行されます。

始める前に LDAP サーバーがインストールされている必要があります。詳細は、『[Oracle Solaris 11.3 ディレクトリサービスとネームサービスでの作業: LDAP](#)』の第 4 章、「LDAP クライアントでの [Oracle Directory Server Enterprise Edition のセットアップ](#)」を参照してください

### 1. 管理者になります。

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

### 2. /etc/openldap/slapd.conf ファイルを次のエントリで更新します。

```
include /usr/lib/fs/nfs/fedfs-11.schema
suffix dc=example,dc=org
rootdn cn=Manager,dc=example,dc=org
rootpw password
```

### 3. FedFS データの識別名を作成します。

```
nsdb-update-nci -l NSDB -r port -D bind_DN -w bind-PW nce
```

次に例を示します。

```
nsdb-update-nci -l localhost -r 389 -D cn=Manager -w\
example.org dc=example,dc=org adding new entry "dc=example,dc=org"
NCE entry created
```

ここで

- l NSDB を実装する LDAP サーバーを指定します
- r NSDB を実装する LDAP サーバーが応答を待つポートを指定します
- D NSDB 情報を変更することを許可されたユーザーの識別名を指定します
- w バインド DN ユーザーのパスワードを指定します

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



## ▼ NSDB へのセキュアな接続を使用する方法

始める前に LDAP サーバーがインストールされている必要があります。詳細は、『Oracle Solaris 11.3 ディレクトリサービスとネームサービスでの作業: LDAP』の第 4 章、「LDAP クライアントでの Oracle Directory Server Enterprise Edition のセットアップ」を参照してください。

### 1. 管理者になります。

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

### 2. LDAP サーバーで、証明書を作成します。

LDAP トラフィックを保護する証明書が必要です。

```
mkdir /etc/openldap/certs
mkdir /etc/openldap/certs/keys
cd /etc/openldap/certs
openssl req -x509 -nodes -days 3650 -newkey rsa:2048 \
 -keyout keys/ldapskey.pem -out ldapscert.pem
chown -R openldap:openldap /etc/openldap/certs/*
chmod 0400 keys/ldapskey.pem
```

### 3. /etc/openldap/slapd.conf ファイルに宣言を追加します。

```
TLSertificateFile /etc/openldap/certs/ldapscert.pem
TLSCertificateKeyFile /etc/openldap/certs/keys/ldapskey.pem
```

### 4. 証明書を NFS サーバーおよびクライアントにコピーします。

```
scp ldap-server:/etc/openldap/certs/keys/ldapskey.pem \
 /etc/openldap/certs/keys/ldapskey.pem
chmod 0400 /etc/openldap/certs/keys/ldapskey.pem
```

### 5. NFS サーバーおよびクライアントで、接続エントリを更新します。

```
nsdbparams update -f ldapscert.pem -t FEDFS_SEC_TLS localhost
```

nsdbparams コマンドで使用可能なオプションの詳細は、[nsdbparams\(1M\)](#) のマニュアルページを参照してください。

## ▼ FedFS リフェラルを作成する方法

始める前に NFS サーバーがインストールされている必要があります。

### 1. 管理者になります。

### 2. NSDB 用の接続エントリを作成します。

このコマンドによって、LDAP サーバーおよび NFS サーバーで定義された NSDB 間の接続エントリが作成されます。

```
nsdbparams update -D cn=Manager,dc=example,dc=org -w example.org nsdb.example.org
```

### 3. FedFS リフェラルを作成します。

```
nfsref -t svc-type add path location
```

-t *svc-type*                   リフェラルのサービスタイプを指定します。

次に例を示します。

```
nfsref -t nfs-fedfs add /share/docs server2:/usr/local/docs server3:/tank/docs
Created reparse point /share/doc
```

# ◆◆◆ 第 4 章

## autofs の管理

---

この章では、ファイルシステムにアクセスしたり、autofs マップを変更したり、autofs でセキュリティ制限を使用したりなどの autofs 管理タスクを実行する方法について説明します。

この章の内容は、次のとおりです。

- 99 ページの「autofs 管理」
- 100 ページの「SMF パラメータを使用して autofs 環境を構成する」
- 102 ページの「マップの変更」
- 104 ページの「オートマウンタのカスタマイズ」

---

注記 - システムでゾーンが有効になっていて、非大域ゾーンでこの機能を使用する場合は、『Oracle Solaris ゾーンの見方』を参照してください。

---

## autofs 管理

次の表に、autofs に関連するタスクについての説明と参照箇所を示します。

表 6 autofs を管理するためのタスク

| タスク                               | 説明                                                               | 手順                                                 |
|-----------------------------------|------------------------------------------------------------------|----------------------------------------------------|
| autofs を起動および停止します                | システムをリブートすることなく自動マウントサービスを起動および停止します                             | 85 ページの「オートマウンタの起動と停止」                             |
| autofs SMF パラメータで autofs 環境を構成します | SMF リポジトリ内のパラメータに値を割り当てます                                        | 100 ページの「SMF パラメータを使用して autofs 環境を構成する」            |
| autofs でファイルシステムにアクセスします          | 自動マウントサービスを使ってファイルシステムにアクセスします                                   | 80 ページの「オートマウンタによるマウント」                            |
| autofs マップを修正します                  | ほかのマップをリストするために使用される、マスターマップを変更します<br>ほとんどのマップに使用される、間接マップを変更します | 102 ページの「マスターマップを修正する方法」<br>102 ページの「間接マップを修正する方法」 |

| タスク                                                              | 説明                                                                                                                                                                                              | 手順                                                                                                                                                 |
|------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                  | クライアントおよびサーバー上のマウントポイント間に直接の関連付け確立するために使用される、直接マップを変更します                                                                                                                                        | 103 ページの「直接マップを修正する方法」                                                                                                                             |
| 非 NFS ファイルシステムにアクセスするために autofs マップを修正します<br><br>/home マップを使用します | CD-ROM アプリケーション用のエントリで autofs マップを設定します<br><br>共通 /home マップを設定します<br><br>複数のファイルシステムを参照する /home マップを設定します                                                                                       | 104 ページの「非 NFS ファイルシステムへのアクセス」<br><br>104 ページの「/home の共通ビューの設定」<br>105 ページの「複数のホームディレクトリファイルシステムで /home を設定する方法」                                   |
| 新しい autofs マウントポイントを使用します                                        | プロジェクト関連 autofs マップを設定します<br><br>さまざまなクライアントアーキテクチャーをサポートする autofs マップを設定します<br><br>さまざまなオペレーティングシステムをサポートする autofs マップを設定します                                                                   | 106 ページの「共通ディレクトリにプロジェクト関連ファイルを統合する方法」<br>108 ページの「共有名前空間にアクセスするために異なるアーキテクチャーを設定する方法」<br>109 ページの「非互換のクライアントオペレーティングシステムのバージョンをサポートする方法」          |
| autofs でファイルシステムを複製します                                           | フェイルオーバーしたファイルシステムへのアクセスを提供します                                                                                                                                                                  | 109 ページの「複数のサーバーを通じて共用ファイルを複製する方法」                                                                                                                 |
| autofs でセキュリティ制限を使用します                                           | ファイルへのリモート root アクセスを制限しながらファイルシステムへのアクセスを提供します                                                                                                                                                 | 110 ページの「autofs セキュリティ制限を適用する方法」                                                                                                                   |
| autofs で公開ファイルハンドルを使用します                                         | ファイルシステムのマウント時に公開ファイルハンドルの使用を強制します                                                                                                                                                              | 110 ページの「autofs で公開ファイルハンドルを使用する方法」                                                                                                                |
| autofs で NFS URL を使用します                                          | オートマウントが使用できるように、NFS URL を追加します                                                                                                                                                                 | 111 ページの「autofs で NFS URL を使用する方法」                                                                                                                 |
| autofs ブラウザビリティを無効にします                                           | ブラウザビリティを無効にして、autofs マウントポイントが単一クライアント上に自動的に設定されないようにします<br><br>ブラウザビリティを無効にして、autofs マウントポイントがすべてのクライアント上に自動的に設定されないようにします<br><br>ブラウザビリティを無効にして、特定の autofs マウントポイントがクライアント上に自動的に設定されないようにします | 111 ページの「1つの NFS クライアントの autofs ブラウズ機能を完全に無効にする方法」<br>112 ページの「すべてのクライアントの autofs ブラウズ機能を無効にする方法」<br>112 ページの「選択したファイルシステムの autofs ブラウズ機能を無効にする方法」 |

## SMF パラメータを使用して autofs 環境を構成する

SMF パラメータを使用して autofs 環境を構成できます。特に、この機能は、autofs コマンドおよび autofs デーモンを構成する追加の方法を提供します。コマンド行で行う

ように、`sharectl` コマンドで同じ指定を行うことができます。指定するには、キーワードに値を割り当てます。

## ▼ SMF パラメータを使用して autofs 環境を構成する方法

### 1. 管理者になります。

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。

### 2. autofs SMF パラメータを追加または変更します。

たとえば、すべての autofs マウントポイントの表示をオフに設定する場合は、次のコマンドを使用します。

```
sharectl set -p nobrowse=on autofs
```

`nobrowse` キーワードは、`automountd` コマンドの `-n` オプションと同等です。autofs でサポートされるパラメータについては、[autofs\(4\)](#) を参照してください。

### 3. autofs デーモンを再起動します。

```
svcadm restart system/filesystem/autofs
```

## マップの管理タスク

マップとネームサービスタイプの選択は、autofs マップに変更を加えるときに使用する必要があるメカニズムに影響します。

---

**注記** - できるだけ間接マップを使用してください。間接マップは構築が簡単で、システムのファイルシステムへの負荷が少なく済みます。また、間接マップは直接マップよりもマウントテーブル内のスペースを必要としません。

---

マップのタイプとその使用法です。

- マスターディレクトリをマップに関連付けます
- 直接 - autofs を特定のファイルシステムに関連付けます
- 間接 - autofs を参照指向ファイルシステムに関連付けます

autofs 環境に変更を加える方法はネームサービスによって異なります。ローカルファイルをネームサービスとして使用している場合に変更を加えるには、テキストエディタを使用します。ネームサービスが NIS または LDAP の場合は、これらのネームサービスに必要な変更を加えます。

マップのタイプに加えた変更によっては、`automount` コマンドの実行が必要になる場合があります。たとえば、直接 (`direct`) マップに対する追加または削除を行なった場合、ローカルシステム上で `automount` コマンドを実行する必要があります。`automount` コマンドを実行すると、変更が反映されます。ただし、既存のエントリを変更した場合は、変更を反映するために `automount` コマンドを実行する必要はありません。マスターマップに変更を加えた場合は、`automount` コマンドを常に実行する必要があります。間接マップに変更を加えた場合は、`automount` コマンドを実行する必要はまったくありません。

## マップの変更

このセクションでは、いくつかのタイプの `automounter` マップを更新する方法について説明します。

### ▼ マスターマップを修正する方法

1. 使用しているネームサービスに基づいて、マップを変更する権限を持つユーザーとしてログインします。ローカルマップファイルを使用している場合は、`root` 役割になります。
2. マスターマップに変更を加えます。  
マップを変更するために必要な具体的な手順は、使用しているネームサービスによって異なります。ローカルファイルをネームサービスとして使用している場合はテキストエディタを使用します。ネームサービスが NIS の場合は、`make` ファイルを使用します。
3. 各クライアントで管理者になります。  
詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。
4. 各クライアントで、`automount` コマンドを実行して変更を反映します。
5. マスターマップからの新しい情報を組み込むために、ユーザーに自分のシステムでスーパーユーザーとして `automount` コマンドを実行するように通知します。

### ▼ 間接マップを修正する方法

1. マップを変更する権限を持つユーザーとしてログインします。

## 2. 間接マップに変更を加えます。

マップを変更するために必要な具体的な手順は、使用しているネームサービスによって異なります。

## ▼ 直接マップを修正する方法

### 1. マップを変更する権限を持つユーザーとしてログインします。

### 2. 直接マップに変更を加えます。

マップを変更するために必要な具体的な手順は、使用しているネームサービスによって異なります。

### 3. マップを変更したことを他のユーザーに通知します。

必要に応じてユーザーが自分のシステムでスーパーユーザーとして `automount` コマンドを実行できるようにするには、通知が必要です。

---

**注記** - 既存の直接マップエントリの内容の変更だけを行なった場合は、`automount` コマンドを実行する必要はありません。

---

たとえば、異なるサーバーから `/usr/src` ディレクトリがマウントされるように `auto_direct` マップを修正するとします。`/usr/src` がその時点でマウントされていない場合、`/usr/src` にアクセスするとすぐにその新しいエントリが反映されます。`/usr/src` がその時点でマウントされている場合、オートアンマウントが実行されるまで待ちます。その後、アクセスが可能になります。

## マウントポイント競合の回避

`/src` にマウントされたローカルディスクパーティションがあり、ほかのソースディレクトリをマウントするために `autofs` サービスを使用する予定の場合は、ローカルパーティションに到達しようとするたびに NFS サービスはそれを隠します。したがって、パーティションをどこかほかの場所にマウントする必要があります。

たとえば、`/export/src` にパーティションにマウントするには、次のように `/etc/vfstab` ファイルにエントリを追加します。

```
/dev/dsk/d0t3d0s5 /dev/rdsk/c0t3d0s5 /export/src ufs 3 yes -
```

`auto_src` にもエントリを追加する必要があります。この例では、システムの名前は `terra` です。

```
terra terra:/export/src
```

## 非 NFS ファイルシステムへのアクセス

autofs は、NFS ファイルでないファイルシステム、たとえばリムーバブルメディア上のファイル (CD-ROM や USB フラッシュドライブなど)、もマウントできます。

サーバーからファイルシステムのマウントを行う代わりに、ドライブに媒体を配置してマップから参照します。たとえば、CD-ROM アプリケーションにアクセスするには、管理者になり、次の例のような CD-ROM ファイルシステム用のエントリと、コロン後に CD-ROM デバイス名を autofs マップに追加します。

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

## オートマウントのカスタマイズ

このセクションでは、使いやすいディレクトリ構造を提供するためにオートマウントマップをカスタマイズする方法について説明します。

### /home の共通ビューの設定

すべてのネットワークユーザーが自分のホームディレクトリまたはほかのユーザーのホームディレクトリが /home で見つけられることが理想的です。このビューは、クライアントやサーバーに関係なく、すべてのシステムで共通にすることをお勧めします。

Oracle Solaris をインストールすると、常にマスターマップ /etc/auto\_master がインストールされます。

```
Master map for autofs
#
+auto_master
/net -hosts -nosuid,nobrowse
/home auto_home -nobrowse
/nfs4 -fedfs -ro,nosuid,nobrowse
```

auto\_home 用のマップも、/etc にインストールされます。新しいローカルユーザーが作成されると、エントリは自動的に /etc/auto\_home に追加されます。次に例を示します。

```
Home directory map for autofs
#
rusty dragon:/export/home/&
+auto_home
```

dragon というサーバーで、/export/home/rusty と /home/rusty 経由で rusty のホームディレクトリにアクセスできます。



auto\_home マップが設定されていれば、ユーザーはパス /home/user で任意のホームディレクトリ (自分のを含む) を参照できます。user はログイン名で、マップ内の鍵です。すべてのホームディレクトリのこの共通ビューは、ほかのユーザーのシステムにログインするときに役立ちます。autofs は、ユーザー自身のホームディレクトリをマウントします。同様に、ほかのシステム上でリモートウィンドウシステムクライアントを実行する場合は、クライアントプログラムには同じビューの /home ディレクトリが表示されます。この共通表示は、サーバーにも拡張されています。

ユーザーは、各ホームディレクトリの実際の位置を意識する必要はありません。ユーザーがより多くのディスク領域を必要としているため、ホームディレクトリをほかのサーバーに移す必要がある場合には、必要なことは auto\_home マップ内でユーザーのエントリを変更して新しい場所を反映するだけです。ほかのユーザーは /home/user パスを引き続き使用できます。

---

**注記** - ユーザーが自分のホームディレクトリから `setuid` 実行可能ファイルを実行することを許可しないでください。この制限がない場合は、任意のユーザーが任意のシステム上でスーパーユーザー権限を持つことができます。

---

## ▼ 複数のホームディレクトリファイルシステムで /home を設定する方法

### 1. 管理者になります。

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

### 2. /export/home にホームディレクトリパーティションをインストールします。

システムにいくつかのパーティションがある場合は、/export/home1、/export/home2 のように、パーティションを別個のディレクトリにインストールしてください。

### 3. auto\_home マップを更新します。

新しいユーザーアカウントを作成するときは必ず、そのユーザーのホームディレクトリの場所を auto\_home マップに入力します。マップのエントリは、次のように単純な形式にすることができます。

```
user1 system1:/export/home1/&
user2 system1:/export/home1/&
user3 system2:/export/home2/&
user4 system1:/export/home3/&
```

マップキーの代わりに &(アンパサンド) を使用しています。アンパサンドは次の例の 2 つ目の user1 の省略です。

```
user1 system1:/export/home1/user1
```

## ▼ 共通ディレクトリにプロジェクト関連ファイルを統合する方法

autofs を使用して、複数のシステムに共通のディレクトリにファイルを統合できます。共通ディレクトリ用にプロジェクト関連ファイルのディレクトリ構造を autofs マップに追加できます。このディレクトリ構造によって、ユーザーはシステム内の物理的およびハードウェア変更に関係なくプロジェクトファイルを使用できます。

### 1. 管理者になります。

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

### 2. 共通ディレクトリ用エントリを `auto_master` マップに追加します。

```
/common-dir auto_common-dir -nosuid
```

`auto_common-dir` マップによって共通ディレクトリの内容が決まります。

### 3. 作業空間に存在する可能性のある `setuid` プログラムをユーザーが実行するのを防ぐために、予防措置として `-nosuid` オプションを追加します。

### 4. `auto_common-dir` マップにエントリを追加します。

`auto_common-dir` マップが、各エントリがサブプロジェクトを表現するように編成されます。最初の操作により、マップが次のようになります。

```
project1 system1:/export/common-dir/&
project2 system1:/export/common-dir/&
app1 system2:/export/common-dir/&
```

各エントリの最後のアンパサンド (&) は、エントリ鍵を省略したものです。

#### 例 7 /ws にプロジェクト関連ファイルを統合する

大きなソフトウェア開発プロジェクトの管理者であるとします。そこで、プロジェクト関連のファイルをすべて /ws というディレクトリの下で利用できるようにすると仮定します。このようなディレクトリは、そのサイトのすべてのワークステーションで共通である必要があります。

/ws ディレクトリに対するエントリを、サイトの `auto_master` マップに追加します。

```
/ws auto_ws -nosuid
```

`auto_ws` マップが、/ws ディレクトリの内容を決定します。`-nosuid` は、作業空間に存在する可能性のある `setuid` プログラムをユーザーが実行するのを防ぎます。各エントリがサブプロジェクトを表現するように、`auto_ws` マップにエントリを追加します。最初の操作により、マップが次のようになります。

```

compiler alpha:/export/ws/&
windows alpha:/export/ws/&
files bravo:/export/ws/&
drivers alpha:/export/ws/&
man bravo:/export/ws/&
tools delta:/export/ws/&

```

各エントリの最後のアンパサンド (&) は、エントリ鍵を省略したものです。たとえば、最初のエントリは次のエントリと同じ意味です。

```
compiler alpha:/export/ws/compiler
```

この最初の試行によってシンプルに見えるマップが提供されますが、さらに調整が必要です。プロジェクトのオーガナイザーが、`man` エントリ内のドキュメントを各サブプロジェクトの下のサブディレクトリとして提供しようとしているとします。さらに、各サブプロジェクトは、ソフトウェアの複数のバージョンを記述するために、複数のサブディレクトリを必要とします。この場合、サーバー上のディスクパーティション全体に対して、これらのサブディレクトリをそれぞれ割り当てる必要があります。

次のように、マップ内のエントリを修正してください。

```

compiler \
 /vers1.0 alpha:/export/ws/&/vers1.0 \
 /vers2.0 bravo:/export/ws/&/vers2.0 \
 /man bravo:/export/ws/&/man
windows \
 /vers1.0 alpha:/export/ws/&/vers1.0 \
 /man bravo:/export/ws/&/man
files \
 /vers1.0 alpha:/export/ws/&/vers1.0 \
 /vers2.0 bravo:/export/ws/&/vers2.0 \
 /vers3.0 bravo:/export/ws/&/vers3.0 \
 /man bravo:/export/ws/&/man
drivers \
 /vers1.0 alpha:/export/ws/&/vers1.0 \
 /man bravo:/export/ws/&/man
tools \
 / delta:/export/ws/&

```

現在のマップはかなり長くなっていますが、まだ5つのエントリを含んでいるだけです。各エントリは、複数のマウントがあるために長くなっています。たとえば、`/ws/compiler` を参照するには、`vers1.0`、`vers2.0`、および `man` ディレクトリ用の3つのマウントが必要です。

各行の末尾のバックスラッシュは、そのエントリが次の行に続くことを示します。実際、エントリは1つの長い行となっていますが、行ブレイクやインデントのいくつかはエントリを読みやすくする目的で使用されています。

`tools` ディレクトリには、すべてのサブプロジェクトに対するソフトウェア開発ツールが含まれているため、同じサブディレクトリ構造の対象とはなっていません。`tools` ディレクトリは単一のマウントのままです。

この配置は、システムの管理者に大きな柔軟性を提供します。ソフトウェアプロジェクトでは、非常に大きなディスクスペースを消費します。プロジェクトのすべての過

程を通じて、さまざまなディスクパーティションを再配置し、拡張することになる可能性もあります。これらの変更が `auto_ws` マップに反映されている場合は、`/ws` のディレクトリ階層は変わらないため、ユーザーに通知する必要はありません。

サーバー `alpha` と `bravo` が同じ `autofs` マップを見るため、これらのシステムにログインするユーザーは予期したとおりに `/ws` 名前空間を見つけることができます。このようなユーザーには、NFS マウントではなく、ループバックマウントを通じてのローカルファイルへの直接アクセスが提供されます。

## ▼ 共有名前空間にアクセスするために異なるアーキテクチャーを設定する方法

ローカル実行可能ファイルやアプリケーション (スプレッドシートアプリケーションやワープロパッケージなど) のための共有名前空間を構築する必要があるとします。この名前空間のクライアントは、異なる実行可能フォーマットを必要とする複数の異なるワークステーションアーキテクチャーを使用します。また、ワークステーションには、異なるリリースのオペレーティングシステムを使用するものもあります。

### 1. `auto_local` マップを作成します。

ネームサービスの詳細は、『[Oracle Solaris 11.3 ディレクトリサービスとネームサービスでの作業: DNS と NIS](#)』を参照してください。

### 2. 共有名前空間について、サイト固有の名称を 1 つ選択します。

この名称により、その名前空間に属するファイルとディレクトリが簡単に識別できるようになります。たとえば、`/usr/local` を名前として選択した場合は、パス `/usr/local/bin` は明確にこの名前空間の一部です。

### 3. `autofs` 間接マップを作成し、それをアンカーするパス名を選択します。

このマップを `/usr/local` にマウントします。NIS の `auto_master` マップ内で、次のエントリを設定します。

```
/usr/local auto_local -ro
```

`-ro` マウントオプションは、クライアントがファイルやディレクトリに書き込めないことを意味します。

### 4. サーバー上の任意のディレクトリをエクスポートします。

### 5. `auto_local` マップ内に `bin` エントリを含めます。

ディレクトリ構造は、次のようになります。

```
bin aa:/export/local/bin
```

`aa` はサーバーの名前です。

6. (オプション) 異なるアーキテクチャーのクライアントを処理するため、**autofs CPU** 変数を加えて、エントリの変更を行います。

```
bin aa:/export/local/bin/$CPU
```

aa はサーバーの名前です。

- SPARC クライアントの場合 – 実行可能ファイルを /export/local/bin/sparc に配置します。
- x86 クライアントの場合 – 実行可能ファイルを /export/local/bin/i386 に配置します。

## ▼ 非互換のクライアントオペレーティングシステムのバージョンをサポートする方法

1. アーキテクチャータイプと、クライアントのオペレーティングシステムのタイプを決定する変数とを結合します。

autofs OSREL 変数と CPU 変数を結合して、CPU タイプと OS リリースの両方を示す名前を作成することができます。

2. 次のようなマップエントリを作成します。

```
bin aa:/export/local/bin/CPUOSREL
```

Version 5.6 を実行しているクライアントのために、次のファイルシステムをエクスポートします。

- SPARC クライアントの場合 – /export/local/bin/sparc5.6 をエクスポートします。
- x86 クライアントの場合 – 実行可能ファイルを /export/local/bin/i3865.6 に配置します。

## ▼ 複数のサーバーを通じて共用ファイルを複製する方法

読み取り専用の複製されたファイルシステムを共有する最良の方法は、フェイルオーバーの利用です。フェイルオーバーの詳細は、[41 ページの「クライアント側フェイルオーバー」](#)を参照してください。

1. 管理者になります。  
詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。
2. **autofs** マップに、すべての複製サーバーのコンマ区切りリストを作成します。

```
bin aa,bb,cc,dd:/export/local/bin/$CPU
```

autofs は、もっとも近いサーバーを選択します。サーバーが複数のネットワークインタフェースを持っている場合は、各インタフェースのリストを作成してください。autofs はクライアントにもっとも近接したインタフェースを選択し、NFS トラフィックの不必要なルーティングを避けるようにしています。

## autofs セキュリティー制限

nosuid オプションは、setuid または setgid ビットが設定されたファイルをユーザーが作成することを防ぎます。

このエントリは、汎用ローカルファイル /etc/auto\_master 内の /home のエントリをオーバーライドします。汎用ローカル /etc/auto\_master ファイルについては、[104 ページの「/home の共通ビューの設定」](#)を参照してください。これは、+auto\_master が、ファイル内の /home エントリより先に、外部のネームサービスマップを参照するためです。auto\_home マップ内のエントリにマウントオプションがある場合、nosuid オプションは無効になります。したがって、auto\_home マップ内でオプションを使用しないようにするか、nosuid オプションを各エントリに含める必要があります。

---

注記 - サーバー上の /home またはその下に、ホームディレクトリのディスクパーティションをマウントしないでください。

---

### ▼ autofs セキュリティー制限を適用する方法

1. 管理者になります。

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. ネームサービス auto\_master ファイル内に -nosuid エントリを作成します。

```
/home auto_home -nosuid
```

### ▼ autofs で公開ファイルハンドルを使用する方法

1. 管理者になります。

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. **autofs** マップ内に次のエントリを作成します。

```
/usr/local -ro,public bee:/export/share/local
```

public オプションは、公開ハンドルの使用を強制します。NFS サーバーが公開ファイルハンドルをサポートしない場合、マウントは失敗します。

## ▼ autofs で NFS URL を使用する方法

1. 管理者になります。

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. **autofs** マップ内に次のエントリを作成します。

```
/usr/local -ro nfs://server-name/export/share/local
```

サービスは、NFS サーバー上で公開ファイルハンドルの使用を試みます。サーバーが公開ファイルハンドルをサポートしない場合、MOUNT プロトコルが使用されます。

## autofs ブラウザビリティの無効化

インストールされるデフォルトバージョンの `/etc/auto_master` には、`~/home` および `/net` 用のエントリに `nobrowse` オプションが追加されています。さらに、アップグレード手順により、`/etc/auto_master` 内の `/home` および `/net` エントリに `-nobrowse` オプションが追加されます (これらのエントリが変更されていなかった場合)。ただし、このような変更を手動で加えるか、あるいはインストール後にサイト固有の `autofs` マウントポイントに対するブラウザ機能をオフにすることが必要な場合もあります。

このセクションでは、単一クライアント、すべてのクライアント、および選択したファイルシステムのブラウザビリティをオフにする方法について説明します。

## ▼ 1 つの NFS クライアントの autofs ブラウズ機能を完全に無効にする方法

1. NFS クライアント上で管理者になります。

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. **autofs** SMF 構成パラメータを変更します。

```
sharectl set -p nobrowse=TRUE autofs
```

3. autofs サービスを再起動します。

```
svcadm restart system/filesystem/autofs
```

## ▼ すべてのクライアントの autofs ブラウズ機能を無効にする方法

すべてのクライアントに対するブラウズ機能を無効にするには、NIS のようなネームサービスを使用する必要があります。それ以外の場合には、各クライアント上でオートマウントマップを手動で編集する必要があります。この例では、/home ディレクトリのブラウズ機能が無効にされています。無効にする必要がある各間接 autofs ノードに対して、この手順を実行してください。

1. ネームサービス auto\_master ファイル内の /home エントリに -nobrowse オプションを追加します。

```
/home auto_home -nobrowse
```

2. 新しい動作を有効にするために、すべてのクライアントで automount コマンドを実行します。

リブート後にも新しい動作が有効になります。

```
/usr/sbin/automount
```

## ▼ 選択したファイルシステムの autofs ブラウズ機能を無効にする方法

この例では、/net ディレクトリのブラウズ機能を無効にします。/home または他の autofs マウントポイントにも、同じ手順を使用できます。

1. 自動マウントのネームサービスの検索順序を確認します。

name-service/switch サービスの config/automount プロパティは、自動マウント情報の検索順序を示します。

```
svcprop -p config svc:/system/name-service/switch
config/value_authorization astring solaris.smf.value.name-service.switch
config/printer astring user\ files
config/default astring files\ nis
config/automount astring files\ nis
```

最後のエントリは、ローカルの自動マウントファイルが最初に検索され、次に NIS サービスが確認されることを示しています。config/default エントリは、明示的に一覧表示されていないすべての名前情報の検索順序を指定します。LDAP ネームサービスを使用している場合、name-service/switch サービスの config/automount プロパティでは、自動マウント情報の検索順序に ldap が示されます。



```
svcprop -p config svc:/system/name-service/switch
config/value_authorization astring solaris.smf.value.name-service.switch
config/printer astring user\ files\ ldap
config/default astring files\ ldap
config/automount astring files\ ldap
```

2. **/etc/auto\_master** 内で **+auto\_master** エントリの位置を確認します。

ローカルファイルが名前空間内のエントリより優先されるように、**+auto\_master** エントリを **/net** のあとに移す必要があります。

```
Master map for automounter
#
/net -hosts -nosuid
/home auto_home
/nfs4 -fedfs -ro,nosuid,nobrowse
+auto_master
```

標準構成では **+auto\_master** エントリはファイルの先頭に置かれ、ローカル変更が使用されるのを防ぎます。

3. **/etc/auto\_master** ファイル内の **/net** エントリに **nobrowse** オプションを追加します。

```
/net -hosts -nosuid,nobrowse
```

4. すべてのクライアント上で、**automount** コマンドを実行します。

新規の動作は、クライアントシステム上で **automount** コマンドを実行した後、またはリブートした後に反映されます。

```
/usr/sbin/automount
```



# ◆◆◆ 第 5 章

## ネットワークファイルシステムを管理するためのコマンド

---

この章では、ネットワークファイルシステムの管理に使用するコマンド行ユーティリティーについて説明します。

この章の内容は、次のとおりです。

- 115 ページの「NFS コマンド」
- 136 ページの「FedFS コマンド」

---

注記 - システムでゾーンが有効なときに非大域ゾーンでこの機能を使用する場合は、『[Oracle Solaris ゾーンの紹介](#)』を参照してください。

---

### NFS コマンド

これらのコマンドの効果を十分に得るには root として実行する必要がありますが、情報を要求することはすべてのユーザーができます。

- 116 ページの「[automount コマンド](#)」
- 117 ページの「[clear\\_locks コマンド](#)」
- 117 ページの「[fsstat コマンド](#)」
- 118 ページの「[mount コマンド](#)」
- 125 ページの「[mountall コマンド](#)」
- 136 ページの「[nfsref コマンド](#)」
- 126 ページの「[sharectl コマンド](#)」
- 128 ページの「[share コマンド](#)」
- 134 ページの「[shareall コマンド](#)」
- 135 ページの「[showmount コマンド](#)」
- 124 ページの「[umount コマンド](#)」
- 125 ページの「[umountall コマンド](#)」
- 134 ページの「[unshare コマンド](#)」

- [134 ページの「unshareall コマンド」](#)

また、FedFS サービスに関連したコマンドについては、[136 ページの「FedFS コマンド」](#)に説明されています。

## automount コマンド

このコマンドは autofs マウントポイントをインストールし、オートマスターファイル内の情報を各マウントポイントに関連付けます。このコマンドの構文は次のとおりです。

```
automount [-t duration] [-v]
```

-t *duration* はファイルシステムがマウントを継続できる時間 (秒) を設定し、-v は冗長モードを選択します。冗長モードでこのコマンドを実行するとトラブルシューティングが容易になります。

継続時間の値は、特に設定しないと 5 分に設定されます。通常はこの値が適切です。しかし、自動マウントされたファイルシステムの多いシステムでは、この値を増やす必要がある場合もあります。特に、サーバーを多くのユーザーが使用中の場合は、自動マウントされたファイルシステムを 5 分ごとにチェックするのは能率的でない場合があります。autofs ファイルシステムは 1800 秒 (30 分) ごとにチェックする方が適しています。5 分おきにファイルシステムマウントを解除しないと、`/etc/mnttab` が大きくなることがあります。df が `/etc/mnttab` 内の各エントリをチェックするときの出力を減らすには、-F オプション ([df\(1M\)](#) のマニュアルページを参照) または `egrep` を使用することで、df からの出力を絞り込むことができます。

この継続時間を調節すると、オートマウントマップへの変更が反映される速さを変更できるということも考慮すべきです。変更はファイルシステムがアンマウントされるまでは見ることはできません。オートマウントマップの変更方法については、[102 ページの「マップの変更」](#)を参照してください。

コマンド行で行う場合と同じ指定を `sharectl` コマンドで行うことができます。ただし、コマンド行オプションと異なり、サービス再起動、システムリブート、およびシステムアップグレードを経由しても SMF リポジトリは指定を保持します。automount コマンドには次のパラメータを設定できます。

### timeout

ファイルシステムがアンマウントされるまでアイドル状態を持続する時間を設定します。このキーワードは、automount の -t 引数と同等です。デフォルト値は 600 です。

### automount\_verbose

マウント、アンマウント、およびその他の重要でないイベントを通知します。このキーワードは、-automount の v 引数と同等です。デフォルト値は FALSE です。

## clear\_locks コマンド

このコマンドを使用すると、ある NFS クライアントのファイル、レコード、または共有のロックをすべて削除できます。このコマンドを実行するには、スーパーユーザーでなければなりません。NFS サーバーから、特定のクライアントに対するロックを解除できます。また、NFS クライアントから、特定のサーバーにおけるそのクライアントに対するロックを解除できます。次の例では、現在のシステム上の tulip という名前の NFS クライアントのロックを解除します。

```
clear_locks tulip
```

-s オプションは、どの NFS ホストからロックを解除するかを指定するために使用します。このオプションは、ロックを作成した NFS クライアントで実行する必要があります。この状況では、クライアントからのロックが bee という名前の NFS サーバーから解除されます。

```
clear_locks -s bee
```



**注意** - このコマンドは、クライアントがクラッシュしてロックを解除できないときだけに実行するようにしてください。データが破壊されるのを避けるため、使用中のクライアントに関するロックは解除しないでください。

## fsstat コマンド

fsstat ユーティリティを使用して、ファイルシステムの種類およびマウントポイントごとに、ファイルシステムオペレーションをモニターできます。さまざまなオプションで出力をカスタマイズできます。

-i                   マウントポイントの入出力操作に関する統計を表示します

-n                   マウントポイントの命名操作に関する統計を表示します

この例では、NFS Version 3、NFS Version 4、およびルートマウントポイントの出力を示しています。

```
% fsstat nfs3 nfs4 /
new name name attr attr lookup rddir read read write write
file remov chng get set ops ops ops bytes ops bytes
3.81K 90 3.65K 5.89M 11.9K 35.5M 26.6K 109K 118M 35.0K 8.16G nfs3
759 503 457 93.6K 1.44K 454K 8.82K 65.4K 827M 292 223K nfs4
25.2K 18.1K 1.12K 54.7M 1017 259M 1.76M 22.4M 20.1G 1.43M 3.77G /
```

次の例では、-i オプションを使って NFS Version 3、NFS Version 4、およびルートマウントポイントの入出力操作に関する統計を提供しています。

```
% fsstat -i nfs3 nfs4 /
read read write write rddir rddir rwlock rwlock
ops bytes ops bytes ops bytes ops ops
```

```

109K 118M 35.0K 8.16G 26.6K 4.45M 170K 170K nfs3
65.4K 827M 292 223K 8.82K 2.62M 74.1K 74.1K nfs4
22.4M 20.1G 1.43M 3.77G 1.76M 3.29G 25.5M 25.5M /

```

次の例では、`-n` オプションを使って NFS Version 3、NFS Version 4、およびルートマウントポイントの命名操作に関する統計を提供しています。

```

% fsstat -n nfs3 nfs4 /
lookup creat remov link renam mkdir rmdir rddir symlnk rdlnk
35.5M 3.79K 90 2 3.64K 5 0 26.6K 11 136K nfs3
454K 403 503 0 101 0 0 8.82K 356 1.20K nfs4
259M 25.2K 18.1K 114 1017 10 2 1.76M 12 8.23M /

```

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

## mount コマンド

このコマンドを使用すると、指定したファイルシステムをローカルまたはリモートで、指定したマウントポイントにマウントできます。詳細は、[mount\(1M\)](#) のマニュアルページを参照してください。引数を指定しないで `mount` を使用すると、現在コンピュータにマウントされているファイルシステムのリストが表示されます。

Oracle Solaris 標準インストールに含まれるファイルシステムタイプごとに、固有の `mount` コマンド用オプションが用意されています。NFS ファイルシステムオプションについては、[mount\\_nfs\(1M\)](#) のマニュアルページを参照してください。UFS ファイルシステムオプションについては、[mount\\_ufs\(1M\)](#) のマニュアルページを参照してください。

標準 `server:/pathname` 構文の代わりに NFS URL を使用することで、NFS サーバーからマウントするパス名を選択できます。詳細は、[83 ページの「NFS URL を使用して NFS ファイルシステムをマウントする方法」](#) を参照してください。



**注意** - `mount` コマンドは無効なオプションについて警告しません。解釈できないオプションがあると無視されるだけです。予想外の結果が生じるのを避けるために、使用するオプションはすべて確認してください。

## NFS ファイルシステム用の mount オプション

このセクションでは、NFS ファイルシステムマウント時に `-o` フラグのあとに指定できるいくつかのオプションについて説明します。オプションの完全なリストについては、[mount\\_nfs\(1M\)](#) のマニュアルページを参照してください。

`bg|fg`

これらのオプションは、マウントが失敗した場合の再試行動作を選択するために使用できます。bg オプションの場合はバックグラウンドで、fg オプションでは、

フォアグラウンドでマウント試行が実行されます。デフォルトは `fg` で、マウントが完了するまで後続の処理を抑止するため、利用可能である必要があるファイルシステムにとって最適な選択です。`bg` は、マウント要求の完了を待機中にクライアントはほかの処理を実行できるため、クリティカルでないファイルシステムに適した選択です。

#### `forcedirectio`

このオプションは、大きなシーケンシャルデータ転送のパフォーマンスを改善します。データは直接ユーザーバッファにコピーされます。クライアント上のカーネル内ではキャッシュへの書き込みは行われません。このオプションはデフォルトでオフになっています (`noforcedirectio`)。

アプリケーションがクライアント上の単一ファイルへの並行書き込みおよび並行読み取り/書き込みを発行することを許可するには、`forcedirectio` マウントオプションを使用します。このオプションによって、マウントされたファイルシステム内のすべてのファイルでこの機能が有効になります。この機能をクライアント上の単一ファイルで有効にするするために、`directio()` インタフェースを使用することもできます。この機能を有効にしないかぎり、ファイルへの書き込みは直列化されます。また、並行書き込みや並行読み取り/書き込みが実行されると、そのファイルに関しては、POSIX のセマンティクスはサポートされなくなります。

このオプションの使用例については、[121 ページの「mount コマンドの使用」](#)を参照してください。

#### `largefiles`

このオプションで、2G バイトより大きいファイルにアクセスできます。大きいファイルにアクセスできるかどうかはサーバーでのみ制御できるため、このオプションは NFS Version 3 マウントでサイレントに無視されます。デフォルトでは、UFS ファイルシステムはすべて `largefiles` オプション付きでマウントされます。NFS Version 2 プロトコルを使用するマウントでの `largefiles` オプションは、マウントがエラーで失敗します。

#### `nolargefiles`

UFS マウント用のこのオプションによって、ファイルシステム上に大きなファイルが存在できないことが保証されます。大きなファイルの存在は NFS サーバー上でのみ制御できるため、NFS マウントを使用しているときに `nolargefiles` 用のオプションは存在しません。このオプションを指定してファイルシステムを NFS マウントしようとする、エラーが発生して拒否されます。

#### `nosuid|suid`

`nosuid` オプションは、`nodevices` オプションを `nosetuid` オプションとともに指定することと同等です。`nodevices` オプションが指定されている場合、マウントされたファイルシステム上のデバイス特殊ファイルを開くことができません。`nosetuid` オプションが指定されている場合、ファイルシステム上に置かれたバイナリファイルの `setuid` ビットと `setgid` ビットは無視されます。プロセスは、バイナリファイルを実行するユーザーの特権で実行します。

suid オプションは、devices オプションを setuid オプションと同時に指定することと同等です。devices オプションが指定されている場合、マウントされたファイルシステムのデバイス特殊ファイルを開くことができます。setuid オプションが指定されている場合、ファイルシステムに置かれたバイナリファイルの setuid ビットと setgid ビットは、カーネルが引き受けます。

いずれのオプションも指定されていない場合、デフォルトのオプションは suid になります。これにより、devices オプションを setuid オプションと同時に指定するデフォルトの動作になります。

次の表で、nosuid または suid を devices または nodevices、および setuid または nosetuid と組み合わせた効果について説明します。オプションの各組み合わせでは、もっとも制限の高いオプションが動作を決定します。

| オプションの組み合わせによる動作              | オプション  | オプション    | オプション     |
|-------------------------------|--------|----------|-----------|
| nosetuid と nodevices の同時指定と同等 | nosuid | nosetuid | nodevices |
| nosetuid と nodevices の同時指定と同等 | nosuid | nosetuid | devices   |
| nosetuid と nodevices の同時指定と同等 | nosuid | setuid   | nodevices |
| nosetuid と nodevices の同時指定と同等 | nosuid | setuid   | devices   |
| nosetuid と nodevices の同時指定と同等 | suid   | nosetuid | nodevices |
| nosetuid と devices の同時指定と同等   | suid   | nosetuid | devices   |
| setuid と nodevices の同時指定と同等   | suid   | setuid   | nodevices |
| setuid と devices の同時指定と同等     | suid   | setuid   | devices   |

nosuid オプションを指定すると、信頼できないサーバーにアクセスする可能性のある NFS クライアントのセキュリティーを向上できます。このオプションでリモートファイルシステムをマウントすることで、信頼できないデバイスのインポートまたは信頼できない setuid バイナリファイルのインポートによって特権がエスカレーションする危険が少なくなります。これらのすべてのオプションは、すべての Oracle Solaris ファイルシステムで使用可能です。



**public**

このオプションによって、NFS サーバーに接続するときに公開ファイルハンドルが強制的に使用されます。NFS サーバーが公開ファイルハンドルをサポートしていれば、MOUNT プロトコルが使用されないため、マウント操作は短時間で行われます。また、MOUNT プロトコルを使用しないため、ファイアウォールを越えたマウントが可能です。

**rw|ro**

**-rw** オプションと **-ro** オプションは、ファイルシステムが読み書きまたは読み取り専用のどちらでマウントされるかを示します。デフォルトは読み書き可能で、これはリモートホームディレクトリやメールプールディレクトリなどの、ユーザーによる変更が必要なファイルシステムに適しています。読み取り専用オプションは、ユーザーが変更してはいけないディレクトリに適しています。具体的には、マニュアルページの共有コピーなどです。

**sec=mode**

このオプションは、マウント時に使用される認証メカニズムを指定します。*mode* に使用可能な値は次のとおりです。

- **krb5**、Kerberos Version 5 認証サービスの場合
- **krb5i**、完全性付き Kerberos Version 5 の場合
- **krb5p**、プライバシー付き Kerberos Version 5 の場合
- **none**、認証なしの場合
- **dh**、Diffie-Hellman (DH) 認証の場合
- **sys**、UNIX 標準認証の場合

モードは、`/etc/nfssec.conf` ファイルにも定義されます。

**soft|hard**

**soft** オプションでマウントされた NFS ファイルシステムは、サーバーが応答しない場合にエラーを返します。**hard** オプションが指定されていると、サーバーが応答するまで続けて再試行が行われます。デフォルトは **hard** です。ほとんどのファイルシステムには **hard** を使用します。ソフトマウントされたファイルシステムからの値を検査しないアプリケーションが多いので、アプリケーションでエラーが発生してファイルが破壊される恐れがあるためです。アプリケーションが戻り値をチェックする場合でも、ルーティング問題やほかの状況によってアプリケーションを混乱させたりファイルが破損したりすることがあります。原則として、**soft** は使用しないでください。**hard** オプションを指定した場合にファイルシステムが使用できなくなると、そのファイルシステムを使用するアプリケーションはファイルシステムが復旧するまでハングアップする可能性があります。

## mount コマンドの使用

次の例は別のシナリオを示しています。

- NFS Version 2 または NFS Version 3 では、次のコマンドはどちらもサーバー `bee` から NFS ファイルシステムを読み取り専用でマウントします。

```
mount -F nfs -r bee:/export/share/man /usr/man
```

```
mount -F nfs -o ro bee:/export/share/man /usr/man
```

NFS Version 4 では、次のコマンド行が同じマウントを実現します。

```
mount -F nfs -o vers=4 -r bee:/export/share/man /usr/man
```

- NFS Version 2 または NFS Version 3 では、次のコマンドの `-o` オプションによって、強制的にマニュアルページがサーバー `bee` からローカルシステムにマウントされます (すでに `/usr/man` がマウントされている場合でも)。

```
mount -F nfs -o bee:/export/share/man /usr/man
```

NFS Version 4 では、次のコマンドが同じマウントを実現します。

```
mount -F nfs -o vers=4 -o bee:/export/share/man /usr/man
```

- NFS Version 2 または NFS Version 3 では、次のコマンドがクライアントフェイルオーバーを使用します。

```
mount -F nfs -r bee,wasp:/export/share/man /usr/man
```

NFS Version 4 では、次のコマンドがクライアントフェイルオーバーを使用します。

```
mount -F nfs -o vers=4 -r bee,wasp:/export/share/man /usr/man
```

---

**注記** - コマンド行から使用する場合、リスト内のサーバーがサポートしている NFS プロトコルは同じバージョンでなければなりません。コマンド行から `mount` を実行するときは、NFS Version 2 と NFS Version 3 サーバーの両方を使用しないでください。autofs が NFS Version 2 または NFS Version 3 サーバーの最適のサブセットを自動的に選択するため、autofs で両方のサーバーを使用できます。

---

- 次の例は、NFS Version 2 または Version 3 で、NFS URL を `mount` コマンドで使用する方法を示しています。

```
mount -F nfs nfs://bee:/export/share/man /usr/man
```

次の例は、NFS Version 4 で、NFS URL を `mount` コマンドで使用する方法を示しています。

```
mount -F nfs -o vers=4 nfs://bee:/export/share/man /usr/man
```

- 次の例は、`quota` マウントオプションを使用して、ユーザーがファイルシステムのディスク割り当て制限と使用量を確認できるようにする方法を示しています。

```
mount -F nfs -o quota bee:/export/share/man /usr/man
```

- 次の例は、`forcedirectio` マウントオプションを使用して、クライアントにファイルへの並行書き込みとファイルへの並行読み取り/書き込みを許可する方法を示しています。

```
mount -F nfs -o forcedirectio bee:/home/somebody /mnt
```

この例では、サーバー `bee` からの NFS ファイルシステムがマウントされ、ディレクトリ `/mnt` にあるファイルごとに並行読み取り/書き込みが有効になります。並行読み取り/書き込みのサポートを有効にすると、次のことが発生します。

- クライアントは、ファイルへの並列した書き込みをアプリケーションに許可します。
- クライアントでのキャッシュが無効になります。その結果、読み取りと書き込みのデータはサーバー上に保持されます。つまり、クライアントは読み取られたデータまたは書き込まれたデータをキャッシュに書き込まないため、アプリケーションがキャッシュに書き込んでいないデータはサーバーから読み取られます。クライアントのオペレーティングシステムは、このデータのコピーを持ちません。通常、NFS クライアントは、アプリケーションが使用するカーネルにデータをキャッシュします。

クライアント側でキャッシュが無効になっているため、先読みと後書きプロセスが無効になります。先読みプロセスは、アプリケーションが次に要求する可能性のあるデータをカーネルが予測したときに、発生します。次に、カーネルはあらかじめデータを収集するプロセスを開始します。カーネルの目標は、アプリケーションがデータを要求する前にそのデータを準備しておくことです。

クライアントは、書き込みのスループットを向上する後書きプロセスを使用します。アプリケーションがデータをファイルに書き込むたびに、入出力操作をただちに開始する代わりに、データはメモリー内にキャッシュされます。のちに、データはディスクに書き込まれます。

後書きプロセスにより、データがより大きな領域に書き込まれたり、アプリケーションから非同期で書き込まれたりする可能性があります。通常、より大きな領域を使用するとスループットが向上します。非同期の書き込みにより、アプリケーション処理と入出力処理間でオーバーラップができるようになります。また、ストレージサブシステムが、より優れた入出力処理を行うことで入出力を最適化できるようにもなります。同期の書き込みは、最適化されていないストレージサブシステムでの入出力を強制的に処理します。

- アプリケーションでキャッシュされていないデータのセマンティクスを処理する準備ができていない場合、著しくパフォーマンスが低下する可能性があります。マルチスレッド化されたアプリケーションは、この問題を回避します。

---

**注記** - 並行書き込みのサポートが有効にされていない場合、すべての書き込み要求は直列化されます。ある書き込み要求が進行中のとき、2 番目の書き込み要求は、最初の処理が完了するのを待ってから処理を続行する必要があります。

---

- 次の例は、`mount` コマンドを引数なしで使用して、クライアントにマウントされたファイルシステムを表示する方法を示しています。

```
% mount
/ on /dev/dsk/c0t3d0s0 read/write/setuid on Wed Apr 7 13:20:47 2004
/usr on /dev/dsk/c0t3d0s6 read/write/setuid on Wed Apr 7 13:20:47 20041995
/proc on /proc read/write/setuid on Wed Apr 7 13:20:47 2004
/dev/fd on fd read/write/setuid on Wed Apr 7 13:20:47 2004
/tmp on swap read/write on Wed Apr 7 13:20:51 2004
/opt on /dev/dsk/c0t3d0s5 setuid/read/write on Wed Apr 7 13:20:51 20041995
/home/kathys on bee:/export/home/bee7/kathys
intr/noquota/nosuid/remote on Wed Apr 24 13:22:13 2004
```

## umount コマンド

`umount` コマンドにより、現在マウントされているリモートファイルシステムを削除できます。`umount` コマンドには次のオプションを使用できます。

- v                    テストを有効にします
- a                    いくつかのファイルシステムを一度にアンマウントします。-a オプションに *mount-points* を指定すると、そのファイルシステムがアンマウントされます。マウントポイントが含まれない場合は、`/etc/mnttab` にリストされているすべてのファイルシステムのアンマウントが試行されます(`/`、`/usr`、`/var`、`/proc`、`/dev/fd`、`/tmp` など、「必要な」ファイルシステムを除く)。このファイルシステムはすでにマウントされていて `/etc/mnttab` にエントリがあるはずなので、そのファイルシステムタイプ用のフラグを含める必要はありません。
- f                    ビジーファイルシステムを強制的にアンマウントします。このオプションを使用して、マウントできないファイルシステムのマウントを試みた最中にハングアップしたクライアントを復帰させることが可能です。



**注意** - ファイルシステムを強制的にアンマウントすることで、ファイルとそのシステムに書き込まれている場合にデータを失うことがあります。

**例 8**                    ファイルシステムをアンマウントする

次の例では、`/usr/man` にマウントされたファイルシステムをアンマウントします。

```
umount /usr/man
```

**例 9** `umount` でオプションを使用する

次の例では、`umount -a -V` の実行結果が表示されます。

```
umount -a -V
umount /home/kathys
umount /opt
umount /home
umount /net
```

このコマンドでは、実際にはファイルシステムはアンマウントされません。

## `mountall` コマンド

`mountall` コマンドは、ファイルシステムテーブルにリストされているすべてのファイルシステムまたは特定のファイルシステムグループをマウントするために使用します。このコマンドには次のオプションが用意されています。

- `-F FSType`            アクセスするファイルシステムタイプを選択します。
- `-r`                    ファイルシステムテーブルにリストされているすべてのリモートファイルシステムを選択します
- `-l`                    すべてのローカルファイルシステムを選択します

NFS ファイルシステムタイプに分類されるすべてのファイルシステムはリモートファイルシステムなので、これらのオプションの一部は冗長です。詳細は、[mountall\(1M\)](#) のマニュアルページを参照してください。

次の 2 つのユーザー入力例は同等です。

```
mountall -F nfs
mountall -F nfs -r
```

## `umountall` コマンド

`umountall` コマンドは、ファイルシステムのグループをアンマウントするために使用します。`umountall` コマンドには次のオプションを使用できます。

- `-k`                    `mount-point` に関連付けられているプロセスを強制終了するために `fuser -k mount-point` コマンドを実行します。
- `-s`                    アンマウントが並行して実行されないことを示します

- l ローカルファイルシステムだけが使用されることを指定します
- r リモートファイルシステムだけが使用されることを指定します
- h *host* 指定されたホストからすべてのシステムファイルがアンマウントされるべきであることを指定します。-h オプションは、-l または -r と同時に指定できません。

次の例では、リモートホストからマウントされたすべてのファイルシステムをアンマウントします。

```
umountall -r
```

次の例では、bee サーバーから現在マウントされているすべてのファイルシステムをアンマウントします。

```
umountall -h bee
```

## sharect1 コマンド

このリリースには、NFS などのファイル共有プロトコルを構成および管理できる管理ツールである、sharect1 ユーティリティーが含まれています。このコマンドを使用して次のことを実行できます。

- クライアントとサーバーの動作プロパティを設定する
- 特定のプロトコルのプロパティ値を表示する
- プロトコルのステータスを取得する

sharect1 ユーティリティーは次の構文を使用します。

```
sharect1 subcommand [option] [protocol]
```

sharect1 ユーティリティーは次のサブコマンドをサポートしています。

- set ファイル共有プロトコルのプロパティを定義します。プロパティとプロパティ値の一覧については、[nfs\(4\)](#) のマニュアルページで説明されているパラメータを参照してください。
- get 指定されたプロトコルのプロパティとプロパティ値を表示します。
- status 指定されたプロトコルが有効か無効かを表示します。プロトコルが指定されていない場合は、すべてのファイル共有プロトコルのステータスが表示されます。

sharect1 ユーティリティーの詳細については、次を参照してください。

- [sharect1\(1M\)](#) のマニュアルページ

- 127 ページの「set サブコマンド」
- 127 ページの「get サブコマンド」
- 128 ページの「status サブコマンド」

## set サブコマンド

ファイル共有プロトコルのプロパティを定義する set サブコマンドは、次のオプションをサポートしています。

-h                    オンラインヘルプの説明を提供します。

-p                    プロトコルのプロパティを定義します。

set サブコマンドは次の構文を使用します。

```
sharectl set [-h] [-p property=value] protocol
```

set サブコマンドを使用するには root の特権が必要です。

このコマンドを追加プロパティ値ごとに繰り返す必要はありません。同じコマンドで -p オプションを複数回使用して複数のプロパティを定義できます。

次の例は、クライアントの NFS プロトコルの最小バージョンを 3 に設定します。

```
sharectl set -p client_versmin=3 nfs
```

次の例は、AUTH\_SYS で共有されているすべてのファイルシステムに対する NFS 呼び出しに予約済みポートを使用するよう、クライアントに要求する方法を示しています。

```
sharectl set -p resvport=on nfs
```

## get サブコマンド

指定されたプロトコルのプロパティとプロパティ値を表示する get サブコマンドは、次のオプションをサポートしています。

-h                    オンラインヘルプの説明を提供します。

-p                    指定されたプロパティのプロパティ値を特定します。-p オプションが使用されていない場合は、すべてのプロパティ値が表示されます。

get サブコマンドは次の構文を使用します。

```
sharectl get [-h] [-p property] protocol
```

get サブコマンドを使用するには root の特権が必要です。

次の例は `servers` を使用します。これは、同時に処理できる NFS 要求の最大数を指定できるプロパティです。

```
sharectl get -p servers nfs
servers=1024
```

次の例では、`-p` オプションが使用されていないため、すべてのプロパティ値が表示されます。

```
sharectl get nfs
servers=1024
listen_backlog=32
protocol=ALL
servers=32
lockd_listen_backlog=32
lockd_servers=20
lockd_retransmit_timeout=5
grace_period=90
nfsmapid_domain=example.com
server_versmin=2
server_versmax=4
client_versmin=2
client_versmax=4
server_delegation=on
max_connections=-1
device=
```

## status サブコマンド

`status` サブコマンドは、指定されたプロトコルが有効か無効かを表示します。オンラインヘルプの説明を提供する、`-h` オプションをサポートします。

`status` サブコマンドは次の構文を使用します。

```
sharectl status [-h] [protocol]
```

次の例は、NFS プロトコルのステータスを表示します。

```
sharectl status nfs
nfs enabled
```

## share コマンド

`share` コマンドは、NFS サーバー上のローカルファイルシステムをマウント可能にするために使用します。`share` コマンドを使用して、システム上のファイルシステムのうち、現在共有されているもののリストを表示することもできます。`share` コマンドが機能するには、NFS サーバーが動作して必要があります。

すべてのディレクトリツリーは共有できるオブジェクトです。ただし、各ファイルシステムの階層構造は、そのファイルシステムが位置するディスクスライスやパーティションで制限されます。



すでに共有している大規模なファイルシステムの一部であるファイルシステムを共有することはできません。たとえば、`/usr` および `/usr/local` が 1 つのディスクスライス上にある場合は、`/usr` を共有したり `/usr/local` を共有したりできます。ただし、異なる共有オプションを指定してこれら両方のディレクトリを共有するには、`/usr/local` を別のディスクスライスに移動する必要があります。

読み取り専用で共有されているファイルシステムへのアクセスを、読み書きで共有されているファイルシステムのファイルハンドル経由で取得できます。ただし、両方のファイルシステムが同じディスクスライスにある必要があります。より安全な状況を作成するには、読み書きである必要があるファイルシステムを、読み取り専用として共有されているファイルシステムとは別のパーティションまたは別のディスクスライスに配置してください。

---

**注記** - ファイルシステムが共有解除されてから再共有される時、NFS Version 4 がどのように機能するかについては、[29 ページの「NFS Version 4 におけるファイルシステムの共有解除と再共有」](#)を参照してください。

---

## share オプション

`-o` フラグと一緒に含めることができるオプションの一部を次に示します。

`rw|ro`

`pathname` ファイルシステムが、すべてのクライアントに読み書きまたは読み取り専用で共有されます。

`rw=access-list`

ファイルシステムが、リストされたクライアントにのみ読み書きで共有されます。それ以外の要求は拒否されます。詳細については、[132 ページの「share コマンドによるアクセスリストの設定」](#)を参照してください。このオプションは `-ro` オプションをオーバーライドします。

## NFS 固有 share オプション

NFS ファイルシステムで使用できるオプションには次が含まれます。

`aclok`

このオプションによって、NFS Version 2 プロトコルをサポートする NFS サーバーが NFS Version 2 クライアントのアクセス制御を行うように構成できます。このオプションを指定しないと、すべてのクライアントは最小限のアクセスしかできません。指定すると、最大限のアクセスができるようになります。たとえば `aclok` オプションを指定して共有したファイルシステムでは、1 人のユーザーが読み取り権を持っていれば全員が読み取りを許可されます。このオプションを指定し

ないと、アクセス権を持つべきクライアントからのアクセスが拒否される可能性があります。ユーザーに与えるアクセス権は、既存のセキュリティーシステムによって決定します。アクセス制御リスト (ACL) の詳細は、『Oracle Solaris 11.3 でのファイルのセキュリティー保護とファイル整合性の検証』の「アクセス制御リストによる UFS ファイルの保護」を参照してください。

---

**注記** - ACL を使用するには、クライアントとサーバーが NFS Version 3 および NFS\_ACL プロトコルをサポートするソフトウェアを実行していることを確認してください。ソフトウェアが NFS Version 3 プロトコルのみをサポートする場合でも、クライアントは正しいアクセスを取得できますが、ACL を操作できません。NFS\_ACL プロトコルをサポートしていれば、正しいアクセス権を取得した上で ACL の操作も可能です。

---

#### anon=uid

anon は、認証されていないユーザーのユーザー ID を選択するために使用します。anon を -1 に設定すると、サーバーは認証されていないユーザーへのアクセスを拒否します。anon=0 を設定して root アクセスを付与することで、認証されていないユーザーに root アクセスを付与できるため、代わりに root オプションを使用してください。

#### index=filename

index=filename オプションによって、ユーザーが NFS URL にアクセスするとき、ディレクトリのリストが表示される代わりに、HTML ファイルが強制的にロードされます。これは、HTTP URL がアクセスしているディレクトリに index.html ファイルが見つかったらブラウザのような動作をするというものです。このオプションを設定することは、httpd に対して DirectoryIndex オプションを指定するのと同じ意味です。たとえば、share コマンドが次を報告するとします。

```
export_web /export/web nfs sec=sys,public,index=index.html,ro
```

このとき、次の URL によって表示される情報はすべて同じです。

```
nfs://server/dir
nfs://server/dir/index.html
nfs://server//export/web/dir
nfs://server//export/web/dir/index.html
http://server/dir
http://server/dir/index.html
```

#### log=tag

このオプションによって、/etc/nfs/nfslog.conf 内に、ファイルシステム用の NFS サーバーロギング構成情報を含むタグを指定します。NFS サーバーログ機能を使用可能にするにはこのオプションを選択する必要があります。

#### nosuid

このオプションによって、setuid または setgid モードを有効にするすべての試行が無視されるべきであることを示します。NFS クライアントは、setuid または setgid ビットがオンのファイルを作成できません。

**public**

**public** オプションは、WebNFS ブラウズのために追加されました。このオプションで共有できるのは、1 台のサーバーにつき 1 つのファイルシステムだけです。

**root=access-list**

サーバーがリスト内のホストに **root** アクセスを付与します。デフォルトでは、サーバーはどのリモートホストにもルートアクセス権は与えません。選択されているセキュリティーモードが **sec=sys** 以外のものである場合は、リストにクライアントホスト名のみを含めることができます。詳細については、[132 ページの「share コマンドによるアクセスリストの設定」](#)を参照してください。



**注意** - ほかのホストに **root** アクセスを付与することは、さまざまなセキュリティー上の影響があります。**-root=** オプションは十分慎重に使用してください。

**root=client-name**

**client-name** 値は、AUTH\_SYS 認証で、**exportfs(1B)** で提供されるアドレスのリストに基づいてクライアントの IP アドレスをチェックするために使用されます。一致が見つかった場合、共有されているファイルシステムに **root** アクセスが付与されます。

**root=hostname**

AUTH\_SYS や RPCSEC\_GSS などのセキュア NFS モードの場合、サーバーは、アクセスリストから派生したホストベース主体名のリストに基づいてクライアントの主体名をチェックします。クライアントの主体名の汎用構文は **root@hostname** です。Kerberos V の場合、構文は **root/hostname.fully.qualified@REALM** です。**hostname** 値を使用するときは、アクセスリスト上のクライアントは主体名用の資格を持っている必要があります。Kerberos V の場合、クライアントはその **root/hostname.fully.qualified@REALM** 主体名用の有効な **keytab** エントリを持っている必要があります。詳細は、『[Oracle Solaris 11.3 での Kerberos およびその他の認証サービスの管理](#)』の「[Kerberos クライアントの構成](#)」を参照してください。

**sec=mode[:mode]**

このオプションによって、ファイルシステムへのアクセスを取得するために必要なセキュリティーモードを設定します。デフォルトのセキュリティーモードは、UNIX の認証です。モードは複数指定できますが、コマンド行に指定するときは 1 行につき 1 つのセキュリティーモードだけにしてください。各 **sec=** オプションは、別の **sec=** が検出されるまで、後続の **rw**、**ro**、**rw=**、**ro=**、**root=**、および **window=** オプションに適用されます。**-sec=none** とすると、すべてのユーザーがユーザー **nobody** にマップされます。

**window=value**

**value** は、NFS サーバーで資格が有効な時間の上限です。デフォルトは 30000 秒 (8.3 時間) です。

### resvport

このオプションは、個々のファイルシステムに対して予約済みポートの使用を強制します。

次の例は、AUTH\_SYS で共有されている NFS 共有に予約済みポートを使用するよう、クライアントに要求する方法を示しています。

```
share -F NFS -o resvport=on /usr/src
```

## share コマンドによるアクセスリストの設定

share コマンドで提供するアクセスリストには、ドメイン名、サブネット番号、およびアクセスを拒否するエントリ、さらに標準の `-ro=`、`-rw=`、または `-root=` オプションを含めることができます。この拡張により、名前空間を変更したり多数のクライアントを定義したリストを使用することなく、ファイルアクセス制御を単一のサーバーで簡単に管理できます。

次の例は、ほとんどのシステムに読み取り専用アクセスを提供しますが、`rose` と `lilac` には読み書きアクセスを許可します。

```
share -F nfs -o ro,rw=rose:lilac /usr/src
```

次の例では、`eng` ネットグループ内のホストに読み取り専用アクセスを割り当てます。`rose` クライアントでは、読み取りと書き込みの両方ができます。

```
share -F nfs -o ro=eng,rw=rose /usr/src
```

---

**注記** - 引数なしで `rw` と `ro` の両方を指定できません。読み書き可能オプションを指定しないと、デフォルトによってすべてのクライアントが読み書き可能になります。

---

複数のクライアントが1つのファイルシステムを共有するには、同じ行にすべてのオプションを入力する必要があります。同じオブジェクトに対して `share` コマンドの複数呼び出しを発行する場合、最後に実行されるコマンドだけが適用されます。次の例では、3つのクライアントシステムで読み書きアクセスが有効になりますが、`rose` と `tulip` だけにはファイルシステムへのアクセスが `root` として付与されます。

```
share -F nfs -o rw=rose:lilac:tulip,root=rose:tulip /usr/src
```

複数の認証メカニズムを使用するファイルシステムを共有するときは、正しいセキュリティモードのあとに `-ro`、`-ro=`、`-rw`、`-rw=`、`-root`、および `-window` オプションを含めていることを確認してください。この例では、`eng` というネットグループ内のすべてのホストに対して UNIX 認証が選択されています。これらのホストは、ファイルシステムを読み取り専用モードでのみマウントできます。ホスト `tulip` と `lilac` は、これらのホストが Diffie-Hellman 認証を使用する場合にファイルシステムを読み書きでマウントできます。これらのオプションでは、`tulip` および `lilac` は、これらのホストが DH 認証を使用していない場合でも、ファイルシステムを読み取り専用で

マウントできます。ただし、ホスト名が `eng` ネットグループに含まれている必要があります。

```
share -F nfs -o sec=dh,rw=tulip:lilac,sec=sys,ro=eng /usr/src
```

デフォルトのセキュリティーモードは UNIX 認証ですが、`-sec` オプションを使用している場合、この UNIX 認証は含まれなくなります。そのため、UNIX 認証をほかの認証メカニズムとともに使用する場合は、`-sec=sys` オプションを指定する必要があります。

実際のドメイン名の前にドットを付けると、アクセスリスト中で DNS ドメイン名を使用できます。ドットの後の文字列はドメイン名です。完全指定のホスト名ではありません。次の例では、`eng.example.com` ドメイン内のすべてのホストにマウントアクセスを許可します。

```
share -F nfs -o ro=.:eng.example.com /export/share/man
```

この例では、単一ドットは NIS 名前空間で照合されたすべてのホストに一致します。ネームサービスから返される結果にはドメイン名は含まれません。`.eng.example.com` エントリは、名前空間解決に DNS を使用するすべてのホストに一致します。DNS は常に完全修飾ホスト名を返すため、DNS とほかの名前空間を組み合わせを使用する場合は長いエントリが必要です。

実際のネットワーク番号またはネットワーク名とアットマーク (@) を後置することで、アクセスリスト内でサブネット番号を使用できます。この文字は、ネットワーク名をネットグループ名や完全指定のホスト名と区別するためです。サブネットは、`/etc/networks` 内または NIS 名前空間内で識別する必要があります。次のエントリは、サブネット `192.168` が `eng` ネットワークと識別されている場合、すべて同じ意味を持ちます。

```
share -F nfs -o ro=@eng /export/share/man
share -F nfs -o ro=@192.168 /export/share/man
share -F nfs -o ro=@192.168.0.0 /export/share/man
```

2 番目と 3 番目のエントリは、ネットワークアドレス全体を指定する必要がないことを表しています。

ネットワークアドレスの先頭部分がバイトによる区切りでなく、CIDR (Classless Inter-Domain Routing) のようになっている場合には、マスクの長さをコマンド行で具体的に指定できます。この長さは、ネットワーク名かネットワーク番号の後ろにスラッシュで区切ってアドレスの接頭辞に有効ビット数として指定します。次に例を示します。

```
share -f nfs -o ro=@eng/17 /export/share/man
share -F nfs -o ro=@192.168.0/17 /export/share/man
```

この例で、「/17」はアドレスの先頭から 17 ビットがマスクとして使用されることを表します。CIDR についての追加情報は、RFC 1519 を参照してください。

また、エントリの前に「-」を指定することでアクセスの拒否を示すこともできます。エントリは左から右に読み込まれるため、アクセス拒否のエントリは次のようにそのエントリを適用するエントリの前に置く必要があることに注意してください。

```
share -F nfs -o ro=-rose:.eng.example.com /export/share/man
```

この例では、eng.example.com ドメイン内のホストにアクセスを許可します (rose という名前のホストを除く)。

## unshare コマンド

unshare は、それまで利用可能であったファイルシステムをクライアントがマウントできなくするコマンドです。NFS ファイルシステムの共有を解除している場合、クライアントから既存マウントへのアクセスは禁止されます。ファイルシステムはクライアントでまだマウントされている可能性があります、ファイルにはアクセスできません。-t オプションを使用してファイルシステムの共有を一時的に解除する場合を除き、unshare コマンドは共有を恒久的に削除します。

---

**注記** - ファイルシステムが共有解除されてから再共有される時、NFS Version 4 がどのように機能するかについては、[29 ページの「NFS Version 4 におけるファイルシステムの共有解除と再共有」](#)を参照してください。

---

次の例では、ファイルシステム /usr/src を共有解除します。

```
unshare /usr/src
```

## shareall コマンド

shareall コマンドでは複数のファイルシステムを共有できます。オプションなしで使用すると、SMF リポジトリ内のすべてのエントリが共有されます。share コマンドを並べたファイルの名前を指定することができます。

次の例では、ローカルファイルにリストされているすべてのファイルシステムを共有します。

```
shareall /etc/dfs/special_dfstab
```

## unshareall コマンド

unshareall コマンドでは、現在共有されているすべてのリソースが利用できなくなります。-F *FSType* は、/etc/dfs/fstypes に定義されているファイルシステムタイプのリストを選択するオプションです。このフラグによって、特定のタイプのファイルシステムだけを共有解除できます。デフォルトファイルシステムタイプは /etc/dfs/

`fstypes` に定義されています。特定のファイルシステムを選択するには、`unshare` コマンドを使います。

次の例では、すべての NFS タイプファイルシステムを共有解除します。

```
unshareall -F nfs
```

## showmount コマンド

`showmount` コマンドは、次の情報を表示するために使用します。

- NFS サーバーから共有している、リモートマウントされたファイルシステムを持つすべてのクライアント
- クライアントによってマウントされたファイルシステムのみ
- 共有されたファイルシステムとクライアントアクセス情報

---

**注記** - `showmount` コマンドでは、NFS Version 2 および NFS Version 3 エクスポートだけが表示されます。このコマンドでは NFS Version 4 エクスポートは表示されません。

---

このコマンドの構文に次のようになります。

```
showmount [-ade] [hostname]
```

- |                       |                                                      |
|-----------------------|------------------------------------------------------|
| <code>-a</code>       | すべてのリモートマウントのリストを出力します。各エントリには、クライアント名とディレクトリが含まれます。 |
| <code>-d</code>       | クライアントがリモートマウントしたディレクトリのリストを表示します。                   |
| <code>-e</code>       | 共有されているファイル、またはエクスポートされたファイルのリストを表示します。              |
| <code>hostname</code> | 表示する情報の取得元 NFS サーバーを指定します。                           |

`hostname` を指定しない場合、ローカルホストが照会されます。

次の例では、すべてのクライアント、およびクライアントがマウントしたローカルディレクトリが一覧表示されます。

```
showmount -a bee
lilac:/export/share/man
lilac:/usr/src
rose:/usr/src
tulip:/export/share/man
```

次の例では、マウントされているディレクトリが一覧表示されます。

```
showmount -d bee
/export/share/man
/usr/src
```

次の例では、共有しているファイルシステムが一覧表示されます。

```
showmount -e bee
/usr/src (everyone)
/export/share/man eng
```

/network/nfs/server:default サービスの nfs\_props/showmount\_info プロパティは、showmount コマンドによってクライアントに表示される情報量を制御します。デフォルト値は full です。この値を none に設定すると、クライアントにはサーバー上のリモートファイルシステムのうち、クライアントがマウントできるもののみが表示されます。ほかのクライアントに関する情報は表示されません。このプロパティの変更方法の例については、例5「[クライアントに表示されるファイルシステム情報を制限する](#)」を参照してください。

## nfsref コマンド

nfsref コマンドは、NFSv4 リフェラルを追加、削除、または一覧表示するために使用されます。このコマンドの構文は次のとおりです。

```
nfsref add path location [location ...]
```

```
nfsref remove path
```

```
nfsref lookup path
```

*path* 再解析ポイントの名前を決定します。

*location* 再解析ポイントに関連付ける 1 つ以上の NFS または SMB 共有ファイルシステムを識別します。

## FedFS コマンド

次のコマンドは、FedFS サービスに関連付けられています。

**nsdb-list** LDAP サーバーに格納されているすべての FedFS データを一覧表示します。

**nsdb-nces** LDAP サーバー上のネーミングコンテキストと相対識別名を一覧表示します。



`nsdb-resolve-fsn` 選択されたファイルセット名のファイルセット場所を表示します。

`nsdb-update-nci` FedFS データの識別名を管理します。

`nsdbparams` FedFS 接続を管理します。

これらのコマンドの使用例については、[95 ページの「FedFS の管理」](#)を参照してください。



## ネットワークファイルシステムのトラブルシューティング

---

この章では、NFS トラブルシューティングの方法、手順、およびコマンドについて説明します。この章には、autofs のトラブルシューティングに関する情報と、NFS エラーメッセージとその意味のリストも含まれています。

この章の内容は、次のとおりです。

- 139 ページの「NFS のトラブルシューティングの方法」
- 140 ページの「NFS のトラブルシューティング用のコマンド」
- 146 ページの「NFS トラブルシューティング手順」
- 152 ページの「autofs のトラブルシューティング」
- 156 ページの「NFS エラーメッセージ」

### NFS のトラブルシューティングの方法

NFS の問題を追跡するときは、障害が主に発生する可能性がある場所 (NFS サーバー、NFS クライアント、およびネットワーク) を覚えておいてください。各コンポーネントを切り分けて、機能しないコンポーネントを見つけるようにしてください。リモートマウントが成功するように、mountd デーモンと nfsd デーモンがサーバー上で常に動作している必要があります。

すべてのマウントに `-intr` オプションがデフォルトで設定されています。プログラムが `server not responding` メッセージでハングアップする場合は、キーボード割り込み (Ctrl-C) でプログラムを終了できます。

ネットワークまたはサーバーに問題があるときは、ハードマウントされたりリモートファイルにアクセスするプログラムは、ソフトマウントされたりリモートファイルにアクセスするプログラムと異なる方法で失敗します。ハードマウントされたりリモートファイルシステムの場合、クライアントのカーネルは、サーバーがふたたび応答するまで要求を再試行します。ソフトマウントされたりリモートファイルシステムでは、クライアントのシステムコールはいくつかの試行後にエラーを返します。エラーによっ

て予期しないアプリケーションエラーやデータ破損が発生する可能性があるため、ソフトマウントは避けてください。

ファイルシステムがハードマウントされているときは、サーバーが応答に失敗した場合に、ファイルシステムにアクセスしようとするプログラムはハングアップします。この場合、NFS は次のメッセージをコンソールに表示します。

```
NFS server hostname not responding still trying
```

サーバーが少し後に応答すると、次のメッセージがコンソールに表示されます。

```
NFS server hostname ok
```

サーバーが応答しないような、ソフトマウントされたファイルシステムにアクセスしているプログラムは、次のメッセージを表示します。

```
NFS operation failed for server hostname: error # (error-message)
```



---

**注意** - 読み取りと書き込みをするデータを持つファイルシステム、または実行可能ファイルを持つファイルシステムは、ソフトマウントしないでください。エラーが発生する可能性があります。アプリケーションがそのようなソフトエラーを無視すれば、書き込み可能なデータが破壊される恐れがあります。またマウントされた実行可能ファイルが正常にロードされず、動作も正常に行われない可能性があります。

---

## NFS のトラブルシューティング用のコマンド

このセクションでは、NFS 問題をトラブルシューティングするために使用できるコマンドについて説明します。

### nfsstat コマンド

このコマンドでは、NFS および RPC 接続に関する統計情報が表示されます。NFS サーバーおよびクライアント統計を表示するには、次の構文を使用します。

```
nfsstat [-cmnrzs]
```

- c                   クライアント側情報を表示します
- m                   NFS マウントされた各ファイルシステムの統計を表示します
- n                   クライアント側とサーバー側の両方の NFS 情報を表示します。
- r                   RPC 統計を表示します
- s                   サーバー側の情報を表示します

-z 統計をゼロに設定するように指定します

オプションを指定しない場合、-cnrs オプションが使用されます。

新しいソフトウェアやハードウェアを処理環境に追加した場合、サーバー側の統計を収集することが、デバッグにたいへん役立ちます。このコマンドを週に最低1度は実行し、履歴を作成するようにしてください。統計を保存しておく、以前のパフォーマンスの有効な記録となります。

#### 例 10 NFS サーバー統計を表示する

```
nfsstat -s

Server rpc:
Connection oriented:
calls badcalls nullrecv badlen xdrCALL dupchecks dupreqs
719949194 0 0 0 0 58478624 33
Connectionless:
calls badcalls nullrecv badlen xdrCALL dupchecks dupreqs
73753609 0 0 0 0 987278 7254

Server NFSv2:
calls badcalls referrals referlinks
25733 0 0 0

Server NFSv3:
calls badcalls referrals referlinks
132880073 0 0 0

Server NFSv4:
calls badcalls referrals referlinks
488884996 4 0 0
Version 2: (746607 calls)
null getattr setattr root lookup readlink read
883 0% 60 0% 45 0% 0 0% 177446 23% 1489 0% 537366 71%
wrcache write create remove rename link symlink
0 0% 1105 0% 47 0% 59 0% 28 0% 10 0% 9 0%
mkdir rmdir readdir statfs
26 0% 0 0% 27926 3% 108 0%

Version 3: (728863853 calls)
null getattr setattr lookup access
1365467 0% 496667075 68% 8864191 1% 66510206 9% 19131659 2%
readlink read write create mkdir
414705 0% 80123469 10% 18740690 2% 4135195 0% 327059 0%
symlink mknod remove rmdir rename
101415 0% 9605 0% 6533288 0% 111810 0% 366267 0%
link readdir readdirplus fsstat fsinfo
2572965 0% 519346 0% 2726631 0% 13320640 1% 60161 0%
pathconf commit
13181 0% 6248828 0%

Version 4: (54871870 calls)
null compound
266963 0% 54604907 99%

Version 4: (167573814 operations)
reserved access close commit
0 0% 2663957 1% 2692328 1% 1166001 0%
create delegpurge delegreturn getattr
167423 0% 0 0% 1802019 1% 26405254 15%
getfh link lock lockt
11534581 6% 113212 0% 207723 0% 265 0%
```

```

locku lookup lookupp nverify
230430 0% 11059722 6% 423514 0% 21386866 12%
open openattr open_confirm open_downgrade
2835459 1% 4138 0% 18959 0% 3106 0%
putfh putpubfh putrootfh read
52606920 31% 0 0% 35776 0% 4325432 2%
readdir readlink remove rename
606651 0% 38043 0% 560797 0% 248990 0%
renew restorefh savefh secinfo
2330092 1% 8711358 5% 11639329 6% 19384 0%
setattr setclientid setclientid_confirm verify
453126 0% 16349 0% 16356 0% 2484 0%
write release_lockowner illegal
3247770 1% 0 0% 0 0%

Server nfs_acl:
Version 2: (694979 calls)
null getacl setacl getattr access getxattrdir
0 0% 42358 6% 0 0% 584553 84% 68068 9% 0 0%
Version 3: (2465011 calls)
null getacl setacl getxattrdir
0 0% 1293312 52% 1131 0% 1170568 47%

```

この例では、RPC および NFS アクティビティの統計を表示する方法を示します。どちらの統計でも、badcalls または calls の平均値、および各週の calls の数がわかるので、問題を特定するのに役立ちます。badcalls 値は、クライアントからの悪いメッセージの数を報告します。この値は、ネットワークのハードウェアに問題が発生したことを示す場合があります。

いくつかの接続では、ディスクに対する書き込みアクティビティが発生します。この数値の急激な上昇は障害の可能性を示すもので、調査が必要です。NFS Version 2 統計では、注目すべき接続は、setattr、write、create、remove、rename、link、symlink、mkdir、および rmdir です。NFS Version 3 と NFS Version 4 統計では、監視すべき値は commit です。ある NFS サーバーの commit レベルが、別のほぼ同等のサーバーと比較して高い場合は、NFS クライアントのメモリーが十分であることを確認してください。サーバーの commit 操作の数は、クライアントに利用可能なリソースがないときに増えます。

## pstack コマンド

pstack コマンドでは、各プロセスのスタックトレースが表示されます。pstack コマンドは、プロセスの所有者によってまたは root で実行される必要があります。pstack コマンドを使用してプロセスがハングアップした場所を判断できます。このコマンドで許可されるオプションは、確認するプロセスのプロセス ID だけです。pstack コマンドの詳細は、[proc\(1\)](#) のマニュアルページを参照してください。

**例 11** NFS プロセスのスタックトレースを表示する

```
/usr/bin/pgrep nfsd
```

```

243
/usr/bin/pstack 243
243: /usr/lib/nfs/nfsd -a 16
ef675c04 poll (24d50, 2, ffffffff)
000115dc ???????? (24000, 132c4, 276d8, 1329c, 276d8, 0)
00011390 main (3, effffff14, 0, 0, ffffffff, 400) + 3c8
00010fb0 _start (0, 0, 0, 0, 0, 0) + 5c

```

この例では、プロセスが新規の接続要求を持っていることが示されています。これは正常な反応です。要求が行われた後でもプロセスがポーリングしていることがスタックからわかった場合、そのプロセスはハングアップしている可能性があります。ハングアッププロセスの修復に関する詳細は、[150 ページの「NFS サービスを再起動する方法」](#)を参照してください。NFS のトラブルシューティングの詳細は、[146 ページの「NFS トラブルシューティング手順」](#)を参照してください。

## rpcinfo コマンド

rpcinfo コマンドは、システムで動作している RPC サービスに関する情報を生成します。RPC サービスに関する情報を表示するときは、次のコマンド構文を使用してください。

```
rpcinfo [-m | -s] [hostname]
```

```
rpcinfo [-T transport] hostname [progname]
```

```
rpcinfo [-t | -u] [hostname] [progname]
```

|           |                                      |
|-----------|--------------------------------------|
| -m        | rpcbind 処理の統計テーブルを表示します              |
| -s        | 登録されているすべての RPC プログラムを簡易リストで表示します    |
| -T        | 特定のトランスポートまたはプロトコルを使用するサービスの情報を表示します |
| -t        | TCP を使用する RPC プログラムを検索します            |
| -u        | UDP を使用する RPC プログラムを検索します            |
| transport | サービスのトランスポートまたはプロトコルを指定します           |
| hostname  | サーバーのホスト名を指定します                      |
| progname  | RPC プログラムの名前を指定します                   |

使用可能なオプションの詳細は、[rpcinfo\(1M\)](#) のマニュアルページを参照してください。

*hostname* を指定しないと、ローカルホスト名が使用されます。*progname* に RPC プログラム番号を代入できますが、名前がより一般的に使用されます。NFS Version 3 ソフトウェアが実行していないシステムでは、*-s* オプションの代わりに *-p* オプションを使用できます。

このコマンドを実行すると、次の項目を含むデータを生成することができます。

- RPC プログラム番号
- 特定プログラムのバージョン番号
- 使用中のトランスポートプロトコル
- RPC サービスの名前
- RPC サービスの所有者

#### 例 12 RPC サービス情報を表示する

```
rpcinfo -s bee |sort -n
program version(s) netid(s) service owner
100000 2,3,4 udp6,tcp6,udp,tcp,ticlts,ticotsord,ticots portmapper superuser
100001 4,3,2 udp6,udp,ticlts rstatd superuser
100003 4,3,2 tcp,udp,tcp6,udp6 nfs 1
100005 3,2,1 ticots,ticotsord,tcp,tcp6,ticlts,udp,udp6 mountd superuser
100007 1,2,3 ticots,ticotsord,ticlts,tcp,udp,tcp6,udp6 ypbind 1
100011 1 udp6,udp,ticlts rquotad superuser
100021 4,3,2,1 tcp,udp,tcp6,udp6 nlockmgr 1
100024 1 ticots,ticotsord,ticlts,tcp,udp,tcp6,udp6 status superuser
100068 5,4,3,2 ticlts - superuser
100083 1 ticotsord - superuser
100133 1 ticots,ticotsord,ticlts,tcp,udp,tcp6,udp6 - superuser
100134 1 ticotsord - superuser
100155 1 ticotsord smserverd superuser
100169 1 ticots,ticotsord,ticlts - superuser
100227 3,2 tcp,udp,tcp6,udp6 nfs_acl 1
100234 1 ticotsord - superuser
390113 1 tcp - superuser
390435 1 tcp - superuser
390436 1 tcp - superuser
1073741824 1 tcp,tcp6 - 1
```

次の例では、サーバーで実行中の RPC サービスに関する情報を表示します。このコマンドによって生成される出力は、情報を読みやすくするために、*sort* コマンドによってプログラム番号でフィルタされます。この例では、RPC サービスの数行を省略しています。

サーバー上で特定のトランスポートを選択することで、特定の RPC サービスについての情報を収集できます。次の例では、TCP 経由で実行中の *mountd* サービスを確認します。

```
rpcinfo -t bee mountd
program 100005 Version 1 ready and waiting
program 100005 Version 2 ready and waiting
```



```
program 100005 Version 3 ready and waiting
```

次の例では、UDP 経由で実行中の NFS サービスを確認します。

```
rpcinfo -u bee nfs
program 100003 Version 2 ready and waiting
program 100003 Version 3 ready and waiting
```

## snoop コマンド

snoop コマンドは、ネットワーク上のパケットを監視するために使用されます。snoop コマンドは、root ユーザーとして実行される必要があります。このコマンドは、NFS クライアントと NFS サーバーの両方で、ネットワークハードウェアが機能しているかどうかを確認する方法としてよく使用されます。

ネットワーク上のパケットを監視するときは、次のコマンド構文を使用してください。

```
snoop [-d device] [-o filename] [host hostname]
```

**-d device**                    ローカルネットワークのインタフェースを指定します

**-o filename**                受信したすべてのパケットを指定したファイルに保存します

**hostname**                    特定のホストが送受信したパケットを表示します

**-d device** オプションは、複数のネットワークインタフェースを持つサーバーで役立ちます。ホストの設定以外にも、使用できる式が多数あります。コマンド正規表現を **grep** で組み合わせることで、十分に使用できるデータを生成できます。使用可能なオプションの詳細は、[snoop\(1M\)](#) のマニュアルページを参照してください。

トラブルシューティングをする場合は、パケットの発信元と送信先のホストが正しいことを確認してください。また、エラーメッセージも調べてください。パケットをファイルに保存すると、データを簡単に参照することができます。

## truss コマンド

truss コマンドを使用して、プロセスがハングアップしているかどうかを確認できます。truss コマンドは、プロセスの所有者によってまたは root で実行される必要があります。

プロセスがハングアップしているかどうかを確認するときは、次のコマンド構文を使用してください。

```
truss [-t syscall] -p pid
```

-t syscall                    追跡するシステムコールを選択します

-p pid                        追跡するプロセスの PID を指定します

syscall は、追跡するシステムコールのコンマ区切りリストです。リストを ! 文字で開始すると、リストされたシステムコールがトレースから除外されます。使用可能なオプションの詳細は、[truss\(1\)](#) のマニュアルページを参照してください。

**例 13**                    プロセスステータスを表示する

```
/usr/bin/truss -p 243
poll(0x00024D50, 2, -1) (sleeping...)
```

この例では、プロセスが別の接続要求を待機していることを示しています (正常な応答)。新規接続の要求が行われた後でも反応が変わらない場合、そのプロセスはハングアップしている可能性があります。

NFS サービスの再起動については、[150 ページ](#)の「[NFS サービスを再起動する方法](#)」を参照してください。ハングアッププロセスのトラブルシューティングについては、[146 ページ](#)の「[NFS トラブルシューティング手順](#)」を参照してください。

## NFS トラブルシューティング手順

NFS サービスがエラーになった場所を判断するには、いくつかの手順を踏まなければなりません。次をチェックします。

- クライアントがサーバーに到達できるかどうか
- クライアントがサーバー上の NFS サービスに接続できるかどうか
- NFS サービスがサーバー上で動作しているかどうか

このプロセス中に、ネットワークのほかの部分機能が機能していないことに気付く場合があります。たとえば、ネームサービスやネットワークのハードウェアが機能していない場合があります。ネームサービスについては、『[Oracle Solaris 11.3 ディレクトリサービスとネームサービスでの作業: DNS と NIS](#)』を参照してください。プロセス中に、問題がクライアント側でないことに気付く場合があります (たとえば、作業エリア内のすべてのサブネットから問題を受け取る場合)。このような状況では、問題がサーバーまたはサーバーの近くのネットワークハードウェアであると想定し、クライアントではなくサーバーでデバッグプロセスを開始してください。

## ▼ NFS クライアントの接続性を確認する方法

1. クライアントで、NFS サーバーに到達できることを確認します。

```
/usr/sbin/ping bee
bee is alive
```

コマンドを入力した結果、サーバーが動作していることがわかったら、NFS サーバーをリモートで確認します。NFS サーバーのリモートチェックについては、[148 ページの「NFS サーバーをリモートで確認する方法」](#)を参照してください。

2. クライアントからサーバーに到達できない場合は、クライアントでローカルネームサービスが動作していることを確認します。

次に例を示します。

- NIS ネームサービスを使用している場合は、ypbind デーモンが動作していることを確認します。詳細は、『[Oracle Solaris 11.3 ディレクトリサービスとネームサービスでの作業: DNS と NIS](#)』の「[ypbind が NIS クライアントで実行されていない](#)」を参照してください。
- LDAP ネームサービスを使用している場合は、ldap\_cachemgr デーモンが動作していることを確認します。詳細は、『[Oracle Solaris 11.3 ディレクトリサービスとネームサービスでの作業: LDAP](#)』の「[LDAP クライアントステータスのモニタリング](#)」を参照してください。

3. ネームサービスが動作している場合は、クライアントが正しいホスト情報を受け取っていることを確認します。

```
/usr/bin/getent hosts system
```

次に例を示します。

```
/usr/bin/getent hosts bee
192.168.83.117 bee.eng.example.com
```

4. ホスト情報は正しいけれどもクライアントからサーバーに到達できない場合は、別のクライアントから ping コマンドを実行します。

2 番目のクライアントからのコマンド実行が失敗する場合は、NFS サービスがサーバー上で有効になっているかどうかを確認してください。詳細は、[149 ページの「サーバーで NFS サービスを確認する方法」](#)を参照してください。

5. 2 番目のクライアントからサーバーに到達できる場合は、ping を使用して 1 番目のクライアントからローカルネット上のほかのシステムへの接続を確認します。

ping コマンドが失敗する場合は、クライアント上のネットワークソフトウェア構成を確認してください (/etc/netmasks ファイルや、svc:/system/name-service/switch サービスに関連付けられたプロパティ情報など)。

6. (オプション) `rpcinfo` コマンドの出力を確認します。

`rpcinfo` コマンドで「program 100003 Version 4 ready and waiting」が表示されない場合は、NFS Version 4 がサーバー上で有効になっていません。NFS Version 4 の有効化については、[表4](#)を参照してください。

7. ソフトウェアに問題がない場合は、ネットワークハードウェアを確認します。

クライアントをほかの物理ネットワーク接続に移動してみてください。

## ▼ NFS サーバーをリモートで確認する方法

NFS Version 4 サーバーを使用している場合は、UDP および MOUNT プロトコルのサポートは必要ありません。

1. NFS デーモンが NFS サーバー上で起動されていることを確認します。

```
rpcinfo -s server-name | egrep 'nfs|mountd'
```

次に例を示します。

```
rpcinfo -s bee | egrep 'nfs|mountd'
100003 3,2 tcp,udp,tcp6,udp6 nfs superuser
100005 3,2,1 ticots,ticotsord,tcp,tcp6,ticlts,udp,udp6 mountd superuser
```

デーモンが起動していない場合は、NFS サービスを再起動してください。詳細は、[150 ページの「NFS サービスを再起動する方法」](#)を参照してください。

2. クライアント上で、サーバーからの UDP NFS 接続をテストします。

```
/usr/bin/rpcinfo -u bee nfs
program 100003 Version 2 ready and waiting
program 100003 Version 3 ready and waiting
```

---

**注記** - NFS Version 4 は UDP をサポートしません。

---

サーバーが動作している場合は、`rpcinfo` コマンドで UDP プロトコルに関連付けられたプログラムとバージョン番号が一覧表示されます。`rpcinfo` コマンドで `-t` オプションを使用して TCP 接続を確認できます。`rpcinfo` コマンドが失敗する場合は、NFS サービスがサーバー上で有効になっているかどうかを確認してください。詳細は、[149 ページの「サーバーで NFS サービスを確認する方法」](#)を参照してください。

3. サーバーの `mountd` デーモンが応答していることを確認します。

```
/usr/bin/rpcinfo -u bee mountd
program 100005 Version 1 ready and waiting
program 100005 Version 2 ready and waiting
program 100005 Version 3 ready and waiting
```

サーバーが動作している場合は、`rpcinfo` コマンドで UDP プロトコルに関連付けられたプログラムとバージョン番号が一覧表示されます。`-t` オプションを使用すると、TCP 接続を検査できます。`nfsd` および `mountd` デーモンが動作しているかどうかを確認します。

4. ローカル `autofs` サービスがクライアントで使用されていることを、正しく動作している `/net` または `/home` マウントポイントに変更することで確認します。

```
cd /net/eng
```

このコマンドが失敗する場合は、クライアントで `root` ユーザーとして、`autofs` サービスを再起動してください。

```
svcadm restart system/filesystem/autofs
```

5. ファイルシステムがサーバー上で予期したとおりに共有されていることを確認します。

```
/usr/sbin/showmount -e bee
/usr/src eng
/export/share/man (everyone)
```

サーバーの項目とローカルマウントエントリにエラーがないことをチェックします。名前空間も確認します。この例では、最初のクライアントが `eng` ネットグループ内がない場合、クライアントは `/usr/src` ファイルシステムをマウントできません。

すべてのローカルファイルを調べて、マウント情報を含むエントリをすべて検査します。リストには `/etc/vfstab` ファイルとすべての `/etc/auto_*` ファイルが含まれています。

## ▼ サーバーで NFS サービスを確認する方法

1. 管理者になります。

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. サーバーがクライアントに到達できることを確認します。

```
ping lilac
lilac is alive
```

3. サーバーからクライアントに到達できない場合は、クライアント上でローカルネームサービスが動作していることを確認します。

4. ネームサービスが実行されている場合は、サーバー上のネットワークソフトウェア構成を確認します。たとえば、`/etc/netmasks` や、`svc:/system/name-service/switch` サービスに関連付けられているプロパティ情報を確認します。

## 5. rpcbind デーモンがサーバーで動作しているかどうかを確認します。

```
/usr/bin/rpcinfo -u localhost rpcbind
program 100000 Version 1 ready and waiting
program 100000 Version 2 ready and waiting
program 100000 Version 3 ready and waiting
```

サーバーが動作している場合は、rpcinfo コマンドで、UDP プロトコルに関連付けられているプログラムとそのバージョン番号が一覧表示されます。

## 6. nfsd デーモンがサーバーで動作しているかどうかを確認します。

```
rpcinfo -u localhost nfs
program 100003 Version 2 ready and waiting
program 100003 Version 3 ready and waiting
ps -ef | grep nfsd
root 101328 0 0 Jul 12 ? 303:25 nfsd_kproc
root 101327 1 0 Jul 12 ? 2:54 /usr/lib/nfs/nfsd
root 263149 131084 0 13:59:19 pts/17 0:00 grep nfsd
```

---

注記 - NFS Version 4 は UDP をサポートしません。

---

サーバーが動作している場合は、rpcinfo コマンドで UDP プロトコルに関連付けられたプログラムとバージョン番号が一覧表示されます。さらに、rpcinfo で -t オプションを使用して TCP 接続を確認してください。これらのコマンドを使用するとエラーになる場合は、NFS サービスを再起動します。詳細は、[150 ページの「NFS サービスを再起動する方法」](#)を参照してください。

## 7. mountd デーモンがサーバーで動作しているかどうかを確認します。

```
/usr/bin/rpcinfo -t localhost mountd
program 100005 Version 1 ready and waiting
program 100005 Version 2 ready and waiting
program 100005 Version 3 ready and waiting
ps -ef | grep mountd
root 145 1 0 Apr 07 ? 21:57 /usr/lib/autofs/automountd
root 234 1 0 Apr 07 ? 0:04 /usr/lib/nfs/mountd
root 3084 2462 1 09:30:20 pts/3 0:00 grep mountd
```

サーバーが動作している場合は、rpcinfo コマンドで UDP プロトコルに関連付けられたプログラムとバージョン番号が一覧表示されます。さらに、rpcinfo で -t オプションを使用して TCP 接続を確認してください。これらのコマンドを使用するとエラーになる場合は、NFS サービスを再起動します。詳細は、[150 ページの「NFS サービスを再起動する方法」](#)を参照してください。

## ▼ NFS サービスを再起動する方法

### 1. 管理者になります。

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. サーバー上で NFS サービスを再起動します。

```
svcadm restart network/nfs/server
```

## NFS サービスを提供するホストの識別

nfsstat コマンドを `-m` オプション付きで使用して現在の NFS 情報を表示します。現在のサーバー名は、「currserver=」のあとに表示されます。

次に例を示します。

```
nfsstat -m
/usr/local from bee,wasp:/export/share/local
Flags: vers=3,proto=tcp,sec=sys,hard,intr,llock,link,synlink,
 acl,rsize=32768,wsiz=32678,retrans=5
Failover: noresponse=0, failover=0, remap=0, currserver=bee
```

## ▼ mount コマンドで使用されたオプションを確認する方法

mount コマンドで無効なオプションが指定されても警告は発行されません。この手順は、コマンド行または `/etc/vfstab` で使用されたオプションが有効であったかどうかを判断するのに役立ちます。

この手順の例では、次のコマンドが実行されたと想定してください。

```
mount -F nfs -o ro,vers=2 bee:/export/share/local /mnt
```

1. オプションを確認します。

```
nfsstat -m
/mnt from bee:/export/share/local
Flags: vers=2,proto=tcp,sec=sys,hard,intr,dynamic,acl,rsize=8192,wsiz=8192,
 retrans=5
```

bee サーバーからのファイルシステムは、プロトコルバージョンが 2 に設定された状態でマウントされました。nfsstat コマンドですべてのオプションの情報が表示されるわけではありません。しかし、オプションを確認するには nfsstat コマンドを使用することがもっとも正確な方法です。

2. `/etc/mnttab` ファイル内のエントリを確認します。

mount コマンドは、無効なオプションをマウントテーブルに追加することができません。そのため、ファイルにリストされているオプションがコマンド行でリストされたオプションと一致していることを確認してください。この方法で、nfsstat コマンドで報告されないオプションを確認できます。

```
grep bee /etc/mnttab
```

```
bee:/export/share/local /mnt nfs ro,vers=2,dev=2b0005e 859934818
```

## autofs のトラブルシューティング

autofs でときどき問題が発生することがあります。このセクションでは、autofs が生成するエラーメッセージのリストを示します。このリストは、2つのパートに分かれています。

- automount の冗長 (-v) オプションで生成されるエラーメッセージ
- 通常表示されるエラーメッセージ

各エラーメッセージの後には、そのメッセージの説明と考えられる原因が続きます。

トラブルシューティング時には、詳細形式 (-v) オプションで autofs プログラムを開始します。

### automount -v で生成されるエラーメッセージ

```
bad key key in direct map mapname
```

**説明:** 直接マップのスキヤン中、autofs が接頭辞 / のないエントリ鍵を発見しました。

**対処方法:** 直接マップ内の鍵は、フルパス名でなくてはなりません。

```
bad key key in indirect map mapname
```

**説明:** 間接マップのスキヤン中、autofs が / を含むエントリ鍵を発見しました。

**対処方法:** 間接マップの鍵は、パス名ではなく、単なる名称でなくてはなりません。

```
can't mount server: pathname: reason
```

**説明:** サーバー上のマウントデーモンが *server:pathname* 用のファイルハンドルを提供するのを拒否しました。

**対処方法:** サーバー上のエクスポートテーブルを確認してください。

```
couldn't create mount point mountpoint: reason
```

**説明:** autofs は、マウントに必要なマウントポイントを作成することができませんでした。この問題は、すべてのサーバーのエクスポートされたファイルシステムを階層的にマウントしようとする場合に頻繁に生じます。



**対処方法:** 必要なマウントポイントは、マウントできないファイルシステム内にだけ存在するため、ファイルシステムはエクスポートできません。エクスポートされる親ファイルシステムは、読み取り専用でエクスポートされるため、マウントポイントを作成できません。

#### leading space in map entry *entry text in mapname*

**説明:** autofs は自動マウントマップ内に先頭にスペースを含むエントリを発見しました。この問題は、通常、マップエントリが不当である場合に発生します。次に例を示します。

```
fake
/blast frobz:/usr/frotz
```

**対処方法:** この例では、autofs が 2 つめの行を検出した場合に警告が生成されます。これは、最初の行がバックスラッシュ (\) で終端されていないためです。

#### *mapname*: Not found

**説明:** 必要とされるマップが配置されていません。このメッセージは、`-v` オプションが使用されている場合にだけ生成されます。

**対処方法:** マップ名のスペルとパス名を確認してください。

#### remount *server:pathname* on *mountpoint*: server not responding

**説明:** autofs が、アンマウントしたファイルシステムの再マウントに失敗しました。

**対処方法:** My Oracle Support に連絡して支援を依頼してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

#### WARNING: *mountpoint* already mounted on

**説明:** autofs が、既存のマウントポイント上にマウントしようとしてしました。このメッセージは、autofs 内で内部エラー (異常) が生じたことを意味しています。

**対処方法:** My Oracle Support に連絡して支援を依頼してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

## その他のエラーメッセージ

#### *dir mountpoint* must start with '/'

**対処方法:** オートマウンタのマウントポイントは、フルパス名で指定しなくてはなりません。マウントポイントのスペルとパス名を確認してください。

**hierarchical mountpoint: *pathname1* and *pathname2***

**対処方法:** autofs は、マウントポイントが階層的な関係を持つことを許可しません。autofs マウントポイントは、他の自動マウントされたファイルシステムに含まれていてはなりません。

**host *server* not responding**

**説明:** autofs が *server* に接続しようとしたのですが、応答を受け取りませんでした。

**対処方法:** NFS サーバーのステータスを確認してください。

**hostname: exports: *rpc-err***

**説明:** このエラーは、*hostname* からエクスポートリストを取得する場合に発生します。このメッセージは、サーバーまたはネットワークに問題があることを示します。

**対処方法:** NFS サーバーのステータスを確認してください。

**map *mapname*, key *key*: bad**

**説明:** マップエントリの形式が間違っているため、autofs がエントリを解釈できません。

**対処方法:** エントリを再確認してください。そのエントリに、エスケープする必要のある文字が含まれている可能性があります。

**mapname: *nis-err***

**説明:** このエラーは、NIS マップのエントリを参照する場合に発生します。このメッセージは、NIS に問題がある可能性があることを示しています。

**対処方法:** NIS サーバーのステータスを確認してください。

**mount of *server:pathname* on *mountpoint:reason***

**説明:** autofs がマウントに失敗しました。サーバーまたはネットワークに問題のある可能性があります。*reason* の文字列によって、問題が特定されます。

**対処方法:** My Oracle Support に連絡して支援を依頼してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

**mountpoint: Not a directory**

説明: *mountpoint* がディレクトリではないため、autofs はその場所で自身をマウントできません。

対処方法: マウントポイントのスペルとパス名を確認してください。

**nfscast: cannot send packet: reason**

説明: autofs が、複製されたファイルシステムの場所を示すリスト内にあるサーバーへの照会パケットを送信できません。reason の文字列によって、問題が特定されません。

対処方法: My Oracle Support に連絡して支援を依頼してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

**nfscast: cannot receive reply: reason**

説明: autofs が、複製されたファイルシステムの場所を示すリスト内にあるいずれのサーバーからも応答を受けられません。reason の文字列によって、問題が特定されません。

対処方法: My Oracle Support に連絡して支援を依頼してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

**nfscast: select: reason**

説明: このエラーメッセージは、複製されたファイルシステムについてサーバーを確認しようとしたときの問題を示します。このメッセージは、ネットワークに問題がある可能性があることを示しています。reason の文字列によって、問題が特定されません。

対処方法: My Oracle Support に連絡して支援を依頼してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

**pathconf: no info for server:pathname**

説明: autofs がパス名の pathconf 情報を取得することに失敗しました。

対処方法: 構成可能なパス名については、[fpathconf\(2\)](#) のマニュアルページを参照してください。

**pathconf: server: server not responding**

説明: autofs が、`pathconf()` に情報を提供する *server* に示されるサーバー上のマウントデーモンにコンタクトできませんでした。

**対処方法:** このサーバーで POSIX マウントオプションを使用しないでください。

## autofs のその他のエラー

/etc/auto\* ファイルが実行ビットセットを持っている場合、オートマウンタは次のようなメッセージを生成するマップの実行を試みます。

```
/etc/auto_home: +auto_home: not found
```

この場合、auto\_home ファイルは不適切な権限をもつこととなります。このファイル内の各エントリは、よく似たエラーメッセージを生成します。ファイルへのアクセス権をリセットするには次のコマンドを入力してください。

```
chmod 644 /etc/auto_home
```

## NFS エラーメッセージ

このセクションでは、NFS エラーメッセージ、そのエラーを作成できる状態の説明、および可能な解決策を一覧表示します。

**Bad argument specified with index option - must be a file**

**対処方法:** index オプションにファイル名を含めるする必要があります。ディレクトリ名は使用できません。

**Cannot establish NFS service over /dev/tcp: transport setup problem**

**説明:** このメッセージは、名前空間内のサービス情報が更新されなかったときによく生成されます。またこのメッセージは、UDP の状態を示すことがあります。

**対処方法:** この問題を解決するには、名前空間の中のサービスデータを更新します。

NIS と /etc/services の場合、エントリは次のようにする必要があります。

```
nfsd 2049/tcp nfs # NFS server daemon
nfsd 2049/udp nfs # NFS server daemon
```

**Could not start daemon: error**

**説明:** このメッセージは、デーモンが異常終了するか、システムコールにエラーが発生した場合に表示されます。error 文字列に問題が記述されます。

**対処方法:** My Oracle Support に連絡して支援を依頼してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

Could not use public filehandle in request to server

説明: このメッセージは、`public` オプションが指定されているけれども NFS サーバーが公開ファイルハンドルをサポートしない場合に表示されます。この場合、マウントが失敗します。

対処方法: 公開ファイルハンドルを使用しないでマウント要求を試行するか、公開ファイルハンドルをサポートするように NFS サーバーを再構成してください。

*daemon running already with pid pid*

説明: デーモンがすでに実行中です。

対処方法: 新しいプロセスを実行する場合は、現在のバージョンを終了して新しいバージョンを開始してください。

error locking *lock-file*

説明: このメッセージは、デーモンに関連付けられている *lock-file* を正しくロックできなかったときに表示されます。

対処方法: My Oracle Support に連絡して支援を依頼してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

error checking *lock-file*: error

説明: このメッセージは、デーモンに関連付けられている *lock-file* を正しく開けなかったときに表示されます。

対処方法: My Oracle Support に連絡して支援を依頼してください。このエラーメッセージが出力されることはほとんどなく、直接的な解決策はありません。

NOTICE: NFS3: failing over from *host1* to *host2*

説明: このメッセージは、フェイルオーバーが発生したときにコンソールに表示されます。報告のためだけのメッセージです。

対処方法: アクションは必要ありません。

*filename*: File too large

説明: NFS Version 2 クライアントが、2G バイトを超えるファイルにアクセスしようとしています。

**対処方法:** NFS Version 2 の使用を避けてください。NFS Version 3 または NFS Version 4 でファイルシステムをマウントしてください。nolargefiles オプションの説明については、[mount\(1M\)](#) マニュアルページを参照してください。

mount: ... server not responding:RPC\_PMAP\_FAILURE - RPC\_TIMED\_OUT

**説明:** マウントしようとしているファイルシステムを共有しているサーバーが、ダウンしているか、到達不能か、または間違っただ実行レベルであるか、rpcbind プロセスが死んでいるまたはハングアップしています。

**対処方法:** サーバーがリブートするまで待機します。サーバーがハングアップしている場合は、サーバーをリブートします。

mount: ... server not responding: RPC\_PROG\_NOT\_REGISTERED

**説明:** マウント要求は rpcbind プロセスによって登録されているけれども、NFS マウントデーモン mountd が登録されていません。

**対処方法:** サーバーがリブートするまで待機します。サーバーがハングアップしている場合は、サーバーをリブートします。

mount: ... No such file or directory

**説明:** リモートディレクトリもローカルディレクトリも存在しません。

**対処方法:** ディレクトリ名のスペルをチェックします。両方のディレクトリで ls コマンドを実行します。

mount: ...: Permission denied

**説明:** システム名が、クライアントのリスト内にはないか、マウントしようとしたファイルシステムへのアクセスを許可されているネットグループ内にはない可能性があります。

**対処方法:** showmount -e コマンドを使ってアクセスリストを確認してください。

NFS file temporarily unavailable on the server, retrying ...

**説明:** NFS Version 4 サーバーはファイルの管理をクライアントに委託できます。このメッセージは、クライアントからの要求と重複するほかのクライアントへの委託を、サーバーが再発信していることを示します。

**対処方法:** サーバーがクライアントの要求を処理する前に、再発信が行われる必要があります。委託の詳細は、[35 ページの「NFS Version 4 における委託」](#)を参照してください。

**NFS fsstat failed for server *hostname*: RPC: Authentication error**

**説明:** さまざまな状況で発生するエラーです。もっともデバッグが困難なのは、ユーザーの属しているグループが多すぎる場合です。現在、ユーザーは最大 16 個のグループに属することができますが、NFS マウントでファイルにアクセスしている場合は、それよりも少なくなります。

**対処方法:** 16 を超えるグループに所属する必要があるユーザーのために、代替策があります。アクセス制御リスト (ACL) を使って、必要なアクセス特権を提供できます。

**nfs mount: NFS can't support "nolargefiles"**

**説明:** NFS クライアントが、`-nolargefiles` オプションを使用して NFS サーバーからファイルシステムをマウントしようとした。

**対処方法:** このオプションは、NFS ファイルシステムタイプに対してはサポートされていません。

**nfs mount: NFS V2 can't support "largefiles"**

**説明:** NFS Version 2 プロトコルは大きなファイルを処理できません。

**対処方法:** 大きなファイルへのアクセスが必要な場合は、NFS Version 3 または NFS Version 4 を使用する必要があります。

**NFS server *hostname* not responding still trying**

**説明:** ファイル関連の作業中にプログラムがハングアップする場合は、NFS サーバーで障害が発生した可能性があります。このメッセージは、NFS サーバー (*hostname*) がダウンしているか、サーバーがネットワークに問題があることを示すものです。

**対処方法:** NFS フェイルオーバーが使用されている場合は、*hostname* はサーバーのリストです。147 ページの「[NFS クライアントの接続性を確認する方法](#)」を参照してください。

**NFS server recovering**

**説明:** NFS Version 4 サーバーリブートの一環で、一部の操作が許可されませんでした。このメッセージは、サーバーがこの操作の続行を許可するまで、クライアントが待機していることを示します。

**対処方法:** アクションは必要ありません。サーバーが操作を許可するまで待機します。

## Permission denied

説明: このメッセージは、次の理由で `ls -l`、`getfacl`、および `setfacl` コマンドによって表示されます。

- NFS Version 4 サーバー上のアクセス制御リスト (ACL) エントリ内に存在するユーザーまたはグループを、NFS Version 4 クライアント上の有効なユーザーまたはグループにマッピングできない場合、ユーザーはクライアント上で ACL を読み取ることができません。
- NFS Version 4 クライアント上で設定されている ACL エントリ内に存在するユーザーまたはグループを、NFS Version 4 サーバー上の有効なユーザーまたはグループにマッピングできない場合、ユーザーはクライアント上の ACL に対して書き込んだり変更したりできません。
- NFS Version 4 クライアントおよびサーバーの `nfsmapid_domain` 値が一致しない場合、ID マッピングが失敗します。

NFS 用の ACL エントリに関する詳細は、[37 ページの「NFS Version 4 での ACL と nfsmapid」](#) を参照してください。

対処方法: 次を実行してください。

- ACL エントリ内のすべてのユーザー ID およびグループ ID がクライアントとサーバーの両方に存在することを確認します。
- `nfsmapid_domain` プロパティの値が SMF リポジトリ内で正しく設定されていることを確認します。

サーバーまたはクライアントでユーザーまたはグループをマッピングできるかどうかを判断するために使用するスクリプトについては、[38 ページの「マッピングされていないユーザー ID またはグループ ID を確認する」](#) を参照してください。

## port number in nfs URL not the same as port number in port option

説明: NFS URL に含まれているポート番号は、マウント上の `-port` オプションに含まれているポート番号と一致する必要があります。一致していないと、マウントは失敗します。

対処方法: 同じポート番号にしてコマンドを再実行するか、ポート番号の指定を省略してください。通常は、NFS URL と `-port` オプションの両方にポート番号を指定する必要はありません。

## replicas must have the same version

説明: NFS フェイルオーバーが正しく機能するには、複製である NFS サーバーが同じバージョンの NFS プロトコルをサポートする必要があります。

対処方法: 複数のバージョンを実行しないでください。



**replicated mounts must be read-only**

**説明:** NFS フェイルオーバーは読み書きでマウントされたファイルシステムでは機能しません。ファイルシステムを読み書き可能としてマウントすると、ファイルが変更される可能性が高くなるためです。

**対処方法:** NFS のフェイルオーバー機能は、ファイルシステムがまったく同じであることが前提です。

**replicated mounts must not be soft**

**説明:** NFS フェイルオーバーが発生する前にタイムアウトを待機することを、複製されたマウントが要求しています。

**対処方法:** `soft` オプションを指定すると、タイムアウトが開始してすぐにマウントが失敗するため、複製されるマウントには `-soft` オプションは指定できません。

**share\_nfs: Cannot share more than one filesystem with 'public' option**

**対処方法:** `share` コマンドを使って、`-public` オプションで共有するために1つのファイルシステムだけが選択されていることを確認してください。公開ファイルハンドルの、サーバーあたり1つしか設定できません。したがって、`public` オプションで共有できるファイルシステムは1つだけです。

**WARNING: No network locking on *hostname:path*: contact admin to install server change**

**説明:** NFS クライアントが NFS サーバー上のネットワークロックマネージャーとの接続を確立しようとしたますが、成功しませんでした。この警告は、マウントできなかったことを知らせるためではなく、ロックが機能しないことを警告するために出力されます。

**対処方法:** 完全なロックマネージャーサポートを提供する新しいバージョンの OS でサーバーをアップグレードしてください。



## ネットワークファイルシステムへのアクセス

この章には、NFS サービスをサポートするファイルとデーモンのリストが含まれています。

この章の内容は、次のとおりです。

- 163 ページの「NFS ファイル」
- 167 ページの「NFS デーモン」

---

注記 - システムでゾーンが有効なときに非大域ゾーンでこの機能を使用する場合は、『Oracle Solaris ゾーンの紹介』を参照してください。

---

### NFS ファイル

システム上で NFS アクティビティをサポートするにはいくつかのファイルが必要です。これらのファイルの多くは ASCII 形式ですが、一部のファイルはデータファイルです。次の表に NFS ファイルとその機能を示します。

表 7 NFS ファイル

| ファイル名                | 機能                                                                                                  | マニュアルページ                               |
|----------------------|-----------------------------------------------------------------------------------------------------|----------------------------------------|
| /etc/default/fs      | ローカルファイルシステムのデフォルトファイルシステムタイプを指定します。クライアントやサーバーでサポートされるファイルシステムタイプは、/kernel/fs でファイルを確認することで判断できます。 | <a href="#">fs(4)</a> のマニュアルページ。       |
| /etc/default/nfslogd | NFS サーバーログデーモン (nfslogd) の構成情報を指定します。                                                               | <a href="#">nfslogd(1M)</a> のマニュアルページ。 |
| /etc/dfs/dfstab      | 廃止: 共有されるローカルリソースを指定します。                                                                            | <a href="#">dfstab(4)</a> のマニュアルページ。   |
| /etc/dfs/fstypes     | リモートファイルシステムのデフォルトファイルシステムタイプを指定します。最初のエントリにデフォルトとして NFS ファイルシステムタイプを定義します。                         | <a href="#">fstypes(4)</a> のマニュアルページ。  |

| ファイル名                | 機能                                                               | マニュアルページ                                  |
|----------------------|------------------------------------------------------------------|-------------------------------------------|
| /etc/dfs/sharetab    | 共有されるローカルおよびリモートリソースを指定します。このファイルは編集しないでください。                    | <a href="#">sharetab(4)</a> のマニュアルページ。    |
| /etc/mnttab          | 自動マウントされるディレクトリを含めて、現在マウントされているファイルシステムを指定します。このファイルは編集しないでください。 | <a href="#">mnttab(4)</a> のマニュアルページ。      |
| /etc/netconfig       | トランスポートプロトコルを指定します。このファイルは編集しないでください。                            | <a href="#">netconfig(4)</a> のマニュアルページ。   |
| /etc/nfs/nfslog.conf | NFS サーバーロギングの一般構成情報を指定します。                                       | <a href="#">nfslog.conf(4)</a> のマニュアルページ。 |
| /etc/nfs/nfslogtab   | nfslogd デモンによるログ後処理の情報を指定します。このファイルは編集しないでください。                  |                                           |
| /etc/nfssec.conf     | NFS セキュリティーサービスを指定します。                                           | <a href="#">nfssec.conf(4)</a> のマニュアルページ。 |
| /etc/rmtab           | NFS クライアントによってリモートでマウントされるファイルシステムを指定します。このファイルは編集しないでください。      | <a href="#">rmtab(4)</a> のマニュアルページ。       |
| /etc/vfstab          | ローカルにマウントするファイルシステムを定義します。                                       | <a href="#">vfstab(4)</a> のマニュアルページ。      |

## /etc/default/nfslogd ファイル

このファイルは、NFS サーバーログ機能を使用するとき使用されるいくつかのパラメータを定義します。

---

**注記** - NFS Version 4 は NFS サーバーロギングをサポートしません。

---

次のパラメータを定義できます。

### CYCLE\_FREQUENCY

ログファイルを循環させる前に経過すべき時間数を決定するパラメータです。デフォルト値は 24 時間です。このオプションはログファイルが大きくなり過ぎるのを防ぐために使用します。

### IDLE\_TIME

nfslogd が、バッファファイル内のさらなる情報を検査する前にスリープすべき秒数を決定するパラメータです。このパラメータは、構成ファイルの検査頻度も決定します。このパラメータと MIN\_PROCESSING\_SIZE によりバッファファイルの処理頻度が決まります。デフォルト値は 300 秒です。この数値を増加させると、検査の回数が減ってパフォーマンスが向上します。

**MAPPING\_UPDATE\_INTERVAL**

ファイルハンドルパスマッピングテーブル内でレコードを更新する間隔を秒数で指定します。デフォルト値は 86400 秒つまり 1 日です。このパラメータは、ファイルハンドルパスマッピングテーブルを継続的に更新することなく、テーブルを最新の状態に保つのに役立ちます。

**MAX\_LOGS\_PRESERVE**

保存するログファイル数を決めます。デフォルト値は 10 です。

**MIN\_PROCESSING\_SIZE**

バッファファイルが処理してログファイルに書き込むための最小限のバイト数を設定します。このパラメータと **IDLE\_TIME** によりバッファファイルの処理頻度が決まります。デフォルト値は 524,288 バイトです。この数値を大きくするとバッファファイルの処理回数が減ってパフォーマンスが向上します。

**PRUNE\_TIMEOUT**

ファイルハンドルパスマッピングレコードを中断して削減できるようになるまでに経過しなければならない時間数を選択するパラメータです。デフォルト値は 168 時間、つまり 7 日間です。

**UMASK**

**nfslogd** によって作成されるログファイルのファイルモード生成マスクを指定します。デフォルト値は 0137 です。

## **/etc/nfs/nfslog.conf** ファイル

このファイルは **nfslogd** で使用するログのパス、ファイル名、およびタイプを定義します。各定義はタグに関連付けられています。NFS サーバーのログを開始するためには、各ファイルシステムについてタグを付ける必要があります。広域タグはデフォルト値を定義します。

---

**注記** - NFS Version 4 は NFS サーバーロギングをサポートしません。

---

必要に応じて、各タグに、次のパラメータを使用することができます。

**defaultdir=path**

ログファイルのデフォルトのディレクトリパスを指定するパラメータです。異なるように指定した場合を除いて、デフォルトディレクトリは **/var/nfs** です。

**log=path/filename**

ログファイルのパスとファイル名を指定するパラメータです。デフォルトは **/var/nfs/nfslog** です。

**fhtable=path/filename**

ファイルハンドルパスデータベースのパスとファイル名を選択するパラメータです。デフォルトは `/var/nfs/fhtable` です。

**buffer=path/filename**

バッファファイルのパスとファイル名を決定するパラメータです。デフォルトは `/var/nfs/nfslog_workbuffer` です。

**logformat=basic|extended**

ユーザーから読み取り可能なログファイルを作成するときに使用するフォーマットを選択します。基本フォーマットでは、`ftpd` デーモンに似たログファイルが作成されます。拡張フォーマットは、より詳細に表示されます。

パスが指定されていない場合は、`defaultdir` によって定義されるパスが使用されます。絶対パスを使用すると `defaultdir` をオーバーライドできます。

ファイルを識別しやすくするために、ファイルを別々のディレクトリに入れておきます。次の例は必要な変更を示しています。

**例 14** サンプル NFS サーバーロギング構成ファイル

```
cat /etc/nfs/nfslog.conf
#ident "@(#)nfslog.conf 1.5 99/02/21 SMI"
#
#
NFS server log configuration file.
#

global defaultdir=/var/nfs \
 log=nfslog fhtable=fhtable buffer=nfslog_workbuffer

publicftp log=logs/nfslog fhtable=fh/fhtables buffer=buffers/workbuffer
```

この例は、`log=publicftp` で共有されるファイルシステムを示しています。 `log=publicftp` で共有されるファイルシステムは次の値を使用します。

- デフォルトのディレクトリは `/var/nfs` です。
- ログファイルは、`/var/nfs/logs/nfslog*` に保存されます。
- ファイルハンドル-パスデータベーステーブルは `/var/nfs/fh/fhtables` に保存されます。
- バッファファイルは `/var/nfs/buffers/workbuffer` に保存されます。

NFS サーバーロギングの有効化については、[77 ページの「NFS サーバーログを有効にする方法」](#)を参照してください。

## NFS デーモン

NFS アクティビティーをサポートするために、システムが実行レベルまたはマルチユーザーモードになるときにいくつかのデーモンが起動されます。mountd デーモンおよび nfsd デーモンは、サーバーであるシステム上で実行されます。サーバーデーモンの自動起動は、少なくとも 1 つの NFS 共有が存在するかどうかで変わります。NFS 共有の現在のリストを表示するには、share -F nfs コマンドを実行します。NFS のファイルロックをサポートするために、lockd および statd デーモンが、NFS クライアントおよびサーバー上で実行されます。ただし、以前のバージョンの NFS と異なり、NFS Version 4 では、デーモン lockd、statd、および nfslogd は使用されません。

このセクションでは、次のデーモンについて説明します。

- [167 ページの「automountd デーモン」](#)
- [168 ページの「lockd デーモン」](#)
- [169 ページの「mountd デーモン」](#)
- [170 ページの「nfs4cbd デーモン」](#)
- [170 ページの「nfsd デーモン」](#)
- [171 ページの「nfslogd デーモン」](#)
- [172 ページの「nfsmapid デーモン」](#)
- [178 ページの「reparseid デーモン」](#)
- [178 ページの「statd デーモン」](#)

## automountd デーモン

automountd デーモンは autofs サービスからのマウントおよびアンマウント要求を処理します。このコマンドの構文は次のとおりです。

```
automountd [-Tnv] [-D name=value]
```

ここで

- |    |                               |
|----|-------------------------------|
| -T | トレースを有効にします。                  |
| -n | すべての autofs ノード上でブラウズを無効にします。 |
| -v | すべてのステータスメッセージログをコンソールに記録します。 |

`-D name=value` `name` で示される自動マウントマップ変数を `value` で置換します。

自動マウントマップのデフォルト値は `/etc/auto_master` です。トラブルシューティングには `-T` オプションを使用してください。

コマンド行で行うように、`sharect1` コマンドで同じ指定を行うことができます。ただし、コマンド行オプションと異なり、サービス再起動、システムリブート、およびシステムアップグレードを経由しても `SMF` リポジトリは指定を保持します。`automountd` デーモンには次のパラメータを設定できます。

#### `automountd_verbose`

ステータスメッセージをコンソールに記録します。このキーワードは `automountd` デーモンの `-v` 引数と同等です。デフォルト値は `FALSE` です。

#### `nobrowse`

すべての `autofs` マウントポイントのブラウズをオンまたはオフにします (`-automountd` の `n` 引数と同等)。デフォルト値は `FALSE` です。

#### `trace`

各リモート手続き呼び出し (RPC) を拡張し、拡張された RPC を標準出力に表示します。このキーワードは、`-automountd` の `T` 引数と同等です。デフォルト値は `0` です。値の範囲は `0` から `5` です。

#### `environment`

さまざまな値をさまざまな環境に割り当てることを許可します。このキーワードは、`-automountd` の `D` 引数と同等です。`environment` パラメータは複数回使用できます。ただし、環境割り当てごとにエントリを分けて使用する必要があります。

## lockd デーモン

`lockd` デーモンは NFS ファイルでのレコードロック操作をサポートします。`lockd` デーモンは、ネットワークロックマネージャー (NLM) プロトコルについて、クライアントとサーバー間の RPC 接続を管理します。通常は、パラメータを指定しないで起動します。使用できるオプションは 3 つあります。これらのオプションは、コマンド行から、または `sharect1` コマンドを使用してパラメータを設定することで設定できます。詳細は、[lockd\(1M\)](#) のマニュアルページを参照してください。

---

**注記** - `LOCKD_GRACE_PERIOD` キーワードと `-g` オプションは非推奨です。非推奨のキーワードは、新しい `grace_period` パラメータに置き換えられています。両方のキーワードが設定されている場合、`grace_period` の値は、`LOCKD_GRACE_PERIOD` の値をオーバーライドします。

---



LOCKD\_GRACE\_PERIOD と同様に、`grace_period=graceperiod` パラメータは、クライアントが NFS Version 3 ロック (NLM によって提供される) と NFS Version 4 ロックを再利用する必要がある、サーバーリブート後の秒数を設定します。

`lockd_retransmit_timeout=timeout` パラメータは、ロック要求をリモートサーバーに再転送する前に待機する秒数を選択します。このオプションは NFS クライアントのサービスに関係します。`timeout` のデフォルト値は 5 秒です。この値を小さくすると、トラフィックの多いネットワーク上の NFS クライアントに対する応答時間を改善できます。ただし、ロック要求が増えることによってサーバーの負荷が増す可能性があります。デーモンに `-t timeout` オプションで開始することで、同じパラメータをコマンド行から使用できます。

`lockd_servers=number` パラメータは、同時 lockd 要求の最大数を指定します。デフォルト値は 1024 です。

`nthreads` パラメータは、サーバーが処理できる同時スレッドの最大数を指定します。UDP を使用するすべての NFS クライアントは、NFS サーバーと 1 つの接続を共有します。その場合、UDP 接続が使用できるスレッドの数を増やさなければならないことがあるかもしれません。各 UDP クライアントには、少なくとも 2 つのスレッドを許可します。ただし、この数は、クライアントの負荷により異なります。そのため、クライアントごとに 2 つのスレッドを許可しても、十分ではない場合があります。多くのスレッドを使用する場合の不利な点は、これらのスレッドを使用すると、NFS サーバー上で使用するメモリの容量が増えるという点です。ただし、スレッドを使用しない場合は、`nthreads` の値を増やしても影響がありません。デーモンに `nthreads` オプションを指定して開始すると、コマンド行から同じパラメータを使用できます。

## mountd デーモン

mountd デーモンは、リモートシステムからのファイルシステムマウント要求を処理し、アクセス制御を提供します。mountd デーモンは、`/etc/dfs/sharetab` を調べて、リモートマウントに使用可能なファイルシステムと、リモートマウントを実行できるシステムを判断します。詳細は、[mountd\(1M\)](#) のマニュアルページを参照してください。

- v                    コマンドを冗長モードで実行します。クライアントが付与されるアクセス権を NFS サーバーが決定するたびに、コンソールにメッセージが表示されます。この情報は、クライアントがファイルシステムにアクセスできない理由を調べるときに役立ちます。
  
- r                    クライアントからの今後のすべてのマウント要求を拒否します。このオプションを指定しても、すでにファイルシステムがマウントされているクライアントには影響しません。

コマンド行オプション以外に、いくつかの SMF パラメータを使用して `mountd` デーモンを構成できます。

#### `client_versmin`

NFS クライアントによって使用される最小バージョン NFS プロトコルを設定します。デフォルトは 2 です。有効な値はほかに 3 と 4 があります。85 ページの「[NFS サービスの設定](#)」を参照してください。

#### `client_versmax`

NFS クライアントによって使用される最大バージョン NFS プロトコルを設定します。デフォルトは 4 です。有効な値はほかに 2 と 3 があります。85 ページの「[NFS サービスの設定](#)」を参照してください。

## nfs4cbd デーモン

`nfs4cbd` デーモンは、NFS Version 4 クライアントを排他的に使用するもので、NFS Version 4 コールバックプログラムのための通信エンドポイントを管理します。デーモンには、ユーザーがアクセス可能なインタフェースがありません。詳細は、[nfs4cbd\(1M\)](#) のマニュアルページを参照してください。

## nfsd デーモン

`nfsd` デーモンは、クライアントファイルシステム要求を処理します。このコマンドに対してはいくつかのオプションを指定できます。完全なリストについては、[nfsd\(1M\)](#) のマニュアルページを参照してください。これらのオプションは、コマンド行から使用するか、`sharectl` コマンドで適切な SMF パラメータを設定して使用することができます。

`listen_backlog=length` NFS および TCP 用の接続指向トランスポート経由接続キューの長さを設定します。デフォルト値は 32 エントリです。`nfsd` に `-l` オプションを指定して開始すると、コマンド行から同じ項目を選択できます。

`max_connections=#conn` 接続指向トランスポート単位の最大接続数を設定します。`#conn` のデフォルト値はありません。コマンド行から `-c #conn` オプションを指定してデーモンを開始すると、同じパラメータを使用できます。

`servers=nservers` サーバーが処理できる最大同時要求数を設定します。`nservers` のデフォルト値は 1024 です。`nfsd` を `nservers` オプションで開始することで、コマンド行から同じ選択を行うことができます。

以前のバージョンの `nfsd` デーモンとは異なり、現在のバージョンの `nfsd` では複数のコピーを作成して要求を同時に処理することはありません。処理テーブルを `ps` でチェックすると、動作しているデーモンのコピーが 1 つしかないことがわかります。

さらに、次の SMF パラメータを使用して `mountd` デーモンを構成できます。これらのパラメータにはコマンド行版はありません。

#### `server_versmin`

サーバーによって登録および提供される最小バージョンの NFS プロトコルを設定します。デフォルトは 2 です。有効な値はほかに 3 と 4 があります。85 ページの「[NFS サービスの設定](#)」を参照してください。

#### `server_versmax`

サーバーによって登録および提供される最大バージョンの NFS プロトコルを設定します。デフォルトは 4 です。有効な値はほかに 2 と 3 があります。85 ページの「[NFS サービスの設定](#)」を参照してください。

#### `server_delegation`

NFS Version 4 委託機能をサーバーで有効にするかどうかを制御します。この機能が有効な場合、サーバーは NFS Version 4 クライアントに委託を提供しようとし、デフォルトでは、サーバー委託は有効になっています。サーバー委託を無効にするには、86 ページの「[サーバー上で異なるバージョンの NFS を選択する方法](#)」を参照してください。詳細は、35 ページの「[NFS Version 4 における委託](#)」を参照してください。

## nfslogd デーモン

---

注記 - NFS Version 4 はこのデーモンを使用しません。

---

`nfslogd` デーモンは操作ロギングを提供します。サーバーに対して記録される NFS 操作は、`/etc/default/nfslogd` に定義されている構成オプションに基づくものです。NFS サーバーのログ機能がオンになると、選択されたファイルシステム上でのすべての RPC 操作の記録がカーネルによりバッファファイルに書き込まれます。その後、`nfslogd` がこれらの要求を後処理します。ログインおよび IP アドレスへの UID をホスト名に割り当てやすくするために、ネームサービススイッチが使用されます。識別されたネームサービスで一致するものが見つからない場合は、その番号が記録されます。

ファイルハンドルからパス名へのマッピングも `nfslogd` によって処理されます。このデーモンは、ファイルハンドルパスマッピングテーブル内でこれらの割り当てを追跡します。`/etc/nfs/nfslogd` で識別される各タグについて 1 つのマッピングテーブルが存在します。後処理の後に、レコードが ASCII ログファイルに書き込まれます。

## nfsmapid デーモン

version 4 の NFS プロトコル (RFC3530) では、クライアントとサーバーの間でユーザー識別子またはグループ識別子を交換する方法が変更されました。このプロトコルは、NFS Version 4 クライアントと NFS Version 4 サーバーとの間で、ファイルの所有者とグループ属性がそれぞれ `user@nfsv4-domain` または `group@nfsv4-domain` 形式の文字列として交換されることを要求します。

たとえば、完全修飾ホスト名が `system.example.com` の NFS Version 4 クライアント上で、ユーザー `known_user` に UID 123456 が割り当てられているとします。クライアントが NFS Version 4 サーバーに要求を行うには、クライアントは UID 123456 を `known_user@example.com` にマッピングしてから、この属性を NFS Version 4 サーバーに送信する必要があります。サーバーがクライアントから `known_user@example.com` を受け取ると、サーバーはこの文字列をローカル UID 123456 にマッピングし、これが配下のファイルシステムによって認識されます。この機能では、ネットワーク内のすべての UID と GID が一意であること、およびクライアント上の NFS Version 4 ドメインがサーバー上の NFS Version 4 ドメインと一致していることを前提としています。

NFS Version 4 クライアントおよびサーバーは、整数-文字列および文字列-整数変換の両方に対応しています。たとえば、NFS Version 4 サーバーは GETATTR 操作に応答して、配下のファイルシステムから取得された UID および GID をそれぞれの文字列表現にマッピングし、この情報をクライアントに送信します。またクライアントでも、UID と GID を文字列表現に割り当てる必要があります。たとえば、クライアントは `chown` コマンドに응答して、新しい UID および GID を文字列表現にマッピングしてから、SETATTR 操作をサーバーに送信します。

ただし、クライアントとサーバーでは、文字列が認識されない場合の対処が異なることに注意してください。

- ユーザーがサーバー上に (同じ NFS Version 4 ドメイン構成内にも) 存在しない場合は、サーバーはリモートプロシージャー呼び出し (RPC) を拒否し、クライアントにエラーメッセージを返します。このような場合は、リモートユーザーが実行できる操作が制限されます。
- ユーザーがクライアントおよびサーバーの両方上に存在するけれども、それらのドメインが一致しない場合は、サーバーが着信ユーザー文字列を配下のファイルシステムが認識できる整数値にマッピングすることを要求する属性変更操作 (SETATTR など) を、サーバーは拒否します。NFS Version 4 クライアントおよびサーバーを正常に機能させるには、それらの NFS Version 4 ドメイン (文字列の、@ 記号のあとの部分) が一致するようにしてください。
- NFS Version 4 クライアントがサーバーからのユーザーまたはグループ名を認識しない場合、クライアントは文字列をその一意 ID (整数値) にマッピングできません。そのような場合は、クライアントは着信ユーザー文字列または着信グループ文字列を `nobody` ユーザーに割り当てます。nobody に割り当てられると、さまざま

なアプリケーションでさまざまな問題が発生します。NFS Version 4 では、ファイル属性を変更する操作は失敗します。

- 指定されたユーザーまたはグループ名をサーバーが認識しない場合 (NFS Version 4 ドメインが一致している場合でも)、サーバーはユーザーまたはグループ名をその一意 ID (整数値) にマッピングできません。そのような場合は、サーバーは着信ユーザー名または着信グループ名を `nobody` ユーザーに割り当てます。このようなことが発生するのを避けるために、管理者は NFS Version 4 クライアントにのみ存在する特殊アカウントの作成を避けるようにしてください。

クライアントおよびサーバーのドメイン名は、`sharectl` コマンドを `nfsmapid_domain` オプション付きで使用することで変更できます。このオプションはクライアントとサーバーの共通ドメインを設定します。ローカル DNS ドメイン名を使用するデフォルト動作をオーバーライドします。タスク情報については、[85 ページの「NFS サービスの設定」](#)を参照してください。

## 構成ファイルと `nfsmapid` デモン

`nfsmapid` デモンは、`svc:system/name-service/switch` および `svc:/network/dns/client` 内で見つかる SMF 構成情報を次のように使用します。

- `nfsmapid` は、標準の C ライブラリ関数を使用して、バックエンドネームサービスにパスワードおよびグループ情報を要求します。これらのネームサービスは、`svc:system/name-service/switch` SMF サービスの設定によって制御されます。サービスのプロパティへの変更は、`nfsmapid` の動作に影響します。`svc:system/name-service/switch` SMF サービスの詳細は、[`nsswitch.conf\(4\)` のマニュアルページ](#)を参照してください。
- NFS Version 4 クライアントがさまざまなドメインからファイルシステムをマウントできることを保証するために、`nfsmapid` は DNS TXT リソースレコード (RR) `_nfsv4idmapdomain` の構成に依存します。`_nfsv4idmapdomain` リソースレコードの構成の詳細については、[175 ページの「`nfsmapid` と DNS TXT レコード」](#)を参照してください。また、次の点にも注意してください。
  - DNS TXT RR は、必要なドメイン情報を使って、DNS サーバー上で明示的に構成するようにしてください。
  - `svc:system/name-service/switch` SMF サービスは、`resolver` が DNS サーバーを検出してクライアントおよびサーバー NFS Version 4 ドメイン用の TXT レコードを検索できるように構成するようにしてください。

詳細については、次を参照してください。

- [174 ページの「優先順位規則」](#)
- [177 ページの「NFS Version 4 のデフォルトドメインを構成する」](#)
- [`resolv.conf\(4\)` のマニュアルページ](#)

## 優先順位規則

nfsmapid が正しく動作するには、NFS Version 4 クライアントおよびサーバーが同じドメインに割り当てられている必要があります。NFS Version 4 ドメインの一致を保証するために、nfsmapid はこれらの厳密な優先順位規則に従います。

1. デーモンは、nfsmapid\_domain パラメータに割り当てられている値を SMF リポジトリで最初に確認します。値が検出された場合、その割り当てられている値は他の設定よりも優先されます。割り当てられている値は、発信属性文字列に追加され、着信属性文字列と比較されます。手順については、[85 ページの「NFS サービスの設定」](#)を参照してください。

---

**注記 - NFSMAPID\_DOMAIN 設定の使用はスケーラブルではないため、大規模な配備には推奨されません。**

---

2. 値が nfsmapid\_domain に割り当てられていなかった場合、デーモンは DNS TXT RR からのドメイン名を確認します。nfsmapid は、resolver 内のルーチンセットによって使用される /etc/resolv.conf ファイル内の指令に依存します。resolver は、構成されている DNS サーバーから \_nfsv4idmapdomain TXT RR を検索します。DNS TXT レコードを使用する方がよりスケーラブルです。このため、SMF リポジトリにパラメータを設定するよりも、TXT レコードを継続して使用することをお勧めします。
3. ドメイン名を提供する DNS TXT レコードが構成されていない場合、nfsmapid デーモンは /etc/resolv.conf ファイル内の domain または search 指令で指定された値を使用します。このとき、最後に指定された指令が優先されます。

次の例では、domain と search の両方の指令が使用されています。nfsmapid デーモンは、search 指令のあとにリストされている最初のドメイン (example.com) を使用します。

```
domain company.example.com
search example.com abc.def.com
```

4. /etc/resolv.conf ファイルが存在しない場合、nfsmapid は domainname コマンドの動作に従って NFS Version 4 ドメイン名を取得します。具体的には、/etc/defaultdomain ファイルが存在する場合には、nfsmapid は NFS Version 4 ドメインにその内容を使用します。/etc/defaultdomain ファイルが存在しない場合には、nfsmapid はネットワークに構成されているネームサービスから渡されるドメイン名を使用します。詳細は、[domainname\(1M\)](#) のマニュアルページを参照してください。

## nfsmapid と DNS TXT レコード

DNS は広く普及しているので、NFS Version 4 ドメイン名にとって効率的な保存および配布メカニズムです。さらに、DNS は本質的に拡張性が高いので、大規模に配備するために NFS Version 4 ドメインを構成する場合には DNS TXT リソースレコードを使用することが推奨されます。エンタープライズレベル DNS サーバーでは、`_nfsv4idmapdomain` TXT レコードを構成することをお勧めします。このような構成することで、NFS Version 4 クライアントまたはサーバーが DNS ツリーをトラバースすればその NFS Version 4 ドメインを見つけられることが保証されます。

次の例は、DNS サーバーが NFS Version 4 ドメイン名を提供するために推奨されるエントリを示しています。

```
_nfsv4idmapdomain IN TXT "abc.def"
```

この例では、構成するドメイン名は二重引用符で囲まれている値です。tt1 フィールドの指定がなく、`_nfsv4idmapdomain` にドメインが付加されていないことから、これは owner フィールドの値です。この構成により、TXT レコードは Start-Of-Authority (SOA) レコードからゾーンの `_${ORIGIN}` エントリを使用できます。たとえば、さまざまなレベルのドメイン名前空間で、レコードは次のように読み取ることができます。

```
_nfsv4idmapdomain.subnet.example.com. IN TXT "abc.def"
_nfsv4idmapdomain.example.com. IN TXT "abc.def"
```

この構成では、DNS クライアントが DNS ツリー階層を検索するときに、`resolv.conf` ファイルを使用して柔軟に検索することができます。[resolv.conf\(4\)](#) のマニュアルページを参照してください。この機能により、TXT レコードの検索での確率がより高くなります。さらに柔軟性を高めるために、低レベル DNS サブドメインは独自の DNS TXT リソースレコード (RR) を定義できます。この機能により、低レベル DNS サブドメインは高レベル DNS ドメインで定義された TXT レコードをオーバーライドできます。

---

**注記** - TXT レコードで指定されたドメインは、NFS Version 4 を使用するクライアントとサーバーの DNS ドメインと一致している必要がない、任意の文字列を使用できます。NFS Version 4 データをほかの DNS ドメインと共有しないこともできます。

---

## NFS Version 4 のドメインを確認する

ネットワークの NFS Version 4 ドメインに値を割り当てる前に、NFS Version 4 ドメインがネットワーク用にすでに構成されているかどうかを確認してください。次の例は、ネットワークの NFS Version 4 ドメインを識別する方法を示しています。

- NFS Version 4 ドメインを DNS TXT RR から識別するには、`nslookup` または `dig` コマンドを使用します。

次の例は、nslookup コマンドのサンプル出力を示しています。

```
nslookup -q=txt _nfsv4idmapdomain
Server: 10.255.255.255
Address: 10.255.255.255#53

_nfsv4idmapdomain.company.example.com text = "example.com"
```

次の例は、dig コマンドのサンプル出力を示しています。

```
dig +domain=company.example.com -t TXT _nfsv4idmapdomain
...
;; QUESTION SECTION:
;_nfsv4idmapdomain.company.example.com. IN TXT

;; ANSWER SECTION:
_nfsv4idmapdomain.company.example.com. 21600 IN TXT "example.com"

;; AUTHORITY SECTION:
...
```

DNS TXT RR の設定方法については、[175 ページの「nfsmapid と DNS TXT レコード」](#)を参照してください。

- ネットワークが NFS Version 4 DNS TXT RR で設定されていない場合は、次のコマンドを使用して NFS Version 4 ドメインを DNS ドメイン名から識別します。

```
egrep domain /etc/resolv.conf
domain company.example.com
```

- /etc/resolv.conf ファイルがクライアントの DNS ドメイン名を提供するように構成されていない場合は、次のコマンドを使用してネットワークの NFS Version 4 ドメイン構成からドメインを識別します。

```
cat /system/volatile/nfs4_domain
example.com
```

- NIS などの別のネームサービスを使用している場合は、次のコマンドを使用して、ネットワークに構成されているネームサービスでドメインを確認します。

```
domainname
it.company.example.com
```

詳細は、次のマニュアルページを参照してください。

- [nslookup\(1M\)](#)
- [dig\(1M\)](#)
- [resolv.conf\(4\)](#)
- [domainname\(1M\)](#)



## NFS Version 4 のデフォルトドメインを構成する

このセクションでは、ネットワークがどのようにして目的のデフォルトドメインを取得するかについて説明します。

- 最新リリースについては、[177 ページの「Oracle Solaris 11 リリースで NFS Version 4 のデフォルトドメインを構成する」](#)を参照してください。
- 初期 Solaris 10 リリースの場合は、[177 ページの「Solaris 10 リリースで NFS Version 4 のデフォルトドメインを構成する」](#)を参照してください。

## Oracle Solaris 11 リリースで NFS Version 4 のデフォルトドメインを構成する

Oracle Solaris 11 リリースでは、次のコマンドを入力することでデフォルト NFS ドメインバージョンを設定します。

```
sharectl set -p nfsmapid_domain=example.com nfs
```

---

**注記** - DNS は広く普及していて拡張性が高いため、大規模な NFS Version 4 配備のドメインを構成する場合には DNS TXT レコードを使用することが引き続き推奨され、強く奨励されています。[175 ページの「nfsmapid と DNS TXT レコード」](#)を参照してください。

---

## Solaris 10 リリースで NFS Version 4 のデフォルトドメインを構成する

NFS Version 4 の初期 Solaris 10 リリースでは、ネットワークが複数の DNS ドメインを含んでいるけれども、単一 UID および GID 名前空間しか持たない場合には、すべてのクライアントが `nfsmapid_domain` に 1 つの値を使用する必要がありました。DNS を使用するサイトでは、`nfsmapid` が、`_nfsv4idmapdomain` に割り当てられた値からドメイン名を取得して、この問題を解決します。詳細は、[175 ページの「nfsmapid と DNS TXT レコード」](#)を参照してください。ネットワークが DNS を使用するよう構成されていない場合は、最初のシステムブート時に、オペレーティングシステムは `sysidconfig` ユーティリティーを使用して NFS Version 4 ドメイン名のために次のプロンプトを提供します。

```
This system is configured with NFS Version 4, which uses a domain name that is automatically derived from the system's name services. The derived domain name is sufficient for most configurations. In a few cases, mounts that cross different domains might cause files to be owned by nobody due to the lack of a common domain name.
```

```
Do you need to override the system's default NFS version 4 domain name (yes/no)? [no]
```

デフォルトの応答は `[no]` です。`[no]` を選択すると、次のメッセージが表示されません。

For more information about how the NFS Version 4 default domain name is derived and its impact, refer to the man pages for `nfsmapid(1M)` and `nfs(4)`, and the System Administration Guide: Network Services.

[yes] を選択すると、次のプロンプトが表示されます。

```
Enter the domain to be used as the NFS Version 4 domain name.
NFS Version 4 domain name []:
```

---

注記 - `nfsmapid_domain` の値が SMF リポジトリに存在する場合は、提供するドメイン名がその値をオーバーライドします。

---

## nfsmapid の追加情報

nfsmapid の詳細は、次を参照してください。

- [nfsmapid\(1M\)](#) のマニュアルページ
- [nfs\(4\)](#) のマニュアルページ
- <http://www.ietf.org/rfc/rfc1464.txt>
- 37 ページの「NFS Version 4 での ACL と nfsmapid」

## reparsed デーモン

reparsed デーモンは再解析ポイントに関連付けられたデータを解釈します。これらのポイントは、SMB および NFS ファイルサーバーで DFS および NFS リフェラルによって使用されます。このサービスは SMF によって管理されるため、手動で起動しないようにしてください。

## statd デーモン

---

注記 - NFS Version 4 はこのデーモンを使用しません。

---

statd デーモンは lockd と連携し、ロックマネージャーにクラッシュおよび回復機能を提供します。statd デーモンは、NFS サーバーにロックを保持するクライアントを追跡します。サーバーがクラッシュした場合は、リブート時に、サーバー上の statd はクライアント上の statd に連絡します。次にクライアント側 statd は、サーバー上のすべてのロックを再要求します。クライアント statd も、クライアントがクラッシュしたときに、サーバー上でクライアントのロックを解除できるようにサーバー

statd に通知します。このデーモンにオプションはありません。詳細は、[statd\(1M\)](#) のマニュアルページを参照してください。



# 索引

---

## 数字・記号

\* (アスタリスク)

autofs マップの中の, 73

# (ポンド記号)

間接マップのコメント, 59

直接マップのコメント, 58

マスターマップ (auto\_master) のコメント, 56, 56, 56

+ (プラス記号)

autofs マップ名の中の, 68, 68, 69, 69

## あ

アクセス

NFS リフェラル, 94

アクセス権

NFS Version 3 の改善, 16

アクセス制御リスト (ACL) と NFS

エラーメッセージ、Permission denied, 160

説明, 18, 37

アスタリスク (\*)

autofs マップ, 73

アプリケーション

ハングアップ, 159

アンパサンド (&)

autofs マップの中の, 72

アンマウント

autofs および, 62

autofs と, 14

ファイルシステムのグループ, 125

ミラーマウントおよび, 51

例, 124

委託

NFS Version 4, 35

エラーメッセージ

automount -v コマンドで生成される, 152

No such file or directory, 158

Permission denied, 158

server not responding

キーボード割り込み, 139

ハングアップしたプログラム, 159

リモートマウントの問題, 158, 159

オープンエラー

NFS および, 16

書き込みエラー

NFS および, 16

その他の automount メッセージ, 153

大きなファイル

NFS サポート, 20

オープンエラー

NFS および, 16

オペレーティングシステム

非互換バージョンのサポート, 109

マップ変数, 67

## か

カーネル

サーバーでの応答の確認, 147

階層型マウント (複数マウント), 62

書き込みエラー

NFS および, 16

間接マップ (autofs)

いつ実行するか automount コマンド, 102

概要, 59, 60

構文, 59, 59

コメント, 59

説明, 101

例, 59, 60

キーワード

- NFS バージョンネゴシエーション, 28
- 揮発性ファイルハンドル
  - NFS Version 4, 32
- キャッシュと NFS Version 3, 16
- 共有 参照 ファイル共有
- 共有解除と再共有
  - NFS Version 4, 29
- クライアント回復
  - NFS Version 4, 33
- クライアント側フェイルオーバー
  - NFS Version 4 における, 43
  - NFS サポート, 20
  - NFS ロックおよび, 43
  - 概要, 41
  - 複製されたファイルシステム, 42
  - 有効化, 81
- 公開鍵暗号手法
  - DH 認証, 48
  - 公開鍵のデータベース, 47, 48
  - 時間同期, 49
  - 秘密鍵
    - データベース, 48
    - リモートサーバーからの削除, 49
- 公開鍵暗号方式
  - DH 認証, 49
  - 共通鍵, 49
  - 対話鍵, 49
- 公開鍵マップ
  - DH 認証, 48
- 公開ファイルハンドル
  - autofs および, 111
  - NFS マウント, 21
  - WebNFS および, 91
  - マウントおよび, 41
- コマンド
  - FedFS, 136
  - NFS, 115
  - ハングアップしたプログラム, 159
- コメント
  - 間接マップ, 59
  - 直接マップ, 58
  - マスターマップ (auto\_master) の, 56

## さ

- / (スラッシュ)
  - / が前に付いたマスターマップ名, 55, 55
  - マスターマップのマウントポイント /-, 55, 58
  - ルートディレクトリ
    - ディスクレスクライアントによるマウント, 14
- サーバー, 110
  - 参照 NFS サーバー
  - autofs によるファイルの選択, 63
  - NFS サーバーおよび vfstab ファイル, 79
  - NFS サービス, 15
  - クラッシュおよび秘密鍵, 49, 49
  - ホームディレクトリサーバー設定, 105
- サーバーとクライアント
  - NFS サービス, 15
- 削除
  - NFS リフェラル, 52, 95
- 作成
  - NFS リフェラル, 52, 94, 97
  - セキュアな接続 (FedFS), 97
  - 名前空間データベース (FedFS), 96
- 資格
  - UNIX 認証, 48
  - 説明, 48
- 時間同期, 49
- 時間の同期, 49
- 実行可能なマップ, 69
- 出力
  - 共有またはエクスポートされたファイルの一覧表示, 135
- シリアルアンマウント, 125
- シングルユーザーモードとセキュリティー, 49
- スーパーユーザー
  - autofs とパスワード, 14
- すでにマウントされたファイルシステムをオーバーレイ, 122
- スラッシュ (/)
  - ルートディレクトリ、ディスクレスクライアントによるマウント, 14
- 制限
  - 表示されるファイルシステム情報, 84
- セキュリティー
  - autofs 制限の適用, 110
  - DH 認証

- 概要, 48, 49
  - パスワード保護, 47
  - ユーザー認証, 47
  - NFS Version 3 および, 16
  - Secure NFS システム
    - 概要, 47
    - 管理, 89
  - Secure RPC
    - DH 認証の問題, 49, 50
    - 概要, 47
  - UNIX 認証, 47, 48
  - ファイル共有の問題, 129, 131
  - セキュリティと NFS
    - エラーメッセージ、Permission denied, 160
    - 説明, 18, 37
  - セキュリティフレーバー, 21
  - セキュリティモード選択と mount コマンド, 121
  - 設定
    - nobrowse パラメータ, 111
- た**
- (ダッシュ)
    - autofs マップ名の中の, 68, 68
  - 対話鍵, 49
  - 直接入出力マウントオプション, 119
  - 直接マップ (autofs)
    - いつ実行するか automount コマンド, 102
    - 概要, 58
    - 構文, 57
    - コメント, 58
    - 説明, 101
    - 例, 57
  - ディスククライアント
    - 手動マウント要件, 14
    - ブートプロセス中のセキュリティ, 49
  - デーモン
    - automountd, 167
      - autofs と, 14
      - 概要, 53, 53
    - lockd, 168
    - mountd, 169
      - rpcbind デーモンに登録されていない, 158
      - サーバーでの応答の確認, 148
      - 動作しているかどうかの確認, 150, 158
    - nfs4cbd, 170
    - nfsd
      - サーバーでの応答の確認, 148
      - 説明, 170
      - 動作しているかどうかの確認, 150
    - nfslogd, 171
    - nfsmapid, 172
    - reparsed, 178
    - rpcbind
      - マウントエラーメッセージ, 158, 158
    - statd, 178
    - リモートマウントの要件, 139
  - ドメイン
    - 定義, 89
  - ドメイン名
    - Secure NFS システムおよび, 89
  - トラブルシューティング
    - autofs, 152
      - automount -v コマンドで生成されるエラーメッセージ, 152
      - その他のエラーメッセージ, 153
      - マウントポイント競合の回避, 103
    - NFS
      - NFS サービスが失敗した場所を特定する, 150
      - サーバーの問題, 147
      - ハングアップしたプログラム, 159
      - 方法, 139
      - リモートマウントの問題, 146, 158
  - トランスポートプロトコル
    - NFS ネゴシエーション, 39
- な**
- 名前空間
    - autofs および, 22
    - 共有へのアクセス, 108
  - 認証
    - DH, 49
    - Diffie-Hellman (DH), 48
    - RPC, 48
    - UNIX, 47, 48
  - ネームサービス
    - autofs マップ保守方法, 101
  - ネゴシエーション

WebNFS セキュリティー, 22  
ファイル転送サイズ, 39  
ネットワークロックマネージャー, 20

## は

バージョンネゴシエーション

NFS, 28

パスワード

autofs とスーパーユーザーパスワード, 14

DH パスワード保護, 47

バックグラウンドファイルマウントオプション,  
118

ハングアップしたプログラム, 159

秘密鍵

サーバーのクラッシュおよび, 49, 49

データベース, 48

リモートサーバーからの削除, 49

表示

共有されるファイルシステム, 132

制限されたファイルシステム情報, 84

マウント可能なファイルシステム, 84

リモートマウントされたディレクトリの一覧表  
示, 135

リモートマウントされたファイルシステムを持  
つクライアント, 135

ファイアウォール

NFS アクセス, 21

経由でファイルシステムをマウント, 82

経由の WebNFS アクセス, 94

ファイルアクセス権

NFS Version 3 の改善, 16

WebNFS および, 92

ファイルおよびファイルシステム

autofs アクセス

非 NFS ファイルシステム, 104

autofs によるファイルの選択, 63, 66

NFS ファイルとその機能, 163

プロジェクト関連ファイルの統合, 106

リモートファイルシステム

グループのアンマウント, 125

ファイルシステムテーブルからのマウント,  
125

リモートマウントされたファイルシステム  
を持つクライアントの一覧表示, 135

ローカルファイルシステム

グループのアンマウント, 125

ファイル共有

NFS Version 3 の改善, 16

NFS Version 3 の改良点, 20

root アクセスの付与, 131

概要, 128

共有解除, 134

自動, 76

セキュリティの問題, 47, 129, 131

認証されていないユーザーおよび, 130

複数のサーバーにまたがる共有ファイルを複製  
する, 110

複数のファイルシステム, 134

読み書きアクセス, 129, 132

読み取り専用アクセス, 129, 129, 132

リストされたクライアントのみ, 129

例, 132

ファイル共有オプション, 129

ファイルシステムと NFS, 15

ファイルシステム名前空間

NFS Version 4, 30

ファイルシステムの共有解除

unshareall コマンド, 134

unshare コマンド, 134

ファイルシステムのマウント

1つのクライアントへのアクセスの無効化, 82

1つのサーバーからすべてをマウント, 81

autofs および, 80

NFS URL, 83

概要, 78

手動 (オンザフライ), 80

タスクマップ, 78

ファイアウォール経由, 82

ブート時の方法, 79

ファイル属性と NFS Version 3, 16

ファイル転送サイズ

ネゴシエーション, 39

ファイルとファイルシステム

NFS による扱い, 15, 16

定義, 16

ファイルシステム, 16

ブート

ディスククライアントのセキュリティ,  
49



- ファイルシステムのマウント, 79
  - フェイルオーバー
    - mount コマンドの例, 122, 122
    - NFS サポート, 20
    - エラーメッセージ, 157
  - フェデレーテッドファイルシステム 参照 FedFS
  - フォアグラウンドファイルマウントオプション, 118
  - 複数のサーバーにまたがる共有ファイルを複製する, 110
  - 複製されたファイルシステム, 42
  - ブラウザビリティ
    - 概要, 22
    - 無効化, 111
  - ブラウザ
    - NFS URL, 93
  - プログラム
    - ハングアップ, 159
  - プロジェクト
    - ファイルの統合, 106
  - プロジェクト関連ファイルの統合, 106
  - プロセッサタイプマップ変数, 67
  - ベリファイア
    - RPC 認証システム, 48
  - 変更
    - NFS リフェラル, 95
  - ポートマッパー
    - マウントおよび, 40
  - ホスト
    - すべてのファイルシステムのアンマウント, 125
  - ポンド記号 (#)
    - 間接マップのコメント, 59, 59
    - 直接マップのコメント, 58, 58
- ま**
- マウント
    - autofs および, 62
    - autofs と, 14
    - FedFS, 95
    - nfsd デーモンおよび, 40
    - キーボード割り込み, 139
    - 強制直接入出力, 119
    - 公開ファイルハンドルおよび, 41
    - すでにマウントされたファイルシステムをオーバーレイ, 122
    - ソフトとハードの比較, 140
    - ディスクレスクライアント要件, 14
    - テーブル内のすべてのファイルシステム, 125
    - バックグラウンド再試行, 118
    - フォアグラウンド再試行, 118
    - ポートマッパーおよび, 40
    - ミラーマウントおよび, 50
    - 読み書き指定, 121
    - 読み取り専用指定, 121, 122
    - リモートマウント
      - トラブルシューティング, 147, 150
      - 必要とされるデーモン, 139
    - 例, 122
    - マウントのキーボード割り込み, 139
    - マウント不可メッセージ, 152
    - マウントポイント
      - /home, 55, 56
      - /net, 56
      - /nfs4, 55, 57
      - 競合回避, 103
      - マスターマップのマウントポイント /-, 55, 58
    - マスターマップ (auto\_master)
      - /etc/mnttab ファイルとの比較, 54
      - automount コマンドをいつ実行するか, 102
      - インストール済み, 104
    - 概要, 55, 55
    - 構文, 55
    - コメント, 56
    - セキュリティ制限, 110
    - 説明, 101
    - 内容, 55, 57
    - マウントポイント /-, 55, 58
  - マッピングされていないユーザーまたはグループ ID
    - 確認, 38
  - マッピングされていないユーザーまたはグループ ID の確認, 38
  - マップエントリ内の変数, 67, 68
  - マップ内のサーバーの重み付け, 67
  - マップの中の特殊文字
    - 引用符記号で囲む, 73
  - マップのバックスラッシュ (\), 56, 58, 59
  - マップを使用したナビゲーション

- 概要, 54
  - プロセスの開始, 56, 60
  - マップ (autofs)
    - automount コマンド
      - いつ実行するか, 102
    - 間接, 59, 60
    - 管理タスク, 101
    - クライアント用の読み取り専用ファイルの選択, 66
    - クライアント用読み取り専用ファイルの選択, 63
    - コメント, 56, 58, 59
    - 実行可能な, 69
    - タイプとその使用法, 101
    - 直接, 57, 58
    - 特殊文字, 73
    - 長い行の分割, 56, 58, 59
    - ナビゲーションプロセスの開始, 56, 60
    - ネットワークナビゲーション, 54
    - 複数マウント, 62
    - 変数, 67, 68
    - ほかのマップの参照, 69
    - ほかのマップへの参照, 68
    - 保守方法, 101
    - マウント競合の回避, 103
    - マスター, 55, 55
  - ミラーマウント
    - 1つのサーバーからすべてのファイルシステムをマウント, 81
    - 概要, 50
  - 無効化
    - 1つのクライアントへのマウントアクセス, 82
    - autofs ブラウザビリティ
      - 概要, 111
- や**
- 有効化
    - NFS サーバーロギング, 77
    - WebNFS サービス, 92
    - クライアント側フェイルオーバー, 81
  - 読み書きタイプ
    - ファイルシステムの共有, 129, 132
    - ファイルシステムのマウント, 121
  - 読み取り専用アクセス
    - ファイルシステムの共有, 129
    - 読み取り専用タイプ
      - autofs によるファイルの選択, 63, 66
      - ファイルシステムの共有, 129, 132
      - ファイルシステムのマウント, 121, 122
- ら**
- リスト
    - マウントされたファイルシステム, 124
  - リフェラル 参照 NFS リフェラル
  - リモート手続き呼び出し (RPC)
    - Secure
      - 概要, 47
  - リモートファイルシステム
    - グループのアンマウント, 125
    - リモートマウントされたファイルシステムを持つクライアントの一覧表示, 135
  - リモートマウント
    - トラブルシューティング, 146, 150
    - 必要とされるデーモン, 139
  - ルートディレクトリ
    - ディスクレスクライアントによるマウント, 14
  - ローカルキャッシュと NFS Version 3, 16
  - ローカルファイル
    - autofs マップの更新, 101
  - ローカルファイルシステム
    - グループのアンマウント, 125
  - ロック
    - NFS Version 3 の改善, 20
    - ロックの削除, 117
- A**
- a オプション
    - showmount コマンド, 135
    - umount コマンド, 124
  - already mounted メッセージ, 153
  - & (アンパサンド)
    - autofs マップの中の, 72
  - anon オプション
    - share コマンド, 130
  - ARCH マップ変数, 67
  - auto\_home マップ
    - /home ディレクトリ, 104

- /home ディレクトリサーバー設定, 105
  - /home マウントポイント, 55, 56
  - auto\_master ファイル
    - nobrowse オプション, 112
  - autofs
    - /home ディレクトリ, 104
    - NFS URL および, 111
    - nobrowse オプション, 112
    - アンマウントプロセス, 62
    - 概要, 14
    - 機能, 22
    - 共有名前空間へのアクセス, 108
    - 公開ファイルハンドルおよび, 111
    - 参照情報, 73
    - 特殊文字, 73
    - トラブルシューティング, 152
    - 名前空間データ, 22
    - 非 NFS ファイルシステムへのアクセス, 104
    - 非互換クライアント OS のサポート, 109
    - ファイルシステムのマウント, 80
    - 複数のサーバーにまたがる共有ファイルを複製する, 110
    - ブラウザビリティ, 22, 111
    - プロジェクト関連ファイルの統合, 106
    - ホームディレクトリサーバー設定, 105
    - マウントプロセス, 61, 62
    - マップ
      - 間接, 59, 60
      - タイプ, 101
      - 直接, 57, 58
      - ナビゲーションプロセスの開始, 56, 60
      - ネットワークナビゲーション, 54
      - ブラウザビリティおよび, 22
      - 変数, 67, 68
      - ほかのマップの参照, 69
      - ほかのマップへの参照, 68
      - マスター, 55, 55
      - 読み取り専用ファイルの選択, 63, 66
    - マップの管理, 101
    - メタキャラクタ, 72, 73
    - リファレンス情報, 72
  - autofs マップの \ (バックスラッシュ), 56, 58, 59
  - automount コマンド, 116
    - autofs と, 14
    - autofs マスターマップ (auto\_master) の変更, 102
    - v オプション, 152
    - いつ実行するか, 102
    - エラーメッセージ, 152
    - 概要, 53
  - automountd デーモン, 167
    - autofs と, 14
    - 概要, 53, 53
    - 説明, 22
    - マウントおよび, 22
- B**
- bad argument specified with index option, 156
  - bad key メッセージ, 152
  - bg オプション
    - mount コマンド, 118
- C**
- cannot receive reply メッセージ, 155
  - cannot send packet メッセージ, 155
  - clear\_locks コマンド, 117
  - client\_versmax パラメータ, 170
  - client\_versmin パラメータ, 170
  - could not use public filehandle メッセージ, 157
  - couldn't create mount point メッセージ, 152
  - CPU マップ変数, 67
- D**
- d オプション
    - showmount コマンド, 135
  - daemon running already メッセージ, 157
  - DH 認証
    - Secure NFS および, 89
    - 概要, 48, 49
    - パスワード保護, 47
    - ユーザー認証, 47
  - dir must start with '/' メッセージ, 153
  - DNS レコード
    - FedFS, 95

**E**

/etc/default/autofs ファイル  
autofs 環境を構成する, 100  
/etc/default/nfslogd ファイル, 164  
/etc/mnttab ファイル  
auto\_master マップとの比較, 54  
/etc/netconfig ファイル, 164  
/etc/nfs/nfslog.conf ファイル, 165  
/etc/services ファイル  
nfsd エントリ, 156  
/etc/vfstab ファイル  
automount コマンドおよび, 54  
NFS サーバーおよび, 79  
クライアント側フェイルオーバーの有効化, 82  
ディスクレスクライアント, 14  
ブート時のファイルシステムのマウント, 79  
-e オプション  
showmount コマンド, 135  
error checking メッセージ, 157  
error locking メッセージ, 157

**F**

-F オプション  
unshareall コマンド, 134  
FedFS  
DNS レコード, 95  
LDAP スキーマ, 96  
管理, 95  
マウント, 95  
マウントポイント, 57  
FedFS コマンド, 136  
fg オプション  
mount コマンド, 118  
file too large メッセージ, 157  
forcedirectio オプション  
mount コマンド, 119  
ftp アーカイブ  
WebNFS および, 92  
fuser コマンド  
umountall コマンドと, 125

**G**

-g オプション

lockd デーモン, 168  
grace\_period パラメータ  
lockd デーモン, 169  
GSS-API  
および NFS, 21

**H**

/home ディレクトリおよび NFS サーバー設定,  
105  
/home マウントポイント, 55, 56  
-h オプション  
umountall コマンド, 125  
hard オプション  
mount コマンド, 121  
hierarchical mount points メッセージ, 154  
host not responding メッセージ, 154  
HOST マップ変数, 67  
HTML ファイル  
WebNFS および, 92  
httpd コマンド  
ファイアウォールアクセスおよび WebNFS, 94

**I**

ID マッピングの失敗  
理由, 38  
index オプション  
bad argument エラーメッセージ, 156  
share コマンド, 93  
WebNFS および, 92  
-intr オプション  
mount コマンド, 139

**K**

/kernel/fs ファイル  
確認, 163  
-k オプション  
umountall コマンド, 125  
KERB 認証  
NFS と, 20  
keylogin コマンド  
リモートログインのセキュリティ問題, 50  
keylogout コマンド

Secure NFS および, 50

## L

-l オプション

umountall コマンド, 125

largefiles オプション

mount コマンド, 119

エラーメッセージ, 159

LDAP スキーマ

FedFS 用, 96

leading space in map entry メッセージ, 153

LOCKD\_GRACE\_PERIOD パラメータ

lockd デーモン, 168

lockd\_retransmit\_timeout パラメータ

lockd デーモン, 169

lockd\_servers パラメータ

lockd デーモン, 169

lockd デーモン, 168

log オプション

share コマンド, 130

login コマンド

Secure NFS および, 50

ls コマンド

ACL エントリおよび, 38

## M

map key bad メッセージ, 154

mnttab ファイル

auto\_master マップとの比較, 54

mount of server:pathname エラー, 154

mount コマンド, 118

autofs と, 14

NFS URL, 84, 122

オプション

public, 83

説明, 118

引数なし, 124

使用, 122

ディスククライアントが必要なとき, 14

ファイルシステムの手動マウント, 80

フェイルオーバー, 122, 122

mountall コマンド, 125

mountd デーモン, 169

rpcbind に登録されていない, 158

サーバーでの応答の確認, 148

動作しているかどうかの確認, 150, 158

## N

/net マウントポイント, 56

/nfs4 マウントポイント, 55, 57

netconfig ファイル

説明, 164

nfs4cbd デーモン, 170

NFS ACL

エラーメッセージ、Permission denied, 160

説明, 18, 37

NFS can't support nolargefiles メッセージ, 159

NFS

コマンド, 115

デーモン, 167

バージョンネゴシエーション, 28

NFS URL

autofs および, 111

mount コマンド例, 122

WebNFS および, 91

構文, 93

ファイルシステムのマウント, 83

マウント, 21

NFS V2 can't support largefiles メッセージ, 159

NFS Version 4

機能, 29

NFS 環境

Secure NFS システム, 47

NFS 管理

管理者の責任, 75, 99

NFS クライアント

NFS サービス, 15

非互換オペレーティングシステムのサポート, 109

NFS サーバー

autofs によるファイルの選択, 66

共有ファイルを複製する, 110

現在の識別, 151

トラブルシューティング

問題の解決, 147

- リモートマウントの問題, 146, 158
  - 保守, 75, 99
  - マップの重み付け, 67
  - リモートマウントで必要とされるデーモン, 139
  - NFS サーバーロギング
    - 概要, 22
    - 有効化, 77
  - NFS サービス
    - クライアント上で異なるバージョンを選択する
      - mount コマンドの使用, 88
      - SMF プロパティの変更, 87
    - サーバー上で異なるバージョンを選択する, 86
    - 再起動, 150
    - タスクマップ, 85
  - NFS での ACL の問題
    - 回避, 38
  - NFS での ACL の問題を回避する, 38
  - NFS トラブルシューティング
    - NFS サービスが失敗した場所を判断する, 150
    - サーバーの問題, 147
    - ハングアップしたプログラム, 159
    - 方法, 139
    - リモートマウントの問題, 158
  - NFS リフェラル
    - 概要, 52
    - 削除, 95
    - 作成, 94, 97
  - NFS ロック
    - クライアント側フェイルオーバーおよび, 43
  - nfscast: cannot receive reply メッセージ, 155
  - nfscast: cannot send packet メッセージ, 155
  - nfscast: select メッセージ, 155
  - nfstd デーモン, 170
    - サーバーでの応答の確認, 148
    - 動作しているかどうかの確認, 150
    - マウントおよび, 40
  - nfslog.conf ファイル, 165
  - nfslogd デーモン
    - 説明, 171
  - nfslogd ファイル, 164
  - NFSMAPID\_DOMAIN キーワード, 38
  - nfsmapid\_domain パラメータ, 173
  - nfsmapid デーモン
    - ACL および, 37
    - DNS TXT レコードと, 175
    - NFSv4 デフォルトドメインの構成, 177
    - NFSv4 ドメインの確認, 175
    - 構成ファイルと, 173
    - 説明, 17, 172
    - 追加情報, 178
    - 優先ルールと, 174
  - nfsref コマンド
    - 説明, 136
    - 例, 97
  - nfsstat コマンド, 140, 151
  - NIS ネームサービス
    - autofs マップの更新, 101
  - no info メッセージ, 155
  - No such file or directory メッセージ, 158
  - nobrowse オプション
    - auto\_master ファイル, 112
  - nobrowse パラメータ
    - 設定, 111
  - nolargefiles オプション
    - mount コマンド, 119
    - エラーメッセージ, 159
  - nosuid オプション
    - share コマンド, 130
  - Not a directory メッセージ, 154
  - Not found メッセージ, 153
  - nsdb-list コマンド
    - 説明, 136
  - nsdb-nces コマンド
    - 説明, 136
  - nsdb-resolve-fsn コマンド
    - 説明, 137
  - nsdb-update-nci コマンド
    - 説明, 137
    - 例, 96
  - nsdbparams コマンド
    - 説明, 137
    - 例, 97
  - nthreads オプション
    - lockd デーモン, 169
- O**
- o オプション
    - mount コマンド, 122

share コマンド, 129, 132  
-o オプション  
  mount コマンド, 122  
OPEN 共有サポート  
  NFS Version 4, 35  
OSNAME マップ変数, 67  
OSREL マップ変数, 67  
OSVERS マップ変数, 67

## P

pathconf: server not responding メッセージ, 155  
pathconf: no info メッセージ, 155  
Permission denied メッセージ, 158  
pstack コマンド, 142  
public オプション  
  dfstab ファイルの, 93  
  mount コマンド, 83, 121  
  share エラーメッセージ, 161  
  WebNFS および, 92

## R

-r オプション  
  mount コマンド, 122  
  umountall コマンド, 125  
remount メッセージ, 153  
repared デーモン, 178  
replicas must have the same version メッセージ, 160  
replicated mounts must be read-only メッセージ, 161  
replicated mounts must not be soft メッセージ, 161  
replicated mounts  
  soft オプションおよび, 161  
rlogin コマンド  
  Secure NFS および, 50  
ro オプション  
  mount コマンド, 121  
  -o フラグ付きの mount コマンド, 122  
  share コマンド, 129, 132  
root オプション

share コマンド, 131  
RPC  
  Secure  
    DH 認証の問題, 49, 50  
    認証, 48  
  rpcbind デーモン  
    mountd デーモンが登録されていない, 158  
    死んでいるまたはハングアップ, 158  
  rpcinfo コマンド, 143  
  RPCSEC\_GSS, 21  
  rw=client オプション  
    umountall コマンド, 129  
  rw オプション  
    mount コマンド, 121  
    share コマンド, 129, 132

## S

-s オプション  
  umountall コマンド, 125  
Secure NFS システム  
  DH 認証および, 89  
  概要, 47  
  管理, 89  
  ドメイン名, 89  
Secure RPC  
  DH 認証の問題, 49, 50  
  概要, 47  
server\_delegation パラメータ, 171  
server\_versmax パラメータ, 171  
server\_versmin パラメータ, 171  
server not responding メッセージ, 153, 155  
  キーボード割り込み, 139  
  ハングアップしたプログラム, 159  
  リモートマウントの問題, 158  
setfacl コマンド  
  NFS, 37  
setgid モード  
  share コマンド, 130  
setuid モード  
  Secure RPC および, 49  
  share コマンド, 130  
share コマンド  
  WebNFS サービスの有効化, 93  
  オプション, 129

セキュリティーの問題, 131  
 説明, 128  
 shareall コマンド, 134  
 showmount\_info プロパティ, 84  
 showmount コマンド, 135  
 例, 84  
 slash (/)  
 マスターマップのマウントポイント /, 55, 58  
 snoop コマンド, 145  
 soft オプション  
 mount コマンド, 121  
 statd デーモン, 178

## T

-t オプション  
 lockd デーモン, 169  
 TCP  
 NFS Version 3 と, 19  
 telnet コマンド  
 Secure NFS および, 50  
 transport setup problem  
 エラーメッセージ, 156  
 truss コマンド, 145

## U

/usr/kvm ディレクトリ  
 ディスクレスクライアントによるマウント, 14  
 /usr/lib/fs/nfs/fedfs-11.schema ファイル,  
 96  
 /usr/sbin/mount コマンド 参照 mount コマンド  
 /usr/sbin/nsdb-list コマンド  
 説明, 136  
 /usr/sbin/nsdb-nces コマンド  
 説明, 136  
 /usr/sbin/nsdb-resolve-fsn コマンド  
 説明, 137  
 /usr/sbin/nsdb-update-nci コマンド  
 説明, 137  
 /usr/sbin/nsdbparams コマンド  
 説明, 137  
 /usr/sbin/showmount コマンド, 135  
 /usr/sbin/unshareall コマンド, 134

/usr ディレクトリ  
 ディスクレスクライアントによるマウント, 14  
 UDP  
 NFS および, 19  
 umount コマンド  
 autofs と, 14  
 説明, 124  
 umountall コマンド, 125  
 UNIX 認証, 47, 48  
 unshare コマンド, 134  
 unshareall コマンド, 134  
 URL サービスタイプ  
 WebNFS および, 94

## V

-v オプション  
 umount コマンド, 124  
 -v オプション  
 automount コマンド, 152  
 vfstab ファイル  
 automount コマンドおよび, 54  
 NFS サーバーおよび, 79  
 クライアント側フェイルオーバーの有効化, 82  
 ディスクレスクライアントによるマウント, 14  
 ブート時のファイルシステムのマウント, 79

## W

WARNING: *mountpoint already mounted on* メッ  
 セージ, 153  
 WebNFS サービス  
 URL サービスタイプおよび, 94  
 概要, 21  
 計画, 91  
 セキュリティーネゴシエーションおよび, 22  
 説明, 44  
 タスクマップ, 91  
 ファイアウォールおよび, 94  
 ブラウズ, 93  
 有効化, 92