Oracle® Solaris 11.1 でのネットワークファイルシステムの管理

Part No: E36745–02
2012 年 10 月

ORACLE

121210@25097

目次

図目次

表目次

例目次

はじめに

『Oracle Solaris 11.1 でのネットワークファイルシステムの管理』は、Oracle Solaris システム管理情報についての重要な情報を説明する複数巻から成るマニュアルセットの 1 巻です。このドキュメントでは、Oracle Solaris オペレーティングシステムがすでにインストールされており、使用する予定のネットワークソフトウェアが設定済みであることを前提としています。

注 – この Oracle Solaris のリリースでは、SPARC および x86 系列のプロセッサアーキテクチャーを使用するシステムをサポートしています。サポートされるシステムについては、Oracle Solaris OS: Hardware Compatibility Lists を参照してください。このドキュメントでは、プラットフォームにより実装が異なる場合は、それを特記します。

Oracle サポートへのアクセス

Oracle のお客様は、My Oracle Support を通じて電子的なサポートを利用することができます。詳細は、`http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` を参照してください。聴覚に障害をお持ちの場合は、`http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` を参照してください。

表記上の規則

次の表では、このドキュメントで使用される表記上の規則について説明します。

表 P–1　表記上の規則

字体	説明	例
`AaBbCc123`	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	`.login` ファイルを編集します。 `ls -a` を使用してすべてのファイルを表示します。 `machine_name% you have mail.`

表 P–1 表記上の規則 *(続き)*

字体	説明	例
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	`machine_name%` **`su`** `Password:`
aabbcc123	プレースホルダ: 実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、 `rm` *filename* と入力します。
AaBbCc123	書名、新しい単語、および強調する単語を示します。	『ユーザーズガイド』の第 6 章を参照してください。 キャッシュは、ローカルに格納されるコピーです。 ファイルを保存しないでください。 注: いくつかの強調された項目は、オンラインでは太字で表示されます。

コマンド例のシェルプロンプト

Oracle Solaris OS に含まれるシェルで使用する、UNIX のシステムプロンプトとスーパーユーザープロンプトを次に示します。コマンド例では、シェルプロンプトはコマンドが標準ユーザーまたは特権ユーザーのどちらによって実行されるべきかを示しています。

表 P–2 シェルプロンプト

シェル	プロンプト
Bash シェル、Korn シェル、および Bourne シェル	`$`
Bash シェル、Korn シェル、および Bourne シェルのスーパーユーザー	`#`
C シェル	`machine_name%`
C シェルのスーパーユーザー	`machine_name#`

◆ ◆ ◆ 第 1 章

1 ネットワークファイルシステムの管理(概要)

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するNFS サービスの概要を説明します。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。

注 – 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、詳細について『Oracle Solaris 11.1 の管理: Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理』を参照してください。

NFS サービスの新機能

このセクションでは、Oracle Solaris OS の各リリースの新機能に関する情報を提供します。

このリリースでの重要な変更点

Oracle Solaris 11.1 リリースには、次の拡張機能が含まれています。

- `showmount` コマンドがリモートクライアントに表示する情報量を制御する新しいプロパティーが `/network/nfs/server:default` サービスに追加されました。詳細は、例 2–3 および 123 ページの「`showmount` コマンド」を参照してください。
- FedFS リフェラルのサポートが追加されました。これにより、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。詳細は、68 ページの「FedFS の管理」を参照してください。

以前のリリースでの重要な変更点

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

- `/etc/default/autofs` と `/etc/default/nfs` を編集することによって設定していた構成パラメータは、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 パラメータについては、それらを使用する手順およびそれらを使用するデーモンの説明を参照してください。
  - 103 ページの「`automount` コマンド」
  - 91 ページの「`automountd` デーモン」
  - 92 ページの「`lockd` デーモン」
  - 93 ページの「`mountd` デーモン」
  - 94 ページの「`nfsd` デーモン」
  - 95 ページの「`nfsmapid` デーモン」
- 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、NFSv4 クライアントはサーバーの名前空間で共有ファイルシステムのマウントポイントをたどることができます。NFSv4 マウントの場合、オートマウンタはサーバーの名前空間のルートをマウントし、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。従来のオートマウンタに対するミラーマウントの主な利点は、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の管理に関連するオーバーヘッドがなくなることです。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については、次を参照してください。
  - 38 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」
  - 156 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。リフェラルはサーバーベースのリダイレクションで、NFSv4 クライアントはそれに従ってファイルシステムを見つけることができます。NFS サーバーは `nfsref(1M)` コマンドによって作成されるリフェラルをサポートし、NFSv4 クライアントはリフェラルを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。この機能を使用すると、オートマウンタマップを編集する代わりにリフェラルを作成することにより、オートマウンタの多くの使用方法を置き換えることができます。NFS リフェラルは次の機能を提供します。
  - 前述した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。オートマウンタには依存しません。
  - クライアントでもサーバーでもセットアップは必要ありません。

  NFS リフェラルの詳細については、次を参照してください。
  - 67 ページの「NFS リフェラルの管理」
  - 158 ページの「NFS リフェラルのしくみ」
- DNS ドメイン単位でフェデレーテッドファイルシステム名前空間のルートをマウントする機能が追加されました。このマウントポイントを NFS リフェラルで使用し、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して、任意の大きな名前空間を構築できます。詳細については、次のトピックを参照してください。
  - 40 ページの「FedFS サーバーの DNS レコードの設定」
  - 161 ページの「`/nfs4` マウントポイント」
- `sharectl` ユーティリティーが含まれています。このユーティリティーでは、NFS などのファイル共有プロトコルの構成と管理を行うことができます。たとえば、このユーティリティーを使用すると、クライアントとサーバーの動作プロパティーの設定、特定のプロトコルのプロパティー値の表示、プロトコルのステータスの取得が可能です。詳細は、`sharectl`(1M) のマニュアルページおよび 114 ページの「`sharectl` コマンド」を参照してください。
- 初期の Solaris 10 リリース以降、NFS version 4 ドメインを定義する方法が変わりました。詳細は、101 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。

NFS の用語

このセクションでは、NFS サービスを使用するために必要な基本用語について説明します。NFS サービスの詳細は、第 3 章「ネットワークファイルシステムへのアクセス (リファレンス)」で説明します。

NFS サーバーとクライアント

クライアントとサーバーという用語は、コンピュータがファイルシステムを共有するときの役割を示すものです。ネットワークを介してファイルシステムを提供するコンピュータは、サーバーの役割を果たします。そのファイルシステムにアクセスしているコンピュータをクライアントと呼びます。NFS を使用することによって、どのコンピュータからも他のコンピュータのファイルシステムにアクセスでき、ネットワーク上では 1 台のコンピュータがクライアントかサーバー、またはその両方の役割として動作することができます。

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のではありません。マウントプロセスでは一連のリモートプロシージャー呼び出しによって、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。ユーザーはファイルシステムがローカルにあるのと同じようにコマンドを入力します。ファイルシステムをマウントするタスクについては、35 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。`share` コマンドと autofs に関連するタスクについては、32 ページの「ファイルシステムの自動共有」および51 ページの「autofs 管理タスクの概要」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は、ディレクトリツリー、つまり、あるファイル階層の全体またはその一部であり、これにはファイルが 1 つのみの場合も含まれます。すでに共有しているファイル階層と重複する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。

多くの UNIX システム環境で共有されるファイル階層構造は、1 つのファイルシステム、またはその一部です。しかし NFS サポートは複数のオペレーティングシステムにまたがって動作しますが、ファイルシステムという概念は UNIX 以外の環境では意味がないかもしれません。したがって、「ファイルシステム」という語は、NFS での共有およびマウントが可能なファイルまたはファイル階層構造を指します。

NFS サービスについて

NFS サービスとは、アーキテクチャーが異なり、別のオペレーティングシステムで動作しているコンピュータが、ネットワークを通じてファイルシステムを共有できるようにするサービスのことです。NFS サポートは、MS-DOS から VMS オペレーティングシステムまで多くのプラットフォームに実装されています。

NFS 環境は、異なるオペレーティングシステムで実現できます。NFS はアーキテクチャーの仕様を定義するのではなく、ファイルシステムの抽象モデルを定義しているためです。それぞれのオペレーティングシステムでは、ファイルシステムセマンティクスに NFS 抽象モデルを適用します。このモデルにより、書き込みや読み取りのようなファイルシステムオペレーションが、ローカルファイルにアクセスするように機能することになります。

NFS サービスには次の利点があります。

- 複数のコンピュータで同一のファイルを使用するため、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
- 各ユーザーアプリケーションがローカルのディスクスペースを占めるのではなく、複数のコンピュータでアプリケーションを共有するため、ストレージを有効利用できる
- 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
- 異機種システム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 サービスを使用すると、ファイルシステムの実際の場所をユーザーとは無関係に決めることができます。ユーザーは場所を気にすることなく、すべての適切なファイルにアクセスできるということです。NFS サービスでは、よく使用されるファイルのコピーをすべてのシステムに置くのではなく、NFS サーバーのファイルシステムから元のファイルを共有できるようにします。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」と呼ばれる方法によってマウントできます。クライアント側のサービスである autofs は、自動マウントを実現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。automountd という常駐型の自動マウントデーモンが、必要に応じてリモートディレクトリのマウントとアンマウントを行います。

automountd を実行しているクライアントコンピュータがリモートのファイルまたはディレクトリにアクセスしようとすると、リモートのファイルシステムがデーモンによってマウントされます。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時間アクセスされないと、自動的にアンマウントされます。

ブート時にマウントする必要はなく、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のパスワードを知る必要はありません。ユーザーが mount と umount コマンドを使用する必要もありません。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 によって一部のファイル階層をマウントするということは、mount によって他の階層をマウントしないということではありません。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 をマウントしなければなりません。

autofs サービスについては、51 ページの「autofs 管理タスクの概要」と165 ページの「autofs のしくみ」で詳しく説明します。

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

version 2 は、一般に広く使用された初めての NFS プロトコルです。version 2 は、引き続き広範囲のプラットフォームで使用できます。Oracle Solaris のすべてのリリースが NFS プロトコルの version 2 をサポートしています。

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 バイト以上のファイルを扱えます。以前の制限はなくなりました。26 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 では、サーバーで非同期の書き込みが可能になります。サーバーがクライアントの書き込み要求をメモリーに保存するので、効率が向上しました。クライアントは、サーバーが変更内容をディスクに反映させるのを待つ必要がないため、応答時間が短縮されます。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。

Solaris NFS version 3 の多くの操作では、ローカルキャッシュに保存されているファイル属性が返されます。キャッシュの更新頻度が増えたため、ローカルキャッシュのデータを更新する操作を独立して行う必要性が少なくなります。したがってサーバーに対する RPC コールの回数が減少し、パフォーマンスが向上します。

ファイルアクセス権の確認処理も改善されました。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と、「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、ファイルを開く前にアクセス権がチェックされるため、「オープンエラー」というメッセージが出力されます。

NFS version 3 プロトコルでは、8K バイトの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 の 8K バイトの制限を受けることなく、サポートされている転送サイズを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。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。

NFS version 4 プロトコル

NFS version 4 は、以前のバージョンでは使用できない機能を備えています。

NFS version 4 プロトコルでは、ユーザー ID とグループ ID が文字列として表されます。`nfsmapid` は、次の目的でクライアントとサーバーが使用します。

- version 4 のこれらの ID 文字列をローカルの数値 ID にマップするには
- ローカルの数値 ID を version 4 の ID 文字列に割り当てる

詳細は、95 ページの「`nfsmapid` デーモン」を参照してください。

NFS version4 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またはグループ ID にマッピングします。逆も同じです。詳細は、142 ページの「NFS version 4 での ACL と `nfsmapid`」を参照してください。

NFS version 4 では、ファイルシステムの共有を解除するとき、そのファイルシステムにあるオープンファイルまたはファイルロックの状態がすべて削除されます。NFS version 3 では、ファイルシステムが共有解除される前に、サーバーはクライアントが取得したロックを保持しました。詳細は、133 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 のサーバーは擬似ファイルシステムを使用して、クライアントがサーバーにエクスポートされたオブジェクトにアクセスできるようにします。NFS version 4 以前のバージョンには、擬似ファイルシステムがありません。詳細は、134 ページの「NFS version 4 におけるファイルシステムの名前空間」を参照してください。

NFS version 2 と version 3 では、サーバーは持続的ファイルハンドルを返しました。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、136 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。詳細は、140 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 では LIPKEY/SPKM のセキュリティーの種類をサポートしていません。

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 での機能の一覧は、133 ページの「NFS version 4 における機能」を参照してください。

NFS version 4 の使用に関する手順については、42 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。詳細は、クライアントパラメータについては 93 ページの「`mountd` デーモン」、サーバーパラメータについては 94 ページの「`nfsd` デーモン」、または `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

Solaris 2.5 で、アクセス制御リスト (ACL) サポートが追加されました。 アクセス制御リスト (ACL) では、ファイルアクセス権を通常の UNIX ファイルアクセス権よりも厳密に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 サーバーへの 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 の実装では、旧 POSIX ドラフトスタイルの ACL をサポートしています。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。UFS ACL の詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』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、新しい NFSv4スタイルの ACL をサポートします。 NFSv4 ACL は、ZFS によりネイティブでサポートされます。NFSv4 ACL の全機能を利用するには、NFSv4 サーバーの基盤となるファイルシステムとして ZFS を使用する必要があります。NFSv4 ACL は、豊富な継承プロパティーセット、および標準の読み取り、書き込み、実行を超えたアクセス権ビットセットを備えています。新しい ACL の概要については、『Oracle Solaris 11.1 の管理: ZFS ファイルシステム』の第 7 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、142 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のトランスポートプロトコルは、Solaris 2.5 で TCP (Transport Control Protocol) に変更されました。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 を利用した NFS は、version 2、version 3、および version 4 で動作します。Solaris 2.5 より前のリリースでは、NFS のデフォルトプロトコルはユーザーデータグラムプロトコル (UDP) でした。

注 – InfiniBand ハードウェアがシステムで使用可能な場合は、デフォルトのトランスポートプロトコルが TCP から Remote Direct Memory Access (RDMA) に変わります。詳細は、130 ページの「RDMA 経由の NFS」を参照してください。ただし、`proto=tcp` マウントオプションを使用する場合、NFS マウントには強制的に TCP のみが使用されます。

UDP 経由の NFS

Solaris 10 以降のリリースでは、NFS クライアントで余分な UDP ポートが使用されなくなりました。これまで、UDP 経由の NFS 転送では、未処理の要求ごとに別々の UDP ポートが使用されていました。これからはデフォルトで、予約済みの UDP ポートが 1 つだけ使用されるようになりました。ただし、このサポートは構成可能です。複数のポートを同時に使用したほうがスケーラビリティーが高まり、結果的にシステムのパフォーマンスが向上するような場合には、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。なお、この機能は、TCP 経由の NFS に最初から備わっていた同種の構成可能なサポートを UDP に移植したものです。詳細は、『Oracle Solaris 11.1 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。

注 – NFS version 4 は、UDP を使用しません。`proto=udp` オプションを使用してファイルシステムをマウントする場合は、NFS version 3 が version 4 の代わりに使用されます。

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、デフォルトのトランスポートプロトコルが TCP から Remote Direct Memory Access (RDMA) に変わります。RDMA プロトコルとは、高速ネットワーク経由でデータのメモリー間転送を行うためのテクノロジです。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 テクノロジと Oracle Solaris オペレーティングシステムを組み合わせます。詳細は、130 ページの「RDMA 経由の NFS」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 経由で共有されるすべてのファイルに UNIX のレコードロック機能を提供します。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 リクエストを互いに同期し合えるようになり、データ整合性が確保されます。

注 – ネットワークロックマネージャーは、NFS version 2 と version 3 の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す。

NFS 大規模ファイルのサポート

Solaris 2.6 の NFS version 3 プロトコルから、2G バイトを超えるサイズのファイル (大規模ファイル) も正しく処理できるようになりました。NFS version 2 プロトコルでは、2G バイトを超えるファイルは処理できませんでした。

NFS クライアントフェイルオーバー

Solaris 2.6 では、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機能が追加されました。フェイルオーバーによって、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あらかじめ複製されている読み取り専用リソースを高度に利用できます。フェイルオーバー機能は、ファイルシステムがマウントされた後ならばいつでも実行可能です。手動マウントでは、今までのリリースのオートマウンタ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。オートマウンタ

は、フェイルオーバーの際にファイルシステムが再マウントされるまで待つ必要がなくなったこと以外は変更されていません。詳細は、38 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 146 ページの「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。

NFS サービスのための Kerberos のサポート

NFS サービスは、Kerberos V5 クライアントをサポートしています。`mount` および `share` コマンドが Kerberos V5 認証を使用する NFS version 3 のマウントをサポートするように変更されました。`share` コマンドもクライアントごとに異なる複数の認証機能を使用できるように変更されました。セキュリティー方式に関連する変更の詳細は、27 ページの「RPCSEC_GSS セキュリティー方式」を参照してください。Kerberos V5 認証については、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「Kerberos NFS サーバーの構成」を参照してください。

WebNFS のサポート

Solaris 2.6 には、インターネット上のファイルシステムにファイアウォール経由でアクセスできるようにする機能も追加されました。この機能は、NFS プロトコルの拡張機能によって実現しました。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、信頼性が高いことです。このサービスは、NFS version 3 と version 2 プロトコルの拡張として構築されています。さらに、WebNFS ではそうしたファイルを共有しても匿名 `ftp` サイトを管理するオーバーヘッドが生じません。WebNFS サービスに関連する変更の詳細は、28 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」を参照してください。タスクの詳細は、49 ページの「WebNFS の管理タスク」を参照してください。

注 – 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 から、RPCSEC_GSS と呼ばれるセキュリティー方式がサポートされています。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、27 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 の詳細については、『Oracle Solaris 11 セキュリティーサービス開発ガイド』を参照してください。

Solaris 7 の NFS に対する拡張機能

Solaris 7 で、`mount` コマンドと `automountd` コマンドが拡張され、マウント要求で MOUNT プロトコルの代わりに公開ファイルハンドルも使用できるようになりました。MOUNT プロトコルは、WebNFS サービスが使用するアクセス方法と同じです。公開ファイルハンドルを使用すると、ファイアウォールを越えたマウントが可能です。さらに、サーバーとクライアント間のトランザクションが少なくて済むため、マウントにかかる時間が短縮されます。

この拡張機能で、標準のパス名の代わりに NFS URL を使用することもできるようになりました。また、`mount` コマンドとオートマウンタのマップに `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なります。WebNFS サービスの変更の詳細は、27 ページの「WebNFS のサポート」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

Solaris 8 で、WebNFS クライアントが 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新しいプロトコルが追加されました。このプロトコルの追加により、WebNFS サービスの使用時に、セキュアなトランザクションを使用できます。詳細は、151 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

Solaris 8 で、NFS サーバーはロギングによって、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るようになりました。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。また、これらのオプションを使用して、ログに記録する処理を選択することもできます。この機能は、NFS クライアントや WebNFS クライアントで匿名 ftp を利用するサイトで特に便利です。詳細は、34 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。

注 – NFS version 4 は、サーバーロギングをサポートしません。

---

autofs の機能

autofs は、ローカルの名前空間に指定されているファイルシステムで動作します。この情報は、NIS またはローカルファイルに保持できます。

完全にマルチスレッド化されたバージョンの automountd が含まれています。この拡張によって autofs はさらに信頼性が高まりました。また、複数のマウントを並行してサービスできるようになったため、あるサーバーが使用できないときにサービスが停止することも避けられます。

automountd は、改善されたオンデマンドマウント機能を提供します。Solaris 2.6 より前のリリースでは、階層に含まれるすべてのファイルシステムがマウントされていました。現在は、いちばん上のファイルシステムしかマウントされません。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 サービスで、間接マップを表示できるようになりました。これによりユーザーは、どのディレクトリがマウントできるかを確認するために各ファイルシステムを実際にマウントする必要がなくなります。autofs マップに -nobrowse オプションが追加されたので、/net や /home などの大きなファイルが自動的に表示されることはありません。また、-automount で n オプションを使用することによって、autofs のブラウズ機能をクライアントごとにオフにすることもできます。詳細は、64 ページの「autofs のブラウズ機能の無効化」を参照してください。

◆ ◆ ◆ 第 2 章

ネットワークファイルシステムの管理 (タスク)

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 の管理タスクの実行方法について説明します。また、Secure NFS システムおよび WebNFS の機能の使用方法についても説明します。章の最後ではトラブルシューティングの手順を説明し、NFS のいくつかのエラーメッセージとその意味を示します。

NFS 管理者の責任は、サイトの要求やネットワーク上に存在するコンピュータの役割によって変わります。管理者がローカルネットワークのコンピュータすべてに責任を持つこともありえます。そのような場合は、次の構成事項について判断する必要があります。

- サーバー専用にするコンピュータの決定
- サーバーとクライアントの両方として動作するコンピュータの決定
- クライアントとしてのみ動作するコンピュータの決定

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、コンピュータが自動的にマウントした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

コンピュータは、サーバーとクライアントのどちらにもなれることに注意してください。つまり、ローカルファイルシステムをリモートコンピュータと共有したり、リモートファイルシステムをマウントしたりできます。

注 – 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、詳細について『Oracle Solaris 11.1 の管理: Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理』を参照してください。

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。`/etc/dfs/dfstab` は使用されなくなりました。

表 2–1 ファイルシステムの共有 (タスクマップ)

タスク	説明	参照先
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する手順	32 ページの「ファイルシステム自動共有を設定する方法」
WebNFS を有効にします	ユーザーが WebNFS でファイルにアクセスできるようにサーバーを構成する手順	33 ページの「WebNFS アクセスを有効にする方法」
NFS サーバーログを有効にします	NFS ロ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する手順	34 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 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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。

   `share` コマンドを使用して、共有される各パスを定義します。この情報はシステムのリブート時に維持されます。

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

   specific-options の完全な一覧については、`share_nfs(1M)` のマニュアルページを参照してください。

3 情報が正しいことを確認します。

share コマンドを実行し、適切なオプションが表示されていることを確認します。

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

参照 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。詳細は、51 ページの「autofs 管理タスクの概要」を参照してください。

▼ WebNFS アクセスを有効にする方法

次の事項に注意してください。

- デフォルトでは、NFS マウントに利用可能なすべてのファイルシステムが、WebNFS アクセス用として自動的に利用可能となります。この手順を使用する必要があるのは、次のいずれかの場合だけです。
  - 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リセットして 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、特定の HTML ファイルが強制的に読み込ま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。sharectl(1M) のマニュアルページおよび114 ページの「sharectl コマンド」を参照してください。

WebNFS サービスを起動する際の注意事項については、49 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 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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。次の例の public タグおよび index タグは省略できます。

```
# share -F nfs -o ro,public,index=index.html /export/ftp
```

オプションの完全な一覧については、share_nfs(1M) のマニュアルページを参照してください。

3 情報が正しいことを確認します。

share コマンドを実行し、適切なオプションが表示されていることを確認します。

```
# 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
```

▼ 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 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 (省略可能) ファイルシステム構成の設定を変更します。

/etc/nfs/nfslog.conf では、2 つの方法のいずれかで設定を変更できます。すべてのファイルシステムについてデフォルトの設定を編集するには、global タグに関連するデータを変更します。または、このファイルシステムについて新しいタグを追加します。これらの変更が必要でない場合には、このファイルを変更する必要はありません。/etc/nfs/nfslog.conf の書式については、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* オプションとともに使用されるタグは、/etc/nfs/nfslog.conf に入力される必要があります。この例では、global タグ内のデフォルト設定を使用します。

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

4 情報が正しいことを確認します。

share コマンドを実行し、適切なオプションが表示されていることを確認します。

```
# 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 (省略可能) 動作していない場合は、**nfslogd** を起動します。

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

ファイルシステムのマウント

ファイルシステムをマウントするには、いくつかの方法があります。システムをブートするときに自動的にマウントされるようにするか、コマンド行から必要に応じてマウントするか、オートマウンタを使用します。オートマウンタには、ブート時のマウントやコマンド行からのマウントに比較していくつもの利点がありますが、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。ファイルシステムのマウントに関するすべてのタスクのリストについては、次の表を参照してください。

表 2–2 ファイルシステムのマウントのタスクマップ

タスク	説明	参照先
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する手順。	36 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	システムの動作時にファイルシステムをマウントする手順。この手順はテストに有効です。	36 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」
オートマウンタによりマウントします	コマンド行を使用せずに、要求に応じてファイルシステムにアクセスする手順。	37 ページの「オートマウンタによるマウント」
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します	ミラーマウントを使用して 1 つ以上のファイルシステムをマウントする手順。	例 2–2
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する手順。	38 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	サーバーの不良時、動作中のファイルシステムへの自動切り換えを有効にする手順。	38 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	任意の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する手順。	39 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 プロトコルでファイアウォールを越えてファイルシステムへのアクセスを許可する手順。	39 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 を使ってファイルシステムへのアクセスを許可する手順。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へのアクセスが可能になります。	40 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」

表 2–2 ファイルシステムのマウントのタスクマップ (続き)

タスク	説明	参照先
FedFS ファイルシステムをマウントします	/nfs4 マウントポイントを通して FedFS ファイルシステムにアクセスできるように DNS レコードを確立する処理。	40 ページの「FedFS サーバーの DNS レコードの設定」

▼ ブート時にファイルシステムにマウントする方法

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には、次の手順に従います。リモートファイルシステムにアクセスするクライアントごとに、この手順を行う必要があります。

1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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** に追加します。

/etc/vfstab ファイルのエントリ構文は、次のとおりです。

```
special  fsckdev  mountp  fstype  fsckpass  mount-at-boot  mntopts
```

詳細は、vfstab(4) のマニュアルページを参照してください。

注意 – NFS クライアントの vfstab エントリも持つ NFS サーバーでは、リブート時のハングアップを避けるために、常に bg オプションを指定する必要があります。詳細は、106 ページの「NFS ファイルシステム用の mount オプション」を参照してください。

例 2–1 クライアントの vfstab ファイル内のエントリ

wasp サーバーの /var/mail ディレクトリをクライアントマシンにマウントさせたいとします。それには、そのファイルシステムをクライアント上の /var/mail としてマウントし、読み取りと書き込みの両方ができるようにします。この場合は、次の項目をクライアントの vfstab ファイルに追加します。

```
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

新規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あります。このようにしてマウントすると、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、一時的にアクセスすることができます。

1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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。

次のコマンドを入力します。

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

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。コマンド行からこのようにマウントすることにより、ファイルシステムを一時的に表示することができます。umount を実行するかローカルホストをリブートすると、このマウントは解除されます。

注意 – mount コマンドのどのバージョンでも、無効なオプションに関する警告は出しません。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。

例 2–2 ファイルシステムをマウントしたあとでミラーマウントを使用する

このリリースにはミラーマウント機能が含まれています。この新しいマウント技術は、NFSv4 サーバーからの 2 つ目のファイルシステムにアクセスする任意の NFSv4 クライアントから使用できます。mount コマンドまたはオートマウンタを使用してサーバーから最初のファイルシステムをマウントしたあとは、そのマウントポイントに追加された任意のファイルシステムにアクセスできます。必要な操作はファイルシステムへのアクセスを試みることだけです。ミラーマウントは自動的に実行されます。詳細は、156 ページの「ミラーマウントのしくみ」を参照してください。

オートマウンタによるマウント

51 ページの「autofs 管理タスクの概要」では、オートマウンタによるマウントの確立とサポートについて詳細に説明します。通常のシステムに変更を加えることなく、リモートファイルシステムが /net マウントポイントでアクセスできるようになります。前述の例の /export/share/local ファイルシステムをマウントする場合は、次のように入力します。

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

オートマウンタでは、すべてのユーザーがファイルシステムをマウントできるので、root としてアクセスする必要はありません。またファイルシステムのマウントを自動的に解除できるので、作業の終了後、ファイルシステムのマウントを解除する必要はありません。

追加のファイルシステムをクライアントにマウントする方法については、例 2–2を参照してください。

▼ 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

このリリースにはミラーマウント機能が含まれるため、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、そのサーバーから NFS を使用して共有され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。詳細は、156 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 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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。この場合は、`/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。

   このコマンドなど、ファイルシステムにアクセスする任意のコマンドを実行すると、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 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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。

   コマンド行からも、オートマウンタを使用しても、また`/etc/vfstab` ファイルに次のようなエントリを追加することによってもマウントできます。

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

   この構文はオートマウンタでも指定できました。しかし、フェイルオーバー機能が使用できるのは単一のサーバーが選択されているときだけで、ファイルシステムがマウントされている間は使用できませんでした。

注 – 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、コマンド行や vfstab のエントリに混在させないでください。NFS version 2、version 3、または version 4 のプロトコルをサポートしているサーバーを混在して使用できるのは、autofs を使用する場合だけです。autofs では、version 2 、version 3、または version 4 のサーバーの最適な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 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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 ro=-rose:eng /export/share/man
```

ro=-rose:eng	rose という名前のホストを除き、eng ネット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
/export/share/man	共有されるファイルシステム。

▼ 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

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するには、次の手順を実行します。

1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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 bee:/export/share/local /mnt
```

この例では、/export/share/local というファイルシステムは、公開ファイルハンドルを使ってローカルクライアントにマウントしています。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。

注 – 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、クライアントと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。

▼ 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 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 (省略可能) ファイルシステムを手動でマウントします。

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

この例では、サーバー bee の /export/share/local というファイルシステムが、NFS ポート番号 3000 を使ってマウントされます。ポート番号を指定する必要はありません。その場合、デフォルトの NFS ポート番号である 2049 が使用されます。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 – ファイルシステムのマウント時に使用される NFS プロトコルのバージョンは、クライアントとサーバーの両方によってサポートされる最上位のバージョンです。vers=# オプションを使用すると、特定の NFS プロトコルバージョンを選択できます。

FedFS サーバーの DNS レコードの設定

適切な DNS レコードを作成したあと、マウントポイントにアクセスすると、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します。サーバーの DNS レコードは次のようになります。

```
% 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.
```

▼ 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する方法

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。一部の環境では、必ずしもすべてのクライアントにこの情報が表示されるべきではありません。手順については、例 2–3 を参照してください。

- マウント可能なファイルシステムに関する情報を表示します。

  -e オプションを使用すると、共有ファイルシステムのリストを出力できます。ほかのオプションについては、123 ページの「showmount コマンド」または showmount(1M) のマニュアルページを参照してください。

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

例 2–3 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

一部の環境では、共有ファイルシステムやそれらをマウントしているシステムに関する情報が表示されるべきではありません。共有ファイルシステムに関するすべての情報を表示するのではなく、showmount_info プロパティーを使用して、クライアントが次になるように設定できます。

- アクセスが許可されているファイルシステムに関する情報のみを表示できます
- 共有されているすべてのファイルシステムに関する情報を必ずしも表示できません
- ファイルシステムをマウントしているほかのシステムに関する情報を表示できません

このプロパティーを設定するには、サーバーで次のコマンドを実行します。

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

これで、クライアント rose では、次の情報が表示されるようになります。

```
% /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っています。

NFS サービスの設定

このセクションでは、次のことを行うために必要なタスクを説明します。

- NFS サーバー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する
- 異なるバージョンの NFS を選択する

注 – Solaris 10 以降のリリースでは、NFS のデフォルトは version 4 です。

表 2–3 NFS サービスのタスクマップ

タスク	説明	参照先
NFS サーバーを起動します	NFS サービスが自動的に起動されていない場合に、NFS サービスを起動する手順。	43 ページの「NFS サービスを起動する方法」
NFS サーバーを停止します	NFS サービスを停止する手順。通常は、サービスを停止する必要はありません。	43 ページの「NFS サービスを停止する方法」
オートマウンタを起動します	オートマウンタを起動する手順。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。	43 ページの「オートマウンタを起動する方法」
オートマウンタを停止します	オートマウンタを停止する手順。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。	44 ページの「オートマウンタを停止する方法」
サーバー上で異なるバージョンの NFS を選択します	サーバー上で異なるバージョンの NFS を選択する手順。NFS version 4 を使用しない場合、この手順を使用します。	44 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」
クライアント上で異なるバージョンの NFS を選択します	SMF パラメータを変更して、クライアント上で異なるバージョンの NFS を選択する手順。NFS version 4 を使用しない場合、この手順を使用します。	45 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」
	コマンド行を使用して、クライアント上で異なるバージョンの NFS を選択する代替手順。NFS version 4 を使用しない場合、この代替手順を使用します。	46 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」

▼ NFS サービスを起動する方法

1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 サーバー上で **NFS** サービスを有効にします。

次のコマンドを入力します。

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

このコマンドを実行すると、NFS サービスが有効になります。

注 – システムのブート時に NFS サーバーが自動的に起動されます。さらに、システムのブート以降は、NFS ファイルシステムを共有すると NFS サービスデーモンが自動的に有効になります。32 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。

▼ NFS サービスを停止する方法

1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 サーバー上で **NFS** サービスを無効にします。

次のコマンドを入力します。

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

▼ オートマウンタを起動する方法

1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 **autofs** デーモンを有効にします。

次のコマンドを入力します。

```
# svcadm enable system/filesystem/autofs
```

▼ オートマウンタを停止する方法

1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 **autofs** デーモンを無効にします。

次のコマンドを入力します。

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

▼ サーバー上で異なるバージョンの NFS を選択する方法

NFS version 4 を使用しない場合、この手順を使用します。

1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

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

たとえば、サーバーが NFS version 3 のみを提供するようにするには、次のように、`server_versmax` と `server_versmin` の値をどちらも 3 に設定します。

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

注 – デフォルトで設定される NFS のバージョンは NFS version 4 です。

---

3 **(省略可能)** サーバー委譲を無効にします。

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

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

注 – 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、140 ページの「NFS version 4 における委託」を参照してください。

---

4 **(省略可能)** 共通ドメインを設定します。

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` プロパティーを変更します。

```
# sharectl set -p nfsmapid_domain=my.company.com nfs
```

my.company.com　　共通ドメイン名を指定します

詳細は、95 ページの「nfsmapid 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ことを確認します。**

次のコマンドを入力します。

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

このコマンドは、NFS サーバーサービスがオンラインか、または無効かをレポートします。

6 **(省略可能) 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であることを前の段階で検出した場合、次のコマンドを入力して、そのサービスを有効にします。

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

注 – NFS サービスを構成する必要がある場合は、32 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 132 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

次の手順は、クライアント上で使用される NFS のバージョンを制御する方法を示しています。

1 **管理者になります。**

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

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

たとえば、ファイルシステムが NFS version 3 プロトコルを使用してマウントされるようにするには、次のように、`client_versmax` と `client_versmin` の値をどちらも 3 に設定します。

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

注 – デフォルトで設定される NFS のバージョンは NFS version 4 です。

---

3 クライアント上で **NFS** をマウントします。

次のコマンドを入力します。

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

server-name サーバーの名前を指定します。

/share-point マウントするリモートディレクトリのパスを指定します。

/local-dir ローカルマウントポイントのパスを指定します。

参照 132 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 のバージョンを制御する方法を示しています。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 のバージョンを変更する場合は、45 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 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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。

次のコマンドを入力します。

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

value バージョン番号を指定します。

server-name サーバーの名前を指定します。

/share-point マウントするリモートディレクトリのパスを指定します。

/local-dir ローカルマウントポイントのパスを指定します。

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

参照 132 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、関与するすべてのコンピュータにドメイン名が必要です。通常、ドメインとは、複数のコンピュータから構成される管理上のエンティティーのことであり、大規模なネットワークの一部です。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。『Oracle Solaris Administration: Naming and Directory Services 』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。Kerberos サービスについては、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の第 19 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も構成できます。この認証サービスについては、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の第 18 章「ネットワークサービスの認証 (タスク)」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コンピュータに知らせます。

2. 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` コマンドを使用します。`chkey` コマンドを使用して、各ユーザーに独自の Secure RPC パスワードを設定してもらいます。

   注 – これらのコマンドについての詳細は、`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。

   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。

3. ネームサービスが応答していることを確認します。

例:

- 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。

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

次のコマンドを入力します。

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

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/keyserv
```

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。ログインパスワードとネットワーク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 を実行しなければなりません。また、keylogin -r コマンドを root として実行し、復号化した秘密鍵を /etc/.rootkey に保存する必要があります。

注 – keylogin -r は、root の秘密鍵が変更されたか、/etc/.rootkey が損失した場合に、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タマップを更新します。

auto_master データを編集し、Diffie-Hellman 認証の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
```

コンピュータの再インストール、移動、またはアップグレードを行うときに、root 用に新しい鍵の確立または鍵の変更を行わない場合は、忘れずに /etc/.rootkey を保存してください。/etc/.rootkey を削除する場合は、常に次を入力できます。

```
# keylogin -r
```

WebNFS の管理タスク

このセクションでは、WebNFS システムを管理する方法について説明します。関連するタスクを次に示します。

表 2–4　WebNFS 管理のタスクマップ

タスク	説明	参照先
WebNFS に関する計画を作成します	WebNFS サービスを有効にする前に考慮する項目。	49 ページの「WebNFS アクセスの計画」
WebNFS を有効にします	WebNFS プロトコルを使用して NFS ファイルシステムのマウントを有効にする手順。	33 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 プロトコルを使用して、ファイアウォール経由でファイルへのアクセスを許可する手順。	51 ページの「ファイアウォール経由で WebNFS アクセスを有効にする方法」
NFS URL を使ってブラウズします	Web ブラウザ内での NFS URL の使用についての説明。	50 ページの「NFS URL を使ってブラウズする方法」
autofs で公開ファイルハンドルを使用します	オートマウンタでファイルシステムをマウントする場合に、公開ファイルハンドルの使用を強制するための手順。	64 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	オートマウンタマップに NFS URL を追加するための手順。	64 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 プロトコルでファイアウォールを越えてファイルシステムへのアクセスを許可する手順。	39 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 を使ってファイルシステムへのアクセスを許可する手順。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へのアクセスが可能になります。	40 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://server/path` のような NFS URL を実行し、読み込め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。

1. サーバーには公開ファイルハンドルが 1 つずつあり、このハンドルは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ます。NFS URL に示されたパスは、この公開ファイルハンドルが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すると、URL は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なります。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。
2. 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。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ているかどうかには関係ありません。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ー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。
3. すでに公開されているファイルシステムは、`public` オプションを使用するのに適しています。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。
4. `share` コマンドで `index` オプションを使用すると、HTML ファイルを強制的に読み込むことができます。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らファイルを確認し、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ます。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、ディレクトリに対しては `755`、ファイルに対しては `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。これについては、152 ページの「Web ブラウザの使用と比較した場合の WebNFS の制約」で説明します。

NFS URL を使ってブラウズする方法

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 – ほとんどのブラウザでは、前のトランザクションで使用した URL サービスのタイプ (`nfs` や `http` など) を次のトランザクションでも使用できます。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。NFS URL を使用したあとに、HTTP URL に対する参照が読み込まれたとします。その場合、次に続くページは NFS プロトコルではなく HTTP プロトコルを使って読み込まれます。

ファイアウォール経由で WebNFS アクセスを有効にする方法

ローカルのサブネットに属していないクライアントに対して 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` に対してアクセスを許可するだけでは、NFS URL が使えるようにはなりません。

autofs 管理タスクの概要

このセクションでは、ユーザー自身の環境で遭遇する可能性のあるもっとも一般的なタスクについて説明します。各シナリオについて、ユーザーのクライアントで必要とする条件にもっとも適合するように autofs を構成するために推奨される手順も示します。

注 – SMF リポジトリ内のパラメータを使用して autofs 環境を構成することもできます。タスクの詳細は、53 ページの「SMF パラメータを使用して autofs 環境を構成する」を参照してください。

---

autofs 管理のタスクマップ

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 2–5　autofs 管理のタスクマップ

タスク	説明	参照先
autofs を起動します	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します	43 ページの「オートマウンタを起動する方法」

表 2–5 autofs 管理のタスクマップ (続き)

タスク	説明	参照先
autofs を停止します	他のネットワークサービスを使用不可にすることなく自動マウントサービスを停止します	44 ページの「オートマウンタを停止する方法」
autofs SMF パラメータによって 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	53 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	37 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	他のマップを一覧表示するために使用されるマスターマップの修正を行う手順	55 ページの「マスターマップを修正する方法」
	ほとんどのマップに対して使用される間接マップの修正を行う手順	55 ページの「間接マップを修正する方法」
	クライアント上のマウントポイントとサーバー間の直接の関係が必要な場合に使用される直接マップの修正を行う手順	56 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	CD-ROM アプリケーション用のエントリで autofs マップを設定する手順	57 ページの「autofs で CD-ROM アプリケーションにアクセスする方法」
	PC-DOS フロッピーディスク用のエントリで autofs マップの設定を行う手順	57 ページの「autofs で PC-DOS データフロッピーディスクにアクセスする方法」
/home を使用します	共通の /home マップの設定方法の例	58 ページの「/home の共通表示の設定」
	複数のファイルシステムを参照する /home マップを設定する手順	58 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	プロジェクト関連の autofs マップを設定する手順	59 ページの「/ws 下のプロジェクト関連ファイルを統合する方法」
	異なるクライアントアーキテクチャーをサポートする autofs マップを設定する手順	61 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」
	異なるオペレーティングシステムをサポートする autofs マップを設定する手順	62 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	63 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」

表 2–5 autofs 管理のタスクマップ (続き)

タスク	説明	参照先
autofs でセキュリティー制限を使用します	ファイルへのリモート root アクセスを制限する一方でファイルシステムへのアクセスを提供します	63 ページの「autofs セキュリティー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	64 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	オートマウンタが使用できるように、NFS URL を追加します	64 ページの「autofs で NFS URL を使用する方法」
autofs のブラウズ機能を無効にします	autofs マウントポイントが 1 つのクライアント上で自動的に生成されないように、ブラウズ機能を無効にする手順	65 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」
	autofs マウントポイントがすべてのクライアント上で自動的に生成されないように、ブラウズ機能を無効にする手順	65 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」
	特定の autofs マウントポイントがある 1 つのクライアント上で自動的に生成されないように、ブラウズ機能を無効にする手順	66 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。コマンド行で行うのと同じ指定を、sharectl コマンドで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。

次の手順は、sharectl コマンドを使用して autofs パラメータを管理する方法を示しています。

▼ 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 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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。

3 **autofs デーモンを再起動します。**

次のコマンドを入力します。

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

マップの管理タスク

次の表は、autofs マップの管理時に認識しておく必要のある事項について示しています。選択したマップのタイプおよびネームサービスにより、autofs マップへの変更を行うために使用する必要があるメカニズムが異なります。

次の表に、マップのタイプとその使用方法を示します。

表 2–6 autofs マップのタイプとその使用方法

マップのタイプ	用途
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向けにします
間接	autofs をリファレンス指向のファイルシステム向けにします

次の表では、使用しているネームサービスごとの、autofs 環境の変更方法を示しています。

表 2–7 マップの保守

ネームサービス	メソッド
ローカルファイル	テキストエディタ
NIS	make ファイル

次の表に、マップのタイプに対して行なった修正に応じた automount コマンドの実行について示します。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、既存のエントリを修正した場合は、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。

表 2–8 automount コマンドを実行する場合

マップのタイプ	**automount** を再実行するか否か	
	追加または削除	修正
auto_master	Y	Y
direct	Y	N
indirect	N	N

マップの修正

次の手順は、いくつかの種類のオートマウンタマップを更新する方法を示します。

▼ マスター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. 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。
3. 各クライアントで管理者になります。
   詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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、変更が反映されるようにします。
5. マップを変更したことを他のユーザーに通知します。
   他のユーザーがコンピュータ上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、通知が必要になります。automount コマンドは、実行時にマスターマップから情報を収集することに注意してください。

▼ 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、他のユーザーがコンピュータ上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、通知が必要になります。

注 – 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。

注 – 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容易であり、コンピュータのファイルシステムへの要求が少なくて済みます。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。

マウントポイントの重複回避

/src 上にマウントされたローカルな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のマウントにもその autofs サービスを使用する場合、問題が発生する可能性があります。マウントポイント /src を指定した場合、ユーザーがローカルパーティションにアクセスしようとするたびに、NFS サービスはそのローカルパーティションを非表示にします。

たとえば /export/src などの他の場所に、パーティションをマウントする必要があります。その後、次のようなエントリを /etc/vfstab に含める必要があります。

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

このエントリは、auto_src にも必要です。

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

terra はコンピュータ名です。

非 NFS ファイルシステムへのアクセス

autofs は NFS ファイル以外のファイルシステムもマウントすることができます。autofs は、フロッピーディスクや CD-ROM など、削除可能な媒体上のファイルをマウントします。

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。autofs を使用し非 NFS ファイルシステムにアクセスを行う場合は、次の手順を参照してください。

▼ autofs で CD-ROM アプリケーションにアクセスする方法

1. 管理者になります。

   詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2. **autofs** マップを更新します。

   次のような CD-ROM のファイルシステム用のエントリを追加します。

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

   マウントする CD-ROM 装置の名前が、コロンのあとに続けて表示されます。

▼ autofs で PC-DOS データフロッピーディスクにアクセスする方法

1. 管理者になります。

   詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2. **autofs** マップを更新します。

   次のようなフロッピーディスクのファイルシステム用のエントリを追加します。

   ```
   pcfs      -fstype=pcfs      :/dev/diskette
   ```

オートマウンタのカスタマイズ

オートマウンタマップの設定方法はいくつかあります。次のタスクに、オートマウンタマップをカスタマイズして簡単に使用できるディレクトリ構造を実現する方法について詳細に説明します。

/home の共通表示の設定

すべてのネットワークユーザーにとっての理想は、自分自身のホームディレクトリ、または他の人のホームディレクトリを /home の下に配置できるようにすることです。この表示方法は通常、クライアントでもサーバーでも、すべてのコンピュータを通じて共通です。

Oracle Solaris をインストールすると、常にマスターマップ /etc/auto_master も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 の下にインストールされます。

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

新しいローカルユーザーが作成されると、エントリが自動的に /etc/auto_home に追加されます。このようにして、dragon というサーバーで、/export/home/rusty と /home/rusty を通じて rusty のホームディレクトリにアクセスできます。

注 – ユーザーは、各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が許可されていません。この制限がないと、すべてのユーザーがすべてのコンピュータ上でスーパーユーザーの権限を持つことになります。

▼ 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 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 **/export/home** の下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。

システムに複数のパーティションがある場合は、/export/home1、/export/home2 のように、別のディレクトリにそれぞれインストールを行います。

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。

```
rusty        dragon:/export/home1/&
gwenda       dragon:/export/home1/&
charles      sundog:/export/home2/&
rich         dragon:/export/home3/&
```

マップ鍵の代替となる & (アンパサンド) の使い方に注意してください。このアンパサンドは、次の例の 2 つ目の rusty の使用を省略した形式です。

```
rusty         dragon:/export/home1/rusty
```

auto_home マップを配置すると、ユーザーは、/home/*user* というパスを使用して、ユーザー自身のホームディレクトリを含むあらゆるホームディレクトリを参照できます。*user* はログイン名で、マップ内での鍵になります。すべてのホームディレクトリを共通に表示するしくみは、他のユーザーのコンピュータにログインする場合に便利です。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、他のコンピュータ上でリモートのウィンドウシステムクライアントを実行するとウィンドウシステムクライアントと同じ /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。前の例を使用すれば、rusty がサーバー dragon にログインする場合、autofs は、/export/home1/rusty を /home/rusty にループバックマウントすることにより、ローカルディスクへの直接アクセスを提供します。

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。rusty がさらにディスク容量を必要とし、自身のホームディレクトリを他のサーバーに再配置する必要がある場合には、単純な変更で十分です。新しい場所を反映するように auto_home マップ内の rusty のエントリを変更することだけが必要になります。他のユーザーは、/home/rusty パスを継続して使用することができます。

▼ /ws 下のプロジェクト関連ファイルを統合する方法

大規模なソフトウェア開発プロジェクトの管理者を想定してください。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。

1 **/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。

2 **-nosuid** オプションを用心のために追加しておきます。

このオプションは、すべての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できないようにします。

3 **auto_ws** マップにエントリを追加します。

auto_ws マップは、各エントリがサブプロジェクトを記述するように構成されています。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
```

この最初の操作により、マップはシンプルなものになりますが、このマップでは不十分です。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 に対する参照は、vers1.0、vers2.0、および man ディレクトリ用に 3 つのマウントを必要とします。各行の最後のバックスラッシュは、エントリが次の行まで続いていることを autofs に伝えるものです。実際、エントリは 1 つの長い行となっていますが、行ブレー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。このような変更が auto_ws マップに反映される場合は、/ws 下のディレクトリ階層構造が変更されることもなく、ユーザーに対する通知の必要はありません。

サーバー alpha と bravo が同一の autofs マップを参照するため、それらのコンピュータにログインするすべてのユーザーは期待通りに /ws 名前空間を確認できます。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

表計算アプリケーションやワードプロセッサパッケージのようなローカルの実行可能ファイルやアプリケーションについて、共有名前空間を作成する必要があります。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。**

『Oracle Solaris Administration: Naming and Directory Services 』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、その名称として /usr/local を選択した場合、/usr/local/bin パスは明らかにこの名前空間の一部です。

3 ユーザーのコミュニティー識別を簡単にするため、**autofs** 間接マップを作成します。

autofs 間接マップを `/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
```

なお、`-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のすべてに対して書き込みができないことを示してます。

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 **1** つ含めます。

ディレクトリ構造は、次のようになります。

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

6 **(省略可能)** 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
```

- SPARC クライアント – 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 – 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 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、アーキテクチャータイプを結合します。

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/$CPU$OSREL
```

SunOS 5.6 を動作させているクライアントについて、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 – `/export/local/bin/sparc5.6` をエクスポートします。
- x86 クライアント – `/export/local/bin/i3865.6` に実行可能ファイルを配置します。

▼ 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。フェイルオーバーについての説明は、146 ページの「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。

1. 管理者になります。

   詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2. **autofs** マップ内のエントリを修正します。

   すべての複製サーバーのリストを、コンマ区切りのリストとして、次のように作成します。

   ```
   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を適用する方法

1. 管理者になります。

   詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2. ネームサービス **auto_master** ファイル内に次のようなエントリを作成します。

   ```
   /home     auto_home     -nosuid
   ```

   nosuid オプションは、setuid または setgid ビットを設定したファイルをユーザーが作成できないようにします。

   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。前述の例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。そのため、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 – 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 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 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 autofs マップに、次のような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 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 次のような autofs エントリを作成します。

```
/usr/local      -ro     nfs://bee/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 のブラウズ機能の無効化

インストールされるデフォルト版の /etc/auto_master には、/home と /net 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 と /net のエントリが変更されていない場合は、-nobrowse オプションがそれらのエントリに追加されます。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ズ機能をオフにすることが必要な場合もあります。

ブラウズ機能をオフにする方法はいくつかあります。automountd デーモンに対してコマンド行オプションを使用してブラウズ機能を無効にすると、そのクライアントに対する autofs ブラウズ機能は完全に無効になります。あるいは、autofs マップを使

用して、すべてのクライアントにおける各マップエントリのブラウズ機能を無効にします。また、ネットワーク規模の名前空間を使用していない場合は、ローカルな autofs を使用して、各クライアントにおける各マップエントリのブラウズ機能を無効にすることができます。

▼ 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 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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タ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 **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
```

▼ 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 プロパティーは、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。

2 **/etc/auto_master** 内の **+auto_master** エントリの位置を確認します。

名前空間内のエントリに優先するローカルファイルへの追加については、+auto_master エントリが /net の下に移動されている必要があります。

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

標準的な構成では、+auto_master エントリがファイルの先頭に配置されます。このように配置することにより、ローカルな変更が使用されなくなります。

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
```

NFS リフェラルの管理

NFS リフェラルは、複数の NFSv4 サーバーを均一な名前空間に接続する手段として、NFSv4 サーバーがほかの NFSv4 サーバー上にあるファイルシステムを指す方法です。

▼ 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。

   ```
   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します。

   ```
   client1% pfexec mount server1:/share/docs /mnt
   ```

4. 正常にマウントされたことを確認します。

   ```
   client1% cd /mnt/docs
   client1% df -k .
   /mnt/docs     (server2:/usr/local/docs):10372284465 blocks 10372284465 files
   ```

例 2–4 既存のリフェラルの変更

server4:/tank/docs などの別のファイルシステムを既存のリフェラルに追加する場合は、この新しいファイルシステムを指定して上記の手順 2 のコマンドを入力します。

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

add サブコマンドは、現在のリフェラルの情報を、コマンドからの新しい情報で単純に置き換えます。add サブコマンドは、既存のリフェラルに関連するファイルシステムをどのように変更するかを指定します。

▼ 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、次の手順に従います。

- リフェラルを削除します。

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

FedFS の管理

FedFS プロトコルを使用すると、フェデレーテッドファイルシステムを構築して維持できます。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含めることができ、異機種混在のグローバル名前空間が可能です。

▼ 名前空間データベース (NSDB) を作成する方法

NSDB は、単一の FedFS 名前空間に統合される各種サーバーからの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 サーバー上で実行されます。

始める前に　この手順では、root 役割になり、LDAP サーバーがインストールされている必要があります。

1. **FedFS の LDAP スキーマを構成します。**

   LDAP 構成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>
```

2. **FedFS データの識別名を作成します。**

   nsdb-update-nci(1M) のマニュアルページを参照してください。

```
# 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
```

▼ NSDB へのセキュアな接続を使用する方法

始める前に　この手順では、root 役割になり、LDAP サーバーがインストールされている必要があります。

1. **LDAP サーバー上:証明書を作成します。**

```
# 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
```

2. **LDAP サーバー上:LDAP 構成ファイルに宣言を追加します。**

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

3. **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
```

4. **NFS サーバーおよびクライアント上:接続エントリを更新します。**

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

▼ FedFS リフェラルを作成する方法

始める前に　この手順では、root 役割になり、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. **NSDB の接続エントリを作成します。**

   このコマンドは、LDAP サーバーに定義された NSDB と NFS サーバーに定義された NSDB の間の接続エントリを作成します。

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

2. **FedFS リフェラルを作成します。**

   -t オプションは、そのリフェラルのサービスタイプを選択します。

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

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、問題が発生する可能性があるのは主に、サーバー、クライアント、およびネットワークであることを覚えておいてください。このセクションで説明するのは、個々のコンポーネントを切り離して、正常に動作しない部分を見つけ出そうというものです。リモートマウントを正常に実行するには、サーバー上で mountd デーモンと nfsd デーモンが動作している必要があります。

デフォルトでは、すべてのマウントに -intr オプションが設定されます。プログラムが「server not responding」(サーバーが応答しません) というメッセージを出してハングアップした場合、キーボード割り込み (Ctrl-C) で終了できます。

ネットワークまたはサーバーに問題がある場合、ハードマウントされたリモートファイルにアクセスするプログラムの障害と、ソフトマウントされたリモートファイルにアクセスするプログラムの障害とは異なります。ハードマウントされた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リモートファイルシステムの場合、クライアントのシステムコールは、しばらく試行した後にエラーを返します。このエラーによって予想外のアプリケーションエラーやデータ破壊が発生する恐れがあるため、ソフトマウントは行わないでください。

ファイルシステムがハードマウントされていると、サーバーが応答に失敗した場合は、これ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 – 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の手順

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。次の項目をチェックしてください。

- 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 サービスを受けられるかどうか
- NFS サービスがサーバー上で動作しているかどうか

上記の項目をチェックする過程で、ネットワークのほかの部分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。いくつかのネームサービスでのデバッグ手順については、『Oracle Solaris Administration: Naming and Directory Services』で説明しています。また、上記の項目をチェックする過程で、クライアント側には問題がないことが判明することもあります。たとえば、作業領域のすべてのサブネットから、少なくとも 1 つの障害が発生したことが通知された場合などです。このような場合は、問題がサーバーかサーバー周辺のネットワークハードウェアで発生しているとみなし、クライアントではなく、サーバーでデバッグを開始する必要があります。

▼ NFS クライアントの接続性を確認する方法

1. クライアントから **NFS** サーバーに到達できることを確認します。クライアントで次のコマンドを入力します。

   ```
   % /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。72 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、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。

3. ネームサービスが実行されている場合は、クライアントが正しいホスト情報を受け取るために次のように入力します。

   ```
   % /usr/bin/getent hosts bee
   129.144.83.117     bee.eng.acme.com
   ```

4. ホスト情報に誤りがなく、クライアントからサーバーに接続できない場合は、別のクライアントから **ping** コマンドを実行します。

   別のクライアントから実行したコマンドが失敗したら、73 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. 別のクライアントとサーバーがソフトウェア的に接続されている場合は、**ping** コマンドを使用して元のクライアントとローカルネット上の他のシステムとの接続性を確認します。

   このコマンドが失敗する場合は、そのクライアントのネットワークソフトウェアの構成を確認します (`/etc/netmasks`、`svc:/system/name-service/switch` サービスに関連付けられたプロパティー情報など)。

6 (省略可能) **rpcinfo** コマンドの出力を確認します。

`rpcinfo` コマンドを使用しても「`program 100003 version 4 ready and waiting`」と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 2–3 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。

クライアントをネットワークの別の場所へ移動して確認します。

▼ NFS サーバーをリモートで確認する方法

NFS version 4 のサーバーを使用している場合は、UDP と MOUNT プロトコルをサポートする必要がないことに注意してください。

1 **NFS** サーバーで **NFS** サービスが実行されていることを、次のコマンドを入力して確認します。

```
% rpcinfo -s bee|egrep 'nfs|mountd'
 100003  3,2    tcp,udp,tcp6,upd6                nfs     superuser
 100005  3,2,1  ticots,ticotsord,tcp,tcp6,ticlts,udp,upd6  mountd  superuser
```

デーモンが起動していない場合は、74 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 サーバーで **nfsd** プロセスが応答することを確認します。

クライアント上で、次のコマンドを入力し、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 – 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、プログラムとバージョン番号が表示されます。`-t` オプションを使用すると、TCP 接続を検査できます。上記コマンドでエラーになる場合は、73 ページの「サーバーで NFS サービスを確認する方法」に進んでください。

3 サーバーで **mountd** が応答することを、次のコマンドを入力して確認します。

```
% /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、UDP プロトコルに関連しているプログラムとそのバージョン番号が出力されます。`-t` オプションを使用すると、TCP 接続を検査できます。エラーになる場合は、73 ページの「サーバーで NFS サービスを確認する方法」に進んでください。

4 ローカル **autofs** サービスを使用していた場合は、そのサービスを確認します。

```
% cd /net/wasp
```

/net か /home マウントポイントのうち、適切に動作する方を確認します。エラーになる場合は、次のコマンドを root としてクライアントから入力し、autofs サービスを再起動します。

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

5 サーバーのファイルシステムの共有が正常に行え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。この例で最初のクライアントが eng ネットグループの中にない場合、/usr/src ファイルシステムは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 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 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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、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 ネームサービスが動作している場合は、サーバーのネットワークソフトウェアの構成を確認します (**/etc/netmasks**、**svc:/system/name-service/switch** サービスに関連付けられたプロパティー情報など)。

5 次のコマンドを入力し、**rpcbind** デーモンが動作していることを確認します。

```
# /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、UDP プロトコルに関連しているプログラムとそのバージョン番号が出力されます。

6 次のコマンドを入力して、**nfsd** デーモンが動作していることを確認します。

```
# 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 – 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、UDP プロトコルに関連しているプログラムとそのバージョン番号が出力されます。`rpcinfo` に `-t` オプションを指定し、TCP 接続も確認します。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。74 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 次のコマンドを入力して、**mountd** デーモンが動作していることを確認します。

```
# /usr/bin/rpcinfo -u 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、UDP プロトコルに関連しているプログラムとそのバージョン番号が出力されます。`rpcinfo` に `-t` オプションを指定し、TCP 接続も確認します。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。74 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 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」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。

次のコマンドを入力します。

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

NFS ファイルサービスを提供しているホストを確認する方法

-m オプションを指定して nfsstat コマンドを実行し、最新の NFS 情報を取得します。現在のサーバー名は、「currserver=」のあとに表示されます。

```
% 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,wsize=32678,retrans=5
 Failover: noresponse=0, failover=0, remap=0, currserver=bee
```

▼ mount コマンドに使用されたオプションを確認する方法

無効なオプションに対する警告は一切発行されません。次の手順は、コマンド行に入力したオプション、または /etc/vfstab から指定したオプションが有効であるかどうかを判断するのに役立ちます。

たとえば、次のコマンドが実行されたとします。

```
# mount -F nfs -o ro,vers=2 bee:/export/share/local /mnt
```

1 次のコマンドを実行し、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,wsize=8192,
        retrans=5
```

bee からマウントされたファイルシステムは、プロトコルのバージョンが 2 に設定されています。nfsstat コマンドを使用しても、一部のオプションの情報は表示されません。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** で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、mnttab ファイルに記述されているオプションとコマンド行のオプションが一致していることを確認してください。このようにすると、nfsstat コマンドにより報告されなかったオプションを特定することができます。

```
# 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 の使用時に、問題の発生することがあります。このセクションでは、問題解決プロセスについてわかりやすく説明します。このセクションは、2 つのパートに分かれています。

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` の詳細形式 (`-v`) オプションにより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。そうしないと、原因がわからないまま問題に遭遇することになります。

次の節は、autofs のエラー時に表示されがちなエラーメッセージと、生じうる問題についての説明です。

`automount -v` により生成されるエラーメッセージ

`bad key` *key* `in direct map` *mapname*
: **説明:** 直接マップのスキャン中、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:** 間接マップのスキャン中、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* のファイルハンドルの提供を 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。例:

```
fake
/blat           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: サポートが必要な場合は、ご購入先に連絡してください。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: サポートが必要な場合は、ご購入先に連絡してください。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 mountpoints:` *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* で示されるサーバーにコンタクトしようとしましたが、応答がありません。

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: マップエントリが不適切な形式であり、autofs が処理できません。

対処方法: その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: サポートが必要な場合は、ご購入先に連絡してください。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: autofs は、ディレクトリではない *mountpoint* に示される場所に自分自身をマウントすることができません。

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、問題が特定されます。

対処方法: サポートが必要な場合は、ご購入先に連絡してください。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、問題が特定されます。

対処方法: サポートが必要な場合は、ご購入先に連絡してください。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: このようなエラーメッセージはすべて、複製されたファイルシステムのサーバーに対して確認を実行した際に問題が発生したことを示します。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、問題が特定されます。

対処方法: サポートが必要な場合は、ご購入先に連絡してください。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、パス名に関する `pathconf` 情報の取得に失敗しました。

対処方法: `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になります。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。ファイルへのこのような権限は、次のコマンドを入力することにより取り消す必要があります。

```
# chmod 644 /etc/auto_home
```

NFS のエラーメッセージ

このセクションでは、エラーメッセージとそのエラーを発生させる原因となった状態について説明し、1 つ以上の解決策を提供しています。

`Bad argument specified with index option - must be a file`
対処方法: index オプションにはファイル名を指定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、名前空間の中のサービス情報が更新されなかったときによく出力されます。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べきです。

```
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* の文字列によって、問題が特定されます。

対処方法: サポートが必要な場合は、ご購入先に連絡してください。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にもかかわらず NFS サーバーが公開ファイルハンドルをサポートしてい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: この問題を解決するには、公開ファイルハンドルを使用しないでマウント要求を行うか、NFS サーバーが公開ファイルハンドルをサポートするように構成し直します。

daemon `running already with pid` *pid*
説明: デーモンがすでに実行されています。

対処方法: 新たにデーモンを実行する場合は、現在のデーモンを終了し、新しいデーモンを開始します。

`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場合に表示されます。

対処方法: サポートが必要な場合は、ご購入先に連絡してください。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* を正しく開くことができなかった場合に表示されます。

対処方法: サポートが必要な場合は、ご購入先に連絡してください。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すると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: 何もする必要はありません。

filename`: File too large`
説明: NFS version 2 のクライアントが、2G バイトを超えるサイズの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 を使用しないでください。 version 3 または version 4 を使用してファイルシステムをマウントします。`nolargefiles` オプションについては、106 ページの「NFS ファイルシステム用の `mount` オプション」を参照してください。

`mount: ... server not responding:RPC_PMAP_FAILURE - RPC_TIMED_OUT`
説明: 実行レベルの誤りか、`rpcbind` の停止かハングアップのため、マウント先の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またはそこに到達できません。

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: ... server not responding: RPC_PROG_NOT_REGISTERED`
説明: マウント要求が `rpcbind` によって登録されているにもかかわらず、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: コンピュータ名が、クライアントのリストに載っていないか、マウントするファイルシステムにアクセスできるネットグループに含まれていません。

対処方法: `showmount -e` を実行し、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 サーバーでは、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、140 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: ただし、ユーザーが 17 個以上のグループに所属する必要がある場合の方法もあります。アクセス制御リストを使用すると、必要なアクセス特権を提供することができます。

`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ました。

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 プロトコルでは、大規模ファイルを処理できません。

対処方法: 大規模ファイルを扱う必要がある場合は、version 3 または version 4 を使用してください。

`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と、NFS サーバーに障害が発生する可能性があります。このメッセージは、NFS サーバー (*hostname*) がダウンしているか、サーバーか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: フェイルオーバー機能を使用している場合、*hostname* はサーバー名のリストになります。71 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 サーバーのリブート中に、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: 何もする必要はありません。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、次の理由により、`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の ACL を読み取ることができない。
- 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 に書き込みや変更を行うことができない。
- NFS version 4 の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 マッピングが失敗する。

詳細は、142 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: 次の手順を実行してください。

- ACL エントリ内のすべてのユーザーおよびグループ 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` 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。

ユーザーまたはグループをサーバーまたはクライアント上でマッピングできるかどうかを判断するには、143 ページの「ACL エントリ内のすべてのユーザーおよ

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。」にあるスクリプトを使用します。

`port` *number* `in nfs URL not the same as port` *number* `in port option`
説明: NFS URL のポート番号は、マウントの `-port` オプションのポート番号と一致していなければなりません。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 フェイルオーバー機能が正しく機能するためには、複製の 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していなければなりません。

対処方法: 複数のバージョンが混在することは許されません。

`replicated mounts must be read-only`
説明: NFS フェイルオーバー機能は、読み書き可能としてマウントされたファイルシステムでは動作しません。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: 複製されるマウントの場合、フェイルオーバーが発生するまでタイムアウトを待つ必要があります。

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` コマンドを使用して、`-public` オプションで共有されるファイルシステムが 1 つだけ選択されているようにする必要があります。公開ファイルハンドルは、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接続を確立できませんでした。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: サーバーを、ロックマネージャーを完全にサポートする新しいバージョンの OS にアップグレードします。

◆ ◆ ◆ 第 3 章

3 ネットワークファイルシステムへのアクセス(リファレンス)

この章では、NFS コマンドについて説明します。また、NFS 環境のさまざまな部分とそれらが互いにどのように連係するかについても説明します。

注 – 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、詳細について『Oracle Solaris 11.1 の管理: Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理』を参照してください。

NFS ファイル

ファイルによっては、いずれのコンピュータ上でも NFS アクティビティーをサポートする必要があるファイルがあります。その多くは ASCII ファイルで、いくつかはデータファイルです。表 3–1 にこのようなファイルとその機能をまとめます。

表 3–1 NFS ファイル

ファイル名	機能
/etc/default/fs	ローカルファイルシステムにおけるデフォルトファイルシステムのタイプを示します。
/etc/default/nfslogd	NFS サーバーログデーモン (nfslogd) の構成情報を示します。
/etc/dfs/dfstab	廃止: 共有されるローカルリソースを示します。
/etc/dfs/fstypes	リモートファイルシステムにおけるデフォルトファイルシステムのタイプを示します。
/etc/dfs/sharetab	共有されるローカルとリモートのリソースを示します。sharetab(4) のマニュアルページを参照してください。このファイルは編集しないでください。
/etc/mnttab	自動マウントしたディレクトリを含む、現在マウントしているファイルシステムを示します。mnttab(4) のマニュアルページを参照してください。このファイルは編集しないでください。
/etc/netconfig	トランスポートプロトコルを示します。このファイルは編集しないでください。
/etc/nfs/nfslog.conf	NFS サーバーログのための一般的な構成情報を示します。
/etc/nfs/nfslogtab	nfslogd によるログ後処理のための情報を示します。このファイルは編集しないでください。
/etc/nfssec.conf	NFS のセキュリティーサービスを示します。
/etc/rmtab	NFS クライアントがリモートでマウントしたファイルシステムを示します。rmtab(4) のマニュアルページを参照してください。このファイルは編集しないでください。
/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。vfstab(4) のマニュアルページを参照してください。

/etc/dfs/fstypes の最初のエントリは、リモートファイルシステムにおけるデフォルトファイルシステムのタイプとして利用されることがよくあります。このエントリは、NFS ファイルシステムのタイプをデフォルトとして定義します。

/etc/default/fs には、エントリが 1 つしかありません。 ローカルディスクにおけるデフォルトファイルシステムのタイプです。クライアントやサーバーでサポートするファイルシステムのタイプは、/kernel/fs のファイルを確認して決定することができます。

/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に使用されるいくつかのパラメータを定義します。次のパラメータを定義することができます。

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。このパラメータはログファイルが大きくなり過ぎないように使用します。

IDLE_TIME
: `nfslogd` が、バッファー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 によりバッファー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。このパラメータを使用すると、ファイルハンドルパスマッピングテーブルを常時更新しないで最新の状態に保つことができます。

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
: バッファー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 によりバッファー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ー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。各定義はタグと関連づけられています。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。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。特に指定しないかぎり、デフォルトのディレクトリは `/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*
バッファー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` が定義す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。次に、必要な変更の例を示します。

```
% 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
```

この例では、`log=publicftp` と共有するファイルシステムはすべて、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。

- バッファーファイルは、/var/nfs/buffers/workbuffer に保存されます。

手順については、34 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 3 またはマルチユーザーモードで稼動し始めたときに、複数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 とは異なり、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。

- 91 ページの「automountd デーモン」
- 92 ページの「lockd デーモン」
- 93 ページの「mountd デーモン」
- 94 ページの「nfs4cbd デーモン」
- 94 ページの「nfsd デーモン」
- 95 ページの「nfslogd デーモン」
- 95 ページの「nfsmapid デーモン」
- 102 ページの「reparsed デーモン」
- 102 ページの「statd デーモン」

automountd デーモン

このデーモンは 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*]

このコマンドは、次のように動作します。

- -T は、トレースを有効にします。
- -n は、すべての autofs ノード上で、ブラウズを無効にします。
- -v は、コンソールへのすべてのステータスメッセージを記録します。
- -D *name*=*value* は、*name* によって示された自動マウントマップ変数の値を *value* に置き換えます。

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。

コマンド行で行うのと同じ指定を、sharectl コマンドを使用して行うことができます。ただし、コマンド行オプションとは異なり、サービスの再起動、システムのリブート、およびシステムのアップグレードを行なったときも SMF リポジトリは指定を保持します。automountd デーモンに設定できるパラメータは次のとおりです。

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。デフォルトの値は 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。このキーワードは -automountd の n 引数と同等です。デフォルトの値は 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

このデーモンは NFS ファイルのレコードロックをサポートします。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。lockd(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 コマンドでパラメータを設定して使用することができます。次に、設定できるパラメータについて説明します。

注 – 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。次の grace_period の説明を参照してください。

LOCKD_GRACE_PERIOD と同様に、grace_period=*graceperiod* パラメータは、クライアントがサーバーのリブート後に NFS version 3 のロック (NLM が提供) と version 4 のロックを再要求するまでの秒数を設定します。つまり、grace_period の値は、NFS version 3 と NFS version 4 の両方についてロックリカバリの猶予期間を制御します。

`lockd_retransmit_timeout=`*timeout* パラメータは、ロックリクエストをリモートサーバーに再転送するまで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。

`lockd_servers=`*number* パラメータは、同時に処理できる `lockd` リクエストの最大数を指定します。デフォルト値は 1024 です。

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違います。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ー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

このデーモンは、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て、アクセス制御を行います。`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。このコマンドでは、`-v` オプションと `-r` オプションを使用できます。`mountd`(1M) のマニュアルページを参照してください。

`-v` オプションは、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` オプションは、その後のクライアントからのマウント要求をすべて拒絶します。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に加え、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。42 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。42 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

NFS version 4 クライアントの排他使用のための nfs4cbd は、NFS version 4 コールバックプログラムでの通信の終端を管理します。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

これは、他のクライアントからのファイルシステム要求を処理するデーモンです。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。オプションをすべて確認するには、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* パラメータは、接続型トランスポートを使用した NFS および TCP の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* パラメータは、接続型トランスポートごとの最大接続数を選択します。*#-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* パラメータは、サーバーが同時に処理できるリクエストの最大数を選択します。*nservers* のデフォルト値は 1024 です。nfsd に nservers オプションを指定して開始すると、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、これらの 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同等のコマンド行オプションはありません。

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。42 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。42 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 の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 のクライアントに委託しようとします。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、44 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、140 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

このデーモンは実行された処理のログ機能を提供します。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ーファイルに書き込まれます。次に `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。

パス名へのファイルハンドルの割り当ても `nfslogd` により行われます。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。

注 – NFS version 4 は、このデーモンを使用しません。

---

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。このプロトコルでは、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の属性をそれぞれ `user@nfsv4_domain`、`group@nfsv4_domain` の形式で文字列として交換する必要があります。

たとえば、`known_user` ユーザーに完全指定のホスト名が `system.example.com` である NFS version 4 クライアント上に UID 123456 が割り当てられているとします。このクライアントが NFS version 4 サーバーに要求を行うには、UID 123456 を

known_user@example.com に割り当ててから、この属性を NFS version 4 サーバーに送信する必要があります。NFS version 4 サーバーは、ユーザーとグループのファイル属性を user_or_group@nfsv4_domain 形式で受信することを予期します。サーバーがクライアントから known_user@example.com を受信すると、サーバーはこの文字列をローカルの UID 123456 に割り当て、配下のファイルシステムがこれを認識します。この機能では、ネットワーク上のすべての 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 のドメインがサーバー上の NFS version 4 のドメインと一致していることを前提としています。

注 – NFS version 4 のドメインが一致している場合でも、渡されたユーザー名またはグループ名をサーバーが認識しない場合、そのサーバーはそのユーザー名またはグループ名を一意の ID (整数値) に割り当てることができません。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。そうした状況が発生することを避けるために、管理者は NFS version 4 クライアントだけに存在する特別なアカウントを作成しないようにしてください。

NFS version 4 のクライアントとサーバーは、整数から文字列への変換と文字列から整数への変換に対応しています。たとえば、NFS version 4 サーバーが GETATTR 処理を受け取ると、配下のファイルシステムから取得した UID および GID をそれぞれの文字列表現に割り当てたうえで、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、クライアントが chown コマンドを受け取ると、新しい UID および GID を文字列表現に割り当ててから、SETATTR 処理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存在しない場合、特に同じ NFS version 4 ドメイン構成の中に存在しない場合には、サーバーはリモート手続き呼び出し(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上とサーバー上に存在している場合でも、そのドメインが一致しない場合には、サーバーが受け取った属性変更処理のうち、着信ユーザー文字列を整数値に割り当てて配下のファイルシステムが認識できるようにする必要がある処理 (SETATTR など) については、サーバーで拒否されます。NFS version 4 のクライアントとサーバーが正常に機能するには、それらの NFS version 4 ドメイン (文字列のうち、@ 記号のあとの部分) が一致しているべきです。
- NFS version 4 クライアントがサーバーから送信されたユーザー名またはグループ名を認識しない場合には、クライアントはその文字列を一意の ID (整数値) に割り当てることができません。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 の機能では、ファイル属性を変更する処理は失敗します。

クライアントとサーバーのドメイン名を変更するには、sharectl コマンドで次のオプションを使用します。

nfsmapid_domain
: クライアントとサーバーに共通の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の動作はオーバーライドされます。タスクの詳細は、42 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

次に、nfsmapid デーモンが、svc:system/name-service/switch と svc:/network/dns/client に見つかった SMF 構成情報をどのように使用するかについて説明します。

- 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ーへの変更は、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のファイルシステムを確実にマウントできるように、nfsmapid は DNS TXT リソースレコード (RR) _nfsv4idmapdomain の構成に依存しています。_nfsv4idmapdomain リソースレコードの構成の詳細については、98 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 サーバーを見つけてクライアントとサーバーの NFS version 4 ドメインの TXT レコードを検索できるように、必要なパラメータを使って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。

  - 98 ページの「優先ルール」
  - 100 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

優先ルール

nfsmapid が正しく動作するには、NFS version 4 のクライアントと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 ドメインが確実に一致するように、nfsmapid は次の厳密な優先ルールに従って動作します。

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、42 ページの「NFS サービスの設定」を参照してください。

   注 – NFSMAPID_DOMAIN 設定を使用する方法はスケーラブルではないため、大規模な配備を行う場合には推奨されません。

   ---

2. 値が nfsmapid_domain に割り当てられていない場合、デーモンは DNS TXT RR でドメイン名を確認します。nfsmapid は、resolver の一連のルーチン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 リポジトリで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薦めします。
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 および search の両方の指令が使用されています。nfsmapid デーモンは、search 指令のあとに最初に記載されているドメイン名である company.com を使用します。

   ```
   domain example.company.com
   search company.com foo.bar.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 ドメインの名前を取得します。より詳しく説明すると、/etc/defaultdomain ファイルが存在する場合には、nfsmapid は NFS version 4 ドメインのためにそのファイル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 は汎用性が高いので、NFS version 4 のドメイン名を格納して配布するための効率的なメカニズムです。また、DNS は本質的にスケーラブルなので、DNS TXT リソースレコードを使用する方法は、大規模な配備の NFS version 4 のドメインを構成

するうえで、もっとも推奨される方法です。エンタープライズレベルの DNS サーバーでは、`_nfsv4idmapdomain` TXT レコードを構成するようにしてください。このように構成すれば、NFS version 4 のクライアントまたはサーバーは DNS ツリーをたどることによって NFS version 4 ドメインを見つけることができます。

DNS サーバーから NFS version 4 のドメイン名を提供するように設定するときは、次の例のように入力することをお勧めします。

```
_nfsv4idmapdomain       IN       TXT           "foo.bar"
```

この例では、構成されるドメイン名は、二重引用符で囲まれている値です。`ttl` フィールドが指定されていないことと、ドメインが `owner` フィールドの値である `_nfsv4idmapdomain` に追加されていないことに注意してください。この構成により、TXT レコードで、Start-Of-Authority (SOA) レコードのゾーンの `${ORIGIN}` エントリを使用できるようになります。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.yourcorp.com.    IN    TXT    "foo.bar"
_nfsv4idmapdomain.yourcorp.com.           IN    TXT    "foo.bar"
```

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。柔軟性の向上により、低いレベルの DNS サブドメインが、自身の DNS TXT リソースレコード (RR) を定義できるようになりました。この機能により、低いレベルの DNS サブドメインが、高いレベルの DNS ドメインの定義した TXT レコードをオーバーライドできます。

注 – TXT レコードで指定したドメインには、任意の文字列を使用できます。この文字列は、NFS version 4 を使用するクライアントとサーバーの DNS ドメインと同じである必要はありません。 NFS version 4 データをほかの DNS ドメインと共有しないようにするオプションがあります。

NFS version 4 のドメインを確認する

ネットワークの NFS version 4 ドメインの値を割り当てる前に、ネットワークに NFS version 4 ドメインがすでに構成されているかどうかを確認します。次の例は、ネットワークの NFS version 4 ドメインを確認する方法を示します。

- NFS version 4 ドメインを DNS TXT RR で確認するには、`nslookup` コマンドまたは `dig` コマンドを使用します。

  `nslookup` コマンドの出力例を次に示します。

  ```
  # nslookup -q=txt _nfsv4idmapdomain
  Server:         10.255.255.255
  Address:        10.255.255.255#53
  ```

```
_nfsv4idmapdomain.example.company.com text = "company.com"
```

dig コマンドの出力例を次に示します。

```
# dig +domain=example.company.com -t TXT _nfsv4idmapdomain
...
;; QUESTION SECTION:
;_nfsv4idmapdomain.example.company.com. IN    TXT

;; ANSWER SECTION:
_nfsv4idmapdomain.example.company.com. 21600 IN TXT   "company.com"

;; AUTHORITY SECTION:
...
```

DNS TXT RR の設定方法については、98 ページの「nfsmapid と DNS TXT レコード」を参照してください。

- ネットワークに NFS version 4 の DNS TXT RR が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 ドメイン名で確認します。

```
# egrep domain /etc/resolv.conf
domain example.company.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 ドメイン構成でドメインを確認します。

```
# cat /system/volatile/nfs4_domain
company.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.example.company.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、101 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、101 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、コマンド行から次のコマンドを入力して、デフォルトの NFS ドメインバージョンを設定できます。

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

注 – DNS 特有のユビキタスでスケーラブルな性質のため、大規模な NFS version 4 配備のドメイン構成には DNS TXT レコードを引き続き使用することを強く推奨します。98 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

初期 Solaris 10 リリースの NFS version 4 では、ネットワーク内に複数の DNS ドメインが存在しているにもかかわらず、単一の UID および GID 名前空間しかない場合、すべてのクライアントが nfsmapid_domain に対して単一の値を使用する必要があります。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、98 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 を使用するように構成されていない場合は、システムの最初のブート時に、OS は sysidconfig ユーティリティーを使用して NFS version 4 のドメイン名に関する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 verion 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、次のプロンプトが表示されます。

```
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 – nfsmapid_domain の値が SMF リポジトリに存在する場合は、指定した [domain_name] 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
- 142 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。データは 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

lockd とともに動作し、ロックマネージャーにクラッシュ/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、サーバーのリブート中に、サーバー側 statd がクライアント側 statd にコンタクトします。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。クライアント側 statd は、サーバー上のクライアントのロックがクリアされるように、サーバー側 statd にクライアントがいつクラッシュしたかを通知します。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。

Solaris 7 で、statd がクライアントを追跡する方法が改善されました。以前のすべての Solaris リリースでは、statd は、クライアントごとにそのクライアントの修飾されていないホスト名を使用して、/var/statmon/sm にファイルを作成しました。そのため、同じホスト名の 2 つのクライアントが異なるドメインに存在する場合や、クライアントが NFS サーバーと異なるドメインに存在する場合に、このファイルのネーミングが原因となり問題が発生していました。修飾されていないホスト名はホスト名のみを表示し、ドメインや IP アドレスの情報がないため、以前のバージョンの statd にはこのようなクライアントを区別する方法がありませんでした。この問題を解決するため、Solaris 7 の statd は、修飾されていないホスト名に対してクライアントの IP アドレスを使用して /var/statmon/sm にシンボリックリンクを作成します。新規リンクは、次のようになります。

```
# ls -l /var/statmon/sm
lrwxrwxrwx   1 daemon          11 Apr 29 16:32 ipv4.192.168.255.255 -> myhost
lrwxrwxrwx   1 daemon          11 Apr 29 16:32 ipv6.fec0::56:a00:20ff:feb9:2734 -> v6host
--w-------   1 daemon          11 Apr 29 16:32 myhost
--w-------   1 daemon          11 Apr 29 16:32 v6host
```

この例では、クライアントのホスト名は myhost で、クライアントの IP アドレスは 192.168.255.255 です。ほかのホストが myhost という名前を持ち、ファイルシステムをマウントしていると、myhost というホスト名に対するシンボリックリンクは 2 つ作成されます。

注 – NFS version 4 は、このデーモンを使用しません。

NFS コマンド

これらのコマンドは、root として実行しないと、十分な効果が得られません。ただし、情報のリクエストは、すべてのユーザーが行うことができます。

- 103 ページの「automount コマンド」
- 104 ページの「clear_locks コマンド」
- 105 ページの「fsstat コマンド」
- 106 ページの「mount コマンド」
- 113 ページの「mountall コマンド」
- 124 ページの「nfsref コマンド」
- 114 ページの「sharectl コマンド」
- 116 ページの「share コマンド」
- 122 ページの「shareall コマンド」
- 123 ページの「showmount コマンド」
- 112 ページの「umount コマンド」
- 113 ページの「umountall コマンド」
- 122 ページの「unshare コマンド」
- 122 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、124 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された状態でいる時間 (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 にある各エントリをチェックした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 を使用して、df の出力にフィルタをかけます。

この継続時間を調節すると、オートマウンタ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ができません。オートマウンタマップの変更方法については、55 ページの「マップの修正」を参照してください。

コマンド行で行うのと同じ指定を、sharectl コマンドを使用して行うことができます。ただし、コマンド行オプションとは異なり、サービスの再起動、システムのリブート、およびシステムのアップグレードを行なったときも SMF リポジトリは指定を保持します。automount コマンドに設定できるパラメータは次のとおりです。

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。デフォルトの値は 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 NFS クライアントに対するロックが解除されます。

```
# clear_locks tulip
```

-s オプションを指定すると、どの NFS ホストからロックを解除するかを指定できます。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。次の場合、クライアントによるロックが 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以外には使用しないでください。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 ユーティリティーを使用して、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監視できます。出力のカスタマイズを可能にするオプションが多数用意されています。次に例を示します。

次の例では、NFS version 3、version 4、およびルートマウントポイントに対する出力を表示しています。

```
% 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、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   rwulock
  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、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 の標準インストールには、さまざまな種類のファイルシステムが含まれています。ファイルシステムの種類ごとにマニュアルページがあり、その種類に対して mount を実行するときに使用可能なオプションのリストが示されています。NFS ファイルシステムについては、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。

Solaris 7 で、server:/pathname という標準の構文の代わりに NFS URL を使用して NFS サーバー上のマウントするパス名を指定することが可能になりました。詳細は、40 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 – このバージョンの mount コマンドでは、無効なオプションに関する警告メッセージは表示されません。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

NFS ファイルシステムのマウント時に -o フラグのあとに指定できるオプションの一部を、次に示します。オプションの完全な一覧については、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ときの再試行の方法を選択するオプションです。bg オプションの場合はバックグラウンドで、fg オプションの場合はフォアグラウンドでマウントが試みられます。デフォルトは fg です。常に使用可能にしておく必要のあるファイルシステムに対しては fg が適しています。fg オプションを指定すると、マウントが完了するまで、次の処理は行われません。bg は、マウント要求が完了しなくても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の処理に適しています。

forcedirectio
: このオプションは、大規模の連続したデータ転送のパフォーマンスを向上させます。データは直接ユーザーバッファーにコピーされます。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。この機能はデフォルトではオフです。

これまで、書き込み要求はすべて、NFS クライアントと NFS サーバーの両方で直列化されていました。今回の NFS クライアントの変更により、単一ファイルに対する並行書き込み、並行読み取り/書き込みを、アプリケーションから実行できるようになりました。この機能をクライアント上で有効にするには、mount コマンドオプション `forcedirectio` を使用します。このオプションを使用した場合、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に対して、この機能が有効になります。この機能をクライアントの単一のファイルに対してのみ有効にするには、 `directio()` インタフェースを使用します。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、109 ページの「`mount` コマンドの使用」を参照してください。

largefiles
: このオプションを使用すれば、2G バイトより大きいファイルにアクセスできます。大規模ファイルにアクセスできるかどうかは、サーバーでしか制御できません。したがって、このオプションは NFS version 3 のマウントでは無視されます。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 プロトコルを使用したマウントで `largefiles` オプションを指定すると、エラーが発生してマウントできません。

nolargefiles
: この UFS マウント用のオプションを指定すると、ファイルシステム上に大規模ファイルが存在できないことが保証されます。`mount_ufs(1M)` のマニュアルページを参照してください。大規模ファイルの存在は 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場合は、`nolargefiles` オプションを指定できません。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と、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。

次の表は、nosuid または suid を devices または nodevices、および setuid または nosetuid と組み合わせることによる結果を示しています。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。このオプションを使用してリモートファイルシステムのマウントを行うと、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拡大する可能性を減らすことができます。これらのオプションはすべて、Oracle Solaris ファイルシステム全体で使用可能です。

public
このオプションを指定すると、NFS サーバーにアクセスするときに必ず公開ファイルハンドルを使用するようになります。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可能と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ス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* の値は、次のいずれかです。

  - Kerberos version 5 認証サービス用の `krb5` を使用する。
  - 整合性を指定する Kerberos version 5 用の `krb5i` を使用する。
  - 機密性を指定する Kerberos version 5 用の `krb5p` を使用する。
  - 認証なしの `none` を使用する。
  - Diffie-Hellman (DH) 認証用の `dh` を使用する。
  - UNIX 標準認証用の `sys` を使用する。

  モードは、`/etc/nfssec.conf` ファイルにも定義されます。

soft|hard
: `soft` オプションを指定してマウントされた NFS ファイルシステムは、サーバーが応答しなくなるとエラーを返します。`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確認する場合は、`soft` が使用されているとルーティングの問題などによってアプリケーションが正しく判断できず、ファイルが破壊されることがあります。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

次の例を参照してください。

- NFS version 2 または 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としてマウントします。

  ```
  # 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、次のコマンド行で同じマウントを行えます。

  ```
  # mount -F nfs -o vers=4 -r bee:/export/share/man /usr/man
  ```

- NFS version 2 または version 3 では、次のコマンドは -O オプションを使用しているため、すでに /usr/man がマウントされている場合でも、強制的にマニュアルページをサーバー bee からローカルシステムにマウントします。次を参照してください。

  ```
  # mount -F nfs -O bee:/export/share/man /usr/man
  ```

  NFS version 4 では、次のコマンド行で同じマウントを行えます。

  ```
  # mount -F nfs -o vers=4 -O bee:/export/share/man /usr/man
  ```

- NFS version 2 または version 3 では、次のコマンドはクライアント側フェイルオーバー機能を使用します。

  ```
  # mount -F nfs -r bee,wasp:/export/share/man /usr/man
  ```

  NFS version 4 では、次のコマンド行はクライアント側フェイルオーバー機能を使用します。

  ```
  # mount -F nfs -o vers=4 -r bee,wasp:/export/share/man /usr/man
  ```

注 – 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、version 2 と version 3 のサーバーを同時に使用しないでください。autofs を実行するときは、両サーバーを同時に使用することができます。autofs により、version 2 または version 3 のサーバーの最適な組み合わせが自動的に選択されます。

- 次に、NFS version 2 または version 3 において、mount コマンドに NFS URL を使用する例を示します。

  ```
  # mount -F nfs nfs://bee//export/share/man /usr/man
  ```

  次に、NFS version 4 において、mount コマンドに NFS URL を使用する例を示します。

  ```
  # mount -F nfs -o vers=4 nfs://bee//export/share/man /usr/man
  ```

- forcedirectio マウントオプションを使用すると、ファイルに対してクライアントが、並行書き込みと並行読み取り/書き込みを行えるようになります。次に例を示します。

  ```
  # 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 – 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。要求が直列化されると、次のことが発生します。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。

- `mount` コマンドに引数を指定しないと、クライアントにマウントされたファイルシステムが表示されます。次を参照してください。

```
% 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

このコマンドにより、現在マウントされているリモートファイルシステムが解除されます。umount コマンドは、テストのために -V オプションをサポートしています。また、-a オプションを使用することによって 1 度に複数のファイルシステムをアンマウントできます。-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。マウントポイントを指定しないと、/etc/mnttab のリストにあるファイルシステムのうち必須でないものすべてのアンマウントが試みられます。必須のファイルシステムとは、/、/usr、/var、/proc、/dev/fd、/tmp などです。ファイルシステムがすでにマウントされていて、/etc/mnttab に項目が指定されている場合、ファイルシステムのタイプのフラグを指定する必要はありません。

-f オプションを指定すると、使用中のファイルシステムが強制的にアンマウントされます。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と、ファイルへの書き込み中だった場合には、データを損失することがあります。

次に例を示します。

例 3–1 ファイルシステムをアンマウントする

次の例は、/usr/man にマウントしたファイルシステムをアンマウントします。

```
# umount /usr/man
```

例 3–2 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、ファイルシステムのアンマウント自体は実行されないことに注意してください。

mountall コマンド

このコマンドを使用すると、ファイルシステムテーブルに一覧表示されたすべてのファイルシステム、または特定グループのファイルシステムをマウントできます。このコマンドを実行すると、次の操作を実行することができます。

- -F *FSType* オプションを使用して、ファイルシステムのタイプを選択する
- -r オプションを使用して、ファイルシステムテーブル中に一覧表示されたリモートファイルシステムをすべて選択する
- -l オプションを使用して、ローカルファイルシステムをすべて選択する

NFS ファイルシステムタイプと指定されているファイルシステムはすべてリモートファイルシステムなので、これらのオプションは余分な指定になることがあります。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では、同じ結果が得られます。

```
# mountall -F nfs
```

```
# mountall -F nfs -r
```

umountall コマンド

このコマンドを使用すると、ファイルシステムのグループをアンマウントできます。-k オプションは、*mount-point* に関連付けられているプロセスを終了させるために fuser -k *mount-point* コマンドを実行します。-s オプションは、アンマウントを並行処理しないことを示します。-l は、ローカルファイルシステムだけを使用することを、-r はリモートファイルシステムだけを使用することを示します。-h *host* オプションは、指定されたホストのファイルシステムをすべてアンマウントす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したすべてのファイルシステムがアンマウントされます。

```
# umountall -r
```

次の例では、bee サーバーからマウントしたすべてのファイルシステムがアンマウントされます。

```
# umountall -h bee
```

sharectl コマンド

このリリースには sharectl ユーティリティーが含まれています。これは、NFS などのファイル共有プロトコルの構成と管理を行うことができる管理ツールです。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ーを設定する
- 特定のプロトコルのプロパティー値を表示する
- プロトコルのステータスを取得する

sharectl ユーティリティーは次の構文を使用します。

```
# sharectl subcommand [option] [protocol]
```

sharectl 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。

表 3–2　sharectl ユーティリティーのサブコマンド

サブコマンド	説明
set	ファイル共有プロトコルのプロパティーを定義します。プロパティーとプロパティー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ーとプロパティー値を表示します。
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。

sharectl 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。

- sharectl(1M) のマニュアルページ
- 114 ページの「set サブコマンド」
- 115 ページの「get サブコマンド」
- 116 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ーを定義する 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ーを定義します。

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
```

注 –

- set サブコマンドを使用するには root の特権が必要です。
- このコマンド行構文を追加のプロパティー値ごとに繰り返す必要はありません。同じコマンド行で -p オプションを複数回使用して、複数のプロパティーを定義できます。

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
```

get サブコマンド

指定されたプロトコルのプロパティーとプロパティー値を表示する 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ーのプロパティー値を特定します。-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ー値が表示されます。

get サブコマンドは次の構文を使用します。

```
# sharectl get [-h] [-p property] protocol
```

注 – 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ーです。

```
# 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ー値が表示されます。

```
# 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=company.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

指定されたプロトコルが有効か無効かを表示する status 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。

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

このコマンドを使用すると、NFS サーバーのローカルファイルシステムをマウントできるようになります。また、システム上のファイルシステムのうち、現在共有しているもののリストを表示します。NFS サーバーが動作していないと、share コマンドは使用できません。

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 が同じディスクスライスにある場合は、/usr または /usr/local を共有できます。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しているファイルシステムに、読み取りと書き込みが可能な状態で共有しているファイルシステムのファイルハンドルでアクセスすることができます。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。より安全にこれらのファイルシステムを使用するには、読み取りと書き込み

が設定されているファイルシステムを、読み取り専用で共有し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ディスクスライスに配置します。

注 – ファイルシステムの共有を解除してから再度共有するとき、NFS version 4 がどのように動作するかについては、133 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。

非ファイルシステム用 share オプション

-o フラグに指定できるオプションの一部を次に示します。

rw|ro
: *pathname* に指定したファイルシステムを、すべてのクライアントに対して読み取りと書き込みの両方が可能な状態で共有するか、読み取り専用で共有するかを指定します。

rw=*accesslist*
: 指定されたクライアントに対してのみ、ファイルシステムが読み取り/書き込みモードで共有されます。それ以外の要求は拒否されます。*accesslist* に定義されるクライアントのリストは、Solaris 2.6 から拡張されました。詳細については、120 ページの「share コマンドを使ってアクセスリストを設定する」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 用 share オプション

NFS ファイルシステムで指定できるオプションは、次のとおりです。

aclok
: このオプションを指定すると、NFS version 2 プロトコルをサポートしている 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 11.1 の管理: セキュリティーサービス』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 – アクセス制御リスト (ACL) を使用するには、クライアントとサーバーが、NFS version 3 プロトコルおよび NFS_ACL プロトコルをサポートしているソフトウェアを実行している必要があります。NFS version 3 プロトコルしかサポートしていないソフトウェアの場合、クライアントは正しいアクセス権を取得できますが、ACL を操作することはできません。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*
uid 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。*uid* を `-1` に設定すると、認証されていないユーザーからのアクセスは拒否されます。`anon=0` とするとルートアクセス権を与えることができますが、これのオプションを指定すると、認証されていないユーザーにルートアクセス権を与えることになるため、代わりに `root` オプションを使用してください。

index=*filename*
`-index=`*filename* オプションを使用すると、ユーザーが NFS URL にアクセスすると、ディレクトリのリストが表示されるのではなく、HTML (HyperText Markup Language) ファイルが強制的に読み込まれます。これは、HTTP URL がアクセスしているディレクトリに `index.html` ファイルが見つかると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*
このオプションは、ファイルシステム用の NFS サーバーログ構成情報の入った `/etc/nfs/nfslog.conf` 内のタグを指定します。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
このオプションを使用すると、`setuid` モードまたは `setgid` モードを有効にしようとしても無視されます。NFS クライアントは、`setuid` か `setgid` のビットがオンの状態では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=*accesslist*
サーバーが、リスト上のホストに対してルートアクセス権を与えます。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` 以外だと、*accesslist* に指定できるのはクライアントホスト名だけです。*accesslist* に定義されたクライアントのリストは、Solaris 2.6 で拡張されました。詳細については、120 ページの「`share` コマンドを使ってアクセスリストを設定する」を参照してください。

注意 – ほかのホストにルートアクセス権を与えるには、広い範囲でセキュリティーが保証されていることが前提です。`-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 の値は、AUTH_SYS 認証で、`exportfs(1B)` で取得されたアドレスのリストにクライアントの IP アドレスが含まれているかどうかを検査するために使用します。一致が見つかった場合、クライアントに共有ファイルシステムへのルートアクセス権が与えられます。

root=*host-name*
AUTH_SYS または RPCSEC_GSS などのセキュアな 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の主体名のリストに、クライアントの主体名が含まれているかどうかを検査します。クライアント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。*host-name* の値を使用する場合、アクセスリスト上のクライアントには主体名の資格が必要になります。Kerberos V の場合、クライアントには `root/`*hostname.fully.qualified*`@REALM` の主体名の有効な keytab エントリが必要です。詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「Kerberos クライアントの構成」を参照してください。

sec=*mode*[:*mode*]
mode は、ファイルシステムへのアクセス権を取得するために必要なセキュリティーモードです。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 `-mode` オプションはほかの `-mode` が検出されるまで、後続のすべての `-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。

share コマンドを使ってアクセスリストを設定する

accesslist には、ドメイン名、サブネット番号、およびアクセス権を拒否するエントリのほか、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。

次のコマンドは、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 には読み取りと書き込みの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 ネットグループのすべてのホストで読み取りだけができるようになります。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 – 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 コマンドを何度も実行しても、最後に実行されたコマンドだけが有効になります。次のコマンドでは、3 つのクライアントシステムで読み取りと書き込みができますが、rose と tulip では、ファイルシステムに root でアクセスできます。

```
# 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場合は、正しいセキュリティーモードの後に -ro、-ro=、-rw、-rw=、-root、および -window オプションを必ず含めるようにしてください。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しかマウントできません。ホスト tulip と lilac は、Diffie-Hellman (DH) 認証を使用すれば読み書き可能でファイルシステムをマウントできます。これらのオプションを指定すると、tulip および lilac は、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することができます。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。次のエントリは、マウントから eng.example.com ドメイン内のすべてのホストへのアクセスを許可するためのものです。

```
# share -F nfs -o ro=.:.eng.example.com /export/share/man
```

この例で、「.」は NIS 名前空間を通じて一致するすべてのホストに対応します。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」というエントリは、名前空間の解決に DNS を使用するすべてのホストに一致します。DNS が返すホスト名は必ず完全指定の名前になるので、DNS と他の名前空間を組み合わせると長いエントリが必要です。

実際のネットワーク番号かネットワーク名の前に「@」を指定すると、アクセスリストの中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 内か 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。例:

```
# 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 の詳細は、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のうち、rose を除いたすべてに対してアクセスが許可されます。

unshare コマンド

このコマンドを使用すると、それまでクライアントでのマウントに使用できたファイルシステムが使用できなくなります。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。クライアントにはファイルシステムがまだマウントされている可能性がありますが、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 – ファイルシステムの共有を解除してから再度共有するとき、NFS version 4 がどのように動作するかについては、133 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、指定したファイルシステムの共有が解除されます。

```
# unshare /usr/src
```

shareall コマンド

このコマンドを使用すると、複数のファイルシステムを共有することができます。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一覧表示されているすべてのファイルシステムが共有されます。

```
# shareall /etc/dfs/special_dfstab
```

unshareall コマンド

このコマンドを使用すると、現在共有されているリソースがすべて使用できなくなります。`-F` *FSType* オプションによって、`/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します。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。デフォルトの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、NFS タイプのファイルシステムの共有がすべて解除されます。

```
# unshareall -F nfs
```

showmount コマンド

このコマンドは、次のいずれかを表示します。

- 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およびクライアントのアクセス情報

注 – showmount コマンドを使用すると、NFS version 2 と version 3 のエクスポートだけが表示され、NFS version 4 のエクスポートは表示されません。

コマンドは、次のような構文になります。

showmount [-ade] [*hostname*]

-a　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。

-d　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。

-e　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。

hostname　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、ローカルホストの情報が表示されます。

次のコマンドでは、すべてのクライアント、およびマウントしたローカルディレクトリが表示されます。

```
# showmount -a bee
lilac:/export/share/man
lilac:/usr/src
rose:/usr/src
tulip:/export/share/man
```

次のコマンドでは、マウントしたディレクトリが表示されます。

```
# showmount -d bee
/export/share/man
/usr/src
```

次のコマンドでは、共有しているファイルシステムが表示されます。

```
# showmount -e bee
/usr/src                                (everyone)
/export/share/man                     eng
```

/network/nfs/server:default サービスの nfs_props/showmount_info プロパティーは、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、そのクライアントがマウントできるものしか表示されません。ほかのクライアントに関する情報は表示されません。このプロパティーの変更手順については、例 2–3 を参照してください。

nfsref コマンド

nfsref コマンドは、NFSv4 リフェラルの追加、削除、または一覧表示に使用します。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選択します。

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

これらは、FedFS サービスに関連したコマンドです。

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の場所を示します。
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、68 ページの「FedFS の管理」を参照してください。

NFS のトラブルシューティング用のコマンド

NFS のトラブルシューティングには次のコマンドを使用します。

nfsstat コマンド

このコマンドを使用すると、NFS と RPC 接続について統計情報を収集できます。このコマンドの構文は次のとおりです。

```
nfsstat [ -cmnrsz ]
```

`-c` クライアント側の情報を表示します

`-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で、NFS の情報が表示されるように指定します

`-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

コマンド行にオプションを指定しないと、`-cnrs` 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と、以前のパフォーマンスの有効な記録となります。

次の例を参照してください。

```
# 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%
```

このリストは、NFS サーバーの統計の例です。最初の 5 行は RPC に関するもので、残りの部分は NFS のアクティビティーのレポートです。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不良メッセージ数を示しています。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ーが発生します。この数値の急激な上昇は障害の可能性を示すものなので、調査が必要です。NFS version 2 の統計で注意が必要なのは、setattr、write、create、remove、rename、link、symlink、mkdir、および rmdir です。NFS version 3 と version 4 では、commit の値に特に注意します。ある NFS サーバーの commit レベルが、それと同等のサーバーと比較して高い場合は、NFS クライアントに十分なメモリーがあるかどうかを確認してください。サーバーの commit オペレーションの数は、クライアントにリソースがない場合に上昇します。

pstack コマンド

このコマンドを使用すると、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、必ずプロセスの所有者、または root として実行してください。pstack を使用して、プロセスがハングアップした場所を判断します。使用できるオプションは、確認するプロセスの PID だけです。proc(1) のマニュアルページを参照してください。

次の例では、実行中の nfsd プロセスを確認しています。

```
# /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, effff14, 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。74 ページの「NFS サービスを再起動する方法」の指示に従って問題を解決してください。ハングアップしたプログラムによって問題が発生しているかどうかを確実に判断するには、70 ページの「NFS のトラブルシューティングの手順」を参照してください。

rpcinfo コマンド

このコマンドは、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 サービスの変更にも使用できます。このコマンドには、たくさんのオプ

ションがあります。rpcinfo(1M) のマニュアルページを参照してください。次は、このコマンドで使用できるオプションの構文です。

```
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	サービスに使用するトランスポートまたはプロトコルを選択します
hostname	必要な情報の取得元のサーバーのホスト名を選択します
progname	情報の取得対象の RPC プログラムを選択します

hostname を指定しないと、ローカルホスト名が使用されます。*progname* の代わりに RPC プログラム番号が使用できますが、ユーザーが覚えやすいのは番号よりも名前です。NFS version 3 が実行され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
- 使用されているトランスポートプロトコル
- RPC サービス名
- RPC サービスの所有者

次の例では、サーバーで実行されている RPC サービスに関する情報を収集しています。生成されたテキストには `sort` コマンドのフィルタをかけ、より読みやすくしています。この例では、RPC サービスの数行を省略しています。

```
% 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、サーバーの特定トランスポートを選択して、RPC サービスの情報を収集する方法について説明しています。最初の例では、TCP で実行されている `mountd` サービスをチェックしています。2 番目の例では、UDP で実行されている NFS サービスをチェックしています。

```
% 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
% rpcinfo -u bee nfs
program 100003 version 2 ready and waiting
program 100003 version 3 ready and waiting
```

snoop コマンド

このコマンドは、ネットワーク上のパケットの監視によく使用されます。`snoop` コマンドは、`root` で実行する必要があります。このコマンドは、クライアントとサーバーの両方で、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。使用できるオプションは多数あります。snoop(1M) のマニュアルページを参照してください。次で、このコマンドの概要を説明します。

`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があるサーバーで特に有効です。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も、十分に使用できるデータを生成できます。

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

このコマンドを使用すると、プロセスがハングアップしたかどうかを確認できます。truss コマンドは、必ずプロセスの所有者、または root として実行してください。このコマンドに指定できるオプションは多数あります。truss(1) のマニュアルページを参照してください。次で、このコマンドの構文を説明します。

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 には、追跡するシステムコールをコンマで区切って指定することもできます。また、*syscall* の指定を ! で始めると、そのシステムコールは追跡されなくなります。

次の例は、プロセスが新しいクライアントからの接続要求を待っていることを示しています。

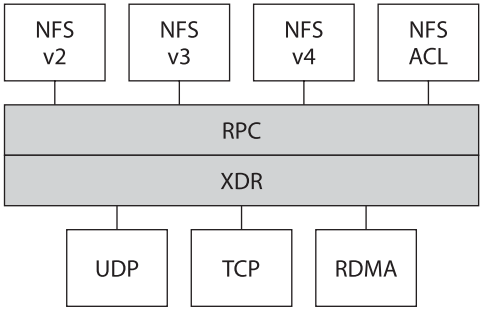
```
# /usr/bin/truss -p 243
poll(0x00024D50, 2, -1)         (sleeping...)
```

これは正常な反応です。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。74 ページの「NFS サービスを再起動する方法」の指示に従ってハングアップの問題を解決してください。ハングアップしたプログラムによって問題が発生しているかどうかを確実に判断するには、70 ページの「NFS のトラブルシューティングの手順」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 のデフォルトのトランスポートプロトコルは RDMA (Remote Direct Memory Access) プロトコルです。RDMA は、高速ネットワーク上でデータのメモリー間転送を行うテクノロジです。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。また、データを直接配置できます。これは、データのコピーを省略し、さらに CPU の介入も省略します。このように、RDMA はホストの CPU を解放するだけでなく、ホストのメモリーと入出力バスの接続も減らします。この機能を提供するために、RDMA は、SPARC プラットフォーム上の InfiniBand のインターコネクト入出力テクノロジと Oracle Solaris オペレーティングシステムを組み合わせます。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 3–1 その他のプロトコルとの RDMA の関係



NFS は RPC の上に重ねて階層化したプロトコル群です。
XDR (eXternal Data Representation) 層は、RPC の引数や結果を、UDP、TCP、RDMA などいくつかの RPC トランスポートの 1 つにエンコードします。

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タマップ vfstab および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で SPARC プラットフォームに InfiniBand 接続が存在する場合、RDMA トランスポート経由の NFS マウントは透過的に実行されます。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 マウントは強制的に RDMA だけを使用するようになります。

TCP と UDP のみが使用されるように指定するには、`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)` のマニュアルページおよび 106 ページの「`mount` コマンド」を参照してください。

注 – 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 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 専用であることに注意してください。

注 – 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、詳細について『Oracle Solaris 11.1 の管理: Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理』を参照してください。

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プロトコル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 version 3 をサポートしていると、version 3 が使用されます。クライアントまたはサーバーが version 2 しかサポートしていないと、version 2 が使用されます。

`sharectl` コマンドを使用して、`client_versmin`、`client_versmax`、`server_versmin`、および `server_versmax` パラメータを設定できます。これらのキーワードのデフォルト値に代わって、クライアントとサーバーに最小値と最大値を指定できます。クライアントとサーバーの最小値は 2 がデフォルトで、最大値は 4 がデフォルトです。サーバーによってサポートされるバージョンを検出するために、NFS クライアントは `client_versmax` の設定から始めて、`client_versmin` のバージョン設定に到るまで各バージョンを試行し続けます。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` および `client_versmin =2` の場合、クライアントは最初に version 4、次に version 3、最後に version 2 を試行します。`client_versmax` と `client_versmax` が同じ値に設定されていると、クライアントは常にこのバージョン

を使用し、その他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 – 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_nfs(1M)` のマニュアルページを参照してください。

手順については、42 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

version 4 の NFS は大幅に変更が行われました。このセクションでは、これらの新しい機能を説明します。

- 133 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」
- 134 ページの「NFS version 4 におけるファイルシステムの名前空間」
- 136 ページの「NFS version 4 における揮発性ファイルハンドル」
- 137 ページの「NFS version 4 におけるクライアント回復」
- 139 ページの「NFS version 4 における `OPEN` 共有サポート」
- 140 ページの「NFS version 4 における委託」
- 142 ページの「NFS version 4 での ACL と `nfsmapid`」
- 148 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」

注 – Solaris 10 以降のリリースでは、NFS version 4 は LIPKEY/SPKM セキュリティー方式をサポートしません。また、`mountd`、`nfslogd`、および `statd` デーモンを使用しません。

NFS version 4 の使用に関する手順については、42 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 では、クライアントが共有を解除されたファイルシステムにアクセスしようとすると、サーバーはエラーコードを返します。ただし、NFS version 3 では、ファイルシステムが共有されなくなる前に、サーバーは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度共有されるとき、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、ファイルシステムの共有を解除するとき、そのファイルシステムにあるオープンファイルまたはファイルロックの状態がすべて削除されます。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りすると、エラーを受け取ります。通常、このエラーは、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、サーバーの状態は削除されません。

関連情報については、137 ページの「NFS version 4 におけるクライアント回復」を参照するか、`unshare_nfs(1M)` のマニュアルページ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名前空間

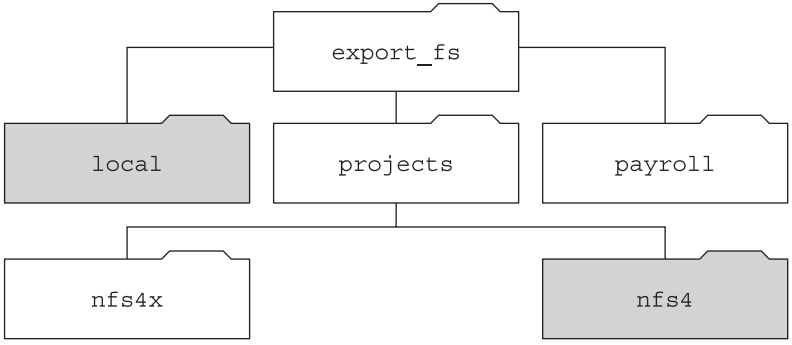
NFS version 4 サーバーは、擬似ファイルシステムを作成し管理します。擬似ファイルシステムにより、クライアントは、サーバー上のエクスポートされた全ファイルにシームレスにアクセスできます。NFS version 4 より前のバージョンには、擬似ファイルシステムがありません。クライアントは、アクセスする各共有サーバーのファイルシステムに強制的にマウントされます。次のような例を考えます。

図 3–2 サーバーのファイルシステムとクライアントのファイルシステムの表示

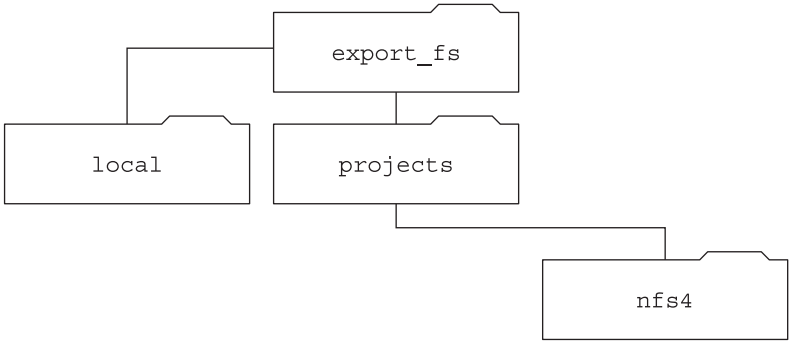
サーバーエクスポート：
/export_fs/local
/export_fs/projects/nfs4

サーバーファイルシステム：
/
/export_fs

サーバーファイルシステム：



サーバーの export_fs **ディレクトリのクライアント表示**：



エクスポートされたディレクトリ

クライアントには payroll ディレクトリと nfs4x ディレクトリは表示されません。これらのディレクトリはエクスポートされておらず、エクスポートされたディレクトリには通じていないためです。ただし、local ディレクトリは、エクスポートされたディレクトリであるため、クライアントに表示されます。projects ディレクトリは、エクスポートされたディレクトリ nfs4 に通じているため、クライアントに表示されます。このように、明示的にエクスポートされていないサーバーの名前空間の部分は、擬似ファイルシステムで橋渡しされます。この擬似ファイルシステムは、エクスポートされたディレクトリ、およびサーバーのエクスポートに通じるディレクトリだけを表示します。

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により、クライアントはエクスポートされたファイルシステムの階層を検索できるようになります。このようにして、クライアントの擬似ファイルシステムの表示は、エクスポートされたファイルシステムに通じるパスに限定されます。

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。しかし、NFS version 4 では、サーバーの名前空間が次のことを行います。

- クライアントのファイルシステム表示を、サーバーのエクスポートに通じるディレクトリに限定します。
- クライアントが配下の各ファイルシステムをマウントしなくても、サーバーのエクスポートにシームレスにアクセスできるようにします。前述の例を参照してください。オペレーティングシステムが異なるとき、クライアントはサーバーの各ファイルシステムをマウントする必要のある場合があります。

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、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 version 3 では、サーバーは持続的ファイルハンドルを返しました。したがって、クライアントは、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。例:

- 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。
- サーバーをリブートする必要があった場合、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 を操作するためのファイルとディレクトリを識別するこの方法は、多くの UNIX ベースのサーバーに適しています。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、NFS version 4 プロトコルは、サーバーがそのファイルハンドルが揮発性であることを宣言できるようにします。したがって、ファイルハンドルが変更されます。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 version 2 と 3 のように、Oracle Solaris NFS version 4 サーバーは常に持続的ファイルハンドルを提供します。ただし、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、そのサーバーが揮発性ファイルハンドルを使用する場合、揮発性ファイルハンドルをサポートする必要があります。特に、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知らせている場合は、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
- 操作をもう一度実行する

注 – サーバーは、どのファイルハンドルが持続的あるいは揮発性かを、クライアントに常に知らせます。

揮発性ファイルハンドルは、次のいずれかの理由により期限切れになります。

- 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検索できない場合、エラーメッセージが `syslog` ファイルに追加されます。このファイルにアクセスしようとすると、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。クライアントとサーバーが次の項目に関する現在の情報を管理するとき、プロトコルはステートフルです。

- オープンファイル
- ファイルロック

サーバーのクラッシュなどの障害が発生したとき、クライアントとサーバーは連携して、障害が発生する前のオープン状態とロック状態を再度確立します。

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して、サーバーの状態の再構築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して知られています。

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たとき、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 basil.example.company.com
```

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。ファイルのオープンやファイルロックの設定の新しい要求は、サーバーが回復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したとき、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 basil.example.company.com
```

クライアントは、サーバーへの状態情報の送信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、その他のクライアントがサーバーに状態情報を送信するプロセスを完了していない可能性があります。したがって、しばらくの間、サーバーはオープンまたはロック要求を受け付けません。この期間は猶予期間として知られており、すべてのクライアントが回復を完了できるように指定されています。

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とき、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、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ることに注意してください。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しません。したがって、これらのコマンドは中断されません。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度確立できません。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。

特定の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 70):  file :  filename
```

番号 70 は 1 つの例です。

回復中にファイルロックの再確立が失敗した場合、次のエラーメッセージが送信されます。

```
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すべてのアプリケーションを再起動する必要があります。次のことに注意してください。

- 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ります。
- ファイルを再度開いたり、または回復の失敗後にオープン操作を実行したその他のプロセスは、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、クライアントがほかのクライアントによるファイルアクセスの制御に使用するファイル共有モードがいくつかあります。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操作または作成操作の開始に失敗すると、 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。

プロトコルにはいくつかの共有モードがありますが、現在のところ、Oracle Solaris でのオープン操作では、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 モードだけを使用します。

また、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 では正しく実装できません。NFS version 4 クライアントで 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。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。読み込み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の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、読み込み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の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。

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

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します。どちらの場合も、競合が存在しているため、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、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の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します。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、version 2 と 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。例:

- NFS version 3 では、サーバーは `JUKEBOX` エラーを返します。これにより、クライアントはアクセス要求を停止し、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出力します。
- NFS version 2 では、`JUKEBOX` エラーと同等のエラーが存在しないため、サーバーは応答しません。これにより、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出力します。

これらの状態は、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、サーバー委託は有効になっています。`server_delegation` パラメータを none に設定すると、委託を無効にできます。手順については、44 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。

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 nfs4cbd 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 アドレスが指定されている場合、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。外部から見ることのできる動作だけになります。

コールバック情報は NFS version 4 要求に組み込まれているため、サーバーは、NAT (Network Address Translation) を使用するデバイスを通してクライアントと連絡を取ることができません。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、その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ーを高めます。ZFS ファイルシステムでは、ACL は、chmod コマンドを使用することで、サーバーおよびクライアント上で設定されます。UFS ファイルシステムの場合は、setfacl コマンドを使用します。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。

ID マッピングが失敗する理由

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「不明」と表示されます。

  たとえば、`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「不明」と表示されます。このコマンドの詳細は、`ls(1)` のマニュアルページを参照してください。
- 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、95 ページの「`nfsmapid` デーモン」を参照してください。

ACL を使用した ID マッピングの問題を回避する

ID マッピングの問題を回避するには、次の処置を行います。

- nfsmapid_domain の値が正しく設定されていることを確認します。
- ACL エントリ内のすべてのユーザーおよびグループ 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。

ACL エントリ内のすべてのユーザーおよびグループ 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。

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る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 – 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、『Solaris Dynamic Tracing Guide』の「Stability Levels」を参照してください。

ACL または nfsmapid の追加情報

次を参照してください。

- 『Oracle Solaris 11.1 の管理: ZFS ファイルシステム』の第 7 章「ACL および属性を使用した Oracle Solaris ZFS ファイルの保護」
- 95 ページの「nfsmapid デーモン」

UDP と TCP のネゴシエーション

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、クライアントと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のリストは、/etc/netconfig にあります。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_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、クライアントと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ーのサイズです。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 には転送サイズに上限はありませんが、クライアントは、必要であればマウント時にこれより小さい転送サイズを提示することができますが、ほとんどの場合その必要はありません。

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。このとき、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。PC クライアントの一部では転送サイズを小さくする必要があります。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 – 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。

ファイルシステムがどのようにマウントされるか

次の説明は、NFS version 3 のマウントに適用されます。NFS version 4 のマウントプロセスは、ポートマップサービスおよび MOUNT プロトコルを含みません。

クライアントがサーバーからファイルシステムをマウントするとき、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、クライアントはサーバーから /home/terry をマウントします。snoop によって追跡したトランザクションは、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=33492
client -> server MOUNT3 C Null
server -> client MOUNT3 R Null
client -> server MOUNT3 C Mount /export/home9/terry
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、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) を指定します。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ポート番号を受け取ると、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。

次の追跡結果では、クライアントは public オプションを使って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/terry
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 – 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、136 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 を組み合わせると、状況がさらに複雑になる可能性があります。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は、次のとおりです。

public オプションと NFS URL – 公開ファイルハンドルが使用されます。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が使用されます。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。公開ファイルハンドルを使用して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ます。

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機能

クライアント側のフェイルオーバー機能を使用すると、NFS クライアントは同じデータを利用できる複数のサーバーを知ることができるため、現在のサーバーが使

用不能になっても、ほかのサーバーに切り替えることができます。ファイルシステムが使用不能になる原因には次のものがあります。

- 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

通常、このような場合のフェイルオーバー機能はユーザーにはわかりません。つまり、フェイルオーバー機能はクライアント上のプロセスを中断することなく実行されます。

フェイルオーバー機能が行われるためには、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一になる条件については、148 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フェイルオーバー機能と CacheFS のどちらか片方の機能しか使用できません。

各ファイルシステムについて用意すべき複製の数を決める要素はさまざまです。理想的には、サーバーを 2 台以上設置します。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。フェイルオーバー処理の際にはリストにある各サーバーが確認されます。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。

フェイルオーバー機能に関する用語

フェイルオーバー機能のプロセスを完全に理解するには、次の 2 つの用語を理解する必要があります。

- フェイルオーバー – 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。通常、ソートされたリストの順番を元に、次のサーバーが応答するならばそのサーバーが使用されます。
- 再マッピング – 新しいサーバーを使用すること。クライアントは、正常な状態のときにリモートファイルシステム上のアクティブなファイルのそれぞれのパス名を格納します。再マッピング時には、そのパス名に基づいて新しいサーバー上のファイルを検出します。

複製されたファイルシステムとは

フェイルオーバー機能に関して、あるファイルシステムのすべてのファイルが元のファイルシステムのファイルとサイズもファイルタイプも同じ場合に、そのファイルシステムを「複製」といいます。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と再マッピングは失敗し、元のサーバーが再び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、動作が異なります。148 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` や `cpio` など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維持できます。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ときは、あらかじめ古いバージョンのファイル名を変更する
- クライアントがほとんど使用しない夜間に更新を実行する
- 更新は小規模にとどめる
- コピーの数を最小限にする

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。サーバー側で意識されないため、再マッピングされてもロックはそのまま残ります。ファイルはもともと変更が許されないので、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が確立されない場合、次のことが起こります。

- ファイルが使用不能とマークされる。
- 警告が出力される。
- アプリケーションがシステムコールの失敗を受け取る。

注 – 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、サイズが異なるディレクトリの複製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正常に行われない場合、次に一覧表示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、正常でない読み取り操作により、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停止しました。

NFS サーバーログ機能のしくみ

NFS サーバーログ機能は 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。このデータは情報へのアクセスを追跡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 データをバッファー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
- クライアントの 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
- 発生した処理のタイプ

`nfslogd` デーモンはこの 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。デー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ーブルに保存して、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` をオフにするとファイルハンドルパステーブルのマッピングが変更されなくなるため、デーモンは常に実行させておく必要があります。

注 – サーバーロギングは NFS version 4 ではサポートされません。

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決まっているため、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っ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。

あるファイルシステムに対するファイルハンドルを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知るには LOOKUP を実行します。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。公開ファイルハンドルからの相対パスに対して LOOKUP を実行する場合には、WebNFS サーバーはマルチコンポーネントルックアップによって 1 度にパス名全体を評価できます。マルチコンポーネントルックアップにより、WebNFS サーバーはパス名の中のディレクトリレベルを 1 つずつファイルハンドルに変換しなくても目的のファイルに対するファイルハンドルを配信できます。

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 アクティビティー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れば、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。パスの先頭にスラッシュを 1 つ追加すると、サーバーのルートファイルシステムからの相対位置に変更できます。次の例では、公開ファイルハンドルが /export/ftp ファイルシステムに設定されていればこの 2 つの NFS URL は同等です。

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

注 – 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、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ています。これは、WebNFS プロトコルの以前のバージョン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版です。

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。デフォルトの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。このメッセージは、そのデフォルトメカニズムが有効ではなく、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。その後、クライアントは、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。

注 – 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。

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 サイトで実現可能な機能のいくつかは、WebNFS ではサポートされていません。この違いは、NFS サーバーはファイルを送るだけであるため、特別な処理はすべてクライアントで行う必要があることが原因です。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。
- ブラウザからは、形式の異なるファイルを扱うために別のビューアが起動されることがあります。NFS URL からそうした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できるならば外部のビューアが起動されます。ブラウザは、NFS URL が使用されている場合、標準の MIME タイプで決まっているファイル名拡張子をすべて認識します。WebNFS ソフトウェアは、ファイルの内容からファイルタイプを判別しません。したがって、ファイルタイプはファイル名の拡張子だけから判別されます。
- NFS によるブラウズでは、サーバー側のイメージマップ (クリック可能なイメージ) は使用できません。ただし、クライアント側のイメージマップ (クリック可能なイメージ) は、場所とともに URL が定義されているため使用できます。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に、有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ー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、要求をしたユーザーではなくコンピュータを認証して、ファイル要求を認証します。そのため、クライアントユーザーは、su を実行し

てファイルの所有者を装ったりすることができます。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。

スーパーユーザーの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、ネットワークから取り出したりできます。ネットワークに対するもっとも危険な攻撃は、データをネットワークに持ち込むような攻撃です。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す。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を保護できます。

ネットワークのセキュリティー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リモート手続き呼び出し (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ーを大幅に向上させるとともに、NFS のセキュリティーを強化します。Secure RPC の機能を利用した NFS システムを Secure NFS システムといいます。

Secure RPC

Secure RPC は Secure NFS システムの基本となるメカニズムです。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 台のコンピュータを共有します。タイムシェアリングシステムはログインパスワードによりユーザーを認証します。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ーへの保証です。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 認証システムを理解するには、「資格 (credential)」と「ベリファイア」という 2 つの用語を理解する必要があります。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 の認証機能は拡張が可能で、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、ベリファイアには何も含まれません。ベリファイアが存在しないため、root 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。UNIX 認証でのもう 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。

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。このマップの設定方法については、『Oracle Solaris Administration: Naming and Directory Services』を参照してください。

DH 認証のセキュリティーは、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。このスキームが機能するには次の条件が必要です。

- 2 つのエージェントの現在時間が一致している。
- 送信側と受信側が同じ暗号化鍵を使用する。

ネットワークが時間同期プログラムを実行する場合、クライアントと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ていないと、サーバーはクライアントの要求を拒否します。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」を推理することによって、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。

KERB 認証

Kerberos は、マサチューセッツ工科大学 (MIT) で開発された認証システムです。Kerberos は、DES を含むさまざまな暗号化タイプを提供します。Kerberos 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 認証の実装についての詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の第 19 章「Kerberos サービスについて」を参照してください。

NFS での Secure RPC の使用

Secure RPC を使用する場合は、次の点に注意してください。

- サーバーがクラッシュしたとき周囲にだれもいない場合 (停電のあとなど) には、システムに格納されていた秘密鍵はすべて削除されます。そのため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ーが不可欠です。
- 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に対する秘密鍵の入力を記録するような、不正なカーネルをだれかがブートすることが考えられます。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。このような制限は重大な問題につながる可能性がありますが、この部分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許したことになります。これは、秘密鍵が相手側のコンピュータの鍵サーバーに渡され、鍵サーバーがその秘密鍵を格納したためです。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共有されていて -o sec=dh 指定されていると、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。パスワードを要求され、入力した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 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。ミラーマウントにより、NFSv4 クライアントは NFSv4 サーバーのファイルシステムが共有されるとすぐに、そのファイルシステム内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、NFSv4 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。

どのような場合にミラーマウントを使用するか

一般に、次の場合を除き、NFSv4 クライアントでミラーマウント機能を使用すると最適です。

- 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必要がある
- 親ファイルシステムのマウントオプションとは異なるマウントオプションを使用する必要がある

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 NFSv4 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たほかのファイルシステムがあれば、それら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じ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。何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 NFSv4 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に自動マウントトリガーポイントが設定されている場合は、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、サーバーの名前空間 `/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。

ミラーマウントを機能させるための具体的な方法については、次を参照してください。

- 例 2–2
- 38 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」

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、アイドル状態で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。この時間は `timeout` パラメータを使用して設定されます。これは、オートマウンタによって同じ目的に使用されるパラメータです。

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アクティブなファイルシステムが含まれている場合は、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達されます。

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。オートマウンタ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ません。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 には、NFS リフェラルと呼ばれる新しい NFS 機能が含まれています。NFS リフェラルは、複数の NFSv4 サーバーを均一な名前空間に接続する手段として、NFSv4 サーバーがほかの NFSv4 サーバー上にあるファイルシステムを指す方法です。

NFSv2、NFSv3、およびその他のクライアントにはリフェラルがシンボリックリンクのように見えるため、クライアントはリフェラルをたどることができます。

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、これを実行するために autofs を使用しない場合に役立ちます。NFSv4 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。リフェラルの作成時にマウントポイントがまだ存在していない場合は、オブジェクトを再解析ポイントとして識別する特別なフラグを含むシンボリックリンクが生成されます。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 サービスデータを置き換えます。

NFS リフェラルの削除

NFS リフェラルの削除にも `nfsref` コマンドを使用します。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、ほかの種類のサービスデータが存在しない場合は再解析ポイントを削除します。

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、ディレクトリからマップへの関連付けを行います。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。auto_master ファイルの内容の例を次に示します。

例 3–3 /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 *mount-point* は、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 *map-name* は、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、160 ページの「/net マウントポイント」を参照してください。

mount-options *mount-options* は省略できます。map-name のエントリにほかのオプションがある場合を除き、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をコンマで区切って並べます。特定のファイルシステムのマウントオプションについては、各ファイルシステムについてのマニュアルページを参照してください。たとえば、NFS に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行です。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、行末にバックスラッシュ (\) を入力します。入力できる文字数の上限は 1024 です。

注 – 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) に記述されたエントリがマウントされるディレクトリです。

注 – 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。このデフォルトは、NIS の auto.master マップのエントリを使用して、またはローカルの /etc/auto_master ファイルを編集することによってオーバーライドできます。

/net マウントポイント

autofs は、特別なマップ -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。これは hosts データベースだけを使用する組み込みマップです。たとえば、hosts データベースにあるコンピュータ gumbo が、ファイルシステムのどれか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
```

なお、autofs はホスト gumbo のエクスポートされたファイルシステムだけをマウントできます。つまり、ローカルディスク上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ファイルシステムです。したがって、gumbo にあるすべてのファイルとディレクトリは、/net/gumbo では利用できない場合があります。

/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。このような場合は /net を使用しないで、そのファイルシステムに対応するエントリをマップの中に設定します。

注 – NFSv3 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。NFSv4 を使用するシステムでは、ミラーマウントによって、サーバー上のエクスポートされたファイルシステムのリストに加えられた動的な変更が反映されます。

/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フェデレーテッドファイルシステムのドメインルートをマウントします。/nfs4/example.net を参照すると、DNS ドメイン example.net のドメインルートの検索と、その場所でのマウントが試みられます。これには、40 ページの「FedFS サーバーの 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。

autofs 直接マップ

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。次に一般的な /etc/auto_direct マップを示します。

```
/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
```

直接マップの行は、次の構文に従っています。

key [*mount-options*] *location*

key *key* は直接マップでのマウントポイントのパス名です。

mount-options *mount-options* は、このマウントに適用するオプションです。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。特定のファイルシステムのマウントオプションについては、各ファイルシステムについてのマニュアルページを参照してください。たとえば、NFS に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。

location *location* はファイルシステムの位置を示します。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 *server*:*pathname* として指定されます。

注 – *pathname* に自動マウントされたマウントポイントを含めることはできません。*pathname* は、ファイルシステム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* としてリストする必要があります。

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、次のようになっているとします。

```
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 – オートマウンタマップの間では、オプションの連結はされません。オートマウンタ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。

この種類のマップについては、これ以外にも重要な機能があります。169 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 (複数ロケーション)」を参照してください。

/– マウントポイント

例 3–3 では、マウントポイント /- は、auto_direct の中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。間接マップの場合は、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あることに注意してください。

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。

autofs 間接マップ

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を対応させます。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一般的な構文になります。

key [*mount-options*] *location*

key	*key* は間接マップでの単純名 (スラッシュなし) です。
mount-options	*mount-options* は、このマウントに適用するオプションです。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。特定のファイルシステムのマウントオプションについては、各ファイルシステムについてのマニュアルページを参照してください。たとえば、NFS に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。
location	*location* はファイルシステムの位置を示します。1 つまたは複数のファイルシステムを *server*: *pathname* で指定します。

> 注 – *pathname* に自動マウントされたマウントポイントを含めることはできません。*pathname* は、ファイルシステム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*:/net/*server*/usr/local ではなく、*server* :/usr/local として指定する必要があります。

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 3–3 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 のもとで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。

```
david                   willow:/export/home/david
rob                     cypress:/export/home/rob
gordon                  poplar:/export/home/gordon
rajan                   pine:/export/home/rajan
tammy                   apple:/export/home/tammy
jim                     ivy:/export/home/jim
linda    -rw,nosuid     peach:/export/home/linda
```

例として、前のマップがホスト oak 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 linda のホームディレクトリが /home/linda であることを示すエントリがあるとします。linda がコンピュータ oak にログインするたびに、autofs は、コンピュータ peach にあるディレクトリ /export/home/linda 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。

次のような状況が発生したと想定してください。 ユーザー linda のホームディレクトリがパスワードデータベースに、/home/linda として表示されます。Linda も含めだれでも、前の例のマップを参照するマスターマップで設定されたどのコンピュータからでも、このパスにアクセスできます。

こうした状況のもとでは、ユーザー linda はこれらのどのコンピュータでも login や rlogin を実行し、代わりに彼女用のホームディレクトリをマウントさせることができます。

さらに、これで linda は次のコマンドも入力できます。

```
% cd ~david
```

autofs は彼女のために David 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で許可されている場合)。

注 – オートマウンタマップの間では、オプションの連結はされません。オートマウンタ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 のしくみ

autofs は、自動的に適切なファイルシステムをマウントするためのクライアント側のサービスです。自動マウントを行うのに、次のコンポーネントが相互に動作します。

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

自動マウントサービス svc:/system/filesystem/autofs は、システムの起動時に呼び出され、マスターマップファイル auto_master を読み取って、autofs マウントの最初のセットを作成します。これらの autofs のマウントは起動時に自動的にはマウントされません。後でファイルシステムがマウントされるポイントです。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。

autofs マウントが設定されると、要求があったときにファイルシステムをマウントすることができます。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。

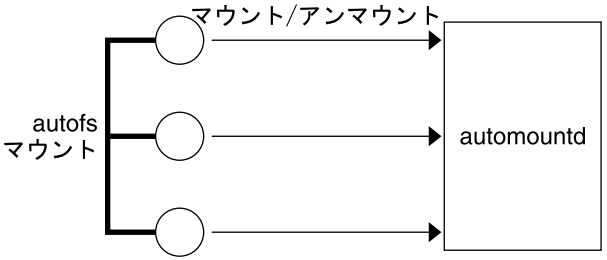
最初に autofs マウントをマウントしたあとは、必要に応じて automount コマンドを実行し、autofs マウントを更新します。このコマンドは、auto_master マップにあるマウントのリストと、マウントテーブルファイル /etc/mnttab (前のバージョンでは /etc/mtab) にあるマウントされたファイルシステムのリストを比較します。その後、automount によって、適切な変更が加えられます。このプロセスにより、システム管理者は auto_master 内のマウント情報を変更し、autofs デーモンを停止したり再起動したりすることなく、それらの変更結果を autofs 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を調べるために /etc/vfstab ファイル (各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。

次の図では、autofs のしくみの概要を簡単に説明します。

自動マウントデーモンである automountd は、ブート時にサービス svc:/system/filesystem/autofs によって起動されます。図 3–3 を参照してください。このサービスは automount コマンドも実行します。このコマンドはマスターマップを読み取り、autofs のマウントポイントをインストールします。詳細は、167 ページの「autofs のナビゲーションプロセス開始法 (マスターマップ)」を参照してください。

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



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。

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 にメッセージを送信します。
3. automountd がマップからファイルシステム情報を見つけ、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。

注 – 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消去されます。

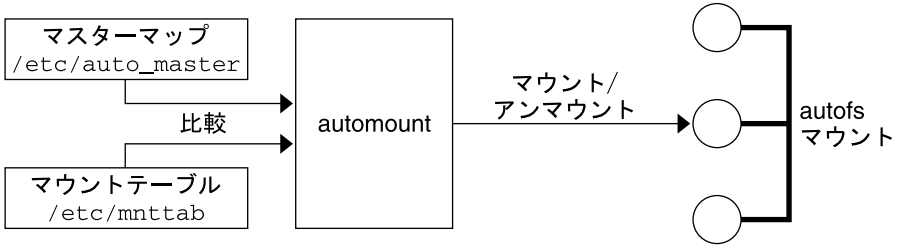
autofs のネットワークナビゲート (マップ)

autofs は一連のマップを探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 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。176 ページの「autofs のネットワークナビゲート法の変更 (マップの変更)」を参照してください。

autofs のナビゲーションプロセス開始法 (マスターマップ)

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。図 3–4 に示すように、マスターマップ内の各エントリは、直接または間接のマップ名、そのパス、およびそのマウントオプションです。エントリの順序は重要ではありません。automount は、マスターマップ内のエントリとマウントテーブル内のエントリを比較して、現在のリストを生成します。

図 3–4　マスターマップによるナビゲーション



autofs マウントプロセス

マウント要求が発生したときに autofs サービスが何を実行するかは、オートマウンタ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 サービスは次の手順を実行します。

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、マウントプロセスはさらに複雑になります。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
```

この場合、/share/ws マウントポイントがアクセスされたときのマウントプロセスは基本的に最初の例と同じです。また、/share/ws ファイルシステムの中に次のレベル (/usr) へのトリガーノードを作成することにより、そのレベルがアクセスされたときにマウントできるようにします。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。この制限についての説明は、169 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

一定時間アクセスがないためにアンマウントされるときは、マウントと逆の順序で実行されます。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。-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 (複数ロケーション)

次は、直接マップの例です。

```
/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 には複数の場所があり、/usr/man のマウントポイントは 3 つ、/usr/spool/news のマウントポイントは 2 つの場所が記述されています。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くなるので、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にだけ意味があります。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、もっとも利用しやすいサーバーが、そのユーザーの手をまったく必要としないで自動的にマウントされるということです。

ファイルシステムを複製として構成してあると (148 ページの「複製されたファイルシステムとは」を参照)、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しても、そのサーバーが動作しており、しかもそのファイルシステムをエクスポートしているかぎり、問題ありません。上の例では、複数のマウント位置は、マップエントリ内のマウント位置のリストになっています。

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

この例では、サーバー oak、rose、willow のどれからでも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。

- 特定レベルの NFS プロトコルをサポートしているサーバーの数
- サーバーとの距離
- 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。

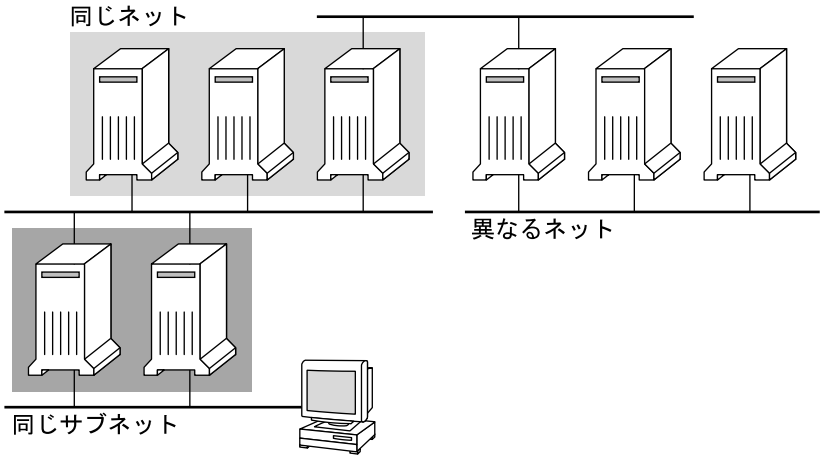
プロトコルが同じバージョンのサーバーの組の中で数がもっとも多いものがわかると、サーバーのリストが距離によってソートされます。距離を判定するために、IPv4 アドレスが調査されます。IPv4 アドレスは、どのサーバーが各サブネットにあるかを示します。ローカルサブネット上のサーバーには、リモートサブネット

上のサーバーよりも高い優先順位が付けられます。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 – 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 3–5 に、サーバーとの距離を示します。

図 3–5 サーバーとの距離



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。優先順位には、重み付けも関係します (173 ページの「autofs と重み付け」を参照してください)。

たとえば、version 4 サーバーの方が多いと、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、優先順位の決定は複雑になります。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に version 3 サーバーがあり、もっとも近い version 4 サーバーがリモートサブネット上にあると、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 version 2 サーバーで構成されていると、version 3 と version 4 サーバーを使用するリモートサブネットよりも優先されます。
- ローカルサブネットがさまざまな数の version 2、version 3、および version 4 サーバーで構成されていると、さらに優先順位付けが必要になります。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、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に、version 4 サーバーよりも version 3 または version 2 サーバーの方が多い場合、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に、version 4 サーバーが 3 台、version 3 サーバーが 3 台、version 2 サーバーが 10 台ある場合、version 3 サーバーが選択されます。
- 同じように、ローカルサブネットがさまざまな数の version 2 と version 3 サーバーで構成されていると、最初にオートマウンタは、どのバージョンがローカルサブネットでもっとも高いバージョンかを見つけます。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い場合、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 version 2 サーバーの方が version 3 サーバーよりも多い場合、version 2 サーバーが選択されます。

注 – 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。特に、`server_versmin`、`client_versmin`、`server_versmax`、および `client_versmax` の値により、いくつかのバージョンを優先順位の決定から除外できます。これらのパラメータの詳細は、93 ページの「mountd デーモン」および 94 ページの「nfsd 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と、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。

多くのサブネットを持つ大規模ネットワークでは、フェイルオーバーは特に便利です。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 – 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。例:

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

括弧内の数値が重み付けを表します。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 – 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

変数名の前にドル記号 ($) 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。この変数は、同じファイルシステムの位置にアクセスする異なるアーキテクチャータイプの調整に役立ちます。変数名を括弧でくくることで、その後に続く文字や数字と変数とを区切ることができます。表 3–3 に定義済みのマップ変数を示します。

表 3–3 定義済みのマップ変数

変数	意味	提供元	例
ARCH	アーキテクチャータイプ	uname -m	sun4

表 3–3 定義済みのマップ変数 (続き)

変数	意味	提供元	例
CPU	プロセッサタイプ	uname -p	sparc
HOST	ホスト名	uname -n	dinky
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のリリース	uname -r	5.8
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 および /usr/local/bin/x86 から、SPARC アーキテクチャーと 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 – 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

ファイルマップで使用されたマップエントリ *+mapname* により、automount は指定されたマップを、あたかも現在のマップ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 プロパティーグループの automount プロパティーは、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 プロパティーが指定されていない場合は、config/default プロパティーで定義された順序が使用されます。例:

```
# 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
```

この例では、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ます。config/automount プロパティー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いくつか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
```

組み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 + エントリを持つことができることに注意してください。

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

注 – 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 マップを作成することもできます。データベースやフラットファイルから autofs 構造を作成しなければならない場合は、実行可能な autofs マップが有効なことがあります。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 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
```

実行可能マップの例を示します。

```
#!/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、実行可能ビットがオンになっ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
```

autofs のネットワークナビゲート法の変更 (マップの変更)

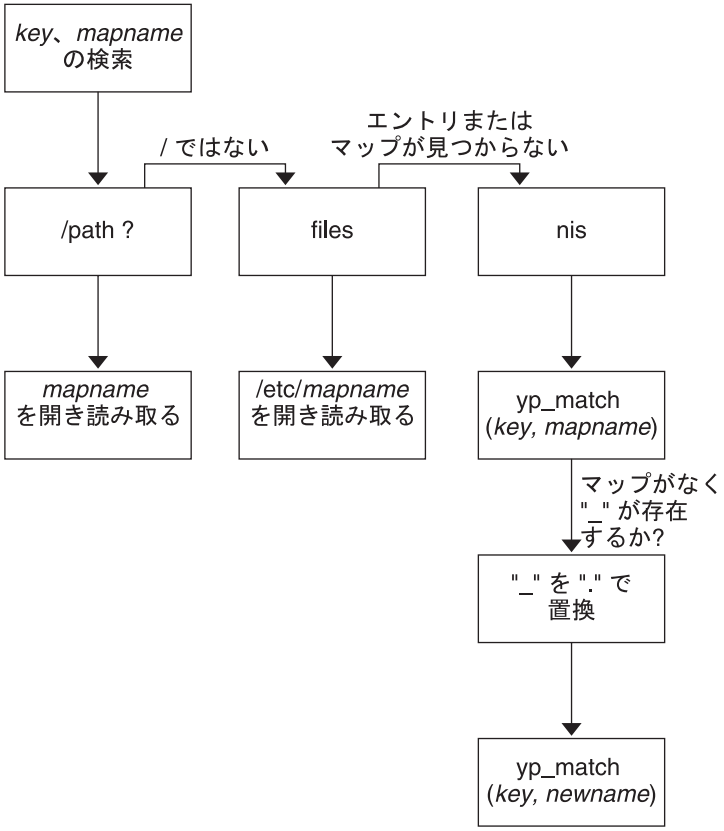
マップへのエントリを変更、削除、または追加して、ユーザーの環境ニーズに合わせることができます。ユーザーが必要とするアプリケーションやその他のファイルシステムがその位置を変更すると、マップはこれらの変更を反映しなければなりません。autofs のマップは、いつでも変更できます。automountd が次にファイルシステムをマウントしたときにその変更内容が有効になるかどうかは、変更したマップと変更内容によって決まります。

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 は、サービス svc:/system/filesystem/autofs によって起動され、マスターマップ auto_master を確認します。次に説明する規則が適用されます。

autofs は、svc:/system/name-service/switch サービスの config/automount プロパティーで指定されたネームサービス順序を使用します。config/automount が定義されていない場合は、config/default プロパティーが使用されます。NIS を選択し、autofs が必要なマップを検出できない場合で、1 つまたは複数のアンダースコアを含むマップ名を検出したときには、それらのアンダースコアがドットに変更されます。こうすることにより、NIS の古いファイル名を利用することができます。図 3–6 に示されているように、次に autofs はもう一度マップを調べます。

図 3–6　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

これ以降のセクションでは、autofs の高度な機能を取り上げます。

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。置き換えに使用する文字や、autofs のマップ構文解析機能から他の文字を保護するために使用する文字もあります。

アンパサンド (&)

たとえば、次のように、多数のサブディレクトリを指定したマップがある場合は、文字列置換を使用できます。

```
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鍵を置換することができます。アンパサンド文字を使用すると、前述のマップは次のようになります。

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

鍵置換はまた、次のような直接マップでも使用できます。

```
/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、次のように指定することはできません。

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

これは、autofs が、この例を次のように解釈するためです。

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

アスタリスク (*)

任意の鍵を一致させるのに、任意の文字を表す置換文字であるアスタリスク (*) を使用できます。このマップエントリを使用して、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

特殊文字が含まれているマップエントリがある場合に、autofs のマップ構文解析機能を混乱させる名前のディレクトリをマウントする必要があるかもしれません。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。これらの名前は二重引用符で囲んでください。

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

索引

B

C

D

E

F

N

O

P

R

S

T

U

V

W

み