Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)

Part No: 819–0382–15
2010 年 9 月

ORACLE

100908@24661

目次

例目次

はじめに

『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』では、Solaris オペレーティングシステム (Solaris OS) のネームサービスおよびディレクトリサービスである DNS、NIS、LDAP の設定および管理について説明します。このガイドは、現行の Solaris リリースのシステム管理およびネットワーク管理のセットの一部です。

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

本書の x86 に関連する用語については、以下を参照してください。

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

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

対象読者

このガイドは、経験豊富なシステム管理者およびネットワーク管理者を対象としています。

このマニュアルは、Solaris のネームサービスおよびディレクトリサービスに関連したネットワークの概念を紹介するものであり、Solaris OS のネットワークの基礎や管理ツールについては説明しません。

内容の紹介

このガイドは、各ネームサービスに対応して複数の部分に分けられています。

パート I「ネームサービスとディレクトリサービスについて」

パート II「DNS の設定と管理」

パート III「NIS の設定と管理」

パート IV「LDAP ネームサービスの設定と管理」

Solaris システム管理マニュアルセットの構成

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

マニュアルのタイトル	トピック
『Solaris のシステム管理 (基本編)』	ユーザーアカウントとグループ、サーバーとクライアントのサポート、システムのシャットダウンとブート、およびサービスの管理
『Solaris のシステム管理 (上級編)』	端末とモデムの設定、システムリソースの管理 (ディスク割り当て、アカウンティング、および crontab ファイルの管理)、システムプロセスの管理、および Oracle Solaris ソフトウェアの障害追跡
『Solaris のシステム管理 (デバイスとファイルシステム)』	リムーバブルメディア、ディスクとデバイス、ファイルシステム、およびデータのバックアップと復元
『Solaris のシステム管理 (IP サービス)』	TCP/IP ネットワーク管理、IPv4 と IPv6 アドレス管理、DHCP、IPsec、IKE、Solaris IP フィルタ、モバイル IP、IP ネットワークマルチのパス化 (IPMP)、および IPQoS
『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』	DNS、NIS、および LDAP のネーミングとディレクトリサービス (NIS から LDAP への移行、および NIS+ から LDAP への移行を含む)
『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』	NIS+ のネーミングとディレクトリサービス
『Solaris のシステム管理 (ネットワークサービス)』	Web キャッシュサーバー、時間関連サービス、ネットワークファイルシステム (NFS と Autofs)、メール、SLP、および PPP
『Solaris のシステム管理 (印刷)』	印刷に関するトピックや、サービス、ツール、プロトコル、およびテクノロジを使って印刷サービスおよびプリンタを設定および管理する方法

マニュアルのタイトル	トピック
『Solaris のシステム管理 (セキュリティサービス)』	監査、デバイス管理、ファイルセキュリティー、BART、Kerberos サービス、PAM、Solaris 暗号化フレームワーク、特権、RBAC、SASL、および Solaris Secure Shell
『Oracle Solaris のシステム管理 (Oracle Solaris コンテナ : 資源管理と Oracle Solaris ゾーン)』	リソース管理に関連する計画と作業、拡張アカウンティング、リソース制御、フェアシェアスケジューラ (FSS)、資源上限デーモン (rcapd) による物理メモリーの制御、および資源プール (Solaris Zones ソフトウェア区分技術と lx ブランドゾーンによる仮想化)
『Oracle Solaris ZFS 管理ガイド』	ZFS ストレージプールおよびファイルシステムの作成と管理、スナップショット、クローン、バックアップ、アクセス制御リスト (ACL) による ZFS ファイルの保護、ゾーンがインストールされた Solaris システム上での ZFS の使用、エミュレートされたボリューム、およびトラブルシューティングとデータ回復
『Oracle Solaris Trusted Extensions 管理の手順』	Oracle Solaris Trusted Extensions 機能固有のシステム管理
『Oracle Solaris Trusted Extensions 構成ガイド』	Solaris 10 5/08 リリース以降での、Oracle Solaris Trusted Extensions 機能の計画、有効化、および初期設定の方法

関連情報

- 『Sun Java System Directory Server 配備ガイド』は Sun Java Enterprise System の文書に収められています。
- 『Sun Java System Directory Server 管理ガイド』は Sun Java Enterprise System の文書に収められています。
- 『DNS & BIND』Paul Albitz/Cricket Liu 著 (高田広章/小島育夫監訳、小舘光正訳、オライリー・ジャパン、2002 年)
- 『Understanding and Deploying LDAP Directory Services』Timothy A. Howes, Ph.D、Mark C. Smith 共著

マニュアル、サポート、およびトレーニング

追加リソースについては、次の Web サイトを参照してください。

- マニュアル (`http://docs.sun.com`)
- サポート (`http://www.oracle.com/us/support/systems/index.html`)
- トレーニング (`http://education.oracle.com`) – 左のナビゲーションバーで「Sun」のリンクをクリックします。

Oracle へのご意見

Oracle はドキュメントの品質向上のために、お客様のご意見やご提案をお待ちしています。誤りを見つけたり、改善に向けた提案などがある場合は、http://docs.sun.com で「Feedback」をクリックしてください。可能な場合には、ドキュメントのタイトルやパート番号に加えて、章、節、およびページ番号を含めてください。返信を希望するかどうかもお知らせください。

Oracle Technology Network (http://www.oracle.com/technetwork/index.html) では、Oracle ソフトウェアに関する広範なリソースが提供されています。

- ディスカッションフォーラム (http://forums.oracle.com) で技術的な問題や解決策を話し合う。
- Oracle By Example (http://www.oracle.com/technology/obe/start/index.html) のチュートリアルで、手順に従って操作を体験する。
- サンプルコード (http://www.oracle.com/technology/sample_code/index.html) をダウンロードする。

表記上の規則

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

表 P–1　表記上の規則

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

表 P–1 表記上の規則 (続き)

字体または記号	意味	例
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	`sun% grep '^#define \` `XV_VERSION_STRING'`

Oracle Solaris OS に含まれるシェルで使用する、UNIX のデフォルト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。コマンド例に示されるデフォルトのシステムプロンプトは、Oracle Solaris のリリースによって異なります。

- C シェル

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

- C シェルのスーパーユーザー

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

- Bash シェル、Korn シェル、および Bourne シェル

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

- Bash シェル、Korn シェル、および Bourne シェルのスーパーユーザー

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

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

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

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

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

パート I

ネームサービスとディレクトリサービスについて

ここでは、Solaris OS のネーミングサービスとディレクトリサービスの概要について説明します。また、異なるサービスと組み合わせて使用する際に利用する `nsswitch.conf` ファイルについても説明します。

◆ ◆ ◆ 第 1 章

1 ネームサービスとディレクトリサービス (概要)

この章では、Solaris で使用されるネームサービスとディレクトリサービスの概要について説明します。また、DNS、NIS、および LDAP ネームサービスについても簡潔に説明します。NIS+ の詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』を参照してください。

ネームサービスとは

「ネームサービス」は、ユーザー、マシン、およびアプリケーションがネットワーク経由で通信するための情報を集中管理することを可能にします。格納される情報には、次のものが含まれます。

- マシン (ホスト) 名とアドレス
- ユーザ名
- パスワード
- アクセス権
- グループのメンバーシップ、プリンタなど

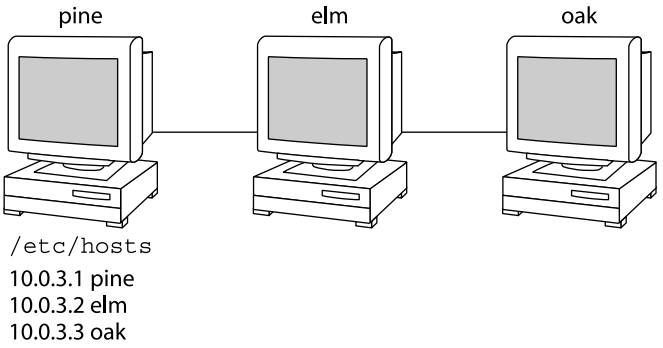
集中化されたネームサービスが存在しない場合、マシンごとに、これらの情報のコピーを管理する必要があります。ネームサービス情報はファイルまたはマップ、データベーステーブルの形で格納できます。すべてのデータを 1 カ所で管理すれば、管理がより簡単になります。

ネームサービスは、どのようなコンピュータネットワークにも欠かせないものです。ネームサービスは、ほかの機能に加え、次の機能を提供します。

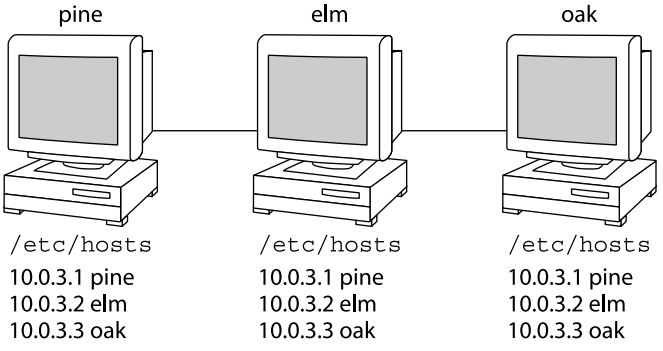
- 名前とオブジェクトを対応付ける (「バインド」する)
- オブジェクトの名前を解決する
- バインドを解除する
- 名前を一覧表示する
- 名前を変更する

ネットワーク情報サービスを使用すると、数値アドレスの代わりに一般的な名前でマシンを識別できます。これにより、ユーザーは 192.168.0.0 のような扱いにくい数値アドレスを記憶して入力する必要がなくなるため、通信が簡単になります。

たとえば、 pine、elm、oak という 3 台のマシンで構成されるネットワークを考えてみましょう。pine が elm または oak にメッセージを送信するには、pine はそれら 2 台のネットワークアドレスを知る必要があります。そのため pine はそれ自体のものを含めたネットワーク内のすべてのマシンのネットワークアドレスを格納する /etc/hosts ファイルまたは /etc/inet/ipnodes ファイルを保持しています。



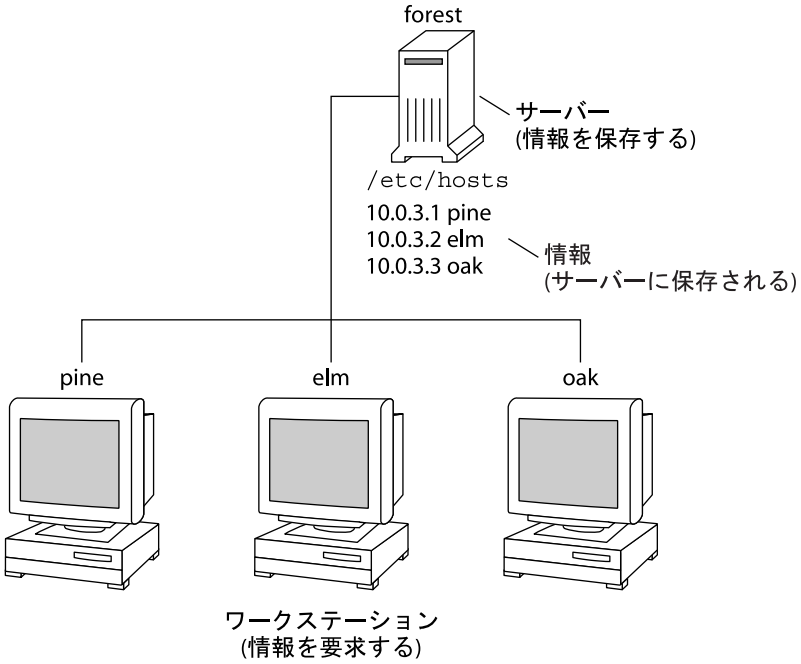
同様に、elm や oak が pine と通信したり、お互いに通信するためには、上記のようなファイルを保持している必要があります。



マシンには、アドレスに加え、セキュリティー情報、メールデータ、ネットワークサービスについての情報なども格納されます。ネットワークによって提供されるサービスが増えるにつれて、格納する情報の種類も増えていきます。その結果、各マシンで /etc/hosts や /etc/inet/ipnodes のようなファイルのセット全部を保持する必要がでてくる可能性があります。

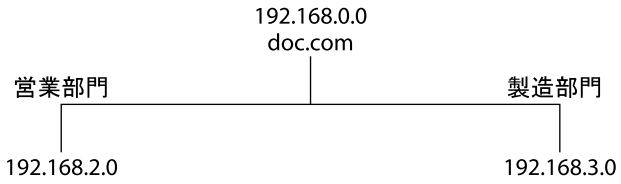
ネットワーク情報サービスは、サーバー上にネットワーク情報を格納し、照会を実行するマシンに情報を提供します。

照会を実行するマシンは、サーバーの「クライアント」と呼ばれます。次の図に、クライアントとサーバーの関係を示します。ネットワークについての情報が変更されるたびに、各クライアントのローカルファイルを変更する代わりに、管理者はネットワーク情報サービスが格納する情報だけを更新します。これによって、エラー、クライアント間の不一致、そして作業量を減らすことができます。

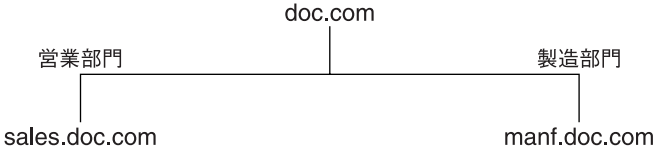


このように、サーバーがネットワークを通してサービスをまとめてクライアントに提供する方法を「クライアントサーバーコンピューティング」と呼びます。

ネットワーク情報サービスの第一の目的は情報の一元管理ですが、もう 1 つの目的はネットワーク名の簡素化です。たとえば、ある会社がネットワークを設定して、インターネットに接続したと仮定します。会社のネットワークには、インターネットのネットワーク番号 `192.168.0.0` とドメイン名 `doc.com` が割り当てられました。会社には「営業 (Sales)」と「製造 (Manf)」という 2 つの部門があるため、このネットワークは 1 つのメインネットと、各部門に 1 つのサブネットに分割されます。各ネットには独自のアドレスがあります。



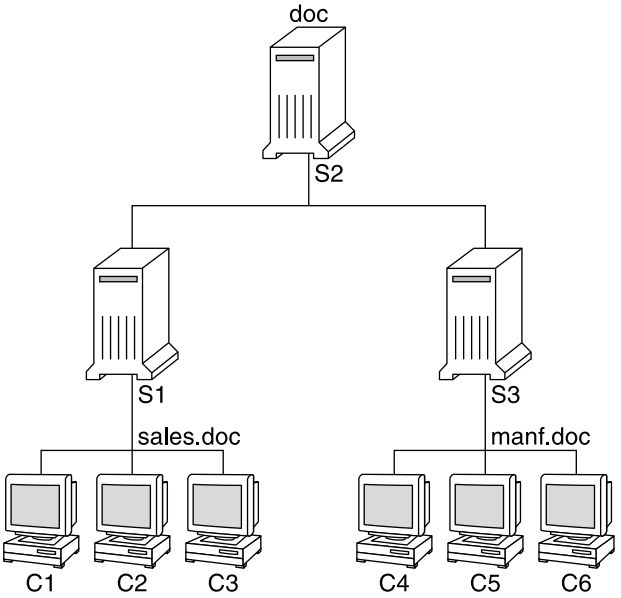
上に示すように、各部はネットワークアドレスで識別することもできますが、ネームサービスによって使用可能となる説明的な名前の方が便利です。



メールやその他のネットワーク通信の送信先は、 198.168.0.0 というアドレスで指定する代わりに、単に doc と指定できます。また、メールの送信先を 192.168.2.0 や 192.168.3.0 と指定する代わりに、sales.doc や manf.doc と指定できます。

名前はまた、物理アドレスよりもはるかに柔軟です。物理的なネットワークはめったに変更されませんが、企業の組織はよく変化します。

たとえば、doc.com ネットワークが、S1、S2、S3 の 3 台のサーバーによってサポートされる場合を考えましょう。そのうち 2 台のサーバー (S1 とS3) がクライアントをサポートしているとします。

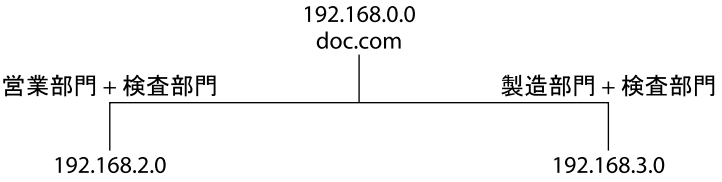


クライアント C1、C2、および C3 はネットワーク情報をサーバー S1 から入手します。クライアント C4、C5、および C6 は、サーバー S3 から情報を入手します。結果として構成されるネットワークの概要を、次の表に示します。表は、前記のネットワークを一般化して表現したもので、実際のネットワーク情報マップとは異なります。

表 1–1　docs.com ネットワークの構成

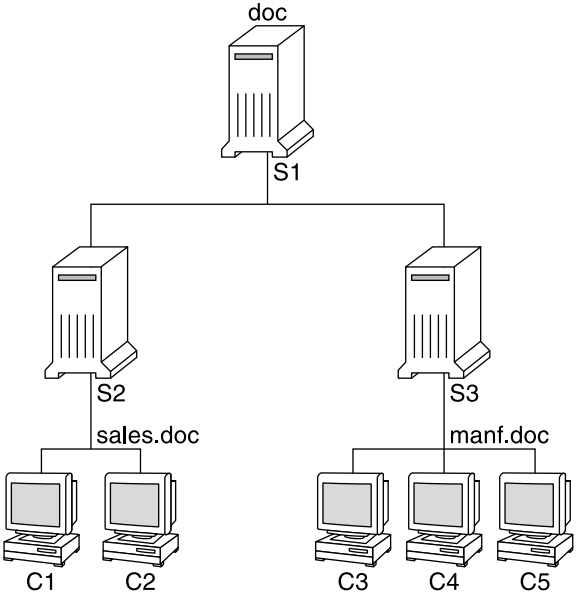
ネットワークアドレス	ネットワーク名	サーバー	クライアント
192.168.1.0	doc	S1	
192.168.2.0	sales.doc	S2	C1、C2、C3
192.168.3.0	manf.doc	S3	C4、C5、C6

2 つの部門からある人数の人材を借りて第 3 の検査部門を新設し、第 3 のサブネットは開設しなかったとします。その結果、物理ネットワークは、企業の組織とは対応しなくなります。



検査部門のトラフィックには専用のサブネットがなく、192.168.2.0 と 192.168.3.0 に分割されます。ここで、ネットワーク情報サービスを使用することにより、検査部門のトラフィックにも専用のネットワークを備えることができます。

このように、組織が変更された場合、そのネットワーク情報サービスでは次に示すようにマッピングを変更できます。



この変更の結果、クライアント C1 と C2 は、サーバー S2 から情報を入手します。クライアント C3、C4、および C5 は、サーバー S3 から情報を入手します。

組織内でそのあとに行われる変更に対しては、ハードウェアのネットワーク構造を再編成することなく、ネットワーク情報構造を変更することにより対応できます。

Solaris のネームサービス

Solaris プラットフォームは、次のネームサービスを提供します。

- DNS (ドメインネームシステム、Domain Name System) - 29 ページの「DNS ネームサービスの説明」を参照してください
- /etc ファイル - 初期の UNIX ネームシステム。30 ページの「/etc ファイルネームサービスの説明」を参照してください
- NIS (ネットワーク情報サービス、Network Information Service) - 30 ページの「NIS ネームサービスの説明」を参照してください
- NIS+ (ネットワーク情報サービスプラス、Network Information Service Plus) - 『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』を参照してください
- LDAP (Lightweight Directory Access Protocol) - パート IV「LDAP ネームサービスの設定と管理」の「LDAP ネームサービスの設定と管理」を参照してください

最近のほとんどのネットワークでは、これらのサービスのうちの 2 つ、またはそれ以上を組み合わせて使用します。複数のサービスを使用するときは、nsswitch.conf ファイルで調整します。このファイルについては、第 2 章「ネームサービススイッチ (概要)」で説明します。

DNS ネームサービスの説明

DNS は TCP/IP ネットワーク用にインターネットが提供するネームサービスです。DNS はネットワーク上のマシンがインターネットアドレスではなく、普通の名前で識別できるように開発されたものです。DNS は、ローカルの管理ドメイン内と、複数の管理ドメイン間においてホスト名の管理を行います。

DNS を使用する、ネットワークに接続されたマシンの集合のことを「DNS 名前空間」と呼びます。DNS 名前空間は階層をなす複数の「ドメイン」に分けることができます。DNS ドメインは、複数のマシンからなるグループです。各ドメインは複数の「ネームサーバー」、つまり、1 つの主サーバーと 1 つまたは複数の副サーバーによってサポートされます。各サーバーは in.named デーモンを実行することによって DNS を実装しています。クライアント側は、「リゾルバ」によって DNS を実装します。リゾルバの機能はユーザーの照会を解決することです。リゾルバがネームサーバーに照会すると、ネームサーバーは要求された情報か、またはほかのサーバーに照会する旨を返します。

/etc ファイルネームサービスの説明

ホストを基本とした初期の UNIX のネームシステムは、スタンドアロンの UNIX マシン用に開発されたあと、ネットワークで使用されるようになりました。UNIX オペレーティングシステムの旧版や UNIX マシンの多くは、現在でもこのシステムを使用していますが、大規模で複雑なネットワークにはあまり適切ではありません。

NIS ネームサービスの説明

ネットワーク情報サービス (NIS) は、DNS とは独立して開発されました。DNS が数値 IP アドレスの代わりにマシン名を使うことによって、通信を簡略化することに焦点を当てているのに対して、NIS は、多様なネットワーク情報を集中管理することによりネットワーク管理機能を高めることに焦点を当てています。NIS には、ネットワーク、マシンの名前とアドレス、ユーザー、およびネットワークサービスに関する情報も格納されます。このようなネットワーク情報の集まりを「NIS の名前空間」と呼びます。

NIS 名前空間情報は NIS マップに格納されています。NIS マップは、UNIX の `/etc` ファイルおよびほかの構成ファイルを置換するように設計されているので、名前やアドレスよりはるかに多くの情報を保存できます。その結果、NIS 名前空間には非常に大きなマップの集合が含まれることになります。詳細については、104 ページの「NIS マップに関する作業」を参照してください。

NIS は DNS に似たクライアントサーバーの配列を持っています。複製の NIS サーバーは NIS クライアントへサービスを提供します。主サーバーは「マスター」サーバーと呼ばれ、信頼性を保証するためにバックアップつまり「スレーブ」サーバーを持っています。どちらのサーバーも NIS 検索ソフトウェアを使用し、NIS マップを格納します。NIS アーキテクチャーおよび NIS の管理方法の詳細については、第 5 章「NIS サービスの設定と構成」および第 6 章「NIS の管理 (手順)」を参照してください。

NIS+ ネームサービスの説明

「ネットワーク情報サービスプラス」(NIS+) は、NIS によく似たネットワークネームサービスですが、より多くの機能を備えています。ただし、NIS+ は NIS の拡張機能ではありません。

NIS+ ネームサービスは、組織の形態に適合するように設計されています。NIS とは異なり、NIS+ の名前空間は動的な構成で、正規ユーザーであればいつでも更新できます。

NIS+ を使用すると、マシンのアドレス、セキュリティー情報、メール情報、Ethernet インタフェース、ネットワークサービスなどの情報を 1 カ所に格納できます。このように構成されたネットワーク情報を、NIS+「名前空間」と呼びます。

NIS+ 名前空間は階層構造となっていて、UNIX のディレクトリファイルシステムによく似ています。階層構造になっていることから、NIS+ 名前空間を企業組織の階層に合わせて構成できます。名前空間における情報の配置は、物理的な配置とは関係ありません。したがって、NIS+ 名前空間は、独立して管理できる複数のドメインに分割できます。クライアントは、適切なアクセス権があれば、自分のドメイン以外のドメインの情報にもアクセスできます。

NIS+ はクライアントサーバーモデルを使用して、NIS+ 名前空間に情報を格納し、またその情報にアクセスできます。各ドメインは複数のサーバーによってサポートされます。メインのサーバーは「主」サーバーと呼ばれ、バックアップサーバーは「副」サーバーと呼ばれます。ネットワーク情報は、内部 NIS+ データベース内にある 16 個の標準 NIS+ テーブルに格納されています。主サーバーと副サーバーの両方で NIS+ サーバーソフトウェアが動作しており、NIS+ テーブルのコピーを管理しています。マスターサーバー上の NIS+ データの変更は、副サーバーにも自動的に伝達されます。

NIS+ には、名前空間の構造とその情報を保護するために、高度なセキュリティーシステムが組み込まれています。NIS+ は、情報にアクセスしようとしているクライアントが正当なものであるかどうかを認証と承認によって確認します。「認証」とは、情報の要求者がネットワークの正当なユーザーであるかどうかを判定することです。「承認」では、特定のユーザーが情報を所有したり修正したりできるかどうかを確認します。NIS+ のセキュリティーの詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』を参照してください。

NIS+ から LDAP への移行についての詳細は、第 16 章「NIS+ から LDAP への移行」を参照してください。

LDAP ネームサービスの説明

Solaris オペレーティングシステムは、Sun Java System Directory Server (以前の名称は Sun ONE Directory Server) およびほかの LDAP Directory Server を使用する場合、LDAP (Lightweight Directory Access Protocol) をサポートします。

LDAP ネームサービスについての詳細は、第 8 章「LDAP ネームサービスの紹介 (概要/リファレンス)」を参照してください。

NIS から LDAP、または NIS+ から LDAP への移行についての詳細は、第 15 章「NIS から LDAP への移行 (概要と手順)」または第 16 章「NIS+ から LDAP への移行」を参照してください。

シングルサインオンと、Kerberos 認証サービスの設定および保守についての詳細は、『Solaris のシステム管理 (セキュリティサービス)』のパート VI「Kerberos サービス」を参照してください。

ネームサービスの比較一覧

	DNS	NIS	NIS+	LDAP
名前空間	階層	一層	階層	階層
データ記憶領域	ファイル/リソースレコード	2 列のマップ	複数列のテーブル	ディレクトリ (可変)
サーバー名	マスター/スレーブ	マスター/スレーブ	ルートマスター/非ルートマスター 主/副 キャッシュ/スタブ	マスター/複製
セキュリティー	なし	なし (root またはなし)	Secure RPC (AUTH_DH) 認証	SSL、可変
トランスポート	TCP/IP	RPC	RPC	TCP/IP
規模	広域	LAN	LAN	広域

◆ ◆ ◆ 第 2 章

2 ネームサービススイッチ (概要)

この章では、ネームサービススイッチについて説明します。ネームサービススイッチは、異なるネームサービスの使用方法を調整するために使います。

ネームサービススイッチについて

ネームサービススイッチは `nsswitch.conf` という名前のファイルです。クライアントのマシンやアプリケーションがネットワーク情報を得る方法を管理します。ネームサービススイッチは、次のような `getXbyY()` インタフェースのいずれかを呼び出すクライアントアプリケーションによって使用されます。

- `gethostbyname()`
- `getpwuid()`
- `getpwnam()`
- `getaddrinfo()`

各マシンの `/etc` ディレクトリには、スイッチファイルがあります。ファイルの各行は、ホスト、パスワード、グループなどの特定タイプのネットワーク情報を識別します。そのあとに 1 つまたは複数のネットワーク情報の場所が続きます。

クライアントは、1 つまたは複数のスイッチのソースからネーミング情報を入手できます。たとえば、NIS のクライアントは、NIS マップからホスト情報を、ローカルの `/etc` ファイルからパスワード情報をそれぞれ入手できます。さらに、クライアントはスイッチが各ソースを使用する条件を指定することもできます。表 2–1 を参照してください。

Solaris システムは、インストールの過程で、`nsswitch.conf` ファイルを各マシンの `/etc` ディレクトリに自動的にロードします。LDAP、NIS、NIS+ またはファイル用にスイッチファイルの 4 つの代替 (テンプレート) バージョンも `/etc` にロードされます。39 ページの「`nsswitch.conf` テンプレートファイル」を参照してください。

これら 4 つのファイルは、代替デフォルトスイッチファイルです。各ファイルはそれぞれ `/etc` ファイル、NIS、NIS+、LDAP という異なる主要なネームサービス用に設

計されています。Solaris ソフトウェアをマシンに最初にインストールするとき、インストール担当者によりマシンのデフォルトのネームサービスが選択されます。(NIS+、NIS、ローカルファイル、または LDAP) を選択します。インストール中に、対応するテンプレートファイルが nsswitch.conf ファイルにコピーされます。たとえば、LDAP を使用するクライアントマシンでは、インストールの過程で nsswitch.ldap が nsswitch.conf にコピーされます。特殊な名前空間を持っている場合を除き、通常の操作には nsswitch.conf にコピーされるデフォルトのテンプレートファイルを使用します。

DNS 用のデフォルトファイルは提供されませんが、これらのファイルのどれでも編集して DNS 用に使用できます。詳細は、44 ページの「DNS とインターネットでのアクセス」を参照してください。

マシンの主要なネームサービスを後から変更する場合は、該当する代替スイッチファイルを nsswitch.conf にコピーします。39 ページの「nsswitch.conf テンプレートファイル」を参照してください。NIS 管理者はまた、/etc/nsswitch.conf ファイルの該当行を編集することによって、クライアントで使用する特定タイプのネットワーク情報のソースを変更できます。構文は次で説明します。また、その他の説明は、43 ページの「ネームサービススイッチを変更する方法」に記載されています。

nsswitch.conf ファイルのフォーマット

nsswitch.conf ファイルは、基本的には 16 種類の情報とそのソース (getXXbyYY() ルーチンの情報検索先) のリストです。16 種類の情報は次のとおりです (順序は、必ずしも次のとおりではありません)。

- aliases
- bootparams
- ethers
- group
- hosts
- ipnodes
- netgroup
- netmasks
- networks
- passwd (シャドウ情報含む)
- protocols
- publickey
- rpc
- services
- automount
- sendmailvars

次の表に、スイッチファイルの中で上記の情報タイプ用に表示できるソースの種類とその説明を示します。

表 2–1 スイッチファイルの情報ソース

ソース	説明
files	クライアントの /etc ディレクトリに格納されているローカルファイル。/etc/passwd など
nisplus	NIS+ テーブル。hosts テーブルなど。
nis	NIS マップ。hosts マップなど。
compat	パスワードとグループ情報を対象に、/etc/passwd、/etc/shadow、/etc/group ファイルで旧形式の「+」または「-」構文をサポートします。
dns	ホスト情報を DNS から入手するように指定します。
ldap	エントリを LDAP ディレクトリから入手するように指定します。

検索基準

単一ソース。nisplus のような情報のソースが 1 つだけの場合、スイッチを使用しているルーチンは、そのソースだけで情報を検索します。情報が見つかった場合、「success」という状態メッセージが返されます。情報が見つからない場合は、検索が停止され、「success」以外の状態メッセージが返されます。状態メッセージに基づいて何をするかは、ルーチンによって異なります。

複数ソース。テーブルに特定の情報タイプのソースが複数ある場合、スイッチは最初のソースから検索を行うようにルーチンに指示します。情報が見つかった場合、「success」という状態メッセージが返されます。最初のソースで情報が見つからない場合は、次のソースが検索されます。ルーチンは情報が見つかるか、return 処理によって中止されるまで全ソースを検索します。必要な情報がどのソースにもなかったとき、ルーチンは検索を停止し、non-success という状態メッセージを返します。

スイッチ状態メッセージ

情報が見つかると、「success」という状態メッセージが返されます。探している情報が見つからない場合は、3 種類のエラー状態メッセージのいずれかが返されます。表示される状態メッセージを次の表に示します。

表 2–2 スイッチ状態メッセージ

状態メッセージ	意味
SUCCESS	要求されたエントリがソース内で発見された。

表 2–2 スイッチ状態メッセージ (続き)

状態メッセージ	意味
UNAVAIL	ソースが応答しない、または使用不可。つまり、NIS+ テーブル、NIS マップ、または /etc ディレクトリのファイルが見つからなかったかアクセスできなかった。
NOTFOUND	ソースが「エントリなし」と応答した。つまり、テーブル、マップ、ファイルにアクセスしたが、必要な情報は見つからなかった。
TRYAGAIN	ソース使用中のため再検索の必要あり。つまり、テーブル、マップ、ファイルは見つかったが、照会に対して応答しなかった。

スイッチの動作に関するオプション

次の表に示すように、状態メッセージに対して 2 つの「動作」のどちらかで応答するようにスイッチに指示できます。

表 2–3 スイッチ状態メッセージへの応答

動作	意味
return	情報の検索を停止します。
continue	次のソースの検索を試みます。

デフォルト検索基準

nsswitch.conf ファイルの状態メッセージと動作オプションの組み合わせによって、ルーチンの各ステップでの動作が決まります。状態と動作を組み合わせて、「検索基準」を構成します。

スイッチのデフォルトの検索基準は、どのソースについても同じです。これらを上記の状態メッセージに基づいて説明すると、次のようになります。

- SUCCESS=return。情報の検索を停止します。見つかった情報を使用して処理を続行します。
- UNAVAIL=continue。次のソース (nsswitch.conf ファイルに指定されたもの) を使用して検索を続行します。次のソースがなければ、「NOTFOUND」という状態メッセージを返します。
- NOTFOUND=continue。次のソース (nsswitch.conf ファイルに指定されたもの) を使用して検索を続行します。次のソースがなければ、「NOTFOUND」という状態メッセージを返します。
- TRYAGAIN=continue。次のソース (nsswitch.conf ファイルに指定されたもの) を使用して検索を続行します。次のソースがなければ、「NOTFOUND」という状態メッセージを返します。

前に示した *STATUS=action* 構文を使用することで、ほかの検索基準を明示的に指定してデフォルトの検索基準を変更できます。たとえば、NOTFOUND 状態に対し、デフォルトの動作では次のソースに対する検索を続行しますが、NOTFOUND 状態の場合に検索を停止するように networks の設定を変更するには、スイッチファイルの networks 行を編集します。この行を次のようにします。

```
networks: nis [NOTFOUND=return] files
```

networks: nis [NOTFOUND=return] files 行は、NOTFOUND 状態に関してデフォルトでない検索基準を設定しています。デフォルト以外の設定をするときは [ ] を使用します。

この例では、検索ルーチンは次のような働きをします。

- networks マップが見つかり必要な情報があった場合、ルーチンは「SUCCESS」という状態メッセージを返します。
- networks マップが見つからなかった場合、ルーチンは「UNAVAIL」という状態メッセージを返し、デフォルトで適切な /etc ファイルの検索を続行します。
- networksマップは見つかったがマップの中に必要な情報がなかった場合、ルーチンは「NOTFOUND」という状態メッセージを返します。そして /etc ファイルの検索を続行する (デフォルトの設定) 代わりに検索を停止します。
- networksマップが使用中の場合、ルーチンは TRYAGAIN という状態メッセージを返し、デフォルトで適当な /etc ファイルの検索を続行します。

注 – nsswitch.conf ファイル内の検索は、項目の記載順に実行されます。ただし、passwd -r *repository* コマンドを使用して特に指定しない限り、パスワードの更新は逆順で実行されます。詳細は、46 ページの「スイッチファイルとパスワード情報」を参照してください。

構文が正しくない場合の処理

クライアントのライブラリルーチンには、nsswitch.conf ファイルにおいて「必要なエントリがない」、「エントリの構文が誤っている」といった場合に使用される、コンパイル時に組み込まれるデフォルトエントリがあります。これらのエントリは nsswitch.conf ファイルのデフォルトエントリと同じものです。

ネームサービススイッチは、テーブル名やソース名のスペルが正しいものとして処理をします。テーブル名やソース名のスペルが正しくない場合は、デフォルト値が使用されます。

auto_home と auto_master

auto_home テーブル、auto_master テーブルとマップのスイッチ検索基準は、automount と呼ばれる1 つのカテゴリに統合されます。

時間帯とスイッチファイル

timezone テーブルはスイッチを使用しないため、スイッチファイルのリストには含まれていません。

nsswitch.conf ファイル中のコメント

nsswitch.conf ファイル中の行のうち、コメント文字 (#) で始まっているものはコメント行として解釈され、ファイルを検索するルーチンでは無視されます。

コメント文字の前の文字列は、nsswitch.conf ファイルを検索するルーチンによって解釈されます。コメント文字よりあとの文字列は、コメントとして解釈され、無視されます。

表 2–4 スイッチファイルのコメント例

行の種類	例
コメント行。	#hosts: nisplus [NOTFOUND=return] files
解釈される行。	hosts: nisplus [NOTFOUND=return] file
部分的に解釈される行。「files」の部分は解釈されません。	hosts: nisplus [NOTFOUND=return] # files

キーサーバーとスイッチファイルの publickey エントリ

注意 – nsswitch.conf に変更を加えたあとは、キーサーバーを再起動する必要があります。

キーサーバーは、起動時にだけネームサービススイッチ構成ファイルの publickey エントリを参照します。スイッチ構成ファイルを変更しても再起動しない限り、キーサーバーは変更を登録しません。

nsswitch.conf テンプレートファイル

Solaris システムでは、さまざまなネームサービスに対応できるように、スイッチテンプレートファイルが 4 つ用意されています。ファイルごとに、異なるデフォルトの情報ソースセットが提供されます。

4 つのテンプレートファイルは、次のとおりです。

- LDAP テンプレートファイル。nsswitch.ldap 構成ファイルでは、マシンの情報の一次ソースとして LDAP ディレクトリが指定されています。

  注 – LDAP ネームサービスを使用するには、すべての LDAP クライアントマシンを正しく設定し、nsswitch.conf を変更する必要があります。詳細は、第 12 章「LDAP クライアントの設定 (手順)」を参照してください。

- NIS+ テンプレートファイル。nsswitch.nisplus 構成ファイルでは、passwd、group、automount、aliases を除くすべての情報の一次ソースとして NIS+ が指定されています。これら 4 つのファイルでは、一次ソースはローカルの /etc ディレクトリのファイルで、二次ソースは NIS+ テーブルです。[NOTFOUND=return] という検索基準は、スイッチが「No such entry」というメッセージを受け取ったら NIS+ テーブルの検索を停止するという意味です。スイッチは、NIS+ サーバーを使用できない場合のみ、ローカルファイルを検索します。
- NIS テンプレートファイル。nsswitch.nis 構成ファイルでは、passwd、group、automount、aliases を除くすべての情報の一次ソースとして NIS が指定されています。これら 4 つのファイルでは、一次ソースはローカルの /etc ディレクトリのファイルで、二次ソースは NIS マップです。[NOTFOUND=return] という検索基準は、スイッチが「No such entry」というメッセージを受け取ったら NIS マップの検索を停止するという意味です。スイッチは、NIS サーバーを使用できない場合のみ、ローカルファイルを検索します。

  passwd、group の情報に関しては files nis という順序で検索するよう指定されているため、/etc/passwd と /etc/group に + エントリを指定する必要はありません。
- files テンプレートファイル。nsswitch.files では、マシンの情報ソースとしてローカルの /etc ディレクトリのファイルだけが指定されています。netgroup に関する files のソースは存在しないため、クライアントがスイッチファイルでこのエントリを使用することはありません。

要件に一番近いテンプレートファイルを nsswitch.conf 構成ファイルにコピーして、必要に応じてファイルを変更します。

たとえば、LDAP テンプレートファイルを使用する場合は、次のコマンドを入力します。

```
mymachine# cp /etc/nsswitch.ldap /etc/nsswitch.conf
```

デフォルトスイッチテンプレートファイル

Solaris 製品で用意されているスイッチファイルは、次のとおりです。

例 2–1 NIS+ スイッチファイルテンプレート (nsswitch.nisplus)

```
#
#
# /etc/nsswitch.nisplus:
#
#
# An example file that could be copied over to /etc/nsswitch.conf;
# it uses NIS+ (NIS Version 3) in conjunction with files.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet"
# transports.

# the following two lines obviate the "+" entry in /etc/passwd
# and /etc/group.
passwd: files nisplus
group: files nisplus
# consult /etc "files" only if nisplus is down.
hosts: nisplus [NOTFOUND=return] files
# Uncomment the following line, and comment out the above, to use
# both DNS and NIS+. You must also set up the /etc/resolv.conf
# file for DNS name server lookup. See resolv.conf(4).
# hosts: nisplus dns [NOTFOUND=return] files
services: nisplus [NOTFOUND=return] files
networks: nisplus [NOTFOUND=return] files
protocols: nisplus [NOTFOUND=return] files
rpc: nisplus [NOTFOUND=return] files
ethers: nisplus [NOTFOUND=return] files
netmasks: nisplus [NOTFOUND=return] files
bootparams: nisplus [NOTFOUND=return] files
publickey: nisplus
netgroup: nisplus
automount: files nisplus
aliases: files nisplus
sendmailvars: files nisplus
```

注 – publickey エントリでは、nisplus を値のリストの先頭にしてください。たとえば、publickey: nisplus files が、複数の NIS+ ドメインから参照される nsswitch.conf ファイルの正しいエントリです。

例 2–2 NIS スイッチファイルテンプレート

```
#
# /etc/nsswitch.nis:
#
```

例 2–2 NIS スイッチファイルテンプレート (続き)

```
# An example file that could be copied over to /etc/nsswitch.conf;
# it uses NIS (YP) in conjunction with files.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet"
# transports.
#
# the following two lines obviate the "+" entry in /etc/passwd
# and /etc/group.
passwd: files nis
group: files nis
# consult /etc "files" only if nis is down.
hosts: nis [NOTFOUND=return] files
networks: nis [NOTFOUND=return] files
protocols: nis [NOTFOUND=return] files
rpc: nis [NOTFOUND=return] files
ethers: nis [NOTFOUND=return] files
netmasks: nis [NOTFOUND=return] files
bootparams: nis [NOTFOUND=return] files
publickey: nis [NOTFOUND=return] files
netgroup: nis
automount: files nis
aliases: files nis
# for efficient getservbyname() avoid nis
services: files nis
sendmailvars: files
```

例 2–3 Files スイッチファイルテンプレート

```
#
# /etc/nsswitch.files:
#
# An example file that could be copied over to /etc/nsswitch.conf;
# it does not use any naming service.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet"
# transports.
passwd: files
group: files
hosts: files
networks: files
protocols: files
rpc: files
ethers: files
netmasks: files
bootparams: files
publickey: files
# At present there isn't a 'files' back end for netgroup;
# the system will figure it out pretty quickly, and will notuse
# netgroups at all.
netgroup: files
automount: files
aliases: files
services: files
sendmailvars: files
```

例 2–4 LDAP スイッチファイルテンプレート

```
#
# /etc/nsswitch.ldap:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# uses LDAP in conjunction with files.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet" transports.

# the following two lines obviate the "+" entry in /etc/passwd
and /etc/group.
passwd:     files ldap
group:      files ldap

hosts:      ldap [NOTFOUND=return] files

networks:   ldap [NOTFOUND=return] files
protocols:  ldap [NOTFOUND=return] files
rpc:        ldap [NOTFOUND=return] files
ethers:     ldap [NOTFOUND=return] files
netmasks:   ldap [NOTFOUND=return] files
bootparams: ldap [NOTFOUND=return] files
publickey:  ldap [NOTFOUND=return] files

netgroup:   ldap

automount:  files ldap
aliases:    files ldap

# for efficient getservbyname() avoid ldap
services:   files ldap
sendmailvars:   files
```

nsswitch.conf ファイル

Solaris ソフトウェアと共にインストールされるデフォルトの nsswitch.conf ファイルは、インストール時に選択したネームサービスで決まります。ファイルの各行は、ネットワーク情報の種類 (ホスト、パスワード、グループなど) と、それに対する情報ソース (NIS+ テーブル、NIS マップ、DNS ホストテーブル、同一マシン上の /etc など) を対応させています。ネームサービスを選択すると、そのサービスのスイッチテンプレートファイルがコピーされ新しい nsswitch.conf ファイルが作成されます。たとえば、NIS を選択した場合は、nsswitch.nis ファイルがコピーされ新しい nsswitch.conf ファイルが作成されます。

nsswitch.conf ファイルは、次の代替 (テンプレート) バージョンと一緒に、Solaris 9 リリースソフトウェアによって各マシンの /etc ディレクトリに自動的にロードされます。

- /etc/nsswitch.nisplus
- /etc/nsswitch.nis
- /etc/nsswitch.files

- `/etc/nsswitch.ldap`

これらの代替テンプレートファイルには、NIS+ および NIS サービス、ローカルファイル、および LDAP によって使用されるデフォルトのスイッチ構成が含まれています。DNS 用のデフォルトファイルは提供されませんが、これら 4 つのファイルのどれでも編集して DNS 用に使用できます。Solaris ソフトウェアをマシンに最初にインストールするとき、インストール担当者はマシンのデフォルトのネームサービスを選択します。インストール中に、対応するテンプレートファイルが `/etc/nsswitch.conf` にコピーされます。たとえば、NIS を使用しているクライアントマシンでは、インストールの過程で `nsswitch.nis` が `nsswitch.conf` にコピーされます。

ネットワークがインターネットに接続されており、ユーザーが DNS を使用してインターネット上のホストにアクセスする必要がある場合は、DNS 転送を有効にする必要があります。

特殊な名前空間を持っている場合を除き、通常の操作には `nsswitch.conf` にコピーされるデフォルトのテンプレートファイルを使用します。

構成ファイルの変更

マシンのネームサービスを変更するときは、そのマシンのスイッチファイルを新しいネームサービスに対応させて変更する必要があります。たとえば、マシンのネームサービスをファイルから NIS に変更する場合、スイッチファイルを NIS に対応したものに変更する必要があります。スイッチファイルを変更するには、対応するテンプレートファイルを `nsswitch.conf` にコピーします。

NIS+ インストールスクリプトを使って NIS+ をマシンにインストールすると、NIS+ テンプレートファイルが自動的に `nsswitch.conf` にコピーされます。この場合、特にスイッチファイルをカスタマイズしたいというのでなければ、スイッチファイルを明示的に変更する必要はありません。

スイッチファイルを変更する前に、ファイルに列挙されている情報ソースが正しく設定されていることを確認してください。たとえば、NIS 用スイッチファイルに変更するのであれば、ワークステーションには NIS サービスへのアクセス権が必要になり、ローカルファイル用スイッチファイルに変更するのであれば、それらのローカルファイルがワークステーション上に正しく設定されている必要があります。

▼ ネームサービススイッチを変更する方法

スイッチファイルを変更する場合は、次の手順に従います。

注 – LDAP ネームサービスを使用するには、すべての LDAP クライアントマシンを正しく設定し、nsswitch.conf を変更する必要があります。詳細は、第 12 章「LDAP クライアントの設定 (手順)」を参照してください。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2 マシンのネームサービスに適した代替ファイルを nsswitch.conf ファイルにコピーします。

NIS+ バージョン (NIS+ スクリプトによって自動的に行われる)

```
client1# cd /etc
client1# cp nsswitch.nisplus nsswitch.conf
```

NIS バージョン

```
client1# cd /etc
client1# cp nsswitch.nis nsswitch.conf
```

ローカルの /etc ファイルバージョン

```
client1# cd /etc
client1# cp nsswitch.files nsswitch.conf
```

3 マシンをリブートします。

nscd デーモンによってスイッチ情報がキャッシュに書き込まれます。nscd(1M) のマニュアルページを参照してください。

ライブラリルーチンの中には、nsswitch.conf ファイルが変更されたかどうかを定期的にチェックしないものがあります。 このため、マシンをリブートして、nscd とこれらのライブラリルーチンが確実に最新スイッチの情報を持つようにする必要があります。

DNS とインターネットでのアクセス

nsswitch.conf ファイルでは、次のいくつかのセクションで説明するとおり、クライアントの DNS 転送も制御されます。DNS 転送によって、クライアントへのインターネットでのアクセスが可能になります。NIS との互換性を持つ NIS+ 用に DNS 転送を設定する方法については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』を参照してください。

IPv6 と Solaris ネームサービス

NIS、NIS+、および LDAP では、IPv6 データを格納できるだけでなく、プロトコルのトラフィックに IPv6 トランスポートを使用することもできます。BIND Version 8.3.3 から、Solaris OS 上の DNS でクライアント側での IPv6 トランスポートの使用がサポートされています。BIND Version 8.4.2 以降は、DNS が Solaris での IPv6 ネットワークに対するクライアントサーバーの完全なソリューションを提供しています。

`nsswitch.conf` ファイルは、IPv6 アドレスの検索基準を制御します。IPv6 は、32 ビットから 128 ビットまで IP アドレスサイズを大きくして、より多くのアドレス階層をサポートし、より多くのノードにアドレス指定できるようにします。IPv6 の構成と実装の詳細については、『Solaris のシステム管理 (IP サービス)』を参照してください。

IPv6 アドレスには、新しい `ipnodes` ソースを使用してください。`/etc/inet/ipnodes` ファイルには、IPv4 と IPv6 のアドレスが格納されています。`/etc/inet/ipnodes` ファイルは、`/etc/hosts` ファイルと同じフォーマットを使用します。

IPv6 のネームサービスでは、検索用に新しい `ipnodes` ソースを使用しています。たとえば、LDAP で IPv6 アドレスを認識させる場合は、次のように指定します。

```
ipnodes: ldap [NOTFOUND=return] files
```

注意 – 起こり得る遅延の問題について

- `ipnodes` は、デフォルトでは `files` です。IPv4 から IPv6 への移行中には、すべてのネームサービスが、IPv6 アドレスを認識できるわけではないので、デフォルトの `files` を使用します。このデフォルトを使用しない場合には、アドレスの解決中に不必要な遅延が生じることがあります (ブート時の遅延など)。
- アプリケーションは、IPv4 のアドレスを `ipnodes` データベースで検索してから、`hosts` データベースを検索します。`ipnodes` を指定する前に、IPv4 アドレスの両方のデータベースを検索する時間を考慮にいれる必要があります。

+/- 構文との互換性を追加する

`/etc/passwd`、`/etc/shadow`,、`/etc/group` の各ファイルで +/- 構文を使用する場合は、`nsswitch.conf` ファイルを変更して互換性を確保する必要があります。

- NIS+。NIS+ で +/- 構文と同じ効果を得るには、`passwd` および `group` のソースを `compat` に変更します。次に、`passwd_compat: nisplus` というエントリを、`nsswitch.conf` ファイルの `passwd` または `group` エントリのあとに追加します。

  ```
  passwd: compat
  passwd_compat: nisplus
  ```

```
group: compat
group_compat: nisplus
```

上記の指定により、/etc ファイルと NIS+ テーブルからファイルの +/- エントリで指定されているとおりにクライアントルーチンでネットワーク情報が入手されます。

- NIS。Solaris 4.x リリースの構文と同じ効果を得るには、passwd と group の各ソースを compat に変更します。

```
passwd: compat
group: compat
```

この指定により、/etc ファイルと NIS マップから +/- エントリで指定されているとおりにネットワーク情報が入手されます。

注 – NIS+ サーバーが NIS 互換モードで動作している場合、クライアントマシンでは netgroup テーブルに対して ypcat を実行できません。実行すると、エントリの有無に関わらず「テーブルが空である」という結果が返されます。

スイッチファイルとパスワード情報

files、nisplus などの複数のリポジトリ内のパスワード情報全体にアクセスできます。nsswitch.conf ファイルを使用して、その情報の検索順序を設定できます。

注意 – nsswitch.conf ファイルでは、files を passwd 情報の最初のソースにしてください。

NIS+ の環境では、nsswitch.conf ファイルの passwd 行に次の順序でリポジトリを指定します。

```
passwd: files nisplus
```

NIS の環境では、nsswitch.conf ファイルの passwd 行に次の順序でリポジトリを指定します。

```
passwd: files nis
```

ヒント – files を最初に指定すると、システムのネットワークまたはネームサービスにいくつか問題があっても、ほとんどの場合 root でログインできます。

同一ユーザーを複数のリポジトリで管理することは、推奨されていません。ユーザーごとに 1 つのリポジトリで集中的にパスワードを管理することに

よって、混乱や誤りを減らすことができます。ユーザーごとに複数のリポジトリを管理する場合は、 `passwd -r` コマンドを使用してパスワード情報を更新します。

```
passwd -r repository
```

`-r` オプションによってリポジトリを指定しないと、`passwd` は `nsswitch.conf` に指定されているリポジトリを逆順に更新します。

パート II

DNSの設定と管理

ここでは、Solaris OSにおけるBIND 9 DNSネームサービスの構成および管理について説明します。

◆ ◆ ◆ 第 3 章

3 DNS の設定と管理 (リファレンス)

Solaris オペレーティングシステム (Solaris OS) には、BIND 9.x DNS ネームサーバーが付属しています。この章では、Solaris オペレーティングシステムで BIND 9 を使用する場合の構成および管理について説明します。BIND および DNS の全般については、51 ページの「関連資料」を含め、豊富な資料が用意されています。

この章の内容は次のとおりです。

関連資料

DNS と BIND の管理については、次の文書を参照してください。

- 『BIND 9 Migration Notes』 (`/usr/share/doc/bind/migration.txt`)
- 『BIND 9 管理者のマニュアル』(ISC (Internet Systems Consortium) の Web サイト (http://www.isc.org))
- BIND の機能、既知のバグと不具合、および ISC の Web サイト (http://www.isc.org) 上の資料へのリンク
- 『DNS & BIND』 Paul Albitz/Cricket Liu 著 (高田広章/小島育夫監訳、小舘光正訳、オライリー・ジャパン、2002 年)

BIND 8 から BIND 9 への移行

BIND 9 は、BIND 8 の大部分の機能と上位互換性があります。ただし、BIND 9 を使用するために既存の BIND 8 インストールをアップグレードする際に知っておく必要のある多数の注意事項があります。BIND 9 のインストールや使用の前には、必ず『Migration Notes』をお読みください。『Migration Notes』は、`/usr/share/doc/bind/migration.txt` から入手可能です。また、BIND のパッケージ名は SUNWbind および SUNWbindr に変更されています。SUNWbindr パッケージに DNS サーバー目録が含まれています。

BIND 8 と BIND 9 の相違点の概要を次に示します。詳細については、『Migration Notes』を参照してください。

- 構成ファイルの互換性
  - 実装されないオプションの警告メッセージ
  - *transfer-format* オプションの変更
  - 構成ファイルのエラー
  - ロギングカテゴリの変更
  - 通知メッセージと更新照会の変更
  - 複数のクラスの変更
- ゾーンファイルの互換性
  - ゾーンファイルにおける TTL に関するより厳密なルール
  - SOA シリアル番号の変更
  - 対になっていない引用符によるエラー
  - 改行および構文の変更
  - ドメイン名における $$ に代わる \$ の使用
- 新しいプロトコルの機能による相互運用性への影響
  - BIND 9 の新規 EDNS0
  - ゾーン転送のデフォルトの変更
- 文字セットの制限解除
  - 文字セットに関する制限の解除
  - セキュリティー上の問題、不適切な命名
- サーバー管理ツール
  - `ndc` プログラムの `rndc` プログラムへの置き換え
  - `nsupdate`: 複数更新における変更
- ゾーン間の情報漏れの防止
  - グルー NS レコードの異なる処理
- 変更されない umask
  - umask のアクセス権に関する問題

DNS とサービス管理機能

DNS/BIND named サービスは、サービス管理機能 (SMF) によって管理できます。SMF の概要については、『Solaris のシステム管理 (基本編)』の第 18 章「サービスの管理 (概要)」を参照してください。さらに、詳細は、svcadm(1M)、svcs(1)、および svccfg(1M) のマニュアルページを参照してください。
/var/svc/manifest/network/dns にある DNS サーバー目録 server.xml も確認してください。

- このサービスに関する有効化、無効化、再起動などの管理アクションは svcadm コマンドを使用して実行できます。

ヒント – -t オプションを使用してサービスを一時的に無効化すると、そのサービス構成に対していくらかの保護を提供できます。-t オプションを指定してサービスを無効にした場合、リブート後に元の設定が復元されます。-t オプションを指定しないでサービスを無効にした場合、リブート後もそのサービスは無効のままです。

- DNS サービスに対する障害管理リソース識別子 (FMRI) は、svc:/network/dns/server:<*instance*> および svc:/network/dns/client:<*instance*> です。
- DNS サーバーおよびクライアントの状態の照会は、svcs コマンドを使用して実行できます。
  - svcs コマンドと出力の例を、次に示します。

```
# svcs \*dns\*
STATE          STIME    FMRI
online         Nov_16   svc:/network/dns/server:default
online         Nov_16   svc:/network/dns/client:default
```

  - svcs -l コマンドと出力の例を、次に示します。

```
# svcs -l /network/dns/server
fmri         svc:/network/dns/server:default
name         Internet domain name server (DNS)
enabled      true
state        online
next_state   none
restarter    svc:/system/svc/restarter:default
contract_id  25
dependency   require_all/none svc:/system/filesystem/minimal (online)
dependency   require_all/none file://localhost/etc/named.conf (online)
dependency   require_any/error svc:/network/loopback (online)
dependency   optional_all/error svc:/network/physical (online)
```

- DNS サービスを異なるオプション (たとえば、/etc/named.conf 以外の構成ファイル) によって開始する必要がある場合、svccfg コマンドを使用して DNS サーバー目録の *start method* のプロパティーを変更します。

- BIND 9 ネームサービスの複数のコピーを実行する場合にのみ、SMF サービスの複数のインスタンスが必要です。追加インスタンスはそれぞれ、異なる開始メソッドを使用して DNS サーバー目録で指定できます。

サーバーの管理には svcadm を使用することをお勧めしますが、rndc も使用できます。管理に svcadm と rndc のどちらを使用しても、SMF は BIND 9 の named サービスの状態変化を認識します。

注 – サービスをコマンド行から手動で実行した場合は、SMF は BIND 9 の named サービスを認識しません。

---

rndc の実装

BIND 8 ndc と BIND 9 rndc のネームサーバー制御ツールは、下位互換性がありません。rndc は BIND 8 ネームサーバーに接続できず、ndc は BIND 9 ネームサーバーに接続できません。機能、オプション、操作のデフォルトモード、および構成ファイルの要件が変更されています。そのため、BIND 9 サーバーで ndc を使用すると、機能が使用できなくなったり操作が不安定になったりすることがあります。詳細は、rndc(1M) のマニュアルページを参照してください。

rndc.conf 構成ファイル

BIND 8 の ndc と BIND 9 の rndc とのもっとも大きな違いは、rndc はそれ自身の構成ファイル rndc.conf を必要とすることです。rndc-confgen コマンドによってこのファイルを生成できます。rndc.conf ファイルは、制御を行うサーバー、およびそのサーバーが使用する必要のあるアルゴリズムを指定します。

例 3–1 rndc.conf ファイルの例

```
options {
             default-server localhost;
             default-key "rndc-key";
     };

     key "rndc-key" {
             algorithm hmac-md5;
             secret "qPWZ3Ndl81aBRY9AmJhVtU==";
     };
```

例 3–2 rndc の named.conf ファイルエントリの例

```
controls {
             inet * allow { any; } keys { "rndc-key"; };
     };

     key "rndc-key" {
```

例 3–2 rndc の named.conf ファイルエントリの例 (続き)

```
            algorithm hmac-md5;
            secret "qPWZ3Ndl81aBRY9AmJhVtU==";
    };
```

制御チャネルの相違点

ndc ユーティリティーと rndc ユーティリティーはどちらも、コマンドをネームサーバーに送信し、情報をネームサーバーから取り出すために制御チャネルを使用します。ただし、これらのユーティリティーには相違点があります。

- BIND 8 の ndc は、AF_UNIX ドメインソケット (UNIX 制御チャネル) または TCP/IP ソケット (inet 制御チャネル) を使用できます。デフォルトでは、BIND 8 サーバーは、in.named にコンパイルされたパス (/var/run/ndc.d/ndc) によって UNIX ドメインソケットを使用するので、ndc は /etc/named.conf のサポートを必要としません。

  これに対して BIND 9 の rndc は、認証された TCP/IP inet 制御チャネルのみを使用するので、BIND 8 と下位互換性がありません。BIND 9 サーバーには、制御チャネルのための UNIX ドメインソケットのサポートがありません。
- rndc を使用する場合、ネームサーバーと通信するために「key」句を指定する必要があります。BIND 9 サーバーと rndc クライアントは、同じキーを共有する必要があります (/etc/named.conf と /etc/rndc.conf で定義される)。BIND 9 で BIND 8 の制御エントリを使用すると、エラーメッセージが表示されます。
- 一部のコマンドオプションが、ndc 実装から rndc 実装に変更されました。-c オプションもそれに含まれ、BIND 9 では構文が異なります。そのため、BIND 9 で制御チャネルを指定するには、rndc -s <*server*> -p <*port*> を使用します。

BIND 9 rndc のコマンド

rndc コマンドの一覧を次に示します。

reload	構成ファイルおよびゾーンを再読み込みする
reload zone [class [view]]	単一ゾーンを再読み込みする
refresh zone [class [view]]	ゾーンの即時保守をスケジュールする
reconfig	構成ファイルと新しいゾーンのみを再読み込みする
stats	サーバー統計を統計ファイルに書き込む
querylog	照会ロギングを切り替える
dumpdb	キャッシュをダンプファイル (named_dump.db) にダンプする

stop	保留中の更新をマスターファイルに保存し、サーバーを停止する
halt	保留中の更新を保存せずにサーバーを停止する
trace	デバッグレベルを 1 ずつ増分する
trace level	デバッグレベルを変更する
notrace	デバッグレベルを 0 に設定する
flush	サーバーのキャッシュをすべてフラッシュする
flush [view]	表示用のサーバーのキャッシュをフラッシュする
status	サーバーの状態を表示する
restart	サーバーを再起動する (未実装)

BIND 9 のコマンド、ファイル、ツールおよびオプション

BIND 9 の一部のコマンド、ファイル、ツール、およびオプションは、BIND 8 と同じです。ただし、その他は変更または追加されています。この項では、BIND 9 の多くのコマンド、ファイル、ツール、オプション、およびそれぞれに関連付けられた新しい動作または変更された動作について説明します。

BIND 9 のツールと構成ファイル

Solaris オペレーティングシステムで使用できる BIND 9.x ツールを次に示します。

```
named
nsupdate
rndc
dnssec-keygen
nslookup
dig
dnssec-makekeyset
dnssec-signkey
dnssec-signzone
named-checkconf
named-checkzone
rndc-confgen
host
```

Solaris 10 リリースでサポートされている BIND 9.x 構成ファイルを次に示します。

/etc/rndc.conf

BIND 8 と BIND 9 のコマンドおよびファイルの比較

BIND 8 と BIND 9 のコマンドおよび構成ファイルの違いを次の表に示します。

BIND 8 のコマンド	BIND 9.x で置き換えられたコマンド
dnskeygen(1M)	dnssec-keygen(1M)
rndc(1M)	rndc(1M)
named-bootconf(1M)	必要なし
nsupdate(1M)	nsupdate(1M)
nslookup(1M)	nslookup(1M)
named-xfer(1M)	必要なし
in.named(1M)	named(1M)
named.conf(4)	named.conf[1]
dig(1M)	dig(1M)

[1] 詳細な named.conf マニュアルページは BIND 9.2.4 には含まれません。58 ページの「named.conf のオプション」に、BIND 9.2.4 でサポートされる named.conf のオプションの概要を示します。

コマンドとオプションの変更の説明

次に示す非互換性はすべて、同等の BIND 9 バイナリでサポートされない BIND 8 の機能およびインタフェースです。BIND 9.x バイナリのオプション、コマンド行オプションまたは機能のすべてを示しているわけではありません。

コマンド	オプションの変更
in.named(1M)	DNS ネームサーバー in.named の一部のコマンド行オプションはサポートされません。
	BIND 9.x ネームサーバーで、-g *group_name*、-q、-r および -w *directory* のオプションがサポートされず、BIND 8.x の -b *config_file* が -c *config_file* で置き換えられます。詳細は、named のマニュアルページを参照してください。
dnssec-keygen(1M)	キーの生成のために使用する BIND 8.x の dnskeygen と BIND 9.x の dnssec-keygen に、共通のオプションはありません。詳細は、dnssec-keygen のマニュアルページを参照してください。

コマンド	オプションの変更
rndc(1M)	BIND 8.x の ndc と BIND 9.x の rndc は大きく異なります。共通のオプションは 1 つもなく、ndc とは異なり、rndc は実行のために /etc/rndc.conf に構成ファイルが必要です。詳細は、rndc、rndc.conf、rndc-confgen のマニュアルページを参照してください。
nsupdate(1M)	BIND 9.x の -nsupdate で、k オプションの構文が変更されました。-k *keydir::keyname* ではなく、 k *keyfile* です。それ以外のもう 1 つの変更は、サーバーへの入力送信の信号として空の行を使用していましたが、明示的に send サブコマンドを使用するようになりました。詳細は、nsupdate のマニュアルページを参照してください。
nslookup(1M)	BIND の 9.x バージョンでサポートされないオプションは、 help、host server、set ignoretc、set noignoretc、set srch[list]=N1[/N2/.../N6]、set ro[ot]=host、root、finger [USER]、ls [opt] DOMAIN [> FILE] です。
named.conf(4)	named.conf の詳細なマニュアルページは、BIND 9.2.4 に含まれていません。サポートされないオプション、実装されないオプション、またはデフォルトが変更されたオプションがあります。オプションの変更のリスト、および BIND 9.2.4 でサポートされる named.conf のすべてのオプションの概要については、58 ページの「named.conf のオプション」を参照してください。

named.conf のオプション

BIND 8 と BIND 9 の named.conf のオプションの違いを次に一覧表示します。変更の簡単な説明も記載しています。「変更」の欄の「変更なし」は、BIND 9 バージョンの named でオプションの機能に変更がないことを表します。

オプション {	変更
[version version_string;]	変更なし
[directory path_name;]	変更なし
[named-xfer path_name;]	旧式[1]
[dump-file path_name;]	変更なし
[memstatistics-file path_name;]	実装されない
[pid-file path_name;]	変更なし
[statistics-file path_name;]	変更なし

[1] アーキテクチャーの違いのために旧式。

オプション {	変更
[auth-nxdomain yes_or_no;]	変更なし[2]
[dialup yes_or_no;	変更なし
[fake-iquery yes_or_no;]	旧式
[fetch-glue yes_or_no;]	旧式
[has-old-clients yes_or_no;]	旧式
[host-statistics yes_or_no;]	実装されない
[host-statistics-max number;]	実装されない
[multiple-cnames yes_or_no;]	旧式
[notify yes_or_no | explicit;]	変更なし
[recursion yes_or_no;]	変更なし
[rfc2308-type1 yes_or_no;]	実装されない
[use-id-pool yes_or_no;]	旧式
[treat-cr-as-space yes_or_no;]	旧式
[also-notify yes_or_no;]	構文の変更[3]
[forward (only | first);]	変更なし[4]
[forwarders { [in_addr ; \ [in_addr ; ...]] };]	変更なし[5]
[check-names (master | slave | \ response) (warn | fail | ignore);]	実装されない
[allow-query { address_match_list };]	変更なし
[allow-recursion { address_match_list };]	変更なし
[allow-transfer { address_match_list };]	変更なし
[blackhole { address_match_list };]	変更なし
[listen-on [port ip_port] \ { address_match_list };]	変更なし
[query-source [address (ip_addr | *)] \	変更なし

[2] デフォルト設定は BIND 8 では *yes*、BIND 9 では *no*。

[3] *yes* の場合は IP アドレスが必要。

[4] 転送機能を指定しないと機能しない。指定しない場合、`no matching 'forwarders' statement` のエラーが表示される。

[5] [forward] 句を参照。

オプション {	変更
[port (ip_port | *)] ;]	変更なし
[lame-ttl number;]	
[max-transfer-time-in number;]	変更なし
[max-ncache-ttl number;]	変更なし
[min-roots number;]	実装されない
[transfer-format (one-answer | \	変更なし[6]
many-answers);]	
[transfers-in number;]	変更なし
[transfers-out number;]	変更なし
[transfers-per-ns number;]	変更なし
[transfer-source ip_addr;]	変更なし
[maintain-ixfr-base yes_or_no;]	旧式
[max-ixfr-log-size number;]	旧式[7]
[coresize size_spec ;]	変更なし
[datasize size_spec ;]	変更なし
[files size_spec ;]	変更なし
[stacksize size_spec ;]	変更なし
[cleaning-interval number;]	変更なし
[heartbeat-interval number;]	変更なし
[interface-interval number;]	変更なし
[statistics-interval number;]	実装されない
[topology { address_match_list };]	実装されない
[sortlist { address_match_list };]	変更なし
[rrset-order { order_spec ; \	実装されない
[order_spec ; ...] };]	
};	

[6] デフォルト設定は BIND 8 では *one-answer*、BIND 9 では *many-answers*。

[7] BIND 9 では、該当するログファイルのサイズを自動的にトリムするので、このオプションは必要ない。

BIND 9 の文

この項では、BIND 8 と BIND 9 の文の相違点について説明します。

Controls 文

unix が ndc のデフォルトであり、引数のすべてがコンパイルされます。*inet* が rndc の唯一のオプションであり、この場合、何もコンパイルされません。

```
Syntax
   controls {
     [ inet ip_addr
       port ip_port
       allow { address_match_list; }; ]    OK
     [ unix path_name
       perm number
       owner number
       group number; ]            Not Implemented
   };
```

ロギング構文は大幅に変更されました。named.conf オプションのリストは、58 ページの「named.conf のオプション」を参照してください。

Zone 文

BIND 8 の named.conf マニュアルページに示されているゾーン文の構文は、次を除いた大部分が BIND 9 でサポートされます。

```
[ pubkey number number number string; ]    Obsolete
[ check-names ( warn | fail | ignore ); ]    Not Implemented
```

ACL 文

BIND 9 で変更なく機能します。

```
Syntax
   acl name {
     address_match_list
   };
```

Key 文

BIND 9 で変更なく機能します。

```
Syntax
   key key_id {
     algorithm algorithm_id;
     secret secret_string;
   };
```

Trusted-Keys 文

変更なく機能しますが、この文を使用するコードは BIND 9.2.4 で使われなくなりました。

```
  Syntax
     trusted-keys {
       [ domain_name flags protocol algorithm key; ]
     };
```

Server 文

support-ixfr は廃止されましたが、次のオプションのすべては BIND 9 で変更なく機能します。*transfer-format* のデフォルトは変更されました。

```
  Syntax
     server ip_addr {
       [ bogus yes_or_no; ]
       [ transfers number; ]
       [ transfer-format ( one-answer | many-answers ); ]
       [ keys { key_id [ key_id ... ] }; ]
       [ edns yes_or_no; ]
     };
```

Include 文

BIND 9 で変更なく機能します。

```
  Syntax
     include path_name;
```

named.conf のオプションの概要

BIND 9.2.4 には named.conf の詳細なマニュアルページが含まれていません。BIND 9.2.4 でサポートされる named.conf のオプションの概要を次に示します。

```
options  {
        blackhole { <address_match_element>; ... };
        coresize <size>;
        datasize <size>;
        deallocate-on-exit <boolean>; // obsolete
        directory <quoted_string>;
        dump-file <quoted_string>;
        fake-iquery <boolean>; // obsolete
        files <size>;
        has-old-clients <boolean>; // obsolete
        heartbeat-interval <integer>;
        host-statistics <boolean>; // not implemented
        host-statistics-max <integer>; // not implemented
        interface-interval <integer>;
        listen-on [ port <integer> ] { <address_match_element>; ... };
```

```
listen-on-v6 [ port <integer> ] { <address_match_element>; ... };
match-mapped-addresses <boolean>;
memstatistics-file <quoted_string>; // not implemented
multiple-cnames <boolean>; // obsolete
named-xfer <quoted_string>; // obsolete
pid-file <quoted_string>;
port <integer>;
random-device <quoted_string>;
recursive-clients <integer>;
rrset-order { [ class <string> ] [ type <string> ] [ name
    <quoted_string> ] <string> <string>; ... }; // not implemented
serial-queries <integer>; // obsolete
serial-query-rate <integer>;
stacksize <size>;
statistics-file <quoted_string>;
statistics-interval <integer>; // not yet implemented
tcp-clients <integer>;
tkey-dhkey <quoted_string> <integer>;
tkey-gssapi-credential <quoted_string>;
tkey-domain <quoted_string>;
transfers-per-ns <integer>;
transfers-in <integer>;
transfers-out <integer>;
treat-cr-as-space <boolean>; // obsolete
use-id-pool <boolean>; // obsolete
use-ixfr <boolean>;
version <quoted_string>;
allow-recursion { <address_match_element>; ... };
allow-v6-synthesis { <address_match_element>; ... };
sortlist { <address_match_element>; ... };
topology { <address_match_element>; ... }; // not implemented
auth-nxdomain <boolean>; // default changed
minimal-responses <boolean>;
recursion <boolean>;
provide-ixfr <boolean>;
request-ixfr <boolean>;
fetch-glue <boolean>; // obsolete
rfc2308-type1 <boolean>; // not yet implemented
additional-from-auth <boolean>;
additional-from-cache <boolean>;
query-source <querysource4>;
query-source-v6 <querysource6>;
cleaning-interval <integer>;
min-roots <integer>; // not implemented
lame-ttl <integer>;
max-ncache-ttl <integer>;
max-cache-ttl <integer>;
transfer-format ( many-answers | one-answer );
max-cache-size <size_no_default>;
check-names <string> <string>; // not implemented
cache-file <quoted_string>;
allow-query { <address_match_element>; ... };
allow-transfer { <address_match_element>; ... };
allow-update-forwarding { <address_match_element>; ... };
allow-notify { <address_match_element>; ... };
notify <notifytype>;
notify-source ( <ipv4_address> | * ) [ port ( <integer> | * ) ];
notify-source-v6 ( <ipv6_address> | * ) [ port ( <integer> | * ) ];
also-notify [ port <integer> ] { ( <ipv4_address> | <ipv6_address>
```

```
            ) [ port <integer> ]; ... };
        dialup <dialuptype>;
        forward ( first | only );
        forwarders [ port <integer> ] { ( <ipv4_address> | <ipv6_address> )
            [ port <integer> ]; ... };
        maintain-ixfr-base <boolean>; // obsolete
        max-ixfr-log-size <size>; // obsolete
        transfer-source ( <ipv4_address> | * ) [ port ( <integer> | * ) ];
        transfer-source-v6 ( <ipv6_address> | * ) [ port ( <integer> | * ) ];
        max-transfer-time-in <integer>;
        max-transfer-time-out <integer>;
        max-transfer-idle-in <integer>;
        max-transfer-idle-out <integer>;
        max-retry-time <integer>;
        min-retry-time <integer>;
        max-refresh-time <integer>;
        min-refresh-time <integer>;
        sig-validity-interval <integer>;
        zone-statistics <boolean>;
};

controls {
        inet ( <ipv4_address> | <ipv6_address> | * ) [ port ( <integer> | *
            ) ] allow { <address_match_element>; ... } [ keys { <string>; ... } ];
        unix <unsupported>; // not implemented
};

acl <string> { <address_match_element>; ... };

logging {
        channel <string> {
                file <logfile>;
                syslog <optional_facility>;
                null;
                stderr;
                severity <logseverity>;
                print-time <boolean>;
                print-severity <boolean>;
                print-category <boolean>;
        };
        category <string> { <string>; ... };
};

view <string> <optional_class> {
        match-clients { <address_match_element>; ... };
        match-destinations { <address_match_element>; ... };
        match-recursive-only <boolean>;
        key <string> {
                algorithm <string>;
                secret <string>;
        };
        zone <string> <optional_class> {
                type ( master | slave | stub | hint | forward );
                allow-update { <address_match_element>; ... };
                file <quoted_string>;
                ixfr-base <quoted_string>; // obsolete
                ixfr-tmp-file <quoted_string>; // obsolete
                masters [ port <integer> ] { ( <ipv4_address> |
                    <ipv6_address> ) [ port <integer> ] [ key <string> ]; ... };
```

```
        pubkey <integer> <integer> <integer> <quoted_string>; //
            obsolete
        update-policy { ( grant | deny ) <string> ( name |
            subdomain | wildcard | self ) <string> <rrtypelist>; ... };
        database <string>;
        check-names <string>; // not implemented
        allow-query { <address_match_element>; ... };
        allow-transfer { <address_match_element>; ... };
        allow-update-forwarding { <address_match_element>; ... };
        allow-notify { <address_match_element>; ... };
        notify <notifytype>;
        notify-source ( <ipv4_address> | * ) [ port ( <integer> | *
            ) ];
        notify-source-v6 ( <ipv6_address> | * ) [ port ( <integer>
            | * ) ];
        also-notify [ port <integer> ] { ( <ipv4_address> |
            <ipv6_address> ) [ port <integer> ]; ... };
        dialup <dialuptype>;
        forward ( first | only );
        forwarders [ port <integer> ] { ( <ipv4_address> |
            <ipv6_address> ) [ port <integer> ]; ... };
        maintain-ixfr-base <boolean>; // obsolete
        max-ixfr-log-size <size>; // obsolete
        transfer-source ( <ipv4_address> | * ) [ port ( <integer> |
            * ) ];
        transfer-source-v6 ( <ipv6_address> | * ) [ port (
            <integer> | * ) ];
        max-transfer-time-in <integer>;
        max-transfer-time-out <integer>;
        max-transfer-idle-in <integer>;
        max-transfer-idle-out <integer>;
        max-retry-time <integer>;
        min-retry-time <integer>;
        max-refresh-time <integer>;
        min-refresh-time <integer>;
        sig-validity-interval <integer>;
        zone-statistics <boolean>;
};
server {
        bogus <boolean>;
        provide-ixfr <boolean>;
        request-ixfr <boolean>;
        support-ixfr <boolean>; // obsolete
        transfers <integer>;
        transfer-format ( many-answers | one-answer );
        keys <server_key>;
        edns <boolean>;
};
trusted-keys { <string> <integer> <integer> <integer>
    <quoted_string>; ... };
allow-recursion { <address_match_element>; ... };
allow-v6-synthesis { <address_match_element>; ... };
sortlist { <address_match_element>; ... };
topology { <address_match_element>; ... }; // not implemented
auth-nxdomain <boolean>; // default changed
minimal-responses <boolean>;
recursion <boolean>;
provide-ixfr <boolean>;
request-ixfr <boolean>;
```

```
        fetch-glue <boolean>; // obsolete
        rfc2308-type1 <boolean>; // not yet implemented
        additional-from-auth <boolean>;
        additional-from-cache <boolean>;
        query-source <querysource4>;
        query-source-v6 <querysource6>;
        cleaning-interval <integer>;
        min-roots <integer>; // not implemented
        lame-ttl <integer>;
        max-ncache-ttl <integer>;
        max-cache-ttl <integer>;
        transfer-format ( many-answers | one-answer );
        max-cache-size <size_no_default>;
        check-names <string> <string>; // not implemented
        cache-file <quoted_string>;
        allow-query { <address_match_element>; ... };
        allow-transfer { <address_match_element>; ... };
        allow-update-forwarding { <address_match_element>; ... };
        allow-notify { <address_match_element>; ... };
        notify <notifytype>;
        notify-source ( <ipv4_address> | * ) [ port ( <integer> | * ) ];
        notify-source-v6 ( <ipv6_address> | * ) [ port ( <integer> | * ) ];
        also-notify [ port <integer> ] { ( <ipv4_address> | <ipv6_address>
            ) [ port <integer> ]; ... };
        dialup <dialuptype>;
        forward ( first | only );
        forwarders [ port <integer> ] { ( <ipv4_address> | <ipv6_address> )
            [ port <integer> ]; ... };
        maintain-ixfr-base <boolean>; // obsolete
        max-ixfr-log-size <size>; // obsolete
        transfer-source ( <ipv4_address> | * ) [ port ( <integer> | * ) ];
        transfer-source-v6 ( <ipv6_address> | * ) [ port ( <integer> | * ) ];
        max-transfer-time-in <integer>;
        max-transfer-time-out <integer>;
        max-transfer-idle-in <integer>;
        max-transfer-idle-out <integer>;
        max-retry-time <integer>;
        min-retry-time <integer>;
        max-refresh-time <integer>;
        min-refresh-time <integer>;
        sig-validity-interval <integer>;
        zone-statistics <boolean>;
};

lwres {
        listen-on [ port <integer> ] { ( <ipv4_address> | <ipv6_address> )
            [ port <integer> ]; ... };
        view <string> <optional_class>;
        search { <string>; ... };
        ndots <integer>;
};

key <string> {
        algorithm <string>;
        secret <string>;
};

zone <string> <optional_class> {
        type ( master | slave | stub | hint | forward );
```

```
        allow-update { <address_match_element>; ... };
        file <quoted_string>;
        ixfr-base <quoted_string>; // obsolete
        ixfr-tmp-file <quoted_string>; // obsolete
        masters [ port <integer> ] { ( <ipv4_address> | <ipv6_address> ) [
            port <integer> ] [ key <string> ]; ... };
        pubkey <integer> <integer> <integer> <quoted_string>; // obsolete
        update-policy { ( grant | deny ) <string> ( name | subdomain |
            wildcard | self ) <string> <rrtypelist>; ... };
        database <string>;
        check-names <string>; // not implemented
        allow-query { <address_match_element>; ... };
        allow-transfer { <address_match_element>; ... };
        allow-update-forwarding { <address_match_element>; ... };
        allow-notify { <address_match_element>; ... };
        notify <notifytype>;
        notify-source ( <ipv4_address> | * ) [ port ( <integer> | * ) ];
        notify-source-v6 ( <ipv6_address> | * ) [ port ( <integer> | * ) ];
        also-notify [ port <integer> ] { ( <ipv4_address> | <ipv6_address>
            ) [ port <integer> ]; ... };
        dialup <dialuptype>;
        forward ( first | only );
        forwarders [ port <integer> ] { ( <ipv4_address> | <ipv6_address> )
            [ port <integer> ]; ... };
        maintain-ixfr-base <boolean>; // obsolete
        max-ixfr-log-size <size>; // obsolete
        transfer-source ( <ipv4_address> | * ) [ port ( <integer> | * ) ];
        transfer-source-v6 ( <ipv6_address> | * ) [ port ( <integer> | * ) ];
        max-transfer-time-in <integer>;
        max-transfer-time-out <integer>;
        max-transfer-idle-in <integer>;
        max-transfer-idle-out <integer>;
        max-retry-time <integer>;
        min-retry-time <integer>;
        max-refresh-time <integer>;
        min-refresh-time <integer>;
        sig-validity-interval <integer>;
        zone-statistics <boolean>;
};

server {
        bogus <boolean>;
        provide-ixfr <boolean>;
        request-ixfr <boolean>;
        support-ixfr <boolean>; // obsolete
        transfers <integer>;
        transfer-format ( many-answers | one-answer );
        keys <server_key>;
        edns <boolean>;
};

trusted-keys { <string> <integer> <integer> <integer> <quoted_string>; ... };
```

パート III

NISの設定と管理

ここでは、NISネームサービスの概要と、Solaris OS内でのNISの設定、管理、そして障害の対処方法について説明します。

◆ ◆ ◆ 第 4 章

ネットワーク情報サービス (NIS) (概要)

この章では、ネットワーク情報サービス (NIS) の概要について説明します。

NIS とは分散型ネームサービスであり、ネットワーク上のオブジェクトおよびリソースを識別し、探索するメカニズムです。NIS は、ネットワーク全体の情報に関する一様な記憶領域と検索方法を、トランスポートプロトコルやメディアに依存しない形式で提供します。

この章の内容は次のとおりです。

NIS の概要

システム管理者は、NIS を実行することにより、「マップ」と呼ばれる管理データベースをさまざまなサーバー (「マスター」と「スレーブ」) に分散させることができます。さらに、これらの管理データベースを一元管理により自動的かつ確実な方法で更新できるため、どのクライアントもネットワーク全体を通して一貫した方法で同じネームサービス情報を共有できます。

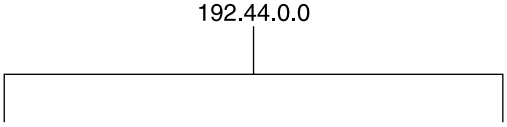
NIS は DNS から独立して開発されたため、その焦点は少し異なっています。DNS は数値 IP アドレスの代わりにマシン名を使うことによって、通信を簡略化することに焦点を当てているのに対して、NIS の場合は、多様なネットワーク情報を集中管理することによりネットワーク管理機能を高めることに焦点を絞っています。NIS には、マシン名とアドレスだけでなく、ユーザー、ネットワークそのもの、ネットワークサービスについての情報も格納されます。このようなネットワーク「情報」の集まりを NIS の「名前空間」と呼びます。

注 – 「マシン」名の代わりに「ホスト」名が使われることがあります。この解説では「マシン」名が使われていますが、一部の画面メッセージまたは NIS マップ名では「ホスト」名または「マシン」名が使われています。

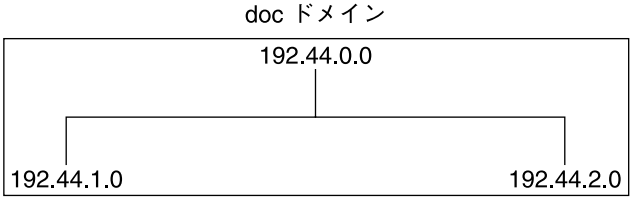
NIS アーキテクチャー

NIS はクライアントサーバー方式を使用します。NIS サーバーが NIS のクライアントへサービスを提供します。主サーバーは「マスター」サーバーと呼ばれ、信頼性を保証するためにバックアップつまり「スレーブ」サーバーを持っています。マスターサーバーとスレーブサーバーは、NIS の情報検索ソフトウェアを使い、NIS のマップを格納します。

NIS はドメインを使用して、マシン、ユーザー、およびネットワークをその名前空間に配置します。しかし、ドメイン階層を使用しないため、NIS の名前空間はフラットになっています。



したがって、上記のような物理ネットワークは、次のように 1 つの NIS ドメインに配置されます。



NIS だけを使っても、NIS ドメインをインターネットに直接接続することはできません。ただし、NIS を使用してインターネットへも接続したいと希望する組織では、NIS と DNS を組み合せることができます。その場合、NIS を使用してすべてのローカル情報を管理し、DNS を使用してインターネットのホストを検索できます。NIS は、NIS マップで情報が見つからない場合にホスト検索の機能を DNS へ転送する転送サービス機能を持っています。Solaris システムでは、ホスト検索要求を DNS だけに転送する、DNS で情報が見つからなければ次に NIS に転送する、あるい

は NIS で情報が見つからなければ次に DNS に転送する、という切り替えも `nsswitch.conf` ファイルで設定できます。詳細については、第 2 章「ネームサービススイッチ (概要)」を参照してください。

NIS マシンのタイプ

NIS マシンには、次の 3 つのタイプがあります。

- マスターサーバー
- スレーブサーバー
- NIS サーバーのクライアント

NIS クライアントにはどのマシンでもなれますが、NIS サーバー (マスターまたはスレーブ) となるのはディスクが装備されているマシンだけです。一般にサーバーは、多くの場合はそのサーバー自身のクライアントでもあります。

NIS サーバー

NIS サーバーは、FNS ファイルサーバーと同じマシンである必要はありません。

NIS サーバーには、マスターサーバーとスレーブサーバーの 2 つの種類があります。マスターサーバーとして指定されているマシンには、NIS 管理者が必要に応じて作成、更新する一群のマップが保存されます。各 NIS ドメインには、マスターサーバーが 1 つだけ存在している必要があります。マスターサーバーは、パフォーマンスの低下を最小限に抑えながら NIS の更新をスレーブサーバーに伝播できます。

ドメインに別の NIS サーバーをスレーブサーバーとして指定できます。各スレーブサーバーには、マスターサーバーの NIS マップセットの完全なコピーが存在します。マスターサーバーの NIS マップが更新されると、必ずこれらの更新がスレーブサーバーに伝播されます。スレーブサーバーは、マスターサーバーからの要求のオーバーフローに対処して、「サーバー使用不可」エラーを最小限に抑えることができます。

通常、システム管理者はすべての NIS マップに対してマスターサーバーを 1 つ指定します。ただし、各 NIS マップ内ではマスターサーバーのマシン名が符合化されているので、異なる複数のマップに対して異なる複数のサーバーを、マスターサーバーやスレーブサーバーとして動作するように指定することもできます。管理の複雑さを最小限に抑えるには、1 つのドメイン内で作成されるすべてのマップに対して、マスターサーバーを 1 つだけ指定します。この章の例では、1 つのサーバーがドメイン内のすべてのマップのマスターサーバーとなっています。

NIS クライアント

NIS クライアントでは、サーバー上のマップのデータを要求するプロセスが動作します。各 NIS サーバーに保存されている情報は同じであるはずなので、クライアントではマスターサーバーとスレーブサーバーの区別は行われません。

注 – Solaris オペレーティングシステムは、NIS クライアントとネイティブな LDAP クライアントが同一のクライアントマシン上に共存する構成をサポートしません。

NIS の要素

NIS ネームサービスは、次の要素から構成されています。

- ドメイン (74 ページの「NIS ドメイン」を参照)
- デーモン (74 ページの「NIS デーモン」を参照)
- ユーティリティー (75 ページの「NIS ユーティリティー」を参照)
- マップ (76 ページの「NIS のマップ」を参照)
- NIS コマンドセット (80 ページの「NIS 関連コマンド」を参照)

NIS ドメイン

NIS「ドメイン」は、共通の NIS マップセットを共有しているマシンの集合です。各ドメインにはドメイン名が指定されており、共通の NIS マップセットを共有している各マシンがそのドメインに属しています。

どのマシンも指定されたドメインに属することができます。ただしこれは、そのドメインのマップに対するサーバーが同一ネットワーク上に存在する場合に限ります。NIS クライアントマシンは、ブートプロセス中にドメイン名を取得して、NIS サーバーにバインドします。

NIS デーモン

NIS サービスは、表 4–1 に示す 5 つのデーモンで提供されます。NIS サービスはサービス管理機能によって管理されます。このサービスに関する有効化、無効化、再起動などの管理アクションは `svcadm` コマンドを使用して実行できます。SMF の概要については、『Solaris のシステム管理 (基本編)』の第 18 章「サービスの管理 (概要)」を参照してください。また、詳細については、`svcadm(1M)` および `svcs(1)` のマニュアルページを参照してください。

表4–1 NISデーモン

デーモン	機能
ypserv	サーバープロセス
ypbind	バインドプロセス
ypxfrd	高速マップ転送
rpc.yppasswdd	NISパスワード更新デーモン ** 下の注を参照 **
rpc.ypupdated	ほかのマップ (publickeyなど) を更新する

注 – rpc.yppasswddは、rで始まるすべてのシェルを制限付きとみなします。たとえば、/bin/rkshで作業しているユーザーはそのシェルを別のシェルに変更できません。rで始まるシェルを持っているが、そのような制約を受けたくない場合は、第7章「NISのトラブルシューティング」の対処方法を参照してください。

NISユーティリティー

NISサービスは、表4–2に示す9つのユーティリティーでサポートされています。

表4–2 NISユーティリティー

ユーティリティー	機能
makedbm	NISマップのdbmファイルを作成します
ypcat	マップのデータを一覧表示します
ypinit	NISデータベースの作成、インストール、およびNISクライアントのypserversリストの初期化を行います
ypmatch	マップの特定エントリを検索します
yppoll	サーバーからマップ順序番号を取得します
yppush	データをNISマスターサーバーからNISスレーブサーバーに伝播します
ypset	特定サーバーにバインドを設定します
ypwhich	NISサーバー名およびニックネーム変換テーブルを表示します
ypxfr	NISマスターサーバーからNISスレーブサーバーにデータを転送します

NIS のマップ

NIS マップの情報は、ndbm フォーマットで保存されます。マップファイルのフォーマットについては、ypfiles(4) と ndbm(3C) のマニュアルページで説明しています。

NIS マップは、UNIX の /etc ファイルおよびほかの構成ファイルを置換するように設計されているので、名前およびアドレスよりはるかに多くの情報を保存できます。NIS が動作しているネットワーク上では、各 NIS ドメインの NIS マスターサーバーは、照会されるドメイン内のほかのマシンの NIS マップセットを保持します。NIS スレーブサーバーは、NIS マスターサーバーのマップのコピーを保持します。NIS クライアントマシンは、マスターサーバーまたはスレーブサーバーから名前空間情報を取得できます。

NIS マップは、本質的には 2 つの列からなるテーブルです。1 つの列は「キー」であり、もう 1 つの列はキーに関連する情報です。NIS は、キーを検索してクライアントに関する情報を見つけます。各マップでは異なるキーが使われるので、一部の情報はいくつかのマップに保存されます。たとえば、マシン名とアドレスは、hosts.byname および hosts.byaddr という 2 つのマップに保存されます。サーバーがマシンの名前を持っており、そのマシンのアドレスを見つける必要がある場合は、サーバーは hosts.byname マップを調べます。サーバーがマシンのアドレスを持っており、そのマシンの名前を見つける必要がある場合は、サーバーは hosts.byaddr マップを調べます。

NIS Makefile は、インストール時に NIS サーバーとして指定されたマシンの /var/yp ディレクトリに保存されます。このディレクトリで make を実行すると、makedbm が入力ファイルからデフォルトの NIS マップを作成または更新します。

注 – マップは必ずマスターサーバー上で作成してください。スレーブサーバーで作成したマップはマスターサーバーに自動的に格納されません。

デフォルトの NIS マップ

Solaris システムには、NIS マップのデフォルトセットが提供されています。システム管理者は、これらのマップをすべて使用することも一部だけを使用することもできます。また、ほかのソフトウェア製品のインストール時にシステム管理者が作成または追加したマップもすべて NIS で使用できます。

NIS ドメインのデフォルトのマップは、各サーバーの /var/yp/*domainname* ディレクトリに入っています。たとえば、test.com ドメインに属しているマップは、各サーバーの /var/yp/test.com ディレクトリにあります。

表 4–3 には、デフォルトの NIS マップ、これらの NIS マップに存在する情報、および NIS 動作時にソフトウェアが対応する管理ファイルを調べているか否かが示されています。

表 4–3 NISマップに関する説明

マップ名	対応するNIS管理ファイル	説明
audit_user	audit_user	ユーザー監査の事前選択データを含みます。
auth_attr	auth_attr	承認名と説明を含みます。
bootparams	bootparams	ブート時にクライアントが必要とするファイルのパス名 (ルート、スワップ、その他) を含みます。
ethers.byaddr	ethers	マシン名とEthernetアドレスを含みます。Ethernetアドレスはマップ内のキーです。
ethers.byname	ethers	ethers.byaddr と同じです。ただしキーは、Ethernetアドレスではなくマシン名です。
exec_attr	exec_attr	プロファイルの実行属性を含みます。
group.bygid	group	グループセキュリティー情報を含みます。キーはグループIDです。
group.byname	group	グループセキュリティー情報を含みます。キーはグループ名です。
hosts.byaddr	hosts	マシン名とIPアドレスを含みます。キーはIPアドレスです。
hosts.byname	hosts	マシン名とIPアドレスを含みます。キーはマシン (ホスト) 名です。
mail.aliases	aliases	エイリアスとメールアドレスを含みます。キーはエイリアスです。
mail.byaddr	aliases	メールアドレスとエイリアスを含みます。キーはメールアドレスです。
netgroup.byhost	netgroup	グループ名、ユーザー名、マシン名を含みます。キーはマシン名です。
netgroup.byuser	netgroup	netgroup.byhost と同じです。ただし、キーはユーザー名です。
netgroup	netgroup	netgroup.byhost と同じです。ただし、キーはグループ名です。

表 4–3 NIS マップに関する説明 (続き)

マップ名	対応する NIS 管理ファイル	説明
netid.byname	passwd, hosts group	UNIX スタイルの認証に使用されます。マシン名とメールアドレスを含みます (ドメイン名も含む)。netid ファイルがある場合には、ほかのファイルを使用して利用できるデータのほかにそれが参照されます。
netmasks.byaddr	netmasks	IP 送出時に使用するネットワークマスクを含みます。キーはアドレスです。
networks.byaddr	networks	システムに認識されているネットワーク名、および IP アドレスを含みます。キーは IP アドレスです。
networks.byname	networks	networks.byaddr と同じです。ただし、キーはネットワーク名です。
passwd.adjunct.byname	passwd と shadow	C2 クライアント用の監査情報と隠蔽されたパスワード情報を含みます。
passwd.byname	passwd と shadow	パスワード情報を含みます。キーはユーザー名です。
passwd.byuid	passwd と shadow	passwd.byname と同じです。ただし、キーはユーザー ID です。
prof_attr	prof_attr	実行プロファイルの属性を含みます。
protocols.byname	protocols	システムに認識されているネットワークプロトコルを含みます。
protocols.bynumber	protocols	protocols.byname と同じです。ただし、キーはプロトコル番号です。
rpc.bynumber	rpc	システムに認識されている RPC のプログラム番号と名前を含みます。キーは RPC のプログラム番号です。
services.byname	services	ネットワークに認識されているインターネットサービスを一覧表示します。キーはポートまたはプロトコルです。
services.byservice	services	ネットワークに認識されているインターネットサービスを一覧表示します。キーはサービス名です。
user_attr	user_attr	ユーザーと役割に関する拡張属性を含みます。
ypservers	なし	ネットワークに認識されている NIS サーバーを一覧表示します。

新しい ipnodes マップ (ipnodes.byaddr および ipnodes.byname) が、NIS に追加されました。このマップには、IPv4 アドレスと IPv6 アドレスの両方が格納されます。

注 – Solaris 10 8/07 リリース以降、Solaris OS には 2 つの別個の hosts ファイルは存在しません。/etc/inet/hosts ファイルが唯一の hosts ファイルであり、この中に IPv4 と IPv6 の両方のエントリが含まれます。常に同期させる必要がある 2 つの hosts ファイル内の IPv4 エントリを維持管理する必要はありません。/etc/inet/ipnodes ファイルは、下位互換性のために、/etc/inet/hosts ファイルへの同名のシンボリックリンクに置き換えられています。

詳細は、hosts(4) のマニュアルページを参照してください。

NIS クライアントとサーバーは、IPv4 または IPv6 のどちらかの RPC トランスポートを使用して通信することができます。

---

ageing.byname マッピングには、yppasswdd によって使用される情報が含まれています。NIS から LDAP への移行時に、DIT とのパスワード有効期限情報の読み取りおよび書き込みのために使用されます。パスワードの有効期限を使用しない場合は、この情報をマッピングファイルからコメントアウトします。NIS から LDAP への移行の詳細については、第 15 章「NIS から LDAP への移行 (概要と手順)」を参照してください。

NIS マップの使用

NIS を使うと、/etc ファイルシステムを使った場合に比べ、ネットワークデータベースの更新がはるかに簡単になります。/etc ファイルシステムではネットワーク環境を更新するたびに各マシンの管理 /etc ファイルを変更する必要がありましたが、NIS ではこのような操作を行う必要はありません。

たとえば、NIS が動作しているネットワークに新しいマシンを追加する場合に必要なのは、マスターサーバーの入力ファイルを更新し、make を実行することだけです。これで、hosts.byname および hosts.byaddr マップが自動的に更新されます。次に、これらのマップはすべてのスレーブサーバーに転送され、ドメインのすべてのクライアントマシン、およびこれらのクライアントマシンのプログラムはこれらのマップを使用することが可能になります。クライアントマシンまたはアプリケーションがマシン名またはアドレスを要求すると、NIS サーバーは必要に応じて hosts.byname または hosts.byaddr マップを参照し、要求された情報をクライアントに送信します。

ypcat コマンドを使うと、マップの値を表示できます。ypcat の基本的な使用形式は、次のとおりです。

```
% ypcat mapname
```

mapname は、調べたいマップ名またはその「ニックネーム」です。ypservers の場合のようにマップがキーだけで構成されている場合は、ypcat -k と入力してください。ypcat -k と入力しない場合は、空白行が出力されます。ypcat のほかのオプションについては、ypcat(1) のマニュアルページに説明されています。

ypwhich コマンドを使うと、どのサーバーが特定のマップのマスターサーバーなのかを判定できます。次のように入力します。

```
% ypwhich -m mapname
```

mapname は、検索するマスターサーバーのマップ名またはニックネームです。mapname を入力すると、マスターサーバー名が表示されます。詳細については、ypwhich(1) のマニュアルページを参照してください。

NIS マップのニックネーム

「ニックネーム」は、マップのフルネームのエイリアスです。使用可能なマップのニックネーム (たとえば、passwd.byname の場合は passwd) を一覧表示するには、ypcat -x または ypwhich -x と入力してください。

ニックネームは、/var/yp/nicknames ファイルに保存されています。このファイルには、マップの完全指定名のあとに、マップのニックネームが空白で区切られて含まれています。ニックネームのリストは、追加または更新できます。ニックネーム数は現在、500 に制限されています。

NIS 関連コマンド

NIS サービスには、特殊なデーモン、システムプログラム、コマンドが含まれています。これらのコマンドについては次の表にまとめてあります。

表 4–4 NIS コマンドについてのまとめ

コマンド	説明
ypserv	NIS クライアントが要求する NIS マップの情報を提供します。ypserv は、完全なマップセットを備えた NIS サーバー上で動作するデーモンです。NIS サービスが機能するには、少なくとも 1 つの ypserv デーモンがネットワークに存在する必要があります。
ypbind	クライアントに NIS サーバーバインド情報を提供します。ypbind は、要求元クライアントのドメイン内のマップにサービスを提供する ypserv プロセスを見つけてバインドを行います。ypbind はすべてのサーバーとクライアント上で実行される必要があります。
ypinit	自動的に入力ファイルから NIS サーバーのマップを作成します。ypinit はまた、クライアント上に /var/yp/binding/ *domain*/ypservers 初期ファイルを作成する際にも使用されます。NIS マスターサーバーおよび NIS スレーブサーバーを初めて設定する場合は、ypinit を使用します。

表4–4 NISコマンドについてのまとめ (続き)

コマンド	説明
make	Makefileを読み込むことでNISマップを更新します(makeを/var/ypディレクトリで実行した場合)。makeを使うと、入力ファイルに基づいてすべてのマップを更新したり、個々のマップを更新したりできます。NISのmakeの機能については、ypmake(1M)のマニュアルページに説明されています。
makedbm	makedbmは入力ファイルを取得し、これをdbm.dirおよびdbm.pagファイルに変換します(これらのファイルは、NISがマップとして使用できる有効なdbmファイル)。また、makedbm -uと入力すると、マップを分解できるため、システム管理者はマップを構成するキーと値のペアを参照できます。
ypxfr	NIS自体を転送媒体として使い、NISマップを遠隔サーバーから/var/yp/*domain*ローカルディレクトリに取り込みます。システム管理者はypxfrを対話形式で実行したり、crontabファイルから定期的に実行したりできます。また、ypxfrがypservによって呼び出されると、転送が開始されます。
ypxfrd	ypxfr要求(一般にスレーブサーバーで発生する)に対してマップ転送サービスを提供します。ypxfrは、マスターサーバー上でだけ動作します。
yppush	NISマップの新バージョンをNISマスターサーバーからそのスレーブサーバーにコピーします。yppushの実行は、NISマスターサーバー上で行います。
ypset	指定されたNISサーバーにバインドするようにypbindプロセスに要求します。ypsetは、セキュリティーの関係上、通常のオペレーションで気軽に使用できるようには設計されていません。したがって、ypsetはできる限り使用しないでください。ypbindプロセスのypsetおよびypsetmeオプションについては、ypset(1M)とypbind(1M)のマニュアルページを参照してください。
yppoll	指定されたサーバー上でNISマップのどのバージョンが動作しているかを通知します。yppollはまた、NISマップのマスターサーバーを一覧表示します。
ypcat	NISマップの内容を表示します。
ypmatch	NISマップ内の指定された1つ以上のキーの値を出力します。システム管理者は、NISサーバーマップのバージョンを指定することはできません。
ypwhich	クライアントが現在どのNISサーバーを使用してNISサービスを取得しているかを表示します。また、-m *mapname*オプションを指定してこのコマンドを起動した場合は、どのNISサーバーが各マップのマスターサーバーであるかが表示されます。-mだけを指定した場合は、使用可能なすべてのマップ名、およびこれらのマップのマスターサーバーが表示されます。

NIS のバインド

NIS クライアントは、バインドプロセスにより NIS サーバーから情報を取得します。バインドプロセスは、サーバーリストおよび同報通信という 2 つのモードのどちらかで動作できます。

- サーバーリスト。サーバーリストモードでは ypbind プロセスは、/var/yp/binding/*domain*/ypservers リストでドメイン内のすべての NIS サーバー名を調べます。ypbindプロセスは、このファイルに存在するサーバーにだけバインドされます。このファイルは、ypinit -c を実行すると作成されます。
- 同報通信。ypbind プロセスはまた、RPC 同報通信を使ってバインドを開始できます。同報通信は、それ以上配信されない唯一のローカルサブネットイベントです。したがって、クライアントと同じサブネット上に少なくとも 1 つのサーバー (マスターまたはスレーブ) が存在しなければなりません。サーバーは、異なる複数のサブネット上に存在できます (マップはサブネット境界を超えて伝播されるため)。サブネット環境での 1 つの一般的方法は、NIS サーバーとしてサブネットルーターを使用することです。この方法を使用すると、ドメインサーバーはどちらかのサブネットインタフェース上でクライアントにサービスを提供できます。

サーバーリストモード

サーバーリストモードでは、バインドプロセスは次のように動作します。

1. NIS マップで提供された情報を必要とする、NIS クライアントマシン上で動作しているプログラムが、ypbind にサーバー名を要求します。
2. ypbind が、/var/yp/binding/*domainname*/ypservers ファイルを調べてドメインの NIS サーバーリストを見つけます。
3. ypbind が、NIS サーバーリストの先頭サーバーへのバインドを開始します。先頭サーバーが応答しない場合、ypbind はサーバーが見つかるまで、あるいは NIS サーバーリストの最後に達するまで、2 番目以降のサーバーへのバインドを順に試みます。
4. ypbind が、どのサーバーにアクセスすべきかをクライアントプロセスに通知します。次に、クライアントプロセスが直接、サーバーに要求を送信します。
5. NIS サーバー上の ypserv デーモンが、該当するマップを調べて要求を処理します。
6. ypserv デーモンが、要求された情報をクライアントに送り返します。

同報通信モード

同報通信モードでは、バインドプロセスは次のように動作します。

1. 同報通信オプション (broadcast) が設定されている状態で ypbind を起動する必要があります。
2. ypbind が、RPC 同報通信を送出して NIS サーバーを探索します。

注 – このようなクライアントをサポートするには、NIS サービスを要求している各サブネット上に 1 つの NIS サーバーが存在する必要があります。

---

3. ypbind が、同報通信に応答する先頭サーバーへのバインドを開始します。
4. ypbind が、どのサーバーにアクセスすべきかをクライアントプロセスに通知します。次に、クライアントプロセスが直接、サーバーに要求を送信します。
5. NIS サーバー上の ypserv デーモンが、該当するマップを調べて要求を処理します。
6. ypserv デーモンが、要求された情報をクライアントに送り返します。

通常、いったんクライアントがサーバーにバインドされると、何らかの原因でバインドが解除されるまではクライアントはサーバーにバインドされたままになります。たとえば、サーバーがサービスを提供できなくなると、このサーバーがサービスを提供していたクライアントは、新しいサーバーにバインドされます。

どの NIS サーバーが現在、特定クライアントにサービスを提供しているかを調べる場合は、次のコマンドを入力してください。

%**ypwhich** *machinename*

machinename は、クライアント名です。マシン名が指定されていない場合は、ypwhich はデフォルトとしてローカルマシン (コマンドが実行されるマシン) を使用します。

◆ ◆ ◆ 第 5 章

NIS サービスの設定と構成

この章では、ネットワーク情報サービス (NIS) の初期設定と構成について説明します。

注 – 「マシン」名の代わりに「ホスト」名が使われることがあります。この解説では「マシン」名が使われていますが、一部の画面メッセージまたは NIS マップ名では「ホスト」名または「マシン」名が使われています。

この章の内容は次のとおりです。

NIS の構成 — 作業マップ

作業	参照先
変換用のソースファイルを準備します。	89 ページの「NIS マップへの変換用のソースファイルを準備する」
`ypinit` を使用してマスターサーバーを設定します	92 ページの「`ypinit` によるマスターサーバーの設定」
マスターサーバーで NIS を起動します。	94 ページの「マスターサーバーでの NIS サービスの開始と停止」

作業	参照先
スレーブサーバーを設定します。	96 ページの「スレーブサーバーを設定する」
NIS クライアントを設定します。	97 ページの「NIS クライアントの設定」

NIS の構成を始める前に

NIS の名前空間を構成する前に、次の操作を行う必要があります。

- NIS を使用する予定のすべてのマシンで、正しく構成された `nsswitch.conf` ファイルをインストールする。詳細については、第 2 章「ネームサービススイッチ (概要)」を参照してください。
- NIS ドメインを設計する。

NIS とサービス管理機能

NIS サービスはサービス管理機能によって管理されます。SMF の概要については、『Solaris のシステム管理 (基本編)』の第 18 章「サービスの管理 (概要)」を参照してください。また、詳細については、`svcadm`(1M) および `svcs`(1) のマニュアルページを参照してください。

- このサービスに関する有効化、無効化、再起動などの管理アクションは `svcadm` コマンドを使用して実行できます。NIS を開始または停止するには、コマンド行から `ypstart` および `ypstop` も使用できます。詳細については、`ypstart`(1M) および `ypstop`(1M) のマニュアルページを参照してください。

  ヒント – `-t` オプションを使用してサービスを一時的に無効化すると、そのサービス構成に対していくらかの保護を提供できます。`-t` オプションを指定してサービスを無効にした場合、リブート後に元の設定が復元されます。`-t` オプションを指定しないでサービスを無効にした場合、リブート後もそのサービスは無効のままです。

- NIS の障害管理リソース識別子 (FMRI) は、NIS サーバーに対しては `svc:/network/nis/server:<`*instance*`>`、NIS クライアントに対しては `svc:/network/nis/client:<`*instance*`>` です。
- `svcs` コマンドを使用して NIS の状態を照会できます。
  - `svcs` コマンドと出力の例を、次に示します。

```
# svcs network/nis/server
STATE          STIME    FMRI
online         Jan_10   svc:/network/nis/server:default
```

```
# svcs \*nis\*
STATE          STIME    FMRI
disabled       12:39:18 svc:/network/rpc/nisplus:default
disabled       12:39:18 svc:/network/nis/server:default
disabled       12:39:20 svc:/network/nis/passwd:default
disabled       12:39:20 svc:/network/nis/update:default
disabled       12:39:20 svc:/network/nis/xfr:default
online         12:42:16 svc:/network/nis/client:default
```

- svcs -l コマンドと出力の例を、次に示します。

```
# svcs -l /network/nis/client
fmri         svc:/network/nis/client:default
enabled      true
state        online
next_state   none
restarter    svc:/system/svc/restarter:default
contract_id  99
dependency   exclude_all/none svc:/network/nis/server (offline)
dependency   require_all/none svc:/system/identity:domain (online)
dependency   require_all/restart svc:/network/rpc/bind (online)
dependency   require_all/none svc:/system/filesystem/minimal (online)
```

- サービスに関してより詳細な情報を得るには、svccfg ユーティリティーを使用します。svccfg(1M) のマニュアルページを参照してください。
- デーモンの存在は ps コマンドを使用して確認できます。

```
# ps -e | grep rpcbind
daemon 100806      1   0   Sep 01 ?       25:28   /usr/sbin/rpcbind
```

注 – -f オプションを ps で使用しないでください。このオプションはユーザー ID を名前に変換しようとするため、より多くのネームサービス検索が失敗する可能性があります。

NIS ドメインの設計

NIS サーバーまたはクライアントとしてマシンを構成する前に、NIS ドメインを設計する必要があります。

まず、NIS ドメインに入れるマシンを決めます。NIS ドメインは、ネットワークと同一である必要はありません。ネットワークには複数の NIS ドメインが存在でき、NIS ドメインに属さないマシンもネットワーク上に存在できます。

NIS ドメイン名を選択します。NIS ドメイン名には、最高 256 文字を指定できます。ドメイン名が 32 文字を超えないように制限するとよいでしょう。ドメイン名は大文字と小文字を区別します。便宜上、インターネットのドメイン名に基づいて NIS ドメイン名を指定することもできます。たとえば、インターネットのドメイン名が doc.com の場合、NIS ドメインも doc.com にすることができます。doc.com を 2 つの NIS ドメインに分けて、1 つを営業部門に、もう 1 つを製造部門に使用する場合は、一方を sales.doc.com とし、もう一方を manf.doc.com とできます。

NIS ドメイン名とマシン名を正しく設定しないと、マシンが NIS サービスを使用できるようになりません。マシン名はマシンの /etc/nodename ファイルによって設定され、マシンのドメイン名はマシンの /etc/defaultdomain ファイルによって設定されます。これらのファイルはブート時に読み取られ、その内容はそれぞれ uname -S コマンドと domainname コマンドによって使用されます。ディスクレスマシンは、そのブートサーバーからこれらのファイルを読み取ります。

NIS サーバーとクライアントを特定する

マスターサーバーになるマシンを 1 つ選択します。スレーブサーバーを作成する場合は、スレーブサーバー用のマシンを決定します。

NIS クライアントになるマシンを決定します。通常は、ドメイン内のすべてのマシンが NIS クライアントになるように設定されますが、これは必須ではありません。

マスターサーバーの準備

以降の節では、マスターサーバーのソースファイルと passwd ファイルを準備する方法を説明します。

ソースファイルディレクトリ

ソースファイルは、マスターサーバーの /etc ディレクトリか、その他のディレクトリにあります。ソースファイルを /etc に入れることは望ましくありません。マップの内容がマスターサーバー上のローカルファイルの内容と同じになるからです。これは passwd ファイルと shadow ファイルに固有の問題です。ユーザー全員がマスターサーバーのマップにアクセスし、passwd マップを通じてすべての NIS クライアントに root パスワードが渡されるためです。詳細については、89 ページの「passwd ファイルと名前空間のセキュリティー」を参照してください。

ただし、ソースファイルをほかのディレクトリに入れた場合は、/var/yp 内の Makefile の DIR=/etc 行を DIR=/*your-choice* に変更する必要があります。*your-choice* はソースファイルを格納するためのディレクトリの名前です。これによって、サーバー上のローカルファイルをクライアント上のファイルのように扱うことができます。編集前の Makefile のコピーを保存しておくことをお勧めします。

また、audit_user、auth_attr、exec_attr、prof_attr がデフォルト以外のディレクトリから取り出される場合は、RBACDIR=*/etc/security* を *RBACDIR=/your-choice* に変更します。

passwd ファイルと名前空間のセキュリティー

passwd マップは特殊なケースです。この NIS 実装では、NIS パスワードマップを作成するための入力として、Solaris 1 の passwd ファイルのフォーマットに加え、/etc/passwd ファイルと /etc/shadow ファイルのフォーマットも使用できます。

セキュリティー上の理由から未承認の root アクセスを防ぐために、NIS のパスワードマップの構築に使用されるファイルには root のエントリを含めないでください。このため、パスワードマップはマスターサーバーの /etc ディレクトリに置かれたファイルから構築しないでください。パスワードマップの構築に使用されるパスワードファイルは、root エントリが削除された上、未承認のアクセスから保護されるディレクトリに置かれている必要があります。

たとえば、マスターサーバーのパスワード入力ファイルは、ファイル自体が別のファイルへのリンクではなく、ファイルの場所が Makefile に指定されている限り、/var/yp/ などのディレクトリに格納されているか、選択したディレクトリに格納されている必要があります。Makefile に指定された構成に従って、適正なディレクトリオプションが自動的に設定されます。

注意 – PWDDIR によってディレクトリ内に指定された passwd ファイルに root のエントリが含まれないようにしてください。

ソースファイルが /etc 以外のディレクトリにある場合は、Makefile の PWDIR パスワードマクロが、passwd ファイルと shadow ファイルが入っているディレクトリを参照するように変更します。この操作を行うには、PWDIR=/etc 行を PWDIR=/*your-choice* に変更します。*your-choice* は、passwd マップソースファイルを格納するのに使用するディレクトリの名前です。

NIS マップへの変換用のソースファイルを準備する

NIS マップへの変換用のソースファイルを準備します。

▼ 変換用のソースファイルを準備する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2 マスターサーバーのソースファイルをチェックして、それらのファイルがシステムの最新の状態を反映しているかどうか確認します。

次のファイルを確認します。

- auto.home または auto_home
- auto.master または auto_master
- audit_user
- auth_attr
- bootparams
- ethers
- exec_attr
- group
- hosts
- ipnodes
- netgroup
- netmasks
- networks
- passwd
- protocols
- rpc
- サービス
- shadow
- user_attr

3 これらのソースファイル (passwd を除く) をすべて、選択した DIR ディレクトリにコピーします。

4 passwd ファイルを、選択した PWDIR ディレクトリにコピーします。

5 audit_user、auth_attr、exec_attr、prof_attr を、選択した RBACDIR ディレクトリにコピーします。

6 /etc/mail/aliases ファイルを確認します。

ほかのソースファイルと異なり、/etc/mail/aliases ファイルは別のディレクトリに移動できません。このファイルは /etc/mail ディレクトリに格納されていなければなりません。詳細については、aliases(4) のマニュアルページを参照してください。

注 – /var/yp/Makefile 内の ALIASES = /etc/mail/aliases エントリを変更して別の場所を指すようにすることで、NIS 固有のメールエイリアスファイルを追加することができます。make を実行すると、ALIASES エントリによって mail.aliases マップが作成されます。/etc/nsswitch.conf ファイルで files のほかに nis が適切に指定されている場合、sendmail サービスは /etc/mail/aliases ファイルに加えてこのマップを使用します。107 ページの「Makefile の更新と使用」を参照してください。

---

7 ソースファイルからすべてのコメントと、その他の余計な行や情報を取り除きます。

これらの操作は、sed または awk の各スクリプトで、またはテキストエディタを使用して実行できます。Makefile はソースファイルから不要なエントリをある程度自動的に削除しますが、これらのファイルを手動で検証し、クリーンアップしてから実行することをお勧めします。

8 すべてのソースファイルのデータが正しい形式になっていることを確認します。

ソースファイルのデータは、それぞれのファイルに適した形式で格納されている必要があります。該当するマニュアルページを参照して、各ファイルが正しい形式になっていることを確認します。

Makefile を準備する

ソースファイルをチェックしてソースファイルディレクトリにコピーしたら、NIS サービスが使用する ndbm 形式のマップにそのソースファイルを変換する必要があります。92 ページの「ypinit によるマスターサーバーの設定」の節で説明しているように、この処理は、マスターサーバーで呼び出されると ypinit によって自動的に行われます。

ypinit スクリプトはプログラム make を呼び出します。このプログラムは、/var/yp ディレクトリに置かれた Makefile を使用します。/var/yp ディレクトリにはデフォルトの Makefile が用意されており、この中には要求された ndbm 形式のマップにソースファイルを変換するためのコマンドが入っています。

デフォルトの Makefile は、そのまま使用することも必要に応じて修正することもできます。(デフォルトの Makefile を修正するときは、将来必要な場合に備えて、必ず最初に修正前の Makefile をコピーして保存するようにしてください。)次に説明する Makefile への修正のうち、必要に応じて 1 つまたは複数を実行します。

- 「デフォルトではないマップ」

  デフォルトではない自分専用のソースファイルを作成して NIS マップに変換する場合は、そのソースファイルを Makefile に追加する必要があります。

- 「DIR の値」

  88 ページの「ソースファイルディレクトリ」で説明しているように、/etc 以外のディレクトリに格納されたソースファイルを Makefile で使用する場合は、Makefile の DIR の値を、使用するディレクトリに変更してください。この値を Makefile で変更するときは行をインデントしないでください。

- 「PWDIR の値」

  /etc 以外のディレクトリに格納された passwd、shadow、adjunct の各ソースファイルを Makefile で使用する場合は、Makefile の PWDIR の値を、使用するディレクトリに変更します。この値を Makefile で変更するときは行をインデントしないでください。

- 「ドメインネームリゾルバ」

  現在のドメインにはないマシンに対して NIS サーバーがドメインネームリゾルバを使用するようにする場合は、`Makefile` の `B=` 行をコメントアウトし、`B=-b` 行のコメントを解除します (有効にする)。

`Makefile` の機能は、`all` の下にリストされるデータベースのそれぞれに対して適切な NIS マップを作成することです。データは `makedbm` で処理され、`mapname.dir` と `mapname.pag` の 2 つのファイルに保存されます。この両ファイルは、マスターサーバーの `/var/yp/`*domainname* ディレクトリに置かれます。

`Makefile` は必要に応じて、`/PWDIR/passwd`、`/PWDIR/shadow`、`/PWDIR/security/passwd.adjunct` の各ファイルから `passwd` マップを構築します。

ypinit によるマスターサーバーの設定

`ypinit` スクリプトは、マスターサーバー、スレーブサーバー、クライアントが NIS を使用するように設定します。また最初に `make` を実行して、マスターサーバー上にマップを作成します。

`ypinit` を使用して新規に NIS マップセットをマスターサーバーに作成する場合は、次の手順に従います。

▼ ypinit を使用してマスターサーバーを設定する方法

1. マスターサーバーで、スーパーユーザーになるか、同等の役割になります。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2. `nsswitch.files` ファイルの内容を `nsswitch.conf` ファイルにコピーします。

   ```
   # cp /etc/nsswitch.files /etc/nsswitch.conf
   ```

3. `/etc/hosts` ファイルまたは `/etc/inet/ipnodes` ファイルを編集して、**NIS** サーバーのそれぞれの名前と **IP** アドレスを追加します。

4. 新しいマップをマスターサーバーに作成します。

   ```
   # /usr/sbin/ypinit -m
   ```

5. `ypinit` によって **NIS** スレーブサーバーになるほかのマシンのリストを求めるプロンプトが表示されたら、作業中のサーバー名と **NIS** スレーブサーバー名を入力します。

6 致命的でないエラーが発生したときにすぐに処理を終了するか、引き続き処理を継続するかを ypinit が尋ねてきたら、**y** と入力します。

y を選択すると、ypinit は最初の問題が発生すると終了します。問題を解決して ypinit を再起動します。ypinit を初めて実行する場合はこの手順に従うようにしてください。処理を継続する場合は、発生する問題をすべて手動で解決してから ypinit を再起動します。

注 – マップファイルの一部が存在しないと、致命的でないエラーが発生することがあります。これは NIS の機能に影響するエラーではありません。マップが自動的に作成されない場合は、必要に応じて手動で追加します。すべてのデフォルトの NIS マップの詳細については、76 ページの「デフォルトの NIS マップ」を参照してください。

---

7 /var/yp/*domainname* ディレクトリ内の既存のファイルを破棄してもよいかどうか ypinit が尋ねてきます。

このメッセージは、NIS が以前に設定されている場合にだけ表示されます。

8 ypinit は、サーバーのリストを作成し終わると make を起動します。

このプログラムは、 /var/yp に置かれた Makefile (デフォルトまたは修正されたもの) に含まれている命令を使用します。make コマンドは、指定したファイルにコメント行があればその行を取り除きます。また、指定したファイルに対して makedbm を実行して適切なマップを作成し、各マップにマスターサーバー名を設定します。

マスターサーバー上で domainname コマンドを実行すると返されるドメイン以外に対するマップの転送を Makefile で行う場合は、ypinit シェルスクリプトの中で make コマンドの変数 DOM に適切なドメイン名を指定して起動すれば、マップを正しいドメインに転送することができます。次のように入力してください。

```
# make DOM=domainname  password
```

このコマンドによって、マスターサーバーが属するドメインではなく目的のドメインに password マップが転送されます。

9 次のように入力してネームサービスとして **NIS** を有効にします。

```
# cp /etc/nsswitch.nis /etc/nsswitch.conf
```

現在のスイッチファイルが、デフォルトの NIS 用スイッチファイルに置き換えられます。このファイルは必要に応じて編集可能です。

複数の NIS ドメインをサポートするマスターサーバー

NIS マスターサーバーは通常、NIS ドメインだけをサポートします。ただし、マスターサーバーを使用して複数のドメインをサポートする場合は、92 ページ

の「ypinit によるマスターサーバーの設定」で説明したように、追加のドメイン用にサーバーを設定するときに手順を若干修正する必要があります。

サーバー上で domainname コマンドを実行します。このコマンドによって返されるドメイン名はサーバーのデフォルトドメインです。92 ページの「ypinit によるマスターサーバーの設定」で説明した手順は、そのドメインへのサービスの設定では正しく機能します。ほかのドメインへのサービスを設定する場合は、ypinit シェルスクリプトを次のように修正する必要があります。

```
# make DOM=correct-domain passwd
```

correct-domain はサービスを設定しているほかのドメインの名前であり、passwd は make のターゲットです。このコマンドによって、マスターサーバーが属するドメインではなく目的のドメインに passwd マップが転送されます。

マスターサーバーでの NIS サービスの開始と停止

マスターサーバーのマップが作成されると、NIS デーモンをマスターサーバーで起動してサービスを開始できます。NIS サービスを有効にすると、ypserv と ypbind がサーバー上で起動されます。クライアントがサーバーの情報を要求すると、ypserv デーモンは NIS マップ内で検索してクライアントからの情報の要求に応答します。ypserv デーモンと ypbind デーモンは 1 単位として管理されます。

サーバー上で NIS サービスを開始または停止するには、次の 3 つの方法があります。

- ブートプロセス中に /usr/lib/netsvc/yp/ypstart スクリプトを自動的に起動する
- コマンド行からサービス管理機能の svcadm enable <*fmri*> コマンドおよび svcadm disable <*fmri*> コマンドを使用する

  SMF の詳細については、svcadm(1M) を参照してください。
- コマンド行から ypstart(1M) コマンドおよび ypstop(1M) コマンドを使用する

NIS サービスを自動的に開始する

ypinit を実行して NIS マスターサーバーを構成し終わると、マシンのブート時に ypstart が自動的に起動され、 ypserv が開始されます。92 ページの「ypinit によるマスターサーバーの設定」を参照してください。

コマンド行から NIS を開始または停止する

サービス管理機能の svcadm コマンド、または ypstart/ypstop コマンドを使用して、コマンド行から NIS を開始および停止します。svcadm コマンドを使用する場

合、サービスのインスタンスを複数実行しているときにのみインスタンス名が必要です。詳細については、86 ページの「NIS とサービス管理機能」、または svcadm(1M)、ypstart(1M)、および ypstop(1M) のマニュアルページを参照してください。

コマンド行から NIS を開始するには、svcadm enable コマンドまたは ypstart コマンドを使用します。

```
# svcadm enable network/nis/server:<instance>
# svcadm enable network/nis/client:<instance>
or
# ypstart
```

注 – 起動後に ypserv が呼び出しに応答できるようになるまでに若干の遅延があるので、プログラムまたはスクリプトの内部から呼び出す場合は、svcadm の実行後に 3 - 5 秒間スリープ状態にしてください。

NIS サービスを停止するには、svcadm disable コマンドまたは ypstop コマンドを実行します。

```
# svcadm disable network/nis/server:<instance>
# svcadm disable network/nis/client:<instance>
or
# ypstop
```

NIS サービスを停止してすぐに再起動するには、svcadm restart コマンドを使用します。

```
# svcadm restart network/nis/server:<instance>
# svcadm restart network/nis/client:<instance>
```

NIS スレーブサーバーの設定

ネットワークは 1 つ以上のスレーブサーバーを持つことができます。スレーブサーバーを持つことで、マスターサーバーが利用できない場合にも NIS サービスを継続して利用できます。

スレーブサーバーを準備する

ypinit を実際に実行してスレーブサーバーを作成する前に、domainname コマンドを NIS スレーブサーバーごとに実行してドメイン名がマスターサーバーと一致していることを確認するようにしてください。

注 – ドメイン名は大文字と小文字を区別します。

ネットワークが正しく機能していることを確認してから、NIS スレーブサーバーを構成してください。特に、rcp を使用して NIS マスターサーバーから NIS スレーブサーバーにファイルを送れるかどうかを確認してください。

スレーブサーバーを設定する

次の手順はスレーブサーバーの設定方法を示しています。

▼ スレーブサーバーを設定する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2. スレーブサーバー上で /etc/hosts ファイルまたは /etc/inet/ipnodes ファイルを編集して、ほかのすべての **NIS** サーバー名と **IP** アドレスを追加します。

3. スレーブサーバー上の /var/yp にディレクトリを変更します。

   注 – まず、新しいスレーブサーバーを NIS クライアントとして構成して、最初にマスターサーバーから NIS マップを入手できるようにします。詳細については、97 ページの「NIS クライアントの設定」を参照してください。

4. スレーブサーバーをクライアントとして初期化します。

   ```
   # /usr/sbin/ypinit -c
   ```

   ypinit コマンドによって、NIS サーバーのリストを求めるプロンプトが表示されます。作業中のローカルマシン (スレーブ) の名前を最初に入力してからマスターサーバーを入力し、そのあとにドメイン内のほかの NIS スレーブサーバーをネットワーク的に近いものから遠いものの順番で入力します。

5. **NIS** クライアントが実行されているかどうかを確認し、必要に応じてクライアントサービスを開始します。

   ```
   # svcs network/nis/client
   STATE          STIME     FMRI
   online         20:32:56  svc:/network/nis/client:default
   ```

svc:/network/nis/client の状態が online と表示される場合、NIS は実行されています。サービスの状態が無効となっている場合、NIS は実行されていません。

a. **NIS クライアントが実行されている場合、クライアントサービスを再起動します。**

```
# svcadm restart network/nis/client
```

b. **NIS クライアントが実行されていない場合、クライアントサービスを開始します。**

```
# svcadm enable network/nis/client
```

6 このマシンをスレーブサーバーとして初期設定します。

```
# /usr/sbin/ypinit -s master
```

master は、既存の NIS マスターサーバーのマシン名です。

この節で説明した手順を、NIS スレーブサーバーとして構成するマシンごとに繰り返します。

▼ スレーブサーバーで NIS を開始する方法

次の手順は、スレーブサーバーで NIS サービスを開始する方法を示しています。

- スーパーユーザーになるか、同等の役割を引き受けます。
  役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

NIS クライアントの設定

ネームサービスとして NIS を使用するようにクライアントマシンを設定するには、次の 2 つの方法があります。

注 – Solaris オペレーティングシステムは、NIS クライアントとネイティブな LDAP クライアントが同一のクライアントマシン上に共存する構成をサポートしません。

- 「ypinit」。NIS を使用するようにクライアントマシンを設定する場合は、マシンに root としてログインして ypinit -c を実行する方法をお勧めします。

```
# ypinit -c
```

NIS サーバーを指定するように求められます。クライアントは NIS サーバーからネームサービス情報を得ます。必要な数だけマスターサーバーやスレーブ

サーバーを指定できます。指定するサーバーはドメイン内のどこにあってもかまいません。クライアントにネットワーク的に近いサーバーから遠いサーバーの順に指定することをお勧めします。

- 「ブロードキャスト方式」。NIS を使用するようにクライアントマシンを設定する旧式の方法です。マシンに root としてログインし、domainname コマンドでドメイン名を設定してから、ypbind を実行します。

  /var/yp/binding/‘domainname‘/ypservers ファイルが存在しない場合、ypstart は NIS クライアントをブロードキャストモードで自動的に起動します (ypbind -broadcast)。

  ```
  # domainname doc.com
  # mv /var/yp/binding/‘domainname‘/ypservers /var/yp/binding/‘domainname‘\
  /ypservers.bak
  # ypstart
  ```

  ypbind を実行すると、NIS サーバーがローカルサブネットで検索されます。NIS サーバーが見つかると、ypbind はそのサーバーにバインドします。この検索を「ブロードキャスト」と呼びます。クライアントのローカルサブネットに NIS サーバーがない場合、ypbind によるバインドは失敗し、クライアントマシンは NIS サービスから名前空間データを入手することができません。

注 – セキュリティーと管理の意味から、クライアントにブロードキャストを使ってサーバーを検索させるのではなく、クライアントの ypservers ファイルでクライアントのバインド先のサーバーを指定してください。ブロードキャストは、ネットワークの速度を落とし、クライアントの速度も落とします。また、異なるクライアントに対して異なるサーバーをリストするため、サーバー負荷の均衡がとれなくなります。

◆ ◆ ◆ 第 6 章

NIS の管理 (手順)

この章では、NIS の管理方法について説明します。この章の内容は次のとおりです。

注 – NIS サービスはサービス管理機能によって管理されます。このサービスに関する有効化、無効化、再起動などの管理アクションは `svcadm` コマンドを使用して実行できます。NIS で SMF を使用する場合の詳細については、86 ページの「NIS とサービス管理機能」を参照してください。SMF の概要については、『Solaris のシステム管理 (基本編)』の第 18 章「サービスの管理 (概要)」を参照してください。また、詳細については、`svcadm(1M)` および `svcs(1)` のマニュアルページを参照してください。

NIS サービスの開始および停止は、`ypstart` および `ypstop` コマンドを使用しても行えます。詳細については、`ypstart(1M)` および `ypstop(1M)` のマニュアルページを参照してください。

パスワードファイルと名前空間のセキュリティー

セキュリティーの関係上、次のガイドラインに従ってください。

- マスターサーバーの NIS マップへのアクセスは制限します。
- 未許可アクセスを防止するためには、NIS パスワードマップの作成に使用されたファイルに root エントリを含めないでください。したがって、root エントリをこのパスワードファイルから削除して、このパスワードファイルをマスターサーバーの /etc ディレクトリ以外のディレクトリにおく必要があります。このディレクトリへの未許可アクセスは、防止しなければなりません。

たとえば、マスターサーバーのパスワード入力ファイルは、別のファイルへのリンクではなく Makefile に指定されている限り、/var/yp などのディレクトリに存在するか選択されたディレクトリに存在します。サービス管理機能または ypstart スクリプトを使用して NIS サービスを開始する場合、Makefile に指定された構成に従って適切なディレクトリオプションが設定されます。

注 – この NIS 実装では、NIS パスワードマップを作成するための入力として、旧 Solaris 1 バージョンの passwd ファイルのフォーマットに加え、Solaris 2 の passwd ファイルと shadow ファイルのフォーマットも使用できます。

NIS ユーザーの管理

この節では、ユーザーパスワードの設定、NIS ドメインへの新しいユーザーの追加、ネットグループ (netgroups) へのユーザーの割り当てについて説明します。

▼ NIS ドメインに新しい NIS ユーザーを追加する方法

1 マスター NIS サーバーで、スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2 useradd コマンドで新しいユーザーのログイン ID を作成します。

```
# useradd userID
```

userID は新しいユーザーのログイン ID です。このコマンドは、NIS マスターサーバー上の /etc/passwd ファイルと /etc/shadow ファイルにエントリを作成します。

3 新しいユーザーの初期パスワードを作成します。

新しいユーザーがログインするための初期パスワードを作成するには、passwd コマンドを実行します。

```
# passwd userID
```

userID は新しいユーザーのログイン ID です。このユーザーに割り当てるパスワードを入力するようにプロンプトが表示されます。

この手順が必要になるのは、 useradd コマンドで作成されたパスワードエントリがロックされ、新しいユーザーがログインできないからです。初期パスワードを指定することで、このパスワードエントリのロックが解除されます。

4 必要に応じて、マスターサーバーの passwd マップ入力ファイルに新しいエントリをコピーします。

マスターサーバー上のマップソースファイルは、/etc 以外のディレクトリにあります。新しい行を /etc/passwd ファイルと /etc/shadow ファイルからマスターサーバー上の passwd マップ入力ファイルにコピーします。詳細については、100 ページの「パスワードファイルと名前空間のセキュリティー」を参照してください。

たとえば、新しいユーザー brown を追加する場合、/etc/passwd ファイルから passwd 入力ファイルにコピーする行は次のようになります。

```
brown:x:123:10:User brown:/home/brown:/bin/csh:
```

/etc/shadow からコピーされる brown 行は次のようになります。

```
brown:W12345GkHic:6445::::::
```

5 パスワード入力ファイルが格納されているディレクトリが Makefile で正しく指定されていることを確認します。

6 必要に応じて、/etc/passwd ファイルと /etc/shadow ファイルから新しいユーザーのエントリを削除します。

セキュリティー上の理由から、NIS マスターサーバーの /etc/passwd ファイルと /etc/shadow ファイルにユーザーエントリを保持しないようにしてください。ほかのディレクトリに存在する NIS マップソースファイルに新しいユーザーのエントリをコピーしたあと、マスターサーバー上で userdel コマンドを使用して新しいユーザーを削除します。

たとえば、マスターサーバーの /etc ファイルから新しいユーザー brown を削除するには次のように入力します。

```
# userdel brown
```

userdel の詳細については、userdel のマニュアルページを参照してください。

7 **NIS の** passwd **マップを更新します。**

マスターサーバー上の passwd 入力ファイルを更新したあと、ソースファイルが存在するディレクトリで make を実行して、passwd マップを更新します。

```
# userdel brown
# cd /var/yp
# /usr/ccs/bin/make passwd
```

8 **新しいユーザーのログイン ID に割り当てられた初期パスワードを新しいユーザーに通知します。**

ログイン後、新しいユーザーはいつでも passwd を実行して別のパスワードに変更できます。

ユーザーパスワードの設定

ユーザーは passwd を実行してパスワードを変更します。

% **passwd** *username*

ユーザーがパスワードを変更する前に、NIS 管理者はマスターサーバー上で rpc.yppasswdd デーモンを起動してパスワードファイルを更新する必要があります。

rpc.yppasswdd デーモンは、マスターサーバー上で自動的に起動します。rpc.yppasswdd に -m オプションが指定された場合は、ファイルが更新されるとすぐ /var/yp の make が実行されます。passwd ファイルが更新されるたびにこの make が実行されることを回避したい場合は、ypstart スクリプトの rpc.yppasswd コマンドから -m オプションを削除して、crontab ファイルで passwd マップの転送を制御してください。

注 – rpc.yppasswd - m コマンドのあとに引数を指定するべきではありません。別の動作のために ypstart スクリプトファイルを編集することは可能ですが、-m オプションを任意に削除すること以外の変更をこのファイルに加えることは望ましくありません。すべてのコマンドおよびデーモンは、適切なコマンド行パラメータのセットが存在するこのファイルで起動されます。このファイルを編集する場合は、rpc.yppasswdd コマンドの編集では特に注意してください。passwd.adjunct ファイルに明示的コールを追加する場合は、パスを $PWDIR/security/passwd.adjunct と正確に指定しなければなりません。正確に指定しないと、不適切な処理が行われます。

NIS ネットグループ

NIS ネットグループは、NIS 管理者が管理目的のために定義するユーザーまたはマシンのグループ (集合) です。たとえば、次のようなネットグループを作成できます。

- 特定マシンにアクセスできる一群のユーザーを定義する

- 特定のファイルシステムにアクセスできる一群の NFS クライアントマシンを定義する
- 特定の NIS ドメインのすべてのマシンに対して管理者権限を持つ一群のユーザーを定義する

各ネットグループには、1 つのネットグループ名が与えられます。ネットグループはアクセス権を直接設定しません。代わりに、ユーザー名またはマシン名が一般に使用される場所ではネットグループ名がほかの NIS マップで使用されます。たとえば、`netadmins` というネットワーク管理者ネットグループを作成したと仮定します。`netadmins` ネットグループのすべてのメンバーに特定マシンへのアクセス権を与えるには、そのマシンの `/etc/passwd` ファイルに `netadmin` エントリを追加するだけで、ネットグループ名を `/etc/netgroup` ファイルに追加して、NIS グループマップに追加することもできます。ネットグループの使い方の詳細については、`netgroup(4)` のマニュアルページを参照してください。

NIS が使用されているネットワーク上では、NIS マスターサーバー上の `netgroup` 入力ファイルを使用して、次の 3 つのファイルが生成されます。`netgroup`、`netgroup.byuser`、および `netgroup.byhost` です。`netgroup` マップには、`netgroup` 入力ファイルの基本情報が入っています。ほかの 2 つの NIS マップには、マシンまたはユーザーが指定されるとネットグループ情報の検索が迅速に行われるフォーマットで情報が入っています。

`netgroup` 入力ファイルのエントリのフォーマットは、 *name ID* です。*name* はネットグループ名であり、*ID* は、ネットグループに属しているマシンまたはユーザーを示します。ネットグループの ID (メンバー) は、コンマで区切っていくつでも指定できます。たとえば、3 つのメンバーが存在するネットグループを作成する場合、`netgroup` 入力ファイルエントリのフォーマットは、*name ID*, *ID*, *ID* となります。`netgroup` 入力ファイルエントリのメンバーID のフォーマットは次のようになります。

```
([-|machine], [-|user], [domain])
```

machine はマシン名、*user* はユーザー ID 、*domain* はマシンまたはユーザーの NIS ドメインです。「ドメイン」エレメントは任意指定ですが、ほかの NIS ドメインのマシンまたはユーザーを示す場合には必ず指定します。各エントリでは「マシン」エレメントと「ユーザー」エレメントは必須ですが、ダッシュ (-) は空であることを示すために使用されます。エントリでは、「マシン」エレメントと「ユーザー」エレメントの関係を示す必要はありません。

`netgroup` 入力ファイルの 2 つのサンプルエントリを次に示します。これらの各サンプルエントリでは、`admins` という名前のネットグループが作成されます。これらの各ネットグループは、遠隔ドメイン `sales` に存在するユーザー `hauri` と `juanita`、およびマシン `altair` と `sirius` で構成されます。

```
admins (altair, hauri), (sirius,juanita,sales)
```

```
admins (altair,-), (sirius,-), (-,hauri), (-,juanita,sales)
```

さまざまなプログラムでは、ログイン、遠隔マウント、遠隔ログイン、および遠隔シェル作成時に NIS ネットグループマップを使用してアクセス権のチェックを行います。さまざまなプログラムとは、`mountd`、`login`、`rlogin`、`rsh` などです。`login` コマンドは、`passwd` データベース内でネットグループ名を見つけた場合に、ネットグループマップでユーザー分類を調べます。`mountd` デーモンは、`/etc/dfs/dfstab` ファイル内でネットグループ名を見つけた場合に、ネットグループマップでマシン分類を調べます。`rlogin` と `rsh` (`ruserok` インタフェースを使用する任意のプログラム) は、`/etc/hosts.equiv` または `.rhosts` ファイル内でネットグループ名を見つけた場合に、ネットグループマップでマシン分類とユーザー分類の両方を調べます。

ネットワークに新しい NIS ユーザーまたはマシンを追加する場合は、`netgroup` 入力ファイルの該当ネットグループに追加してください。次に、`make` でネットグループマップを作成し、これを `yppush` コマンドですべての NIS サーバーに転送してください。ネットグループおよびネットグループ入力ファイル構文の使い方の詳細については、`netgroup(4)` のマニュアルページを参照してください。

NIS マップに関する作業

この節には次の情報が含まれます。

マップ情報の取得

マップ情報は、`ypcat`、`ypwhich`、`ypmatch` コマンドを使っていつでも取得できます。次の例では、*mapname* はマップの正式名とニックネーム (存在する場合) の両方を意味します。

マップのすべての値を表示するには、次のように入力します。

```
% ypcat mapname
```

マップのキーと値 (存在する場合) の両方を表示するには、次のように入力します。

```
% ypcat -k mapname
```

マップのすべてのニックネームを表示するには、次のいずれかのコマンドを入力します。

```
% ypcat -x
% ypmatch -x
% ypwhich -x
```

使用可能なすべてのマップとそのマスターサーバーを表示するには、次のように入力します。

```
% ypwhich -m
```

特定のマップのマスターサーバーを表示するには、次のように入力します。

```
% ypwhich -m mapname
```

キーをマップのエントリと照合するには、次のように入力します。

```
% ypmatch key mapname
```

検索している項目がマップのキーでない場合は、次のように入力します。

```
% ypcat mapname | grep item
```

item は検索している情報です。ほかのドメインに関する情報を取得するには、これらのコマンドの `-d domainname` オプションを指定します。

デフォルト以外のドメインの情報を要求するマシンが、そのドメインに対するバインドを持っていない場合、`ypbind` は `/var/yp/binding/`*domainname*`/ypservers` ファイルを参照して、そのドメインのサーバーリストを検索します。このファイルが存在しない場合、ypbind は RPC 同報通信を送出してサーバーを検索します。この場合、検索先であるドメインのサーバーは要求元マシンと同じサブネットに存在している必要があります。

マップのマスターサーバーの変更

選択されたマップのマスターサーバーを変更するには、まず新しい NIS マスターサーバー上にマップを構築する必要があります。古いマスターサーバー名は既存のマップにキーと値のペアとして発生するので (このペアは `makedbm` で自動的に挿入される)、`ypxfr` でマップを新しいマスターサーバーにコピーしたり、コピーを新しいマスターサーバーに転送するだけでは不十分です。キーと新しいマスターサーバー名との対応づけをし直す必要があります。マップに ASCII ソースファイルが存在する場合は、このファイルを新しいマスターサーバーにコピーします。

▼ マップのマスターサーバーを変更する方法

1. 新しいマスターで、スーパーユーザーになるか、同等の役割になります。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2 ディレクトリを変更します。

```
newmaster# cd /var/yp
```

3 作成するマップを指定する前に、Makefile にこの新しいマップのエントリが必要です。新しいマップのエントリがない場合は、最初に、sites.byname というマップを使用して Makefile を編集します。

4 マップを更新または再作成するには、次のように入力します。

```
newmaster# make sites.byname
```

5 古いマスターサーバーが **NIS** サーバーとして残っている場合は、古いマスターサーバーに遠隔ログイン (rlogin) してから、Makefile を編集します。sites.byname を作成した Makefile 内のセクションをコメントアウトして、このセクションで **sites.byname** が再び作成されないようにします。

6 sites.byname だけが ndbm ファイルとして存在している場合は、新しいマスターサーバー上に作成し直します。まず任意の **NIS** サーバーからコピーを分解し、次に makedbm を使ってそれを実行します。

```
newmaster# cd /var/yp
newmaster# ypcat sites.byname | makedbm -domain-/sites.byname
```

新しいマスターサーバー上でマップが作成されたら、そのコピーをほかのスレーブサーバーに送信する必要があります。この場合、yppush を使用しないでください。yppush を使用すると、スレーブサーバーは新しいマスターサーバーからではなく古いマスターサーバーから新しいコピーを取得します。このような動作を回避するには、一般にマップのコピーを新しいマスターサーバーから古いマスターサーバーに送り返すという方法が使用されます。これを行うには、古いマスターサーバーでスーパーユーザーになるか、同等の役割になり、次のように入力します。

```
oldmaster# /usr/lib/netsvc/yp/ypxfr -h newmaster sites.byname
```

これで、yppush を使用できます。スレーブサーバーは古いマスターサーバーを現行のマスターサーバーとして認識しているので、マップの現行バージョンを古いマスターサーバーから取得しようとします。クライアントがこの処理を行うときは、新しいマスターサーバーが現行のマスターサーバーとして指定されている新しいマップを取得します。

この方法が失敗した場合は、各 NIS サーバーの root としてログインして上記の ypxfr コマンドを実行できます。

構成ファイルの変更

NIS は設定ファイルを正確に構文解析します。このため NIS 管理は容易になりますが、設定ファイルおよび構成ファイルにおける変更により、NIS の動作は影響を受けます。

次のファイルを変更する場合は、この節の手順を使用します。

- `/var/yp/Makefile`。このファイルは、サポートされているマップを追加または削除するために使用します
- `/etc/resolv.conf`。このファイルを追加または削除することで、DNS 転送が可能または不可になります
- `$PWDIR/security/passwd.adjunct`。このファイルを追加したり削除したりすることで、C2 セキュリティーが可能または不可になります (`$PWDIR` は、`/var/yp/Makefile` で定義される)

▼ 構成ファイルを更新する方法

NIS のマップまたはマップソースファイルを更新する場合は、NIS を停止および起動する必要はありません。

次の点に注意してください。

- NIS マスターサーバーからマップまたはソースファイルを削除しても、スレーブサーバー上の対応するマップまたはソースファイルは自動的には削除されません。スレーブサーバー上の対応するマップまたはソースファイルの削除は、NIS 管理者が手作業で行う必要があります。
- 新しいマップは、自動的には既存のスレーブサーバーに転送されません。新しいマップを既存のスレーブサーバーに転送するには、NIS 管理者がそのスレーブサーバーで `ypxfr` を実行してください。

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2. NIS サーバーを停止します。

   ```
   # svcadm disable network/nis/server
   ```

3. 必要に応じてファイルを変更します。

4. NIS サーバーを起動します。

   ```
   # svcadm enable network/nis/server
   ```

Makefile の更新と使用

`/var/yp` で提供されたデフォルトの `Makefile` を更新することにより、NIS 管理者のニーズを満たすことができます。マップを追加したり削除したり、一部のディレクトリの名前を変更できます。

ヒント – 将来の参照のために、変更していない、元の Makefile のコピーを保存しておいてください。

Makefile での作業

新しい NIS マップを追加するには、マップの ndbm ファイルのコピーをドメインに存在する各 NIS サーバーの /var/yp/*domainname* ディレクトリに転送する必要があります。通常これは、Makefile によって行われます。どの NIS サーバーがマップのマスターサーバーであるかを決定したら、マップを容易に作成し直せるようにマスターサーバーの Makefile を更新してください。異なるサーバーを異なるマップのマスターサーバーとして設定することも可能ですが、このようにするとたいていの場合、管理上の混乱を招きます。したがって、1 つのサーバーだけをすべてのマップのマスターサーバーとして設定するようにしてください。

一般に、人が読めるテキストファイルは、makedbm に対する入力として適したものにするために awk 、 sed、grep でフィルタリングされます。デフォルトの Makefile を参照してください。make コマンドの概要については、make(1S) のマニュアルページを参照してください。

make が認識する依存性の作成方法を決定する際には、Makefile にすでに備わっているメカニズムを使用してください。make では、依存ルール内の行の始まりにタブが存在するか否かが重要であることに注意してください。ほかの設定が正しくても、タブが存在しないというだけでエントリが無効になることがあります。

Makefile にエントリを追加する場合は、次の作業を行ってください。

- データベース名を all ルールに追加する
- time ルールを作成する
- データベースのルールを追加する

たとえば、Makefile をオートマウンタ入力ファイルで動作させるには、auto_direct.time および auto_home.time マップを NIS データベースに追加する必要があります。

これらのマップを NIS データベースに追加するには、Makefile を修正する必要があります。

Makefile のマクロおよび変数の変更

Makefile の先頭で定義されている変数の設定値を変更するには、等号 (=) の右側の値を変更します。たとえば、マップへの入力として /etc に存在するファイルではなく、別のディレクトリに存在するファイル (たとえば /var/etc/domainname など) を使用する場合は、DIR を DIR=/etc から DIR=/var/etc/*domainname* に変更します。また、PWDIR を PWDIR=/etc から PWDIR=/var/etc/*domainname* に変更します。

変数は次のとおりです。

- *DIR*=。passwd と shadow を除くすべての NIS 入力ファイルが存在するディレクトリ。デフォルト値は /etc です。マスターサーバーの /etc ディレクトリのファイルを NIS 入力ファイルとして使用することは望ましくないので、この値は変更しなければなりません。
- *PWDIR*=。NIS 入力ファイル passwd と shadow が存在するディレクトリ。マスターサーバーの /etc ディレクトリのファイルを NIS 入力ファイルとして使用することは望ましくないので、この値は変更しなければなりません。
- *DOM*=。NIS ドメイン名。DOM のデフォルト値は、domainname コマンドで設定されます。ただし、大部分の NIS コマンドでは現在のマシンのドメイン (現在のマシンの /etc/defaultdomain ファイルに設定されている) が使用されます。

Makefile エントリの変更

次の手順では、Makefile にデータベースを追加したり削除したりする方法を説明します。

▼ 特定のデータベースを使用するために Makefile を変更する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2. all という語で始まる行に、追加したいデータベース名 (1 つまたは複数) を追加します。

   ```
   all: passwd group hosts ethers networks rpc services protocols \
       netgroup bootparams aliases netid netmasks \
       audit_user auth_attr exec_attr prof_attr \
     auto_direct auto_home auto_direct.time auto_home.time
   ```

   エントリの順序は任意ですが、継続行の始まりの空白はスペースではなくタブにしてください。

3. Makefile の終わりに次の行を追加します。

   ```
   auto_direct: auto_direct.time
   auto_home: auto_home.time
   ```

4. ファイル中央に auto_direct.time エントリを追加します。

   ```
   auto_direct.time: $(DIR)/auto_direct
    @(while read L; do echo $$L; done < $(DIR)/auto_direct
    $(CHKPIPE)) | \ (sed -e "/^#/d" -e "s/#.*$$//" -e "/^ *$$/d"
    $(CHKPIPE)) | \ $(MAKEDBM) - $(YPDBDIR)/$(DOM)/auto_direct;
    @touch auto_direct.time;
    @echo "updated auto_direct";
   ```

```
@if [ ! $(NOPUSH) ]; then $(YPPUSH) auto_direct; fi
@if [ ! $(NOPUSH) ]; then echo "pushed auto_direct"; fi
```

次に、各引数について説明します。

- CHKPIPE は、次のコマンドに結果を渡す (パイピングする) 前に、パイプ (|) の左側の動作が正しく行われたことを確認します。パイプの左側の動作が正しく行われなかった場合は、「NIS make terminated」というメッセージが表示されてプロセスは終了します。
- NOPUSH は、makefile が yppush を呼び出して新しいマップをスレーブサーバーに転送することを防止します。NOPUSH が設定されていない場合は、転送は自動的に行われます。

継続行の始まりにある while ループは、バックスラッシュで拡張された行を入力ファイルから削除するためのものです。sed スクリプトはコメント行と空行を削除します。

ほかのすべてのオートマウンタマップ (auto_home やほかのデフォルトでないマップなど) でも、同じ手順が必要となります。

5 make を実行します。

```
# make mapname
```

mapname は、作成するマップの名前です。

▼ データベースを削除するために Makefile を変更する方法

Makefile で特定データベースのマップを作成しない場合は、Makefile を次のように編集してください。

1 all ルールからデータベース名を削除します。

2 削除するデータベースのデータベースルールを削除またはコメントアウトします。

たとえば、hosts データベースを削除するには、hosts.time エントリを削除します。

3 **time** ルールを削除します。

たとえば、hosts データベースを削除するには、hosts: hosts.time エントリを削除します。

4 マスターサーバーとスレーブサーバーからマップを削除します。

既存のマップの更新

NIS のインストール終了後、頻繁に更新しなければならないマップとまったく更新する必要がないマップがあることに気づくかもしれません。たとえば、passwd.byname マップは、大企業のネットワークで頻繁に更新されることがありますが、auto_master マップはほとんど更新されません。

76 ページの「デフォルトの NIS マップ」で説明されているように、デフォルトの NIS マップのデフォルトの位置は、/var/yp/*domainname* のマスターサーバー上です。*domainname* は NIS ドメインの名前です。マップを更新する必要がある場合は、マップがデフォルトのマップか否かによって 2 つの更新手順のどちらかを使用できます。

- デフォルトのマップは、ネットワークデータベースから ypinit で作成されたデフォルトセットに存在するマップです。
- デフォルトでないマップは次のいずれかです。
  - ベンダーから購入したアプリケーションに付属のマップ
  - ユーザーサイト用に特別に作成されたマップ
  - テキスト以外のファイルから作成されたマップ

この節では、さまざまな更新ツールの使用方法について説明します。実際にはこれらの更新ツールはシステム起動後に、デフォルトでないマップを追加する場合または一群の NIS サーバーを変更する場合にだけ使用します。

▼ デフォルトセットに付いているマップを更新する方法

デフォルトセットに付いているマップを更新する場合は、次の手順に従います。

1. マスターサーバー上でスーパーユーザーになります。
   必ずマスターサーバー上だけで NIS マップを更新します。
2. 更新するマップのソースファイルを編集します (このファイルが /etc に存在しているか、選択されたほかのディレクトリに存在しているかは問題ではありません)。
3. 次のように入力します。

   ```
   # cd /var/yp
   # make mapname
   ```

   make コマンドは、対応するファイルに対して NIS 管理者が行った変更に従ってマップを更新します。make コマンドはまた、これらの変更をほかのサーバーに伝播します。

更新したマップを管理する

以降の節では、デフォルトセットで提供されているマップの更新完了後に実行する手順について説明します。

NIS マップを伝播する

マップが更新されると、Makefile は yppush を使用して新しいマップをスレーブサーバーに伝播します (ただし、NOPUSH が Makefile に設定されている場合を除く)。これは、ypserv デーモンに通知してマップ転送要求を送ることで実行されます。次に、スレーブサーバー上の ypserv デーモンが ypxfr プロセスを起動し、マスターサーバー上の ypxfrd デーモンに連絡します。いくつかの基本検査 (マップが実際に更新されているかどうかの確認など) が行われてマップが転送されます。そのあと、スレーブサーバー上の ypxfr が、転送が成功したかどうかを yppush プロセスに通知します。

注 – 上記手順は、新しく作成されたマップがスレーブサーバー上に存在しない場合は動作しません。スレーブサーバー上で ypxfr を実行して、新しいマップをスレーブサーバーに転送する必要があります。

マップの伝播は失敗することがありますが、失敗した場合は ypxfr を使って手動で新しいマップ情報を転送する必要があります。ypxfr は、2 つの方法で使用できます。1 つは root の crontab ファイルを定期的に使用する方法であり、もう 1 つはコマンド行から対話形式で使用する方法です。これらの方法については、以降の節で説明します。

cron を使ってマップ転送を行う

マップの更新頻度はマップによってそれぞれ異なります。たとえば、デフォルトのマップである protocols.byname やデフォルトでないマップの auto_master など一部のマップは何ヶ月も更新されないことがありますが、passwd.byname など一部のマップは 1 日に数回更新されることがあります。crontab コマンドでマップ転送をスケジュールすると、個々のマップに対して特定の更新時間を設定できます。

マップに適切な頻度で ypxfr を定期的に実行するには、各スレーブサーバー上の root の crontab ファイルに、該当する ypxfr エントリを入れる必要があります。ypxfr は、マスターサーバー上のコピーがローカルのコピーより新しい場合に限り、マスターサーバーと連絡をとりマップを転送します。

注 – デフォルトの m オプションが指定されている -rpc.yppasswdd をマスターサーバー上で実行すると、どこかで yp パスワードが変更されるたびに、 passwd デーモンが make を実行して passwd マップを作成し直します。

cron と ypxfr でシェルスクリプトを使用する

NIS 管理者は、各マップに対する crontab エントリを個々に作成するという方法ではなく、root の crontab コマンドでシェルスクリプトを実行してすべてのマップを定期的に更新するという方法を使用することもできます。マップ更新シェルスクリプトのサンプルは、/usr/lib/netsvc/yp ディレクトリに入っています。スクリプト名は、ypxfr_1perday、ypxfr_1perhour、ypxfr_2perday です。これらのシェルスクリプトを更新または置換することによって、容易にサイト要件に適合させることができます。例 6–1 は、デフォルトの ypxfr_1perday シェルスクリプトを示しています。

例 6–1 ypxfr_1perday シェルスクリプト

```
#! /bin/sh
#
# ypxfr_1perday.sh - Do daily yp map check/updates
PATH=/bin:/usr/bin:/usr/lib/netsvc/yp:$PATH
export PATH
# set -xv
ypxfr group.byname
ypxfr group.bygid
ypxfr protocols.byname
ypxfr protocols.bynumber
ypxfr networks.byname
ypxfr networks.byaddr
ypxfr services.byname
ypxfr ypservers
```

このシェルスクリプトは、root の crontab が毎日実行される場合はマップを 1 日に 1 回更新します。NIS 管理者は、1 週間に 1 回、1 ヶ月に 1 回、または 1 時間に 1 回などといった頻度でマップを更新するスクリプトを使用できますが、マップを頻繁に伝播するとパフォーマンスが低下するので注意してください。

NIS ドメインに対して構成された各スレーブサーバー上で同じシェルスクリプトを root で実行してください。各サーバー上の実行時間を変更して、マスターサーバーが動作不能にならないようにしてください。

特定のスレーブサーバーからマップを転送する場合は、シェルスクリプトの ypxfr の -h *machine* オプションを使用してください。シェルスクリプトに記述するコマンドの構文は、次のとおりです。

```
# /usr/lib/netsvc/yp/ypxfr -h machine [ -c ] mapname
```

machine は転送するマップが存在するサーバー名です。*mapname* は要求されたマップ名です。マシンを指定することなく -h オプションを指定すると、ypxfr はマス

ターサーバーからマップを取得しようとします。ypxfr 実行時に ypserv がローカルに実行されていない場合は、ypxfr がローカル ypserver に現在のマップ要求の取消しを送信しないよう、-c フラグを使用してください。

-s *domain* オプションを使用すると、別のドメインからローカルドメインにマップを転送できます。これらのマップは、各ドメインにおいて同じでなければなりません。たとえば、2 つのドメインが同じ services.byname マップおよび services.byaddr マップを共有することがあります。また、より細かい制御を行うための rcp または rdist を使用してファイルを複数のドメインに転送することもできます。

ypxfr を直接起動する

2 番目の方法である ypxfr の起動とは、ypxfr をコマンドとして実行することです。一般に、ypxfr をコマンドとして実行するのは例外的状況においてだけです。たとえば、一時的に NIS サーバーを設定して試験環境を作成する場合や、動作不能になっていた NIS サーバーをほかのサーバーと迅速に整合させようとする場合などです。

ypxfr のアクティビティーを記録する

ypxfr が試みた転送およびその転送結果は、ログファイルに記録されます。/var/yp/ypxfr.log というファイルが存在する場合は、転送結果はこのファイルに記録されます。このログファイルのサイズには制限がありません。このログファイルのサイズが無限に大きくなることを防止するには、ときどき次のように入力してこのログファイルを空にしてください。

```
# cd /var/yp
# cp ypxfr.log ypxfr.log.old
# cat /dev/null > /var/yp/ypxfr.log
```

これらのコマンドは、crontab で 1 週間に 1 回実行させることができます。記録を取らないようにするには、ログファイルを削除してください。

デフォルトでないマップの更新

デフォルトでないマップを更新する場合は、次の手順に従います。

1. 対応するテキストファイルを作成または編集します。
2. 新しいマップまたは更新されたマップを作成 (または再作成) します。マップ作成には 2 つの方法があります。
   - Makefile を使用する方法。デフォルトでないマップを作成するには、この方法をお勧めします。Makefile にマップのエントリが存在する場合は、make *name* を実行します (*name* は作成するマップ名)。Makefile にマップのエントリが存在しない場合は、107 ページの「Makefile の更新と使用」を参照してエントリを作成してください。

- /usr/sbin/makedbm プログラムを使用する方法。このコマンドの詳細については、makedbm(1M) のマニュアルページに説明されています。

デフォルトでないマップを makedbm で更新する

入力ファイルが存在しない場合は、makedbm でマップを更新する方法は 2 つあります。

- makedbm -u の出力先を一時ファイルに変更し、一時ファイルを更新して更新済みの一時ファイルを makedbm の入力として使用します。
- makedbm -u の出力を makedbm に渡されるパイプライン内で動作させます。分解されたマップを awk、sed、または cat で更新できる場合は、この方法をお勧めします。

テキストファイルからマップを新たに作成する

テキストファイル /var/yp/mymap.asc がマスターサーバー上のエディタまたはシェルスクリプトで作成されていると仮定します。この場合、このファイルから NIS マップを作成し、作成された NIS マップを *homedomain* サブディレクトリに入れるには、マスターサーバー上で次のように入力してください。

```
# cd /var/yp
# makedbm mymap.asc homedomain/mymap
```

mymap マップは現在、マスターサーバーの *homedomain* ディレクトリに存在しています。この新しいマップをスレーブサーバーに転送するには、ypxfr を実行してください。

ファイルをベースとしたマップにエントリを追加する

mymap へのエントリの追加は簡単です。まず、テキストファイル /var/yp/*mymap*.asc を更新します。対応するテキストファイルを更新しないで実際の dbm ファイルを更新した場合は、更新内容が失われます。次に、上記のように makedbm を実行してください。

標準入力からマップを作成する

オリジナルのテキストファイルが存在しない場合は、キーボードから makedbm に次のように入力して NIS マップを作成します (最後に Control + D を入力します)。

```
ypmaster# cd /var/yp
ypmaster# makedbm -homedomain-/mymapkey1 value1  key2 value2  key3 value3
```

標準入力から作成されたマップを更新する

あとでマップを更新する必要がある場合は、makedbm でマップを分解して一時的な中間テキストファイルを作成できます。マップを分解して一時ファイルを作成するには、次のように入力します。

```
% cd /var/yp
% makedbm -u homedomain/mymap > mymap.temp
```

作成される一時ファイル *mymap*.temp には、1 行につき 1 つのエントリが存在します。このファイルは、任意のテキストエディタで必要に応じて編集できます。

マップを更新するには、次のように入力して更新後の一時ファイルの名前を makedbm に指定します。

```
% makedbm mymap.temp homedomain/mymap
% rm mymap.temp
```

次に root になり、次のように入力してマップをスレーブサーバーに伝播します。

```
# yppush mymap
```

ここまでは makedbm でマップを作成する方法について説明してきましたが、実際に必要な作業はほとんど、ypinit と Makefile で行うことができます。ただし、システム起動後にデフォルトでないマップをデータベースに追加したり NIS サーバーセットを変更したりしない場合に限ります。

/var/yp の Makefile を使用してもほかの手順を使用しても、最終目的は同じです。正しく作成された dbm ファイルの新しいペアをマスターサーバー上の maps ディレクトリに配置しなければなりません。

スレーブサーバーの追加

NIS の実行後、ypinit に指定された初期リストに含まれていなかった NIS スレーブサーバーを必要に応じて作成します。

NIS スレーブサーバーを追加する場合は、次の手順に従います。

▼ スレーブサーバーを追加する方法

1. マスターサーバーで、スーパーユーザーになるか、同等の役割になります。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2. NIS ドメインディレクトリに移動します。

   ```
   # cd /var/yp/domainname
   ```

3. ypservers ファイルを分解します。

   ```
   # makedbm -u ypservers >/tmp/temp_file
   ```

   makedbm コマンドは、ypservers を ndbm フォーマットから一時 ASCII ファイル、/tmp/temp_file に変換します。

4. テキストエディタで /tmp/temp_file ファイルを編集します。つまり、新しいスレーブサーバー名をサーバーリストに追加します。このあと、**/tmp/temp_file** ファイルを保存し、閉じます。

5. 入力ファイルに temp_file を指定し、出力ファイルに ypservers を指定して、makedbm コマンドを実行します。

   ```
   # makedbm /tmp/temp_file ypservers
   ```

   makedbm は、ypservers を変換して ndbm フォーマットに戻します。

6. スレーブサーバーで次のように入力して、ypservers マップが正しいことを確認します (ypservers の **ASCII** ファイルは存在しないため)。

   ```
   slave3# makedbm -u ypservers
   ```

   makedbm コマンドは、画面に ypservers の各エントリを表示します。

   注 – ypservers にマシン名が存在しない場合は、ypservers はマップファイルの更新を受信しません。これは、yppush がこのマップでスレーブサーバーリストを調べるからです。

   ---

7. 新しい **NIS** スレーブサーバーで、スーパーユーザーになるか、同等の役割になります。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

8 新しいスレーブサーバーの **NIS** ドメインディレクトリを設定します。

マスターサーバーから NIS マップセットをコピーし、NIS クライアントを起動します。`ypinit` コマンドを実行 (起動) したら、プロンプトに従って、優先順に NIS サーバーを指定してください。

```
slave3# cd /var/yp
slave3# ypinit -c
slave3# svcadm enable network/nis/client
```

9 このマシンをスレーブサーバーとして初期設定します。

```
slave3# /usr/sbin/ypinit -s ypmaster
```

ypmaster は、既存の NIS マスターサーバーのマシン名です。

10 **NIS** クライアントとして実行されているマシンを停止します。

```
# svcadm disable network/nis/client
```

11 **NIS** スレーブサービスを開始します。

```
# svcadm enable network/nis/server
```

C2 セキュリティーが装備されている NIS の使用

`$PWDIR/security/passwd.adjunct` ファイルが存在する場合は、C2 セキュリティーが自動的に起動されます。`$PWDIR` は `/var/yp/Makefile` で定義されます。C2 セキュリティーモードでは、`passwd.adjunct` ファイルを使って `passwd.adjunct` NIS マップが作成されます。C2 セキュリティーモードでは、`passwd.adjunct` ファイルと `shadow` ファイルの両方を使用してセキュリティーを管理できます。`passwd.adjunct` ファイルは、次のように入力した場合にだけ処理されます。

```
# make passwd.adjunct
```

`make passwd` コマンドは、NIS 管理者が C2 セキュリティーモードで `make` を手動で実行した場合に passwd マップのみを処理します。passwd.adjunct マップは処理されません。

特定の NIS サーバーへのバインド

指定した NIS サーバーにバインドするには、次の手順に従います。詳細は、`ypinit(1M)`、`ypstart(1M)`、および `svcadm(1M)` のマニュアルページを参照してください。

1. NIS サーバーのホスト名と IP アドレスを `/etc/hosts` ファイルに追加します。
2. `domainname` コマンドを実行して、`/etc/defaultdomain` ファイルを生成します。

   ```
   # /usr/bin/domainname name-of-NIS-domain
   ```

3. NIS サーバーホスト名を要求します。

```
# /usr/sbin/ypinit -c
Server name:      Type the NIS server hostname
```

4. 次のいずれかの手順を実行して、NIS サービスを再起動します。
   - リブート後を繰り返しても持続するサービスの場合は、`svcadm` コマンドを実行します。

     ```
     # svcadm enable -r svc:/network/nis/client
     ```
   - 次のリブートまでしか持続しないサービスの場合は、`ypstop` および `ypstart` コマンドを実行します。

     ```
     # /usr/lib/netsvc/yp/ypstop
     # /usr/lib/netsvc/yp/ypstart
     ```

マシンの NIS ドメインの変更

マシンの NIS ドメイン名を変更する場合は、次の手順に従います。

▼ マシンの NIS ドメイン名を変更する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2 マシンの `/etc/defaultdomain` ファイルを編集して、現在のドメイン名をそのマシンの新しいドメイン名に置き換えます。

たとえば、現在のドメイン名である `sales.doc.com` を `research.doc.com` に変更します。

3 `domainname 'cat /etc/defaultdomain'` を実行します。

4 マシンを NIS クライアント、スレーブサーバー、またはマスターサーバーとして再初期化します。

詳細は、第 5 章「NIS サービスの設定と構成」を参照してください。

NIS を DNS と組み合わせて使用する

一般に NIS クライアントは、マシン名とアドレスの検索に NIS だけが使用されるように、nsswitch.conf ファイルで構成されます。このような検索が失敗した場合は、NIS サーバーはこれらの結果を DNS に転送します。

▼ NIS と DNS によるマシン名とアドレスの検索を設定する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2 hosts.byname と hosts.byaddr という 2 つのマップファイルには、YP_INTERDOMAIN キーが必要です。このキーを検査するために、Makefile を編集し、次の行を変更します。

```
#B=-b
B=
```

から

```
B=-b
#B=
```

これで、マップの作成時に makedbm が -b フラグで起動され、YP_INTERDOMAIN キーが ndbm ファイルに挿入されます。

3 make コマンドを実行してマップを作成し直します。

```
# /usr/ccs/bin/make hosts
```

4 NIS サーバーのすべての /etc/resolv.conf ファイルが有効なネームサーバーを指していることを確認します。

注 – Solaris リリース 2 を実行していない NIS サーバーがある場合は、YP_INTERDOMAIN キーがホストマップに存在することを確認してください。

---

5 DNS 転送を有効にするために、各サーバーを再起動します。

```
# svcadm restart network/nis/server:<instance>
```

この NIS 実装では、ypserv が -d オプションを付けることで自動的に起動し、DNS に要求が転送されます。

混在 NIS ドメインの処理

マスターサーバーとスレーブサーバーのどちらも Solaris リリース 2 を実行していない場合は、次の表を参考にして問題が発生しないように対処してください。「4.0.3+」という表記は、「Solaris OS のリリース 4.0.3 以降」であることを意味します。`makedm -b` コマンドは、`Makefile` の「B」変数への参照です。

表 6–1 異機種システムが混在する NIS ドメインにおける NIS/DNS

	マスターサーバー		
スレーブサーバー	**4.0.3+**	**Solaris NIS**	
4.0.3+	マスターサーバー: `makedbm -b` スレーブサーバー: `ypxfr`	マスターサーバー: `makedbm -b` スレーブサーバー: `ypxfr -b`	マスターサーバー: `ypserv -d` スレーブサーバー: `ypxfr -b`
Solaris NIS	マスターサーバー: `makedbm -b` スレーブサーバー: `ypxfr`	マスターサーバー: `makedbm -b` スレーブサーバー: `ypxfr`	マスターサーバー: `ypserv -d` スレーブサーバー: `resolv.conf` が存在する `ypxfr` または `ypxfr -b`

NIS サービスをオフにする

NIS マスターサーバー上の `ypserv` が使用不可になっている場合は、NIS マップを更新できません。

- クライアント上の NIS を無効にするには、次のように入力します。

```
# svcadm disable network/nis/client
```

- 特定のスレーブサーバーまたはマスターサーバー上の NIS を無効にするには、そのサーバー上で次のように入力します。

```
# svcadm disable network/nis/server
```

◆ ◆ ◆ 第 7 章

7 NISのトラブルシューティング

この章では、NISを実行しているネットワーク上で発生する問題の解決方法について説明します。NISクライアントで発生する問題と、NISサーバーで発生する問題を取り上げます。

NISサーバーやクライアントで問題を解決しようとする前に、NIS環境について説明している第4章「ネットワーク情報サービス(NIS)(概要)」を参照してください。そのあとこの節で、各問題を適切に解説している節を参照してください。

注 – NISサービスはサービス管理機能によって管理されます。このサービスに関する有効化、無効化、再起動などの管理アクションは`svcadm`コマンドを使用して実行できます。NISでSMFを使用する場合の詳細については、86ページの「NISとサービス管理機能」を参照してください。SMFの概要については、『Solarisのシステム管理(基本編)』の第18章「サービスの管理(概要)」を参照してください。また、詳細については、`svcadm(1M)`および`svcs(1)`のマニュアルページを参照してください。

NISサービスの開始および停止は、`ypstart`および`ypstop`コマンドを使用しても行えます。詳細については、`ypstart(1M)`および`ypstop(1M)`のマニュアルページを参照してください。

NISのバインドに関する問題

症状

NISのバインドに関する一般的な問題には、次のようなものがあります。

- `ypbind`がサーバーを見つけられない、あるいはサーバーと通信できないというメッセージが表示される
- サーバーが応答していないというメッセージが表示される

- NIS が使用できないというメッセージが表示される
- クライアントのコマンドがバックグランドモードでゆっくりと処理されているか、通常よりも機能に時間がかかる
- クライアントのコマンドがハングする。システム全体は正常で新しいコマンドを実行できる場合でも、コマンドがハングすることがあります
- クライアントのコマンドがあいまいなメッセージとともに、またはまったくメッセージなしでクラッシュする

1 台のクライアントに影響する NIS の問題

1 台か 2 台のクライアントだけで、NIS のバインドに関する問題を示す症状が発生している場合は、そのクライアントに問題があると考えられます。複数のクライアントが正しくバインドできない場合は、1 台以上の NIS サーバーに問題があると考えられます。128 ページの「複数のクライアントに影響する NIS の問題」を参照してください。

ypbind がクライアントで実行されていない

1 台のクライアントに問題があっても、同じサブネット上のほかのクライアントは正常に機能しています。問題のあるクライアント上で、`ls -l` をたとえば `/usr` のような、多くのユーザーが所有するファイル (クライアント `/etc/passwd` ファイルにはないものも含む) を持つディレクトリで実行します。この結果の表示に、ローカルの `/etc/passwd` には、名前ではなく番号として入ってないファイルの所有者が含まれる場合には、NIS サービスがクライアントで機能していないことを示します。

通常これらの症状は、クライアント `ypbind` プロセスが実行されていないことを示します。NIS クライアントサービスが実行されているかを確認してください。

```
client# svcs network/nis/client
STATE          STIME    FMRI
disabled       Sep_01   svc:/network/nis/client:default
```

クライアントが無効である場合、スーパーユーザーとしてログインするか、同等の役割になり、NIS クライアントサービスを開始します。

```
client# svcadm enable network/nis/client
```

ドメイン名が指定されていないか間違っている

あるクライアントに問題があり、ほかのクライアントは正常に機能していますが、`ypbind` は問題のあるクライアント上で実行されています。クライアントのドメインの設定が間違っている可能性があります。

クライアント上で `domainname` コマンドを実行して、どのドメイン名が設定されているのかを調べます。

```
client7# domainname neverland.com
```

NISのマスターサーバー上の/var/yp内の実際のドメイン名と、出力を比較します。実際のNISドメインは、/var/ypディレクトリ内のサブディレクトリとして表示されます。

```
Client7# ls /var/yp...
-rwxr-xr-x 1 root Makefile
drwxr-xr-x 2 root binding
drwx------ 2 root doc.com ...
```

マシン上でのdomainnameの実行によって得たドメイン名が、/var/yp内のディレクトリとして示されたサーバードメイン名と同じではない場合には、マシンの/etc/defaultdomainファイルで指定されたドメイン名が間違っています。スーパーユーザーとしてログインするか、同等の役割になり、マシンの/etc/defaultdomainファイルでクライアントのドメイン名を修正します。これによって、マシンを起動するたびに、ドメイン名が正しいかどうかが確認されます。ここでマシンをリブートします。

注– ドメイン名では大文字と小文字を区別します。

クライアントがサーバーにバインドされない

ドメイン名が正しく設定されていてypbindが実行中でもコマンドがまだハングする場合には、ypwhichコマンドを実行してクライアントがサーバーにバインドされていることを確認してください。ypbindを起動したばかりのときは、ypwhichを数回実行します。通常、1回目ではドメインがバインドされていないことが通知され、2回目は成功します。

サーバーが使用できない

ドメイン名が正しく設定されていてypbindが実行中のときに、クライアントがサーバーと通信できないというメッセージを受け取った場合には、いくつかの問題が考えられます。

- バインドするサーバーのリストを含む/var/yp/binding/*domainname*/ypserversファイルがクライアントにあるかどうかを確認します。ない場合には、ypinit -cを実行して、設定の順番にクライアントのバインド先のサーバーを指定します。
- クライアントに/var/yp/binding/*domainname*/ypserversファイルがあり、1つ以上のサーバーが使用できない場合には、十分な数のサーバーがあるかどうかを調べます。ない場合には、ypinit -cを実行して、リストにサーバーを追加します。
- クライアントのypserversファイルにリストされたサーバーが、/etc/hostsファイルにエントリを持っているかどうかを確認します。持っていない場合には、NISマップホストの入力ファイルにサーバーを追加して、Working With NIS Mapsで説明しているように、-yppinit cまたは-ypinit 104ページの「NISマップに関する作業」を実行してマップを再構築します。

- /etc/nsswitch.conf ファイルが設定されていて、NIS の他にマシンのローカルの hosts ファイルを参照できるかどうかを確認します。スイッチについての詳細は、第 2 章「ネームサービススイッチ (概要)」を参照してください。
- /etc/nsswitch.conf ファイルが設定されていて、services と rpc に対して files を参照できるかどうかを確認します。スイッチについての詳細は、第 2 章「ネームサービススイッチ (概要)」を参照してください。

ypwhich の表示に一貫性がない

ypwhich を同じクライアントで数回使うと、NIS サーバーが変わるので結果の表示も変わります。これは正常な状態です。NIS クライアントから NIS サーバーへのバインドは、ネットワークや NIS サーバーを使用中の場合は時間の経過に伴って変化します。ネットワークは、すべてのクライアントが受け入れ可能な応答時間を NIS サーバーから得られる点で安定した状態になります。クライアントのマシンが NIS サービスを得ている限りは、サービスの供給元は問題にはなりません。たとえば、NIS サーバーマシンがそれ自体の NIS サービスを、ネットワーク上の別の NIS サーバーから受けることもあります。

サーバーのバインドが不可能な場合

ローカルなサーバーのバインドが不可能な場合には ypset コマンドを使用すると、別のネットワークまたはサブネットの別のサーバーが使用可能な場合には、そのサーバーへのバインドが一時的に可能になります。ただし、-ypset オプションを使用するためには、ypbind を -ypset または -ypsetme オプションのどちらかを指定して、実行する必要があります。詳細は、ypbind(1M) のマニュアルページを参照してください。

```
# /usr/lib/netsvc/yp/ypbind -ypset
```

別の方法については、118 ページの「特定の NIS サーバーへのバインド」を参照してください。

注 – セキュリティーの目的のために、-ypset と -ypsetme のオプションの使用は、制御された状態でのデバッグだけに限定してください。-ypset と -ypsetme のオプションを使用すると、セキュリティーが侵害される恐れがあります。これらのデーモンの実行中はサーバーのバインドをだれでも変更できるため、ほかのユーザーの作業に問題が生じたり重要なデータへの未承認のアクセスが認められたりするためです。これらのオプションで ypbind を起動する場合には、いったん問題を修正したら ypbind を終了して、これらのオプションを指定しないで再起動してください。

ypbind を再起動するには、サービス管理機能 (SMF) を使用します。

```
# svcadm enable -r svc:/network/nis/client:default
```

ypbind のクラッシュ

ypbind が、起動するたびにすぐにクラッシュする場合には、システムのほかの部分で問題を調べます。次のように入力して、rpcbind デーモンが存在するかどうか確認します。

```
% ps -e | grep rpcbind
```

rpcbind が存在しない、安定しない、あるいは動作に異常がある場合には、RPC のマニュアルを参照してください。

正常に機能しているマシンから、問題のあるクライアント上の rpcbind と通信ができる場合があります。正常に機能しているマシンから、次のように入力してください。

```
% rpcinfo client
```

問題のあるクライアント上の rpcbind に問題がない場合には、rpcinfo によって次のように出力されます。

```
program    version    netid    address    service    owner
...
100007    2    udp    0.0.0.0.2.219    ypbind    superuser
100007    1    udp    0.0.0.0.2.219    ypbind    superuser
100007    1    tcp    0.0.0.0.2.220    ypbind    superuser
100007    2    tcp    0.0.0.0.128.4    ypbind    superuser
100007    2    ticotsord    \000\000\020H    ypbind    superuser
100007    2    ticots    \000\000\020K    ypbind    superuser
...
```

使用中のマシンには異なる複数のアドレスがあります。それらのアドレスが表示されない場合は、ypbind によってそのサービスが登録できていません。マシンをリブートして再度 rpcinfo を実行します。ypbind プロセスがあり、NIS サービスを再起動するたびに変更される場合は、rpcbind デーモンが実行中であってもシステムをリブートしてください。

複数のクライアントに影響する NIS の問題

1 台か 2 台のクライアントだけで、NIS のバインドに関する問題を示す症状が発生している場合は、そのクライアントに問題があると考えられます。124 ページの「1 台のクライアントに影響する NIS の問題」を参照してください。複数のクライアントが正しくバインドできない場合は、1 台以上の NIS サーバーに問題があると考えられます。

rpc.yppasswdd が r で始まる制限のないシェルを制限付きとみなしている

1. 次のような特殊な文字列が含まれている /etc/default/yppasswdd を作成します。
   "check_restricted_shell_name=1"
2. 「check_restricted_shell_name=1」文字列をコメント扱いにすると、「r」のチェックは行われません。

ネットワークまたはサーバーが過負荷

ネットワークまたは NIS サーバーが過負荷状態で、クライアント ypbind プロセスに ypserv が時間以内に応答を戻せない場合には、NIS がハングする場合があります。

こういった状態では、ネットワーク上のすべてのクライアントで同じまたは類似した問題が発生します。ほとんどの場合、この状態は一時的です。NIS サーバーが再起動して ypserv を再起動するか、または NIS サーバーまたはネットワーク自体の負荷が減少すると、通常、メッセージは消えます。

サーバーの誤動作

サーバーが起動して実行中であることを確認してください。サーバーが物理的に近くにない場合には、ping コマンドを使ってください。

NIS デーモンが実行されていない

サーバーが起動されていて実行中の場合には、クライアントマシンが正常に動作していることを調べて、ypwhich コマンドを実行します。ypwhich が応答しない場合は、そのコマンドを強制終了します。次に、NIS サーバーで root としてログインし、次のように入力して NIS プロセスが実行中かどうかをチェックします。

```
# ps -e | grep yp
```

注 – -f オプションを ps で使用しないでください。このオプションはユーザー ID を名前に変換しようとするため、より多くのネームサービス検索が失敗する可能性があります。

NISサーバーデーモン(ypserv)もNISクライアントデーモン(ypbind)も実行されていない場合、次のいずれかを入力してデーモンを再起動します。

```
# svcadm restart network/nis/server
or
# /usr/lib/netsvc/yp/ypstop
# /usr/lib/netsvc/yp/ypstart
```

ypbindとypservの両プロセスがNISサーバーで実行中の場合は、ypwhichを実行します。ypwhichが応答しない場合は、ypservがハングしたと考えられるため再起動が必要です。サーバーにrootとしてログインし、次のいずれかを入力してNISサービスを再起動します。

```
# svcadm restart network/nis/server
or
# /usr/lib/netsvc/yp/ypstop
# /usr/lib/netsvc/yp/ypstart
```

サーバーに別のバージョンの**NIS**マップが存在する

NISはマップをサーバー間で伝播するので、ネットワーク上のさまざまなNISサーバーに、同じマップの異なるバージョンが存在することがあります。相違点が長時間継続しない場合には、このバージョンの違いは許容可能です。

マップの不一致のもっとも一般的な原因は、マップの正常な伝播を妨げる何かが存在するためです。たとえば、NISサーバーまたはルーターが、NISサーバー間でダウンしている場合です。すべてのNISサーバーとそれらの間に存在するルーターが実行中の場合には、ypxfrは成功します。

サーバーとルーターが正常に機能している場合には、次のことをチェックします。

- ypxfr出力のログをとります(129ページの「ypxfr出力のログ」を参照)。
- 制御ファイルをチェックします(130ページの「crontabファイルとypxfrシェルスクリプトをチェックする」を参照)。
- マスターサーバー上のypserversマップをチェックします(130ページの「ypserversマップをチェックする」を参照)。

ypxfr出力のログ

特定のスレーブサーバーでマップの更新に問題がある場合には、そのサーバーにログインしてypxfrを対話形式で実行します。ypxfrが失敗するとypxfrがその理由を通知するので、問題の修正が可能になります。ypxfrが成功するが時々失敗するような場合には、メッセージのログをとるためのログファイルを作成します。ログファイルを作成する場合は、スレーブサーバーで次のように入力します。

```
ypslave# cd /var/yp
ypslave# touch ypxfr.log
```

これによって、ypxfrからのすべての出力を保存するypxfr.logファイルが作成されます。

この出力は、ypxfr が対話形式で実行しているときに表示する出力と似ていますが、ログファイルの各行にはタイムスタンプが記録されます。タイムスタンプは通常とは異なる順番になる可能性がありますが、問題はありません。タイムスタンプは、ypxfr が実行し始めたことを示します。ypxfr のコピーが同時に実行されても作業時間が異なる場合は、起動時とは異なる順番でサマリーステータス行がログファイルに書き込まれることがあります。断続的に発生するあらゆる種類の障害がログに記録されます。

注 – 問題を解決したら、ログファイルを削除してログを停止します。削除しないと、ログは制限なく大きくなります。

---

crontab ファイルと ypxfr シェルスクリプトをチェックする

root の crontab ファイルを調べて、それが起動した ypxfr シェルスクリプトをチェックします。これらファイルにタイプミスがあると、伝播に関する問題が発生します。/var/spool/cron/crontabs/root ファイル内でシェルスクリプトを参照できない場合や、任意のシェルスクリプト内でマップを参照できない場合にも、エラーが発生します。

ypservers マップをチェックする

NIS スレーブサーバーが、ドメインに対するマスターサーバー上の ypservers マップにリストされていることも確認してください。リストされていない場合には、スレーブサーバーはサーバーとして正しく機能しますが、yppush はマップの変更をスレーブサーバーに伝播しません。

対策

NIS スレーブサーバーの問題が明白ではない場合には、rcp または ftp を使ってデバッグし、一貫性のないマップの最新バージョンを問題のない NIS サーバーからコピーして問題を解決できます。次に問題のあるマップを転送する方法を示します。

```
ypslave# rcp ypmaster:/var/yp/mydomain/map.\* /var/yp/mydomain
```

* の文字はコマンド行でエスケープされて、ypslave でローカルにではなく ypmaster で展開されます。

ypserv のクラッシュ

ypserv プロセスがほとんど即座にクラッシュして、何度再起動しても安定しないときは、デバッグプロセスは、127 ページの「ypbind のクラッシュ」で説明する内容と実質的に同じです。rpcbind デーモンの存在を次のようにチェックしてください。

```
ypserver% ps -e | grep rpcbind
```

デーモンが見つからない場合は、サーバーをリブートします。デーモンが見つかった場合は、デーモンが実行中であれば次のように入力して同様の出力を検索します。

```
% rpcinfo -p ypserver

% program     vers     proto     port     service
100000    4    tcp    111    portmapper
100000    3    tcp    111    portmapper
100068    2    udp    32813    cmsd
...
100007    1    tcp    34900    ypbind
100004    2    udp    731    ypserv
100004    1    udp    731    ypserv
100004    1    tcp    732    ypserv
100004    2    tcp    32772    ypserv
```

使用中のマシンには、異なる複数のポート番号があることがあります。ypserv プロセスを表す 4 つのエントリは、次のとおりです。

```
100004     2     udp     731     ypserv
100004     1     udp     731     ypserv
100004     1     tcp     732     ypserv
100004     2     tcp     32772     ypserv
```

エントリが 1 つもなく、ypserv がそのサービスを rpcbind で登録できない場合にはマシンをリブートしてください。エントリがある場合には、rpcbind からサービスの登録を解除してから ypserv を再起動します。rpcbind からサービスの登録を解除するには、サーバーで次のように入力します。

```
# rpcinfo -d number 1
# rpcinfo -d number 2
```

number は、rpcinfo によって通知される ID 番号です (前述の例では、100004)。

パート IV

LDAP ネームサービスの設定と管理

ここでは、LDAP ネームサービスの概要を説明します。さらに、Sun Java System Directory Server (以前の名称は Sun ONE Directory Server) の使用に焦点を当てて、Solaris OS での LDAP ネームサービスの設定、構成、管理、そして障害の対処方法についても説明します。

◆ ◆ ◆ 第 8 章

LDAP ネームサービスの紹介 (概要/リファレンス)

この LDAP の章では、Sun Java System Directory Server (以前の名称は Sun ONE Directory Server) で動作する Solaris LDAP ネームサービスクライアントの設定方法について説明します。Sun Java System Directory Server の使用を推奨しますが、必須ではありません。一般的なディレクトリサーバー要件については、第 14 章「LDAP の一般的なリファレンス」で簡潔に説明します。

注 – ディレクトリサーバーは、必ずしも LDAP サーバーである必要はありません。しかし、この章では「ディレクトリサーバー」という言葉は「LDAP サーバー」と同じ意味で使っています。

対象読者

LDAP ネームサービスに関するこれらの章は、LDAP に関する実務上の知識を持つシステム管理者を対象としています。次のリストはこの章を読む前によく理解しておく必要のある概念の一部です。次の概念に関する知識がない場合は、このマニュアルを使って Solaris システムに LDAP ネームサービスを導入することは難しいかもしれません。

- LDAP 情報モデル (エントリ、オブジェクトクラス、属性、タイプ、値)
- LDAP ネームモデル (ディレクトリ情報ツリー (DIT) 構造)
- LDAP 機能モデル (検索パラメータ: ベースオブジェクト (DN)、スコープ、サイズ制限、時間制限、フィルタ (Sun Java System Directory Server のインデックスを表示する)、属性リスト)
- LDAP セキュリティーモデル (認証方式、アクセス制御モデル)
- データの計画方法と DIT、トポロジ、複製、セキュリティーの設計方法を含む LDAP ディレクトリサービスの計画と設計全般

推奨される前提知識

前述の概念についての詳細、また一般的な LDAP とディレクトリサービスの導入について知りたい場合は、次の文書を参照してください。

- 『Understanding and Deploying LDAP Directory Services』Timothy A. Howes, Ph.D、Mark C. Smith 共著

  この本には LDAP ディレクトリサービスの扱い方に関する詳細だけでなく、LDAP を配備する上で役に立つケーススタディが書かれています。配備事例には、大規模な大学、多国籍企業、そしてエクストラネットを使った企業などがあります。
- 『Sun Java System Directory Server 配備ガイド』は Sun Java Enterprise System の文書に収められています。

  このマニュアルでは、基本的なディレクトリ計画 (ディレクトリ設計、スキーマ設計、ディレクトリツリー、トポロジ、複製、およびセキュリティーを含む) が説明されています。最後の章では、単純で小規模な配備計画と、複雑な世界に広がる配備計画の両方のシナリオを説明しています。
- 『Sun Java System Directory Server 管理ガイド』は Sun Java Enterprise System の文書に収められています。

その他の前提条件

Sun Java System Directory Server をインストールする場合は、ご使用のバージョンの Sun Java System Directory Server の『インストールガイド』を参照してください。

LDAP ネームサービスとその他のネームサービスの比較

次の表は、DNS、 NIS、 NIS+、LDAP ネームサービスを比較したものです。

	DNS	NIS	NIS+	LDAP
名前空間	階層	一層	階層	階層
データ記憶領域	ファイル/リソースレコード	2 列のマップ	複数列のテーブル	ディレクトリ (可変) インデックス化したデータベース

	DNS	NIS	NIS+	LDAP
サーバー	マスター/スレーブ	マスター/スレーブ	ルートマスター/非ルートマスター 主/副 キャッシュ/スタブ	マスター/複製 マルチマスター複製
セキュリティー	なし	なし (root または なし)	Secure RPC (AUTH_DH) 認証	SSL、可変
トランスポート	TCP/IP	RPC	RPC	TCP/IP
規模	広域	LAN	LAN	広域

LDAP ネームサービスの利点

- LDAP を使用すると、アプリケーション固有の情報を置き換えて情報の整理統合を実行し、管理するデータベースの数を減らすことができる。
- LDAP を使用すると、異なる複数のネームサービス間でデータを共有できる。
- LDAP により、データの集中的なリポジトリ (格納場所) が提供される。
- LDAP を使用すると、マスターと複製との間でより頻繁にデータの同期を取ることができる。
- LDAP では、プラットフォーム間およびベンダー間の互換性が維持されている。

LDAP ネームサービスの欠点

次に、その他のネームサービスと比較して LDAP の欠点を示します。

- Solaris 8 以前のクライアントはサポートしていない。
- LDAP サーバーをそのクライアントとして使用することはできない。
- LDAP ネームサービスの設定および管理がより複雑なため、注意深い計画が必要である。
- NIS クライアントとネイティブな LDAP クライアントは、同一のクライアントマシン上で共存できない。

注 – ディレクトリサーバー (LDAP サーバー) をそのクライアントとして使用することはできません。つまり、ディレクトリサーバーソフトウェアを実行中のマシンを、LDAP ネームサーバークライアントにすることはできません。

LDAP ネームサービスの設定 (作業マップ)

作業	参照先
パッチがインストールされていることを確認する	
ネットワークモデルを計画する	172 ページの「LDAP ネットワークモデルの計画」
DIT を計画する	第 10 章「LDAP ネームサービスの計画要件 (手順)」
複製サーバーを設定する	174 ページの「LDAP と複製サーバー」
セキュリティーモデルを計画する	175 ページの「LDAP セキュリティーモデルの計画」
クライアントプロファイルおよびデフォルト属性値を選択する	177 ページの「LDAP 用のクライアントプロファイルおよびデフォルト属性値の計画」
データ生成を計画する	178 ページの「LDAP データ生成の計画」
LDAP ネームサービスで使用する前に Sun Java System Directory Server を構成する	『Sun ONE Directory Server 5.2 (Solaris Edition)』
LDAP ネームクライアントで使用するために Sun Java System Directory Server を設定する	第 11 章「LDAP クライアントと Sun Java System Directory Server の設定 (手順)」
プリンタエントリを管理する	191 ページの「プリンタエントリの管理」
LDAP クライアントを初期化する	201 ページの「LDAP クライアントの初期化」
プロファイルを使用してクライアントを初期化する	202 ページの「プロファイルを使用してクライアントを初期化する」
手動でクライアントを初期化する	206 ページの「クライアントを手動で初期設定する」
クライアントの初期化を解除する	207 ページの「クライアントの初期設定を解除する」
サービス検索記述子を使用して、クライアントプロファイルを変更する	184 ページの「サービス検索記述子を使用してさまざまなサービスへのクライアントアクセスを変更する」

作業	参照先
ネームサービス情報を取得する	211 ページの「LDAP ネームサービス情報の検出」
クライアント環境をカスタマイズする	212 ページの「LDAP クライアント環境のカスタマイズ」

◆ ◆ ◆ 第 9 章

9 LDAP 基本コンポーネントおよび概念 (概要)

この章の内容は次のとおりです。

LDAP データ交換フォーマット (LDIF)

LDIF は、ディレクトリサービスのエンティティーおよびその属性を記述するためのテキストベースのフォーマットです。LDIF フォーマットを使用することで、ldapadd や ldapmodify などのコマンドを実行して、ディレクトリ間で情報を移動できます。次に、各サービスの LDIF フォーマットの例を示します。この情報を表示するには、-l オプションを指定して、ldaplist(1) を実行してください。

```
% ldaplist -l hosts myhost


hosts

dn: cn=myhost+ipHostNumber=7.7.7.115,ou=Hosts,dc=mydc,dc=mycom,dc=com
cn: myhost
iphostnumber: 7.7.7.115
objectclass: top
objectclass: device
objectclass: ipHost
description: host 1 - floor 1 - Lab a - building b

% ldaplist -l passwd user1
```

```
passwd

dn: uid=user1,ou=People,dc=mydc,dc=mycom,dc=com
uid: user1
cn: user1
userpassword: {crypt}duTx91g7PoNzE
uidnumber: 199995
gidnumber: 20
gecos: Joe Smith [New York]
homedirectory: /home/user1
loginshell: /bin/csh
objectclass: top
objectclass: shadowAccount
objectclass: account
objectclass: posixAccount

% ldaplist -l services name


services

dn: cn=name+ipServiceProtocol=udp,ou=Services,dc=mydc,dc=mycom,dc=com
cn: name
cn: nameserver
ipserviceprotocol: udp
ipserviceport: 42
objectclass: top
objectclass: ipService

% ldaplist -l group mygroup


group

dn: cn=mygroup,ou=Group,dc=mydc,dc=mycom,dc=com
cn: mygroup
gidnumber: 4441
memberuid: user1
memberuid: user2
memberuid: user3
userpassword: {crypt}duTx91g7PoNzE
objectclass: top
objectclass: posixGroup

% ldaplist -l netgroup mynetgroup


netgroup

cn=mynetgroup,ou=netgroup,dc=central,dc=sun,dc=com objectclass=nisNetgroup
-objectclass: -top
-cn: -mynetgroup
-nisnetgrouptriple: -(user1..mydc.mycom.com,-,) nisnetgrouptriple=(user1.,-,)
-membernisnetgroup: -mylab

% ldaplist -l networks 200.20.20.0


networks

dn: ipNetworkNumber=200.20.20.0,ou=Networks,dc=mydc,dc=mycom,dc=com
```

```
cn: mynet-200-20-20
ipnetworknumber: 200.20.20.0
objectclass: top
objectclass: ipNetwork
description: my Lab Network
ipnetmasknumber: 255.255.255.0
```

% **ldaplist** -l **netmasks 201.20.20.0**

```
netmasks

dn: ipNetworkNumber=201.20.20.0,ou=Networks,dc=mydc,dc=mycom,dc=com
cn: net-201
ipnetworknumber: 201.20.20.0
objectclass: top
objectclass: ipNetwork
description: my net 201
ipnetmasknumber: 255.255.255.0
```

% **ldaplist** -l **rpc ypserv**

```
rpc

dn: cn=ypserv,ou=Rpc,dc=mydc,dc=mycom,dc=com
cn: ypserv
cn: ypprog
oncrpcnumber: 100004
objectclass: top
objectclass: oncRpc
```

% **ldaplist** -l **protocols tcp**

```
protocols

dn: cn=tcp,ou=Protocols,dc=mydc,dc=mycom,dc=com
cn: tcp
ipprotocolnumber: 6
description: transmission control protocol
objectclass: top
objectclass: ipProtocol
```

% **ldaplist** -l **bootparams myhost**

```
bootparams

dn: cn=myhost,ou=Ethers,dc=mydc,dc=mycom,dc=com
bootparameter: root=boothost:/export/a/b/c/d/e
objectclass: top
objectclass: device
objectclass: bootableDevice
cn: myhost
```

% **ldaplist** -l **ethers myhost**

```
ethers

dn: cn=myhost,ou=Ethers,dc=mydc,dc=mycom,dc=com
```

```
macaddress: 8:1:21:71:31:c1
objectclass: top
objectclass: device
objectclass: ieee802Device
cn: myhost
```

```
% ldaplist -l publickey myhost
```

```
publickey

dn: cn=myhost+ipHostNumber=200.20.20.99,ou=Hosts,dc=mydc,dc=mycom,dc=com
cn: myhost
iphostnumber: 200.20.20.99
description: Joe Smith
nispublickey: 9cc01614d929848849add28d090acdaa1c78270aeec969c9
nissecretkey: 9999999998769c999c39e7a6ed4e7afd687d4b99908b4de99
objectclass: top
objectclass: NisKeyObject
objectclass: device
objectclass: ipHost
```

```
% ldaplist -l aliases myname
```

```
aliases

dn: mail=myname,ou=aliases,dc=mydc,dc=mycom,dc=com
cn: myname
mail: myname
objectclass: top
objectclass: mailgroup
mgrprfc822mailmember: my.name
```

LDAP での完全指定ドメイン名の使用

NIS や NIS+ クライアントと違い、LDAP クライアントは常にホスト名として完全指定のドメイン名 (FQDN、Fully Qualified Domain Name) を返します。LDAP の FQDN は、DNS によって返される FQDN に似ています。たとえば、次のドメイン名を考えてみましょう。

```
west.example.net
```

ホスト名 *server* を検索する場合、`gethostbyname()` および `getnameinfo()` はホスト名を FQDN で返します。

```
server.west.example.net
```

また、`server-<番号>` のようなインタフェース固有のエイリアスを使用した場合も、完全指定ホスト名の長いリストが返されます。ホスト名を使用してファイルシステムを共有したりほかの検査を実行する場合、この点に留意する必要があります。たとえば、ローカルホストは FQDN ではなく、遠隔 DNS で解決されるホストだけが FQDN であると想定している場合は、その違いに留意する必要がありま

す。DNS とは違うドメイン名で LDAP を設定する場合、参照先によって同じホストが結果的に 2 つの異なる FQDN になる可能性があります。

デフォルトのディレクトリ情報ツリー (DIT)

Solaris LDAP クライアントはデフォルトで、DIT がある特定の構造を持っていると想定して情報にアクセスします。LDAP サーバーがサポートするドメインごとに、想定された構造を持つサブツリーがあります。ただしこのデフォルト構造は、サービス検索記述子 (Service Search Descriptor、SSD) を指定することで上書きできます。指定されたドメインでは、デフォルト DIT がベースコンテナを保持し、ベースコンテナに特定の情報タイプのエントリを含む既知のコンテナが多数含まれます。これらのサブツリー名については次の表を参照してください。(この情報は RFC 2307 などで確認できます)。

表 9–1　DIT のデフォルト位置

デフォルトコンテナ	情報タイプ
ou=Ethers	bootparams(4)、ethers(4)
ou=Group	group(4)
ou=Hosts	hosts(4)、ipnodes(4)、publickey (ホスト用)
ou=Aliases	aliases(4)
ou=Netgroup	netgroup(4)
ou=Networks	networks(4)、netmasks(4)
ou=People	passwd(1)、shadow(4)、user_attr(4)、audit_user(4)、publickey (ユーザー用)
ou=printers	printers(4)
ou=Protocols	protocols(4)
ou=Rpc	rpc(4)
ou=Services	services(4)
ou=SolarisAuthAttr	auth_attr(4)
ou=SolarisProfAttr	prof_attr(4)、exec_attr(4)
ou=projects	project
automountMap=auto_*	auto_*

デフォルトの LDAP スキーマ

スキーマは、LDAP ディレクトリ内にエントリとして格納可能な情報タイプの定義です。LDAP ネームサービスクライアントをサポートするために、ディレクトリサーバースキーマの拡張が必要な場合があります。IETF および Solaris 固有のスキーマの詳細については、第 14 章「LDAP の一般的なリファレンス」を参照してください。IETF Web サイト `http://www.ietf.org` で、さまざまな RFC にアクセスできます。

サービス検索記述子 (SSD) とスキーママッピング

注 – スキーママッピングは、注意深くかつ一貫した方法で使用する必要があります。マッピングされた属性の構文が、マッピング先の属性との一貫性を保持していることを確認してください。つまり、単一値の属性が単一値の属性にマッピングされ、属性の構文が一致しており、マッピングされたオブジェクトクラスが適正な必須 (通常はマッピングされた) 属性を保持することを確認します。

前述したように、LDAP ネームサービスはデフォルトで、DIT が特定の方法で構築されているという想定のもとに動作します。必要に応じて、DIT 内のデフォルト以外の場所を検索するように Solaris LDAP ネームサービスに指示することができます。また、デフォルトのスキーマで指定された属性やオブジェクトクラスの代わりに、別の属性やオブジェクトクラスを指定して使用することもできます。デフォルトフィルタの詳細については、247 ページの「LDAP ネームサービスで使用されるデフォルトフィルタ」を参照してください。

SSD の説明

`serviceSearchDescriptor` 属性は、LDAP ネームサービスクライアントが特定のサービスに関する情報を検索する方法および場所を定義します。`serviceSearchDescriptor` には、サービス名に続き、1 つ以上のセミコロンで区切られたベース - スコープ - フィルタのセットが含まれます。これらのベース - スコープ - フィルタのセットは特定のサービス専用の検索定義に使用され、指定された順番で検索されます。特定のサービスに対して複数のベース - スコープ - フィルタが指定されている場合、このサービスは、特定のエントリを検索する際、指定されたスコープおよびフィルタを保持する各ベースを検索します。

注 – SSD では、デフォルト位置は SSD に含められていない限り、サービス (データベース) の検索対象にはなりません。サービスに複数の SSD が指定されている場合、予期しない結果になることがあります。

次の例では、Solaris LDAP ネームサービスクライアントが、passwd サービスに対して、ou=west,dc=example,dc=com で 1 レベルの検索を実行し、次に ou=east,dc=example,dc=com で 1 レベルの検索を実行します。ユーザーの username に対して passwd データを検索する場合、各 BaseDN に対してデフォルトの LDAP フィルタ (&(objectClass=posixAccount)(uid=username)) が使用されます。

```
serviceSearchDescriptor: passwd:ou=west,dc=example,dc=com;ou=east,
dc=example,dc=com
```

次の例では、Solaris LDAP ネームサービスクライアントは、ou=west,dc=example,dc=com 内で passwd サービスのサブツリー検索を実行します。ユーザー username の passwd データを検索する場合、LDAP フィルタ (&(fulltimeEmployee=TRUE)(uid=username)) を使用して、サブツリー ou=west,dc=example,dc=com が検索されます。

```
serviceSearchDescriptor: passwd:ou=west,dc=example,
dc=com?sub?fulltimeEmployee=TRUE
```

特定のサービスタイプに複数のコンテナを関連付けることも可能です。次の例では、サービス検索記述子が 3 つのコンテナでパスワードエントリを検索することを指定しています。

```
ou=myuser,dc=example,dc=com
ou=newuser,dc=example,dc=com
ou=extuser,dc=example,dc=com
```

例の末尾の「,」は、SSD の相対ベースに defaultSearchBase が付加されることを意味します。

```
defaultSearchBase: dc=example,dc=com
serviceSearchDescriptor: \
passwd:ou=myuser,;ou=newuser,;ou=extuser,dc=example,dc=com
```

スキーママップ

Solaris LDAP ネームサービスを使用すると、1 つ以上の属性名をいずれかのサービスに再マッピングできます(Solaris LDAP クライアントは、第 14 章「LDAP の一般的なリファレンス」に記載されているよく知られている属性を使用します)。属性を対応づける場合、その属性が元の属性と同じ意味および構文を必ず保持するようにしてください。userPassword 属性を対応づけると、問題が発生する場合があります。

スキーママッピングを使用する理由として、次の 2 つが挙げられます。

- 既存のディレクトリサーバー内の属性を対応づけしたい
- 大文字小文字のみが異なるユーザー名を使用する場合、大文字小文字を無視する uid 属性を、大文字小文字を無視しない属性に対応づける必要があります

この属性の書式は、service:attribute-name=mapped-attribute-name です。

指定されたサービスに対して複数の属性を対応づける場合は、複数の attributeMap 属性を定義できます。

次の例では、uid および homeDirectory 属性を passwd サービスで利用する場合、常に employeeName および home 属性が使用されます。

```
attributeMap: passwd:uid=employeeName
attributeMap: passwd:homeDirectory=home
```

passwd サービスの gecos 属性を複数の属性に対応づける場合、特殊なケースが 1 つ存在します。次に例を示します。

```
attributemap: gecos=cn sn title
```

上の例では、gecos 値が、空白で区切られた cn、sn、および title 属性値のリストに対応づけられます。

objectClass マップ

Solaris LDAP ネームサービスを使用すると、オブジェクトクラスを任意のサービス用に対応づけしなおすことができます。特定のサービス用に複数のオブジェクトクラスを対応づける場合、複数の objectclassMap 属性を定義できます。次の例では、posixAccount オブジェクトクラスを使用する場合、常に myUnixAccount オブジェクトクラスが使用されます。

```
objectclassMap: passwd:posixAccount=myUnixAccount
```

LDAP クライアントプロファイル

Solaris クライアントのセットアップを容易にし、各クライアントに同じ情報を再入力する手間を省くために、ディレクトリサーバー上に単一のクライアントプロファイルを作成します。この単一のプロファイルに、使用するすべてのクライアントの構成を定義します。プロファイル属性への以降の変更はすべて、定義されたリフレッシュ頻度でクライアントに送信されます。

これらのクライアントプロファイルは、LDAP サーバー上のよく知られた位置に格納されます。指定されたドメインのルート DN は、nisDomainObject のオブジェクトクラス、およびクライアントのドメインを含む nisDomain 属性を保持する必要があります。すべてのプロファイルは、このコンテナと相対的な関係にある ou=profile コンテナ内に配置されます。これらのプロファイルは、匿名で読み取り可能にする必要があります。

クライアントのプロファイル属性

次の表に、Solaris LDAP クライアントのプロファイル属性を示します。このプロファイル属性は、idsconfig の実行時に自動的に設定されます。クライアントプロファイルを手動で設定する方法については、206 ページの「クライアントを手動で初期設定する」と idsconfig(1M) のマニュアルページを参照してください。

表 9–2 クライアントのプロファイル属性

属性	説明
cn	プロファイル名。デフォルト値はありません。必ず指定する必要があります。
preferredServerList	優先使用されるサーバーのホストアドレスの、空白で区切られたリスト。(ホスト名は使用しない)。defaultServerList 内のサーバーより「前に」、接続が成功するまで、このリスト内のサーバーへの接続が順番に試みられます。デフォルト値はありません。preferredServerList または defaultServerList に 1 つ以上のサーバーを指定する必要があります。
defaultServerList	デフォルトサーバーのホストアドレスの、空白で区切られたリスト。(ホスト名は使用しない)。preferredServerlist 内のサーバーへの接続試行後に、接続が確立されるまで、クライアントのサブネット上のデフォルトサーバーへの接続、続いて残りのデフォルトサーバーへの接続が試みられます。preferredServerList または defaultServerList に 1 つ以上のサーバーを指定する必要があります。このリスト内のサーバーへの接続は、優先サーバーリストのサーバーへの接続試行後に試みられます。デフォルト値はありません。
defaultSearchBase	よく知られたコンテナの検索に使用する相対識別名。デフォルト値はありません。ただしこの値は、serviceSearchDescriptor 属性で指定されたサービスで置き換えることが可能です。
defaultSearchScope	クライアントによるデータベース検索の適用範囲を定義します。この値は、serviceSearchDescriptor 属性で置き換えることが可能です。指定可能な値は one または sub です。デフォルト値は 1 レベルの検索 (値はone) です。
authenticationMethod	クライアントが使用する認証方式を示します。デフォルト値は none (匿名) です。詳細については、158 ページの「認証方式の選択」を参照してください。
credentialLevel	クライアントが認証に使用する証明書タイプを示します。anonymous、proxy、または self (「ユーザー別」とも呼ばれる) を選択できます。デフォルトは anonymous です。
serviceSearchDescriptor	クライアントがネームデータベースを検索する方法および場所を定義します (例、クライアントが DIT 内の 1 つ以上の場所を検索する)。デフォルトでは、SSD は定義されていません。

表 9–2 クライアントのプロファイル属性 (続き)

属性	説明
serviceAuthenticationMethod	クライアントが特定のサービスで使用する認証メソッド。デフォルトでは、サービス認証メソッドは定義されていません。サービスで serviceAuthenticationMethod が定義されていない場合、authenticationMethod の値がデフォルトになります。
attributeMap	クライアントが使用する属性マッピング。デフォルトでは、attributeMap は定義されていません。
objectclassMap	クライアントが使用するオブジェクトクラスマッピング。デフォルトでは、objectclassMap は定義されていません。
searchTimeLimit	クライアントが許可する、タイムアウトまでの最長検索時間 (秒)。この値は、LDAP サーバーが許可する、検索完了までの時間に影響を与えません。デフォルト値は 30 秒
bindTimeLimit	クライアントがサーバーとのバインドに許可する最長時間 (秒)。デフォルト値は 30 秒です。
followReferrals	クライアントが LDAP 参照に準拠するかどうかを指定します。指定可能な値は TRUE または FALSE です。デフォルト値は TRUE です。
profileTTL	ldap_cachemgr(1M) により実行される、LDAP サーバーからのクライアントプロファイルの更新間隔。デフォルト値は 43200 秒 (12 時間) です。値が 0 の場合、プロファイルは更新されません。

ローカルのクライアント属性

次の表に、ldapclient を使用してローカルに設定可能なクライアント属性を示します。詳細については、ldapclient(1M) のマニュアルページを参照してください。

Solaris 10 10/09 リリース以降では、enableShadowUpdate スイッチが使用できます。詳細は、157 ページの「enableShadowUpdate スイッチ」を参照してください。

表 9–3 ローカルのクライアント属性

属性	説明
adminDN	管理者資格の管理者エントリの識別名を指定します。クライアントシステムの enableShadowUpdate スイッチの値が true で、credentialLevel の値が self 以外の場合、adminDN を指定する必要があります。
adminPassword	管理者資格の管理者エントリのパスワードを指定します。クライアントシステムの enableShadowUpdate スイッチの値が true で、credentialLevel の値が self 以外の場合、adminPassword を定義する必要があります。

表 9–3 ローカルのクライアント属性 (続き)

属性	説明
domainName	クライアントのドメイン名 (クライアントシステムのデフォルトドメインになる) を指定します。デフォルト値はなく、必ず指定する必要があります。
proxyDN	プロキシの識別名。proxy の credentialLevel を使用してクライアントシステムを構成する場合、proxyDN を指定する必要があります。
proxyPassword	プロキシのパスワード。プロキシの credentialLevel を使用してクライアントシステムを構成する場合、proxyPassword を定義する必要があります。
certificatePath	証明書データベースを含む、ローカルファイルシステム上のディレクトリ。TLS を使用し、authenticationMethod または serviceAuthenticationMethod を指定してクライアントシステムを構成する場合、この属性が使用されます。デフォルト値は /var/ldap です。

注 – SSD 内の BaseDN に「末尾のコンマが含まれる」場合、*defaultSearchBase* の相対値として処理されます。検索実行前に、defaultSearchBase の値が BaseDN に付加されます。

ldap_cachemgr デーモン

ldap_cachemgr は、LDAP クライアントマシン上で稼働するデーモンです。LDAP クライアントを起動すると、ldap_cachemgr デーモンが起動されます。このデーモンは、次の主要機能を実行します。

- root として稼働し、構成ファイルへのアクセスを取得する
- サーバー上のプロファイルに格納されたクライアント構成情報を更新して、クライアントからこのデータを引き出す
- 使用可能な LDAP サーバーのソート済みリストを管理する
- さまざまなクライアントから送信される一般的な検索要求をキャッシュして、検索効率を向上させる
- ホスト検索の効率を向上させる
- Solaris 10 10/09 リリース以降では、enableShadowUpdate スイッチが true に設定されている場合、構成した管理者資格にアクセスし、shadow データの更新を実行します。

注 – LDAP ネームサービスを機能させるには、ldap_cachemgr が常に実行されている必要があります。

詳細については、ldap_cachemgr(1M) のマニュアルページを参照してください。

LDAP ネームサービスのセキュリティーモデル

はじめに

Solaris LDAP ネームサービスは、LDAP リポジトリを 2 つの異なる方法で使用できます。1 つは、ネームサービスと認証サービスの両方のソースとして使用する方法です。もう 1 つは、厳密にネームデータのソースとして使用する方法です。この節では、LDAP リポジトリをネームサービスと認証サービスの両方のソースとして使用する場合の、クライアント識別情報、認証方式、pam_ldap と pam_unix モジュール、およびアカウント管理の概念について説明します。また、LDAP ネームサービスを Kerberos 環境 (『Solaris のシステム管理 (セキュリティサービス)』のパート VI「Kerberos サービス」) および pam_krb5(5) モジュールと組み合わせて使用する方法についても説明します。

注 – 以前は、pam_ldap アカウント管理を有効にすると、システムにログインする際には、常にすべてのユーザーが認証用にログインパスワードを入力する必要がありました。そのため、rsh、rlogin、ssh などのツールによるパスワードを使用しないログインは失敗します。

一方、pam_ldap(5) を Sun Java System Directory Server DS5.2p4 以降のリリースで使用することで、ユーザーはパスワードを入力せずに、rsh、rlogin、rcp、および ssh を使ってログインできるようになりました。

pam_ldap(5) は変更され、ユーザーのログイン時に Directory Server への認証を実行せずに、アカウントの管理およびユーザーのアカウント状態の取得を実行できるようになりました。Directory Server 上でこの機能を制御するのは、1.3.6.1.4.1.42.2.27.9.5.8 です。これは、デフォルトで有効になっています。

この制御をデフォルト以外に変更する場合は、Directory Server 上でアクセス制御情報 (ACI) を追加します。

```
dn: oid=1.3.6.1.4.1.42.2.27.9.5.8,cn=features,cn=config
objectClass: top
objectClass: directoryServerFeature
oid:1.3.6.1.4.1.42.2.27.9.5.8
cn:Password Policy Account Usable Request Control
aci: (targetattr != "aci")(version 3.0; acl "Account Usable";
     allow (read, search, compare, proxy)
     (groupdn = "ldap:///cn=Administrators,cn=config");)
creatorsName: cn=server,cn=plugins,cn=config
modifiersName: cn=server,cn=plugins,cn=config
```

注 – Kerberos を認証システムとして使用し、LDAP ネームシステムに統合する場合は、Kerberos を利用して企業内でシングルサインオン (SSO) 環境を実現できます。また、ユーザーまたはホストごとに LDAP ネームデータのクエリー検索を実行する際にも、同じ識別システムを使用できます。

LDAP リポジトリ内の情報にアクセスする場合、クライアントは最初にディレクトリサーバーで識別情報を確立できます。この識別情報は、匿名にも、LDAP サーバーの認識するオブジェクトにもできます。クライアントの識別情報およびサーバーのアクセス制御情報(ACI) に基づいて、LDAP サーバーはクライアントに対してディレクトリ情報の読み取りまたは書き込みを許可します。ACI の詳細については、ご使用のバージョンの Sun Java System Directory Server の『管理者ガイド』を参照してください。

特定の要求に関して匿名以外で接続している場合、クライアントは、クライアントとサーバーの両方がサポートする認証方式でサーバーに識別情報を証明する必要があります。クライアントは識別情報を確立後に、さまざまな LDAP 要求を実行できます。

pam_ldap を使用する場合、ネームサービスと認証サービス(pam_ldap) がディレクトリにアクセスする方法には違いがあります。ネームサービスは、事前定義された識別情報に基づくディレクトリから、さまざまなエントリおよびその属性を読み取ります。認証サービスは、ユーザーの名前とパスワードを使用して LDAP サーバーへの認証を行い、ユーザーが適正なパスワードを入力したかどうかを確認します。認証サービスについての詳細は、pam_ldap(5) のマニュアルページを参照してください。

Kerberos を認証に使用する場合、および LDAP ネームサービス内の認証も有効にする場合 (ユーザー別の認証方式で必要)、Kerberos は二重の機能を提供できます。ディレクトリへの認証に、サーバーへの Kerberos 認証、および主体 (ユーザーまたはホスト) に対する Kerberos 識別情報が使用されます。これにより、システムの認証に使用されるのと同じユーザー識別情報がディレクトリの認証にも使用され、検索と更新が実行されます。管理者は、必要に応じ、アクセス制御情報 (ACI) をディレクトリ内で使用して、ネームサービスで得られる結果を制限できます。

Transport Layer Security (TLS)

注 – Solaris LDAP ネームサービス用の TLS を使用する場合、ディレクトリサーバーは、LDAP および SSL 用にデフォルトポート 389 および 636 をそれぞれ使用する必要があります。ディレクトリサーバーがこれらのポートを使用しない場合、TLS を使用することはできません。

TLS を LDAP クライアントとディレクトリサーバー間の通信のセキュリティー保護に使用すると、機密性とデータ整合性を確保することができます。TLS プロトコルは、Secure Sockets Layer (SSL) プロトコルのスーパーセットです。Solaris LDAP ネームサービスは、TLS 接続をサポートします。SSL を使用すると、ディレクトリサーバーおよびクライアントに負荷がかかることに留意してください。

SSL 対応のディレクトリサーバーを設定する必要があります。SSL 対応の Sun Java System Directory Server の設定方法の詳細については、ご使用のバージョンの Sun Java System Directory Server の『管理者ガイド』を参照してください。SSL 対応の LDAP クライアントも設定する必要があります。

TLS を使用する場合は、必要なセキュリティーデータベースをインストールしなければなりません。具体的には証明書ファイルと鍵データベースファイルが必要です。たとえば、Netscape Communicator の古いデータベースフォーマットを使用する場合、cert7.db と key3.db の 2 つのファイルが必要です。あるいは、Mozilla の新しいデータベースフォーマットを使用する場合、cert8.db、key3.db、および secmod.db の 3 つのファイルが必要です。cert7.db ファイルまたは cert8.db ファイルには、信頼された証明書が入ります。key3.db ファイルには、クライアントの鍵が入ります。LDAP ネームサービスクライアントがクライアントの鍵を使用しない場合で

も、このファイルは必要です。secmod.db ファイルには、PKCS#11 などのセキュリティーモジュールが入ります。このファイルは、古いフォーマットを使用する場合には必要ありません。

詳細については、208 ページの「TLS のセキュリティーの設定」を参照してください。

クライアント資格レベルの割り当て

LDAP ネームサービスクライアントは、クライアントの資格レベルに従って LDAP サーバーを認証します。LDAP クライアントには、次の 4 つの資格レベルの 1 つを割り当てて、ディレクトリサーバーの認証を行うことができます。

- anonymous
- proxy
- proxy anonymous
- self (このマニュアルでは「ユーザー別」と呼ぶ)

匿名 *(anonymous)*

匿名でのアクセスを利用する場合、すべてのユーザーが使用可能なデータだけにアクセスできます。匿名モードでは、LDAP BIND 操作は実行されません。また、セキュリティーの問題も考慮する必要があります。ディレクトリの特定部分に匿名アクセスを許可する場合、そのディレクトリへのアクセス権を保持するすべてのユーザーが読み取りアクセスを保持することになります。資格レベルとして anonymous を使用する場合、すべての LDAP ネームエントリおよび属性に読み取りアクセスを許可する必要があります。

注意 – 匿名でのディレクトリへの書き込みは、決して許可してはなりません。この書き込みを許可すると、どのユーザーでも書き込みアクセス権のある DIT 内の情報 (別のユーザーのパスワードや自分自身の識別情報など) を変更することが可能になります。

注 – Sun Java System Directory Server を使用すると、IP アドレス、DNS 名、認証方式、および時刻に基づいてアクセスを制限できます。さらに指定を加えて、アクセスを制限することもできます。詳細については、ご使用のバージョンの Sun Java System Directory Server の『管理者ガイド』のアクセス権の管理に関する章を参照してください。

プロキシ *(Proxy)*

クライアントは、単一のプロキシアカウントを使用して、ディレクトリへの認証またはバインドを行います。このプロキシアカウントには、ディレクトリへのバインドを許可されるエントリを設定できます。このプロキシアカウントは、LDAP サーバー上でネームサービス機能を実行するのに十分のアクセス権を必要とします。プロキシアカウントは、システムごとに共有されるリソースです。つまり、root ユーザーを含む、プロキシアクセスを使ってシステムにログインした各ユーザーには、そのシステム内のほかのすべてのユーザーと同じ結果が表示されます。プロキシ資格レベルを使用して、すべてのクライアントで `proxyDN` および `proxyPassword` を構成する必要があります。暗号化された `proxyPassword` はローカルのクライアントに格納されます。別のクライアントグループに対しては別のプロキシを設定できます。たとえば全営業クライアント用のプロキシを構成する場合、企業全体からアクセス可能なディレクトリと営業ディレクトリの両方へのアクセスを許可しつつ、給与情報を保持する人事ディレクトリへのアクセスを許可しない、という方法が可能です。最も極端な例として、各クライアントに別個のプロキシを割り当てることや、すべてのクライアントに同じプロキシを割り当てることも可能です。一般的な LDAP 配備はこの両極端の中間に位置します。選択は慎重に行なってください。プロキシエージェントが不足していると、リソースへのユーザーアクセスを制御する能力が制限されます。ただし、プロキシが多過ぎる場合、システムの設定および保守が困難になります。適切な権限をプロキシユーザーに付与する必要がありますが、その程度は環境によって異なります。使用する構成に最適の認証方式を決定するための情報については、158 ページの「資格の保存」を参照してください。

プロキシユーザーのパスワードを変更した場合、そのプロキシユーザーを使用するすべてのクライアントで情報を更新する必要があります。LDAP アカウントのパスワード有効期間を設定する場合、プロキシユーザーに関してはこの設定を解除してください。

注 – プロキシ資格レベルは、指定されたシステムのすべてのユーザーおよびプロセスに適用されます。2 人のユーザーが異なるネーミングポリシーを使用する場合は、別個のマシンを使用するか、ユーザー別の認証モデルを使用する必要があります。

また、クライアントが認証にプロキシ資格を使用する場合、`proxyDN` はすべてのサーバーで同じ `proxyPassword` を保持する必要があります。

匿名プロキシ(proxy anonymous)

匿名プロキシは複数値のエントリで、複数の資格レベルが内部に定義されています。匿名プロキシレベルを割り当てられたクライアントは、最初にそのプロキシ識別情報を使用して認証を試みます。ユーザーのロックアウト、パスワードの有効期限切れなどの何らかの理由でクライアントがプロキシユーザーとしての認証ができなかった場合、クライアントは匿名アクセスを使用します。この場合、ディレクトリの構成に応じて、別のサービスレベルに移行する可能性があります。

ユーザー別 (Per User)

ユーザー別 (self) の認証では、ディレクトリサーバーの認証時に Kerberos 識別情報 (主体) を使用して各ユーザーまたは各システムの検索が実行されます。ユーザー別の認証では、システム管理者は、アクセス制御情報 (ACI)、アクセス制御リスト (ACL)、役割、グループ、またはその他のディレクトリアクセス制御機構を使用して、特定のユーザーまたはシステムの特定のネームサービスデータへのアクセスを許可または拒否できます。

注 – ユーザー別のモードを設定する場合は、このモードを表す設定値「self」を使用します。

ユーザー別の認証モデルを使用するには、Kerberos シングルサインオンサービスを配備する必要があります。また、配備に使用する 1 つ以上のディレクトリサーバーで SASL および SASL/GSSAPI 認証機構をサポートする必要があります。Kerberos では、ホスト名の検索に LDAP ではなく、ファイルおよび DNS を使用することを前提としているため、この環境には DNS を配備するようにしてください。また、ユーザー別の認証を使用するには、`nscd` を有効にする必要があります。この構成では、`nscd` デーモンはオプションの構成要素ではありません。

`enableShadowUpdate` スイッチ

Solaris 10 10/09 リリース以降では、クライアントで `enableShadowUpdate` スイッチが `true` に設定されている場合、管理者資格がシャドウデータの更新に使用されます。シャドウデータは、ディレクトリサーバーの `shadowAccount` オブジェクトクラスに格納されます。管理者資格は、150 ページの「ローカルのクライアント属性」で説明しているように、`adminDN` および `adminPassword` 属性の値によって定義されます。これらの管理者資格は、それ以外の目的には使用されません。

管理者資格のプロパティーは `Proxy` 資格のプロパティと類似しています。管理者資格の場合、シャドウデータを読み取ったり更新するには、ユーザーはゾーンのすべての特権を持つか、`root` の有効な UID を持っている必要があるという例外があります。管理者資格は、ディレクトリへのバインドが許可されるエントリに割り当てることができます。ただし、LDAP サーバーのディレクトリマネージャー識別情報 (`cn=Directory Manager`) には同じものを使用しないでください。

管理者資格が設定されたこのエントリは、ディレクトリ内のシャドウデータに対する十分な読み取りおよび書き込みアクセスを持っている必要があります。エントリはシステムごとに共有されるリソースであるため、`adminDN` および `adminPassword` 属性はクライアントごとに構成する必要があります。暗号化された `adminPassword` はローカルのクライアントに格納されます。パスワードには、クライアント用に構成された認証方式と同じ方式が使用されます。管理者資格は、シャドウデータの読み取りと更新を行うために、特定のシステムのすべてのユーザーおよびプロセスによって使用されます。

資格の保存

プロキシ識別情報を使用するようクライアントを設定する場合、クライアントは proxyDN および proxyPassword を /var/ldap/ldap_client_cred 内に保存します。セキュリティー保護のため、このファイルへのアクセスは root のみに許可され、proxyPassword の値は暗号化されます。過去の LDAP 実装ではプロキシ資格はクライアントのプロファイル内に格納されましたが、Solaris 9 LDAP ネームサービスではこれは行われません。初期化時に ldapclient を使用して設定されたプロキシ資格は、すべてローカルに保存されます。このため、プロキシの DN およびパスワード情報に関するセキュリティーが向上します。クライアントプロファイルの設定方法の詳細については、第 12 章「LDAP クライアントの設定 (手順)」を参照してください。

同様に、シャドウデータの更新が有効になるようにクライアントを構成し、クライアント資格レベルが self ではない場合、クライアントは adminDN および adminPassword 属性を /var/ldap/ldap_client_cred ファイルに保存します。adminPassword の値も暗号化され、ldap_cachemgr デーモンプロセスによってのみ使用されます。

ユーザー別の認証を使用するようクライアントを構成している場合、認証時に各主体 (各ユーザーまたはホスト) 用の Kerberos 識別情報および Kerberos チケット情報が使用されます。この環境では、ディレクトリサーバーは Kerberos 主体を DN にマッピングします。この DN の認証には、Kerberos 資格が使用されます。次に、ディレクトリサーバーは、必要に応じアクセス制御情報 (ACI) 機構を使用して、ネームサービスデータへのアクセスを許可または拒否します。この状況では、ディレクトリサーバーの認証に Kerberos チケット情報が使用されます。システムが、認証 DN またはパスワードをシステムに保存することはありません。したがって、この種類の構成では、クライアントを ldapclient コマンドを使用して初期化するときに、adminDN および adminPassword 属性を指定する必要はありません。

認証方式の選択

プロキシ資格または匿名プロキシ資格を割り当てる場合、プロキシによるディレクトリサーバーへの認証方式も選択する必要があります。デフォルトの認証方式は none (匿名によるアクセス) です。認証方式には、関連するトランスポートセキュリティーオプションも含まれます。

認証方式には、資格レベルと同様、複数値を指定できます。たとえば、クライアントプロファイルを設定することにより、クライアントが TLS でセキュリティー保護された simple メソッドを最初に使用してバインドを試みるようにできます。これが成功しない場合、クライアントは sasl/digest-MD5 メソッドを使用してバインドを試みます。そのあと、authenticationMethod は tls:simple;sasl/digest-MD5 になります。

LDAP ネームサービスは、いくつかの Simple Authentication and Security Layer (SASL) 機構をサポートします。これらの機構を使用すると、TLS なしでセキュリティー保護されたパスワードを交換できます。ただし、これらの機構はデータの完全性や機密性を保証するものではありません。SASL の詳細については、RFC 2222 を参照してください。

次の認証機構がサポートされています。

- `none`

  クライアントは、ディレクトリへの認証を行いません。これは、`anonymous` 資格レベルと等価です。

- `simple`

  認証方式 `simple` を使用する場合、クライアントシステムはユーザーのパスワードを平文で送信してサーバーへのバインドを実行します。このため、セッションが IPsec により保護されていない限り、パスワードが漏洩しやすくなります。認証方式 `simple` を使用する主な利点は、すべてのディレクトリサーバーがこの方式をサポートしていること、および設定が容易であるという点です。

- `sasl/digest-MD5`

  認証時にクライアントのパスワードは保護されますが、セッションは暗号化されません。Sun Java System Directory Server を含むいくつかのディレクトリサーバーは、`sasl/digest-MD5` 認証方式もサポートします。`digest-MD5` の主な利点は、認証時にパスワードが平文のままネットワーク上を流れないため、`simple` よりも安全であるという点です。`digest-MD5` の詳細については、RFC 2831 を参照してください。`digest-MD5` は、`cram-MD5` のセキュリティーが改善されたものと見なされます。

  `sasl/digest-MD5` を使用する場合、認証はセキュリティー保護されますがセッションは保護されません。

  注 – Sun Java System Directory Server を使用している場合、パスワードをディレクトリ内に「平文」で格納する必要があります。

- `sasl/cram-MD5`

  この場合、LDAP セッションは暗号化されませんが、`sasl/cram-MD5` を使用して認証が行われるため、認証時にクライアントのパスワードが保護されます。

  `cram-MD5` 認証方式の詳細については、RFC 2195 を参照してください。すべてのディレクトリサーバーが `cram-MD5` をサポートしているわけではありません。たとえば、Sun Java System Directory Server は `cram-MD5` をサポートしません。

- `sasl/GSSAPI`

  この認証方式は、ユーザー別の検索を有効にする場合に、self 資格モードとともに使用されます。クライアントの資格を使用するために割り当てられたユーザー別の nscd は、sasl/GSSAPI 方式およびクライアントの Kerberos 資格を使

用して、ディレクトリサーバーへのバインドを実行します。ディレクトリサーバーでは、アクセスをユーザー別に制御できます。

- `tls:simple`

  クライアントは、`simple` を使用してバインドを行い、セッションは暗号化されます。パスワードは保護されます。

- `tls:sasl/cram-MD5`

  `sasl/cram-MD5` を使用して、LDAP セッションの暗号化およびクライアントによるディレクトリサーバーへの認証が行われます。

- `tls:sasl/digest-MD5`

  `sasl/digest-MD5` を使用して、LDAP セッションの暗号化およびクライアントによるディレクトリサーバーへの認証が行われます。

注意 – Sun Java System Directory Server で `digest-MD5` を使用する場合、パスワードを平文で格納する必要があります。認証方式を `sasl/digest-MD5` または `tls:sasl/digest-MD5` に設定する場合、プロキシユーザーのパスワードを平文で格納する必要があります。平文で格納する場合には、`userPassword` 属性が適切な ACI を保持するようにして読み取り不可にするよう、特に注意してください。

次の表に、さまざまな認証方式およびその特性の概要を示します。

表 9–4 認証方式

	バインド	通信時のパスワード	Sun Java System Directory Server でのパスワード	セッション
`none`	いいえ	なし	なし	暗号化しない
`simple`	はい	平文	任意	暗号化しない
`sasl/digest-MD5`	はい	暗号化	平文	暗号化しない
`sasl/cram-MD5`	はい	暗号化	なし	暗号化しない
`sasl/GSSAPI`	はい	Kerberos	Kerberos	暗号化
`tls:simple`	はい	暗号化	任意	暗号化
`tls:sasl/cram-MD5`	はい	暗号化	なし	暗号化
`tls:sasl/digest-MD5`	はい	暗号化	平文	暗号化

認証とサービス

認証方式を特定のサービスに対して `serviceAuthenticationMethod` 属性に指定できます。現在この機能をサポートしているサービスを次に示します。

- passwd-cmd

  このサービスは、passwd(1) により、ログインパスワードおよびパスワード属性の変更に使用されます。

- keyserv

  このサービスは、chkey(1) および newkey(1M) ユーティリティーにより、ユーザーの Diffie-Hellman 鍵ペアの作成および変更に使用されます。

- pam_ldap

  このサービスは、pam_ldap(5) を使用したユーザーの認証に使用されます。

  pam_ldap は、アカウントの管理をサポートします。

注 – サービスが serviceAuthenticationMethod セットを保持しない場合、authenticationMethod 属性の値がデフォルトになります。

注 – ユーザー別のモードでは、163 ページの「pam_krb5 サービスモジュール」 (pam Kerberos) が認証サービスとして使用されます。この操作モードでは、ServiceAuthenticationMethod は不要です。

注 – enableShadowUpdate スイッチを true に設定した場合、ldap_cachemgr デーモンは passwd-cmd の serviceAuthenticationMethod パラメータに定義された認証方式を使用して LDAP サーバーにバインドします (この認証方式が定義されている場合)。このパラメータが存在しない場合、authenticationMethod が使用されます。デーモンは認証方式 none を使用しません。

次に示す例は、クライアントプロファイルの 1 セクションです。ここで、ユーザーはディレクトリサーバーへの認証に sasl/digest-MD5 を使用しますが、パスワードの変更には SSL セッションを使用します。

```
serviceAuthenticationMethod=pam_ldap:sasl/digest-MD5
serviceAuthenticationMethod=passwd-cmd:tls:simple
```

プラグイン可能な認証方式

PAM フレームワークを使用することで、pam_unix、pam_krb5、pam_ldap などの中からサーバー認証サービスを選択できます。

ユーザー別の認証方式を使用する場合、上記の 3 つの認証サービスの中で最も強力な pam_krb5 を有効にする必要があります。pam_krb5(5) および『Solaris のシステム管理 (セキュリティサービス)』を参照してください。

ユーザー別の認証が有効でない場合でも、pam_krb5 認証システムを使用できます。プロキシまたは匿名の資格レベルを使用してディレクトリサーバーのデータにアクセスする場合、ディレクトリデータへのアクセスをユーザーごとに制限することはできません。

匿名またはプロキシ認証方式を使用する場合は、pam_unix を使用するよりも、柔軟性の向上、より強力な認証方式、およびアカウント管理機能を備えた、pam_ldap を使用することをお勧めします。

pam_unix サービスモジュール

pam.conf(4) ファイルを変更していない場合、デフォルトで pam_unix の機能が有効になっています。

注 – pam_unix モジュールは削除されたので、Solaris ではサポートされていません。その他のサービスモジュールによって、同等またはそれ以上の機能が提供されます。したがって、このマニュアルでは、pam_unix は pam_unix モジュールではなくその同等の機能を指します。

---

pam_unix と同等の機能を提供するモジュールは、次のとおりです。

pam_authtok_check(5)
pam_authtok_get(5)
pam_authtok_store(5)
pam_dhkeys(5)
pam_passwd_auth(5)
pam_unix_account(5)
pam_unix_auth(5)
pam_unix_cred(5)
pam_unix_session(5)

pam_unix は従来の UNIX 認証モデルに従い、次のように動作します。

1. クライアントは、ネームサービスからユーザーの暗号化されたパスワードを取得します。
2. ユーザーは、ユーザーパスワードの入力を求められます。
3. ユーザーのパスワードが暗号化されます。
4. クライアントは、暗号化された 2 つのパスワードを比較して、ユーザーを認証するかどうかを決定します。

pam_unix を使用する場合、次の 2 つの制限が存在します。

- パスワードは、平文を含むほかの暗号化方式ではなく、UNIX crypt 形式で格納する必要があります。

- userPassword 属性は、ネームサービスから読み取り可能でなければなりません。
  たとえば、資格レベルを匿名に設定する場合、すべてのユーザーに対してuserPassword 属性を読み取り可能にする必要があります。同様に、資格レベルを proxy に設定する場合、userPassword 属性の読み取りをプロキシユーザーに許可する必要があります。

注 – pam_unix は、sasl 認証方式 digest-MD5 と互換性がありません。これは、Sun Java System Directory Server では digest-MD5 を使用するためにパスワードを平文で格納する必要があるのに対し、pam_unix ではパスワードを crypt 形式で格納する必要があるためです。

注 – Solaris 10 10/09 リリース以降では、enableShadowUpdate スイッチが true に設定されていると、pam_unix でアカウント管理がサポートされます。リモート LDAP ユーザーアカウントのコントロールは、passwd および shadow ファイルに定義されたローカルユーザーアカウントに適用されるコントロールと同じように適用されます。enableShadowUpdate モードでは、LDAP アカウントについてはシステムが更新を行い、パスワードの有効期限管理とアカウントのロックのために LDAP サーバー上のシャドウデータを使用します。ローカルアカウントのシャドウデータはローカルクライアントシステムに適用されるのに対して、LDAP ユーザーアカウントのシャドウデータはすべてのクライアントシステムのユーザーに適用されます。

パスワードの履歴チェックは、ローカルクライアントに対してのみサポートされ、LDAP ユーザーアカウントに対してはサポートされません。

---

pam_krb5 サービスモジュール

pam_krb5(5) および『Solaris のシステム管理 (セキュリティサービス)』を参照してください。

pam_ldap サービスモジュール

pam_ldap を実装すると、ユーザーは pam_ldap の serviceAuthenticationMethod パラメータに定義された認証方式を使用して LDAP サーバーにバインドします (このパラメータが存在する場合)。このパラメータが存在しない場合、authenticationMethod が使用されます。

pam_ldap が、ユーザーの識別情報および指定されたパスワードでサーバーにバインドできれば、ユーザーが認証されたことになります。

注 – 以前は、pam_ldap アカウント管理を有効にすると、システムにログインする際には、常にすべてのユーザーが認証用にログインパスワードを入力する必要がありました。そのため、rsh、rlogin、ssh などのツールによるパスワードを使用しないログインは失敗します。

一方、pam_ldap(5) を Sun Java System Directory Server DS5.2p4 以降のリリースで使用することで、ユーザーはパスワードを入力せずに、rsh、rlogin、rcp、および ssh を使ってログインできるようになりました。

pam_ldap(5) は変更され、ユーザーのログイン時に Directory Server への認証を実行せずに、アカウントの管理およびユーザーのアカウント状態の取得を実行できるようになりました。Directory Server 上でこの機能を制御するのは、1.3.6.1.4.1.42.2.27.9.5.8 です。これは、デフォルトで有効になっています。

この制御をデフォルト以外に変更する場合は、Directory Server 上でアクセス制御情報 (ACI) を追加します。

```
dn: oid=1.3.6.1.4.1.42.2.27.9.5.8,cn=features,cn=config
objectClass: top
objectClass: directoryServerFeature
oid:1.3.6.1.4.1.42.2.27.9.5.8
cn:Password Policy Account Usable Request Control
aci: (targetattr != "aci")(version 3.0; acl "Account Usable";
     allow (read, search, compare, proxy)
     (groupdn = "ldap:///cn=Administrators,cn=config");)
creatorsName: cn=server,cn=plugins,cn=config
modifiersName: cn=server,cn=plugins,cn=config
```

pam_ldap は、userPassword 属性を読み取りません。このため、pam_unix を使用するほかのクライアントが存在しない限り、userPassword 属性の読み取りアクセス権を付与する必要はありません。また、pam_ldap は、認証方式 none をサポートしません。このため、クライアントが pam_ldap を使用できるように、serviceAuthenticationMethod または authenticationMethod 属性を定義する必要があります。詳細については、pam_ldap(5) のマニュアルページを参照してください。

注意 – 認証方式 simple を使用する場合、第三者がネットワーク上で userPassword 属性を読み取ることができます。

225 ページの「pam_ldap に対応した pam.conf ファイルの例」を参照してください。

次の表に、pam_unix、pam_ldap、および pam_krb5 の主な相違点を示します。

表 9–5 pam_unix、pam_ldap、および pam_krb5 の使用における LDAP の認証動作

	pam_unix	pam_ldap	pam_krb5
パスワードの送信	passwd サービス認証方式を使用します	passwd サービス認証方式を使用します	パスワードではなく、Kerberos シングルサインオンテクノロジを使用します
新規パスワードの送信	暗号化します	暗号化しません (TLS を使用しない場合)	Kerberos を使用します。パスワードはネットワークに送信されません
新規パスワードの格納	crypt 形式	Sun Java System Directory Server で定義されたパスワード格納スキーマ	パスワードは Kerberos を使って管理されます
パスワードの読み取り	はい	いいえ	いいえ
パスワード変更後の sasl/digestMD5 の互換性	ありません。パスワードは平文では格納されません。ユーザーを認証できません。	あります。デフォルトのストレージスキーマが平文 (clear) に設定されていれば、ユーザーを認証できます。	ありません。sasl/GSSAPI が使用されます。Kerberos kdc を使用して LDAP ディレクトリサーバー内のパスワードデータベースを管理する場合を除き、パスワードがネットワーク上に送信されることも、ディレクトリサーバーに保存されることもありません。
パスワードポリシーのサポート	あります。enableShadowUpdate を true に設定する必要があります。	はい (構成されている場合)。	pam_krb5(5)、Kerberos V5 アカウント管理モジュールを参照してください。

PAM およびパスワードの変更

パスワードを変更するには、passwd コマンドを使用します。enableShadowUpdate スイッチが true に設定されていない場合、userPassword 属性がユーザーによって書き込み可能である必要があります。enableShadowUpdate スイッチが true に設定されている場合、管理者資格で userPassword 属性を更新できる必要があります。passwd-cmd 用の serviceAuthenticationMethod が、この操作の authenticationMethod を無効にすることに留意してください。使用する認証方式によっては、現行のパスワードの暗号化解除がネットワーク上で行われる場合があります。

pam_unix の場合、UNIX crypt フォーマットを使用して新しい userPassword 属性が暗号化されタグ付けされてから、LDAP への書き込みが行われます。このため、新規パスワードは、サーバーへのバインドに使用される認証方式に関係なく、ネットワーク上で暗号化されます。詳細は、pam_authtok_store(5) のマニュアルページを参照してください。

enableShadowUpdate スイッチが true に設定されている場合、ユーザーのパスワードが変更されると、pam_unix も関連するシャドウ情報を更新します。pam_unix は、ローカルユーザーのパスワードが変更されたときに pam_unix が更新するローカル shadow ファイル内の同じ shadow フィールドを更新します。

Solaris 10 ソフトウェアリリース以降、pam_ldap ではパスワード更新がサポートされません。pam_ldap パスワード更新機能は、以前に推奨されていた pam_authtok_store と server_policy オプションによって置き換えられます。pam_authtok_store を使用すると、新しいパスワードは平文で LDAP サーバーに送信されます。このため、機密性を保つために TLS を使用する必要があります。TLSを使用しない場合、新しい userPassword が漏洩する危険性があります。Sun Java System Directory Server でタグ付けされていないパスワードを設定すると、passwordStorageScheme 属性を使用してパスワードが暗号化されます。passwordStorageScheme の詳細については、ご使用のバージョンの Sun Java System Directory Server の『管理者ガイド』のユーザーアカウントの管理に関する節を参照してください。

注 – passwordStorageScheme 属性を設定する際、次の構成上の問題を考慮する必要があります。pam_unix を使用する NIS、NIS+、またはほかのクライアントがリポジトリとして LDAP を使用する場合、passwordStorageScheme に対して crypt を実行する必要があります。また、Sun Java System Directory Server で sasl/digest-MD5 に対して pam_ldap を使用する場合、passwordStorageScheme を平文に設定する必要があります。

アカウント管理

アカウントおよびパスワードの管理システムとして pam_krb5 を選択すると、アカウント、パスワード、アカウントロックアウト、およびアカウント管理のその他の詳細情報がすべて Kerberos 環境により管理されます。pam_krb5(5) および『Solaris のシステム管理 (セキュリティサービス)』を参照してください。

pam_krb5 を使用しない場合は、LDAP ネームサービスを構成して、Sun Java System Directory Server のパスワードおよびアカウントロックアウトポリシーのサポートを活用できます。pam_ldap(5) を構成して、ユーザーアカウント管理をサポートすることが可能です。passwd(1) を正しい PAM 構成で使用すると、Sun Java System Directory Server パスワードポリシーによるパスワードの構文規則が適用されます。

次のアカウント管理機能が、pam_ldap(5) でサポートされます。 これらの機能は、Sun Java System Directory Server のパスワードとアカウントのロックアウトポリシー構成を利用しています。必要な機能を必要な数だけ利用できます。

- 古くなったり、有効期限の切れたパスワードを通知する

  パスワードは、予定にしたがって変更する必要があります。パスワードを定められた期間内に変更しないとそのパスワードは無効になります。期限切れのパスワードでは、ユーザーが認証されません。

  期限切れの警告期間内のログイン時には、常に警告メッセージを表示します。メッセージには期限切れまでの日数と時間が表示されます。

- パスワードの構文チェック

  新規パスワードは、最小文字数の条件を満たしている必要があります。また、ユーザーのディレクトリエントリにある uid、cn、sn、および mail と同じ値をパスワードに設定することはできません。

- パスワードの履歴チェック

  パスワードの再利用はできません。以前使われていたパスワードに変更しようとすると、passwd(1) コマンドは失敗します。LDAP 管理者は、サーバーの履歴リストに保持するパスワードの数を設定することができます。

- ユーザーアカウントのロックアウト

  認証の失敗が設定された回数に達すると、そのユーザーアカウントはロックアウトされます。管理者がアカウントを非アクティブにした場合も、そのユーザーはロックアウトされます。アカウントのロックアウト期間が経過するか、管理者が再びアカウントをアクティブにするまで、認証は成功しません。

注 – 以上のアカウント管理機能は、Sun Java System Directory Server だけで有効です。サーバー上のパスワードとアカウントのロックアウトポリシーの構成についての詳細は、ご使用のバージョンの Sun Java System Directory Server の『管理者ガイド』の「ユーザーアカウントの管理」の章を参照してください。227 ページの「アカウント管理のために pam_ldap を構成した pam.conf ファイル例」も参照してください。proxy アカウントでは、アカウント管理を有効にしないでください。

Sun Java System Directory Server 上でパスワードとアカウントのロックアウトポリシーを構成する前に、すべてのホスト上で pam_ldap アカウント管理に基づいた「最新の」 LDAP クライアントが使われていることを確認します。

さらに、クライアントが正しい構成の pam.conf(4) ファイルを保持していることを確認します。正しい構成ファイルを保持していない場合、 LDAP ネームサービスは proxy やユーザーパスワードが期限切れの時に動作しません。

注 – 以前は、pam_ldap アカウント管理を有効にすると、システムにログインする際には、常にすべてのユーザーが認証用にログインパスワードを入力する必要がありました。そのため、rsh、rlogin、ssh などのツールによるパスワードを使用しないログインは失敗します。

一方、pam_ldap(5) を Sun Java System Directory Server DS5.2p4 以降のリリースで使用することで、ユーザーはパスワードを入力せずに、rsh、rlogin、rcp、および ssh を使ってログインできるようになりました。

pam_ldap(5) は変更され、ユーザーのログイン時に Directory Server への認証を実行せずに、アカウントの管理およびユーザーのアカウント状態の取得を実行できるようになりました。Directory Server 上でこの機能を制御するのは、1.3.6.1.4.1.42.2.27.9.5.8 です。これは、デフォルトで有効になっています。

この制御をデフォルト以外に変更する場合は、Directory Server 上でアクセス制御情報 (ACI) を追加します。

```
dn: oid=1.3.6.1.4.1.42.2.27.9.5.8,cn=features,cn=config
objectClass: top
objectClass: directoryServerFeature
oid:1.3.6.1.4.1.42.2.27.9.5.8
cn:Password Policy Account Usable Request Control
aci: (targetattr != "aci")(version 3.0; acl "Account Usable";
     allow (read, search, compare, proxy)
     (groupdn = "ldap:///cn=Administrators,cn=config");)
creatorsName: cn=server,cn=plugins,cn=config
modifiersName: cn=server,cn=plugins,cn=config
```

pam_unix を使用したアカウント管理

Solaris 10 10/09 リリース以降では、enableShadowUpdate スイッチが使用できます。enableShadowUpdate を true に設定すると、LDAP はアカウント管理のファイルネームサービスと同じ機能を提供します。

クライアントで enableShadowUpdate スイッチが true に設定されている場合、ローカルアカウントで使用可能なアカウント管理機能が LDAP アカウントでも使用できます。この機能には、パスワードの有効期限管理、アカウントの有効期限管理および通知、ログインに失敗したアカウントのロックなどが含まれます。また、passwd コマンドの -dluNfnwx オプションが LDAP でサポートされるようになりました。これにより、passwd コマンドの完全な機能と、ファイルネームサービスの pam_unix* モジュールが LDAP ネームサービスでサポートされます。enableShadowUpdate スイッチは、ファイルと LDAP スコープの両方に定義されたユーザーに対して一貫したアカウント管理を実装する 1 つの方法を提供します。

ユーザーが自身のアカウント管理データを変更するのを防ぐため、また、パスワードポリシーを回避するために、LDAP サーバーは、サーバー上にあるユーザー自身のシャドウデータに対するユーザーの書き込みアクセスを防止するように構成さ

れています。管理者資格を持つ管理者は、クライアントシステムに対してシャドウデータの更新を実行します。しかし、この構成は、ユーザーによるパスワードの変更が必要な pam_ldap モジュールと競合してしまいます。したがって、pam_ldap と pam_unix によるアカウント管理には互換性がありません。

注意 – 同じ LDAP ネームドメインで pam_ldap と pam_unix の両方を使用しないでください。すべてのクライアントが pam_ldap を使用するか、またはすべてのクライアントが pam_unix を使用します。この制限により、専用の LDAP サーバーが必要になる場合があります。たとえば、Web または電子メールアプリケーションでは、ユーザーが LDAP サーバー上にあるパスワードを変更する必要がある場合があります。

また、enableShadowUpdate の実装では、管理者資格 (adminDN と adminPassword) が、各クライアント上にローカルで格納されている必要があります。adminPassword は暗号化されており、ldap_cachemgr デーモンによって /var/ldap/ldap_client_cred ファイルからのみ読み取りが可能ですが、管理者資格を保護するために細心の注意が必要です。管理者資格を保護するために、サーバーのディレクトリマネージャー (cn=directory manager) とは異なる情報を使用してください。別の保護方法としては、serviceAuthenticationMethod を構成する際に、passwd-cmd サービスに対して tls:simple またはより保護されたレベルを使用します。このようにすることで、adminPassword の値が平文で送信されず、漏洩に対して脆弱になりません。

pam_ldap をアカウント管理に対して使用するのとは異なり、pam_unix をアカウント管理に対して使用する場合は、/etc/pam.conf ファイルの変更は必要ありません。デフォルトの /etc/pam.conf ファイルで十分です。

◆ ◆ ◆ 第 10 章

10 LDAP ネームサービスの計画要件 (手順)

この章では、サーバーとクライアントの設定およびインストール処理を開始する前に実行する必要のある上流工程の計画について説明します。

この章の内容は次のとおりです。

LDAP の計画の概要

LDAP クライアントプロファイルは、LDAP クライアントが使用する構成情報の集合体です。 LDAP クライアントは、このプロファイルを使用して、サポートする LDAP サーバーについての LDAP ネームサービス情報にアクセスします。この章では、LDAP ネームサービスのさまざまな分野での計画方法を説明します。その中には、ネットワークモデル、ディレクトリ情報ツリー、セキュリティーモデル、さまざまなプロファイル属性のデフォルト値、およびデータ生成の準備が含まれます。

LDAP ネットワークモデルの計画

可用性およびパフォーマンスを考慮すると、企業規模のネットワークの各サブネットが LDAP サーバーを独自に保持して、サブネット内のすべての LDAP クライアントにサービスを提供する方法が最善です。これらのサーバーの 1 つだけをマスター LDAP サーバーにする必要があります。残りはすべてマスターサーバーの複製にできます。

ネットワーク構成を計画する前に、使用可能なサーバーの数、クライアントがサーバーにアクセスする方法、複数のサーバーへのアクセス順序について考慮する必要があります。サブネットごとに 1 つのサーバーが存在する場合、`defaultServerList` 属性を使用してすべてのサーバーのリストを作成し、LDAP クライアントからアクセス順序をソートおよび操作できます。速度またはデータ管理上の理由から、特定の順序でサーバーにアクセスする必要がある場合は、`preferredServerList` 属性を使用してサーバーへの固定されたアクセス順序を定義します。マスターサーバーをこれらのリストに配置しないことで、マスターサーバーへの負荷を軽減できます。

さらに、サーバーおよびネットワーク構成を計画する際に考慮するに値する 3 つの属性があります。`bindTimeLimit` 属性は TCP 接続要求のタイムアウト値の設定に使用されます。`searchTimeLimit` 属性は LDAP 検索操作のタイムアウト値の設定に、`profileTTL` 属性は LDAP クライアントによるサーバーからのプロファイルのダウンロード頻度の制御に、それぞれ使用できます。速度が遅いか不安定なネットワークの場合、`bindTimeLimit` および `searchTimeLimit` 属性にデフォルト値より大きい値を設定することが必要な場合があります。配備の初期テスト段階で、`profileTTL` 属性値を引き下げて、頻繁に行われる LDAP サーバー内のプロファイルの変更をクライアントが取得するようにしてもよいでしょう。

ディレクトリ情報ツリー (DIT) の計画

LDAP ネームサービスは、デフォルトのディレクトリ情報ツリー (DIT) および関連するデフォルトのスキーマを保持します。たとえば、`ou=people` コンテナには、ユーザーアカウント、パスワード、およびシャドウ情報が含まれます。`ou=hosts` コンテナには、ネットワーク内のシステムに関する情報が含まれます。`ou=people` コンテナ内の各エントリは、`objectclass` の `posixAccount` および `shadowAccount` のエントリになります。

デフォルト DIT は、巧みに設計されたディレクトリ構造であり、オープンな標準に基づいています。これは、ほとんどのネームサービスのニーズに応えるもので、変更せずに使用することが推奨されています。デフォルト DIT の使用を選択する場合、決定する必要があるのは、ディレクトリツリー内のどのノード (起点識別名) から、指定されたドメインのネームサービス情報を検索するかという点だけです。このノードは、`defaultSearchBase` 属性を使用して指定されます。さらに、`defaultSearchScope` 属性を設定して、ネームサービスが実行する検索範囲をク

ライアントに指定することもできます。検索範囲には、識別名 (DN) 内の 1 レベルだけを検索するか (one)、DN 内のサブツリー全体を選択するか (sub) を指定できます。

ただし、既存の DIT を利用する場合でも、ディレクトリツリー内に散在するネームサービスデータを使用してより複雑な DIT を処理する場合でも、LDAP ネームサービスにより高度な柔軟性が求められる場合があります。たとえば、ユーザーアカウントエントリがツリーの別の場所に存在する場合があります。クライアントプロファイル内で `serviceSearchDescriptor`、`attributeMap`、および `objectclassMap` 属性を設定して、これらの状況に対応します。

サービス検索記述子を使用して、特定のサービスのデフォルト検索ベース、検索範囲、および検索フィルタを置き換えることができます。146 ページの「サービス検索記述子 (SSD) とスキーママッピング」を参照してください。

`AttributeMap` および `ObjectclassMap` 属性を使用して、スキーママッピングを行うことができます。これらの属性を使用すると、既存の DIT で LDAP ネームサービスを動作させることができます。たとえば、`posixAccount` オブジェクトクラスを既存のオブジェクトクラス `myAccount` へマップできます。`posixAccount` オブジェクトクラス内の属性を `myAccount` オブジェクトクラス内の属性へマップできます。

複数のディレクトリサーバー

複数の LDAP サーバーで 1 つの DIT を構成することも可能です。たとえば、DIT のいくつかのサブツリーを、ほかの LDAP サーバー上に配置できます。この場合、LDAP サーバーは、既知ではあるが自身のデータベース内に存在しないネームデータを求める LDAP クライアントを、別のサーバーに委ねることができます。この種の DIT 構成を計画する場合、LDAP ネームサービスがサーバー参照に従ってネームサービス検索を続行することを示すように、クライアントのプロファイル属性 `followReferrals` を設定する必要があります。ただし可能であれば、指定されたドメインのネームデータすべてを単独のディレクトリサーバー上に配置するのが最善です。

クライアントが通常は読み取り専用の複製にアクセスし、必要な場合にのみ読み取り/書き込み可能なマスターサーバーへの参照を利用する場合、参照が役に立ちます。この方法では、要求が複製により処理されるため、マスターサーバーに過度の負荷がかかることはありません。

ほかのアプリケーションとのデータ共有

LDAP を最大限に活用するには、論理エントリごとに 1 つの LDAP エントリが存在する必要があります。たとえば、ユーザーは、企業白書に関する情報だけでなく、Solaris アカウント情報やアプリケーション固有のデータも保持できます。`posixAccount` および `shadowAccount` は補助オブジェクトクラスであるため、これらのデータをディレクトリ内のエントリに追加できます。このため、注意深い計画、設定、および管理が必要になります。

ディレクトリ接尾辞の選択

適切なディレクトリ接尾辞の選択方法については、Sun Java System Directory Server (以前の名称は Sun ONE Directory Server) のマニュアルを参照してください。

LDAP と複製サーバー

複製サーバーを設定する場合、次の 3 つの方法が存在します。

- 単一マスター複製 (Single-master replication)
- 浮動マスター複製 (Floating-master replication)
- 複数マスター複製 (Multi-master replication)

「単一マスター」

単一マスター複製では、指定されたパーティションまたはパーティション化されていないネットワークに対して、1 つのマスターサーバーだけが、ディレクトリエントリの書き込み可能なコピーを保持します。複製サーバーは、ディレクトリエントリの読み込み専用コピーを保持します。複製とマスターの両方が検索、比較、およびバインド操作を実行できますが、書き込み操作を実行できるのはマスターサーバーだけです。

単一マスター複製の不利な点は、マスターサーバーで単一点障害が発生した場合です。マスターサーバーがダウンした場合、どの複製サーバーからも書き込み操作を実行できません。

「浮動マスター」

浮動マスターは、指定されたパーティション化されたネットワークまたはパーティション化されていないネットワークに対し、書き込み権限を保持するマスターサーバーは常に 1 つだけである点で、単一マスターを使用する場合と似ています。ただし浮動マスターを使用すると、マスターサーバーがダウンした場合、アルゴリズムにより複製の 1 つが自動的にマスターサーバーに変化します。

浮動マスター複製の不利な点は、ネットワークがパーティション化され、どちらの側のパーティション上の複製もマスターになった場合、ネットワークを再結合する際、新規マスター間の調整が非常に複雑になり得ることです。

「複数マスター」

複数マスター複製では、ディレクトリエントリデータの独自の読み取り/書き込み複製を保持する、複数のマスターサーバーが存在します。複数マスターを使用すると、単一点障害を防ぐことができますが、サーバー間で更新による競合が発生する可能性があります。つまり、2 つのマスター上でエントリの属性が同時に変更される場合、競合による障害の解決ポリシー (最後の書き込みを優先するなど) の適用が必要になります。

複製サーバーの設定方法については、ご使用のバージョンの Sun Java System Directory Server の『管理者ガイド』を参照してください。

LDAP セキュリティーモデルの計画

セキュリティーモデルを計画する場合、最初に、LDAP クライアントが LDAP サーバーとの通信に使用する識別情報を考慮する必要があります。たとえば、企業全体でシングルサインオンソリューションを使用するかどうか、ネットワークにパスワードを送信しないかどうか、ネットワークに流れるデータの暗号化、およびディレクトリサーバーで生成される制御データへのユーザー別のアクセス機能などを決定する必要があります。また、強力な認証を使用してネットワーク上を流れるユーザーパスワードを保護するかどうか、また LDAP クライアントと LDAP サーバー間のセッションを暗号化して送信される LDAP データを保護する必要があるかなども決定する必要があります。

セキュリティーモデルの計画には、プロファイル内の `credentialLevel` および `authenticationMethod` 属性が使用されます。`credentialLevel` には、`anonymous`、`proxy`、`proxy anonymous`、および `self` の 4 つの資格レベルのうちの 1 つを指定できます。LDAP ネームサービスのセキュリティー概念については、152 ページの「LDAP ネームサービスのセキュリティーモデル」を参照してください。

注 – 以前は、pam_ldap アカウント管理を有効にすると、システムにログインする際には、常にすべてのユーザーが認証用にログインパスワードを入力する必要がありました。そのため、rsh、rlogin、ssh などのツールによるパスワードを使用しないログインは失敗します。

一方、pam_ldap(5) を Sun Java System Directory Server DS5.2p4 以降のリリースで使用することで、ユーザーはパスワードを入力せずに、rsh、rlogin、rcp、および ssh を使ってログインできるようになりました。

pam_ldap(5) は変更され、ユーザーのログイン時に Directory Server への認証を実行せずに、アカウントの管理およびユーザーのアカウント状態の取得を実行できるようになりました。Directory Server 上でこの機能を制御するのは、1.3.6.1.4.1.42.2.27.9.5.8 です。これは、デフォルトで有効になっています。

この制御をデフォルト以外に変更する場合は、Directory Server 上でアクセス制御情報 (ACI) を追加します。

```
dn: oid=1.3.6.1.4.1.42.2.27.9.5.8,cn=features,cn=config
objectClass: top
objectClass: directoryServerFeature
oid:1.3.6.1.4.1.42.2.27.9.5.8
cn:Password Policy Account Usable Request Control
aci: (targetattr != "aci")(version 3.0; acl "Account Usable";
     allow (read, search, compare, proxy)
     (groupdn = "ldap:///cn=Administrators,cn=config");)
creatorsName: cn=server,cn=plugins,cn=config
modifiersName: cn=server,cn=plugins,cn=config
```

注 – 企業全体のシングルサインオンソリューションとして pam_krb5 および Kerberos を有効にする場合、セッション開始時にのみログインパスワードを必要とするシステムを設計できます。詳細については、『Solaris のシステム管理 (セキュリティサービス)』を参照してください。一般に、Kerberos を有効にする場合には、DNS も有効にする必要があります。詳細については、このマニュアルの DNS に関する章を参照してください。

セキュリティーモデルを計画する際の主要な決定事項を次に示します。

- Kerberos およびユーザー別の認証を使用するか。
- LDAP クライアントは、どの資格レベルおよび認証方式を使用するか。
- TLS を使用するか。
- NIS や NIS+ との下位互換性を必要とするか。言い換えれば、クライアントは pam_unix または pam_ldap を使用するか。
- サーバーの passwordStorageScheme 属性をどのように設定するか。

- アクセス制御情報をどのように設定するか。

  ACI の詳細については、ご使用のバージョンの Sun Java System Directory Server の『管理者ガイド』を参照してください。
- アカウント管理を実行するために、クライアントが `pam_unix` または `pam_ldap` を使用するか。

LDAP 用のクライアントプロファイルおよびデフォルト属性値の計画

前述の計画手順 (ネットワークモデル、DIT、およびセキュリティーモデル) を理解することにより、次のプロファイル属性の値についてアイデアを得ることができるでしょう。

- `cn`
- `defaultServerList`
- `preferredServerList`
- `bindTimeLimit`
- `searchTimeLimit`
- `profileTTL`
- `defaultSearchBase`
- `defaultSearchScope`
- `serviceSearchDescriptor`
- `attributeMap`
- `objectclassMap`
- `followReferrals`
- `credentialLevel`
- `authenticationMethod`
- `serviceCredentialLevel`
- `serviceAuthenticationMethod`

上記の属性の中で、必須属性は `cn`、`defaultServerList`、および `defaultSearchBase` だけです。これらの属性には、デフォルト値は存在しません。残りの属性はオプションであり、デフォルト値がないオプションも存在します。

LDAP クライアントの設定の詳細については、第 12 章「LDAP クライアントの設定 (手順)」を参照してください。

LDAP データ生成の計画

データを使用して LDAP サーバーを生成する場合、適切な DIT およびスキーマを使用して LDAP サーバーを構成したあとで、新しい ldapaddent ツールを使用します。このツールは、対応する /etc ファイルから LDAP コンテナ内にエントリを作成します。このツールを使用して、次のデータタイプ用のコンテナ内にデータを生成することができます。 aliases、auto_*、bootparams、ethers、group、hosts (IPv6 アドレスを含
む)、netgroup、netmasks、networks、passwd、shadow、protocols、publickey、rpc、および services。

デフォルトでは、ldapaddent は標準入力からこのデータを読み取って、コマンド行で指定されたデータベースに関連付けられた LDAP コンテナに追加します。ただし、データを読み取る入力ファイルは、-f オプションを使用して指定できます。

エントリはクライアントの構成に基づき、ディレクトリ内に格納されるため、LDAP ネームサービスを使用するようにクライアントを構成する必要があります。

パフォーマンスを向上させるため、次の順序でデータベースをロードしてください。

1. passwd データベースの次に shadow データベース
2. networks データベースの次に netmasks データベース
3. bootparams データベースの次に ethers データベース

オートマウントエントリを追加する場合、データベース名は auto_* (たとえば auto_home) の形式で指定します。

別のホストの /etc ファイルを LDAP サーバーに追加する場合、それらすべてを 1 つの /etc ファイルにマージして、1 台のホスト上で ldapaddent を使用して追加できます。あるいは、各ホストが LDAP クライアントとして構成済みであることを想定して各ホストで ldapaddent を実行することもできます。

使用するネームサービスデータが NIS サーバー上にすでに存在し、データを LDAP ネームサービス用の LDAP サーバーに移動する場合、ypcat (または niscat) コマンドを使用して NIS マップをファイル内にダンプします。続いて、これらのファイルに対して ldapaddent を実行してデータを LDAP サーバーに追加します。

注 – ldapaddent は LDAP クライアント上でしか実行できません。

次の作業は、テーブルが yp クライアントから抽出されることを想定しています。

▼ ldapaddent を使用して hosts エントリを持つサーバーを生成する方法

1. idsconfig を使用し、**Sun Java System Directory Server** が設定されていることを確認します。

2. クライアントマシンで、スーパーユーザーになるか、同等の役割になります。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

3. そのマシンを **LDAP** クライアントに設定します。

   ```
   # ldapclient init -a profileName=new -a domainName=west.example.com \
   192.168.0.1
   ```

4. データを指定してサーバーを生成します。

   ```
   # ldapaddent -D “cn=directory manager” -f /etc/hosts hosts
   ```

   パスワードの入力を求められます。

   この例では、ldapaddent は、プロファイル「new」で設定されている認証方式を使用します。「simple」を選択した場合、パスワードは平文で送信されます。詳細については、ldapaddent(1M) のマニュアルページを参照してください。

◆ ◆ ◆ 第 11 章

LDAP クライアントと Sun Java System Directory Server の設定 (手順)

この章では、Sun Java System Directory Server (以前の名称は Sun ONE Directory Server) を構成して、Solaris LDAP ネームサービスクライアントのネットワークをサポートする方法について説明します。この情報は、Sun Java System Directory Server に固有の情報です。ディレクトリサーバーのインストールおよび構成については、Sun Java Enterprise System に収められている Sun Java System Directory Server のマニュアルを参照してください。

注 – Sun Java System Directory Server を構成して Solaris LDAP クライアントを使用する前に、Sun Java System Directory Server に付属するインストールおよび構成のマニュアルで説明されているすべての手順を実行しておく必要があります。

注 – ディレクトリサーバー (LDAP サーバー) をそのクライアントとして使用することはできません。

この章の内容は次のとおりです。

idsconfig を使用した Sun Java System Directory Server の構成

サーバーのインストール用チェックリストの作成

サーバーのインストール時に定義する重要な変数を使用して、次に示すようなチェックリストを作成してから、idsconfig を起動してください。221 ページの「記入用のチェックリスト」で提供されるチェックリストを使用できます。

注 – 次の情報は、以降の LDAP 関連の章に示されるすべての例の基礎になります。サンプルのドメインは、全国規模で店舗を展開する部品会社である Example, Inc. のものです。例の中では、その West Coast Division (ドメインは west.example.com) を対象に説明します。

表 11–1　サーバーで定義する変数

変数	サンプルネットワークの定義
インストールしたディレクトリサーバーインスタンスのポート番号	389 (デフォルト)
サーバー名	**myserver** (完全指定ドメイン名 myserver.west.example.com または 192.168.0.1)
複製サーバー (IP 番号: ポート番号)	**192.168.0.2** [myreplica.west.example.com 用]
ディレクトリマネージャー	cn=directory manager (デフォルト)
サービスされるドメイン名	**west.example.com**
クライアント要求の処理がタイムアウトするまでの時間 (秒)	**-1-**
各検索要求で返されるエントリの最大数	**-1-**

注 – defaultServerList または preferredServerList の定義でホスト名を使用する場合、ホストの検索に LDAP を使用してはなりません。これは、/etc/nsswitch.conf の hosts 行に ldap を含めることはできないことを意味します。

表 11–2　クライアントプロファイルで定義する変数

変数	サンプルネットワークの定義
プロファイル名 (デフォルト名は「default」)	**WestUserProfile**

表 11–2 クライアントプロファイルで定義する変数 (続き)

変数	サンプルネットワークの定義
サーバーリスト (デフォルトはローカルサブネット)	**192.168.0.1**
優先されるサーバーリスト (優先順に記載)	**none**
検索範囲 (検索するディレクトリツリーレベルの数、「One」 (デフォルト) または「Sub」)	**one** (デフォルト)
サーバーへのアクセスに使用する資格。デフォルトは anonymous	**proxy**
参照に従うかどうか。(メインサーバーが使用できない場合の別のサーバーへのポインタ)。デフォルトは no	**Y**
検索時にサーバーが情報を返すまでの待機時間の制限 (デフォルトは 30)	**default**
サーバーとの通信時のバインド時間の制限 (デフォルトは 10 秒)	**default**
認証方式。デフォルトは none	**simple**

注 – クライアントプロファイルはドメインごとに定義されます。指定されたドメインで、1 つ以上のプロファイルを定義する必要があります。

属性インデックス

idsconfig が作成する次の属性リストにより、パフォーマンスが向上します。

```
membernisnetgroup    pres,eq,sub
nisnetgrouptriple    pres,eq,sub
ipHostNumber         pres,eq,sub
uidNumber            pres,eq
gidNumber            pres,eq
ipNetworkNumber      pres,eq
automountkey         pres,eq
oncRpcNumber         pres,eq
```

スキーマ定義

idsconfig(1M) により、必要なスキーマ定義が自動的に追加されます。LDAP 管理に精通しているユーザー以外、サーバースキーマを手動で変更してはなりません。LDAP ネームサービスの使用するスキーマ拡張リストについては、第 14 章「LDAP の一般的なリファレンス」を参照してください。

インデックス表示の使用

Sun Java System Directory Server のインデックス表示機能は、仮想リスト表示 (VLV) とも呼ばれます。クライアントは、インデックス表示を使用して、非常に長いリストから、グループや複数のエントリを選択して表示できます。このため、各クライアントの検索処理時間を短縮できます。インデックス表示により最適化かつ事前定義された検索パラメータが提供されるため、Solaris LDAP ネームクライアントは、さまざまなサービスから特定の情報により素早くアクセスできるようになります。インデックス表示を作成しない場合、サーバーが検索時間を制限するため、またはエントリの数を特定される場合があるために、クライアントが指定されたタイプのエントリすべてを取得できない場合があります。

VLV はディレクトリサーバー上に構成されるため、プロキシユーザーはこれらのインデックスに読み取りアクセス権限を保持します。

Sun Java System Directory Server 上でインデックス表示を構成する前に、これらのインデックスの使用に関連したパフォーマンスのコストを検討してください。詳細については、ご使用のバージョンの Sun Java System Directory Server の『管理者ガイド』を参照してください。

idsconfig は、複数の VLV インデックスのエントリを作成します。サーバーを停止し、実際の VLV インデックスを作成するには、directoryserver スクリプトを使用してください。詳細については、idsconfig(1M) および directoryserver(1M) のマニュアルページを参照してください。idsconfig によって作成された VLV エントリと、実行する必要がある、対応する directoryserver コマンドの構文を確認するには、idsconfig コマンドの出力を参照してください。idsconfig 出力の例は、187 ページの「idsconfig 設定の例」を参照してください。

サービス検索記述子を使用してさまざまなサービスへのクライアントアクセスを変更する

サービス検索記述子 (SSD) を使用すると、LDAP 内の指定された操作のデフォルト検索要求を、ユーザーが定義した検索に変更できます。たとえば、これまでカスタマイズしたコンテナ定義や別のオペレーティングシステムで LDAP を使用してきた

ユーザーが、最新の Solaris リリースに移行する場合などに、SSD は特に役に立ちます。SSD を使用すると、既存の LDAP データベースおよびデータを変更せずに、Solaris LDAP ネームサービスを構成できます。

idsconfig を使用して SSD を変更する

前出の Example, Inc. が LDAP を構成済みで、ユーザーを ou=Users コンテナに格納しているものとします。これを最新の Solaris リリースにアップグレードします。定義では、Solaris LDAP クライアントは、ユーザーエントリが ou=People コンテナに格納されていると想定しています。このままでは、LDAP クライアントは passwd サービス検索時に DIT の ou=people レベルを検索するため、適切な値を検出できません。

この問題を解決する手のかかる方法の 1 つは Example, Inc. の既存の DIT を完全に置き換え、Example, Inc. のネットワーク上の既存アプリケーションすべてを書き換えて、新規 LDAP ネームサービスとの互換性を持たせる方法です。別のはるかに望ましい解決策は、SSD を使用して、LDAP クライアントに対しユーザー情報をデフォルトの ou=people コンテナ内ではなく ou=Users コンテナ内で検索するよう指示する方法です。

idsconfig を使用して、Sun Java System Directory Server の構成時に必要な SSD を定義します。プロンプト行は次のようになります。

```
Do you wish to setup Service Search Descriptors (y/n/h? y
  A  Add a Service Search Descriptor
  D  Delete a SSD
  M  Modify a SSD
  P  Display all SSD's
  H  Help
  X  Clear all SSD's

  Q  Exit menu
Enter menu choice: [Quit] a
Enter the service id: passwd
Enter the base: service ou=user,dc=west,dc=example,dc=com
Enter the scope: one[default]
  A  Add a Service Search Descriptor
  D  Delete a SSD
  M  Modify a SSD
  P  Display all SSD's
  H  Help
  X  Clear all SSD's

  Q  Exit menu
Enter menu choice: [Quit] p

Current Service Search Descriptors:
==================================
Passwd:ou=Users,ou=west,ou=example,ou=com?

Hit return to continue.
```

```
  A  Add a Service Search Descriptor
  D  Delete a SSD
  M  Modify a SSD
  P  Display all SSD's
  H  Help
  X  Clear all SSD's

  Q  Exit menu
Enter menu choice: [Quit] q
```

idsconfig の実行

注 – idsconfig の実行には特別な権限は不要であり、LDAP ネームサービスクライアントになる必要もありません。182 ページの「サーバーのインストール用チェックリストの作成」 を実行する準備として、Creating a Checklist Based on Your Server Installationで説明したチェックリストを作成してください。idsconfig を、サーバーまたは LDAP ネームサービスクライアントマシンから実行する必要はありません。ネットワーク上の任意の Solaris マシンから idsconfig を実行できます。

注意 – idsconfig は、ディレクトリマネージャーのパスワードを平文で送信します。これを防ぐには、idsconfig をクライアント上ではなくディレクトリサーバー上で実行する必要があります。

▼ idsconfig を使用して Sun Java System Directory Server を構成する方法

1 ターゲットの **Sun Java System Directory Server** が起動して実行中であることを確認してください。

2 idsconfig を実行します。

```
# /usr/lib/ldap/idsconfig
```

この章の冒頭の 182 ページの「サーバーのインストール用チェックリストの作成」で示した、サーバーおよびクライアントのチェックリストの定義を使用した idsconfig の実行例として、例 11–1 を参照してください。

3 表示される質問に答えます。

ユーザー入力のデフォルトは「no」です。質問の詳細を表示する場合は、

h

と入力します。すると、簡単なヘルプが表示されます。

idsconfig によるディレクトリの設定が完了したら、サーバー設定を完了してサーバーをクライアント対応にする前に、サーバー上で指定されたコマンドを実行する必要があります。

idsconfig 設定の例

この節では、多くのデフォルト値を使用した基本的な idsconfig 設定の例を示します。クライアントプロファイルを変更する最も複雑な方法は、SSD を作成する方法です。詳細については、184 ページの「サービス検索記述子を使用してさまざまなサービスへのクライアントアクセスを変更する」を参照してください。

プロンプトの後ろにある [ ] 内のデータは、そのプロンプトのデフォルト値を表しています。デフォルト値を使用する場合は、Return キーを押します。

注 – サマリー画面で空白になっているパラメータは設定されません。

---

idsconfig によるディレクトリの設定が完了したら、サーバー設定を完了してサーバーをクライアント対応にする前に、サーバー上で指定されたコマンドを実行する必要があります。

例 11–1 Example, Inc. ネットワークでの idsconfig の実行

次の例では、サーバーインスタンスが LDAP サーバーに作成された直後に、idsconfig ユーティリティーが実行されます。

```
# usr/lib/ldap/idsconfig
It is strongly recommended that you BACKUP the directory server
before running idsconfig.

Hit Ctrl-C at any time before the final confirmation to exit.

Do you wish to continue with server setup (y/n/h)? [n] y
Enter the JES Directory Server's  hostname to setup: myserver
Enter the port number for iDS (h=help): [389]
Enter the directory manager DN: [cn=Directory Manager]
Enter passwd for cn=Directory Manager :
Enter the domainname to be served (h=help): [west.example.com]
Enter LDAP Base DN (h=help): [dc=west,dc=example,dc=com]
  Checking LDAP Base DN ...
  Validating LDAP Base DN and Suffix ...
  No valid suffixes were found for Base DN dc=west,dc=example,dc=com
```

例 11–1　Example, Inc. ネットワークでの idsconfig の実行　　(続き)

```
Enter suffix to be created (b=back/h=help): [dc=west,dc=example,dc=com]
Enter ldbm database name (b=back/h=help): [west]
  sasl/GSSAPI is not supported by this LDAP server
Enter the profile name (h=help): [default] WestUserProfile
Default server list (h=help): [192.168.0.1]
Preferred server list (h=help):
Choose desired search scope (one, sub, h=help):  [one]
The following are the supported credential levels:
  1  anonymous
  2  proxy
  3  proxy anonymous
  4  self
  5  self proxy
  6  self proxy anonymous
Choose Credential level [h=help]: [1] 2
The following are the supported Authentication Methods:
  1  none
  2  simple
  3  sasl/DIGEST-MD5
  4  tls:simple
  5  tls:sasl/DIGEST-MD5
  6  sasl/GSSAPI
Choose Authentication Method (h=help): [1] 2


Current authenticationMethod: simple
Do you want to add another Authentication Method? n
Do you want the clients to follow referrals (y/n/h)? [n]
Do you want to modify the server timelimit value (y/n/h)? [n] y
Enter the time limit for iDS (current=3600): [-1]
Do you want to modify the server sizelimit value (y/n/h)? [n] y
Enter the size limit for iDS (current=2000): [-1]
Do you want to store passwords in "crypt" format (y/n/h)? [n] y
Do you want to setup a Service Authentication Methods (y/n/h)? [n]
Client search time limit in seconds (h=help): [30]
Profile Time To Live in seconds (h=help): [43200]
Bind time limit in seconds (h=help): [10]
Do you want to enable shadow update (y/n/h)? [n]
Do you wish to setup Service Search Descriptors (y/n/h)? [n]

              Summary of Configuration

  1  Domain to serve               : west.example.com
  2  Base DN to setup              : dc=west,dc=example,dc=com
        Suffix to create           : dc=west,dc=example,dc=com
        Database to create         : west
  3  Profile name to create        : WestUserProfile
  4  Default Server List           : 192.168.0.1
  5  Preferred Server List         :
  6  Default Search Scope          : one
  7  Credential Level              : proxy
  8  Authentication Method         : simple
  9  Enable Follow Referrals       : FALSE
 10  iDS Time Limit                : -1
 11  iDS Size Limit                : -1
```

例 11–1 Example, Inc. ネットワークでの idsconfig の実行 (続き)

```
 12  Enable crypt password storage : TRUE
 13  Service Auth Method pam_ldap  :
 14  Service Auth Method keyserv   :
 15  Service Auth Method passwd-cmd:
 16  Search Time Limit             : 30
 17  Profile Time to Live          : 43200
 18  Bind Limit                    : 10
 19  Enable shadow update          : FALSE
 20  Service Search Descriptors Menu

Enter config value to change: (1-20 0=commit changes) [0]
Enter DN for proxy agent: [cn=proxyagent,ou=profile,dc=west,dc=example,dc=com]
Enter passwd for proxyagent:
Re-enter passwd:

WARNING: About to start committing changes. (y=continue, n=EXIT) y

  1. Changed timelimit to -1 in cn=config.
  2. Changed sizelimit to -1 in cn=config.
  3. Changed passwordstoragescheme to "crypt" in cn=config.
  4. Schema attributes have been updated.
  5. Schema objectclass definitions have been added.
  6. Database west successfully created.
  7. Suffix dc=west,dc=example,dc=com successfully created.
  8. NisDomainObject added to dc=west,dc=example,dc=com.
  9. Top level "ou" containers complete.
  10. automount maps: auto_home auto_direct auto_master auto_shared processed.
  11. ACI for dc=west,dc=example,dc=com modified to disable self modify.
  12. Add of VLV Access Control Information (ACI).
  13. Proxy Agent cn=proxyagent,ou=profile,dc=west,dc=example,dc=com added.
  14. Give cn=proxyagent,ou=profile,dc=west,dc=example,dc=com read permission
      for password.
  15. Generated client profile and loaded on server.
  16. Processing eq,pres indexes:
      uidNumber (eq,pres)   Finished indexing.
      ipNetworkNumber (eq,pres)   Finished indexing.
      gidnumber (eq,pres)   Finished indexing.
      oncrpcnumber (eq,pres)   Finished indexing.
      automountKey (eq,pres)   Finished indexing.
  17. Processing eq,pres,sub indexes:
      ipHostNumber (eq,pres,sub)   Finished indexing.
      membernisnetgroup (eq,pres,sub)   Finished indexing.
      nisnetgrouptriple (eq,pres,sub)   Finished indexing.
  18. Processing VLV indexes:
      west.example.com.getgrent vlv_index   Entry created
      west.example.com.gethostent vlv_index   Entry created
      west.example.com.getnetent vlv_index   Entry created
      west.example.com.getpwent vlv_index   Entry created
      west.example.com.getrpcent vlv_index   Entry created
      west.example.com.getspent vlv_index   Entry created
      west.example.com.getauhoent vlv_index   Entry created
      west.example.com.getsoluent vlv_index   Entry created
      west.example.com.getauduent vlv_index   Entry created
      west.example.com.getauthent vlv_index   Entry created
      west.example.com.getexecent vlv_index   Entry created
```

例 11–1 Example, Inc. ネットワークでの idsconfig の実行 *(続き)*

```
      west.example.com.getprofent vlv_index   Entry created
      west.example.com.getmailent vlv_index   Entry created
      west.example.com.getbootent vlv_index   Entry created
      west.example.com.getethent vlv_index   Entry created
      west.example.com.getngrpent vlv_index   Entry created
      west.example.com.getipnent vlv_index   Entry created
      west.example.com.getmaskent vlv_index   Entry created
      west.example.com.getprent vlv_index   Entry created
      west.example.com.getip4ent vlv_index   Entry created
      west.example.com.getip6ent vlv_index   Entry created

idsconfig: Setup of iDS server myserver is complete.


Note: idsconfig has created entries for VLV indexes.

      For DS5.x, use the directoryserver(1m) script on myserver
      to stop the server.  Then, using directoryserver, follow the
      directoryserver examples below to create the actual VLV indexes.

      For DS6.x, use dsadm command delivered with DS6.x on myserver
      to stop the server.  Then, using dsadm, follow the
      dsadm examples below to create the actual VLV indexes.

  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getgrent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.gethostent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getnetent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getpwent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getrpcent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getspent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getauhoent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getsoluent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getauduent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getauthent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getexecent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getprofent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getmailent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getbootent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getethent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getngrpent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getipnent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getmaskent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getprent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getip4ent
  directoryserver -s <server-instance> vlvindex -n west -T west.example.com.getip6ent

<install-path>/bin/dsadm reindex -l -t west.example.com.getgrent <directory-instance-path>
 dc=west,dc=example,dc=com
<install-path>/bin/dsadm reindex -l -t west.example.com.gethostent <directory-instance-path>
 dc=west,dc=example,dc=com
.
.
.
<install-path>/bin/dsadm reindex -l -t west.example.com.getip6ent <directory-instance-path>
 dc=west,dc=example,dc=com
```

ldapaddent を使用したディレクトリサーバーの生成

注 – pam_unix を使用している場合、データを使用してディレクトリサーバーを生成する前に、パスワードを UNIX Crypt 形式で格納するようにサーバーを構成してください。pam_ldap を使用している場合、任意の形式でパスワードを格納できます。UNIX crypt 形式でパスワードを設定する方法については、Sun Java System Directory Server のマニュアルを参照してください。

---

ldapaddent は、標準入力から (/etc/filename passwd など) データを読み取り、このデータをサービスに関連付けられたコンテナに配置します。クライアント構成により、デフォルトのデータ書き込み方法が決定されます。

注 – ldapaddent(1M) は LDAP クライアント上でのみ実行できます。LDAP ネームサービス用のクライアントの構成方法については、第 12 章「LDAP クライアントの設定 (手順)」を参照してください。

▼ ldapaddent を使ったユーザーパスワードデータによる Sun Java System Directory Server の生成方法

ldapaddent(1M) のマニュアルページを参照してください。LDAP セキュリティーおよび Directory Server への書き込みアクセスの詳細については、第 9 章「LDAP 基本コンポーネントおよび概念 (概要)」を参照してください。

- ldapaddent コマンドを使用して、/etc/passwd エントリをサーバーに追加します。

```
# ldapaddent -D "cn=directory manager" -f /etc/passwd passwd
```

プリンタエントリの管理

プリンタの追加

プリンタエントリを LDAP ディレクトリに追加する場合、printmgr 構成ツールまたは lpset -n ldap コマンド行ユーティリティーを使用します。lpset(1M) のマニュアルページを参照してください。ディレクトリに追加されるプリンタオブジェクトは、プリンタの印刷システムクライアントが必要とする接続パラメータのみを定義することに留意してください。ローカルのプリントサーバー構成データはファイル内に保持されます。典型的なプリンタエントリは、次のようになります。

```
printer-uri=myprinter,ou=printers,dc=mkg,dc=example,dc=com
objectclass=top
objectclass=printerService
objectclass=printerAbstract
objectclass=sunPrinter
printer-name=myprinter
sun-printer-bsdaddr=printsvr.example.com,myprinter,Solaris
sun-printer-kvp=description=HP LaserJet (PS)
printer-uri=myprinter
```

lpget の使用

lpget(1M) を使用して、LDAP クライアントの LDAP ディレクトリが認識するプリンタエントリすべてをリスト表示できます。LDAP クライアントの LDAP サーバーが複製サーバーの場合、更新複製規約 (update replication agreement) によって、リスト表示されたプリンタはマスター LDAP サーバーのプリンタと同じでない場合があります。詳細については、lpget(1M) のマニュアルページを参照してください。

たとえば、指定されたベース DN のプリンタすべてを一覧表示するには、次のように入力します。

```
# lpget -n ldap list
myprinter:
    dn=myprinter,ou=printers,dc=mkt,dc=example,dc=com
    bsdaddr=printsvr.example.com,myprinter,Solaris
    description=HP LaserJet (PS)
```

追加プロファイルを使用してディレクトリサーバーを生成する

ldapclient を genprofile オプションとともに使用すると、指定された属性に基づいて、構成プロファイルの LDIF 表現を作成できます。作成したプロファイルは、次に LDAP サーバーに読み込まれ、クライアントプロファイルとして使用されます。クライアントプロファイルは、 ldapclient init を使うことによりクライアントからダウンロードできます。

ldapclient genprofile の使用方法の詳細については、ldapclient(1M) のマニュアルページを参照してください。

▼ ldapclient を使った追加プロファイルによるディレクトリサーバーの生成方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2. ldapclient コマンドを genprofile オプションとともに使用します。

```
# ldapclient genprofile \
-a profileName=myprofile \
-a defaultSearchBase=dc=west,dc=example,dc=com \
-a "defaultServerList=192.168.0.1 192.168.0.2:386" \

> myprofile.ldif
```

3. 新規プロファイルをサーバーにアップロードします。

```
# ldapadd -h 192.168.0.1 -D “cn=directory manager” -f myprofile.ldif
```

ディレクトリサーバーを構成してアカウント管理を有効にする

Solaris 10 10/09 リリース以降では、pam_ldap を使用するクライアントと pam_unix を使用するクライアントに対してアカウント管理を実装できます。

注意 – 同じ LDAP ネームドメインで pam_ldap と pam_unix の両方を使用しないでください。すべてのクライアントが pam_ldap を使用するか、またはすべてのクライアントが pam_unix を使用します。この制限により、専用の LDAP サーバーが必要になる場合があります。

pam_ldap を使用するクライアントの場合

pam_ldap が正しく動作するには、パスワードとアカウントのロックアウトポリシーがサーバー上で正しく構成されている必要があります。ディレクトリサーバーコンソール、または ldapmodify を使用して、LDAP ディレクトリのアカウント管理ポリシーを構成できます。手順と詳細については、ご使用のバージョンの Sun Java System Directory Server の『管理者ガイド』の「ユーザーアカウントの管理」の章を参照してください。

注 – 以前は、pam_ldap アカウント管理を有効にすると、システムにログインする際には、常にすべてのユーザーが認証用にログインパスワードを入力する必要がありました。そのため、rsh、rlogin、ssh などのツールによるパスワードを使用しないログインは失敗します。

一方、pam_ldap(5) を Sun Java System Directory Server DS5.2p4 以降のリリースで使用することで、ユーザーはパスワードを入力せずに、rsh、rlogin、rcp、および ssh を使ってログインできるようになりました。

pam_ldap(5) は変更され、ユーザーのログイン時に Directory Server への認証を実行せずに、アカウントの管理およびユーザーのアカウント状態の取得を実行できるようになりました。Directory Server 上でこの機能を制御するのは、1.3.6.1.4.1.42.2.27.9.5.8 です。これは、デフォルトで有効になっています。

この制御をデフォルト以外に変更する場合は、Directory Server 上でアクセス制御情報 (ACI) を追加します。

```
dn: oid=1.3.6.1.4.1.42.2.27.9.5.8,cn=features,cn=config
objectClass: top
objectClass: directoryServerFeature
oid:1.3.6.1.4.1.42.2.27.9.5.8
cn:Password Policy Account Usable Request Control
aci: (targetattr != "aci")(version 3.0; acl "Account Usable";
     allow (read, search, compare, proxy)
     (groupdn = "ldap:///cn=Administrators,cn=config");)
creatorsName: cn=server,cn=plugins,cn=config
modifiersName: cn=server,cn=plugins,cn=config
```

proxy ユーザー用のパスワードは、期限が切れてはいけません。proxy パスワードが期限切れになった場合、 proxy 資格レベルを使用するクライアントはサーバーからネームサービス情報を取り出すことができません。proxy ユーザーのパスワードの期限が切れないことを保証するために、次のスクリプトを記述して proxy アカウントを変更します。

```
# ldapmodify -h ldapserver -D administrator DN \
-w administrator password <<EOF
dn: proxy user DN
DNchangetype: modify
replace: passwordexpirationtime
passwordexpirationtime: 20380119031407Z
EOF
```

注 – pam_ldap のアカウント管理は、Sun Java System Directory Server をもとにユーザーのパスワードやアカウントの有効期限情報を維持し、ユーザーに知らせます。ディレクトリサーバーは、ユーザーアカウントを検査する際、対応するシャドウエントリを解釈しません。しかし、pam_unix がシャドウデータを調査して、アカウントがロックされているか、パスワードが古くなっているかを判断します。LDAP ネームサービスやディレクトリサーバーはシャドウデータを最新の状態に維持しているわけではないので、pam_unix はシャドウデータに基づいたアクセスを許可するべきではありません。シャドウデータは proxy 識別情報を使って検出します。そのため、proxy ユーザーに userPassword 属性への読み取りアクセスを許可しないでください。proxy ユーザーを userPassword へ読み取りアクセスさせないことにより、pam_unix が無効なアカウントの検証を行わないようになります。

pam_unix を使用するクライアントの場合

Solaris LDAP クライアントがアカウント管理に対して pam_unix を使用できるようにするには、シャドウデータの更新を有効にするようにサーバーを設定する必要があります。この機能は、Solaris 10 10/09 リリース以降で使用できます。pam_ldap アカウント管理とは異なり、pam_unix は追加の構成手順が必要がありません。すべての構成は、idsconfig ユーティリティーを実行して行うことができます。基本的な idsconfig の実行については、例 11–1を参照してください。

次に 2 つの idsconfig 実行の出力を示します。

最初の idsconfig 実行では、既存のクライアントプロファイルを使用します。

```
# /usr/lib/ldap/idsconfig

It is strongly recommended that you BACKUP the directory server
before running idsconfig.

Hit Ctrl-C at any time before the final confirmation to exit.

Do you wish to continue with server setup (y/n/h)? [n] y
Enter the JES Directory Server's  hostname to setup: myserver
Enter the port number for iDS (h=help): [389]
Enter the directory manager DN: [cn=Directory Manager]
Enter passwd for cn=Directory Manager :
Enter the domainname to be served (h=help): [west.example.com]
Enter LDAP Base DN (h=help): [dc=west,dc=example,dc=com]
  Checking LDAP Base DN ...
  Validating LDAP Base DN and Suffix ...
  sasl/GSSAPI is not supported by this LDAP server

Enter the profile name (h=help): [default] WestUserProfile

Profile 'WestUserProfile' already exists, it is possible to enable
shadow update now. idsconfig will exit after shadow update
```

```
is enabled. You can also continue to overwrite the profile
or create a new one and be given the chance to enable
shadow update later.

Just enable shadow update (y/n/h)? [n] y
Add the administrator identity (y/n/h)? [y]
Enter DN for the administrator: [cn=admin,ou=profile,dc=west,dc=example,dc=com]
Enter passwd for the administrator:
Re-enter passwd:
  ADDED: Administrator identity cn=admin,ou=profile,dc=west,dc=example,dc=com.
         Proxy ACI LDAP_Naming_Services_proxy_password_read does not
         exist for dc=west,dc=example,dc=com.
  ACI SET: Give cn=admin,ou=profile,dc=west,dc=example,dc=com read/write access
           to shadow data.
  ACI SET: Non-Admin access to shadow data denied.

  Shadow update has been enabled.
```

2つ目の idsconfig 実行では、後で使用するための新しいプロファイルを作成します。出力の一部のみが表示されています。

```
# /usr/lib/ldap/idsconfig

It is strongly recommended that you BACKUP the directory server
before running idsconfig.

Hit Ctrl-C at any time before the final confirmation to exit.

Do you wish to continue with server setup (y/n/h)? [n] y
Enter the JES Directory Server's  hostname to setup: myserver
Enter the port number for iDS (h=help): [389]
Enter the directory manager DN: [cn=Directory Manager]
Enter passwd for cn=Directory Manager :
Enter the domainname to be served (h=help): [west.example.com]
Enter LDAP Base DN (h=help): [dc=west,dc=example,dc=com]
  Checking LDAP Base DN ...
  Validating LDAP Base DN and Suffix ...
  sasl/GSSAPI is not supported by this LDAP server

Enter the profile name (h=help): [default] WestUserProfile-new
Default server list (h=help): [192.168.0.1]
.
.
.
Do you want to enable shadow update (y/n/h)? [n] y

              Summary of Configuration

  1  Domain to serve              : west.example.com
  2  Base DN to setup             : dc=west,dc=example,dc=com
       Suffix to create           : dc=west,dc=example,dc=com
  3  Profile name to create       : WestUserProfile-new
.
.
.
 19  Enable shadow update         : TRUE
```

```
.
.
.
Enter DN for the administrator: [cn=admin,ou=profile,dc=west,dc=example,dc=com]
Enter passwd for the administrator:
Re-enter passwd:


WARNING: About to start committing changes. (y=continue, n=EXIT) y

  1. Changed timelimit to -1 in cn=config.
  2. Changed sizelimit to -1 in cn=config.
.
.
.
  11. ACI for dc=test1,dc=mpklab,dc=sfbay,dc=sun,dc=com modified to
      disable self modify.
.
.
.
  15. Give cn=admin,ou=profile,dc=west,dc=example,dc=com write permission for shadow.
...
```

Sun Java System Directory Server の移行

Sun Java System Directory Server 5.1 (以前の名称は Sun ONE Directory Server) リリースと Sun Java System Directory Server 5.2 リリース間のスキーマ変更が実装されました。`ldapaddent` コマンドは、`ethers/bootparams` のエントリに `objectclass: device` を追加します。したがって、LDAP コマンドを使用してディレクトリデータを Sun Java System Directory Server 5.1 から 5.2 に移行する場合、`ldapaddent -d` を使用してデータをエクスポートし、`ldapaddent` を使用してデータをインポートする必要があります。あるいは、Sun Java System Directory Server ツール `db2ldif` および `ldif2db` を使用してデータを移行する場合、データの移行の前にすべてのパッチを Sun Java System Directory Server 5.2 に適用する必要があります。パッチを適用しないと、データのインポートに失敗する可能性があります。

Sun Java System Directory Server 5.2 の構成については、Sun Java Enterprise System に収められている Sun Java System Directory Server のマニュアルを参照してください。

◆ ◆ ◆ 第 12 章

12 LDAP クライアントの設定 (手順)

この章では、Solaris LDAP ネームサービスクライアントの設定方法について説明します。この章で扱う内容は、次のとおりです。

LDAP クライアント設定の前提条件

Solaris クライアントで LDAP をネームサービスとして使用するためには、次の要件が満たされている必要があります。

- クライアントのドメイン名が LDAP サーバーによって処理される
- `nsswitch.conf` ファイルが、必要なサービスの LDAP を指している
- クライアントが、その動作を定義するための特定のパラメータをすべて使って構成されている
- `ldap_cachemgr` がクライアントで実行されている
- クライアントが構成されているサーバーが 1 つ以上起動され、実行されている

`ldapclient` ユーティリティーは、サーバーの起動を除き、上記の手順をすべて実行するので、LDAP クライアントを設定するための鍵となります。この章の後半では、`ldapclient` ユーティリティーを使用して LDAP クライアントを設定する方法や、それ以外の各種 LDAP ユーティリティーを使用して LDAP クライアントに関する情報を取得したり LDAP クライアントの状態をチェックしたりする方法について、例を挙げて説明します。

LDAP とサービス管理機能

LDAP クライアントサービスは、サービス管理機能を使用して管理されます。SMF の概要については、『Solaris のシステム管理 (基本編)』の第 18 章「サービスの管理 (概要)」を参照してください。また、詳細については、svcadm(1M) および svcs(1) のマニュアルページを参照してください。

- このサービスに関する有効化、無効化、再起動などの管理アクションは svcadm コマンドを使用して実行できます。

  ヒント – -t オプションを使用してサービスを一時的に無効化すると、そのサービス構成に対していくらかの保護を提供できます。-t オプションを指定してサービスを無効にした場合、リブート後に元の設定が復元されます。-t オプションを指定しないでサービスを無効にした場合、リブート後もそのサービスは無効のままです。

- LDAP クライアントサービスに対する障害管理リソース識別子 (FMRI) は、svc:/network/ldap/client:<*instance*> です。
- svcs コマンドを使用することによって、LDAP クライアントおよび ldap_cachemgr の状態を照会できます。
  - svcs コマンドと出力の例を、次に示します。

    ```
    # svcs \*ldap\*
    STATE          STIME    FMRI
    online         15:43:46  svc:/network/ldap/client:default
    ```

  - svcs -l コマンドと出力の例を、次に示します。次に示す出力を得るには、FMRI でインスタンス名を使用する必要があります。

    ```
    # svcs -l network/ldap/client:default
    fmri         svc:/network/ldap/client:default
    enabled      true
    state        online
    next_state   none
    restarter    svc:/system/svc/restarter:default
    contract_id  1598
    dependency   require_all/none file://localhost/var/ldap/ldap_client_file (-)
    dependency   require_all/none svc:/network/initial (online)
    dependency   require_all/none svc:/system/filesystem/minimal (online)
    ```

- デーモンの存在は ps コマンドを使用して確認できます。

  ```
  # ps -e | grep slapd
    root 23320     1   0   Aug 27 ?       16:30 ./ns-slapd -D \
    /usr/iplanet/ds5/slapd-lastrev -i /usr/iplanet/ds5/slapd-lastrev/
    root 25367 25353   0 15:35:19 pts/1     0:00 grep slapd
  ```

注 – -f オプションを ps で使用しないでください。このオプションはユーザー ID を名前に変換しようとするため、より多くのネームサービス検索が失敗する可能性があります。

LDAP クライアントの初期化

ldapclient(1M) は、Solaris システムで LDAP クライアントを設定するためのユーティリティーです。ldapclient ユーティリティーでは、サーバーがすでに適切なクライアントプロファイルで構成されていることを前提としています。サーバーをインストールして、適切なプロファイルで構成してからクライアントを設定する必要があります。

注 – Solaris オペレーティングシステムは、NIS クライアントとネイティブな LDAP クライアントが同一のクライアントマシン上に共存する構成をサポートしません。

ldapclient を使用してクライアントを設定するには、主に次の 2 つの方法があります。

- 「プロファイル」

  少なくとも、使用するプロファイルとドメインを含むサーバーアドレスを指定する必要があります。プロファイルが指定されていない場合は、デフォルトのプロファイルが使用されます。プロキシと認証データベースの情報を除いて、必要な情報はサーバーから入手できます。クライアントの資格レベルがプロキシまたは匿名プロキシである場合は、プロキシのバインド DN とパスワードを入力してください。詳細については、155 ページの「クライアント資格レベルの割り当て」を参照してください。

  Solaris 10 10/09 リリース以降では、enableShadowUpdate スイッチが使用できます。シャドウデータの更新を有効にするには、管理者資格 (adminDN と adminPassword) を入力する必要があります。

- 「手動」

  クライアント自体でプロファイルを設定します。つまり、コマンド行からすべてのパラメータを定義します。このため、プロファイル情報はキャッシュファイルに格納されサーバーによってリフレッシュされることはありません。

注 – クライアントを手動で構成することも可能ですが、お勧めしません。構成用のプロファイルを使用すると、クライアントの管理が容易になりコストも削減できます。

プロファイルを使用してクライアントを初期化する

▼ プロファイルを使用してクライアントを初期化する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2 `init` を指定して `ldapclient` を実行します。

```
# ldapclient init \
-a profileName=new \
-a domainName=west.example.com 192.168.0.1
System successfully configured
```

ユーザー別の資格を使用する

注 – どちらのクライアント構成ファイルも直接編集しないでください。これらのファイルの内容を作成または変更する場合は、`ldapclient` コマンドを使用してください。

▼ ユーザー別の資格を使用してクライアントを初期化する方法

始める前に ユーザー別の資格を使用してクライアントを設定する前に、次の項目が構成済みである必要があります。

- 1 つ以上の Kerberos KDC サーバーが構成され、稼働している
- DNS、DNS サーバーへのクライアントアクセス、および 1 つ以上の DNS サーバーが構成され、稼働している
- クライアントマシン上の Kerberos が構成され、有効にされている
- Kerberos クライアントのインストールプロファイルが存在している。次にプロファイルの例を示す

```
# cat /usr/tmp/krb5.profile
REALM SPARKS.COM
KDC kdc.example.com
ADMIN super/admin
FILEPATH /usr/tmp/krb5.conf
NFS 1
DNSLOOKUP none
```

- LDAP サーバーがインストールおよび構成され、`sasl/GSSAPI` がサポートされている

- 適切な識別情報マッピング構成が存在する
- ディレクトリサーバーおよび KDC 用の Kerberos ホスト主体が KDC 内で設定されている
- 使用するディレクトリサーバー DIT 上で `idsconfig` が稼働している
- ユーザー別の適切な `gssapi` プロファイル (`gssapi_EXAMPLE.COM` など) が作成済みである

  次に、`idsconfig` で表示されるユーザー別プロファイルの例の一部を示します。

```
# /usr/lib/ldap/idsconfig
Do you wish to continue with server setup (y/n/h)? [n] y
Enter the iPlanet Directory Server's (iDS) hostname to setup: kdc.example.com
Enter the port number for iDS (h=help): [389] <Enter your port>
Enter the directory manager DN: [cn=Directory Manager] <Enter your DN>
Enter passwd for cn=Directory Manager : <Enter your password>
Enter the domainname to be served (h=help): [example.com] <Enter your domain>
Enter LDAP Base DN (h=help): [dc=example,dc=com] <Enter your DN>
GSSAPI is supported. Do you want to set up gssapi:(y/n) [n] y
Enter Kerberos Realm: [EXAMPLE.COM] EXAMPLE.COMYou can create a sasl/GSSAPI enabled profile with default values n
want to create a sasl/GSSAPI default profile ? [n] y Enter. the profile name
(h=help): [gssapi_EXAMPLE.COM] <Enter>
GSSAPI setup is done. ...
```

- 必須のユーザー主体が鍵配布センター (KDC) 内に存在する
- クライアントマシン上で、次に示すようなコマンドにより、クライアントプロファイルを使って Kerberos が初期化される

  ```
  # /usr/sbin/kclient -p /usr/tmp/krb5.profile
  ```

- `/etc/nsswitch.ldap` が、`hosts` および `ipnodes` 用の `dns` を使用するように構成される。必要に応じ、エディタを使用してこのファイルを次のように修正します。

  ```
  host: files dns
  ipnodes: files dns
  ```

- `/etc/resolv.conf` が構成され、dns サービスが稼働している。詳細は、このマニュアルの DNS に関する章を参照してください。
- ディレクトリサーバー DIT が、(少なくとも) このクライアントマシンのユーザー、クライアントホスト、および必須の auto_home LDAP エントリで読み込み済みである。`ldapaddent` を使用してエントリを追加する方法については、このマニュアルのほかの節を参照してください。

1 **`ldapclient init`** を実行し、**`gssapi`** プロファイルを使用してクライアントを初期化します。

  ```
  # /usr/sbin/ldapclient init -a profilename=gssapi_SPARKS.COM -a \
  domainname=example.com 9.9.9.50
  ```

2 ユーザーとしてログインを試みます。

  `kinit -p user` を実行します。

  ユーザーのログインセッションで **`ldaplist -l passwd`** user を実行します。「userpassword」が表示されるはずです。

一方、**ldaplist -l passwd bar** を実行すると、**userpassword** なしでエントリを取得できます。デフォルトでは、root はすべてのユーザーの userpassword を引き続き表示できます。

参考 ユーザー別の資格の使用について

- syslog でメッセージ libsldap: Status: 7 Mesg: openConnection: GSSAPI bind failed - 82 Local error が表示される場合、Kerberos が初期化されていないか、チケットの有効期限が切れていることが考えられます。**klist** を実行して、表示される内容を確認します。**kinit -p foo** または **kinit -R -p foo** を実行して、再度試みてください。
- 必要に応じて、**pam_krb5.so.1** を /etc/pam.conf に追加することで、ログイン時に kinit を自動的に実行できます。
- 次に例を示します。

```
login   auth optional pam_krb5.so.1
rlogin  auth optional pam_krb5.so.1
other   auth optional pam_krb5.so.1
```

- ユーザーが kinit され、syslog メッセージに Invalid credential が表示される場合は、原因として、ホストのエントリ (root) かユーザーエントリが LDAP ディレクトリ内に存在しない、またはマッピング規則が正しくないことが考えられます。
- ldapclient init の実行時に、LDAP プロファイルに self/ sasl/GSSAPI 構成が含まれるかどうかがチェックされます。/etc/nsswitch.ldap のチェックに失敗する場合の一般的な原因は、dns が host: および ipnodes: に追加されていないことです。
- DNS クライアントが有効でないために失敗する場合は、**svcs -l dns/client** を実行して、/etc/resolv.conf が存在しないのか、単に無効になっているのかを確認します。これを有効にするには、**svcadm enable dns/client** を実行します。
- sasl/GSSAPI バインドが原因でチェックが失敗する場合は、syslog で問題の箇所を確認します。

詳細については、このマニュアルのほかの箇所および『Solaris のシステム管理 (セキュリティサービス)』を参照してください。

プロキシの資格を使用する

▼ プロキシの資格を使用してクライアントを初期化する方法

注 – どちらのクライアント構成ファイルも直接編集しないでください。これらのファイルの内容を作成または変更する場合は、ldapclient を使用してください。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2 ldapclient を実行します (プロキシ値を定義します)。

```
# ldapclient init \
-a proxyDN=cn=proxyagent,ou=profile,dc=west,dc=example,dc=com \
-a domainName=west.example.com \
-a profileName=pit1 \
-a proxyPassword=test1234 192.168.0.1
System successfully configured
```

使用するプロファイルが proxy 用に設定されている場合は、-a proxyDN と -a proxyPassword が必須です。サーバーに保存されているプロファイルにはこの資格情報が含まれていないため、クライアントを初期設定するときは資格情報を入力する必要があります。この方法は、プロキシの資格情報をサーバーに保存していた従来の方法に比べて安全性が高くなります。

プロキシ情報は、/var/ldap/ldap_client_cred の作成に使用されます。それ以外の情報は、/var/ldap/ldap_client_file に格納されます。

LDAP でのシャドウ更新を有効にする

Solaris 10 10/09 リリース以降では、enableShadowUpdate スイッチが使用できます。

▼ クライアントを初期化してシャドウデータの更新を有効にする方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2 enableShadowUpdate スイッチを設定し、管理者資格を定義するには、ldapclient コマンドを実行します。

- 既に実行中のクライアントを更新するには、次のコマンドを使用します。

```
# ldapclient mod -a enableShadowUpdate=TRUE \
-a adminDN=cn=admin,ou=profile,dc=west,dc=example,dc=com \
-a adminPassword=admin-password
System successfully configured
```

- クライアントを初期化するには、次のコマンドを使用します。

```
# ldapclient init \
-a adminDN=cn=admin,ou=profile,dc=west,dc=example,dc=com \
-a adminPassword=admin-password
-a domainName=west.example.com \
-a profileName=WestUserProfile \
-a proxyDN=cn=proxyagent,ou=profile,dc=west,dc=example,dc=com \
-a proxyPassword=i<proxy_password> \
192.168.0.1
System successfully configured
```

3 構成を検証するには、/var/ldap/ldap_client_cred ファイルの内容を表示します。
出力には、次のような行を含むべきです。

```
# cat /var/ldap/ldap_client_cred

   NS_LDAP_BINDDN= cn=proxyagent,ou=profile,dc=west,dc=example,dc=com
   NS_LDAP_BINDPASSWD= {NS1}4a3788f8eb85de11
   NS_LDAP_ENABLE_SHADOW_UPDATE= TRUE
   NS_LDAP_ADMIN_BINDDN= cn=admin,ou=profile,dc=west,dc=example,dc=com
   NS_LDAP_ADMIN_BINDPASSWD= {NS1}4a3788f8c053434f
```

クライアントを手動で初期設定する

スーパーユーザー、または同等の役割の管理者は、クライアントを手動で構成できます。ただし、この処理では多数のチェックが省略されるため、システムを正しく構成できないことがよくあります。また、プロファイルを使用するときのように一括に設定するのではなく、「マシンごとに」設定を変更する必要があります。

▼ クライアントを手動で初期設定する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2 ldapclient manual を実行してクライアントを初期化します。

```
# ldapclient manual \
-a domainName=dc=west.example.com \
-a credentialLevel=proxy \
-a defaultSearchBase=dc=west,dc=example,dc=com \
-a proxyDN=cn=proxyagent,ou=profile,dc=west,dc=example,dc=com \
-a proxyPassword=testtest 192.168.0.1
```

3 ldapclient list を使用して確認します。

```
NS_LDAP_FILE_VERSION= 2.0
NS_LDAP_BINDDN= cn=proxyagent,ou=profile,dc=west,dc=example,dc=com
NS_LDAP_BINDPASSWD= {NS1}4a3788e8c053424f
```

```
NS_LDAP_SERVERS= 192.168.0.1
NS_LDAP_SEARCH_BASEDN= dc=west,dc=example,dc=com
NS_LDAP_CREDENTIAL_LEVEL= proxy
```

手動によるクライアント構成を変更する

▼ 手動による構成を変更する方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2. ldapclient mod コマンドを使用して、認証方法を simple に変更します。

   ```
   # ldapclient mod -a authenticationMethod=simple
   ```

3. ldapclient list を実行して、更新が行われたことを確認します。

   ```
   # ldapclient list
   NS_LDAP_FILE_VERSION= 2.0
   NS_LDAP_BINDDN= cn=proxyagent,ou=profile,dc=west,dc=example,dc=com
   NS_LDAP_BINDPASSWD= {NS1}4a3788e8c053424f
   NS_LDAP_SERVERS= 192.168.0.1
   NS_LDAP_SEARCH_BASEDN= dc=west,dc=example,dc=com
   NS_LDAP_AUTH= simple
   NS_LDAP_CREDENTIAL_LEVEL= proxy
   ```

注意事項　LDAP クライアント設定には、mod サブコマンドでは変更できない属性があります。たとえば、profileName 属性や profileTTL 属性は変更できません。これらの属性を変更するには、202 ページの「プロファイルを使用してクライアントを初期化する」で説明されているように、ldapclient init コマンドを使用して新しいプロファイルを作成します。206 ページの「クライアントを手動で初期設定する」で説明されているように、ldapclient manual コマンドを実行することもできます。

クライアントの初期設定を解除する

ldapclient uninit コマンドは、クライアントのネームサービスを元の状態 (init、modify、または manual の最後の操作が行われる前の状態) に復元します。言い換えれば、最後に行われた手順を「元に戻します」。たとえば、profile1 を使用するようにクライアントを設定したあとで profile2 を使用するように変更した場合、ldapclient uninit を実行すると、クライアントで profile1 を使用するように設定が元に戻ります。

▼ クライアントの初期設定を解除する方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2 `ldapclient uninit` を実行します。

```
# ldapclient uninit
System successfully recovered
```

TLS のセキュリティーの設定

注 – セキュリティーデータベースファイルは、すべてのユーザーから読み取れるようにする必要があります。`key3.db` に非公開鍵を含めないようにしてください。

TLS を使用する場合は、必要なセキュリティーデータベースをインストールしなければなりません。具体的には証明書ファイルと鍵データベースファイルが必要です。たとえば、Netscape Communicator の古いデータベースフォーマットを使用する場合、`cert7.db` と `key3.db` の 2 つのファイルが必要です。あるいは、Mozilla の新しいデータベースフォーマットを使用する場合、`cert8.db`、`key3.db`、および `secmod.db` の 3 つのファイルが必要です。`cert7.db` ファイルまたは `cert8.db` ファイルには、信頼された証明書が入ります。`key3.db` ファイルには、クライアントの鍵が入ります。LDAP ネームサービスクライアントがクライアントの鍵を使用しない場合でも、このファイルは必要です。`secmod.db` ファイルには、`PKCS#11` などのセキュリティーモジュールが入ります。このファイルは、古いフォーマットを使用する場合には必要ありません。

注 – `ldapclient` を実行する前に、この節に記述されている必要なセキュリティーデータベースファイルを設定およびインストールしておく必要があります。

これらのファイルを作成および管理する方法については、ご使用のバージョンの Sun Java System Directory Server の『管理者ガイド』の「SSL 管理」の章の中の LDAP クライアントで SSL を利用するための構成に関する節を参照してください。これらのファイルを構成したら、LDAP ネームサービスクライアントで使用できるように所定の場所にそれらを格納する必要があります。この場所を判断するために、属性 `certificatePath` が使用されます。この属性はデフォルトで `/var/ldap` です。

たとえば、Netscape Communicator を使用して必要な `cert7.db` ファイルと `key3.db` ファイルを設定したあとで、これらのファイルをデフォルトの位置にコピーします。

```
# cp $HOME/.netscape/cert7.db /var/ldap
# cp $HOME/.netscape/key3.db /var/ldap
```

次に、すべてのユーザーに読み取り権を付与します。

```
# chmod 444 /var/ldap/cert7.db
# chmod 444 /var/ldap/key3.db
```

注 – Netscape は cert7.db ファイルと key3.db ファイルを $HOME/.netscape ディレクトリで管理し、Mozilla は cert8.db ファイル、key3.db ファイル、および secmod.db ファイルを $HOME/.mozilla のサブディレクトリで管理します。このため、それらのセキュリティーデータベースを LDAP ネームサービスクライアントで使用する場合は、そのコピーをローカルファイルシステム上に格納する必要があります。

PAM を構成する

pam_ldap は、LDAP の認証およびアカウント管理用 PAM モジュールオプションの 1 つです。pam_ldap で現在サポートされている機能の詳細については、pam_ldap(5) のマニュアルページと付録 A 「Solaris 10 ソフトウェアの DNS、NIS、および LDAP の更新」を参照してください。

ユーザー別モードと自己資格オプションの両方を選択した場合は、PAM Kerberos pam_krb5(5) pam モジュールも有効にする必要があります。詳細については、pam_krb5(5) のマニュアルページおよび『Solaris のシステム管理 (セキュリティサービス)』を参照してください。

UNIX policy を使用するための PAM の構成

UNIX policy を使用するように PAM を構成するには、225 ページの「pam_ldap に対応した pam.conf ファイルの例」に示す例に従ってください。pam_ldap.so.1 を含む行をクライアントの /etc/pam.conf ファイルに追加します。詳細については、pam.conf(4) のマニュアルページを参照してください。

LDAP server_policy を使用するための PAM の構成

LDAP server_policy を使用するように PAM を構成するには、227 ページの「アカウント管理のために pam_ldap を構成した pam.conf ファイル例」に示す例に従ってください。pam_ldap.so.1 を含む行をクライアントの /etc/pam.conf ファイルに追加します。さらに、サンプルの pam.conf ファイルの中でいずれかの PAM モジュールが binding フラグと server_policy オプションを定義している場合は、クライアントの /etc/pam.conf ファイルの対応するモジュールに、同じフラグとオプションを記述します。また、サービスモジュール pam_authtok_store.so.1 を含む行に、server_policy オプションを追加します。

注 – 以前は、pam_ldap アカウント管理を有効にすると、システムにログインする際には、常にすべてのユーザーが認証用にログインパスワードを入力する必要がありました。そのため、rsh、rlogin、ssh などのツールによるパスワードを使用しないログインは失敗します。

一方、pam_ldap(5) を Sun Java System Directory Server DS5.2p4 以降のリリースで使用することで、ユーザーはパスワードを入力せずに、rsh、rlogin、rcp、および ssh を使ってログインできるようになりました。

pam_ldap(5) は変更され、ユーザーのログイン時に Directory Server への認証を実行せずに、アカウントの管理およびユーザーのアカウント状態の取得を実行できるようになりました。Directory Server 上でこの機能を制御するのは、1.3.6.1.4.1.42.2.27.9.5.8 です。これは、デフォルトで有効になっています。

この制御をデフォルト以外に変更する場合は、Directory Server 上でアクセス制御情報 (ACI) を追加します。

```
dn: oid=1.3.6.1.4.1.42.2.27.9.5.8,cn=features,cn=config
objectClass: top
objectClass: directoryServerFeature
oid:1.3.6.1.4.1.42.2.27.9.5.8
cn:Password Policy Account Usable Request Control
aci: (targetattr != "aci")(version 3.0; acl "Account Usable";
     allow (read, search, compare, proxy)
     (groupdn = "ldap:///cn=Administrators,cn=config");)
creatorsName: cn=server,cn=plugins,cn=config
modifiersName: cn=server,cn=plugins,cn=config
```

- binding 管理フラグ

  binding 管理フラグを使うことにより、遠隔 (LDAP) パスワードよりローカルパスワードが優先されます。たとえば、ローカルファイルと LDAP 名前空間の両方にユーザーアカウントが見つかった場合、遠隔パスワードよりローカルアカウントのパスワードの方が優先されます。したがって、ローカルパスワードの期限が切れているときは、たとえ LDAP パスワードがまだ有効であっても認証に失敗します。

- server_policy オプション

  server_policy オプションによって、pam_unix_auth、 pam_unix_account、および pam_passwd_auth は LDAP 名前空間で検出されたユーザーを無視し、pam_ldap による認証やアカウント検証が可能になります。pam_authtok_store は、新しいパスワードを暗号化せずに LDAP サーバーに渡します。そのため、パスワードはサーバー上で構成されるパスワードの暗号化方式に基づいたディレクトリに保存されます。詳細については、pam.conf(4) および pam_ldap(5) のマニュアルページを参照してください。

LDAP ネームサービス情報の検出

ldaplist ユーティリティーを使用して、LDAP ネームサービスについての情報を取得できます。この LDAP ユーティリティーは、LDAP サーバーから取得したネームサービス情報を LDIF フォーマットで表示します。このユーティリティーは、トラブルシューティングに役立ちます。詳細については、ldaplist(1) のマニュアルページを参照してください。

すべての LDAP コンテナを表示する

ldaplist は、レコードを空行で区切って出力を表示します。この表示方法は、複数行にまたがる大きなレコードに有効です。

注 – ldaplist の出力は、クライアントの構成によって変わります。たとえば、ns_ldap_search の値が one ではなく sub である場合は、ldaplist によって現在の検索 baseDN の下にあるエントリがすべて表示されます。

次に ldaplist の出力例を示します。

```
# ldaplist
dn: ou=people,dc=west,dc=example,dc=com

dn: ou=group,dc=west,dc=example,dc=com

dn: ou=rpc,dc=west,dc=example,dc=com

dn: ou=protocols,dc=west,dc=example,dc=com

dn: ou=networks,dc=west,dc=example,dc=com

dn: ou=netgroup,dc=west,dc=example,dc=com

dn: ou=aliases,dc=west,dc=example,dc=com

dn: ou=hosts,dc=west,dc=example,dc=com

dn: ou=services,dc=west,dc=example,dc=com

dn: ou=ethers,dc=west,dc=example,dc=com

dn: ou=profile,dc=west,dc=example,dc=com

dn: automountmap=auto_home,dc=west,dc=example,dc=com

dn: automountmap=auto_direct,dc=west,dc=example,dc=com

dn: automountmap=auto_master,dc=west,dc=example,dc=com

dn: automountmap=auto_shared,dc=west,dc=example,dc=com
```

すべてのユーザーエントリ属性を表示する

ユーザーの passwd エントリなど特定の情報を表示する場合は、次のように getent を使用します。

```
# getent passwd user1
user1::30641:10:Joe Q. User:/home/user1:/bin/csh
```

すべての属性を表示する場合は、-l オプションを指定して ldaplist コマンドを実行します。

```
# ldaplist -l passwd user1dn: uid=user1,ou=People,dc=west,dc=example,dc=com
        uid: user1
        cn: user1
        uidNumber: 30641
        gidNumber: 10
        gecos: Joe Q. User
        homeDirectory: /home/user1
        loginShell: /bin/csh
        objectClass: top
        objectClass: shadowAccount
        objectClass: account
        objectClass: posixAccount
        shadowLastChange: 6445
```

LDAP クライアント環境のカスタマイズ

以降の節では、クライアント環境をカスタマイズする方法について説明します。

どのサービスも変更できますが注意が必要です。変更したサービスのデータがサーバー上に生成されない場合、カスタマイズは無効になります。また、ファイルがデフォルトで設定されない場合もあります。

LDAP 用の nsswitch.conf ファイルを変更する

/etc/nsswitch.conf ファイルを変更して、各サービスが情報を取得する場所をカスタマイズできます。デフォルトの設定は /etc/nsswitch.ldap に保存されており、クライアントの初期化時に ldapclient がこのファイルを使って /etc/nsswitch.conf ファイルを作成します。

LDAP で DNS を有効にする

/etc/resolv.conf ファイルを設定して DNS を使用可能にする場合は、次に示すように、DNS を hosts 行に追加します。

```
hosts:      ldap dns [NOTFOUND=return] files
```

推奨構成を次に示します。

```
hosts: files dns

ipnodes: files dns
```

ユーザー別の認証を使用する場合、sasl/GSSAPI および Kerberos 機構は dns ネームサービスが構成され、有効になっていることを前提に動作します。詳細については、この管理ガイドの DNS に関する章を参照してください。

◆ ◆ ◆ 第 13 章

13 LDAP のトラブルシューティング (参照情報)

この章では、LDAP の構成で発生する問題とその解決方法を示します。

注 – LDAP サービスはサービス管理機能によって管理されます。このサービスに関する有効化、無効化、再起動などの管理アクションは svcadm コマンドを使用して実行できます。LDAP でこの機能を使用する場合の詳細については、200 ページの「LDAP とサービス管理機能」を参照してください。この機能の概要については、『Solaris のシステム管理 (基本編)』の第 18 章「サービスの管理 (概要)」を参照してください。また、詳細については、svcadm(1M) および svcs(1) のマニュアルページを参照してください。

LDAP クライアントステータスの監視

以降の節では、LDAP クライアント環境の状態判定に使用するさまざまなコマンドを紹介します。使用可能なオプションの詳細については、マニュアルページも参照してください。

サービス管理機能の概要は、『Solaris のシステム管理 (基本編)』の第 18 章「サービスの管理 (概要)」を参照してください。また、詳細については、svcadm(1M) および svcs(1) のマニュアルページを参照してください。

ldap_cachemgr が実行中であることを確認する

ldap_cachemgr デーモンは、常に実行中で適切に機能している必要があります。このデーモンが機能していない場合、システムは動作しません。LDAP クライアントを起動すると、ldap_cachemgr デーモンが自動的に起動します。したがって、ldap_cachemgr が実行されていない場合は、LDAP クライアントが使用不能になります。LDAP クライアントがオンラインであるかどうかを確認する 2 つの方法を次に示します。

- svcs コマンドを使用します。

```
# svcs \*ldap\*
STATE          STIME    FMRI
disabled       Aug_24   svc:/network/ldap/client:default
```

または

```
# svcs -l network/ldap/client:default
fmri         svc:/network/ldap/client:default
enabled      true
state        online
next_state   none
restarter    svc:/system/svc/restarter:default
contract_id  1598
dependency   require_all/none file://localhost/var/ldap/ldap_client_file (-)
dependency   require_all/none svc:/network/initial (online)
dependency   require_all/none svc:/system/filesystem/minimal (online)
```

- -g オプションを ldap_cachemgr に渡します。

このオプションによって、問題の診断に役立つより広範な状態情報がダンプされます。

```
# /usr/lib/ldap/ldap_cachemgr -g
cachemgr configuration:
server debug level          0
server log file "/var/ldap/cachemgr.log"
number of calls to ldapcachemgr         19

cachemgr cache data statistics:
Configuration refresh information:
  Previous refresh time: 2001/11/16 18:33:28
  Next refresh time:     2001/11/16 18:43:28
Server information:
  Previous refresh time: 2001/11/16 18:33:28
  Next refresh time:     2001/11/16 18:36:08
  server: 192.168.0.0, status: UP
  server: 192.168.0.1, status: ERROR
    error message: Can't connect to the LDAP server
Cache data information:
  Maximum cache entries:          256
  Number of cache entries:          2
```

ldap_cachemgr デーモンの詳細については、ldap_cachemgr(1M) のマニュアルページを参照してください。

現在のプロファイル情報の確認

スーパーユーザーになるか、同等の役割になり、ldapclient を list オプションで実行します。

```
# ldapclient list
NS_LDAP_FILE_VERSION= 2.0
NS_LDAP_BINDDN= cn=proxyagent,ou=profile,dc=west,dc=example,dc=com
```

```
NS_LDAP_BINDPASSWD= {NS1}4a3788e8c053424f
NS_LDAP_SERVERS= 192.168.0.1, 192.168.0.10
NS_LDAP_SEARCH_BASEDN= dc=west,dc=example,dc=com
NS_LDAP_AUTH= simple
NS_LDAP_SEARCH_REF= TRUE
NS_LDAP_SEARCH_SCOPE= one
NS_LDAP_SEARCH_TIME= 30
NS_LDAP_SERVER_PREF= 192.168.0.1
NS_LDAP_PROFILE= pit1
NS_LDAP_CREDENTIAL_LEVEL= proxy
NS_LDAP_SERVICE_SEARCH_DESC= passwd:ou=people,?sub
NS_LDAP_SERVICE_SEARCH_DESC= group:ou=group,dc=west,dc=example,dc=com?one
NS_LDAP_BIND_TIME= 5
```

/var/ldap ファイルは現在 ASCII 形式ですが、バイナリに変更される可能性があるため、このファイルを連結すると問題が発生する可能性があります。この情報へのアクセスにサポートされている方式は、ldapclient list です。詳細については、ldapclient(1M) のマニュアルページを参照してください。

基本的なクライアント/サーバー間通信の検証

クライアントが LDAP サーバーに対して通信を行なっていることを確認する最善の方法は、ldaplist コマンドを使用することです。引数を付けずに ldaplist だけを指定して実行すると、サーバー上のすべてのコンテナがダンプされます。この方法はコンテナが存在している限り可能で、コンテナを生成する必要がありません。詳細については、ldaplist(1) のマニュアルページを参照してください。

最初の手順が成功したら、ldaplist passwd *username* または ldaplist hosts *hostname* を実行できますが、大量のデータが含まれている場合には、生成量の少ないサービスを選ぶか、head や more コマンドを使用してデータをパイプ処理することもできます。

クライアント以外のマシンからのサーバーデータの確認

前述のコマンドの大半は、LDAP クライアントを作成済みであることが前提です。クライアントを作成していない状態でサーバー上のデータをチェックする場合は、ldapsearch コマンドを使用します。次の例では、すべてのコンテナをリスト表示します。

```
# ldapsearch -h server1 -b "dc=west,dc=example,dc=com" -s one "objectclass=*"
```

Solaris 9 およびそれ以前のリリースでは、ldapsearch コマンドはデフォルトで、非標準のテキスト表現で出力を生成していました。それより後の Solaris リリースの ldapsearch のデフォルト出力は、RFC-2849 で定義されている業界標準の LDIF フォーマットです。ldapsearch のすべてのバージョンで、-L オプションを使用することによって LDIF フォーマットを出力できます。

LDAP の構成で発生する問題とその解決方法

以降の節では、LDAP の構成で発生する問題とそれらの解決方法について説明します。

未解決のホスト名

Solaris プラットフォームの LDAP クライアントバックエンドは、ホストの検索で、`gethostbyname()` や `getaddrinfo()` で返されるホスト名のような、完全指定ホスト名を返します。格納済みの名前が指定されている (1 つ以上のドットが含まれている) 場合、クライアントはその名前をそのまま返します。たとえば、格納されている名前が `hostB.eng` であれば、返される名前も `hostB.eng` です。

LDAP ディレクトリに格納された名前が指定されていない (ドットが含まれない) 場合、クライアントのバックエンドは、その名前にドメイン部分を追加します。たとえば、格納されている名前が `hostA` であれば、返される名前は `hostA.domainname` となります。

LDAP ドメイン内のシステムに遠隔アクセスできない

DNS ドメイン名が LDAP ドメイン名とは異なる場合、格納されたホスト名が完全指定でない限り LDAP ネームサービスをホスト名に対して使用することはできません。

ログインできない

LDAP クライアントは、ログイン時に PAMモジュールを使用してユーザーを認証します。UNIX 標準の PAM モジュールでは、パスワードをサーバーから読み込みクライアント側で検査します。この動作は、次のいずれかの理由で失敗する場合があります。

1. `/etc/nsswitch.conf` ファイル内の `passwd` サービスが `ldap` を使用しない
2. プロキシエージェントが、サーバーリスト上のユーザーの `userPassword` 属性を読み取ることができない。プロキシエージェントが比較のためにパスワードをクライアントに返すので、少なくともプロキシエージェントはパスワードを読めなければならない。`pam_ldap` に関しては、パスワードへの読み取りアクセスを必要としない
3. プロキシエージェントが適切なパスワードを保持していない
4. 該当するエントリに `shadowAccount` オブジェクトクラスが定義されていない

5. パスワードが定義されていない

   `ldapaddent` を使用する場合、`-p` オプションを使用してパスワードをユーザーエントリに確実に追加する必要があります。`ldapaddent` を `-p` オプションなしで実行した場合、`ldapaddent` を使用して `/etc/shadow` ファイルを追加しない限り、ユーザーのパスワードはディレクトリに格納されません。

6. LDAP サーバーに到達することができない

   サーバーの状態を確認します。

   ```
   # /usr/lib/ldap/ldap_cachemgr -g
   ```

7. `pam.conf` の構成が不正である
8. LDAP 名前空間でユーザーが定義されていない
9. `pam_unix` で `NS_LDAP_CREDENTIAL_LEVEL` が `anonymous` に設定されており、匿名ユーザーが `userPassword` を使用できない
10. パスワードが `crypt` 形式で格納されていない
11. アカウント管理をサポートするように `pam_ldap` が構成されている場合は、次のいずれかの原因でログインに失敗します。
    - ユーザーのパスワード期限が切れている
    - ログインを何回も行ったために、ユーザーアカウントがロックされる
    - 管理者がユーザーアカウントを非アクティブにした
    - `rsh`、`rlogin`、`ssh`、`sftp` などのパスワードを使用しないプログラムによってユーザーがログインしようとした
12. ユーザー別の認証および sasl/GSSAPI を使用している場合、一部の Kerberos コンポーネントまたは `pam_krb5` 構成が正しく設定されません。この問題を解決する方法については、『Solaris のシステム管理 (セキュリティサービス)』を参照してください。

検索が遅い

LDAP データベースは、検索パフォーマンス向上にインデックスを使用します。インデックスが正しく作成されていない場合、大幅にパフォーマンスが低下することがあります。このマニュアルには、インデックスを作成する必要のある共通の属性セットを記述しています。また、独自のインデックスを追加して、パフォーマンスの向上を図ることができます。

`ldapclient` がサーバーにバインドできない

`profileName` 属性を指定して `init` オプションを使用すると、`ldapclient` がクライアントの初期化に失敗することがあります。考えられる失敗の原因は次のとおりです。

1. コマンド行で不正なドメイン名が指定された
2. 指定されたクライアントドメインのエントリポイントを表す nisDomain 属性が DIT (ディレクトリ情報ツリー) 内に設定されていない
3. アクセス制御情報がサーバー上で適正に設定されていないため、LDAP データベース内の匿名検索が許可されない
4. ldapclient コマンドに渡されたサーバーアドレスが間違っている。ldapsearch を使用してサーバーのアドレスを確認する
5. ldapclient コマンドに渡されたプロファイル名が間違っている。ldapsearch を使用して、DIT 内のプロファイル名を確認する
6. クライアントのネットワークインタフェースに対して snoop を実行して外向きのトラフィックを検査して、どのサーバーにアクセスしているかを確認する

デバッグに ldap_cachemgr を使用する

ldap_cachemgr を -g オプションを付けて使用すると、現在のクライアント構成および統計を表示できるため、デバッグするのに便利です。たとえば、次のように指定します。

```
# ldap_cachemgr -g
```

この結果、すでに説明したように、すべての LDAP サーバーの状態を含む現在のクライアント構成および統計が標準出力に出力されます。このコマンドを実行するのに、スーパーユーザーになる必要はありません。

セットアップ中に ldapclient がハングアップする

ldapclient コマンドがハングアップした場合、Ctrl-C キーを押すと以前の環境を復元したあとで終了します。この状況が発生する場合、サーバーが動作していることをサーバー管理者に確認してください。

プロファイルまたはコマンド行に指定されたサーバーリスト属性で、サーバー情報が適正であることを確認してください。

◆ ◆ ◆ 第 14 章

LDAP の一般的なリファレンス

この章の内容は次のとおりです。

記入用のチェックリスト

表 14–1 サーバーで定義する変数

変数	_____ネットワークの定義
インストールしたディレクトリサーバーインスタンスのポート番号 (389)	
サーバー名	
複製サーバー(IP 番号: ポート番号)	
ディレクトリマネージャー [dn: cn=directory manager]	

表 14–1 サーバーで定義する変数 (続き)

変数	_____ネットワークの定義
サービスされるドメイン名	
クライアント要求の処理がタイムアウトするまでの時間 (秒)	
各検索要求で返されるエントリの最大数	

表 14–2 クライアントプロファイルで定義する変数

変数	_____ネットワークの定義
プロファイル名	
サーバーリスト (デフォルトはローカルサブネット)	
優先されるサーバーリスト (優先順に記載)	
検索範囲 (検索するディレクトリツリーレベルの数、「One」または「Sub」)	
サーバーへのアクセスに使用する資格。デフォルトは anonymous	
参照に従うかどうか。(メインサーバーが利用不可能な場合に使用される、別のサーバーへのポインタ)。デフォルトは no	
検索時にサーバーが情報を返すまでの待機制限時間 (秒、デフォルトは 30)	
サーバーとの通信時のバインド制限時間 (秒、デフォルトは 30)。デフォルトは秒	
認証方式。デフォルトは none	

LDAP のアップグレード情報

この節では、Solaris 8 リリースから Solaris 9 以降のリリースにアップグレードする際の考慮事項について説明します。

互換性

Solaris 9 以降の Solaris ソフトウェアリリースで構成されたクライアントは、バージョン 1 のプロファイルのみに対応する Solaris 8 クライアント用に設定されたディレクトリサーバーと完全な互換性があります。ただし、Solaris 9 以降のリ

リースで導入された新しい機能を利用し、新しいセキュリティーモデルを使用するには、バージョン 2 のプロファイルを使用する必要があります。

サーバーは、旧クライアントと新クライアントの混在環境に対応します。スキーママッピングが無効であり、バージョン 2 のプロファイルが serviceSearchDescriptors 属性の特殊フィルタを使用しないように構成されている限り、どちらのクライアントでも同じ結果を得ることができます。サーバーがデフォルトのスキーマを使用しない場合、Solaris 8 クライアントはデフォルト以外のスキーマを任意に対応づけできないため、旧クライアントはそのサーバーを使用できません。

ldap_cachemgr デーモンの実行

Solaris 9 リリース以降では、ldap_cachemgr デーモンを常に実行している必要があります。このデーモンは、クライアントが適正に動作するために「必須」です。サービス管理機能の svcadm コマンドを使用して LDAP クライアントを起動すると、ldap_cachemgr デーモンが自動的に起動します。詳細については、ldap_cachemgr(1M) のマニュアルページを参照してください。

新しい automount スキーマ

Solaris 9 リリース以降、Solaris ソフトウェアは、automount エントリ用の新しいスキーマをデフォルトで使用します。この新しいスキーマは、Solaris 8 クライアントが使用していた汎用の NIS マップスキーマに置き換わります。このため Solaris 9 以降のソフトウェアツールを使用してサーバーを設定した場合、Solaris 8 クライアントから automount エントリを表示できなくなります。サイトで Solaris 8 クライアントとそれ以降の Solaris ソフトウェアクライアントの両方に対応するサーバーを設定する場合、自動マウントエントリを追加する前に、プロファイルを作成してスキーマを以前のスキーマに対応づけてください。この操作により、ldapaddent(1M) が、以前のスキーマを使用してエントリを追加することが保証されます。ただし、Solaris 9 以降のソフトウェアに基づくすべてのクライアントで、automount 用スキーマに対応づけられたプロファイルを使用する必要があることに注意してください。

このマッピングを有効にするため、次のマッピング属性をプロファイルに追加する必要があります。

```
attributeMap:         automount:automountMapName=nisMapName
attributeMap:         automount:automountKey=cn
attributeMap:         automount:automountInformation=nisMapEntry
objectclassMap:       automount:automountMap=nisMap
objectclassMap:       automount:automount=nisObject
```

pam_ldap の変更点

Solaris 10 OS リリースでは、pam_ldap が次に示すように変更されました。詳細については、pam_ldap(5) のマニュアルページも参照してください。

- 以前サポートされていた use_first_pass および try_first_pass のオプションは、Solaris 10 ソフトウェアリリースでサポートされなくなりました。このオプションは不要のため pam.conf から削除しても問題はなく、そのままにしておいても構いません。将来のリリースで削除されることもあります。
- パスワードプロンプトに対応する必要があります。これは、認証およびパスワードモジュールのスタックで pam_ldap の前に pam_authtok_get をスタックし、passwd サービスの auth スタックに pam_passwd_auth を含めることによって行います。
- 以前サポートされていたパスワード更新機能は、以前使用が推奨されていた、pam_authtok_store と server_policy オプションにこのリリースで置き換わります。

このリリースにアップグレードしても、既存の pam.conf ファイルは自動的に更新されず、上記の変更は反映されません。既存の pam.conf ファイルに pam_ldap の構成が含まれる場合は、アップグレード後で CLEANUP ファイルによって通知できます。pam.conf ファイルを調べて、必要に応じて変更してください。

パスワードプロンプトおよびパスワード更新を主とする上記の変更に対して完全な自動更新ができないのは、同じスタックで使用されるその他のモジュールとの関係のためと、Sun 以外のモジュールがあるためです。

詳細については、pam_passwd_auth(5)、pam_authtok_get(5)、pam_authtok_store(5)、および pam.conf(4) のマニュアルページを参照してください。

LDAP コマンド

Solaris システムには、LDAP 関連のコマンドセットが 2 つ存在します。1 つのセットは一般的な LDAP ツールで、LDAP ネームサービスを使用してクライアントを構成する必要はありません。もう 1 つのセットはクライアント上の共通 LDAP 構成を使用するため、クライアントがネームサービスに LDAP を使用する場合にのみ使用できます。

一般的な LDAP ツール

LDAP コマンド行ツールは、認証やバインドパラメータを含む、一般的なオプションセットをサポートします。次のツールは、LDAP データ交換フォーマット (LDIF) というディレクトリ情報を表現する共通のテキストベース書式をサポートします。これらのコマンドを使用して、ディレクトリエントリを直接操作できます。

ldapsearch(1)
ldapmodify(1)
ldapadd(1)
ldapdelete(1)

LDAP ネームサービスを必要とする LDAP ツール

表 14–3 LDAP ツール

ツール	機能
ldapaddent(1M)	LDAP コンテナ内に、/etc 内のファイルに対応するエントリを作成する。このツールを使用して、ファイルからディレクトリを生成できる。たとえば、/etc/passwd 形式のファイルを読み込んで、ディレクトリ内に passwd エントリを生成する
ldaplist(1)	ディレクトリから、さまざまなサービスの内容をリスト表示するのに使用する
idsconfig(1M)	LDAP ネームサービスクライアント対応の Sun Java System Directory Server の設定に使用する

pam_ldap に対応した pam.conf ファイルの例

```
#
# Authentication management
#
# login service (explicit because of pam_dial_auth)
#
login    auth requisite        pam_authtok_get.so.1
login    auth required        pam_dhkeys.so.1
login    auth required        pam_dial_auth.so.1
login    auth required        pam_unix_cred.so.1
login    auth sufficient        pam_unix_auth.so.1
login    auth required        pam_ldap.so.1
#
# rlogin service (explicit because of pam_rhost_auth)
#
rlogin    auth sufficient        pam_rhosts_auth.so.1
rlogin    auth requisite        pam_authtok_get.so.1
rlogin    auth required        pam_dhkeys.so.1
rlogin    auth required        pam_unix_cred.so.1
rlogin    auth sufficient        pam_unix_auth.so.1
rlogin    auth required        pam_ldap.so.1
#
# rsh service (explicit because of pam_rhost_auth,
# and pam_unix_auth for meaningful pam_setcred)
#
rsh    auth sufficient        pam_rhosts_auth.so.1
rsh    auth required        pam_unix_cred.so.1
```

```
#
# PPP service (explicit because of pam_dial_auth)
#
ppp    auth requisite        pam_authtok_get.so.1
ppp    auth required        pam_dhkeys.so.1
ppp    auth required        pam_dial_auth.so.1
ppp    auth sufficient        pam_unix_auth.so.1
ppp    auth required        pam_ldap.so.1
#
# Default definitions for Authentication management
# Used when service name is not explicitly mentioned for authentication
#
other    auth requisite        pam_authtok_get.so.1
other    auth required        pam_dhkeys.so.1
other    auth required        pam_unix_cred.so.1
other    auth sufficient        pam_unix_auth.so.1
other    auth required        pam_ldap.so.1
#
# passwd command (explicit because of a different authentication module)
#
passwd    auth sufficient        pam_passwd_auth.so.1
passwd    auth required        pam_ldap.so.1
#
# cron service (explicit because of non-usage of pam_roles.so.1)
#
cron    account required    pam_unix_account.so.1
#
# Default definition for Account management
# Used when service name is not explicitly mentioned for account management
#
other    account requisite    pam_roles.so.1
other    account required    pam_unix_account.so.1
#
# Default definition for Session management
# Used when service name is not explicitly mentioned for session management
#
other    session required    pam_unix_session.so.1
#
# Default definition for  Password management
# Used when service name is not explicitly mentioned for password management
#
other    password required    pam_dhkeys.so.1
other    password requisite    pam_authtok_get.so.1
other    password requisite    pam_authtok_check.so.1
other    password required    pam_authtok_store.so.1
#
# Support for Kerberos V5 authentication and example configurations can
# be found in the pam_krb5(5) man page under the "EXAMPLES" section.
#
```

アカウント管理のために pam_ldap を構成した pam.conf ファイル例

注 – 以前は、pam_ldap アカウント管理を有効にすると、システムにログインする際には、常にすべてのユーザーが認証用にログインパスワードを入力する必要がありました。そのため、rsh、rlogin、ssh などのツールによるパスワードを使用しないログインは失敗します。

一方、pam_ldap(5) を Sun Java System Directory Server DS5.2p4 以降のリリースで使用することで、ユーザーはパスワードを入力せずに、rsh、rlogin、rcp、および ssh を使ってログインできるようになりました。

pam_ldap(5) は変更され、ユーザーのログイン時に Directory Server への認証を実行せずに、アカウントの管理およびユーザーのアカウント状態の取得を実行できるようになりました。Directory Server 上でこの機能を制御するのは、1.3.6.1.4.1.42.2.27.9.5.8 です。これは、デフォルトで有効になっています。

この制御をデフォルト以外に変更する場合は、Directory Server 上でアクセス制御情報 (ACI) を追加します。

```
dn: oid=1.3.6.1.4.1.42.2.27.9.5.8,cn=features,cn=config
objectClass: top
objectClass: directoryServerFeature
oid:1.3.6.1.4.1.42.2.27.9.5.8
cn:Password Policy Account Usable Request Control
aci: (targetattr != "aci")(version 3.0; acl "Account Usable";
     allow (read, search, compare, proxy)
     (groupdn = "ldap:///cn=Administrators,cn=config");)
creatorsName: cn=server,cn=plugins,cn=config
modifiersName: cn=server,cn=plugins,cn=config
```

```
#
# Authentication management
#
# login service (explicit because of pam_dial_auth)
#
login   auth requisite        pam_authtok_get.so.1
login   auth required         pam_dhkeys.so.1
login   auth required         pam_unix_cred.so.1
login   auth required         pam_dial_auth.so.1
login   auth binding          pam_unix_auth.so.1 server_policy
login   auth required         pam_ldap.so.1
#
# rlogin service (explicit because of pam_rhost_auth)
#
rlogin  auth sufficient       pam_rhosts_auth.so.1
rlogin  auth requisite        pam_authtok_get.so.1
rlogin  auth required         pam_dhkeys.so.1
rlogin  auth required         pam_unix_cred.so.1
```

```
rlogin  auth binding          pam_unix_auth.so.1 server_policy
rlogin  auth required         pam_ldap.so.1
#
# rsh service (explicit because of pam_rhost_auth,
# and pam_unix_auth for meaningful pam_setcred)
#
rsh     auth sufficient       pam_rhosts_auth.so.1
rsh     auth required         pam_unix_cred.so.1
rsh     auth binding          pam_unix_auth.so.1 server_policy
rsh     auth required         pam_ldap.so.1
#
# PPP service (explicit because of pam_dial_auth)
#
ppp     auth requisite        pam_authtok_get.so.1
ppp     auth required         pam_dhkeys.so.1
ppp     auth required         pam_dial_auth.so.1
ppp     auth binding          pam_unix_auth.so.1 server_policy
ppp     auth required         pam_ldap.so.1
#
# Default definitions for Authentication management
# Used when service name is not explicitly mentioned for authentication
#
other   auth requisite        pam_authtok_get.so.1
other   auth required         pam_dhkeys.so.1
other   auth required         pam_unix_cred.so.1
other   auth binding          pam_unix_auth.so.1 server_policy
other   auth required         pam_ldap.so.1
#
# passwd command (explicit because of a different authentication module)
#
passwd  auth binding          pam_passwd_auth.so.1 server_policy
passwd  auth required         pam_ldap.so.1
#
# cron service (explicit because of non-usage of pam_roles.so.1)
#
cron    account required      pam_unix_account.so.1
#
# Default definition for Account management
# Used when service name is not explicitly mentioned for account management
#
other   account requisite     pam_roles.so.1
other   account binding       pam_unix_account.so.1 server_policy
other   account required      pam_ldap.so.1
#
# Default definition for Session management
# Used when service name is not explicitly mentioned for session management
#
other   session required      pam_unix_session.so.1
#
# Default definition for  Password management
# Used when service name is not explicitly mentioned for password management
#
other   password required     pam_dhkeys.so.1
other   password requisite    pam_authtok_get.so.1
other   password requisite    pam_authtok_check.so.1
other   password required     pam_authtok_store.so.1 server_policy
#
# Support for Kerberos V5 authentication and example configurations can
# be found in the pam_krb5(5) man page under the "EXAMPLES" section.
#
```

LDAP 用の IETF スキーマ

スキーマは、サーバーのディレクトリ内にエントリとして格納可能な情報タイプを記述した定義です。

ディレクトリサーバーが Solaris LDAP ネームサービスクライアントをサポートするには、クライアントのスキーママッピング機能を使用してスキーマをマッピングしていない限り、この章で定義されたスキーマをサーバー内で構成する必要があります。

IETF により定義された必須 LDAP スキーマは次の 3 つです。 RFC 2307 ネットワーク情報サービススキーマ、LDAP メールグループインターネットドラフト、および LDAP Internet Pint Protocol (IPP) ドラフトスキーマ。ネーム情報サービスをサポートするには、これらのスキーマ定義をディレクトリサーバーに追加する必要があります。IETF Web サイト `http://www.ietf.org` で、さまざまな RFC にアクセスできます。

注 – インターネットドラフトとは、最長 6 カ月間有効なドラフトの文書で、ほかの文書によっていつでも更新または廃止される可能性があります。

RFC 2307 ネットワーク情報サービススキーマ

LDAP サーバーは改訂版 RFC 2307 をサポートするように構成する必要があります。

`nisSchema` OID は `1.3.6.1.1` です。RFC 2307 属性を次に示します。

```
( nisSchema.1.0 NAME 'uidNumber'
DESC 'An integer uniquely identifying a user in an
        administrative domain'
EQUALITY integerMatch SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.1 NAME 'gidNumber'
DESC 'An integer uniquely identifying a group in an
        administrative domain'
EQUALITY integerMatch SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.2 NAME 'gecos'
DESC 'The GECOS field; the common name'
EQUALITY caseIgnoreIA5Match
SUBSTRINGS caseIgnoreIA5SubstringsMatch
SYNTAX 'IA5String' SINGLE-VALUE )

( nisSchema.1.3 NAME 'homeDirectory'
DESC 'The absolute path to the home directory'
EQUALITY caseExactIA5Match
SYNTAX 'IA5String' SINGLE-VALUE )

( nisSchema.1.4 NAME 'loginShell'
```

```
DESC 'The path to the login shell'
EQUALITY caseExactIA5Match
SYNTAX 'IA5String' SINGLE-VALUE )

( nisSchema.1.5 NAME 'shadowLastChange'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.6 NAME 'shadowMin'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.7 NAME 'shadowMax'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.8 NAME 'shadowWarning'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.9 NAME 'shadowInactive'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.10 NAME 'shadowExpire'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.11 NAME 'shadowFlag'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.12 NAME 'memberUid'
EQUALITY caseExactIA5Match
SUBSTRINGS caseExactIA5SubstringsMatch
SYNTAX 'IA5String' )

( nisSchema.1.13 NAME 'memberNisNetgroup'
EQUALITY caseExactIA5Match
SUBSTRINGS caseExactIA5SubstringsMatch
SYNTAX 'IA5String' )

( nisSchema.1.14 NAME 'nisNetgroupTriple'
DESC 'Netgroup triple'
SYNTAX 'nisNetgroupTripleSyntax' )

( nisSchema.1.15 NAME 'ipServicePort'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.16 NAME 'ipServiceProtocol'
SUP name )

( nisSchema.1.17 NAME 'ipProtocolNumber'
EQUALITY integerMatch
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.18 NAME 'oncRpcNumber'
EQUALITY integerMatch
```

```
SYNTAX 'INTEGER' SINGLE-VALUE )

( nisSchema.1.19 NAME 'ipHostNumber'
DESC 'IP address as a dotted decimal, eg. 192.168.1.1
        omitting leading zeros'
SUP name )

( nisSchema.1.20 NAME 'ipNetworkNumber'
DESC 'IP network as a dotted decimal, eg. 192.168,
        omitting leading zeros'
SUP name SINGLE-VALUE )

( nisSchema.1.21 NAME 'ipNetmaskNumber'
DESC 'IP netmask as a dotted decimal, eg. 255.255.255.0,
         omitting leading zeros'
EQUALITY caseIgnoreIA5Match
SYNTAX 'IA5String{128}' SINGLE-VALUE )

( nisSchema.1.22 NAME 'macAddress'
DESC 'MAC address in maximal, colon separated hex
      notation, eg. 00:00:92:90:ee:e2'
EQUALITY caseIgnoreIA5Match
SYNTAX 'IA5String{128}' )

( nisSchema.1.23 NAME 'bootParameter'
DESC 'rpc.bootparamd parameter'
SYNTAX 'bootParameterSyntax' )

( nisSchema.1.24 NAME 'bootFile'
DESC 'Boot image name'
EQUALITY caseExactIA5Match
SYNTAX 'IA5String' )

( nisSchema.1.26 NAME 'nisMapName'
SUP name )

( nisSchema.1.27 NAME 'nisMapEntry'
EQUALITY caseExactIA5Match
SUBSTRINGS caseExactIA5SubstringsMatch
SYNTAX 'IA5String{1024}' SINGLE-VALUE )

( nisSchema.1.28 NAME 'nisPublicKey'
DESC 'NIS public key'
SYNTAX 'nisPublicKeySyntax' )

( nisSchema.1.29 NAME 'nisSecretKey'
DESC 'NIS secret key'
SYNTAX 'nisSecretKeySyntax' )

( nisSchema.1.30 NAME 'nisDomain'
DESC 'NIS domain'
SYNTAX 'IA5String' )

( nisSchema.1.31 NAME 'automountMapName'
DESC 'automount Map Name'
EQUALITY caseExactIA5Match
SUBSTR caseExactIA5SubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

```
( nisSchema.1.32 NAME 'automountKey'
DESC 'Automount Key value'
EQUALITY caseExactIA5Match
SUBSTR caseExactIA5SubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )

( nisSchema.1.33 NAME 'automountInformation'
DESC 'Automount information'
EQUALITY caseExactIA5Match
SUBSTR caseExactIA5SubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

nisSchema OID は 1.3.6.1.1 です。RFC 2307 objectClasses を次に示します。

```
( nisSchema.2.0 NAME 'posixAccount' SUP top AUXILIARY
  DESC 'Abstraction of an account with POSIX attributes'
  MUST ( cn $ uid $ uidNumber $ gidNumber $ homeDirectory )
  MAY ( userPassword $ loginShell $ gecos $ description ) )

( nisSchema.2.1 NAME 'shadowAccount' SUP top AUXILIARY
  DESC 'Additional attributes for shadow passwords'
  MUST uid
  MAY ( userPassword $ shadowLastChange $ shadowMin
        shadowMax $ shadowWarning $ shadowInactive $
        shadowExpire $ shadowFlag $ description ) )

( nisSchema.2.2 NAME 'posixGroup' SUP top STRUCTURAL
  DESC 'Abstraction of a group of accounts'
  MUST ( cn $ gidNumber )
  MAY ( userPassword $ memberUid $ description ) )

( nisSchema.2.3 NAME 'ipService' SUP top STRUCTURAL
  DESC 'Abstraction an Internet Protocol service.
        Maps an IP port and protocol (such as tcp or udp)
        to one or more names; the distinguished value of
        the cn attribute denotes the service's canonical
        name'
  MUST ( cn $ ipServicePort $ ipServiceProtocol )
  MAY ( description ) )

( nisSchema.2.4 NAME 'ipProtocol' SUP top STRUCTURAL
  DESC 'Abstraction of an IP protocol. Maps a protocol number
        to one or more names. The distinguished value of the cn
        attribute denotes the protocol's canonical name'
  MUST ( cn $ ipProtocolNumber )
  MAY  description )

( nisSchema.2.5 NAME 'oncRpc' SUP top STRUCTURAL
  DESC 'Abstraction of an Open Network Computing (ONC)
        [RFC1057] Remote Procedure Call (RPC) binding.
        This class maps an ONC RPC number to a name.
        The distinguished value of the cn attribute denotes
        the RPC service's canonical name'
  MUST ( cn $ oncRpcNumber $ description )
  MAY  description )

( nisSchema.2.6 NAME 'ipHost' SUP top AUXILIARY
  DESC 'Abstraction of a host, an IP device. The distinguished
```

```
        value of the cn attribute denotes the host's canonical
        name. Device SHOULD be used as a structural class'
  MUST ( cn $ ipHostNumber )
  MAY ( l $ description $ manager $ userPassword ) )

( nisSchema.2.7 NAME 'ipNetwork' SUP top STRUCTURAL
  DESC 'Abstraction of a network. The distinguished value of
        the cn attribute denotes the network's canonical name'
  MUST ipNetworkNumber
  MAY ( cn $ ipNetmaskNumber $ l $ description $ manager ) )

( nisSchema.2.8 NAME 'nisNetgroup' SUP top STRUCTURAL
  DESC 'Abstraction of a netgroup. May refer to other netgroups'
  MUST cn
  MAY ( nisNetgroupTriple $ memberNisNetgroup $ description ) )

( nisSchema.2.9 NAME 'nisMap' SUP top STRUCTURAL
  DESC 'A generic abstraction of a NIS map'
  MUST nisMapName
  MAY description )

( nisSchema.2.10 NAME 'nisObject' SUP top STRUCTURAL
  DESC 'An entry in a NIS map'
  MUST ( cn $ nisMapEntry $ nisMapName )
  MAY description )

( nisSchema.2.11 NAME 'ieee802Device' SUP top AUXILIARY
  DESC 'A device with a MAC address; device SHOULD be
        used as a structural class'
  MAY macAddress )

( nisSchema.2.12 NAME 'bootableDevice' SUP top AUXILIARY
  DESC 'A device with boot parameters; device SHOULD be
  used as a structural class'
  MAY ( bootFile $ bootParameter ) )

( nisSchema.2.14 NAME 'nisKeyObject' SUP top AUXILIARY
  DESC 'An object with a public and secret key'
  MUST ( cn $ nisPublicKey $ nisSecretKey )
  MAY ( uidNumber $ description ) )

( nisSchema.2.15 NAME 'nisDomainObject' SUP top AUXILIARY
  DESC 'Associates a NIS domain with a naming context'
  MUST nisDomain )

( nisSchema.2.16 NAME 'automountMap' SUP top STRUCTURAL
  MUST ( automountMapName )
  MAY description )

( nisSchema.2.17 NAME 'automount' SUP top STRUCTURAL
  DESC 'Automount information'
  MUST ( automountKey $ automountInformation )
  MAY description )
```

メールエイリアススキーマ

メールエイリアス情報は、LDAP メールグループインターネットドラフト (以前は draft-steinback-ldap-mailgroups ドラフトと呼ばれていた) で定義されたスキーマを使用します。新しいスキーマが使用可能になるまで、Solaris LDAP クライアントは、このメールエイリアス情報のスキーマの使用を続けます。

インターネットドラフトに定義された LDAP メールグループスキーマには、多数の属性とオブジェクトクラスが含まれています。このうち、Solaris クライアントが使用するのは、2 つの属性と 1 つのオブジェクトクラスだけです。次にその内容を示します。

メールエイリアス属性を次に示します。

```
( 0.9.2342.19200300.100.1.3
  NAME 'mail'
  DESC 'RFC822 email address for this person'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 'IA5String(256)'
  SINGLE-VALUE )

( 2.16.840.1.113730.3.1.30
  NAME 'mgrpRFC822MailMember'
  DESC 'RFC822 mail address of email only member of group'
  EQUALITY CaseIgnoreIA5Match
  SYNTAX 'IA5String(256)' )
```

メールエイリアス `objectClass` を次に示します。

```
( 2.16.840.1.113730.3.2.4
  NAME 'mailGroup'
  SUP top
  STRUCTURAL
  MUST mail
  MAY ( cn $ mailAlternateAddress $ mailHost $ mailRequireAuth $
   mgrpAddHeader $ mgrpAllowedBroadcaster $ mgrpAllowedDomain $
   mgrpApprovePassword $ mgrpBroadcasterModeration $ mgrpDeliverTo $
   mgrpErrorsTo $ mgrpModerator $ mgrpMsgMaxSize $
   mgrpMsgRejectAction $ mgrpMsgRejectText $ mgrpNoMatchAddrs $
   mgrpRemoveHeader $ mgrpRFC822MailMember ))
```

ディレクトリユーザーエージェントのプロファイル (DUAProfile) スキーマ

`DUAConfSchemaOID` は、`1.3.6.1.4.1.11.1.3.1` です。

```
DESC 'Default LDAP server host address used by a DUA'
            EQUALITY caseIgnoreMatch
            SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
```

```
  SINGLE-VALUE )

( DUAConfSchemaOID.1.1 NAME 'defaultSearchBase'
  DESC 'Default LDAP base DN used by a DUA'
  EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
  SINGLE-VALUE )

( DUAConfSchemaOID.1.2 NAME 'preferredServerList'
  DESC 'Preferred LDAP server host addresses to be used by a
  DUA'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE )

( DUAConfSchemaOID.1.3 NAME 'searchTimeLimit'
  DESC 'Maximum time in seconds a DUA should allow for a
  search to complete'
  EQUALITY integerMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
  SINGLE-VALUE )

( DUAConfSchemaOID.1.4 NAME 'bindTimeLimit'
  DESC 'Maximum time in seconds a DUA should allow for the
  bind operation to complete'
  EQUALITY integerMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
  SINGLE-VALUE )

( DUAConfSchemaOID.1.5 NAME 'followReferrals'
  DESC 'Tells DUA if it should follow referrals
  returned by a DSA search result'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
  SINGLE-VALUE )

( DUAConfSchemaOID.1.6 NAME 'authenticationMethod'
  DESC 'A keystring which identifies the type of
  authentication method used to contact the DSA'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE )

( DUAConfSchemaOID.1.7 NAME 'profileTTL'
  DESC 'Time to live, in seconds, before a client DUA
  should re-read this configuration profile'
      'serviceSearchDescriptor'
  DESC 'LDAP search descriptor list used by a DUA'
  EQUALITY caseExactMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

( DUAConfSchemaOID.1.9 NAME 'attributeMap'
  DESC 'Attribute mappings used by a DUA'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

( DUAConfSchemaOID.1.10 NAME 'credentialLevel'
  DESC 'Identifies type of credentials a DUA should
  use when binding to the LDAP server'
```

```
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE )

( DUAConfSchemaOID.1.11 NAME 'objectclassMap'
  DESC 'Objectclass mappings used by a DUA'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

( DUAConfSchemaOID.1.12 NAME 'defaultSearchScope' SINGLE-VALUE )

( DUAConfSchemaOID.1.13 NAME 'serviceCredentialLevel'
  DESC 'Identifies type of credentials a DUA
  should use when binding to the LDAP server for a
  specific service'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

( DUAConfSchemaOID.1.15 NAME 'serviceAuthenticationMethod'
  DESC 'Authentication Method used by a service of the DUA'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

    ( DUAConfSchemaOID.2.4 NAME 'DUAConfigProfile'
        SUP top STRUCTURAL
     DESC 'Abstraction of a base configuration for a DUA'
     MUST ( cn )
     MAY ( defaultServerList $ preferredServerList $
    defaultSearchBase $ defaultSearchScope $
    searchTimeLimit $ bindTimeLimit $
    credentialLevel $ authenticationMethod $
    followReferrals $ serviceSearchDescriptor $
    serviceCredentialLevel $ serviceAuthenticationMethod $
    objectclassMap $ attributeMap $
    profileTTL ) )
```

Solaris スキーマ

Solaris プラットフォームに必要なスキーマを次に示します。

- Solaris プロジェクトスキーマ
- アクセス制御および実行プロファイルスキーマに基づく役割
- プリンタスキーマ

Solaris プロジェクトスキーマ

`/etc/project` は、プロジェクトと関連のある属性のローカルソースです。詳細については、`user_attr(4)` のマニュアルページを参照してください。

プロジェクト属性を次に示します。

```
( 1.3.6.1.4.1.42.2.27.5.1.1 NAME 'SolarisProjectID'
  DESC 'Unique ID for a Solaris Project entry'
```

```
  EQUALITY integerMatch
  SYNTAX INTEGER SINGLE )

( 1.3.6.1.4.1.42.2.27.5.1.2 NAME 'SolarisProjectName'
  DESC 'Name of a Solaris Project entry'
  EQUALITY caseExactIA5Match
  SYNTAX IA5String SINGLE )

( 1.3.6.1.4.1.42.2.27.5.1.3 NAME 'SolarisProjectAttr'
  DESC 'Attributes of a Solaris Project entry'
  EQUALITY caseExactIA5Match
  SYNTAX IA5String )

( 1.3.6.1.4.1.42.2.27.5.1.30 NAME 'memberGid'
  DESC 'Posix Group Name'
  EQUALITY caseExactIA5Match
  SYNTAX 'IA5String' )
```

プロジェクト `objectClass` を次に示します。

```
( 1.3.6.1.4.1.42.2.27.5.2.1 NAME 'SolarisProject'
  SUP top STRUCTURAL
  MUST ( SolarisProjectID $ SolarisProjectName )
  MAY ( memberUid $ memberGid $ description $ SolarisProjectAttr ) )
```

役割ベースのアクセス制御と実行プロファイルスキーマ

ユーザーと役割に関する拡張属性のシステムごとの設定は、`/etc/user_attr` に置かれます。詳細については、`user_attr(4)` のマニュアルページを参照してください。

役割によるアクセス制御属性を次に示します。

```
( 1.3.6.1.4.1.42.2.27.5.1.4 NAME 'SolarisAttrKeyValue'
  DESC 'Semi-colon separated key=value pairs of attributes'
  EQUALITY caseIgnoreIA5Match
  SUBSTRINGS caseIgnoreIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )

( 1.3.6.1.4.1.42.2.27.5.1.7 NAME 'SolarisAttrShortDesc'
  DESC 'Short description about an entry, used by GUIs'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )

( 1.3.6.1.4.1.42.2.27.5.1.8 NAME 'SolarisAttrLongDesc'
  DESC 'Detail description about an entry'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )

( 1.3.6.1.4.1.42.2.27.5.1.9 NAME 'SolarisKernelSecurityPolicy'
  DESC 'Solaris  kernel security policy'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )
```

```
( 1.3.6.1.4.1.42.2.27.5.1.10 NAME 'SolarisProfileType'
  DESC 'Type of object defined in profile'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )

( 1.3.6.1.4.1.42.2.27.5.1.11 NAME 'SolarisProfileId'
  DESC 'Identifier of object defined in profile'
  EQUALITY caseExactIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )

( 1.3.6.1.4.1.42.2.27.5.1.12 NAME 'SolarisUserQualifier'
  DESC 'Per-user login attributes'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )

( 1.3.6.1.4.1.42.2.27.5.1.13 NAME 'SolarisReserved1'
  DESC 'Reserved for future use'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )

( 1.3.6.1.4.1.42.2.27.5.1.14 NAME 'SolarisReserved2'
  DESC 'Reserved for future use'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE )
```

役割によるアクセス制御 `objectClassses` を次に示します。

```
( 1.3.6.1.4.1.42.2.27.5.2.3 NAME 'SolarisUserAttr' SUP top AUXILIARY
  DESC 'User attributes'
  MAY ( SolarisUserQualifier $ SolarisAttrReserved1 $ \
        SolarisAttrReserved2 $ SolarisAttrKeyValue ) )

( 1.3.6.1.4.1.42.2.27.5.2.4 NAME 'SolarisAuthAttr' SUP top STRUCTURAL
  DESC 'Authorizations data'
  MUST cn
  MAY ( SolarisAttrReserved1 $ SolarisAttrReserved2 $ \
        SolarisAttrShortDesc $ SolarisAttrLongDesc $ \
        SolarisAttrKeyValue ) )

( 1.3.6.1.4.1.42.2.27.5.2.5 NAME 'SolarisProfAttr' SUP top STRUCTURAL
  DESC 'Profiles data'
  MUST cn
  MAY ( SolarisAttrReserved1 $ SolarisAttrReserved2 $ \
        SolarisAttrLongDesc $ SolarisAttrKeyValue ) )

( 1.3.6.1.4.1.42.2.27.5.2.6 NAME 'SolarisExecAttr' SUP top AUXILIARY
  DESC 'Profiles execution attributes'
  MAY ( SolarisKernelSecurityPolicy $ SolarisProfileType $ \
        SolarisAttrReserved1 $ SolarisAttrReserved2 $ \
        SolarisProfileId $ SolarisAttrKeyValue ) )
```

LDAP 用の Internet Printing Protocol 情報

以降の節では、Internet Printing Protocol と Sun プリンタの属性と ObjectClasses について説明します。

Internet Print Protocol (IPP) 属性

```
( 1.3.18.0.2.4.1140
NAME 'printer-uri'
DESC 'A URI supported by this printer.
This URI SHOULD be used as a relative distinguished name (RDN).
If printer-xri-supported is implemented, then this URI value
MUST be listed in a member value of printer-xri-supported.'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )

( 1.3.18.0.2.4.1107
NAME 'printer-xri-supported'
DESC 'The unordered list of XRI (extended resource identifiers) supported
by this printer.
Each member of the list consists of a URI (uniform resource identifier)
followed by optional authentication and security metaparameters.'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

( 1.3.18.0.2.4.1135
NAME 'printer-name'
DESC 'The site-specific administrative name of this printer, more end-user
friendly than a URI.'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127}  SINGLE-VALUE )

( 1.3.18.0.2.4.1119
NAME 'printer-natural-language-configured'
DESC 'The configured language in which error and status messages will be
generated (by default) by this printer.
Also, a possible language for printer string attributes set by operator,
system administrator, or manufacturer.
Also, the (declared) language of the "printer-name", "printer-location",
"printer-info", and "printer-make-and-model" attributes of this printer.
For example: "en-us" (US English) or "fr-fr" (French in France) Legal values of
language tags conform to [RFC3066] "Tags for the Identification of Languages".'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127}  SINGLE-VALUE )
```

```
( 1.3.18.0.2.4.1136
NAME 'printer-location'
DESC 'Identifies the location of the printer. This could include
things like: "in Room 123A", "second floor of building XYZ".'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} SINGLE-VALUE )

( 1.3.18.0.2.4.1139
NAME 'printer-info'
DESC 'Identifies the descriptive information about this printer.
This could include things like: "This printer can be used for
printing color transparencies for HR presentations", or
"Out of courtesy for others, please print only small (1-5 page)
jobs at this printer", or even "This printer is going away on July 1, 1997,
please find a new printer".'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127}
SINGLE-VALUE )

( 1.3.18.0.2.4.1134
NAME 'printer-more-info'
DESC 'A URI used to obtain more information about this specific printer.
For example, this could be an HTTP type URI referencing an HTML page
accessible to a Web Browser.
The information obtained from this URI is intended for end user consumption.'
EQUALITY caseIgnoreMatch ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )

( 1.3.18.0.2.4.1138
NAME 'printer-make-and-model'
DESC 'Identifies the make and model of the device.
The device manufacturer MAY initially populate this attribute.'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{127}  SINGLE-VALUE )

( 1.3.18.0.2.4.1133
NAME 'printer-ipp-versions-supported'
DESC 'Identifies the IPP protocol version(s) that this printer supports,
including major and minor versions,
i.e., the version numbers for which this Printer implementation meets
the conformance requirements.'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} )

( 1.3.18.0.2.4.1132
NAME 'printer-multiple-document-jobs-supported'
DESC 'Indicates whether or not the printer supports more than one
document per job, i.e., more than one Send-Document or Send-Data
operation with document data.'
EQUALITY booleanMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE )
```

```
( 1.3.18.0.2.4.1109
NAME 'printer-charset-configured'
DESC 'The configured charset in which error and status messages will be
generated (by default) by this printer.
Also, a possible charset for printer string attributes set by operator,
system administrator, or manufacturer.
For example: "utf-8" (ISO 10646/Unicode) or "iso-8859-1" (Latin1).
Legal values are defined by the IANA Registry of Coded Character Sets and
the "(preferred MIME name)" SHALL be used as the tag.
For coherence with IPP Model, charset tags in this attribute SHALL be
lowercase normalized.
This attribute SHOULD be static (time of registration) and SHOULD NOT be
dynamically refreshed attributetypes: (subsequently).'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{63} SINGLE-VALUE )

( 1.3.18.0.2.4.1131
NAME 'printer-charset-supported'
DESC 'Identifies the set of charsets supported for attribute type values of
type Directory String for this directory entry.
For example: "utf-8" (ISO 10646/Unicode) or "iso-8859-1" (Latin1).
Legal values are defined by the IANA Registry of Coded Character Sets and
the preferred MIME name.'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{63} )

( 1.3.18.0.2.4.1137
NAME 'printer-generated-natural-language-supported'
DESC 'Identifies the natural language(s) supported for this directory entry.
For example: "en-us" (US English) or "fr-fr" (French in France).
Legal values conform to [RFC3066], Tags for the Identification of Languages.'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{63} )

( 1.3.18.0.2.4.1130
NAME 'printer-document-format-supported'
DESC 'The possible document formats in which data may be interpreted
and printed by this printer.
Legal values are MIME types come from the IANA Registry of Internet Media Types.'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} )

( 1.3.18.0.2.4.1129
NAME 'printer-color-supported'
DESC 'Indicates whether this printer is capable of any type of color printing
at all, including highlight color.'
EQUALITY booleanMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.7  SINGLE-VALUE )

( 1.3.18.0.2.4.1128
NAME 'printer-compression-supported'
DESC 'Compression algorithms supported by this printer.
For example: "deflate, gzip".  Legal values include; "none", "deflate"
attributetypes: (public domain ZIP), "gzip" (GNU ZIP), "compress" (UNIX).'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{255} )
```

```
( 1.3.18.0.2.4.1127
NAME 'printer-pages-per-minute'
DESC 'The nominal number of pages per minute which may be output by this
printer (e.g., a simplex or black-and-white printer).
This attribute is informative, NOT a service guarantee.
Typically, it is the value used in marketing literature to describe this printer.'
EQUALITY integerMatch
ORDERING integerOrderingMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.27  SINGLE-VALUE )

( 1.3.18.0.2.4.1126 NAME 'printer-pages-per-minute-color'
DESC 'The nominal number of color pages per minute which may be output by this
printer (e.g., a simplex or color printer).
This attribute is informative, NOT a service guarantee.
Typically, it is the value used in marketing literature to describe this printer.'
EQUALITY integerMatch
ORDERING integerOrderingMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.27  SINGLE-VALUE )

( 1.3.18.0.2.4.1125 NAME 'printer-finishings-supported'
DESC 'The possible finishing operations supported by this printer.
Legal values include; "none", "staple", "punch", "cover", "bind", "saddle-stitch",
"edge-stitch", "staple-top-left", "staple-bottom-left", "staple-top-right",
"staple-bottom-right", "edge-stitch-left", "edge-stitch-top", "edge-stitch-right",
"edge-stitch-bottom", "staple-dual-left", "staple-dual-top", "staple-dual-right",
"staple-dual-bottom".'
EQUALITY caseIgnoreMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{255} )

( 1.3.18.0.2.4.1124 NAME 'printer-number-up-supported'
DESC 'The possible numbers of print-stream pages to impose upon a single side of
an instance of a selected medium. Legal values include; 1, 2, and 4.
Implementations may support other values.'
EQUALITY integerMatch
ORDERING integerOrderingMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.27 )

( 1.3.18.0.2.4.1123 NAME 'printer-sides-supported'
DESC 'The number of impression sides (one or two) and the two-sided impression
rotations supported by this printer.
Legal values include; "one-sided", "two-sided-long-edge", "two-sided-short-edge".'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} )

( 1.3.18.0.2.4.1122 NAME 'printer-media-supported'
DESC 'The standard names/types/sizes (and optional color suffixes) of the media
supported by this printer.
For example: "iso-a4",  "envelope", or "na-letter-white".
Legal values  conform to ISO 10175, Document Printing Application (DPA), and any
IANA registered extensions.'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{255} )

( 1.3.18.0.2.4.1117 NAME 'printer-media-local-supported'
DESC 'Site-specific names of media supported by this printer, in the language in
"printer-natural-language-configured".
```

```
For example: "purchasing-form" (site-specific name) as opposed to
(in "printer-media-supported"): "na-letter" (standard keyword from ISO 10175).'
EQUALITY caseIgnoreMatch SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{255} )

( 1.3.18.0.2.4.1121 NAME 'printer-resolution-supported'
DESC 'List of resolutions supported for printing documents by this printer.
Each resolution value is a string with 3 fields:
1) Cross feed direction resolution (positive integer), 2) Feed direction
resolution (positive integer), 3) Resolution unit.
Legal values are "dpi" (dots per inch) and "dpcm" (dots per centimeter).
Each resolution field is delimited by ">".  For example:  "300> 300> dpi>".'
EQUALITY caseIgnoreMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{255} )

( 1.3.18.0.2.4.1120 NAME 'printer-print-quality-supported'
DESC 'List of print qualities supported for printing documents on this printer.
For example: "draft, normal".  Legal values include; "unknown", "draft", "normal",
"high".'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} )

( 1.3.18.0.2.4.1110 NAME 'printer-job-priority-supported'
DESC 'Indicates the number of job priority levels supported.
An IPP conformant printer which supports job priority must always support a
full range of priorities from "1" to "100"
(to ensure consistent behavior), therefore this attribute describes the
"granularity".
 Legal values of this attribute are from "1" to "100".'
EQUALITY integerMatch
ORDERING integerOrderingMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.27  SINGLE-VALUE )

( 1.3.18.0.2.4.1118
NAME 'printer-copies-supported'
DESC 'The maximum number of copies of a document that may be printed as a single job.
A value of "0" indicates no maximum limit.
A value of "-1" indicates unknown.'
EQUALITY integerMatch
ORDERING integerOrderingMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.27  SINGLE-VALUE )

( 1.3.18.0.2.4.1111
NAME 'printer-job-k-octets-supported'
DESC 'The maximum size in kilobytes (1,024 octets actually) incoming print job that
this printer will accept.
A value of "0" indicates no maximum limit.  A value of "-1" indicates unknown.'
EQUALITY integerMatch
ORDERING integerOrderingMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.27  SINGLE-VALUE )

( 1.3.18.0.2.4.1113
NAME 'printer-service-person'
DESC 'The name of the current human service person responsible for servicing this
printer.
It is suggested that this string include information that would enable other humans
```

```
to reach the service person, such as a phone number.'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127}
SINGLE-VALUE )

( 1.3.18.0.2.4.1114
NAME 'printer-delivery-orientation-supported'
DESC 'The possible delivery orientations of pages as they are printed and ejected
from this printer.
Legal values include; "unknown", "face-up", and "face-down".'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} )

( 1.3.18.0.2.4.1115
NAME 'printer-stacking-order-supported'
DESC 'The possible stacking order of pages as they are printed and ejected from
this printer.
Legal values include; "unknown", "first-to-last", "last-to-first".'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} )

( 1.3.18.0.2.4.1116
NAME 'printer-output-features-supported'
DESC 'The possible output features supported by this printer.
Legal values include; "unknown", "bursting", "decollating", "page-collating",
"offset-stacking".'
EQUALITY caseIgnoreMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} )

( 1.3.18.0.2.4.1108
NAME 'printer-aliases'
DESC 'Site-specific administrative names of this printer in addition the printer
name specified for printer-name.'
EQUALITY caseIgnoreMatch
ORDERING caseIgnoreOrderingMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15{127} )

( 1.3.6.1.4.1.42.2.27.5.1.63
NAME 'sun-printer-bsdaddr'
DESC 'Sets the server, print queue destination name and whether the client generates
protocol extensions.
"Solaris" specifies a Solaris print server extension. The value is represented b the
following value: server "," destination ", Solaris".'
SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' SINGLE-VALUE )

( 1.3.6.1.4.1.42.2.27.5.1.64
NAME 'sun-printer-kvp'
DESC 'This attribute contains a set of key value pairs which may have meaning to the
print subsystem or may be user defined.
Each value is represented by the following: key "=" value.'
SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )
```

Internet Print Protocol (IPP) ObjectClasses

```
objectclasses: ( 1.3.18.0.2.6.2549
NAME 'slpService'
DESC 'DUMMY definition'
SUP 'top' MUST (objectclass) MAY ())

objectclasses: ( 1.3.18.0.2.6.254
NAME 'slpServicePrinter'
DESC 'Service Location Protocol (SLP) information.'
AUXILIARY SUP 'slpService')

objectclasses: ( 1.3.18.0.2.6.258
NAME 'printerAbstract'
DESC 'Printer related information.'
ABSTRACT SUP 'top' MAY ( printer-name
$ printer-natural-language-configured
$ printer-location
$ printer-info
$ printer-more-info
$ printer-make-and-model
$ printer-multiple-document-jobs-supported
$ printer-charset-configured
$ printer-charset-supported
$ printer-generated-natural-language-supported
$ printer-document-format-supported
$ printer-color-supported
$ printer-compression-supported
$ printer-pages-per-minute
$ printer-pages-per-minute-color
$ printer-finishings-supported
$ printer-number-up-supported
$ printer-sides-supported
$ printer-media-supported
$ printer-media-local-supported
$ printer-resolution-supported
$ printer-print-quality-supported
$ printer-job-priority-supported
$ printer-copies-supported
$ printer-job-k-octets-supported
$ printer-current-operator
$ printer-service-person
$ printer-delivery-orientation-supported
$ printer-stacking-order-supported $ printer! -output-features-supported ))

objectclasses: ( 1.3.18.0.2.6.255
NAME 'printerService'
DESC 'Printer information.'
STRUCTURAL SUP 'printerAbstract' MAY ( printer-uri
$ printer-xri-supported ))

objectclasses: ( 1.3.18.0.2.6.257
NAME 'printerServiceAuxClass'
DESC 'Printer information.'
AUXILIARY SUP 'printerAbstract' MAY ( printer-uri $ printer-xri-supported ))
```

```
objectclasses: ( 1.3.18.0.2.6.256
NAME 'printerIPP'
DESC 'Internet Printing Protocol (IPP) information.'
AUXILIARY SUP 'top' MAY   ( printer-ipp-versions-supported $
printer-multiple-document-jobs-supported ))

objectclasses: ( 1.3.18.0.2.6.253
NAME 'printerLPR'
DESC 'LPR information.'
AUXILIARY SUP 'top' MUST ( printer-name ) MAY ( printer-aliases))

objectclasses: ( 1.3.6.1.4.1.42.2.27.5.2.14
NAME 'sunPrinter'
DESC 'Sun printer information'
SUP 'top' AUXILIARY MUST (objectclass $ printer-name)  MAY
(sun-printer-bsdaddr $ sun-printer-kvp))
```

Sun プリンタ属性

```
ATTRIBUTE ( 1.3.6.1.4.1.42.2.27.5.1.63
NAME sun-printer-bsdaddr
DESC 'Sets the server, print queue destination name and whether the
     client generates protocol extensions. "Solaris" specifies a
     Solaris print server extension.  The value is represented by
     the following value: server "," destination ", Solaris".'
EQUALITY caseIgnoreIA5Match
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE
)


ATTRIBUTE ( 1.3.6.1.4.1.42.2.27.5.1.64
NAME sun-printer-kvp
DESC 'This attribute contains a set of key value pairs which may have
      meaning to the print subsystem or may be user defined.  Each
      value is represented by the following: key "=" value.'
EQUALITY caseIgnoreIA5Match
SYNTAX  1.3.6.1.4.1.1466.115.121.1.15  )
```

Sun プリンタ ObjectClasses

```
OBJECTCLASS ( 1.3.6.1.4.1.42.2.27.5.2.14
NAME sunPrinter
DESC 'Sun printer information'
SUP  top
AUXILIARY
MUST ( printer-name )
MAY  ( sun-printer-bsdaddr $ sun-printer-kvp ))
```

LDAP 用の汎用ディレクトリサーバーの要件

Solaris 9 以降のリリースで LDAP クライアントをサポートする場合、サーバーの種類に関係なく、LDAP v3 プロトコル、複合ネーミングおよび補助オブジェクトクラスをサポートする必要があります。また、次の制御を 1 つ以上サポートする必要があります。

- 単純ページモード (RFC 2696)
- 仮想リスト表示制御

  サーバーは、次の認証方式を 1 つ以上サポートする必要があります。

```
anonymous
simple
sasl/cram-MD5
sasl/digest-MD5
sasl/GSSAPI
```

`pam_unix` を使用する場合、サーバーは UNIX 暗号形式 (crypt) でのパスワード保管をサポートする必要があります。

TLS を使用する場合、サーバーは SSL または TLS をサポートする必要があります。

`sasl/GSSAPI` を使用する場合、サーバーは SASL、GSSAPI、Kerberos 5 認証をサポートする必要があります。ネットワーク上の GSS 暗号化のサポートは、オプションです。

LDAP ネームサービスで使用されるデフォルトフィルタ

SSD を使用して個々のサービスにパラメータを手動で指定しないと、デフォルトフィルタが使用されます。特定のサービスのデフォルトフィルタを表示するには、`-v` オプションを指定して `ldaplist` を実行してください。

次の例では、`filter=(&(objectclass=iphost)(cn=abcde)` によってデフォルトフィルタが定義されます。

```
database=hosts
filter=(&(objectclass=iphost)(cn=abcde)
user data=(&(%s) (cn=abcde))
```

`ldaplist` は、次に示す一連のデフォルトフィルタを生成します (`%s` は文字列を意味し、`%d` は数値を意味する)。

```
hosts
(&(objectclass=iphost)(cn=%s))
--------------
```

```
passwd
(&(objectclass=posixaccount)(uid=%s))
--------------
services
(&(objectclass=ipservice)(cn=%s))
--------------
group
(&(objectclass=posixgroup)(cn=%s))
--------------
netgroup
(&(objectclass=nisnetgroup)(cn=%s))
--------------
networks
(&(objectclass=ipnetwork)(ipnetworknumber=%s))
--------------
netmasks
(&(objectclass=ipnetwork)(ipnetworknumber=%s))
--------------
rpc
(&(objectclass=oncrpc)(cn=%s))
--------------
protocols
(&(objectclass=ipprotocol)(cn=%s))
--------------
bootparams
(&(objectclass=bootableDevice)(cn=%s))
--------------
ethers
(&(objectclass=ieee802Device)(cn=%s))
--------------
publickey
(&(objectclass=niskeyobject)(cn=%s))
or
(&(objectclass=niskeyobject)(uidnumber=%d))
--------------
aliases
(&(objectclass=mailGroup)(cn=%s))
--------------
```

表 14–4　getXbyY 呼び出しで使用される LDAP フィルタ

フィルタ	定義
bootparamByName	(&(objectClass=bootableDevice)(cn=%s))
etherByHost	(&(objectClass=ieee802Device)(cn=%s))
etherByEther	(&(objectClass=ieee802Device)(macAddress=%s))
groupByName	(&(objectClass=posixGroup)(cn=%s))
groupByGID	(&(objectClass=posixGroup)(gidNumber=%ld))
groupByMember	(&(objectClass=posixGroup)(memberUid=%s))
hostsByName	(&(objectClass=ipHost)(cn=%s))
hostsByAddr	(&(objectClass=ipHost)(ipHostNumber=%s))

表 14–4 getXbyY 呼び出しで使用される LDAP フィルタ (続き)

フィルタ	定義
keyByUID	(&(objectClass=nisKeyObject)(uidNumber=%s))
keyByHost	(&(objectClass=nisKeyObject)(cn=%s))
netByName	(&(objectClass=ipNetwork)(cn=%s))
netByAddr	(&(objectClass=ipNetwork)(ipNetworkNumber=%s))
nisgroupMember	(membernisnetgroup=%s)
maskByNet	(&(objectClass=ipNetwork)(ipNetworkNumber=%s))
printerByName	(& (objectClass=sunPrinter)(|(printer-name=%s)(printer-aliases=%s)))
projectByName	(&(objectClass=SolarisProject)(SolarisProjectName=%s))
projectByID	(&(objectClass=SolarisProject)(SolarisProjectID=%ld))
protoByName	(&(objectClass=ipProtocol)(cn=%s))
protoByNumber	(&(objectClass=ipProtocol)(ipProtocolNumber=%d))
passwordByName	(&(objectClass=posixAccount)(uid=%s))
passwordByNumber	(&(objectClass=posixAccount)(uidNumber=%ld))
rpcByName	(&(objectClass=oncRpc)(cn=%s))
rpcByNumber	(&(objectClass=oncRpc)(oncRpcNumber=%d))
serverByName	(&(objectClass=ipService)(cn=%s))
serverByPort	(&(objectClass=ipService)(ipServicePort=%ld))
serverByNameAndProto	(&(objectClass=ipService)(cn=%s)(ipServiceProtocol=%s))
specialByNameserver	(ipServiceProtocol=%s))
ByPortAndProto	(&(objectClass=shadowAccount)(uid=%s))
netgroupByTriple	(&(objectClass=nisNetGroup)(cn=%s))
netgroupByMember	(&(objectClass=nisNetGroup)(cn=%s))
authName	(&(objectClass=SolarisAuthAttr)(cn=%s))
auditUserByName	(&(objectClass=SolarisAuditUser)(uid=%s))
execByName	(&(objectClass=SolarisExecAttr)(cn=%s) (SolarisKernelSecurityPolicy=%s)(SolarisProfileType=%s))

表 14–4 getXbyY 呼び出しで使用される LDAP フィルタ (続き)

フィルタ	定義
execByPolicy	(&(objectClass=SolarisExecAttr)(SolarisProfileId=%s) (SolarisKernelSecurityPolicy=%s)(SolarisProfileType=%s))
profileByName	(&(objectClass=SolarisProfAttr)(cn=%s))
userByName	(&(objectClass=SolarisUserAttr)(uid=%s))

次の表に getent 属性フィルタの一覧を示します。

表 14–5 getent 属性フィルタ

フィルタ	定義
aliases	(objectClass=rfc822MailGroup)
auth_attr	(objectClass=SolarisAuthAttr)
audit_user	(objectClass=SolarisAuditUser)
exec_attr	(objectClass=SolarisExecAttr)
group	(objectClass=posixGroup)
hosts	(objectClass=ipHost)
networks	(objectClass=ipNetwork)
prof_attr	(objectClass=SolarisProfAttr)
protocols	(objectClass=ipProtocol)
passwd	(objectClass=posixAccount)
printers	(objectClass=sunPrinter)
rpc	(objectClass=oncRpc)
services	(objectClass=ipService)
shadow	(objectclass=shadowAccount)
project	(objectClass=SolarisProject)
usr_attr	(objectClass=SolarisUserAttr)

◆ ◆ ◆ 第 15 章

15 NIS から LDAP への移行 (概要と手順)

この章では、LDAP ディレクトリに格納されたネーム情報を使用する NIS クライアントの、サポートを有効にする方法について説明します。この章の手順に従うことで、NIS ネームサービスから LDAP ネームサービスへ移行できます。

LDAP への移行の利点を判定するには、136 ページの「LDAP ネームサービスとその他のネームサービスの比較」を参照してください。

この章の内容は、次のとおりです。

NIS から LDAP への移行サービス (概要)

NIS から LDAP への移行サービス (「N2L サービス」) は、NIS マスターサーバー上の既存の NIS デーモンを、NIS から LDAP への移行用デーモンに置き換えます。また、N2L サービスは、NIS から LDAP への移行用マッピングファイルを、その NIS マスターサーバー上に作成します。マッピングファイルでは、NIS マップエントリと、LDAP での同等なディレクトリ情報ツリー (DIT) との間のマッピングを指定します。この移行を完了した NIS マスターサーバーは、「N2L サーバー」と呼ばれます。スレーブサーバーには、`NISLDAPmapping` ファイルはありません。したがって、引き続きそのまま動作します。スレーブサーバーのデータは、N2L サーバーから、通常の NIS マスターからと同様に、定期的に更新されます。

N2L サービスの動作は、構成ファイル ypserv および NISLDAPmapping によって制御されます。スクリプト inityp2l は、これらの構成ファイルの作成を支援します。いったん N2L サーバーが確立されたあとは、構成ファイルを直接編集して N2L を管理できます。

N2L サービスは、次の操作をサポートします。

- LDAP ディレクトリ情報ツリー (DIT) 内に NIS マップをインポートする
- NIS の速度および拡張性を維持しつつ、クライアントから DIT 情報にアクセスする

あらゆるネームシステムで、1 つのソースの情報だけが正規のソースになります。従来の NIS では、正規の情報は NIS ソースです。N2L サービスを使用する場合、LDAP ディレクトリが正規のデータソースになります。ディレクトリは、第 9 章「LDAP 基本コンポーネントおよび概念 (概要)」で説明するディレクトリ管理ツールを使用して管理されます。

NIS ソースは、緊急時のバックアップまたはバックアウト (LDAP に移行するのではなく、NIS の使用をやめる) にのみ使用します。N2L サービスを使い始めてから、NIS クライアントを徐々に減らすことができます。最終的に、すべての NIS クライアントを Solaris LDAP ネームサービスクライアントに置き換えることができます。

以降の節では、さらに概要情報を説明します。

- 253 ページの「この章の対象読者」
- 253 ページの「NIS から LDAP への移行サービスを使用しない場合」
- 253 ページの「NIS から LDAP への移行サービスがユーザーに与える影響」
- 254 ページの「NIS から LDAP への移行で使用される用語」
- 255 ページの「NIS から LDAP への移行コマンド、ファイル、およびマップ」
- 256 ページの「サポートされる標準マッピング」

NIS から LDAP への移行用ツールとサービス管理機能

NIS と LDAP のサービスはサービス管理機能によって管理されます。これらのサービスに関する有効化、無効化、再起動などの管理アクションは、svcadm コマンドを使用して実行できます。svcs コマンドを使用してサービスの状態を照会できます。LDAP および NIS での SMF の使用の詳細については、200 ページの「LDAP とサービス管理機能」および86 ページの「NIS とサービス管理機能」を参照してください。SMF の概要については、『Solaris のシステム管理 (基本編)』の第 18 章「サービスの管理 (概要)」を参照してください。また、詳細については、svcadm(1M) および svcs(1) のマニュアルページを参照してください。

この章の対象読者

この章の手順を実行するには、NIS および LDAP の概念、用語、および ID を理解する必要があります。NIS および LDAP のネームサービスについての詳細は、このマニュアルの以降の節を参照してください。

- NIS の概要については、第 4 章「ネットワーク情報サービス (NIS) (概要)」。
- LDAP の概要については、第 8 章「LDAP ネームサービスの紹介 (概要/リファレンス)」。

NIS から LDAP への移行サービスを使用しない場合

次の状況では、N2L サービスを使用しないでください。

- NIS と LDAP ネームサービスクライアント間でデータを共有する予定がない環境。

  このような環境では、N2L サーバーは、過度に複雑な NIS マスターサーバーとして機能します。

- NIS ソースファイルを変更するツール (`yppasswd` 以外のツール) で NIS マップを管理している環境。

  DIT マップから NIS ソースを再生成する処理は、必ずしも正確ではないため、生成されたマップを手動で確認する必要があります。いったん N2L サービスを使用し始めたあとは、NIS ソースの再生成は NIS をバックアウトするため、または NIS に戻すためにだけ提供されます。

- NIS クライアントのない環境。

  このような環境では、Solaris LDAP ネームサービスクライアントおよびそれに対応するツールを使用してください。

NIS から LDAP への移行サービスがユーザーに与える影響

N2L サービスに関連するファイルをインストールするだけでは、NIS サーバーのデフォルトの動作は変更されません。インストール時に、サーバー上の NIS のマニュアルページの一部が変更され、N2L のヘルパースクリプト `inityp2l` および `ypmap2src` が追加されます。しかし、NIS サーバー上で `inityp2l` を実行したり、N2L 構成ファイルを手動で作成したりしないと、NIS コンポーネントは従来の NIS モードで起動し、通常通りに機能します。

`inityp2l` の実行後に、サーバーとクライアントの動作が少し変更されます。次の表に、NIS および LDAP のユーザータイプと、N2L サービスの配備後に各タイプのユーザーが注意しなければならない部分の説明を示します。

ユーザータイプ	N2L サービスの影響
NIS マスターサーバー管理者	NIS マスターサーバーは、N2L サーバーに変換される。構成ファイル NISLDAPmapping および ypserv が、N2L サーバー上にインストールされる。N2L サーバーの確立後は、LDAP コマンドを使用してネーム情報を管理できる
NIS スレーブサーバー管理者	N2L の変換後も、NIS スレーブサーバーは NIS を通常の方法で動作する。ypmake によって yppush が呼び出されると、N2L サーバーは、更新された NIS マップをスレーブサーバーに転送する。ypmake(1M) のマニュアルページを参照
NIS クライアント	NIS の読み取り動作は、従来の NIS と同様。Solaris LDAP ネームサービスクライアントが DIT 内の情報を変更すると、情報が NIS マップ内にコピーされる。コピー操作は、変更可能なタイムアウトの期限が切れると完了する。このような動作は、クライアントが NIS スレーブサーバーに接続された場合の通常の NIS クライアントの動作と同じ
	N2L サーバーが読み取りのために LDAP サーバーにバインドできない場合、N2L サーバーはローカルにキャッシュされたコピーから情報を返す。また、N2L サーバーは内部サーバーエラーを返す場合もある。N2L サーバーの構成によって、どちらの方法で応答することも可能。詳細については、ypserv(1M) のマニュアルページを参照
すべてのユーザー	NIS クライアントがパスワードの変更を要求すると、N2L マスターサーバーとネイティブの LDAP クライアントに変更がただちに反映される
	NIS クライアントでのパスワードの変更を試みて、LDAP サーバーが利用できない場合は、変更は拒絶され N2L サーバーは内部サーバーエラーを返す。この動作によって、キャッシュに誤った情報が書き込まれることを防止する

NIS から LDAP への移行で使用される用語

N2L サービスの実装に関連する用語を次に示します。

表 15–1 N2L の移行の関連用語

用語	説明
N2L 構成ファイル	/var/yp/NISLDAPmapping および /var/yp/ypserv ファイル。ypserv デーモンが N2L モードでマスターサーバーを起動するために使用する。詳細は、NISLDAPmapping(4) および ypserv(4) のマニュアルページを参照
マップ	N2L サービスでは、「マップ」は、次の 2 とおりの意味で使用される。 ■ NIS が特定の種類の情報を格納するデータベースファイル ■ LDAP DIT との間の NIS 情報のマッピングプロセス
マッピング	LDAP DIT エントリとの間の NIS エントリの変換プロセス
マッピングファイル	NISLDAPmapping ファイル。NIS と LDAP のファイル間のエントリのマッピング方法を確立する

表 15–1 N2L の移行の関連用語 (続き)

用語	説明
標準マップ	通常使用される NIS マップ。マッピングファイルへの手動修正が不要で、N2L サービスによってサポートされる。サポートされる標準マップのリストは、256 ページの「サポートされる標準マッピング」を参照
非標準マップ	標準の NIS マップであるが、RFC 2307 やその後継で指定されたマッピング以外の、NIS と LDAP DIT 間のマッピングを使用するようにカスタマイズされたマップ
カスタムマップ	標準のマップではないマップ。したがって、NIS から LDAP への移行時にはマッピングファイルの手動修正が必要
LDAP クライアント	従来の LDAP クライアント。LDAP サーバーとの間で読み書きを行う。従来の LDAP クライアントは、任意の LDAP サーバーに対して読み取りおよび書き込みを行うシステム。Solaris LDAP ネームサービスクライアントは、カスタマイズされたネーム情報のサブセットを処理する
LDAP ネームサービスクライアント	Solaris LDAP クライアント。カスタマイズされたネーム情報のサブセットを処理する
N2L サーバー	N2L サービスを使用して、N2L サーバーとして再構成された NIS マスターサーバー。再構成には、NIS デーモンの置き換えと新しい構成ファイルの追加が含まれる。

NIS から LDAP への移行コマンド、ファイル、およびマップ

N2L の移行に関連して 2 つのユーティリティー、2 つの構成ファイル、および 1 つのマッピングがあります。

表 15–2 N2L のコマンド、ファイル、およびマップの説明

コマンド/ファイル/マップ	説明
`/usr/lib/netsvc/yp/inityp2l`	`NISLDAPmapping` および `ypserv` の構成ファイルの作成を支援するユーティリティー。このユーティリティーは、これらのファイルを管理するための汎用ツールではない。熟練したユーザーであれば、`inityp2l` の出力をテキストエディタを使って検証したりカスタマイズしたりすることで、N2L 構成ファイルの管理や、カスタムマッピングの作成を行うことも可能。`inityp2l(1M)` のマニュアルページを参照
`/usr/lib/netsvc/yp/ypmap2src`	標準の NIS マップを同等な NIS ソースファイルに近似したファイルに変換するユーティリティー。`ypmap2src` の主要な用途は、N2L の移行サーバーから従来の NIS への変換。`ypmap2src(1M)` のマニュアルページを参照
`/var/yp/NISLDAPmapping`	NIS マップエントリと、これと同等な LDAP でのディレクトリ情報ツリー (DIT) エントリとの間のマッピングを指定する構成ファイル。`NISLDAPmapping(4)` のマニュアルページを参照
`/var/yp/ypserv`	NIS から LDAP への移行用デーモンの構成情報を指定するファイル。`ypserv(4)` のマニュアルページを参照

表 15–2 N2L のコマンド、ファイル、およびマップの説明 (続き)

コマンド/ファイル/マップ	説明
ageing.byname	NIS から LDAP への移行の実行時に、DIT でのパスワード有効期限情報の読み取りおよび書き込みのために yppasswdd によって使用されるマッピング

サポートされる標準マッピング

デフォルトでは、N2L サービスは次のリストのマップと RFC 2307 (またはその後継) LDAP エントリとの間のマッピングをサポートします。これらの標準マップでは、マッピングファイルへの手動修正は不要です。システム上で次のリストにないマップは、カスタムマップと見なされ、マッピングファイルの手動修正が必要です。

また、N2L サービスは、auto.* マップの自動マッピングもサポートします。ただし、ほとんどの auto.* ファイル名とそのコンテンツは、各ネットワーク構成に固有なので、このリストではこれらのファイルは指定していません。例外は、auto.home マップと auto.master マップです。これらは標準マップとしてサポートされます。

```
audit_user
auth_attr
auto.home
auto.master
bootparams
ethers.byaddr ethers.byname
exec_attr
group.bygid group.byname group.adjunct.byname
hosts.byaddr hosts.byname
ipnodes.byaddr ipnodes.byname
mail.byaddr mail.aliases
netgroup netgroup.byprojid netgroup.byuser netgroup.byhost
netid.byname
netmasks.byaddr
networks.byaddr networks.byname
passwd.byname passwd.byuid passwd.adjunct.byname
printers.conf.byname
prof_attr
project.byname project.byprojectid
protocols.byname protocols.bynumber
publickey.byname
rpc.bynumber
services.byname services.byservicename
timezone.byname
user_attr
```

NIS から LDAP への移行時に、yppasswdd デーモンは、N2L 固有のマップ ageing.byname を使用して、 DIT でのパスワード有効期限情報の読み取りと書き込みを行います。パスワード有効期限を使用していない場合は、ageing.byname マッピングは無視されます。

NIS から LDAP への移行 (作業マップ)

次の表に、標準およびカスタムの NIS から LDAP への変換マッピングによって、N2L サービスをインストールし管理するために必要な手順を示します。

作業	説明	参照先
すべての前提条件の完了	NIS サーバーと Sun Java System Directory Server (LDAP サーバー) を正しく構成すること	257 ページの「NIS から LDAP への移行のための前提条件」
N2L サービスの設定	NIS マスターサーバーで、`inityp2l` を実行して、次のいずれかのマッピングを設定する	
	標準マッピング	259 ページの「標準マッピングを使用して N2L サービスを設定する方法」
	カスタムまたは非標準マッピング	261 ページの「カスタムマッピングまたは非標準マッピングを使用して N2L サービスを設定する方法」
マップのカスタマイズ	N2L の移行のためのカスタムマップの作成方法の例を参照する	263 ページの「カスタムマップの例」
N2L のための Sun Java System Directory Server の構成	N2L 移行のための、LDAP サーバーとして Sun Java System Directory Server を構成し調整する	265 ページの「Sun Java System Directory Server を使用した NIS から LDAP への移行の最良の実践原則」
システムのトラブルシューティング	一般的な N2L の問題を特定し解決する	268 ページの「NIS から LDAP への移行のトラブルシューティング」
NIS に戻す方法	次のいずれか適切なマップを使用して NIS に戻す	
	以前の NIS ソースファイルに基づくマップ	273 ページの「以前のソースファイルに基づくマップに戻す方法」
	現在の DIT に基づくマップ	274 ページの「現在の DIT 内容に基づくマップに戻す方法」

NIS から LDAP への移行のための前提条件

N2L サービスを実装する前に次の項目をチェックし、完了する必要があります。

- `inityp2l` スクリプトを実行して N2L モードを有効にする前に、システムが従来の NIS サーバーで動作するように設定されていること
- システムで LDAP ディレクトリサーバーを構成していること

NIS から LDAP への移行ツールでは、Sun Java System Directory Server (以前の名称は Sun ONE Directory Server) と、Sun から提供されるその互換バージョンのディレクトリサーバーがサポートされています。Sun Java System Directory Server を使用している場合は、N2L サービスを設定する前に、`idsconfig` コマンドを使用してサーバーを構成してください。`idsconfig` についての詳細は、第 11 章「LDAP クライアントと Sun Java System Directory Server の設定 (手順)」と `idsconfig(1M)` のマニュアルページを参照してください。

その他の (他社製の) の LDAP サーバーが、N2L サービスで動作する場合もありますが、Sun はそれらをサポートしていません。Sun Java System Directory Server または Sun 互換サーバー以外の LDAP サーバーを使用している場合は、N2L サービスを設定する前に、RFC 2307 (またはその後継) スキーマをサポートするようにサーバーを手動で構成してください。

- `nsswitch.conf` ファイルで少なくとも `hosts` エントリおよび `ipnodes` エントリに対して、検索順序として `nis` の前に `files` がリストされていることを確認すること
- N2L マスターサーバーと LDAP サーバーのアドレスが、N2L マスターサーバー上の `hosts` ファイルまたは `ipnodes` ファイルに存在することを確認すること。ローカルホスト名を解決するためのシステムの構成方法に応じて、サーバーアドレスは `hosts` ファイル、`ipnodes` ファイル、またはその両方にリストされる必要があります。

  代わりに、`ypserv` で、ホスト名ではなく LDAP サーバーのアドレスをリストする方法もあります。このことは、LDAP サーバーのアドレスが別の場所にもリストされていることを意味しています。したがって、LDAP サーバーと N2L マスターサーバーのどちらかのアドレスを変更するには、別のファイルの修正も必要です。

NIS から LDAP への移行サービスの設定

次の 2 つの手順に示すように、標準のマッピングとカスタムマッピングのどちらかを使用して、N2L サービスを設定できます。

NIS から LDAP への変換作業の一部として、`inityp2l` コマンドを実行する必要があります。このコマンドは、対話型で、構成情報を入力するスクリプトを実行します。次のリストに、構成に必要な情報の種類を示します。これらの属性の詳細については、`ypserv(1M)` のマニュアルページを参照してください。

- 作成する構成ファイルの名前 (デフォルト = `/etc/default/ypserv`)
- LDAP の構成情報を格納する DN (デフォルト = `ypserv`)
- LDAP との間でデータをマッピングするための優先サーバーリスト
- LDAP との間でデータをマッピングするための認証方式
- LDAP との間でデータをマッピングするための TLS (Transport Layer Security)
- LDAP との間でデータを読み書きするためのプロキシのユーザーバインド DN

- LDAP との間でデータを読み書きするためのプロキシのユーザーパスワード
- LDAP バインド動作のタイムアウト値 (秒単位)
- LDAP 検索動作のタイムアウト値 (秒単位)
- LDAP 変更動作のタイムアウト値 (秒単位)
- LDAP 追加動作のタイムアウト値 (秒単位)
- LDAP 削除動作のタイムアウト値 (秒単位)
- LDAP サーバーでの検索動作の制限時間 (秒単位)
- LDAP サーバーでの検索動作の制限サイズ (バイト単位)
- N2L が LDAP 照会に従うかどうか
- LDAP 検索のエラー処理、検索試行回数、および各試行間のタイムアウト (秒単位)
- 格納のエラー処理、検索試行回数、および各試行間のタイムアウト (秒単位)
- マッピングファイル名
- `auto_direct` マップのマッピング情報を生成するかどうか

  スクリプトは、マッピングファイル内の適切な位置にカスタムマップについての情報を配置します。
- ネーミングコンテキスト
- パスワードの変更を有効にするかどうか
- 任意のマップのデフォルトの TTL 値を変更するかどうか

注 – `sasl/cram-md5` 認証は、Sun Java System Directory Server を含むほとんどの LDAP サーバーではサポートされません。

▼ 標準マッピングを使用して N2L サービスを設定する方法

256 ページの「サポートされる標準マッピング」にリストされているマップを移行する場合は、この手順に従います。カスタムマップまたは非標準マップを使用している場合 は、261 ページの「カスタムマッピングまたは非標準マッピングを使用して N2L サービスを設定する方法」を参照してください。

LDAP サーバーの設定が終わったら、`inityp2l` スクリプトを実行して、プロンプトに従って構成情報を入力します。`inityp2l` は構成を行い、標準および `auto.*` マップのためのマッピングファイルを設定します。

1 **257 ページの「NIS から LDAP への移行のための前提条件」**にリストされた前提条件の手順を完了します。

2 **NIS マスターサーバーで、スーパーユーザーになるか、同等の役割になります。**

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

3 **NIS マスターサーバーを N2L サーバーに変換します。**

```
# inityp2l
```

NIS マスターサーバーで `inityp2l` スクリプトを実行して、プロンプトに従います。指定が必要な情報のリストは、258 ページの「NIS から LDAP への移行サービスの設定」を参照してください。

詳細については、`inityp2l(1M)` のマニュアルページを参照してください。

4 **LDAP ディレクトリ情報ツリー (DIT) が完全に初期化されているかどうかを判定します。**

`NISLDAPmapping` ファイルにリストされたすべてのマップの配備に必要な情報がすでに DIT 内に存在する場合、DIT は完全に初期化されています。

- 初期化されていない場合、手順 5 を続行して手順 6 をスキップします。
- 初期化されている場合、手順 5 をスキップして手順 6 に進みます。

5 **NIS ソースファイルから移行するため、DIT を初期化します。**

DIT が完全に初期化されていない場合に限って、次の手順を実行してください。

a. **以前の NIS マップが最新の状態になっていることを確認してください。**

```
# cd /var/yp
# make
```

詳細については、`ypmake(1M)` のマニュアルページを参照してください。

b. **NIS デーモンを停止します。**

```
# svcadm disable network/nis/server:default
```

c. **以前のマップを DIT にコピーしてから、マップ用の N2L サポートを初期化します。**

```
# ypserv -Ir
```

`ypserv` が終了するまで待ちます。

ヒント – 元の NIS `dbm` ファイルは上書きされません。必要に応じて、これらのファイルを回復できます。

---

d. **NIS** デーモンを起動し、デーモンが新しいマップを使用していることを確認します。

```
# svcadm enable network/nis/server:default
```

これで、標準マップでの N2L サービスの設定を完了します。手順 6 を行う必要はありません。

6 **NIS** マップを初期化します。

DIT が完全に初期化され、手順 5 をスキップした場合に限って、次の手順を実行してください。

a. **NIS** デーモンを停止します。

```
# svcadm disable network/nis/server:default
```

b. **DIT** 内の情報に従って **NIS** マップを初期化します。

```
# ypserv -r
```

`ypserv` が終了するまで待ちます。

ヒント – 元の NIS `dbm` ファイルは上書きされません。必要に応じて、これらのファイルを回復できます。

---

c. **NIS** デーモンを起動し、デーモンが新しいマップを使用していることを確認します。

```
# svcadm enable network/nis/server:default
```

▼ カスタムマップピングまたは非標準マップピングを使用して **N2L** サービスを設定する方法

次の状況に適合する場合、この手順を実行してください。

- 256 ページの「サポートされる標準マッピング」にリストされていないマップがある
- RFC 2307 とは異なるマッピングで LDAP にマップしたい標準の NIS マップがある

1 **257** ページの「**NIS** から **LDAP** への移行のための前提条件」にリストされた前提条件の手順を完了します。

2 **NIS** マスターサーバーで、スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

3 **NIS マスターサーバーを N2L サーバーに構成します。**

```
# inityp2l
```

NIS マスターサーバーで `inityp2l` スクリプトを実行して、プロンプトに従います。指定が必要な情報のリストは、258 ページの「NIS から LDAP への移行サービスの設定」を参照してください。

詳細については、`inityp2l(1M)` のマニュアルページを参照してください。

4 `/var/yp/NISLDAPmapping` ファイルを修正します。

マッピングファイルの修正方法の例は、263 ページの「カスタムマップの例」を参照してください。

5 **LDAP ディレクトリ情報ツリー (DIT) が完全に初期化されているかどうかを判定します。**

`NISLDAPmapping` ファイルにリストされたすべてのマップの配備に必要な情報がすでに DIT 内に存在する場合、DIT は完全に初期化されています。

- 初期化されていない場合、手順 6、手順 8、および手順 9 を完了します。
- 初期化されている場合、手順 6 をスキップして、手順 7、手順 8、および手順 9 を完了します。

6 **NIS ソースファイルから移行するため、DIT を初期化します。**

a. **以前の NIS マップが最新の状態になっていることを確認してください。**

```
# cd /var/yp
# make
```

詳細については、`ypmake(1M)` のマニュアルページを参照してください。

b. **NIS デーモンを停止します。**

```
# svcadm disable network/nis/server:default
```

c. **以前のマップを DIT にコピーしてから、マップ用の N2L サポートを初期化します。**

```
# ypserv -Ir
```

`ypserv` が終了するまで待ちます。

ヒント – 元の NIS `dbm` ファイルは上書きされません。必要に応じて、これらのファイルを回復できます。

---

d. **NIS デーモンを起動し、デーモンが新しいマップを使用していることを確認します。**

```
# svcadm enable network/nis/server:default
```

e. 手順 **7** をスキップして、手順 **8** から続行します。

7 **NIS** マップを初期化します。

DIT が完全に初期化されている場合に限って、この手順を実行します。

a. **NIS** デーモンを停止します。

```
# svcadm disable network/nis/server:default
```

b. **DIT** 内の情報に従って **NIS** マップを初期化します。

```
# ypserv -r
```

ypserv が終了するまで待ちます。

ヒント – 元の NIS dbm ファイルは上書きされません。必要に応じて、これらのファイルを回復できます。

---

c. **NIS** デーモンを起動し、デーモンが新しいマップを使用していることを確認します。

```
# svcadm enable network/nis/server:default
```

8 **LDAP** エントリが正しいことを確認します。

エントリが間違っている場合、LDAP ネームサービスクライアントからはそのエントリを見つけられません。

```
# ldapsearch -h server -s sub -b "ou=servdates, dc=..." \
"objectclass=servDates"
```

9 LDAP_ マップの内容を確認します。

次に、makedbm を使用して servdate.bynumber マップの内容を確認する方法の例を示します。

```
# makedbm -u LDAP_servdate.bynumber
plato: 1/3/2001
johnson: 2/4/2003,1/3/2001
yeats: 4/4/2002
poe: 3/3/2002,3/4/2000
```

出力結果が期待どおりの内容であれば、NIS から LDAP への移行は成功です。

元の NIS dbm ファイルは上書きされないことに注意してください。したがって、いつでもこれらのファイルは回復できます。詳細については、273 ページの「NIS に戻す方法」を参照してください。

カスタムマップの例

次の 2 つの例に、マップをカスタマイズする方法を示します。必要に応じて、任意のテキストエディタを使用して、/var/yp/NISLDAPmapping ファイルを修正しま

す。ファイルの属性と構文については、NISLDAPmapping(4) のマニュアルページと第 9 章「LDAP 基本コンポーネントおよび概念 (概要)」の LDAP ネームサービス情報を参照してください。

例 1-ホストエントリの移動

この例では、DIT でデフォルトの位置から別の (非標準の) 位置にホストエントリを移動する方法を示します。

NISLDAPmapping ファイルの nisLDAPobjectDN 属性を、新しいベース LDAP 識別名 (DN) に変更します。この例では、LDAP オブジェクトの内部構造は変更されません。したがって、objectClass エントリは変更されません。

変更前:

```
nisLDAPobjectDN hosts: \
                        ou=hosts,?one?, \
                        objectClass=device, \
                        objectClass=ipHost
```

変更後:

```
nisLDAPobjectDN hosts: \
                        ou=newHosts,?one?, \
                        objectClass=device, \
                        objectClass=ipHost
```

この変更によって、エントリは次のようにマッピングされます。

```
dn: ou=newHosts, dom=domain1, dc=sun, dc=com
```

元は、次のようでした。

```
dn: ou=hosts, dom=domain1, dc=sun, dc=com.
```

例 2-カスタムマップの実装

この例では、カスタムマップの実装方法を示します。

仮想のマップ「servdate.bynumber」には、システムのサービス日付についての情報が含まれます。このマップには、マシンのシリアル番号でインデックスが付けられます。この例では、123 です。各エントリは、マシンの所有者名、コロン、およびサービス日付のコンマ区切りのリストで構成されます。たとえば、John Smith:1/3/2001,4/5/2003 のようになります。

古いマップ構造は、次の形式の LDAP エントリにマップされます。

```
dn: number=123,ou=servdates,dc=... \
                number: 123 \
                userName: John Smith \
```

```
                date: 1/3/2001 \
                date: 4/5/2003 \
                 .
                 .
                 .
                objectClass: servDates
```

NISLDAPmapping ファイルを調べると、必要なパターンに最も近いマッピングが group であることがわかります。カスタムマッピングは group マッピングを参考にできます。マップは 1 つだけなので、nisLDAPdatabaseIdMapping 属性は不要です。NISLDAPmapping に追加される属性は、次のとおりです。

```
nisLDAPentryTtl servdate.bynumber:1800:5400:3600

nisLDAPnameFields servdate.bynumber: \
                        ("%s:%s", uname, dates)

nisLDAPobjectDN servdate.bynumber: \
                        ou=servdates, ?one? \
                        objectClass=servDates:

nisLDAPattributeFromField servdate.bynumber: \
                        dn=("number=%s,", rf_key), \
                        number=rf_key, \
                        userName=uname, \
                        (date)=(dates, ",")

nisLDAPfieldFromAttribute servdate.bynumber: \
                        rf_key=number, \
                        uname=userName, \
                        dates=("%s,", (date), ",")
```

Sun Java System Directory Server を使用した NIS から LDAP への移行の最良の実践原則

N2L サービスは、Sun Java System Directory Server (以前の名称は Sun ONE Directory Server) と、Sun の提供するその互換バージョンのディレクトリサーバーをサポートしています。その他の (他社製の) LDAP サーバーが、N2L サービスで動作する場合もありますが、Sun はそれらをサポートしていません。Sun Java System Directory Server または Sun の互換サーバー以外の LDAP サーバーを使用している場合は、RFC 2307 (またはその後継スキーマ) に準拠するように、サーバーを手動で構成してください。

Sun Java System Directory Server を使用すれば、ディレクトリサーバーを強化してパフォーマンスを改善できます。これらの強化を行うには、Sun Java System Directory Server 上に LDAP の管理者権限が必要です。また、ディレクトリサーバーのリブートが必要な場合もあります。リブートは、サーバーの LDAP クライアントとの間で調整が必要な作業です。Sun Java System Directory Server (および SunONE Directory Server、iPlanet Directory Server) のドキュメントは、Web サイト Sun Java System Directory Server Enterprise Edition 6.2 (http://docs.sun.com/coll/1224.3) で入手できます。

Sun Java System Directory Server を使用した仮想リスト表示インデックスの作成

大規模なマップでは、LDAP の仮想リスト表示 (VLV) インデックスを使用して、LDAP の検索から正しい結果が得られることを保証しなければなりません。Sun Java System Directory Server での VLV インデックスの設定についての詳細は、Sun Java System Directory Server Enterprise Edition 6.2 (`http://docs.sun.com/coll/1224.3`) のドキュメントを参照してください。

VLV の検索結果では、固定ページサイズ 50000 を使用します。Sun Java System Directory Server で VLV を使用する場合は、LDAP サーバーと N2L サーバーの両方でこのサイズの転送を処理できるようにしてください。すべてのマップがこの制限より小規模であることが明らかな場合は、VLV インデックスを使用する必要はありません。ただし、マップがこのサイズ制限より大きい場合、またはすべてのマップのサイズが明確な場合以外には、VLV インデックスを使用して、結果が不完全となることを防止しなければなりません。

VLV インデックスを使用している場合は、次のように適切なサイズ制限を設定します。

- Sun Java System Directory Server では、 `nsslapd-sizelimit` 属性を 50000 以上、または `-1` に設定する必要があります。`idsconfig(1M)` のマニュアルページを参照してください。
- N2L サーバーでは、 `nisLDAPsearchSizelimit` 属性を 50000 以上、または 0 に設定する必要があります。詳細については、`NISLDAPmapping(4)` のマニュアルページを参照してください。

VLV インデックスが作成されたら、Sun Java System Directory Server で `vlvindex` オプションを付けて `directoryserver` を実行してインデックスを有効にします。詳細については、`directoryserver(1M)` のマニュアルページを参照してください。

標準マップ用 VLV

次の状況に適合する場合、Sun Java System Directory Server の `idsconfig` コマンドを使用して、VLV を設定してください。

- Sun Java System Directory Server を使用している場合
- 標準マップを RFC 2307 LDAP エントリにマッピングしている場合

VLV はドメイン固有です。よって、`idsconfig` を実行するたびに、1 つの NIS ドメインに VLV が作成されます。したがって、NIS から LDAP への移行中に、`NISLDAPmapping` ファイルに含まれる各 `nisLDAPdomainContext` 属性に対して、`idsconfig` を 1 回実行しなければなりません。

カスタムマップおよび非標準マップ用 VLV

次の状況に適合する場合、マップ用に新しい Sun Java System Directory Server の VLV を手動で作成するか、既存の VLV インデックスをコピーして修正しなければなりません。

- Sun Java System Directory Server を使用している場合
- 大規模なカスタムマップがあるか、非標準の DIT 位置にマップされる標準のマップがある場合

既存の VLV インデックスを表示するには、次のように入力します。

```
# ldapsearch -h hostname -s sub -b "cn=ldbm database,cn=plugins,cn=config" \
"objectClass=vlvSearch"
```

Sun Java System Directory Server によるサーバーのタイムアウトの防止

N2L サーバーがマップをリフレッシュすると、その結果、大規模な LDAP ディレクトリアクセスが行われる場合があります。Sun Java System Directory Server が正しく構成されていない場合、リフレッシュ動作は完了前にタイムアウトになることがあります。ディレクトリサーバーのタイムアウトを防止するには、次の Sun Java System Directory Server の属性を手動で修正するか、`idsconfig` コマンドを実行します。

たとえば、サーバーでの検索リクエストの実行にかかる最小時間を秒単位で増やすには、次の属性を修正します。

```
dn: cn=config
nsslapd-timelimit: -1
```

テストのためには、属性値として `-1` を使用できます。この値は、制限がないことを示しています。最適な制限値が決まったら、属性値を変更します。稼働サーバーに、`-1` の属性値が設定されていてはなりません。制限がないと、サーバーがサービス妨害攻撃に無防備になる場合があります。

LDAP での Sun Java System Directory Server の構成の詳細については、第 11 章「LDAP クライアントと Sun Java System Directory Server の設定 (手順)」を参照してください。

Sun Java System Directory Server 使用時のバッファーオーバーランの防止

バッファーオーバーランを防止するには、Sun Java System Directory Server の属性を手動で修正するか、`idsconfig` コマンドを実行します。

1. たとえば、クライアント検索照会に返されるエントリの最大数を増やすには、次の属性を修正します。

```
dn: cn=config
nsslapd-sizelimit: -1
```

2. クライアント検索照会で確認されるエントリの最大数を増やすには、次の属性を修正します。

```
dn: cn=config, cn=ldbm database, cn=plugins, cn=config
nsslapd-lookthroughlimit: -1
```

テストのためには、属性値として -1 を使用できます。この値は、制限がないことを示しています。最適な制限値が決まったら、属性値を変更します。稼働サーバーに、-1 の属性値が設定されていてはなりません。制限がないと、サーバーがサービス妨害攻撃に無防備になる場合があります。

VLV を使用している場合は、sizelimit 属性値を266 ページの「Sun Java System Directory Server を使用した仮想リスト表示インデックスの作成」での定義に合わせて設定する必要があります。VLV を使用していない場合、最も大きなコンテナを格納できるようにサイズ制限を設定する必要があります。

LDAP での Sun Java System Directory Server の構成の詳細については、第 11 章「LDAP クライアントと Sun Java System Directory Server の設定 (手順)」を参照してください。

NIS から LDAP への移行に関する制限

N2L サーバーの設定が完了すると、以降 NIS ソースファイルは使用されません。したがって、N2L サーバーで ypmake を実行しないでください。既存の cron ジョブなどで誤って ypmake を実行した場合、N2L サービスは影響を受けません。ただし、yppush を明示的に呼び出すことを推奨する警告がログに記録されます。

NIS から LDAP への移行のトラブルシューティング

この節では、トラブルシューティングの 2 つの領域を説明します。

よくある LDAP エラーメッセージ

N2L サーバーが LDAP 内部の問題に関連するエラーをログに記録して、LDAP 関連のエラーメッセージが表示される場合があります。エラーは致命的なものではありませんが、調査すべき問題を示しています。たとえば、N2L サーバーは動作を継続していても、返される結果が古かったり、不完全になる場合があります。

次のリストに、N2L サービスを実装するときに発生する可能性のある、よくある LDAP エラーメッセージをいくつか示します。エラーの説明、考えられる原因、およびエラーの対策も含みます。

`Administrative limit exceeded`

エラー番号: 11

原因: ディレクトリサーバーの `nsslapd-sizelimit` 属性で許可されたものより大きな LDAP 検索が実行されました。情報の一部だけが返されます。

対策: `nsslapd-sizelimit` 属性の値を増やすか、失敗した検索に VLV インデックスを実装します。

`Invalid DN Syntax`

エラー番号: 34

原因: 不正な文字を含む DN を使用して LDAP エントリの書き込みが試みられました。N2L サーバーは、DN 内で生成される + 記号などの不正な文字のエスケープを試みます。

対策: LDAP サーバーのエラーログをチェックして、どのような不正な DN が書き込まれたかを調べます。それから、不正な DN を生成した `NISLDAPmapping` ファイルを修正します。

`Object class violation`

エラー番号: 65

原因: 無効な LDAP エントリの書き込みが試みられました。通常、次のいずれかの状況で生じる `MUST` 属性が見つからないことがこのエラーの原因です。

- 見つからない属性のエントリを作成する `NISLDAPmapping` ファイルのバグ
- 存在しないオブジェクトへの `AUXILIARY` 属性の追加の試み

  たとえば、ユーザー名がまだ `passwd.byxxx` マップから作成されていない場合、そのユーザーに対する補足情報の追加の試みは失敗します。

対策: `NISLDAPmapping` ファイルのバグである場合は、サーバーエラーログへの書き込みをチェックして、問題の原因を判断します。

`Can’t contact LDAP server`

エラー番号: 81

原因: `ypserv` ファイルが正しく構成されず、間違った LDAP ディレクトリサーバーを指定していることがあります。または、ディレクトリサーバーが稼働していません。

対策:

- `ypserv` ファイルを再構成して、正しい LDAP ディレクトリサーバーを指定します。
- LDAP サーバーが実行中であることを確認するには、ディレクトリサーバーでスーパーユーザーになるか、同等の役割になり、次のように入力します。

  ```
  # pgrep -l slapd
  ```

`Timeout`

エラー番号: 85

原因: LDAP 動作がタイムアウトしました。多くの場合、DIT からマップを更新している間に発生します。古い情報がマップに含まれている可能性があります。

対策: `ypserv` 構成ファイルの各 `nisLDAPxxxTimeout` 属性を増やします (xxx の部分は何種類かある)。

NIS から LDAP への移行に関する問題

N2L サーバーの実行中に、次の問題が発生する場合があります。考えられる原因と対策を説明します。

`NISLDAPmapping` ファイルのデバッグ

マッピングファイル `NISLDAPmapping` は複雑なファイルです。多くの潜在的なエラーによって、マッピングが予期しない動作をする場合があります。次の方法を用いて、この問題を解決してください。

`ypserv -ir` (または `-Ir`) を実行したときのコンソールメッセージの表示

問題: コンソールに簡単なメッセージが表示され、サーバーが終了します (詳細な説明は、`syslog` に書き込まれます)。

原因: マッピングファイルの構文が間違っている場合があります。

対策: `NISLDAPmapping` ファイルの構文をチェックして修正します。

起動時に **NIS** デーモンが終了する

問題: `ypserv` またはその他の NIS デーモンを実行すると、LDAP 関連のエラーメッセージがログに記録され、デーモンが終了します。

原因: 原因は次のいずれかが考えられます。

- LDAP サーバーと通信できない
- NIS マップまたは DIT 内のエントリが、指定されたマッピングと互換性がない
- LDAP サーバーへの読み書きの試みがエラーを返す

対策: LDAP サーバーのエラーログを調査します。268 ページの「よくある LDAP エラーメッセージ」にリストされた LDAP エラーを参照してください。

NIS 動作からの予期しない結果

問題: NIS 動作が予期した結果を返さず、エラーはログに記録されません。

原因: 間違ったエントリが LDAP または NIS マップに存在する場合があります。この結果、マッピングが期待したとおりに完了しません。

対策: LDAP DIT と NIS マップの N2L バージョンのエントリをチェックして、修正します。

1. LDAP DIT に正しいエントリが存在するかをチェックしてから、必要に応じてエントリを修正します。

   Sun Java System Directory Server を使用している場合は、`directoryserver startconsole` を実行して管理コンソールを起動します。

2. 新しく生成されたマップと元のマップを比較して、`/var/yp` ディレクトリの N2L バージョンの NIS マップに期待どおりのエントリが含まれていることをチェックします。必要に応じてエントリを修正します。

   ```
   # cd /var/yp/domainname
   # makedbm -u test.byname
   # makedbm -u LDAP_test.byname
   ```

   マップの出力をチェックする場合は、次のことに注意してください。

   - 両方のファイルでのエントリの順序が異なる可能性

     出力を比較する前に、`sort` コマンドを使用します。

   - 両方のファイルでの空白の使い方が異なる可能性

     出力を比較するときに、`diff -b` コマンドを使用します。

NIS マップの処理順序

問題 : オブジェクトクラス違反が発生しています。

原因 : `ypserv -i` コマンドを実行すると、各 NIS マップが読み取られ、その内容が DIT に書き込まれます。複数のマップが、同一の DIT オブジェクトに属性を提供する場合もあります。通常、オブジェクトは、1 つのマップによってそのオブジェクトの `MUST` 属性のすべてを含む大部分を生成されます。ほかのマップは、ほかの `MAY` 属性を提供します。

マップは、`NISLDAPmapping` ファイルに定義されている `nisLDAPobjectDN` 属性と同じ順序で処理されます。`MAY` 属性を含むマップが `MUST` 属性を含むマップより先に処理されると、オブジェクトクラス違反が発生します。このエラーについての詳細は、268 ページの「よくある LDAP エラーメッセージ」のエラー 65 を参照してください。

対策 : マップが正しい順序で処理されるように、`nisLDAPobjectDN` 属性の順序を変更します。

一時的に問題を回避するには、`ypserv -i` コマンドを複数回実行します。コマンドを実行するたびに、より多くの LDAP エントリが作られます。

注 – 1 つのマップからオブジェクトのすべての `MUST` 属性を作成できないマッピングはサポートされていません。

N2L サーバーのタイムアウトの問題

問題 : サーバーがタイムアウトします。

原因 : N2L サーバーがマップをリフレッシュすると、その結果、大規模な LDAP ディレクトリアクセスが行われる場合があります。Sun Java System Directory Server が正しく構成されていない場合、この動作は完了前にタイムアウトになることがあります。

対策: ディレクトリサーバーのタイムアウトを防止するには、Sun Java System Directory Server の属性を手動で修正するか、`idsconfig` コマンドを実行します。詳細は、268 ページの「よくある LDAP エラーメッセージ」と265 ページの「Sun Java System Directory Server を使用した NIS から LDAP への移行の最良の実践原則」を参照してください。

N2L のロックファイルの問題

問題 : `ypserv` コマンドは起動しますが、NIS リクエストに応答しません。

原因 : N2L サーバーのロックファイルが、NIS マップへのアクセスと正しく同期していません。このような状況が発生してはなりません。

対策: N2L サーバーで次のコマンドを入力します。

```
# svcadm disable network/nis/server:default
# rm /var/run/yp_maplock /var/run/yp_mapupdate
# svcadm enable network/nis/server:default
```

N2L のデッドロックの問題

問題 : N2L サーバーがデッドロックします。

原因 : N2L マスターサーバーのアドレスと LDAP サーバーのアドレスが `hosts`、`ipnodes`、または `ypserv` ファイルに正しくリストされていない場合、デッドロックの発生することがあります。N2L の正しいアドレス構成についての詳細は、257 ページの「NIS から LDAP への移行のための前提条件」を参照してください。

デッドロックの発生する例として、次の一連の事柄を考えてみてください。

1. NIS クライアントが IP アドレスの検索を試みます。
2. N2L サーバーが、`hosts` エントリは最新ではないことを検出します。
3. N2L サーバーが LDAP からの `hosts` エントリの更新を試みます。
4. N2L サーバーは、`ypserv` から LDAP サーバーの名前を取得してから、`libldap` を使用して検索を実行します。
5. `libldap` は、ネームサービススイッチを呼び出して、LDAP サーバー名の IP アドレスへの変換を試みます。
6. ネームサービススイッチの設定に基づき、N2L サーバーへの NIS 呼び出しを行い、デッドロックが発生します。

対策 : N2L マスターサーバー上の `hosts` ファイルまたは `ipnodes` ファイルに N2L マスターサーバーと LDAP サーバーのアドレスをリストします。`hosts` ファイルおよび `ipnodes` ファイルがローカルホスト名を解決するためにどのようにして構成

されているかに応じて、サーバーアドレスは、各ファイルに、またはその両方にリストされなければなりません。また、nsswitch.conf ファイルの hosts および ipnodes エントリで、検索順序として nis の前に files をリストしていることをチェックしてください。

別の方法として、このデッドロックに対して、ypserv ファイルでホスト名ではなく LDAP サーバーアドレスを記述する方法もあります。これは、LDAP サーバーアドレスが別の場所に記述されていることを意味しています。したがって、LDAP サーバーと N2L サーバーのどちらかでアドレスを変更する場合には、さらに少し作業が必要になります。

NIS に戻す方法

N2L サービスを使用して NIS から LDAP に移行したサイトでは、すべての NIS クライアントを Solaris LDAP ネームサービスクライアントに徐々に置き換えていくことが求められます。最終的には、NIS クライアントに対するサポートは不要になります。ただし、必要に応じて、N2L サービスは、次の 2 つの手順に示すように、従来の NIS に復帰するための 2 種類の方法を提供します。

ヒント – 従来の NIS は、N2L バージョンの NIS マップが存在しても、それを無視します。NIS に戻したあとで、サーバー上の N2L バージョンのマップをそのままにしておいた場合でも問題を起こしません。したがって、あとで再度 N2L を有効にしたい場合に備えて、N2L マップを保管しておくことができます。ただし、マップの保管はディスクスペースを消費します。

▼ 以前のソースファイルに基づくマップに戻す方法

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2 NIS デーモンを停止します。

```
# svcadm disable network/nis/server:default
```

3 N2L を無効にします。

このコマンドは、N2L マッピングファイルをバックアップして、移動します。

```
# mv /var/yp/NISLDAPmapping backup_filename
```

4 NOPUSH 環境変数を設定して、ypmake によって新しいマップが転送されないようにします。

```
# NOPUSH=1
```

5 以前のソースに基づいて、NIS マップの新しいセットを作成します。

```
# cd /var/yp
# make
```

6 (オプション) N2L バージョンの NIS マップを削除します。

```
# rm /var/yp/domainname/LDAP_*
```

7 NIS デーモンを起動します。

```
# svcadm enable network/nis/server:default
```

▼ 現在の DIT 内容に基づくマップに戻す方法

この手順を実行する前に、従来の NIS ソースファイルをバックアップします。

1 スーパーユーザーになるか、同等の役割を引き受けます。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2 NIS デーモンを停止します。

```
# svcadm disable network/nis/server:default
```

3 DIT に基づいてマップを更新します。

```
# ypserv -r
```

ypserv が終了するまで待ちます。

4 N2L を無効にします。

このコマンドは、N2L マッピングファイルをバックアップして、移動します。

```
# mv /var/yp/NISLDAPmapping backup_filename
```

5 NIS ソースファイルを再生成します。

```
# ypmap2src
```

6 再生成された NIS ソースファイルの内容と構造が正しいことを手動でチェックしてください。

7 再生成された NIS ソースファイルを適切なディレクトリに移動します。

8 (オプション) **N2L** バージョンのマッピングファイルを削除します。

```
# rm /var/yp/domainname/LDAP_*
```

9 NIS デーモンを起動します。

```
# svcadm enable network/nis/server:default
```

◆ ◆ ◆ 第 16 章

NIS+ から LDAP への移行

この章では、NIS+ ネームサービスから LDAP ネームサービスへの移行方法について説明します。

NIS+ から LDAP への移行の概要

NIS+ サーバーデーモン `rpc.nisd` は、`/var/nis/data` ディレクトリにある独自フォーマットのファイルに NIS+ データを保存します。NIS+ データは、LDAP と同期化することができます。従来は、そのために外部エージェントが必要でした。しかし、新しい NIS+ デーモンでは、LDAP サーバーを NIS+ データのデータリポジトリとして使用できるようになりました。これにより、NIS+ および LDAP クライアントが同一のネームサービス情報を共有できるため、メインネームサービスを NIS+ から LDAP に移行する作業がより簡単になりました。

デフォルトの `rpc.nisd` デーモンは、従来と同様に機能し、`/var/nis/data` の NIS+ データベースにデータを格納します。システム管理者は、必要に応じて、NIS+ データベースの一部の権限を LDAP サーバーに譲渡し、NIS+ データのリポジトリとして使用することができます。この場合、`/var/nis/data` ファイルは `rpc.nisd` デーモンのキャッシュとして機能するため、LDAP 検索トラフィックが減少します。また、LDAP サーバーが一時的に使用できなくなった場合でも、`rpc.nisd` デーモンは動作を継続できます。NIS+ および LDAP は常に同期化されるだけでなく、NIS+ および LDAP 間でデータをアップロードまたはダウンロードすることができます。

LDAP に対するデータのマッピングは、構成ファイルの柔軟な構文を使用して行います。`client_info.org_dir` および `timezone.org_dir` 以外のすべての標準 NIS+ テーブルは、テンプレートマッピングファイル `/var/nis/NIS+LDAPmapping.template` で対応できます。ほとんどの NIS+ インストールのテーブルは、変更する必要がないか、わずかな変更で済みます(306 ページの「`client_info` および `timezone` テーブル (NIS+ から LDAP への移行)」 と `timezone.org_dir` については、`client_info and timezone Tables (NIS+ to LDAP)` を参照)。NIS+ データは、LDAP ディレクトリ情報ツリー (DIT) に配置されます。また、マッピングファイルでは、LDAP から入力された NIS+

データに対して生存期間 (TTL) を設定できます。多くの場合、NIS+ 列値および LDAP 属性値は 1 対 1 で対応づけられますが、マッピングファイルはより複雑な関係にも対応できます。

`/etc/default/rpc.nisd` ファイルは、LDAP サーバーと認証を選択するときに使用し、`rpc.nisd` の一般的な動作をいくつか制御します。`rpc.nisd(4)` を参照してください。マッピングの詳細は、`/var/nis/NIS+LDAPmapping` ファイル内で指定します。詳細については、`NIS+LDAPmapping(4)` のマニュアルページを参照してください。マッピングファイルの名前を変更するときは、`/lib/svc/method/nisplus` ファイルを編集します。詳細については、279 ページの「NIS+ から LDAP への移行用ツールとサービス管理機能」を参照してください。

この章では、次の用語を使用します。

- コンテナ

  すべての関連エントリが格納される LDAP DIT 内の場所。たとえば、ユーザーアカウント情報は、多くの場合、`ou=People` コンテナに格納される。また、ホストアドレス情報は、 `ou=Hosts` コンテナに格納される
- ネット名

  認証可能な SecureRPC 内のエンティティー (ユーザーまたはマシン)
- マッピング

  NIS+ オブジェクトと LDAP エントリとの関係。たとえば、`passwd.org_dir` NIS+ テーブルの `name` 列のデータ (アカウントのユーザー名など) が、`ou=People` コンテナ内の `posixAccount` オブジェクトクラスの LDAP `uid` 属性に対応しているとする。構成によって、`name` 列と `uid` 属性が対応づけられる。`name` 列が `uid` 属性に対応づけられる (またはその逆) とも表現できる
- 主体

  認証可能な NIS+ のエンティティー (ユーザーまたはマシン)。通常、ネット名と主体名は 1 対 1 で対応づけられる

`rpc.nisd` 構成ファイル

2 つの構成ファイルを使用して、`rpc.nisd` 処理を制御します。

- `/etc/default/rpc.nisd`

  LDAP サーバーと認証、NIS+ ベースドメイン、LDAP デフォルト検索ベース、例外処理、および `rpc.nisd` の一般的な構成に関する情報を含みます。このファイルは、LDAP マッピングが有効であるかどうかにかかわらず適用されます。
- `/var/nis/NIS+LDAPmapping`

  NIS+ データと LDAP との間のマッピング情報を含みます。テンプレートファイル (`/var/nis/NIS+LDAPmapping.template`) は、`client_info.org_dir` と `timezone.org_dir` 以外のすべての標準 NIS+ オブジェクトに対応しています。306

ページの「client_info および timezone テーブル (NIS+ から LDAP への移行)」および NIS+LDAPmapping(4) のマニュアルページを参照してください。

構成とは、値を定義済みの属性に割り当てることです。構成ファイル以外に、構成属性を LDAP から読み取ることもできます (314 ページの「構成情報を LDAP に格納する」を参照)。また、rpc.nisd コマンドの -x オプションに構成属性を指定することもできます。複数の場所で同じ属性が指定されている場合、優先順位 (高から低) は次のとおりです。

1. rpc.nisd -x オプション
2. 構成ファイル
3. LDAP

NIS+ から LDAP への移行用ツールとサービス管理機能

NIS+ から LDAP への移行に関連するコマンド行管理タスクの大部分は、サービス管理機能によって管理されます。SMF の概要については、『Solaris のシステム管理 (基本編)』の第 18 章「サービスの管理 (概要)」を参照してください。また、詳細については、svcadm(1M) および svcs(1) のマニュアルページを参照してください。

- NIS+ から LDAP への移行サービスに関する有効化、無効化、再起動などの管理アクションは svcadm コマンドを使用して実行できる

  ヒント – -t オプションを使用してサービスを一時的に無効化すると、そのサービス構成に対していくらかの保護を提供できます。-t オプションを指定してサービスを無効にした場合、リブート後に元の設定が復元されます。-t オプションを指定しないでサービスを無効にした場合、リブート後もそのサービスは無効のままです。

- NIS+ の障害管理リソース識別子 (FMRI) は、svc:/network/rpc/nisplus:<*instance*> です。LDAP クライアントサービスの FMRI は、svc:/network/ldap/client:<*instance*> です。
- svcs コマンドを使用して NIS+ の状態を照会できる。
  - svcs コマンドと出力の例を、次に示します。

```
# svcs \*nisplus\*
STATE          STIME    FMRI
online         Sep_01   svc:/network/rpc/nisplus:default
```

- svcs -l コマンドと出力の例を、次に示します。次に示す出力を得るには、FMRI でインスタンス名を使用する必要があります。

```
# svcs -l network/rpc/nisplus:default
fmri         svc:/network/rpc/nisplus:default
```

```
enabled      false
state        disabled
next_state   none
restarter    svc:/system/svc/restarter:default
dependency   require_all/none svc:/network/rpc/keyserv (online)
```

- デーモンの存在は ps コマンドを使用して確認できます。

```
# ps -e | grep rpc.nisd
  root 23320     1   0   Aug 27 ?        16:30 ./ns-slapd -D \
  /usr/iplanet/ds5/slapd-lastrev -i /usr/iplanet/ds5/slapd-lastrev/
  root 25367 25353   0 15:35:19 pts/1     0:00 grep slapd
```

注 – -f オプションを ps で使用しないでください。このオプションはユーザー ID を名前に変換しようとするため、より多くのネームサービス検索が失敗する可能性があります。

NIS+ から LDAP への移行で SMF を使用しない場合

通常、/usr/sbin/rpc.nisd デーモンは、svcadm コマンドを使用して管理します。ただし、rpc.nisd デーモンは、-x nisplusLDAPinitialUpdateOnly=yes を指定して起動すると、指定された初期更新アクションを実行して終了します。つまり、rpc.nisd はデーモン化されません。-x nisplusLDAPinitialUpdateOnly=yes を指定した上で、サービス管理機能を使用してはなりません。それ以外の場合で、rpc.nisd デーモンを起動、停止または再起動するときにはいつでも SMF を使用できます。

次の例は、-x nisplusLDAPinitialUpdateOnly=yes を指定した rpc.nisd です。

```
# /usr/sbin/rpc.nisd -m mappingfile \
-x nisplusLDAPinitialUpdateAction=from_ldap \
-x nisplusLDAPinitialUpdateOnly=yes
```

/lib/svc/method/nisplus ファイルの変更

rpc.nisd デーモンをサービス管理機能によって起動するときに特定のオプションを含める場合、svcprop コマンドを使用するか、/lib/svc/method/nisplus ファイルを変更できます。svcprop コマンドの使用方法の詳細については、svcprop(1) のマニュアルページを参照してください。/lib/svc/method/nisplus ファイルを変更する手順を次に示します。

▼ /lib/svc/method/nisplus ファイルの変更方法

1. スーパーユーザーになるか、同等の役割を引き受けます。

   役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の第 9 章「役割によるアクセス制御の使用 (手順)」を参照してください。

2 **NIS+ サービスを停止します。**

```
# svcadm disable network/rpc/nisplus:default
```

3 `/lib/svc/method/nisplus` ファイルを開きます。

任意のエディタを使用してください。

4 ファイルを編集して必要なオプションを追加します。

変更前:

```
if [ -d /var/nis/data -o -d /var/nis/$hostname ]; then
                    /usr/sbin/rpc.nisd || exit $
```

変更後:

```
if [ -d /var/nis/data -o -d /var/nis/$hostname ]; then
                      /usr/sbin/rpc.nisd -Y -B || exit $?
```

この例では、`-Y` および `-B` オプションが `rpc.nisd` に追加され、起動時に自動的に実装されます。

5 `/lib/svc/method/nisplus` ファイルを保存して終了します。

6 **NIS+ サービスを開始します。**

```
# svcadm enable network/rpc/nisplus:default
```

属性とオブジェクトクラスの作成

NIS+ と LDAP のマッピングの構成によっては、新しい LDAP 属性とオブジェクトクラスをいくつか作成しなければならないことがあります。次の例では、これらの作成方法として、`ldapadd` コマンドの入力として使用できる LDIF データを指定する方法を示します。LDIF データを含むファイルを作成してから、`ldapadd(1)` を起動します。

```
# ldapadd -D bind-DN -f ldif -file
```

この方法は、Sun Java System Directory Server で機能します。また、その他の LDAP サーバーでも機能することがあります。

注 – ただし、`defaultSearchBase`、`preferredServerList`、および `authenticationMethod` 属性を除き、この章で使用されているオブジェクト識別子 (OID) は、SYNTAX 仕様と同様に、説明用に挙げているだけです。OID の基準はありません。任意の OID を使用できます。

NIS+ から LDAP への移行の開始前に必要な処置

NIS+ データを LDAP リポジトリに格納するために必要な構成の概要については、NIS+LDAPmapping(4) のマニュアルページを参照してください。ここでは、構成ファイルの編成について詳細に説明します。

/etc/default/rpc.nisd ファイル

/etc/default/rpc.nisd ファイルに値を割り当てるときは、すべて attributeName=value タイプとします。

一般的な構成

次の属性は、rpc.nisd の一般的な構成を制御し、LDAP マッピングが有効かどうかにかかわらずアクティブになります。これらの属性は通常、デフォルトのままにしておきます。詳細については、rpc.nisd(4) のマニュアルページを参照してください。

- nisplusNumberOfServiceThreads
- nisplusThreadCreationErrorAction
- nisplusThreadCreationErrorAttempts
- nisplusThreadCreationErrorTimeout
- nisplusDumpErrorAction
- nisplusDumpErrorAttempts
- nisplusDumpErrorTimeout
- nisplusResyncService
- nisplusUpdateBatching
- nisplusUpdateBatchingTimeout

LDAP からの構成データ

次の属性は、LDAP からのその他の構成属性の読み込みを制御します。これらの属性自体を LDAP に常駐させることはできません。コマンド行または構成ファイルから読み込む必要があります。詳細については、rpc.nisd(4) のマニュアルページを参照してください。

- nisplusLDAPconfigDN
- nisplusLDAPconfigPreferredServerList
- nisplusLDAPconfigAuthenticationMethod
- nisplusLDAPconfigTLS
- nisplusLDAPconfigTLSCertificateDBPath
- nisplusLDAPconfigProxyUser
- nisplusLDAPconfigProxyPassword

サーバーの選択

- `preferredServerList`

  LDAP サーバーとポート番号を指定します。

```
# LDAP server can be found at port 389
# LDAP server can be found at port 389
on the local machine
# preferredServerList=127.0.0.1
# Could also be written
# preferredServerList=127.0.0.0.1:389
LDAP server on the machine at IP
# address "1.2.3.4", at port 65042
# preferredServerList=1.2.3.4:65042
```

認証とセキュリティー

- `authenticationMethod`
- `nisplusLDAPproxyUser`
- `nisplusLDAPproxyPassword`

認証方式と、その方式に適切なプロキシユーザー (バインド識別名 DN) とパスワード (鍵またはその他の共有された機密情報)。これらは、`rpc.nisd` デーモンと LDAP サーバーの間で使用されます。詳細については、297 ページの「セキュリティーと認証」を参照してください。

- `nisplusLDAPTLS`
- `nisplusLDAPTLSCertificateDBPath`

必要に応じて、SSL を使用し、証明書ファイルの場所を指定します。詳細については、298 ページの「SSL の使用」を参照してください。

LDAP および NIS+ 内のデフォルトの場所

- `defaultSearchBase`

  LDAP DIT 内で、RFC 2307 に準拠したネームサービスデータのコンテナが配置される場所。コンテナ DN の完全な検索ベースを個別に指定しなかった場合は、この場所がデフォルトで使用されます。詳細については、287 ページの「`nisplusLDAPobjectDN` 属性」を参照してください。

- `nisplusLDAPbaseDomain`

  NIS+ オブジェクト仕様 (285 ページの「`nisplusLDAPdatabaseIdMapping` 属性」を参照) を完全指定しなかった場合は、このデフォルト NIS+ ドメイン名が使用されます。

LDAP 通信のタイムアウト制限、サイズ制限、および参照アクション

- nisplusLDAPbindTimeout
- nisplusLDAPmodifyTimeout
- nisplusLDAPaddTimeout
- nisplusLDAPdeleteTimeout

上のパラメータ (順番に、ldap bind、modify、add、および delete 操作) でタイムアウトを設定します。これらの属性は通常、デフォルトのままにしておきます。

- nisplusLDAPsearchTimeout
- nisplusLDAPsearchTimeLimit

上のパラメータには、LDAP 検索処理のタイムアウトを設定します。下のパラメータでは、サーバー側の検索時間制限を要求します。nisplusLDAPsearchTimeLimit では、LDAP サーバーが検索要求に使用する時間を制御します。このため、nisplusLDAPsearchTimeLimit には nisplusLDAPsearchTimeout 以上の値を設定してください。NIS+ サーバー、LDAP サーバー、および 2 つのサーバー間の接続のパフォーマンスに応じて、検索制限をデフォルト値より大きくしなければならないことがあります。rpc.nisd から送信されたタイムアウトに関するシステムログメッセージを基にして、これらの値を大きくするかどうかを判断します。

- nisplusLDAPsearchSizeLimit

  このパラメータでは、LDAP 検索要求に対して返される LDAP データ量に対する制限を要求します。デフォルトでは、制限は要求しません。この制限は、サーバー側に適用されます。LDAP サーバーでは、最大データ量に対して制限を適用することがあります。データ量の制限は、使用されているプロキシユーザー (バインド DN) に関連づけられることがあります。最も大きいコンテナに対して十分なデータが送信されるように、LDAP サーバーで rpc.nisd を設定してください。サイトによりますが、多くの場合、passwd.org_dir、mail_aliases.org_dir、または netgroup.org_dir のコンテナが使用されます。詳細については、LDAP サーバーのマニュアルを参照してください。

- nisplusLDAPfollowReferral

  このパラメータには、LDAP の処理中に別の LDAP サーバーへの参照が発生したときに、実行する処理を指定します。デフォルトでは、参照は行いません。参照を希望するか必要な場合は、参照を有効にします。参照は便利ですが、参照要求が発生するたびに rpc.nisd と複数の LDAP サーバーが対話する必要があるため、処理速度が遅くなることがあります。rpc.nisd には通常、発行する可能性のあるすべての LDAP 要求を処理できる LDAP サーバーを直接指定してください。

エラー処理

次のパラメータには、LDAP 処理中にエラーが発生したときに、実行する処理を指定します。これらのパラメータは通常、デフォルトのままにしておきます。詳細については、rpc.nisd(4) のマニュアルページを参照してください。

- nisplusLDAPinitialUpdateAction
- nisplusLDAPinitialUpdateOnly
- nisplusLDAPretrieveErrorAction
- nisplusLDAPretrieveErrorAttempts
- nisplusLDAPretrieveErrorTimeout
- nisplusLDAPstoreErrorAction
- nisplusLDAPstoreErrorAttempts
- nisplusLDAPstoreErrorTimeout
- nisplusLDAPrefreshErrorAction
- nisplusLDAPrefreshErrorAttempts
- nisplusLDAPrefreshErrorTimeout

一般的な LDAP 処理の制御

- nisplusLDAPmatchFetchAction

  このパラメータでは、NIS+ 照合処理のために、LDAP データを事前に取得するかどうかを決定します。ほとんどの場合、この値はデフォルトのままにしておきます。詳細については、rpc.nisd(4) のマニュアルページを参照してください。

/var/nis/NIS+LDAPmapping ファイル

デフォルトの NIS+LDAPmapping ファイルは、NIS+ および LDAP マッピングのマスタースイッチとして機能します。

デフォルト以外のマッピングファイルを使用する場合、-m *mappingfile* オプションを使用して /lib/svc/method/nisplus スクリプトを編集し、rpc.nisd 行にマッピングファイル名を指定する必要があります。詳細については、279 ページの「NIS+ から LDAP への移行用ツールとサービス管理機能」を参照してください。

LDAP に対応づける NIS+ オブジェクトごとに、NIS+LDAPmapping ファイルに 2 から 5 個の属性を指定します。指定する属性値は、オブジェクトとデフォルト値によって異なります。

nisplusLDAPdatabaseIdMapping 属性

エイリアスは、オブジェクトがほかのマッピング属性で使用されるときに設定する必要があります。NIS+ オブジェクト名が完全指定されていない場合 (ドットで終わっていない場合) は、nisplusLDAPbaseDomain の値が付加されます。

たとえば、次のように指定します。

```
nisplusLDAPdatabaseIdMapping    rpc:rpc.org_dir
```

このパラメータでは、データベース ID rpc を NIS+ rpc.org_dir テーブルのエイリアスとして定義しています。

NIS+ テーブルオブジェクトを 2 つの異なるデータベース ID ごとに 2 回定義する場合、テーブルオブジェクト自体 (このオブジェクトを LDAP に対応づける必要がある場合) として定義し、次にテーブルエントリとして定義します。たとえば、次のように指定します。

```
nisplusLDAPdatabaseIdMapping    rpc_table:rpc.org_dir
nisplusLDAPdatabaseIdMapping    rpc:rpc.org_dir
```

まず、データベース ID rpc_table と rpc を、rpc.org_dir テーブルのエイリアスとして定義します。次に、rpc_table を rpc.org_dir テーブルオブジェクトに使用し、rpc をそのテーブルのエントリに使用することを定義します。

nisplusLDAPentryTtl 属性

rpc.nisd デーモンのローカルデータベースは、メモリ内およびディスク上で LDAP データのキャッシュとして機能します。nisplusLDAPentryTtl 属性を使用すれば、そのキャッシュ内のエントリの生存期間 (TTL) 値を設定できます。各データベース ID には、3 つの TTL があります。最初の 2 つの TTL は、rpc.nisd が、対応する NIS+ オブジェクトデータをディスクから最初に読み込むときの初期 TTL を制御します。3 番目の TTL は、NIS+ オブジェクトデータを LDAP から読み込んだとき (更新したとき) にオブジェクトに割り当てられます。

たとえば、次の場合、rpc.org_dir テーブルオブジェクトは、21600 - 43200 秒の範囲からランダムに選択された初期 TTL を取得します。

```
nisplusLDAPentryTtl    rpc_table:21600:43200:43200
```

初期 TTL の生存期間が切れると、テーブルオブジェクトが LDAP から再度読み込まれ、TTL が 43200 秒に設定されます。

同様に、次の場合は、テーブルオブジェクトが最初に読み込まれたときに、1800 - 3600 秒から選択された初期 TTL が、 rpc.org_dir テーブルのエントリに割り当てられます。

```
nisplusLDAPentryTtl    rpc:1800:3600:3600
```

各エントリは、指定された範囲からランダムに選択された TTL を取得します。テーブルエントリが期間切れになり、更新されると、TTL は 3600 秒に設定されます。

TTL 値を選択するときは、パフォーマンスと整合性のバランスを考慮してください。rpc.nisd によって LDAP データがキャッシュされているときは、その TTL が大きい場合、rpc.nisd に LDAP データとの関連付けを設定していない場合とパ

フォーマンスは同じになります。しかし、rpc.nisd 以外のエンティティーによって LDAP データが変更されると、変更が NIS+ に反映されるまでにかなりの時間がかかります。

逆に、小さな値 (またはゼロ) を TTL に設定すると、LDAP データに対する変更が NIS+ にすばやく反映されますが、パフォーマンスが低下する可能性があります。通常、NIS+ 上で LDAP データの読み込みまたは書き込みを行うときは、LDAP 通信を行わない場合と比較して、2 - 3 倍の時間に加えて LDAP 検索の時間がかかります。パフォーマンスはハードウェアリソースによって大きく異なりますが、大きな LDAP コンテナ (数万から数十万のエントリ) を走査して、更新が必要な NIS+ エントリを識別するのは、かなりの時間を必要とします。rpc.nisd デーモンは、この走査をバックグラウンドで実行して、走査の実行中も可能なデータを供給し続けます。しかし、バックグラウンドで走査している場合でも、NIS+ サーバーの CPU とメモリは消費されます。

NIS+ データと LDAP を同期化する重要性を十分に考慮して、適用可能な最も長い TTL を NIS+ オブジェクトごとに選択してください。デフォルト (nisplusLDAPentryTtl を指定しないとき) は 1 時間です。テンプレートマッピングファイル /var/nis/NIS+LDAPmapping.template を適用すると、テーブルエントリ以外のオブジェクトの TTL が 12 時間に変更されます。ただし、テーブルエントリ以外のオブジェクトは自動的に認識されないため、テーブルエントリ以外のオブジェクトのマッピングを追加すると、TTL はデフォルトの 1 時間に設定されます。

注 – 存在しないオブジェクトには、TTL はありません。このため、NIS+ テーブル内で LDAP に対応づけられたエントリの TTL を有効にしても、NIS+ に存在しないエントリを要求すると、常に LDAP を照会してそのエントリを取得しようとします。

---

nisplusLDAPobjectDN 属性

nisplusLDAPobjectDN には、対応づけられた NIS+ オブジェクトごとに、オブジェクトデータが常駐する LDAP DIT 内の対応する場所を設定します。また、LDAP エントリが削除されたときに実行する処理も指定できます。nisplusLDAPobjectDN 値は、3 つの部分から構成されます。最初の部分には、LDAP データの読み込み元を指定します。2 番目の部分には、LDAP データの書き込み先を指定します。3 番目の部分には、LDAP データが削除されたときの処理を指定します。次の例を参照してください。

```
nisplusLDAPobjectDN    rpc_table:\
           cn=rpc,ou=nisPlus,?base?\
              objectClass=nisplusObjectContainer:\
           cn=rpc,ou=nisPlus,?base?\
              objectClass=nisplusObjectContainer,\
              objectClass=top
```

この例では、rpc.org_dir テーブルオブジェクトが DN cn=rpc,ou=nisPlus から読み込まれます。このとき、DN 値がコンマで終了しているため、defaultSearchBase 属性

(検索範囲) の値として base が付加されます。また、ObjectClass 属性の値が nisplusObjectContainer であるエントリが選択されます。

このテーブルオブジェクトは、読み込み元と同じ場所に書き込まれます。削除については指定されていないため、デフォルトの処理が実行されます。NIS+ テーブルオブジェクトが削除されると、LDAP エントリ全体も削除されます。

データを読み込むだけで LDAP に書き込まない場合は、書き込み部分を省略し、読み込み部分との区切り文字であるコロンも省略します。

```
nisplusLDAPobjectDN    rpc_table:\
             cn=rpc,ou=nisPlus,?base?\
             objectClass=nisplusObjectContainer
```

nisplusObjectContainer オブジェクトクラスは、RFC 2307 に準拠していません。このオブジェクトクラスを使用するには、LDAP サーバーを 300 ページの「テーブルエントリ以外の NIS+ オブジェクトのマッピング」で説明するように構成します。

rpc.org_dir テーブルエントリには、次の例も使用できます。

```
nisplusLDAPobjectDN rpc:ou=Rpc,?one?objectClass=oncRpc:\
             ou=Rpc,?one?objectClass=onRpc,objectClass=top
```

この例では、テーブルエントリの読み込みおよび書き込みが、ベース ou=Rpc に対して行われます。コンマで終了しているため、defaultSearchBase 値が付加されます。objectClass 属性の値が oncRpc であるエントリを選択してください。LDAP の ou=Rpc コンテナ内にエントリを作成するときは、objectClass の値として top も指定する必要があります。

デフォルト以外の削除を指定する場合は、次の例を参照してください。

```
nisplusLDAPobjectDN    user_attr:\
             ou=People,?one?objectClass=SolarisUserAttr,\
                solarisAttrKeyValue=*:\
             ou=People,?one?objectClass=SolarisUserAttr:\
                dbid=user_attr_del
```

user_attr.org_dir データは、ou=People LDAP コンテナに存在します。このコンテナは、ほかのソース (passwd.org_dir NIS+ テーブルなど) のアカウント情報も入ります。

そのコンテナ内のエントリから、solarisAttrKeyValue 属性を持つものを選択してください。user_attr.org_dir データが、これらのエントリにだけ含まれるためです。nisplusLDAPobjectDN の dbid=user_attr_del 部分の定義によって、user_attr.org_dir NIS+ テーブル内のエントリが削除されると、対応する LDAP エントリが存在する場合は、データベース ID が user_attr_del のルールセットのルールに基づいて削除されます。詳細については、289 ページの「nisplusLDAPcolumnFromAttribute 属性」を参照してください。

nisplusLDAPattributeFromColumn 属性

nisplusLDAPattributeFromColumn には、NIS+ データを LDAP に対応づけるときのルールを指定します。LDAP から NIS+ データへのマッピングルールは、nisplusLDAPcolumnFromAttribute に指定します。

nisplusLDAPcolumnFromAttribute 属性

nisplusLDAPcolumnFromAttribute には、LDAP データを NIS+ に対応づけるときのルールを指定します。

エントリマッピングの完全な構文については、NIS+LDAPmapping(4) のマニュアルページを参照してください。ここでは、わかりやすい例をいくつか挙げます。

NIS+ rpc.org_dir テーブルには、cname、name、numbe、および comment という列が含まれます。たとえば、NIS+ RPC プログラム番号 (100300) に対して、正規名として nisd が指定され、エイリアスとして rpc.nisd と nisplusd が指定されているとします。このエントリは、rpc.org_dir の次の NIS+ エントリを使用して表現できます。

```
nisd nisd 100300    NIS+ server
nisd rpc.nisd 100300    NIS+ server
nisd nisplusd 100300    NIS+ server
```

defaultSearchBase の値を dc=some,dc=domain とすると、対応する LDAP エントリは、ldapsearch(1) で次のように表示されます。

```
dn: cn=nisd,ou=Ppc,dc=some,dc=domain
cn: nisd
cn: rpc.nsid
cn: nisplusd
oncRpcNumber: 100300
description: NIS+ server
objectClass: oncRpc
```

この例は、NIS+ と LDAP が単純に 1 対 1 で対応づけられている場合です。この場合、NIS+ から LDAP へのマッピング属性値は、次のようになります。

```
nisplusLDAPattributeFromColumn \
rpc:        dn=("cn=%s,", name), \
                cn=cname, \
                cn=name, \
                oncRpcNumber=number, \
                description=comment
```

このエントリの DN として、cn=%s が構成されます。cname 列の値が %s に代入されます。

```
cn=nisd,
```

値がコンマで終了しているため、nisplusObjectDN の読み取りベース値が付加され、次のようになります。

```
cn=nisd,ou=Rpc,dc=some,dc=domain
```

oncRpcNumber 属性および description 属性の値には、対応する NIS+ 列の値が代入されます。rpc.nisd によって、複数の NIS+ エントリが 1 つの LDAP エントリに収集され、異なる name 列値を表す複数の cn 値が生成されます。

同様に、LDAP から NIS+ へのマッピングは、次のようになります。

```
nisplusLDAPcolumnFromAttribute \
       rpc:      cname=cn, \
                 (name)=(cn), \
                 number=oncRpcNumber, \
                 comment=description
```

この例では、oncRpcNumber および description の値が、対応する NIS+ 列に割り当てられます。(cn) で示される複数値列 cn は、(name) で示される複数の name 列値にマップされています。name 列は複数値列でないため、rpc.nisd によって cn 値ごとに 1 つの NIS+ エントリが作成されます。

最後に、削除に使用するルールセットの例を、nisplusLDAPattributeFromColumn 値を使って説明します。

```
nisplusLDAPattributeFromColumn \
user_attr_del:    dn=("uid=%s,", name), \
           SolarisUserQualifier=, \
           SolarisAttrReserved1=, \
           SolarisAttrReserved2=, \
           SolarisAttrKeyValue=
```

すでに述べたように、user_attr.org_dir データは、ほかのテーブル (passwd.org_dir など) のアカウント情報と、ou=People コンテナを共有しています。user_attr.org_dir テーブルのエントリが削除されたときに、ou=People エントリ全体を削除したいとはおそらく考えないでしょう。この例ではエントリ全体を削除する代わりに、user_attr.org_dir エントリが削除されたときに、SolarisUserQualifier、SolarisAttrReserved1、SolarisAttrReserved2、および SolarisAttrKeyValue 属性が存在する場合、次のルールに指定されている ou=People エントリから削除されます。

```
dn=("uid=%s,", name)
```

これ以外の LDAP エントリは変更されません。

NIS+ から LDAP への移行シナリオ

NIS+ から LDAP への移行シナリオの例を挙げます。

- すべての NIS+ クライアントを 1 回の操作で LDAP に変換する場合。rpc.nisd デーモンを使用すれば、LDAP に存在しないすべての NIS+ データをアップロードできます。291 ページの「すべての NIS+ データを 1 回の操作で LDAP に変換する方法」を参照してください。

- NIS+ から LDAP に段階的に移行する場合。まず、NIS+ データを LDAP に変換します (291 ページの「すべての NIS+ データを 1 回の操作で LDAP に変換する方法」を参照)。NIS+ クライアントと LDAP クライアントで、同じネームサービスを共有することができます。NIS+ および LDAP のデータは、`rpc.nisd` によって自動的に同期化されます。移行の初期段階では場合によって、NIS+ が認証されたサーバーとして機能し、LDAP サーバーは、NIS+ データを複製して、LDAP クライアントに提供します。適切な段階で、LDAP を認証されたネームサービスに移行します。NIS+ サービスは、段階的に処理を停止していき、NIS+ クライアントの移行が完了した時点で完全になくなります。
- LDAP がすでにネームサービスとして使用されている場合。NIS+ データと LDAP データをマージする必要があります。次の 3 つのマージ方法があります。
  - NIS+ データを LDAP に追加する方法。NIS+ に存在するが LDAP に存在しないエントリが、LDAP に追加されます。エントリが NIS+ および LDAP の両方に存在するが、データが異なる場合は、NIS+ データが優先されます。291 ページの「すべての NIS+ データを 1 回の操作で LDAP に変換する方法」を参照してください。
  - NIS+ データを LDAP データで上書きする方法。NIS+ に存在するが LDAP に存在しないエントリが、NIS+ から削除されます。NIS+ および LDAP の両方に存在するエントリでは、 LDAP データが優先されます。292 ページの「すべての LDAP データを 1 回の操作で NIS+ に変換する方法」を参照してください。
  - NIS+ データと LDAP データをマージする方法。衝突が発生した場合は、個別に解決します。292 ページの「NIS+ データと LDAP データのマージ」を参照してください。

▼ すべての NIS+ データを 1 回の操作で LDAP に変換する方法

- `rpc.nisd` を使用して、**LDAP** に存在しない **NIS+** データをすべてアップロードします。

  NIS+ と LDAP のすべてのデータマッピングが、デフォルトの場所 (`/var/nis/NIS+LDAPmapping`) に設定されている場合は、次のコマンドを使用します。

  ```
  # /usr/sbin/rpc.nisd -D \
   -x nisplusLDAPinitialUpdateAction=to_ldap \
   -x nisplusLDAPinitialUpdateOnly=yes
  ```

  上記のコマンドによって、`rpc.nisd` デーモンによりデータが LDAP にアップロードされて、変換が終了します。この処理を実行しても、NIS+ データは変更されません。

  `rpc.nisd(4)` のマニュアルページの `nisplusLDAPinitialUpdateAction` 属性を参照してください。

▼ すべての LDAP データを 1 回の操作で NIS+ に変換する方法

- rpc.nisd を使用して、すべての LDAP データを NIS+ にダウンロードし、既存の NIS+ データを上書きします。

  NIS+ と LDAP のすべてのデータマッピングが、デフォルトの場所 (/var/nis/NIS+LDAPmapping) に設定されている場合は、次のコマンドを使用します。

  ```
  # /usr/sbin/rpc.nisd -D \
  -x nisplusLDAPinitialUpdateAction=from_ldap \
  -x nisplusLDAPinitialUpdateOnly=yes
  ```

  上記のコマンドによって、rpc.nisd デーモンによりデータが LDAP からダウンロードされて、変換が終了します。この処理を実行しても、LDAP データは変更されません。

  rpc.nisd(4) のマニュアルページの nisplusLDAPinitialUpdateAction 属性を参照してください。

NIS+ データと LDAP データのマージ

NIS+ および LDAP 間でデータの衝突が発生したときは、NIS+ または LDAP データのどちらかを正式なものとして解決しなければなりません。290 ページの「NIS+ から LDAP への移行シナリオ」 では、NIS+ データと LDAP データを同期化する方法について説明しています。データをマージするときは、ほかの方法と比べて複雑な手順が必要になります。

この節で挙げた手順では、次の点を前提としています。

- NIS+ データのバックアップを /nisbackup ディレクトリに作成する
- 有効なマッピング構成が /etc/default/rpc.nisd および /var/nis/tmpmap (テーブルをマージする場合) にすでに存在する
- マージ前の NIS+ データのフラットファイル表現を /before に格納し、マージ後は /after に格納する
- niscat は、nisaddent(1M) でサポートされないカスタム NIS+ テーブルを、フラットファイル表現でダンプするときに使用する。このようなカスタムテーブルを、NIS+ からまたは NIS+ にダンプして読み込むために、独自のコマンドまたはスクリプトを作成することがある。この場合は、niscat に優先して、独自のコマンドまたはスクリプトを使用する。niscat コマンドには、フラットファイル表現のデータを NIS+ に戻す便利な方法がないためである

  niscat(1) を使用してデータをダンプした場合は、nistbladm(1) を使用すれば、エントリを 1 つずつ NIS+ に戻すことができる
- コマンドパスに /usr/lib/nis (nisaddent(1M) が存在する場所) を含める

▼ NIS+ データと LDAP データをマージする方法

注意 – 手順 4 のダウンロードデータと、手順 10 のアップロードデータが一致しない場合は、アップロードデータによって変更が上書きされます。このため、この手順を実行しているときは、LDAP データの変更はできるだけ避けてください。詳細については、LDAP サーバーのマニュアルを参照してください。

1 nisbackup コマンドを使用して、すべての NIS+ データをバックアップします。

```
# nisbackup -a /nisbackup
```

2 LDAP とマージするデータが含まれる NIS+ テーブルを特定します。これらのテーブルの内容をフラットファイルにダンプします。たとえば、次のように nisaddent を使用して group.org_dir の内容をダンプします。

```
# nisaddent -d group | sort > /before/group
```

パイプを使って nisaddent の出力を sort の入力として渡すと、比較処理が簡単になります。

3 LDAP データを NIS+ にダウンロードします。

```
# /usr/sbin/rpc.nisd -D -m tmpmap \
-x nisplusLDAPinitialUpdateAction=from_ldap \
-x nisplusLDAPinitialUpdateOnly=yes
```

4 NIS+ サービスを開始します。

```
# svcadm enable network/rpc/nisplus:default
```

rpc.nisd デーモンが、LDAP からダウンロードしたデータを提供するようになります。解決を必要とする衝突を NIS+ クライアント上で発生させないようにする必要がある場合は、これ以降の手順は、ほとんどまたはすべての NIS+ クライアントが動作していないときに実行してください。

5 影響を受けるテーブルの NIS+ データをダンプします。

次の例では、group.org_dir テーブルをダンプします。

```
# nisaddent -d group | sort > /after/group
```

6 マージしたテーブルを作成します。

任意のマージ手順を使用して、マージ済みテーブルを作成できます。diff(1) 以外のツールを使用できない場合は、diff(1) コマンドを使用して /before ファイルと /after ファイルとの相違点を収集し、テキストエディタを使用して手動でマージすることができます。

次の例では、マージ後のテーブルが /after に格納されていることを前提としています。

7 マージ後のデータを NIS+ に読み込みます。次の例では、group テーブルを読み取ります。

```
# nisaddent -m -f /after/group group
```

8 マージ後のテーブルから、不要な LDAP エントリを削除します。

A. マージ後の NIS+ データ内に存在しない LDAP エントリが、アップロード後の LDAP に必要がない場合、これらの LDAP エントリは削除する必要があります。

LDAP サーバーには、コンテナのすべてのエントリを削除する方法など、複数のエントリを削除する便利な方法が提供されていることがあります。提供されていない場合は、ldapsearch(1) を使用して、各コンテナのエントリの一覧を生成することができます。たとえば、ou=Rpc コンテナに含まれるすべてのエントリの一覧を生成するには、ldapsearch(1) を次のように使用します。

```
# ldapsearch -h server-address -D bind-DN -w password \
 -b ou=Rpc,search-base 'objectClass=*' dn | \
grep -i ou=Rpc | grep -v -i \^ou=Rpc > /tmp/delete-dn
```

メタ引数 (*server-address*、 *bind-DN* など) については、299 ページの「パフォーマンスとインデックス処理」を参照してください。

B. 結果ファイル (/tmp/delete-dn) を編集して、削除するエントリだけを指定します。コンテナのすべてのエントリを削除する場合は、該当するファイルは操作しないで、NIS+ アップロードを使用して LDAP データを復元することもできます。どちらの方法を使用する場合でも、LDAP データをバックアップしてから、次の ldapdelete 操作を実行してください。

C. ldapdelete を使用して、LDAP エントリを削除します。stdout (通常は、削除したエントリごとに空白行が 1 行ずつ出力される) は、/dev/null にリダイレクトします。

```
# ldapdelete -h server-address -D bind-DN -w password \
/tmp/delete-dn /dev/null
```

D. 削除するエントリが 1 つ以上含まれるコンテナごとに、前述の手順を繰り返します。

マスターと複製 (NIS+ から LDAP への移行)

NIS+ マスターだけが、データを LDAP に書き込むことができます。NIS+ 複製は、NIS+ マスターから更新を取得するか (LDAP から取得する場合を含む)、LDAP サーバーから直接データを読み込みます。この 2 つの方法を組み合わせることもできます。つまり、NIS+ 複製を使用するときは、主に 2 つの方法があります。

- NIS+ 複製は変更せずに使用し、更新データは NIS+ マスターから取得する

  この方法の場合は、NIS+ マスター以外は、LDAP サーバーと接続する必要がないため、構成が単純になります。従来の複製関係も変更されません。つまり、新しいデータはまずマスターに反映され、次に複製に反映されます。ネームサービス

のデータを従来どおり NIS+ が管理するときは、ほとんどの場合、この方法が最も便利な方法です。ただし、LDAP と NIS+ 複製サーバーとの間のパスが長くなります。

- NIS+ 複製は、更新データを NIS+ マスターから取得しないで、LDAP から直接取得する

  この場合、複製のデータの更新は、LDAP から取得したデータの検索トラフィックおよび TTL に基づいて、NIS+ マスターの更新前または更新後に行われます。この方法はより複雑ですが、LDAP がネームサービスリポジトリを管理するときは、便利な方法です。NIS+ データに対する直接の更新は、ほとんどまたはまったく発生しません。

複製タイムスタンプ

NIS+ 複製が特定の NIS+ ディレクトリに含まれる 1 つ以上のオブジェクトのデータを LDAP から取得しているときは、`nisping(1M)` によって出力される更新タイムスタンプが NIS+ マスターおよび NIS+ 複製間のデータの整合性を示しているとは限りません。たとえば、NIS+ ディレクトリ `dir1` に `table1` および `table2` テーブルが含まれているとします。複製が `table1` および `table2` のデータを NIS+ マスターから取得しているときは、次のようなタイムスタンプが出力されます。

```
# nisping dir1

Master server is "master.some.domain."
Last update occurred at Mon Aug  5 22:11:09 2002

Replica server is "replica.some.domain."
    Last Update seen was Mon Aug  5 22:11:09 2002
```

これらのタイムスタンプは、マスターおよび複製のデータが完全に一致していることを示しています。しかし、複製が `table1` または `table2`、あるいはその両方のデータを LDAP から取得しているとします。この場合、この出力には、複製がマスターから NIS_PING を受け取り、再同期化のタイムスタンプをシステム管理用に更新したことだけが示されます。LDAP に対応づけられているテーブルのデータは、次のどちらかの場合、NIS+ マスター上のデータと異なることがあります。

- LDAP データが NIS+ マスター上のデータと異なる
- 複製は、ローカルな NIS+ データベースであるキャッシュにデータを格納している。このキャッシュは期限切れではないが、LDAP と同期がとれていない。

このようなデータの不一致を許容できない場合は、NIS+ 複製が常に NIS+ マスターからデータを取得するようにします。NIS+ マスターが LDAP からデータを取得するように構成した場合は、複製を変更する必要はありません。

ディレクトリサーバー (NIS+ から LDAP への移行)

rpc.nisd デーモンに含まれる LDAP マッピングでは、LDAP プロトコルバージョン 3 を使用して LDAP サーバーと対話します。デフォルトのマッピング構成 (/var/nis/NIS+LDAPmapping.template) では、LDAP サーバーが RFC 2307 の拡張版に準拠していることを前提としています。RFC は、http://www.ietf.org/rfc.html から入手できます。NIS+ データと LDAP データとのマッピングは、NIS+LDAPmapping(4) を使用して変更できます。ただし、LDAP のデータ編成が RFC 2307 の規定に準拠していることを、基本的な前提としています。

たとえば、LDAP クライアントと NIS+ クライアントとの間でアカウント情報を共有するには、UNIX crypt 書式のアカウント (ユーザー) パスワードを LDAP サーバーに格納できるようにする必要があります。LDAP サーバーをこのように構成できない場合でも、アカウントを含む NIS+ データを LDAP に格納することはできます。その場合、NIS+ ユーザーと LDAP bindDN との間でアカウント情報を完全に共有することはできません。

Sun Java System Directory Server の構成

Sun Java System Directory Server のインストール、設定、および管理の詳細については、Sun Java System Directory Server collection を参照してください。

Sun Java System Directory Server を構成して、LDAP クライアントが LDAP をネームサービスとして使用できるようにするには、idsconfig(1M) を使用します。idsconfig(1M) を使用して設定した構成は、NIS+ で LDAP データリポジトリを使用する場合にも適しています。

注 – Sun Java System Directory Server 以外の LDAP サーバーを使用している場合は、RFC 2307 に準拠するように、サーバーを手動で構成する必要があります。

サーバーアドレスとポート番号の割り当て

/etc/default/rpc.nisd ファイルは、ローカル LDAP サーバーをポート 389 で使用するように設定されています。この設定が現在の構成に適していない場合は、preferredServerList 属性に新しい値を設定します。たとえば、LDAP サーバーを IP アドレス 192.0.0.1 とポート 65535 で使用するには、次のように指定します。

```
preferredServerList=192.0.0.1:65535
```

セキュリティーと認証

NIS+ クライアントおよび NIS+ サーバー間の認証は、NIS+ サーバーが LDAP からデータを取得する場合でも、影響することはありません。ただし、NIS+ データを LDAP に格納するときの整合性を保持するには、`rpc.nisd` デーモンおよび LDAP サーバー間の認証を必要に応じて設定する必要があります。LDAP サーバーの機能に応じて、さまざまなタイプの認証を利用できます。

`rpc.nisd` デーモンでは、次の LDAP 認証を利用できます。

- `none`

  `none` は、デフォルトの認証方式です。`none` には、固有の設定は必要ありません。ただし、セキュリティーは保証されません。セキュリティーを考慮する必要がない環境だけで使用してください。

  `none` 認証を使用するときは、`authenticationMethod` 属性に次の値を設定してください。

  ```
  authenticationMethod=none
  ```

この認証方式を利用するときに一定のセキュリティーを保証するには、多くの場合、共有された機密情報 (パスワードまたは鍵) と LDAP の DN を関連付ける必要があります。`rpc.nisd` デーモンで使用する DN は一意なものであり、ほかの目的で使用することもできます。予測される LDAP トラフィックに対応するために、DN には適切な権限を割り当てる必要があります。たとえば、`rpc.nisd` デーモンが LDAP にデータを書き込む場合は、NIS+ データに使用されるコンテナ内で LDAP データを追加、更新、および削除する権限を、選択した DN に割り当てる必要があります。また、LDAP サーバーでは、リソースの使用方法がデフォルトで制限されている場合があります (検索時間制限、検索結果のサイズ制限など)。この制限がある場合は、必要な数の NIS+ データコンテナをサポートできるように、選択した DN に対して必要な設定をする必要があります。

- `simple`

  `simple` 認証方式では、暗号化されていないパスワード文字列が交換されます。パスワードは、LDAP クライアント (`rpc.nisd` デーモン) および LDAP サーバー間をプレーンテキストとして送信されます。このため、`simple` 方式は、NIS+ と LDAP サーバー間の情報交換が別の方式で保護されている場合にだけ使用してください。

  たとえば、LDAP トラフィックのトランポート層を暗号化するときに使用します。また、NIS+ サーバーと LDAP サーバーが同一システム上にあり、NIS+ および LDAP のトラフィックがカーネル内で処理され、認証されていないユーザーから保護されている場合にも使用できます。

  `simple` 認証を使用するときは、`rpc.nisd` デーモンで使用する DN とパスワードの構成を変更してください。たとえば、DN が `cn=nisplusAdmin, ou=People, dc=some, dc=domain` で、パスワードが `aword` の場合は、次のように設定します。

```
authenticationMethod=simple
nisplusLDAPproxyUser=cn=nisplusAdmin,ou=People,dc=some,dc=domain
nisplusLDAPproxyPassword=aword
```

パスワードが格納されている場所は、認証されないアクセスから確実に保護する必要があります。パスワードを `rpc.nisd` コマンド行で指定した場合は、ps(1) などのコマンドによってシステム上の任意のユーザーに見られる可能性があります。

- `sasl/digest-md5`

  `sasl/digest-md5` 認証方式では、`digest/md5` アルゴリズムを使用して認証が行われます。

  `digest-md5` で使用する承認 ID を設定する方法と、`/etc/default/rpc.nisd` ファイルに承認 ID とそのパスワードを指定する方法については、LDAP サーバーのマニュアルを参照してください。

  ```
  authenticationMethod=sasl/digest-md5
  nisplusLDAPproxyUser=cn=nisplusAdmin,ou=People,dc=some,dc=domain
  nisplusLDAPproxyPassword=aword
  ```

  パスワードを格納するファイルを、承認されていないアクセスから確実に保護してください。

- `sasl/cram-md5`

  `cram/md5` アルゴリズムを使用した認証方式。通常は、現在使用されていない SunDS LDAP サーバー以外では使用されません。

  `cram-md5` を使用してバインド DN を設定する方法と、`/etc/default/rpc.nisd` ファイルにバインド DN とそのパスワードを指定する方法については、LDAP サーバーのマニュアルを参照してください。

  ```
  authenticationMethod=sasl/cram-md5
  nisplusLDAPproxyUser=cn=nisplusAdmin,ou=People,dc=some,dc=domain
  nisplusLDAPproxyPassword=aword
  ```

  パスワードを格納するファイルを、承認されていないアクセスから確実に保護してください。

SSL の使用

`rpc.nisd` デーモンは、SSL を使用した LDAP トラフィックのトランスポート層の暗号化にも対応しています。LDAP サーバー認証用の SSL 証明書の生成については、LDAP サーバーのマニュアルを参照してください。SSL 証明書は、NIS+ サーバー上のファイル (`/var/nis/cert7.db` など) に格納します。`/etc/default/rpc.nisd` は、次のように変更します。

```
nisplusLDAPTLS=ssl
nisplusLDAPTLSCertificateDBPath=/var/nis/cert7.db
```

SSL 証明書は、承認されていないアクセスから確実に保護する必要があります。この例では、セッションの暗号化と LDAP サーバーの認証が rpc.nisd に提供されます。SSL 証明書では、LDAP サーバーに対する rpc.nisd の認証は提供されません。この証明書には、この LDAP クライアント (rpc.nisd) の識別情報が含まれていないためです。ただし、rpc.nisd と LDAP サーバーが相互に認証するには、SSL と別の認証方式 (simple、sasl/digest-md5) を組み合わせることができます。

パフォーマンスとインデックス処理

niscat(1) などを使用して、LDAP に対応づけられた NIS+ テーブルの列挙を rpc.nisd デーモンに要求すると、テーブル内のエントリの TTL が 1 つでも期限切れになっている場合は、対応する LDAP コンテナが列挙されます。コンテナの列挙はバックグラウンドで実行されるため、LDAP のパフォーマンスはそれほど重要ではありません。ただし、LDAP にインデックスを設定すれば、コンテナが大きい場合でもすばやく列挙することができます。

特定のコンテナの列挙に必要な時間を見積もるには、次のようなコマンドを使用します。

% **/bin/time ldapsearch** -h *server-address* -D *bind-DN* -w *password* ****

-b *container* **,** *search-base 'cn=*'* **/dev/null**

次に、各引数について説明します。

- *server-address*

  /etc/default/rpc.nisd の preferredServerList 値の IP アドレス部分
- *bind-DN*

  /etc/default/rpc.nisd の nisplusLDAPproxyUser 値
- *password*

  /etc/default/rpc.nisd の nisplusLDAPproxyPassword 値
- *container*

  RFC 2307 に準拠したコンテナ名 (ou=Services、ou=Rpc など)
- *search-base*

  /etc/default/rpc.nisd の defaultSearchBase 値

/bin/time から出力される実際の値は、経過時間です。この値が、対応するテーブルエントリの TTL を 25 パーセント以上占めている場合は (283 ページの「認証とセキュリティー」 を参照)、LDAP コンテナにインデックスを設定すると有効です。

rpc.nisd では、simple page と VLV インデックス方式がサポートされます。ご使用の LDAP サーバーでサポートされているインデックス方式、およびそのインデックスの作成方法については、LDAP サーバーのマニュアルを参照してください。

テーブルエントリ以外の NIS+ オブジェクトのマッピング

テーブルエントリ以外の NIS+ オブジェクトを LDAP に格納できます。ただし、NIS+ 複製が LDAP からこれらの NIS+ オブジェクトを取得しない限り、LDAP に格納しても値は設定されません。次の方法をお勧めします。

- 複製がない場合、または複製が NIS+ データを NIS+ マスターだけから取得する場合。

  マッピング構成ファイル (NIS+LDAPmapping(4) のマニュアルページを参照) を編集して、テーブルエントリ以外のすべてのオブジェクトから次の属性値を削除します。

  ```
  nisplusLDAPdatabaseIdMapping
  nisplusLDAPentryTtl
  nisplusLDAPobjectDN
  ```

  たとえば、/var/nis/NIS+LDAPmapping.template ファイルの場合は、次のセクションを削除するか、コメントにして無効にします。

  ```
  # Standard NIS+ directories
  nisplusLDAPdatabaseIdMapping    basedir:
  .
  .
  .

  nisplusLDAPdatabaseIdMapping    user_attr_table:user_attr.org_dir

  nisplusLDAPdatabaseIdMapping     audit_user_table:audit_user.org_dir

  # Standard NIS+ directories
  nisplusLDAPentryTtl             basedir:21600:43200:43200
  .
  .
  .

  nisplusLDAPentryTtl    user_attr_table:21600:43200:43200
  nisplusLDAPentryTtl    audit_user_table:21600:43200:43200

  # Standard NIS+ directories
  nisplusLDAPobjectDN    basedir:cn=basedir,ou=nisPlus,?base?\

         objectClass=nisplusObjectContainer:\
         cn=basedir,ou=nisPlus,?base?\
         objectClass=nisplusObjectContainer,\
         objectClass=top
  .
  .
  .

  nisplusLDAPobjectDN    audit_user_table:cn=audit_user,ou=nisPlus,?base?\
         objectClass=nisplusObjectContainer:\
         cn=audit_user,ou=nisPlus,?base?\
         objectClass=nisplusObjectContainer,\
         objectClass=top
  ```

- NIS+ 複製が NIS+ データを LDAP サーバーから取得する場合。

nisplusObject 属性と nisplusObjectContainer オブジェクトクラスを次の例に従って作成します。LDIF データは ldapadd(1) に適しています。属性とオブジェクトクラス OID は、例として挙げているだけです。

```
dn: cn=schema
changetype: modify
add: attributetypes
attributetypes: ( 1.3.6.1.4.1.42.2.27.5.42.42.1.0 NAME 'nisplusObject'
       DESC 'An opaque representation of an NIS+ object'
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.5 SINGLE-VALUE )

dn: cn=schema
changetype: modify
add: objectclasses

objectclasses: (1.3.6.1.4.1.42.2.27.5.42.42.2.0 NAME'nisplusObjectContainer'

SUP top STRUCTURAL DESC 'Abstraction of an NIS+ object'
MUST ( cn $ nisplusObject ) )
```

NIS+ オブジェクトのコンテナも作成する必要があります。次の LDIF 構文は、ou=nisPlus,dc=some,dc=domain コンテナの作成方法を示しています。このコンテナは、ldapadd(1) の入力として使用できます。

```
dn: ou=nisPlus,dc=some,dc=domain
ou: nisPlus
objectClass: top
objectClass: organizationalUnit
```

NIS+ エントリの所有者、グループ、アクセス権、および TTL

NIS+ テーブルエントリを LDAP データから作成するときは、そのエントリオブジェクトが存在するテーブルオブジェクトの対応する値を使用して、所有者、グループ、アクセス権、および TTL を初期化する必要があります。環境によっては、これらの NIS+ エントリ属性を個別に設定する必要があります。たとえば、rpc.nispasswdd(1M) デーモンを使用しない環境では、この操作が必要になります。ユーザー自身が NIS+ パスワードを変更して、cred.org_dir テーブルに格納されている Diffie-Hellman キーを再暗号化できるようにするには、passwd.org_dir および cred.org_dir エントリの所有者をそのユーザーに設定し、その所有者に変更権限を割り当てる必要があります。

1 つ以上の NIS+ テーブルエントリの所有者、グループ、アクセス権、または TTL を LDAP に格納するには、次の操作を実行する必要があります。

▼ エントリ属性を LDAP に追加するには

1 **LDAP** サーバーのマニュアルを参照して、次の新しい属性とオブジェクトクラスを作成します。**LDIF** データは、`ldapadd` に適用できます。属性とオブジェクトクラス **OID** は、例として挙げているだけです。

```
dn: cn=schema
changetype: modify
add: attributetypes
attributetypes: ( 1.3.6.1.4.1.42.2.27.5.42.42.4.0 NAME 'nisplusEntryOwner' \
                DESC 'Opaque representation of NIS+ entry owner' \
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.4.1 NAME 'nisplusEntryGroup' \
                DESC 'Opaque representation of NIS+ entry group' \
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.4.2 NAME 'nisplusEntryAccess' \
                DESC 'Opaque representation of NIS+ entry access' \
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.4.3 NAME 'nisplusEntryTtl' \
                DESC 'Opaque representation of NIS+ entry TTL' \
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )


dn: cn=schema
changetype: modify
add: objectclasses


objectclasses:(1.3.6.1.4.1.42.2.27.5.42.42.5.0 NAME 'nisplusEntryData'\
SUP top STRUCTURAL DESC 'NIS+ entry object non-column data'\


MUST ( cn ) MAY ( nisplusEntryOwner $ nisplusEntryGroup $\
nisplusEntryAccess $ nisplusEntryTtl ) )
```

2 関連するテーブルの `nisplusLDAPobjectDN` 属性値を変更して、新しく作成した `nisplusEntryData` オブジェクトクラスを書き込み部分に含めます。

たとえば、`passwd.org_dir` テーブルの場合、`/var/nis/NIS+LDAPmapping.template` をベースにしたテンプレートファイルを使用しているときは、次のように編集します。

```
nisplusLDAPobjectDN    passwd:ou=People,?one?objectClass=shadowAccount,\

                objectClass=posixAccount:\
                ou=People,?one?objectClass=shadowAccount,\
                objectClass=posixAccount,\
                objectClass=account,objectClass=top
```

属性値を次のように編集します。

```
nisplusLDAPobjectDN    passwd:ou=People,?one?objectClass=shadowAccount,\

                objectClass=posixAccount:\
                ou=People,?one?objectClass=shadowAccount,\
                objectClass=posixAccount,\
                objectClass=nisplusEntryData,\
                objectClass=account,objectClass=top
```

3 nisplusLDAPattributeFromColumn 属性値および nisplusLDAPcolumnFromAttribute 属性値を編集して、所有者、グループ、アクセス権、または **TTL** を必要に応じて指定します。

手順 2 で、これらの値を格納する LDAP 属性を作成しました。NIS+ には、zo_owner、zo_group、zo_access、および zo_ttl と呼ばれる定義済みの擬似列名が、あらかじめ定義されています。たとえば、passwd.org_dir エントリの所有者、グループ、およびアクセス権を LDAP に格納するには、次の nisplusLDAPattributeFromColumn 値を変更します。

```
nisplusLDAPattributeFromColumn \
        passwd:        dn=("uid=%s,", name), \
                cn=name, \
                uid=name, \
                userPassword=("{crypt$}%s", passwd), \
                uidNumber=uid, \
                gidNumber=gid, \
                gecos=gcos, \
                homeDirectory=home, \
                loginShell=shell, \
                (shadowLastChange,shadowMin,shadowMax, \
                 shadowWarning, shadowInactive,shadowExpire)=\
                    (shadow, ":")
```

次のように編集します。

```
nisplusLDAPattributeFromColumn \
        passwd:        dn=("uid=%s,", name), \
                cn=name, \
                uid=name, \
                userPassword=("{crypt$}%s", passwd), \
                uidNumber=uid, \
                gidNumber=gid, \
                gecos=gcos, \
                homeDirectory=home, \
                loginShell=shell, \
                (shadowLastChange,shadowMin,shadowMax, \
                 shadowWarning, shadowInactive,shadowExpire)=\
                    (shadow, ":"), \
                nisplusEntryOwner=zo_owner, \
                nisplusEntryGroup=zo_group, \
                nisplusEntryAccess=zo_access
```

同様に、 NIS+ エントリの所有者、グループ、LDAP データからのアクセス権を passwd.org_dir テーブルに設定するには、次の値を変更します。

```
nisplusLDAPcolumnFromAttribute \
        passwd:        name=uid, \
                ("{crypt$}%s", passwd)=userPassword, \
                uid=uidNumber, \
                gid=gidNumber, \
                gcos=gecos, \
                home=homeDirectory, \
                shell=loginShell, \
                shadow=("%s:%s:%s:%s:%s:%s", \
                    shadowLastChange, \
```

```
                    shadowMin, \
                    shadowMax, \
                    shadowWarning, \
                    shadowInactive, \
                    shadowExpire)
```

次のように編集します。

```
nisplusLDAPcolumnFromAttribute \
        passwd:         name=uid, \
                ("crypt$%s", passwd)=authPassword, \
                uid=uidNumber, \
                gid=gidNumber, \
                gcos=gecos, \
                home=homeDirectory, \
                shell=loginShell, \
                shadow=("%s:%s:%s:%s:%s:%s", \
                    shadowLastChange, \
                    shadowMin, \
                    shadowMax, \
                    shadowWarning, \
                    shadowInactive, \
                    shadowExpire), \
                zo_owner=nisplusEntryOwner, \
                zo_group=nisplusEntryGroup, \
                zo_access=nisplusEntryAccess
```

4 所有者、グループ、アクセス権、および **TTL** エントリデータのどれか、またはすべてを **LDAP** にアップロードします。

詳細については、291 ページの「すべての NIS+ データを 1 回の操作で LDAP に変換する方法」を参照してください。

5 マッピングの変更を有効にするために、**NIS+** サービスを再起動します。

```
# svcadm restart network/rpc/nisplus:default
```

主体名とネット名 (NIS+ から LDAP への移行)

NIS+ 認証は、主体名 (ドメイン名で指定されたユーザー名またはホスト名) とネット名 (SecureRPC での主体名) に基づいて認証可能なエンティティー (主体) を一意に識別します。RFC 2307 では、NIS+ 認証に使用する Diffie-Hellman 鍵の格納場所は規定していますが、主体名またはネット名の格納場所は規定していません。

`/var/nis/NIS+LDAPmapping.template` ファイルでは、この問題を回避するために、`cred.org_dir` テーブルの所有者名 (主体名) から主体名およびネット名のドメイン部分を派生します。つまり、NIS+ ドメインが `x.y.z.` で、`cred.org_dir` テーブルの所有者が `aaa.x.y.z.` の場合、LDAP データから作成された NIS+ エントリの主体名は、次の形式になります。

user or system`.x.y.z.`

ネット名は次の形式になります。

unix.*uid***@x.y.z.**

unix.*nodename***@x.y.z.**

ほとんどの NIS+ インストールでは、主体名とネット名を作成するときは、この方式でかまいません。ただし、次のような場合は、この方式では成功しません。

- cred.org_dir テーブルの所有者名が属しているドメインが、cred.org_dir テーブル内の主体名およびネット名によって共有されるドメインと一致していない場合。所有者が親ドメインの主体である場合は、サブドメインの cred.org_dir テーブルも同様です。この問題は、次のいずれかの方法で解決できます。
  - cred.org_dir テーブルの所有者を変更して、テーブルエントリのドメインと一致させます。
  - cred.org_dir データベース ID のマッピングルールを変更して、ほかの NIS+ オブジェクトの所有者を使用します。適切なオブジェクトが存在しない場合は、この目的のために NIS+ オブジェクトを作成します。

    たとえば、sub.dom.ain. ドメインの cred.org_dir テーブルを master.dom.ain. が所有し、cred.org_dir.sub.dom.ain. の主体とネット名が sub.dom.ain に属している場合は、次のようなリンクオブジェクトを作成します。

    ```
    # nisln cred.org_dir.sub.dom.ain. \
    credname.sub.dom.ain.
    ```

    リンクオブジェクトの所有者を、次のように sub.dom.ain. 内の適切な主体に設定します。

    ```
    # nischown trusted.sub.dom.ain. credname.sub.dom.ain.
    ```

    マッピングファイルを編集します。次の箇所を変更します。

    ```
    (nis+:zo_owner[]cred.org_dir, "*.%s")), \
    ```

    から

    ```
    (nis+:zo_owner[]credname.sub.dom.ain., "*.%s")), \
    ```

    ここで使用しているリンクオブジェクト credname は、例として挙げています。エントリオブジェクト以外の、任意の有効なオブジェクトタイプとオブジェクト名が使用できます。オブジェクトの所有者に正しいドメイン名を設定することが重要です。
  - 主体およびネット名に使用されるドメインの主体に対して、特別な目的のオブジェクトであってもその所有権を与えたくない場合は、次に示すように nisplusPrincipalName 属性と nisplusNetname 属性を作成します。
- cred.org_dir テーブルに、複数のドメインに属している主体とネット名が設定されている場合。

LDAP サーバーのマニュアルを参照して、nisplusPrincipalName 属性および nisplusNetname 属性と、nisplusAuthName オブジェクトクラスを作成します。次のデータは ldapadd への LDIF データになっています。属性とオブジェクトクラス OID は、例として挙げているだけです。

```
dn: cn=schema
changetype: modify
add: attributetypes
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.7.0 NAME 'nisplusPrincipalName' \
          DESC 'NIS+ principal name' \
          SINGLE-VALUE \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.9.0 NAME 'nisplusNetname' \
          DESC 'Secure RPC netname' \
          SINGLE-VALUE \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

dn: cn=schema
changetype: modify
add: objectclasses
objectclasses:    ( 1.3.6.1.4.1.42.2.27.5.42.42.10.0 NAME 'nisplusAuthName' \
          SUP top AUXILLIARY DESC 'NIS+ authentication identifiers' \
          MAY ( nisplusPrincipalName $ nisplusNetname ) )
```

新しく作成した nisplusNetname 属性および nisplusPrincipalName 属性を使用するために cred.org_dir マッピングを有効にします。テンプレートマッピングファイル /var/nis/NIS+LDAPmapping.template では、この目的に対応した行がコメントになっています。credlocal、creduser、および crednode データベース ID については、nisplusObjectDN、nisplusLDAPattributeFromColumn 属性、および nisplusLDAPcolumnFromAttribute 属性の値を参照してください。マッピングファイルの編集が終了したら、NIS+ サービスを再起動します。

client_info および timezone テーブル (NIS+ から LDAP への移行)

RFC 2307 では、NIS+ の client_info.org_dir および timezone.org_dir テーブルに保存する情報のスキーマは規定していません。このため、これらのテーブルのマッピングは、テンプレートマッピングファイル (/var/nis/NIS+LDAPmapping.template) ではデフォルトで無効になっています。client_info および timezone の情報を LDAP に保存する場合は、LDAP サーバーのマニュアルを参照しながら、以降の節で説明する新しい属性とオブジェクトクラスを作成します。

client_info 属性とオブジェクトクラス

次のような属性とオブジェクトクラスを作成し、client_info データのコンテナを作成します。推奨コンテナ名は ou=ClientInfo です。LDIF データは ldapadd(1) に適用します。属性とオブジェクトクラス OID は、例として挙げています。

```
dn: cn=schema
changetype: modify
add: attributetypes
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.12.0 \
    NAME 'nisplusClientInfoAttr' \
    DESC 'NIS+ client_info table client column' \
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.12.1 \
    NAME 'nisplusClientInfoInfo' \
    DESC 'NIS+ client_info table info column' \
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.12.2 \
    NAME 'nisplusClientInfoFlags' \
    DESC 'NIS+ client_info table flags column' \
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )

dn: cn=schema
changetype: modify
add: objectclasses
objectclasses:    ( 1.3.6.1.4.1.42.2.27.5.42.42.13.0 \
    NAME 'nisplusClientInfoData' \
    DESC 'NIS+ client_info table data' \
    SUP top STRUCTURAL MUST ( cn ) \
    MAY ( nisplusClientInfoAttr $ nisplusClientInfoInfo $ nisplusClientInfoFlags ) )
```

コンテナを作成するには、次の LDIF データをファイルに入力します。実際の検索ベースを *searchBase* に代入します。

```
dn: ou=ClientInfo, searchBase

objectClass: organizationalUnit

ou: ClientInfo

objectClass: top
```

`ou=ClientInfo` コンテナを作成するために、上記のファイルを `ldapadd` コマンドの入力として使用します。たとえば、LDAP 管理者の DN が `cn=directory manager` で、LDIF データが含まれるファイルが `cifile` の場合は、次のコマンドを実行します。

```
# ldapadd -D cn="directory manager" -f cifile
```

必要な認証によっては、`ldapadd` コマンドを実行すると、パスワードプロンプトが表示されることがあります。

`/var/nis/NIS+LDAPmapping.template` ファイルでは、`client_info.org_dir` テーブルの定義はコメントになっています。これらの定義を実際のマッピングファイルにコピーし、コメント文字「`#`」を削除して定義を有効にしてから、`rpc.nisd` デーモンを再起動します。

```
# svcadm restart network/rpc/nisplus:default
```

必要に応じて、NIS+ データと LDAP データを同期化します。方法については、290 ページの「NIS+ から LDAP への移行シナリオ」を参照してください。

timezone 属性とオブジェクトクラス

次のような属性とオブジェクトクラスを作成し、タイムゾーンデータのコンテナを作成します。推奨コンテナ名は ou=Timezone です。LDIF データは ldapadd(1) に適用します。属性とオブジェクトクラス OID は、例として挙げています。

```
dn: cn=schema
changetype: modify
add: attributetypes
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.15.0 NAME 'nisplusTimeZone' \
          DESC 'tzone column from NIS+ timezone table' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )

dn: cn=schema
changetype: modify
add: objectclasses
objectclasses:    ( 1.3.6.1.4.1.42.2.27.5.42.42.16.0 NAME 'nisplusTimeZoneData' \
          DESC 'NIS+ timezone table data' \
          SUP top STRUCTURAL MUST ( cn ) \
          MAY ( nisplusTimeZone $ description ) )
```

ou=Timezone コンテナを作成するには、次の LDIF データをファイルに入力します。実際の検索ベースを *searchBase* に代入します。

```
dn: ou=Timezone,searchBase ou: Timezone objectClass: top

objectClass: organizationalUnit
```

ou=Timezone コンテナを作成するために、上記のファイルを ldapadd(1) の入力として使用します。たとえば、LDAP 管理者の DN が cn=directory manager で、LDIF データが含まれるファイルが tzfile の場合は、次のコマンドを実行します。

```
# ldapadd -D cn="directory manager" -f tzfile
```

必要な認証によっては、ldapadd コマンドを実行すると、パスワードプロンプトが表示されることがあります。

/var/nis/NIS+LDAPmapping.template ファイルでは、timezone.org_dir テーブルの定義はコメントになっています。これらの定義を実際のマッピングファイルにコピーし、コメント文字「#」を削除して定義を有効にしてから、rpc.nisd デーモンを再起動します。

```
# svcadm restart network/rpc/nisplus:default
```

必要に応じて、NIS+ データと LDAP データを同期化します。方法については、290 ページの「NIS+ から LDAP への移行シナリオ」を参照してください。

新しいオブジェクトマッピングの追加 (NIS+ から LDAP への移行)

テンプレートマッピングファイル /var/nis/NIS+LDAPmapping.template には、すべての標準 NIS+ オブジェクトのマッピング情報が含まれます。サイトまたはアプリケーション固有のオブジェクトのマッピングをサポートするには、新しいマッピングエントリを追加する必要があります。エントリ以外のオブジェクト (ディレクトリ、グループ、リンク、またはテーブル) の場合は、簡単に追加できます。しかし、エントリオブジェクトの場合、対応するエントリデータの LDAP 編成が NIS+ で使用される編成と大きく異なるときは、エントリの追加が複雑になることがあります。ここでは簡単な例を挙げます。

▼ エントリ以外のオブジェクトを対応づけるには

1. 対応づけるオブジェクトの完全指定名を検索します。

   このオブジェクト名が nisplusLDAPbaseDomain 属性で指定されるドメイン名に存在する場合は、nisplusLDAPbaseDomain 値に等しい部分は省略できます。

   たとえば、nisplusLDAPbaseDomain の値が some.domain. で、マッピング先のオブジェクトが nodeinfo.some.domain. と呼ばれるテーブルの場合、オブジェクト名は nodeinfo に短縮できます。

2. オブジェクトを識別するデータベース **ID** を作成します。

   データベース ID は、使用するマッピング構成に対して一意でなければなりません。一意でない場合は解釈されません。LDAP データには、データベース ID がありません。エントリオブジェクトのマッピングと混同しないように、テーブルエントリではなくテーブルオブジェクト自体を識別するデータベース ID を作成します。ID の末尾には、_table などのわかりやすい文字列を付加します。

   たとえば、データベース ID nodeinfo_table を使用して、データベース ID とオブジェクトの接続を標準のマッピングファイルの場所 (/var/nis/NIS+LDAPmapping) で確立するには、次のものを追加します。

   ```
   nisplusLDAPdatabaseIdMapping    nodeinfo_table:nodeinfo.some.domain.
   ```

   nisplusLDAPbaseDomain の値を some.domain. と想定します。次のものも機能します。

   ```
   nisplusLDAPdatabaseIdMapping    nodeinfo_table:nodeinfo
   ```

3. オブジェクトの **TTL** を決定します。

   TTL とは、rpc.nisd デーモンがオブジェクトのローカルコピーを有効とみなす期間のことです。TTL が期限切れになると、オブジェクトが次に参照されるときに LDAP 検索が初期化され、オブジェクトが更新されます。

2 つの TTL 値があります。1 番目の TTL は、リブートまたは再起動したあとに、rpc.nisd デーモンがディスクからオブジェクトを最初に読み込んだときに設定されます。2 番目の TTL は、LDAP から更新されたときに設定されます。1 番目の TTL は、設定した範囲からランダムに選択されます。たとえば、nodeinfo_table の生存期間を、最初に読み込まれたときには 1 - 3 時間、次回以降に読み込まれたときは 12 時間に設定する場合は、次のように指定します。

```
nisplusLDAPentryTtl        nodeinfo_table:3600:10800:43200
```

4 オブジェクトデータを **LDAP** のどこに格納するかを決定します。

テンプレートマッピングファイルでは、エントリ以外のオブジェクトの格納先が ou=nisPlus コンテナに設定されています。

この設定を使用する場合に、適切な属性、オブジェクトクラス、およびコンテナをまだ作成していないときは、300 ページの「テーブルエントリ以外の NIS+ オブジェクトのマッピング」を参照してください。

たとえば、nodeinfo オブジェクトを ou=nisPlus,dc=some,dc=domain コンテナに格納し、その LDAP エントリを cn nodeinfo にするとします。次の nisplusLDAPobjectDN を作成してください。

```
nisplusLDAPobjectDN    nodeinfo_table:\
                cn=nodeinfo,ou=nisPlus,dc=some,dc=domain?base?\
                objectClass=nisplusObjectContainer:\
                cn=nodeinfo,ou=nisPlus,dc=some,dc=domain?base?\
                    objectClass=nisplusObjectContainer,\
                    objectClass=top
```

NIS+ 複製は LDAP にデータを書き込まないため、この nisplusLDAPobjectDN はマスターおよび複製の両方に対して使用できます。

5 **(**マッピング先の **NIS+** オブジェクトがまだ **NIS+** に作成されていない場合は、この手順を省略できます。**)**オブジェクトデータを **LDAP** に格納します。この操作には、rpc.nisd デーモンを使用できます。ただし、nisldapmaptest**(1M)** ユーティリティーを使用すると rpc.nisd デーモンを停止する必要がないので、この操作をより簡単に行うことができます。

```
# nisldapmaptest -m /var/nis/NIS+LDAPmapping -o -t nodeinfo -r
```

-o オプションには、テーブルエントリではなく、テーブルオブジェクト自体を指定します。

6 オブジェクトデータが **LDAP** に格納されたことを確認します。この例では、**LDAP** サーバーがローカルマシンのポート 389 で動作していることを前提としています。

```
# ldapsearch -b ou=nisPlus,dc=some,dc=domain cn=nodeinfo
```

出力は次のようになります。

```
dn: cn=nodeinfo,ou=nisPlus,dc=some,dc=domain
objectclass: nisplusObjectContainer
objectclass: top
```

```
cn: nodeinfo
nisplusobject=<base 64 encoded data>
```

エントリオブジェクトの追加

NIS+LDAPmapping(4) には、テーブルエントリマッピングの構文および意味論が詳細に指定されています。また、構文要素ごとの使用例も提供されています。ただし、多くの場合、既存のマッピングから目的のマッピングに近いものを選択し、そのマッピングをコピーして変更すれば、最も簡単に行うことができ、エラーも少なくなります。

たとえば、ノードの資産情報と所有者情報を格納する nodeinfo という NIS+ テーブルを想定します。NIS+ テーブルは、次のコマンドを使って作成されたとします。

```
# nistbladm -c -D access=og=rmcd,nw=r -s : nodeinfo_tbl \
cname=S inventory=S owner= nodeinfo.‘domainname‘.
```

cname 列には、ノードの正式名が格納されます。つまり、ノードの hosts.org_dir テーブルの cname 列と同じ値が格納されます。

また、対応する情報が LDAP の ou=Hosts コンテナに格納され、nodeInfo オブジェクトクラス (この例のための仮想クラスで、RFC では定義されていません) の MUST 属性が cn で、MAY 属性が nodeInventory と nodeOwner であるとします。

既存の nodeinfo データを LDAP にアップロードするときは、別のファイルに新しいマッピング属性を作成すれば、簡単に行うことができます。たとえば、/var/nis/tmpmapping を使用します。

1. マッピング先の NIS+ テーブルを識別するデータベース ID を作成します。

   ```
   nisplusLDAPdatabaseIdMapping    nodeinfo:nodeinfo
   ```

2. nodeinfo テーブルのエントリに TTL を設定します。この情報はほとんど変更されないため、TTL を 12 時間に設定します。rpc.nisd デーモンがディスクから nodeinfo テーブルを最初に読み取ると、テーブルエントリの TTL が 6 - 12 時間からランダムに選択されます。

   ```
   nisplusLDAPentryTtl        nodeinfo:21600:43200:43200
   ```

3. 既存のマッピングから、作成するマッピングに似ているものを選択します。この例では、属性値の割り当ては簡単で、直接割り当てるだけです。ただし、既存のコンテナに LDAP データを格納する処理が複雑です。このため、nodeinfo データの削除は、慎重に行う必要があります。ou=Hosts エントリ全体を削除せずに、nodeInventory および nodeOwner 属性だけを削除します。このため、特別の削除ルールが必要になります。

   つまり、コンテナを共有し削除ルールを持つマッピングを探します。この候補として、netmasks マッピングがあります。このマッピングは、ou=Networks コンテナを共有し、削除ルールを持っています。

4. /var/nis/NIS+LDAPmapping.template の netmasks テンプレートマッピングでは、次のマッピングがデフォルトになっています。

```
nisplusLDAPobjectDN    netmasks:ou=Networks,?one?objectClass=ipNetwork,\
          ipNetMaskNumber=*:\
             ou=Networks,?one?objectClass=ipNetwork:
             dbid=netmasks_del
```

このテンプレートマッピングを nodeinfo の新しいマッピングにコピーし、データベース ID を nodeinfo、コンテナを ou=Hosts、オブジェクトクラスを nodeInfo に変更します。つまり、nodeinfo マッピングの最初の行は、次のようになります。

```
nisplusLDAPobjectDN    nodeinfo:ou=Hosts,?one?objectClass=nodeInfo,\
```

netmasks マッピングの 2 行目は、検索フィルタ部分になっています。ipNetMaskNumber 属性を含む ou=Networks エントリだけを選択します。この例では、次の nodeInventory 属性を持つ ou=Hosts エントリを選択します。

```
nodeInventory=*:\
```

3、4 行目は nisplusLDAPobjectDN の書き込み部分になっています。LDAP nodeinfo データの書き込み先と、nodeinfo データを削除するときのルールが指定されています。ここでは、データベース ID が nodeinfo_del の削除ルールを作成します。ou=Hosts の既存のエントリに常に書き込むため、次のように nodeinfo データ自体のオブジェクトクラスを指定するだけです。

```
ou=Hosts,?one?objectClass=nodeInfo:\
                 dbid=nodeinfo_del
```

この結果、nisplusLDAPobjectDN は次のようになります。

```
nisplusLDAPobjectDN    nodeinfo:ou=Hosts,?one?objectClass=nodeInfo,\
              nodeInventory=*:\
                  ou=Hosts,?one?objectClass=nodeInfo:\
                  dbid=nodeinfo_del
```

5. nodeinfo データを NIS+ から LDAP に対応づけるマッピングルールを作成します。netmasks を使用するテンプレートは、次のようになります。

```
nisplusLDAPattributeFromColumn \
        netmasks:    dn=("ipNetworkNumber=%s,", addr), \
                        ipNetworkNumber=addr, \
                        ipNetmaskNumber=mask, \
                        description=comment
```

ここでは、ou=Hosts コンテナはより複雑な構成になります。RFC 2307 の規定では、dn に IP アドレスを含める必要があるためです。しかし、IP アドレスは nodeinfo テーブルに格納されないため、別の方法で取得する必要があります。テンプレートファイルの crednode マッピングには、IP アドレスの取得方法が記述されています。

```
nisplusLDAPattributeFromColumn \
   crednode:    dn=("cn=%s+ipHostNumber=%s,", \
                                (cname, "%s.*"), \
   ldap:ipHostNumber:?one?("cn=%s", (cname, "%s.*"))), \
```

crednode マッピングの部分をコピーできます。ただし、ここでは、cname 列値は主体名ではなく実際のホスト名です。cname の一部を抽出する必要はありません。属性および列名を明示的に代入します。nodeinfo マッピングは次のようになります。

```
nisplusLDAPattributeFromColumn \
   nodeinfo:    dn=("cn=%s+ipHostNumber=%s,", cname, \
   ldap:ipHostNumber:?one?("cn=%s", cname)), \
           nodeInventory=inventory, \
           nodeOwner=owner
```

6. LDAP のデータを NIS+ にマッピングするときは、netmasks エントリのテンプレートは次のようになります。

```
nisplusLDAPcolumnFromAttribute \
        netmasks:    addr=ipNetworkNumber, \
                mask=ipNetmaskNumber, \
                comment=description
```

属性および列名を代入すると、次のようになります。

```
nisplusLDAPcolumnFromAttribute \
        nodeinfo:    cname=cn, \
                inventory=nodeInventory, \
                owner=nodeOwner
```

7. netmasks の削除ルールは、次のようになっています。

```
nisplusLDAPattributeFromColumn \
        netmasks_del:    dn=("ipNetworkNumber=%s,", addr), \
                ipNetmaskNumber=
```

この例では、NIS+ の netmasks エントリが削除されると、対応する ou=Networks LDAP エントリの ipNetmaskNumber 属性が削除されます。ここでは、nodeInventory および nodeOwner 属性を削除します。つまり、手順 (5) の dn 指定を使用して、次のように編集します。

```
nisplusLDAPattributeFromColumn \
        nodeinfo_del:    dn=("cn=%s+ipHostNumber=%s,", cname, \
            ldap:ipHostNumber:?one?("cn=%s", cname)), \
                nodeInventory=, \
                nodeOwner=
```

マッピング情報はこれで完了です。

8. NIS+ nodeinfo テーブルにすでにデータが存在する場合は、そのデータを LDAP にアップロードします。新しい nodeinfo マッピング情報を、別のファイル /var/nis/tmpmapping に格納します。

```
# /usr/sbin/rpc.nisd -D -m /var/nis/tmpmapping \
-x nisplusLDAPinitialUpdateAction=to_ldap \
-x nisplusLDAPinitialUpdateOnly=yes
```

9. 一時ファイル /var/nis/tmpmapping のマッピング情報を実際のマッピングファイルに追加します。エディタを使用するか、次の方法でデータを追加します。実際のマッピングファイルは、/var/nis/NIS+LDAPmapping とします。

```
# cp -p /var/nis/NIS+LDAPmapping \
/var/nis/NIS+LDAPmapping.backup

# cat /var/nis/tmpmapping >> /var/nis/NIS+LDAPmapping
```

注意 – リダイレクトに二重矢印「>>」を使っている点に注意してください。矢印「>」を使った場合は対象ファイルを上書きします。

構成情報を LDAP に格納する

NIS+ および LDAP の構成情報は、構成ファイルとコマンド行で格納できますが、構成属性は LDAP にも格納できます。構成情報が多くの NIS+ サーバーによって共有され、定期的に変更される場合は、LDAP に格納すると便利です。

構成属性を LDAP に格納するには、LDAP サーバーのマニュアルを参照して、次の新しい属性とオブジェクトクラスを作成します。構成情報は、`rpc.nisd` コマンド行または `/lib/svc/method/nisplus` の `nisplusLDAPconfigDN` 値に指定された場所に存在することが前提となっています。また、`cn` が `nisplusLDAPbaseDomain` 値であることも前提です (LDAP から構成情報を読み取る前に `rpc.nisd` デーモンに認識されるため)。

LDIF データは、`ldapadd(1)` に適用できます。属性とオブジェクトクラス OID は、例として挙げています。

`defaultSearchBase`、`preferredServerList`、および `authenticationMethod` 属性は、「DUA config」スキーマの原案に準拠しています。このスキーマは、IETF 標準となる見込みです。`NIS+LDAPmapping(4)` で使用する場合は、次の定義で十分です。

```
dn: cn=schema
changetype: modify
add: attributetypes
attributetypes:    ( 1.3.6.1.4.1.11.1.3.1.1.1 NAME 'defaultSearchBase' \
          DESC 'Default LDAP base DN used by a DUA' \
          EQUALITY distinguishedNameMatch \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.11.1.3.1.1.2 NAME 'preferredServerList' \
          DESC 'Preferred LDAP server host addresses to be used by a DUA' \
          EQUALITY caseIgnoreMatch \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.11.1.3.1.1.6 NAME 'authenticationMethod' \
          DESC 'Identifies the authentication method used to connect to the DSA'\
          EQUALITY caseIgnoreMatch \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
```

NIS+ および LDAP の構成属性は、次のようになっています。

```
dn: cn=schema
changetype: modify
add: attributetypes
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.0 \
```

```
          NAME 'nisplusLDAPTLS' \
          DESC 'Transport Layer Security' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.1 \
          NAME 'nisplusLDAPTLSCertificateDBPath' \
          DESC 'Certificate file' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.2 \
          NAME 'nisplusLDAPproxyUser' \
          DESC 'Proxy user for data store/retrieval' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.3 \
          NAME 'nisplusLDAPproxyPassword' \
          DESC 'Password/key/shared secret for proxy user' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.4 \
          NAME 'nisplusLDAPinitialUpdateAction' \
          DESC 'Type of initial update' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.5 \
          NAME 'nisplusLDAPinitialUpdateOnly' \
          DESC 'Exit after update ?' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.6 \
          NAME 'nisplusLDAPretrieveErrorAction' \
          DESC 'Action following an LDAP search error' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.7 \
          NAME 'nisplusLDAPretrieveErrorAttempts' \
          DESC 'Number of times to retry an LDAP search' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.8 \
          NAME 'nisplusLDAPretrieveErrorTimeout' \
          DESC 'Timeout between each search attempt' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.9 \
          NAME 'nisplusLDAPstoreErrorAction' \
          DESC 'Action following an LDAP store error' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.10 \
          NAME 'nisplusLDAPstoreErrorAttempts' \
          DESC 'Number of times to retry an LDAP store' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.11 \
          NAME 'nisplusLDAPstoreErrorTimeout' \
          DESC 'Timeout between each store attempt' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.12 \
          NAME 'nisplusLDAPrefreshErrorAction' \
          DESC 'Action when refresh of NIS+ data from LDAP fails' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.13 \
          NAME 'nisplusLDAPrefreshErrorAttempts' \
          DESC 'Number of times to retry an LDAP refresh' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.14 \
          NAME 'nisplusLDAPrefreshErrorTimeout' \
          DESC 'Timeout between each refresh attempt' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

```
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.15 \
          NAME 'nisplusNumberOfServiceThreads' \
          DESC 'Max number of RPC service threads' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.16 \
          NAME 'nisplusThreadCreationErrorAction' \
          DESC 'Action when a non-RPC-service thread creation fails' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.17 \
          NAME 'nisplusThreadCreationErrorAttempts' \
          DESC 'Number of times to retry thread creation' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.18 \
          NAME 'nisplusThreadCreationErrorTimeout' \
          DESC 'Timeout between each thread creation attempt' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.19 \
          NAME 'nisplusDumpErrorAction' \
          DESC 'Action when an NIS+ dump fails' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.20 \
          NAME 'nisplusDumpErrorAttempts' \
          DESC 'Number of times to retry a failed dump' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.21 \
          NAME 'nisplusDumpErrorTimeout' \
          DESC 'Timeout between each dump attempt' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.22 \
          NAME 'nisplusResyncService' \
          DESC 'Service provided during a resync' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.23 \
          NAME 'nisplusUpdateBatching' \
          DESC 'Method for batching updates on master' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.24 \
          NAME 'nisplusUpdateBatchingTimeout' \
          DESC 'Minimum time to wait before pinging replicas' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.25 \
          NAME 'nisplusLDAPmatchFetchAction' \
          DESC 'Should pre-fetch be done ?' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.26 \
          NAME 'nisplusLDAPbaseDomain' \
          DESC 'Default domain name used in NIS+/LDAP mapping' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.27 \
          NAME 'nisplusLDAPdatabaseIdMapping' \
          DESC 'Defines a database id for an NIS+ object' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.28 \
          NAME 'nisplusLDAPentryTtl' \
          DESC 'TTL for cached objects derived from LDAP' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.29 \
          NAME 'nisplusLDAPobjectDN' \
          DESC 'Location in LDAP tree where NIS+ data is stored' \
```

```
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.30 \
          NAME 'nisplusLDAPcolumnFromAttribute' \
          DESC 'Rules for mapping LDAP attributes to NIS+ columns' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
attributetypes:    ( 1.3.6.1.4.1.42.2.27.5.42.42.18.31 \
          NAME 'nisplusLDAPattributeFromColumn' \
          DESC 'Rules for mapping NIS+ columns to LDAP attributes' \
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

dn: cn=schema
changetype: modify
add: objectclasses
objectclasses:    ( 1.3.6.1.4.1.42.2.27.5.42.42.19.0 NAME 'nisplusLDAPconfig' \
          DESC 'NIS+/LDAP mapping configuration' \
          SUP top STRUCTURAL MUST ( cn ) \
          MAY ( preferredServerList $ defaultSearchBase $
authenticationMethod $ nisplusLDAPTLS $ nisplusLDAPTLSCertificateDBPate
$ nisplusLDAPproxyUser $ nisplusLDAPproxyPassword $ nisplusLDAPinitialUpdateAction
$ nisplusLDAPinitialUpdateOnly $ nisplusLDAPretrieveErrorAction
$ nisplusLDAPretrieveErrorAttempts $ nisplusLDAPretrieveErrorTimeout
$ nisplusLDAPstoreErrorAction $ nisplusLDAPstoreErrorAttempts
$ nisplusLDAPstoreErrorTimeout $ nisplusLDAPrefreshErrorAction
$ nisplusLDAPrefreshErrorAttempts $ nisplusLDAPrefreshErrorTimeout
$ nisplusNumberOfServiceThreads $nisplusThreadCreationErrorAction
$ nisplusThreadCreationErrorAttempts $ nisplusThreadCreationErrorTimeout
$ nisplusDumpErrorAction $ nisplusDumpErrorAttempts
$ nisplusDumpErrorTimeout $ nisplusResyncService $ nisplusUpdateBatching
$ nisplusUpdateBatchingTimeout $ nisplusLDAPmatchFetchAction
$ nisplusLDAPbaseDomain $ nisplusLDAPdatabaseIdMapping $ nisplusLDAPentryTtl
$ nisplusLDAPobjectDN $ nisplusLDAPcolumnFromAttribute !
$ nisplusLDAPattributeFromColumn ) )
```

次の LDIF データを含むファイルを作成します。実際の検索ベースを *searchBase* に、完全指定ドメイン名を *domain* に代入します。

```
dn: cn=domain, searchBase

cn: domain

objectClass: top objectClass: nisplusLDAPconfig
```

上のファイルを `ldapadd(1)` の入力として使用し、NIS+ および LDAP の構成エントリを作成します。最初は、エントリは空になっています。`ldapmodify(1)` を使用して、構成属性を追加します。たとえば、`nisplusNumberOfServiceThreads` 属性に「32」を設定するには、`ldapmodify(1)` の入力として次のファイルを作成します。

```
dn: cn=domain, searchBase nisplusNumberOfServiceThreads: 32
```

◆ ◆ ◆ 付 録 A

Solaris 10 ソフトウェアの DNS、NIS、および LDAP の更新

『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の Solaris 10 バージョンには、DNS BIND および pam_ldap の更新が含まれています。その他の内容に関する細部の変更や追加、および記述上の誤りの訂正も行われています。

サービス管理機能の変更点

DNS、NIS、LDAP のサービスは、サービス管理機能によって管理されます。これらのサービスに関する有効化、無効化、再起動などの管理アクションは、svcadm コマンドを使用して実行できます。サービスの状態は、svcs コマンドを使用して照会できます。SMF の概要については、『Solaris のシステム管理 (基本編)』の第 18 章「サービスの管理 (概要)」を参照してください。また、詳細については、svcadm(1M) および svcs(1) のマニュアルページを参照してください。

各サービス固有の情報は、このマニュアルの次の節で説明されています。

- 53 ページの「DNS とサービス管理機能」
- 86 ページの「NIS とサービス管理機能」
- 200 ページの「LDAP とサービス管理機能」
- 252 ページの「NIS から LDAP への移行用ツールとサービス管理機能」
- 279 ページの「NIS+ から LDAP への移行用ツールとサービス管理機能」

NIS+ およびサービス管理機能については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』を参照してください。

DNS BIND

Solaris 10 リリースには、BIND 8.4.2 が同梱されています。このバージョンの BIND は、Solaris ソフトウェアでの IPv6 ネットワークに対する完全な DNS クライアントサーバーのソリューションを提供します。このマニュアルでは、DNS BIND の手順に関する変更はありません。

BIND 9 も Solaris 10 リリースでサポートされ、/usr/sfw ディレクトリにインストールされます。移行に関するマニュアルは /usr/sfw/doc/bind ディレクトリにあります。移行のマニュアルで規定されている点以外は、パート II「DNS の設定と管理」の情報と手順が BIND 9 にも適用されます。

pam_ldap の変更点

Solaris 10 OS リリースでは、pam_ldap が次に示すように変更されました。詳細については、pam_ldap(5) のマニュアルページを参照してください。

- 以前サポートされていた use_first_pass および try_first_pass のオプションは、Solaris 10 ソフトウェアリリースでサポートされなくなりました。このオプションは不要のため pam.conf から削除しても問題はなく、そのままにしておいても構いません。将来のリリースで削除されることもあります。
- パスワードプロンプトに対応する必要があります。これは、認証およびパスワードモジュールのスタックで pam_ldap の前に pam_authtok_get をスタックし、passwd サービスの auth スタックに pam_passwd_auth を含めることによって行います。
- 以前サポートされていたパスワード更新機能は、以前使用が推奨されていた、pam_authtok_store と server_policy オプションにこのリリースで置き換わります。
- pam_ldap のアカウント管理機能は、LDAP ネームサービスのセキュリティー全般を強化します。特に、アカウント管理機能により次のようなことが行われます。
  - 古いパスワードや、期限切れのパスワードを追跡できます
  - ありふれたパスワードや、以前使ったことのあるパスワードをユーザーが選択できないようにします。
  - パスワードの期限が切れそうなユーザーに警告を出します。
  - 続けてログインに失敗したユーザーをロックします。
  - 許可されたシステム管理者以外のユーザーが、初期化されたアカウントを無効にできないようにします。

上記の変更に対して完全な自動更新は行うことができません。すなわち、Solaris 10 以降のリリースにアップグレードしても、既存の pam.conf ファイルは自動的に更新されず、pam_ldap の変更は反映されません。既存の pam.conf ファイルに pam_ldap の

構成が含まれる場合は、アップグレード後の CLEANUP ファイルによって確認できます。pam.conf ファイルを調べて、必要に応じて変更してください。

詳細については、pam_passwd_auth(5)、pam_authtok_get(5)、pam_authtok_store(5)、および pam.conf(4) のマニュアルページを参照してください。

記述の誤りの訂正

このリリースでは、いくつかの記述上の誤りが訂正されています。

用語集

attribute 各 LDAP エントリは多数の名前付き属性で構成される。各属性は 1 つ以上の値を保持する。

baseDN DIT の一部のベースとなる DN。NIS ドメインエントリの baseDN の場合、「コンテキスト」とも呼ぶ。

DBM NIS マップを格納するために当初使用されるデータベース。

DES 「データ暗号化規格 (DES)」の項を参照。

DIT 「ディレクトリ情報ツリー」の項を参照。

DN LDAP 内の識別名。ツリー構造を持つ LDAP ディレクトリのアドレススキーマ。各 LDAP エントリに一意の名前を付与する。

DNS 「ドメインネームシステム」の項を参照。

DNS ゾーン ネットワークドメイン内の管理境界であり、通常 1 つまたは複数のサブドメインから構成される。

DNS ゾーンファイル DNS ソフトウェアがドメイン内の全ワークステーションの名前と IP アドレスを格納する一連のファイル。

DNS 転送 NIS サーバーまたは NIS 互換設定の NIS+ サーバーは、自分で応答できない要求を DNS サーバーに転送する。

GID 「グループ ID」の項を参照。

IP インターネットプロトコル。インターネットプロトコル体系の「ネットワーク層」プロトコル。

IP アドレス ネットワーク内の各ホストを識別する一意な番号。

LDAP Lightweight Directory Access Protocol。LDAP ネームサービスクライアントとサーバー間の通信に使用される標準の拡張可能なディレクトリアクセスプロトコル。

MIS 経営情報システムまたはサービス。

N2L サーバー NIS-to-LDAP サーバー。N2L サービスを使用して、N2L サーバーとして再構成された NIS マスターサーバー。再構成には、NIS デーモンの置き換えと新しい構成ファイルの追加が含まれる。

NDBM DBM の改良されたバージョン。

NIS	ネットワーク上のシステムとユーザーについての重要な情報が収められている分散型ネットワーク情報サービス。NIS データベースは、「マスターサーバー」とすべての「スレーブサーバー」に格納されている。
NIS+	ネットワーク上のシステムとユーザーについての階層情報が収められている分散型ネットワーク情報サービス。NIS+ データベースは、「マスターサーバー」とすべての「複製サーバー」に格納されている。
NIS 互換モード	NIS+ の構成の 1 つであり、このモードでは NIS クライアントは NIS+ テーブルに格納されたデータにアクセスできる。また、NIS+ サーバーは NIS クライアントと NIS+ クライアントからの情報要求に応答できる。
NIS マップ	NIS によって使用されるファイルであり、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マシンの名前など特定種類の情報を格納する。NIS サービスの一部であるプログラムはこれらのマップを参照する。「NIS」の項も参照。
RDN	相対識別名。DN の一部。
RFC 2307	標準の NIS マップから DIT エントリへの情報のマッピングを指定した RFC。
RPC	遠隔手続き呼び出し (RPC)の項を参照。
SASL	簡単な認証およびセキュリティー層 (Simple Authentication and Security Layer)。アプリケーション層プロトコルにおける認証およびセキュリティー層の意味上の取り決め。
searchTriple	特定の属性を DIT 内のどこで検索するかを示す記述。searchTriple は、「ベース DN」、「スコープ」、および「フィルタ」で構成される。これは、RFC 2255 で定義された LDAP URL 形式の一部である。
Secure RPC パスワード	Secure RPC プロトコルによって要求されるパスワード。非公開鍵の暗号化に使用される。このパスワードはユーザーのログインパスワードと同じでなければならない。
SSL	Secure Sockets Layer プロトコル。LDAP セキュアなどのアプリケーションプロトコルを作成するためのトランスポート層のセキュリティー機構の総称。
TCP	「Transport Control Protocol (TCP)」の項を参照。
TCP/IP	Transport Control Protocol/Interface Program の頭字語を使った略語。このプロトコル群は、最初はインターネット用に開発された。「インターネット」プロトコル群とも呼ばれる。Solaris ネットワークは、デフォルトでは TCP/IP 上で動作する。
Transport Control Protocol (TCP)	インターネットプロトコル群での主要なトランスポートプロトコルであり、高信頼性でコネクション型の全二重ストリームを提供する。配信には IP を使用する。「TCP/IP」の項を参照。
Transport Layer Security (TLS)	LDAP クライアントとディレクトリサーバーとの通信を保護して、機密性とデータ整合性を確保する。TLS プロトコルは、Secure Sockets Layer (SSL) プロトコルのスーパーセットである。
X.500	開放型システム間相互接続 (OSI) 規格で定義されたグローバルディレクトリサービス。LDAP の前身。

yp	Yellow Pages。NIS コード内部で今も使用される NIS の古い名前。
アプリケーションレベルのネームサービス	ファイル、メール、印刷などのサービスを提供するアプリケーションに組み込まれているネームサービスのこと。アプリケーションレベルのネームサービスは、企業レベルのネームサービスの下に位置する。企業レベルのネームサービスが提供するコンテキストの中に、アプリケーションレベルのネームサービスのコンテキストを組み込むことができる。
暗号化	データの機密性を保護するための手段。
暗号化鍵	「データ暗号化鍵」の項を参照。
インターネットアドレス	「TCP/IP」を使用するホストに割り当てられた 32 ビットのアドレス。「ドット表記」の項を参照。
インデックス付き名前	テーブル内のエントリの識別に使用されるネーミング形式。
遠隔手続き呼び出し **(RPC)**	分散コンピューティングのクライアントサーバーモデルを実現する簡単で一般的なパラダイム。与えられた引数を使用することによって、要求が遠隔システムに送信され、指定された手順が実行される。そのあと、その結果が呼び出し側に返される。
エントリ	データベーステーブル内の一列のデータのこと (DIT 内の LDAP 要素など)。
親ドメイン	「ドメイン」の項を参照。
鍵 (暗号化)	鍵の管理および配布システムの一部として、ほかの鍵の暗号化と復号化に使用される鍵。「データ暗号化鍵」の項も参照。
キーサーバー	非公開鍵を格納する Solaris オペレーティング環境のプロセス。
企業レベルのネットワーク	「企業レベル」のネットワークといっても、ケーブルや赤外線ビーム、無線などを利用した単一の LAN (Local Area Network) の場合もあれば、ケーブルや直通電話接続を利用して複数の LAN を結んだクラスタもある。企業レベルのネットワーク内では、DNS や X.500/LDAP などのグローバルネームサービスを使用せずに、どのマシンからでも任意のマシンにアクセスできる。
逆解決	DNS ソフトウェアを使用して、ワークステーションの IP アドレスをワークステーション名に変換するプロセス。
キャッシュマネージャー	NIS+ クライアントのローカルキャッシュ (`NIS_SHARED_DIRCACHE`) を管理するプログラム。これらのクライアントによって最も頻繁に使用されるディレクトリをサポートする NIS+ サーバーについての位置情報 (トランスポートアドレス、認証情報、生存期間など) を格納するために使用される。
クライアント	(1) クライアントとは、ネームサーバーに対してネームサービスを要求する主体 (マシンまたはユーザー)。 (2) ファイルシステムのクライアントサーバーモデルでは、クライアントとは、計算パワーや大きな記憶容量などの計算サーバーのリソースに遠隔アクセスするマシン。

(3) クライアントサーバーモデルでは、「サーバープロセス」からサービスにアクセスする「アプリケーション」がクライアント。このモデルでは、クライアントとサーバーは同じマシン上または別のマシン上で動作可能。

クライアントサーバーモデル
: ネットワークサービスおよびこれらのモデルユーザープロセス (プログラム) を説明する一般的な方法の 1 つ。たとえば、「ドメインネームシステム (DNS)」のネームサーバー/ネームリゾルバパラダイムなど。*client* も参照してください。

グループ **ID**
: ユーザーのデフォルト「グループ」を識別する番号。

グローバルネームサービス
: 電話回線、衛星回線、その他の通信システムにより連結された世界中の企業レベルネットワークの名前を管理するサービスのこと。この世界中に相互接続されたネットワークの集合体がいわゆる「インターネット」である。グローバルネームサービスでは、ネットワーク名だけでなく、任意のネットワーク内の個々のマシンやユーザーも識別できる。

広域ネットワーク **(WAN)**
: 地理的に離れた複数のローカルエリアネットワーク (Local-Area Network、LAN) またはシステムを、電話回線、光ファイバ、衛星などを使用して接続したネットワークのこと。

公開鍵
: 数学的に生成された 1 対の番号の公開構成要素であり、非公開鍵と組み合わせれば DES 鍵が生成される。この DES 鍵を使用すれば、情報の暗号化と復号化を行える。公開鍵は、すべてのユーザーとマシンが使用できる。どのユーザーやマシンにも、固有の公開鍵と非公開鍵が 1 対ある。

子ドメイン
: 「ドメイン」の項を参照。

サーバー
: (1) NIS+ 、NIS 、DNS 、LDAP では、ネットワークに NIS+ サービスを提供するホストマシンのこと。

(2) ファイルシステムの「クライアントサーバーモデル」では、サーバーとは計算資源 (計算サーバーとも呼ばれる) と大きな記憶容量を備えたマシン。クライアントマシンは遠隔アクセスが可能であり、これらのリソースを使用できる。ウィンドウシステムのクライアントサーバーモデルでは、サーバーとはアプリケーションまたは「クライアントプロセス」にウィンドウサービスを提供するプロセス。このモデルでは、クライアントとサーバーは同じマシン上または別のマシン上で動作可能。

(3) ファイルの提供を実際に処理する「デーモン」。

サーバーリスト
: 「優先サーバーリスト」の項を参照。

サブネット
: 経路指定を簡単にするため、1 つの論理ネットワークを小さな物理ネットワークに分割する方式。

資格
: クライアントソフトウェアが各要求とともにネームサーバーに送信する認証情報。この情報によって、ユーザーまたはマシンの ID が検査される。

識別名
: X.500 ディレクトリ情報ベース (DIB) のエントリ。ルートから指定エントリまでのパスに沿ったツリーの各エントリから選択した属性で構成される。

スキーマ
: 特定の LDAP DIT に格納可能なデータの種類を定義したルールのセット。

スレーブサーバー　ネットワーク情報サービス (NIS) データベースのコピーを管理するサーバーシステム。このシステムには、ディスクと動作環境の完全なコピーが存在する。

(2) スレーブサーバーは、NIS+ では「複製サーバー」と呼ばれる。

接尾辞　LDAP では、DIT の識別名 (DN)。

ソース　NIS ソースファイル

ディレクトリ　(1) LDAP ディレクトリは LDAP オブジェクトのコンテナのこと。(2) UNIX では、ファイルまたはサブディレクトリのコンテナのこと。

ディレクトリキャッシュ　ディレクトリオブジェクトに関連付けられたデータの格納に使用されるローカルファイル。

ディレクトリ情報ツリー　ある特定のネットワークの分散型ディレクトリ構造のこと。Solaris LDAP クライアントはデフォルトで、DIT がある特定の構造を持っていると想定して情報にアクセスします。LDAP サーバーがサポートするドメインごとに、想定された構造を持つ想定されたサブツリーがある。

データ暗号化鍵　暗号化を行うプログラムに使用されるデータを暗号化および復号化するための鍵。「鍵 (暗号化)」の項も参照。

データ暗号化規格 **(DES)**　アメリカ商務省標準局によって開発された、データの暗号化と復号化のために一般的に使用される高度なアルゴリズム。「SUN-DES-1」の項も参照。

テーブル　NIS+ においては、NIS+ データを行および列の中に持つ 2 次元的な (リレーショナルでない) データベースオブジェクトのこと (NIS における「NIS マップ」は、「列を 2 つ持つ NIS+ テーブルに似ている)。NIS+ データは、テーブルの形で保存される。NIS+ では定義済み (システム) テーブルが 16 個提供される。保存される情報のタイプはテーブルごとに異なる。

ドット形式の **10** 進表記　32 ビット整数用の構文表現であり、10 進表記された 4 つの 8 ビット数が小数点 (ドット) で区切って表現される。192.67.67.20 のように、インターネットでの IP アドレスを表現するために使用される。

ドメイン　(1) NIS+ では、NIS+ によって管理されるオブジェクト (階層構造になっている) のグループ。最上位のドメイン (ルートドメイン) 1 つと、サブドメイン 0 個以上からなる。ドメインおよびサブドメインは、地理的、組織的、機能的な基準によって編成される。

- 「親ドメイン」階層構造の中で、現在のドメインのすぐ上のドメインを表す相対的な名称。
- 「子ドメイン」階層構造の中で、現在のドメインのすぐ下のドメインを表す相対的な名称。
- 「ルートドメイン」現在の NIS+ 階層の最上位のドメイン。

(1) インターネットではネーミング階層の一部で、通常、Local Area Network (LAN)、Wide Area Network (WAN)、またはその一部に相当する。構文上、インターネットドメイン名は小数点 (ドット) によって区切られた一連の名前 (ラベル) から構成される。たとえば、`sales.doc.com`。

	(2) ISO の開放型システム間相互接続 (OSI) では、「ドメイン」は、MHS プライベート管理ドメイン (PRMD) やディレクトリ管理ドメイン (DMD) などのように、複雑な分散システムの管理パーティションとして使用されるのが普通。
ドメインネームサービス **(DNS)**	インターネットで使用されるネットワーク情報サービスのこと。すなわち DNS は、ドメイン名とマシン名をインターネットなどの企業外部のアドレスにマッピングする場合のネーミングポリシーとメカニズムを提供する。
ドメイン名	ローカルネットワーク上のシステムグループに割り当てられた名前であり、DNS 管理ファイルを共有する。ネットワーク情報サービスのデータベースが正常に動作するためにはドメイン名が必要。「ドメイン」の項を参照。
名前解決	ワークステーションやユーザーの名前をアドレスに変換するプロセス。
名前空間	(1) 名前空間はユーザー、ワークステーション、およびアプリケーションがネットワーク上で必ず必要とする情報を格納している。
	(2) ネーミングシステムで使用される名前セット。
認証	サーバーがクライアントの ID を確認するための手段。
ネームサーバー	1 つ以上のネットワークネームサービスを実行するサーバー。
ネームサービス	マシン、ユーザー、プリンタ、ドメイン、ルーターなどの、ネットワーク上の名前とアドレスを管理するネットワークサービスのこと。
ネームサービススイッチ	ネームサービスクライアントがそのネットワーク情報を獲得できるソースを定義する構成ファイル (`/etc/nsswitch.conf`)。
ネットワークパスワード	「Secure RPC パスワード」の項を参照。
ネットワークマスク	ソフトウェアが、ローカルサブネットアドレスをそれ以外のインターネットプロトコルアドレスから分離するために使用する番号。
非公開鍵	数学的に生成された 1 対の番号の非公開構成要素であり、公開鍵と組み合わせれば DES 鍵が生成される。この DES 鍵を使用すれば、情報の暗号化と復号化を行える。送信側の非公開鍵は、その鍵の所有者だけが使用できる。どのユーザーやマシンにも、固有の公開鍵と非公開鍵が 1 対ある。
フィールド	NIS マップエントリは、多数の構成要素と区切り文字で構成される場合がある。
マスターサーバー	ドメイン内の NIS データベースのマスターコピーを保持しているサーバーのこと。名前空間に対する変更は、必ずマスターサーバーのネームサービスデータベース上で行う。ドメイン中に複数のマスターサーバーを作成できない。
マッピング	DIT エントリとの間の NIS エントリの変換プロセス。この処理は、「マッピング」ファイルにより制御される。
メール交換レコード	DNS ドメイン名、およびこれらに対応するメールホストのリストが収められているファイル。
メールホスト	サイトの電子メールのルーターおよび受信側として機能するワークステーション。

優先サーバーリスト	`client_info` テーブルまたは `client_info` ファイルのこと。優先サーバーリストには、あるクライアントマシンまたはドメインから見た優先サーバーが指定される。
ルートドメイン	「ドメイン」の項を参照。
レコード	「エントリ」の項を参照。
ローカルエリアネットワーク **(LAN)**	1 つの地理的なサイトの中にある複数のシステムをデータやソフトウェアの共有や交換の目的で接続したもの。

索引

N

O

P

あ

い