管理ガイド

Sun™ ONE Directory Server

Version 5.2

816-6851-10
2003 年 6 月

Copyright © 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

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

この配布には、第三者が開発したソフトウェアが含まれている可能性があります。本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。

UNIX は、X/Open Company, Ltd が独占的にライセンスしている米国およびその他の国における登録商標です。

Sun 、Sun Microsystems 、Sun のロゴマーク、Java、Solaris、SunTone、Sun™ ONE, The Network is the Computer、SunTone 認定ロゴマークおよび Sun™ ONE のロゴマークは、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

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

このサービスマニュアルに含まれる製品および情報は、米国の輸出規制に関する法規の適用および管理下にあり、また、米国以外の国の輸出および輸入規制に関する法規の制限を受ける場合があります。核、ミサイル、生物化学兵器もしくは原子力船に関連した使用またはかかる使用者への提供は、直接的にも間接的にも、禁止されています。このソフトウェアを、米国の輸出禁止国へ輸出または再輸出すること、および米国輸出制限対象リスト (輸出が禁止されている個人リスト、特別に指定された国籍者リストを含む) に指定された、法人、または団体に輸出または再輸出することは一切禁止されています。

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われないものとします。

目 次

本書について

Sun™ ONE Directory Server 5.2 は、業界標準の LDAP (Lightweight Directory Access Protocol) に基づいたスケーラブルで強力な分散型ディレクトリサーバーです。Sun ONE Directory Server ソフトウェアは、Sun Open Net Environment (Sun ONE) の一部です。Sun ONE は、必要に応じたサービスの構築と配備のための、Sun の標準ベースのソフトウェアのヴィジョン、アーキテクチャ、プラットフォーム、および専門技術の総称です。

Sun ONE Directory Server は、社内イントラネットや、取引先とのエクストラネット、顧客との窓口となる公共のインターネット上で使用できる、集中・分散型のデータリポジトリを構築するための基盤となります。

このマニュアルの目的

この『管理者ガイド』では、Sun ONE Directory Server に基づくディレクトリサービスの設定と保守に必要なすべての手順について説明します。Directory Server のすべての機能について、コンソールやコマンド行を適切に使用して設定する手順も含まれています。

事前の確認事項

このマニュアルでは、ディレクトリサーバーとその内容を管理する方法について説明します。ただし、ディレクトリサービスを適切に設計および導入するのに必要な、ディレクトリに関する基本的な事柄やアーキテクチャの概念については説明していません。これらの基本概念については、『Sun ONE Directory Server Deployment Guide』で説明しています。

ディレクトリの導入計画をある程度立ててから、システムを設定し、Sun ONE Directory Server をインストールしてください。Directory Server の各種コンポーネントをインストールする方法については、『Sun ONE Directory Server インストールおよびチューニングガイド』で説明しています。

このマニュアルでは、Directory Server コンソールと基本的なコマンドに精通していることを前提としています。基本的なコマンドについては、『Sun ONE Directory Server Getting Started Guide』で説明しています。特に、コマンド行による手順では `ldapmodify` コマンドが利用されるため、このツールで使用される LDIF (LDAP Data Interchange Format) 入力について理解しておく必要があります。『Sun ONE Server Console Server Management Guide』には、Sun ONE サーバーの使い方に関する一般的な基本情報も記載されています。

表記規則

ここでは、このマニュアルで使用する表記規則について説明します。

クーリエ (等倍) フォント : このフォントは、属性およびオブジェクトクラスの名前などを本文中で使用する場合など、リテラル文字列で使用します。また、URL、ファイル名、および例にも使用します。

斜体文字 (*Italic*) : このフォントは、強調、新出用語、および可変部分 (パス名など実際の値に置き換える必要がある文字列) で使用します。

大なり記号 (>) は、メニューまたはサブメニューの項目を指定するときの区切り文字として使用します。たとえば、「オブジェクト」>「新規」>「ユーザー」は、「オブジェクト」メニューの「新規」サブメニューにある「ユーザー」を選択することを意味します。

注	「注」、「注意」、および「ヒント」は、重要な条件または制限を強調するためのものです。必ずこれらの注意事項を読んでから、作業を続けるようにしてください。

デフォルトパスおよびファイル名

Sun ONE Directory Server 製品マニュアルのパスおよびファイル名の例は、すべて次の 2 つの形式のいずれかです。

- *ServerRoot*/... : *ServerRoot* は、Sun ONE Directory Server のインストール先です。このパスには、Directory Server、管理サーバー、および LDAP コマンドの共有バイナリファイルが置かれています。

  実際の *ServerRoot* パスは、使用するプラットフォーム、インストール内容、および設定によって異なります。15 ページの表 1 に示すように、デフォルトパスはプラットフォームとパッケージによって異なります。

- *ServerRoot*/slapd-*serverID*/... : *serverID* は、インストールや設定のときにユーザーが定義した Directory Server インスタンスの名前です。このパスには、特定のインスタンスに固有のデータベースおよび設定ファイルが含まれます。

注　このマニュアルでは、スラッシュで区切られた UNIX 形式のパスを使用します。また、コマンドの表記にファイル拡張子はありません。Windows ベースの Sun ONE Directory Server を使用している場合は、同等の Windows 形式の円記号 (¥) に読み替えます。Windows プラットフォームの実行可能ファイルには、UNIX 形式と同じ名前に `.exe` または `.bat` という拡張子が付いています。

表 1　デフォルトの ServerRoot パス

製品のインストール	*ServerRoot* パス
Solaris パッケージ[1]	`/var/mps/serverroot` - 設定が完了すると、このディレクトリには次の場所へのリンクが含まれます。 • `/etc/ds/v5.2` (スタティック設定ファイル) • `/usr/admserv/mps/admin` (Sun ONE 管理サーバーバイナリ) • `/usr/admserv/mps/console` (サーバーコンソールバイナリ) • `/usr/ds/v5.2` (Directory Server バイナリ)
Solaris およびその他の UNIX システムでの圧縮アーカイブインストール	`/var/Sun/mps`
Windows システムでの Zip インストール	`C:¥Program Files¥Sun¥MPS`

1. Solaris オペレーティング環境で作業しており、インストールされている Sun ONE Directory Server のバージョンが不確かな場合は、pkginfo コマンドを使用して SUNWdsvu などのキーパッケージの有無を確認してください。例:pkginfo | grep SUNWdsvu

Directory Server インスタンスは、*ServerRoot*/slapd-*serverID*/ の下に配置されます。ここで、*serverID* は作成するインスタンスに指定されたサーバー識別子を表します。たとえば、Directory Server の名前を dirserv にした場合、実際のパスは表 2 に示されるようになります。Directory Server インスタンスを別の場所に作成した場合は、それに対応してパスを変更する必要があります。

表 2　　dirserv インスタンスの場所

製品のインストール	インスタンスの場所
Solaris パッケージ	/var/mps/serverroot/slapd-dirserv
Solaris およびその他の UNIX システムでの圧縮アーカイブインストール	/usr/Sun/mps/slapd-dirserv
Windows システムでの Zip インストール	C:¥Program Files¥Sun¥MPS¥slapd-dirserv

Directory Server ツールのダウンロード

サポートされているプラットフォームの一部には、Directory Server にアクセスするためのネイティブツールが用意されています。LDAP ディレクトリサーバーのテストとメンテナンス用のツールを利用するには、Sun ONE Directory Server Resource Kit (DSRK) をダウンロードしてください。このソフトウェアは、次の Web サイトから入手できます。

```
http://wwws.sun.com/software/download/
```

DSRK ツールのインストール方法と参照情報は、『Sun ONE Directory Server Resource Kit Tools Reference』に記載されています。

ディレクトリクライアントアプリケーションを開発する場合には、Sun ONE LDAP SDK for C および Sun ONE LDAP SDK for Java を同じ Web サイトからダウンロードすることもできます。

さらに、JNDI (Java Naming and Directory Interface) テクノロジは、Java アプリケーションから LDAP および DSML v2 を使用して Directory Server へのアクセスをサポートします。JNDI に関する情報は次の Web サイトから入手できます。

```
http://java.sun.com/products/jndi/
```

JNDI チュートリアルには、JNDI の使用方法に関する詳細な説明と事例が記載されています。JNDI チュートリアルは次の Web サイトから入手できます。

```
http://java.sun.com/products/jndi/tutorial/
```

推奨参考文献

Sun ONE Directory Server 製品マニュアルには、HTML と PDF の両方の形式で提供される次のドキュメントが含まれています。

- 『Sun ONE Directory Server Getting Started Guide』: Directory Server 5.2 の多くの重要な機能を要約して説明します。
- 『Sun ONE Directory Server Deployment Guide』: ディレクトリトポロジ、データ構造、セキュリティ、および監視の設計方法について説明し、導入事例を検討します。
- 『Sun ONE Directory Server インストールおよびチューニングガイド』: インストールおよびアップグレード手順について説明し、Directory Server のパフォーマンスの最適化に関するヒントを提供します。
- 『Sun ONE Directory Server 管理ガイド』: コンソールおよびコマンド行を使用して、ディレクトリの内容を管理し、Directory Server のすべての機能を設定する手順について説明します。
- 『Sun ONE Directory Server Reference Manual』: Directory Server の設定パラメータ、コマンド、ファイル、エラーメッセージ、およびスキーマの詳細について説明します。
- 『Sun ONE Directory Server Plug-In API Programming Guide』: Directory Server プラグインの開発方法について説明します。
- 『Sun ONE Directory Server Plug-In API Reference』: Directory Server プラグイン API のデータ構造と機能の詳細について説明します。
- 『Sun ONE Server Console Server Management Guide』: Sun ONE 管理サーバーおよび Java ベースのコンソールを使用してサーバーを管理する方法について説明します。
- 『Sun ONE Directory Server Resource Kit Tools Reference』: 多くの有用なツールを含む Sun ONE Directory Server Resource Kit のインストールと機能について説明します。

その他の有用な情報は、次の Web サイトから入手できます。

- 製品マニュアル : `http://docs.sun.com`

- Sun ソフトウェア : http://wwws.sun.com/software/
- Sun ONE サービス : http://www.sun.com/service/sunps/sunone/
- Sun サポートサービス : http://www.sun.com/service/support/
- Sun ONE 開発者向けサイト : http://sunonedev.sun.com/
- トレーニング : http://suned.sun.com/

第 1 章

Sun™ ONE Directory Server の概要

Sun™ ONE Directory Server 製品には、Directory Server、複数のディレクトリを管理するための管理サーバー、および両方のサーバーを管理するグラフィカルインタフェースを提供する Sun ONE サーバーコンソールが含まれています。この章では、Directory Server の概要と、コンソールを使用してディレクトリサービスの管理を開始するときの基本的な作業について説明します。

この章で説明する Directory Server 5.2 の新しい 2 つの機能は、プラグイン署名と DSML-over-HTTP プロトコルです。プラグイン署名の検証は新たに追加されたセキュリティ機能の 1 つで、未承認のプラグインがロードされようとしていることをサーバーが検出し防止することができます。DSML (Directory Server Markup Language) は、ディレクトリサーバーに要求を送信するための、XML ベースの新しい形式です。

この章は、次の節で構成されます。

- Directory Server の管理の概要
- Directory Server の起動と停止
- SSL が有効になった状態でのサーバーの起動
- Directory Server コンソールの使用
- LDAP パラメータの設定
- プラグインの署名の検証
- DSML の設定

Directory Server の管理の概要

Sun ONE Directory Server は、企業全体のユーザーおよびリソースのディレクトリの管理用に設計された、堅牢かつスケーラブルなサーバーです。LDAP (Lightweight Directory Access Protocol) というオープンシステムサーバープロトコルに基づいています。Directory Server は、`ns-slapd` プロセスまたはサービスとしてマシン上で動作します。このサーバーは、ディレクトリの内容を管理し、クライアントからの要求を処理します。

Directory Server のほとんどの管理作業は、管理サーバーを経由して実行できます。管理サーバーは、Directory Server (およびその他の Sun ONE サーバーすべて) を管理できるようにするために Sun ONE が提供するもう 1 つのサーバーです。Sun ONE サーバーコンソールは、管理サーバーのグラフィカルインタフェースです。Directory Server コンソールは Sun ONE サーバーコンソールの一部であり、特に Sun ONE Directory Server で使うために設計されたものです。

ほとんどの Directory Server 管理タスクは、Directory Server コンソールから実行できます。設定ファイルを編集するか、コマンド行ユーティリティを使って、手動で管理タスクを実行することもできます。Sun ONE サーバーコンソールについては、『Sun ONE Server Console Server Management Guide』を参照してください。

Directory Server の起動と停止

SSL (Secure Sockets Layer) を使っていない場合は、次の方法で Directory Server を起動および停止します。SSL を使っている場合は、22 ページの「SSL が有効になった状態でのサーバーの起動」を参照してください。

注 UNIX システムでは、Solaris パッケージからインストールした場合を除き、システムを再起動しても `slapd` プロセスは自動的に起動されません。これは、Directory Server では起動スクリプトやランコマンド (`rc`) スクリプトを自動的に作成しないためです。スクリプトの記述方法についての詳細は、オペレーティングシステムのマニュアルを参照してください。

コマンド行からのサーバーの起動と停止 (UNIX)

ディレクトリサーバーが停止されている状態で、Directory Server コンソールが稼動していないときは、コマンド行からサーバーを起動する必要があります。Directory Server コンソールを使いたくないときは、コマンド行からサーバーを停止することもできます。root 権限で、次のいずれかのコマンドを実行します。

Solaris パッケージ
```
# /usr/sbin/directoryserver start
```
その他のインストール
```
# ServerRoot/slapd-serverID/start-slapd
```

または

Solaris パッケージ
```
# /usr/sbin/directoryserver stop
```
その他のインストール
```
# ServerRoot/slapd-serverID/stop-slapd
```

ここで *serverID* は、インストール時に指定したサーバーの識別子を示します。

UNIX では、どちらのスクリプトも Directory Server と同じ UID と GID を使って実行する必要があります。たとえば、Directory Server を `nobody` として実行する場合は、`start-slapd` および `stop-slapd` ユーティリティも `nobody` として実行する必要があります。

リフェラルモードは使用できなくなりました。グローバルリフェラルを設定するには Directory Server コンソールを使います。この手順については、73 ページの「デフォルトリフェラルの設定」を参照してください。

コントロールパネルからのサーバーの起動と停止 (Windows)

Windows システムでは、コントロールパネルの「サービス」から次の手順を実行します。

1. デスクトップから、「スタート」> 「設定」> 「コントロールパネル」の順に選択します。
2. 「サービス」アイコンをダブルクリックします。
3. サービスのリストをスクロールし、Sun ONE Directory Server を選択します。

   サービス名は `Sun ONE Directory Server 5.2` (*serverID*) です。ここで *serverID* は、サーバーのインストール時または設定時に指定した識別子を示します。
4. 実行する操作に応じて「開始」または「停止」ボタンをクリックします。

   Directory Server を停止する場合は、サービスの停止を確認するメッセージが表示されます。

コンソールからのサーバーの起動と停止 (すべてのプラットフォーム)

Directory Server コンソールが稼動しているときは、グラフィカルインタフェースを使ってディレクトリサーバーを起動、停止、再起動することができます。コンソールを実行する方法については、23 ページの「Directory Server コンソールの起動」を参照してください。

1. Directory Server コンソールの最上位にある「タスク」タブで、「Directory Server の起動」、「Directory Server の停止」、「Directory Server の再起動」の隣にあるボタンをクリックします。

Directory Server コンソールからの Directory Server の起動または停止が正常に完了すると、サーバーが正常に起動または停止したことを示すダイアログが表示されます。エラーが発生した場合は、エラーに関連するすべてのメッセージがコンソールに表示されます。

SSL が有効になった状態でのサーバーの起動

SSL を有効化する前に、サーバーに証明書をインストールし、それを設定する必要があります。証明書の管理と SSL の有効化については、第 11 章「セキュリティの実装」を参照してください。証明書、証明書データベース、サーバー証明書の取得については、『Sun ONE Server Console Server Management Guide』の第 10 章「Using SSL and TLS with Sun ONE Servers」を参照してください。

SSL を有効にしてサーバーを起動するには、サーバーの証明書を保護するパスワードを指定する必要があります。

- Windows では、サーバーのホストマシンからサーバーを起動する必要があります。セキュリティ上の理由から、パスワードを要求するダイアログボックスは、サーバーのホストマシン上だけに表示されます。
- UNIX では、コマンド行からサーバーを起動する必要があります。

また、どちらのプラットフォーム上でも、パスワードファイルを作成して、証明書のパスワードを格納できます。証明書用のデータベースパスワードをファイルに格納することによって、サーバーコンソールからサーバーを起動できます。さらに、無人でサーバーを実行している場合も、サーバーを自動的に再起動させることができます。

警告 パスワードファイル内のパスワードは、暗号化をされていないテキスト形式で格納されています。したがって、この方法を使うと、セキュリティ上のリスクを負うことになります。サーバーが動作している環境のセキュリティが十分に保護されている場合を除き、パスワードファイルは使わないでください。

パスワードファイルは、次の位置に置く必要があります。

ServerRoot/alias/slapd-*serverID*-pin.txt

ここで、*serverID* は、インストール時に指定したサーバーの識別子を示します。

ファイルには、次のようにセキュリティトークンの名前とパスワードを含める必要があります。

deviceName Token:*password*

次の例には、内部証明書データベースのデバイス名が示されています (大文字と小文字の区別、および空白文字の有無は、この例のとおりに指定する必要があります)。

Internal (Software) Token:*password*

証明書を別のデバイスに格納しているときは、「証明書の管理」ダイアログ上部のドロップダウンメニューに表示されるデバイス名を指定します。証明書データベースを作成するには、管理サーバーと証明書設定ウィザードを使う必要があります。Directory Server 上での SSL の使い方については、第 11 章「セキュリティの実装」を参照してください。

Directory Server コンソールの使用

Directory Server コンソールは、Sun ONE サーバーコンソールの別のウィンドウとして表示されるインタフェースです。次の手順に従って Sun ONE サーバーコンソールから Directory Server コンソールを起動します。

Directory Server コンソールの起動

1. ディレクトリサーバーデーモン slapd-*serverID* が動作していることを確認します。起動していない場合は、root ユーザーまたは管理ユーザーとして次のコマンドを入力し、デーモンを起動します。

Solaris パッケージ
```
# /usr/sbin/directoryserver start
```
その他のインストール
```
# ServerRoot/slapd-serverID/start-slapd
```

2. 管理サーバーデーモン admin-serv が動作していることを確認します。起動していない場合は、root ユーザーまたは管理ユーザーとして次のコマンドを入力し、デーモンを起動します。

Solaris パッケージ
```
# /usr/sbin/directoryserver start-admin
```
その他のインストール
```
# ServerRoot/start-admin
```

3. 次のコマンドを入力して Sun ONE サーバーコンソールを起動します。

Solaris パッケージ
```
# /usr/sbin/directoryserver startconsole
```
その他のインストール
```
# ServerRoot/startconsole
```

Sun ONE 管理サーバーがインストールされているマシンとは異なるマシンで Sun ONE サーバーコンソールを実行しているときは、管理サーバーに接続制限を設定する必要があります。詳細は、『Sun ONE Server Console Server Management Guide』の第 7 章にある「Network Settings」を参照してください。

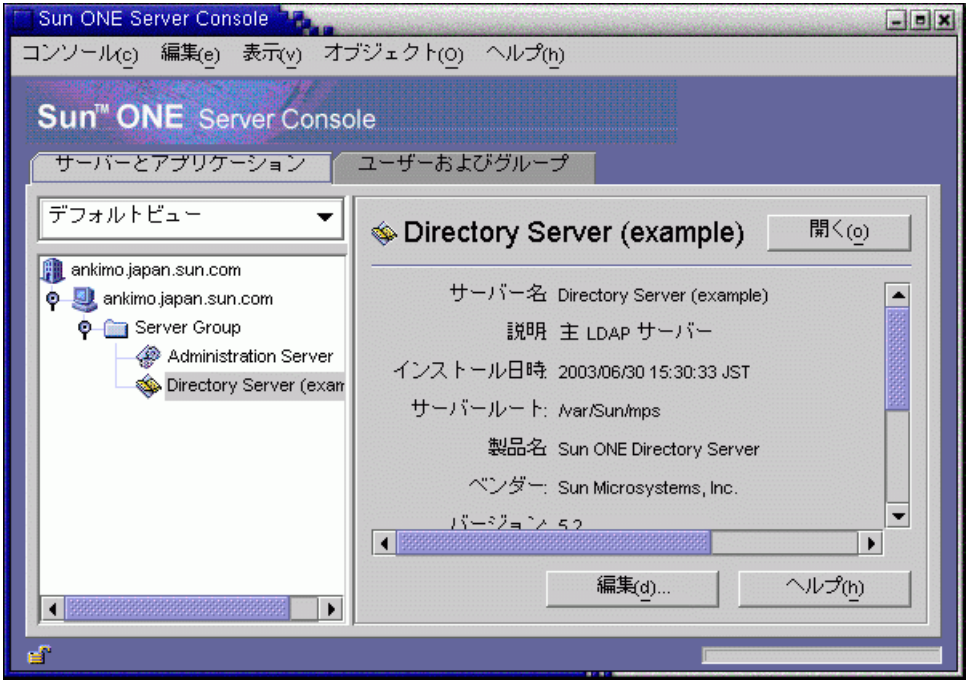
コンソールのログインウィンドウが表示されます。設定ディレクトリ (`o=NetscapeRoot` サフィックスを含むディレクトリ) が Directory Server の別のインスタンスに保存されている場合は、表示されたウィンドウに、管理ユーザー DN、パスワード、およびそのディレクトリサーバーの管理サーバーの URL を入力する必要があります。

4. 目的の操作を実行するために必要なアクセス権を持つユーザーのバインド DN とパスワードを使ってログインします。たとえば、`cn=Directory Manager` とこれに対応する適切なパスワードを使用します。

   Sun ONE サーバーコンソールが表示されます。

5. 左側のパネルのツリーを使って Directory Server のホストマシンを検索し、ホストマシンの名前またはアイコンをクリックして全般的なプロパティを表示します。

図 1-1　Sun ONE サーバーコンソール



ディレクトリサーバーの名前および説明を編集するには、「編集」ボタンをクリックします。テキストボックスに新しい名前および説明を入力します。「了解」をクリックして、新しい名前と説明を設定します。図のように左のツリーに名前が表示されます。

6. ツリー内の Directory Server 名をダブルクリックまたは「開く」ボタンをクリックして、このディレクトリサーバーを管理する Directory Server コンソールを表示します。

Directory Server コンソールの操作

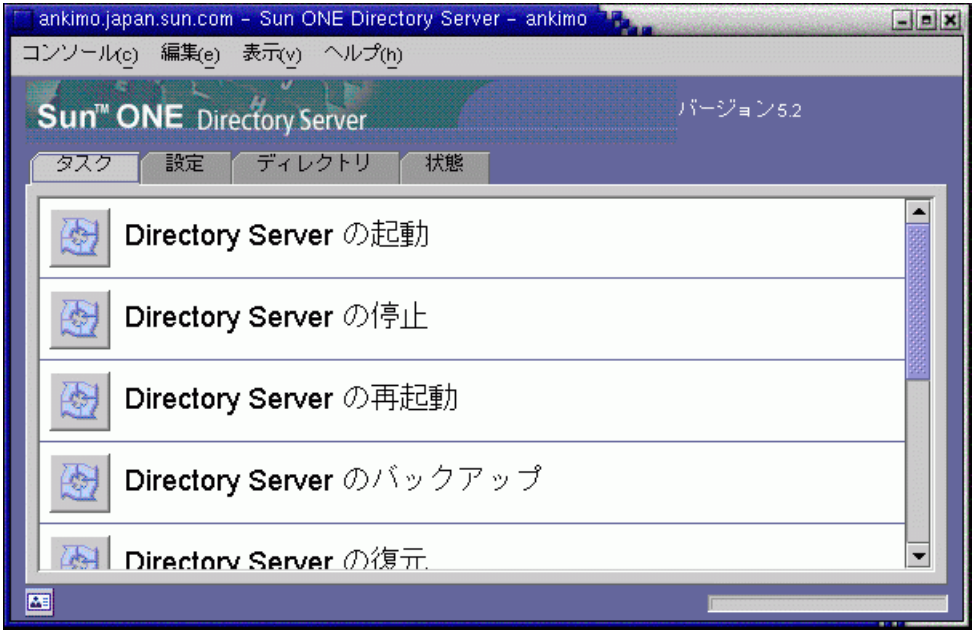
Directory Server コンソールは、Directory Server インスタンスで参照および管理操作を実行するためのインタフェースを提供します。常時表示されている 4 つのタブを使うと、Directory Server のすべての機能にアクセスできます。

- 「タスク」タブ : サーバーの再起動などの管理タスクを実行するボタンが表示されています。
- 「設定」タブ : サーバーを管理するためのパラメータを使用できます。
- 「ディレクトリ」タブ : ディレクトリにあるデータエントリを表示および編集できます。
- 「状態」タブ : サーバーの統計情報、ログ、およびレプリケーション状態が表示されます。

タスクタブ

「タスク」タブは、Directory Server コンソールを開くと最初に表示されるタブです。このタブには、次の図に示すように、Directory Server の起動または終了など主要な管理タスクすべてを実行するボタンが表示されています。すべてのタスクとボタンを表示するには、リストをスクロールする必要があります。

図 1-2　Directory Server コンソールの「タスク」タブ



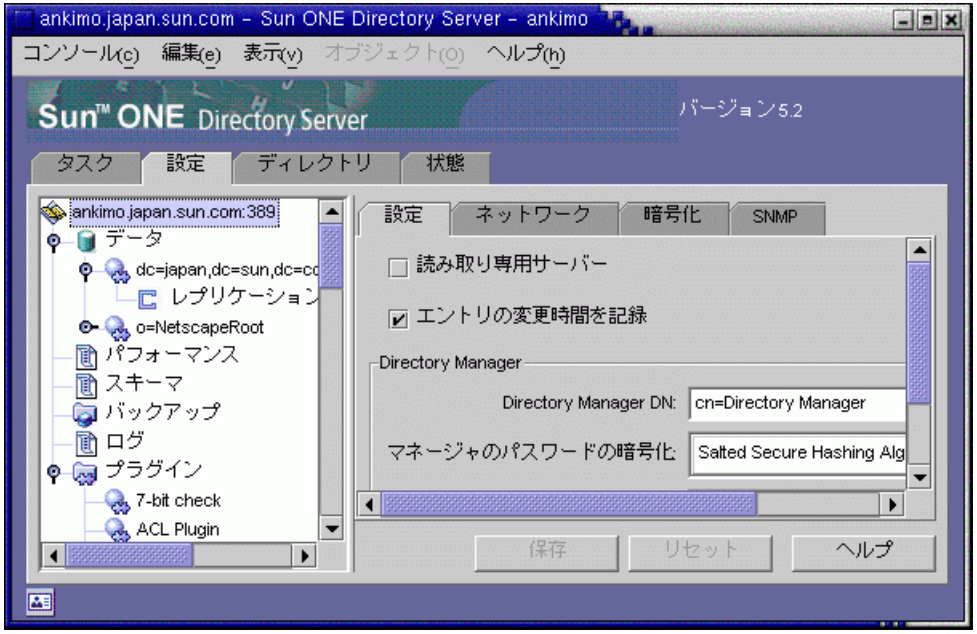
これらのタスクを実行するには、管理者の権限を持つユーザーとしてログインする必要があります。適切な権限を持たないユーザーがアクセスしたときは、タスクボタンは表示されません。

設定タブ

Directory Server コンソールの「設定」タブでは、サフィックス、レプリケーション、スキーマ、ログ、およびプラグインなどのすべてのディレクトリ設定を表示および変更するためのインタフェースおよびダイアログボックスが提供されます。これらのダイアログボックスは、管理者の権限を持つユーザーとしてログインした場合にだけ有効となります。

このタブの左側にはすべての設定機能のツリー、右側には各機能に特有の管理用インタフェースが表示されます。これらのインタフェースには、通常、ほかのタブ、ダイアログボックス、またはポップアップウィンドウがあります。次の図にディレクトリ全体の一般設定を示します。

図 1-3　Directory Server コンソールの「設定」タブ



左側のツリーから設定可能な項目を選択すると、選択した項目の現在の設定が右側のパネルの 1 つ以上のタブに表示されます。これらの設定の説明および動作については、このマニュアルの各機能について説明した章を参照してください。設定に応じて、保存すると変更がすぐに反映される場合と、サーバーが再起動されるまで反映されない場合があります。サーバーの再起動が必要な場合は、それを示すダイアログが表示されます。

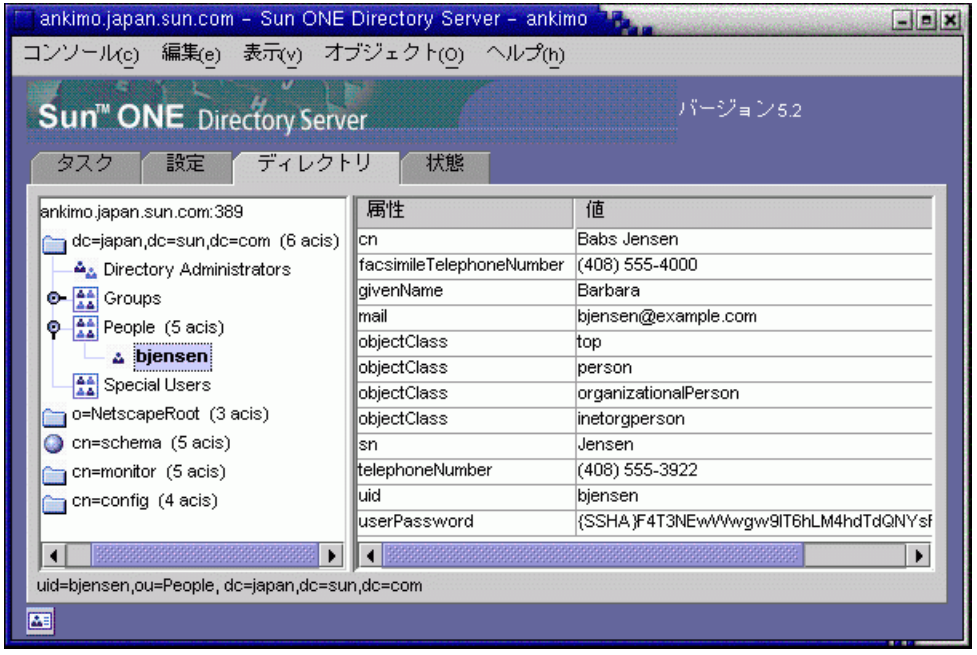
変更が保存されていないタブでは、タブ名の隣に赤のマークが表示されます。変更を保存しないで別の項目を設定したり、他の主要なタブを表示したりしても、行なった変更はタブに残ります。「保存」および「リセット」ボタンは、指定された設定項目のすべてのタブに適用されますが、他の項目の未保存の設定には影響を与えません。

ほとんどのテキストフィールドには、正しい構文でだけ設定値を入力できます。デフォルトでは、設定および値が正しい構文で入力されるまで、そのラベルが赤でハイライト表示されます。すべての設定に有効な構文が指定されるまで、「保存」ボタンは無効になります。32 ページの「視覚的な設定」で説明する方法で、不正な値を斜体で強調するように設定できます。

ディレクトリタブ

コンソールの「ディレクトリ」タブには、移動しやすいようにディレクトリエントリがツリー構造で表示されます。このタブでは、すべてのエントリとその属性を参照、表示、および編集できます。

図 1-4　Directory Server コンソールの「ディレクトリ」タブ



ログイン時に指定されたバインド DN のアクセス権限が十分な場合には、設定エントリが通常のエントリとみなされ、直接変更することが可能です。ただし、設定を安全に変更するには、「設定」タブから利用できるダイアログボックスを使う必要があります。

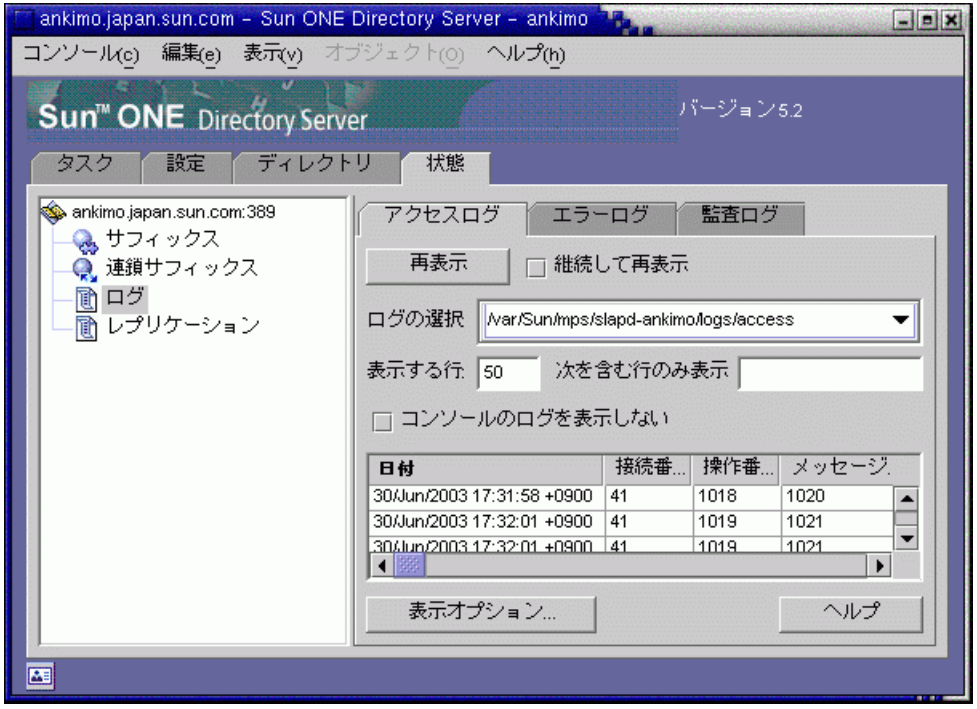
「ディレクトリ」タブのレイアウトおよび内容を変更する場合は、「表示」メニューのオプションを使用できます。新しいレイアウトオプションを使うと、最下位のエントリを含むすべてのエントリを 1 つのツリーに表示したり、右側のパネルに属性を表示したりできます。デフォルトでは、最下位のエントリは、左側のツリーではなく、右側に表示されます。

「表示」>「表示」オプションでは、ディレクトリツリーのすべてのエントリの ACI カウント、ロールカウント、および「アクティブでない状態」アイコンを使用できます。前の図では、ACI カウントおよび最下位のエントリは左側のツリーに表示され、選択したエントリの属性値が右側のパネルに表示されています。詳細は、32 ページの「ディレクトリツリーの表示オプション」を参照してください。

状態タブ

「状態」タブには、サーバー統計およびログメッセージが表示されます。左側のツリーには、すべての状態項目が一覧表示されます。項目が選択されると、その内容が右側のパネルに表示されます。次の図ではログエントリのテーブルを表示しています。

図 1-5　Directory Server コンソールの「状態」タブ



コンソールからの現在のバインド DN の表示

ディスプレイの左下隅にあるログインアイコンをクリックすると、Directory Server コンソールへのログインに使ったバインド DN を表示できます。次に示すように、現在のバインド DN がログインアイコンの隣に表示されます。

ログイン ID の変更

Directory Server コンソールからエントリの作成や管理を行う場合や、はじめて Sun ONE サーバーコンソールにアクセスする場合は、バインド DN とパスワードを入力してログインすることもできます。これによって、ディレクトリツリーにアクセスしているユーザーを特定し、操作を実行するために許可されているアクセス権を決定できます。

最初に Sun ONE サーバーコンソールを起動するときは、Directory Manager DN を使ってログインできます。ログイン後は、コンソールを停止して再起動しなくても、いつでも別のユーザーとしてログインできます。

Sun ONE サーバーコンソールでログイン名を変更するには、次の手順を実行します。

1. Directory Server コンソールの「タスク」タブを選択し、「Directory Server に新規ユーザーとしてログイン」というラベルの隣のボタンをクリックします。コンソールの別のタブでは、「コンソール」>「新規ユーザーとしてログイン」メニューの順に選択します。

   ログインダイアログボックスが表示されます。

2. 新しい DN とパスワードを入力し、「了解」をクリックします。

   サーバーにバインドするエントリの絶対識別名を入力します。たとえば、Directory Manager としてバインドする場合は、「識別名」テキストボックスに次のように DN を入力します。

   cn=Directory Manager

Directory Manager DN およびパスワードの詳細は、次の節で説明します。

オンラインヘルプの使用

オンラインヘルプは、Directory Server コンソールのほとんどのタブとダイアログについて、状況に適した情報を提供します。「ヘルプ」ボタンは、通常は各インタフェースの右下端にあります。状況に即したヘルプを呼び出すには、キーボードショートカットとしてどの画面でも `Alt-P` を使用します。

オンラインヘルプを呼び出すと、コンソールの内蔵ブラウザに HTML ベースのページが表示されます。ここで「ブラウザで開く」ボタンをクリックすると、同じページを Netscape Communicator などの外部ブラウザで表示できます。オンラインヘルプ内でさらに詳しい情報へのリンクをクリックした場合も、外部ブラウザのウィンドウにその情報が表示されます。

オンラインヘルプの各ページには、対応するタブまたはダイアログに含まれるフィールドとボタンの説明が表示されます。コンソールで値の入力または変更を行なったり、値の意味を調べたりするときは、この情報を参考にしてください。

Sun ONE Directory Server のヘルプシステムは、Sun ONE 管理サーバーに依存しています。管理サーバーからはリモートとなるマシンで Directory Server コンソールを稼動している場合は、次の項目を確認する必要があります。

- 使用マシンからのアクセスを許可するように、管理サーバーに設定されている接続制限を変更する必要がある場合があります。手順については、『Sun ONE Server Console Server Management Guide』の第 7 章にある「Network Settings」を参照してください。
- オンラインヘルプのページを外部のブラウザで表示する場合、そのブラウザがプロキシを使用するように設定されているときは、次のいずれかの処理を行う必要があります。
  - ブラウザ設定のプロキシを無効にします。Netscape Communicator の場合は、「編集」メニューの「設定」を選択します。次に、「詳細」の「プロキシ」を選択して、プロキシ設定を表示します。Internet Explorer の場合は、「ツール」メニューの「インターネットオプション」を選択します。
  - プロキシサーバーからのアクセスを許可するように、管理サーバーの接続制限を設定します。

警告 プロキシサーバーからのアクセスを許可するように管理サーバーを設定すると、システムにセキュリティホールが生じる可能性があります。

コンソールクリップボード

Directory Server コンソールでは、システムのクリップボードを使ってテキストのコピー、カット、およびペーストを行います。「ディレクトリ」タブ内での操作中に、エントリの DN または URL をクリップボードにコピーすることで、入力する文字数を減らすことができます。

DN や URL をテキストイールドにペーストする必要のあるダイアログや別のタブを開く前に、次の手順を実行します。

1. Directory Server コンソールの最上位にある「ディレクトリ」タブでツリーを表示し、DN または URL をコピーするエントリを選択 (左クリック) します。
2. 次にメニューの「編集」>「DN のコピー」または「編集」>「URL のコピー」の順に選択します。

コンソールの設定

Directory Server コンソールには、「設定」タブと「ディレクトリ」タブの情報の表示をカスタマイズするために、多数の設定が用意されています。

視覚的な設定

「設定」タブの最上位にあるフィールドで設定パラメータを変更したり、値を入力すると、Directory Server コンソールは有効な入力を色分けしたテキストで表示します。たとえば、設定値の入力を要求する機能を有効にすると、入力が必要なフィールドのラベルは赤で表示され、有効な値を入力すると青に変わります。

デフォルトでは、コンソールは赤と青で色分けされますが、次の手順でこの設定を変更することができます。

1. Directory Server コンソールのいずれかのタブで、「編集」メニューから「プリファレンス」を選びます。「コンソールのプリファレンス」ダイアログで、「その他」タブを選択します。
2. 視覚的な設定のラジオボタンを選択します。フォントの色分け、別フォントによる表示、または両方を指定できます。
3. 「コンソールのプリファレンス」ダイアログのその他のタブの設定については、『Sun ONE Server Console Server Management Guide』の第 3 章にある「Customizing Sun ONE Server Cosole」を参照してください。

   「了解」をクリックして、変更を保存します。
4. Sun ONE サーバーコンソールのすべてのウィンドウを閉じ、再起動します。

ディレクトリツリーの表示オプション

Directory Server コンソールの最上位にある「ディレクトリ」タブで「表示」メニューの項目を使用することで、ディレクトリツリーに追加情報を表示したり、右側のパネルに表示する情報を選択したりすることができます。

「ディレクトリ」タブに表示される内容には、次の「表示」オプションが適用されます。

- リフェラルをたどる : このチェックボックスを選択すると、エントリと、リフェラルのターゲットのすべての子がディレクトリに含まれているようにディレクトリツリーに表示されます。チェックボックスを選択しない場合は、リフェラルはリフェラルのエントリとして表示されます。詳細は、74 ページの「スマートリフェラルの作成」を参照してください。

- オブジェクトのソート : このチェックボックスを選択しない場合は、エントリはサーバーから返される順序で表示されます。このチェックボックスを選択すると、後述する表示属性に従って、同じレベルのエントリがディレクトリツリーでソートされます。サーバーのパフォーマンスに影響を与えずに大規模なサブツリーをソートする方法については、366 ページの「コンソール用のブラウズインデックス」を参照してください。

  エントリは、cn、givenname、o、ou、sn、uid の属性でソートされて表示されます。その他の属性によって表示されるエントリはソートされません。

- 表示 > ACI カウント : エントリが aci 属性に 1 つまたは複数の ACI (アクセス制御命令) を含んでいるときは、ディレクトリツリーのエントリの隣にその数が表示されます。詳細については、第 6 章「アクセス制御の管理」を参照してください。

- 表示 > ロール数 : エントリが 1 つまたは複数のロールのメンバーである場合は、ディレクトリツリーのエントリの隣にその数が表示されます。詳細は、157 ページの「ロールの割り当て」を参照してください。

- 表示 > アクティブでない状態 : サーバーへのバインドを防止するためにユーザーまたはグループのエントリが無効化されている場合は、ディレクトリツリーのエントリのアイコンに赤いボックスと線が表示されます。詳細は、269 ページの「ユーザーとロールの無効化と有効化」を参照してください。

- レイアウト > 子を表示 : このレイアウトオプションを選択すると、左側のパネルのツリーにディレクトリの最下位エントリが表示されなくなり、左側のパネルで親ノードを選択すると、最下位エントリを含め、そのノードのすべての子が右側のパネルに表示されます。どちらのパネルでもエントリを選択できます。

- レイアウト > ツリーのみを表示 : このレイアウトオプションを選択すると、「ディレクトリ」タブにはディレクトリ内のすべてのエントリを含むツリーを示す 1 つのパネルだけが表示されます。

- レイアウト > 属性の表示 : このレイアウトでは、左側のパネルにはディレクトリ内のすべてのエントリを含むツリーが表示され、右側のパネルにはツリーで選択しているエントリに設定されている属性とその値が表示されます。

- 属性の表示 : このメニュー項目を選択すると、「ディレクトリ」タブに表示される属性のラベルを選択するための「属性の表示」ダイアログが表示されます。デフォルトでは、たとえば People のように、エントリの最初の RDN 属性の値がラベルとなります。RDN を持たないベースエントリでは、たとえば dc=example,dc=com のように、DN 全体がラベルとなります。

  別の属性を使ってディレクトリツリーに含まれるエントリを表示するときは、その他のラジオボタンを選んでから属性を選択します。選択されている属性を持たないエントリは、エントリの最初の RDN 属性の値を使い続けます。デフォルトでは、属性値だけがラベルに使われます。「属性名を表示」チェックボックスを選択すると、ラベルは ou=People のように表示されます。

- 再表示 : 特定の操作の後に新しい値を表示するには、ディレクトリツリーの表示を更新する必要があります。この項目を選択すると、ディレクトリツリー全体がサーバーから再度読み込まれます。

LDAP パラメータの設定

Directory Manager の識別名 (DN)、グローバルな読み取り専用設定、ポートの設定、すべてのディレクトリ変更日時の追跡など、LDAP パラメータはディレクトリサーバーの基本的な設定です。

Directory Manager の設定

Directory Manager とは、特権を持つサーバー管理者のことで、UNIX の root ユーザーにあたります。このため、Directory Manager として定義したエントリには、アクセス制御は適用されません。このエントリは、インストール時に初めて定義されます。デフォルト値は `cn=Directory Manager` です。

Directory Manager の DN は `nsslapd-rootDN` 属性に格納され、パスワードは `cn=config` ブランチの `nsslapd-rootpw` 属性に格納されます。

Directory Server コンソールを使って、Directory Manager DN、パスワード、およびパスワードの暗号化スキーマを変更します。

1. Directory Manager としてコンソールにログインします。

   すでにコンソールにログインしている場合に、別のユーザー名でログインする方法については、30 ページの「ログイン ID の変更」を参照してください。

2. 最上位の「設定」タブで、ナビゲーションツリーのルートでサーバーのノードを選択し、右側のパネルで「設定」タブを選択します。
3. 「DN」フィールドに、Directory Manager の新しい識別名を入力します。デフォルト値は、インストール時に定義した識別名です。
4. 「マネージャのパスワードの暗号化」プルダウンメニューから、サーバー上で Directory Manager のパスワードを格納するために使う保存スキーマを選択します。
5. 新しいパスワードを入力し、該当するテキストフィールドで、入力したパスワードを確認します。
6. 「保存」をクリックします。

Directory Server のポート番号の変更

Directory Server コンソールを使うか、`cn=config` エントリの下にある `nsslapd-port` 属性の値を変更して、ユーザーディレクトリサーバーのポート番号またはセキュリティ保護されたポート番号を変更できます。

Sun ONE の設定情報 (`o=NetscapeRoot` サブツリー) を含む Sun ONE Directory Server のポートまたはセキュリティ保護されたポートを変更する場合は、Directory Server コンソールを使います。

設定ディレクトリ、またはユーザーディレクトリのポート番号やセキュリティ保護されたポート番号を変更する場合は、次の点に注意してください。

- 管理サーバーの設定ディレクトリ、またはユーザーディレクトリのポート番号やセキュリティ保護されたポート番号を変更する必要があります。『Sun ONE Server Console Server Management Guide』の第 7 章にある「Network Settings」を参照してください。
- この設定ディレクトリまたはユーザーディレクトリを使用するように指定しているほかの Sun ONE サーバーがインストールされている場合は、これらのサーバーを更新して、新しいポート番号を指定する必要があります。

ディレクトリサーバーが LDAP 要求の受信を待機するポートまたはセキュリティ保護されたポートを変更するには、次の手順を実行します。DSML 要求のポートを変更する方法については、40 ページの「DSML の設定」を参照してください。

1. Directory Server コンソールの最上位の「設定」タブで、サーバー名のルートノードを選び、右側のパネルで「ネットワーク」タブを選びます。

   サーバーの現在の LDAP プロトコル用のポート設定が表示されます。

2. 「ポート」フィールドに、サーバーが SSL 以外の通信に使用するポート番号を入力します。デフォルトは 389 です。

3. 第 11 章「セキュリティの実装」で説明する方法で、このサーバーで SSL を有効化したときは、セキュリティ保護されたポートへの接続も指定できます。

   a. セキュリティ保護されたポートと保護されていないポートの両方を使用するオプションを選択します。

   b. 「セキュリティ保護されたポート」フィールドに、サーバーが SSL 通信に使用するポート番号を入力します。デフォルトは 636 です。

      指定する暗号化ポート番号は、通常の LDAP 通信に使うポート番号とは異なるものにする必要があります。

4. 「保存」をクリックして、サーバーを再起動します。

   詳細は、20 ページの「Directory Server の起動と停止」を参照してください。

グローバルな読み取り専用モードの設定

ディレクトリの各サフィックスは個別に読み取り専用モードにすることができ、固有のリフェラルが定義されていれば、それを取得できます。Directory Server には、すべてのサフィックスに適用されるグローバルな読み取り専用モードも用意されています。このモードでは、グローバルリフェラルが定義されていれば、それを取得できます。

グローバルな読み取り専用モードは、管理者がサフィックスのインデックスを作成し直すときなどに、途中でディレクトリの内容が変更されないようにするために使います。このため、グローバルな読み取り専用モードは、次の設定分岐には適用されません。

- `cn=config`
- `cn=monitor`
- `cn=schema`

これらの分岐は、読み取り専用の設定にかかわらず管理ユーザー以外のユーザーによって変更されることがないよう、常に ACI (アクセス制御命令) で保護されているべきです (第 6 章「アクセス制御の管理」を参照)。グローバルな読み取り専用モードでは、ディレクトリの他のサフィックスに対する更新操作を防止できます。Directory Manager によって開始される更新操作も行うことはできません。

読み取り専用モードでは、このモードが適用されているサフィックスについてはレプリケーションも中断されます。レプリケーションの対象となる変更がマスターレプリカに加えられることはなくなります。ただし、読み取り専用モードが適用される前に加えられた変更は、引き続きレプリケートされます。読み取り専用モードが無効になるまでは、コンシューマレプリカが更新を受け取ることはありません。マルチマスターレプリケーション環境のマスターは、レプリケーションの対象となる変更が加えられることも、他のマスターから更新を受け取ることもありません。

グローバルな読み取り専用モードを有効または無効にするには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで、設定ツリーのルートノードを選択します。次に、右側のパネルで「設定」タブを選択します。
2. 「読み取り専用サーバー」チェックボックスを選択または選択解除します。
3. 「保存」をクリックします。変更は直ちに適用されます。

個別のサフィックスを読み取り専用モードに設定する方法については、133 ページの「サフィックスの読み取り専用モードの設定」を参照してください。

ディレクトリエントリへの変更の記録

Directory Server は、新しく作成されたエントリや変更されたエントリに対して、特別な属性を維持するように設定できます。

- `creatorsName` : エントリを最初に作成したユーザーの識別名
- `createTimestamp` : エントリの作成時刻を GMT (グリニッジ標準時) で記録したタイムスタンプ
- `modifiersName` : 最後にエントリを変更したユーザーの識別名
- `modifyTimestamp` : エントリの最終変更時刻を GMT (グリニッジ標準時) で記録したタイムスタンプ

注

クライアントアプリケーションから連鎖サフィックスのエントリを作成または変更した場合、属性 `creatorsName` と `modifiersName` には、エントリを実際に作成または変更したユーザーの識別名は反映されません。これらの属性には、リモートサーバーにバインドするために必要な連鎖プロキシの名前が含まれます。プロキシ承認については、103 ページの「プロキシ ID の作成」を参照してください。

レプリケートされたサフィックスの変更日時を追跡するときは、名前とタイムスタンプの属性は通常の属性としてレプリケートされます。このため、これらの属性は、エントリがコンシューマにレプリケートされた日時ではなく、マスターサーバー上のエントリに元の変更が適用された日時を反映します。

これらの情報を記録できるように Directory Server を設定するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで、設定ツリーのルートノードを選択します。次に、右側のパネルで「設定」タブを選択します。
2. 「エントリの変更時間を記録」チェックボックスを選択します。

   新しく作成されたエントリまたは変更されたエントリに対して、`creatorsName`、`createTimestamp`、`modifiersName`、および `modifyTimestamp` 属性が追加されます。既存のエントリには creation 属性は含まれません。
3. 「保存」をクリックして、サーバーを再起動します。

   詳細は、20 ページの「Directory Server の起動と停止」を参照してください。

プラグインの署名の検証

プラグイン署名の検証は、Directory Server 5.2 の新しい機能です。Directory Server が提供するプラグインは、それぞれがデジタル署名を持ち、サーバーの起動時にこれが検証されます。デフォルトでは、サーバーはプラグイン署名を検証します。ただし、署名の存在や有効性に関係なく、すべてのプラグインがロードされます。

署名の検証には、次のような利点があります。

- Directory Server が提供するプラグインの署名は、それが厳密にテストされ、正式にサポートされていることを示します。
- プラグインバイナリ自体のチェックサムを利用することで、署名の検証によって署名が改変されているかどうかを検出できます。このため、サーバー自体で実行される機密コードも署名によって保護されます。
- 署名されたプラグインだけをロードするようにサーバーを設定できます。これは、未署名のプラグインやサポートしていないプラグインによる問題の特定に役立ちます。

プラグイン署名の検証の設定

1. Directory Server コンソールの最上位にある「設定」タブで、設定ツリーから「プラグイン」ノードを選択します。右側のパネルに現在の署名検証ポリシーが表示されます。
2. 次のいずれかのオプションを選択します。
   - プラグインの署名を確認しない : 署名に関係なく、サーバーの設定に定義されているすべてのプラグインをロードします。プラグイン署名が原因で警告やエラーが表示されることはありません。
   - 署名が無効なプラグインにフラグをつける : サーバーの設定に定義されているすべてのプラグインをロードしますが、サーバーはそれぞれの署名を検証します。何らかの方法で、プラグインのバイナリが変更されているときは、署名は無効とされ、起動時にサーバーは警告メッセージを表示し、エラーログに記録します。署名を持たないプラグインにもフラグがつけられます。

     未署名のカスタムプラグインを利用するときは、このオプションの指定をお勧めします。このようなプラグインもロードされますが、同時にすべての署名つきプラグインの状態も確認できます。
   - 無効な署名のプラグインを拒否 : サーバーの設定に定義されているすべてのプラグインの署名を検証し、有効な署名を持つプラグインだけをロードします。サーバーの起動時に、どのプラグインが無効な署名を持つか、または署名を持たないかを示す警告メッセージが表示され、エラーログに記録されます。

これは最も安全なオプションですが、未署名のカスタムプラグインをロードできなくなります。

3. 「保存」をクリックし、20 ページの「Directory Server の起動と停止」で説明している方法でディレクトリサーバーを再起動します。

プラグインの状態の確認

1. Directory Server コンソールの最上位の「設定」タブの設定ツリーで「プラグイン」ノードを展開し、確認するプラグインを選択します。プラグインの現在の設定が右側のパネルに表示されます。
2. 「署名の状態」フィールドには、プラグイン署名の検証結果が次のいずれかの値で示されます。
   - 不明 : これは、プラグイン署名を検証しないようにサーバーを設定した場合にすべてのプラグインで表示される状態です。次の状態は、プラグイン署名の検証を行なった場合にだけ表示されます。
   - Valid signature : プラグインの設定は、署名がプラグインバイナリのチェックサムと一致することを示します。このプラグインは、正式にサポートされています。次の状態は、無効な署名にフラグをつけるが、それを拒否しない場合にだけ表示されます。
   - Invalid signature : プラグインの設定に、プラグインバイナリのチェックサムと一致しない署名が含まれます。この状態は、プラグインが改変されている可能性があることを示します。
   - No signature : プラグインの設定に、サーバーが検証する署名が含まれていません。

DSML の設定

Sun ONE Directory Server 5.2 は、LDAP (Lightweight Directory Access Protocol) で要求を処理するほか、DSMLv2 (Directory Service Markup Language バージョン 2) で送信された要求も処理します。クライアントでは、ディレクトリ操作をエンコードするために DSML も使われます。サーバーでは、DSML も他の要求と同様に、同じアクセス制御とセキュリティ機能によって処理されます。この DSML 処理により、さまざまな種類のクライアントがディレクトリの内容にアクセスできるようになります。

Directory Server では、ハイパーテキスト転送プロトコル (HTTP/1.1) で DSMLv2 を使用できます。また、DSML の内容を転送するためのプログラミングプロトコルとして SOAP (Simple Object Access Protocol) バージョン 1.1 が使われます。これらのプロトコルの詳細と、DSML 要求の例については、『Sun ONE Directory Server Deployment Guide』の付録 A 「Accessing Data using DSMLv2 over HTTP/SOAP」を参照してください。

DSML 要求の有効化

ディレクトリへのアクセスに使われる標準プロトコルは LDAP なので、Directory Server のインストール時にデフォルトでは、DSML 要求は無効になっています。HTTP/SOAP で送信された DSML 要求をサーバーで処理するには、この機能を明示的に有効にする必要があります。

DSML 要求をコンソールから有効にするには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで、設定ツリーのルートノードを選択します。次に、右側のパネルで「ネットワーク」タブを選択します。
2. 「DSML を有効」チェックボックスを選択し、次のいずれかのセキュリティオプションを選択します。セキュリティ保護されたポートのオプションは、第 11 章「セキュリティの実装」で説明する方法で SSL を有効化した場合にだけ表示されます。
   - セキュリティ保護されていないポートのみ : 暗号化されない HTTP 経由の DSML 要求だけが、セキュリティ保護されていないポートで受け付けられる
   - セキュリティ保護されたポートのみ : HTTPS 経由の DSML 要求だけが、セキュリティ保護されているポートで受け付けられる
   - セキュリティ保護されたポートと保護されていないポートの両方 : 両方のポートが有効となり、クライアントがいずれかを選択できる
3. 次のフィールドを編集します。
   - ポート : DSML 要求を受け取る HTTP ポート

- セキュリティ保護されたポート : 暗号化された DSML 要求を受け取る、SSL を使った HTTPS ポート
- 相対 URL : 相対 URL をホストとポートに付加することで得られる完全な URL は、DSML 要求を送信するためにクライアントで使われる

デフォルトでは、サーバーは次の URL に送信された要求を処理します。

`http://`*host*`:80/dsml`

4. 「保存」をクリックすると、DSML 要求の処理を開始するにはサーバーを再起動する必要があるというメッセージが表示されます。

DSML 要求をコマンド行から有効にするには、次の手順を実行します。

1. DSML フロントエンドプラグインを有効にしてその設定を変更するには、次の `ldapmodify` コマンドを実行します。`ds-hdsml-port`、`ds-hdsml-secureport`、および `ds-hdsml-rooturl` の各属性の変更は必要に応じて行います。

```
% ldapmodify -h host -p LDAPport -D "cn=Directory Manager" -w passwd
dn:cn=DSMLv2-SOAP-HTTP,cn=frontends,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
-
replace: ds-hdsml-port
ds-hdsml-port: DSMLport
-
add: ds-hdsml-secureport
ds-hdsml-port: secureDSMLport
-
replace: ds-hdsml-rooturl
ds-hdsml-root: relativeURL
-
^D
```

ユーザーによって定義されたパラメータと属性値に従い、DSML クライアントは次の URL を使って、このサーバーに要求を送信できます。

`http://`*host*`:`*DSMLport*`/`*relativeURL*

`https://`*host*`:`*secureDSMLport*`/`*relativeURL*

2. DSML フロントエンドプラグインを変更したら、変更を反映させるためにサーバーを再起動する必要があります。サーバーを再起動する前に、DSML 認証のセキュリティと ID マッピングを設定できます。これらについては、次の節で説明します。

DSML セキュリティの設定

前節で説明したセキュリティ保護されたポートの設定に加え、DSML 要求を受け入れるために必要なセキュリティレベルも設定できます。DSML フロントエンドプラグインの `ds-hdsml-clientauthmethod` 属性によって、必要なクライアントの認証方法が決まります。この属性に指定できる値は次のとおりです。

- `httpBasicOnly`: サーバーは HTTP `Authorization` ヘッダーの内容を使って、ディレクトリ内のエントリに対応付けるユーザー名を見つけます。このプロセスと設定についての詳細は、43 ページの「DSML の ID マッピング」を参照してください。この設定では、セキュリティ保護された HTTPS ポートへの DSML 要求は SSL で暗号化されますが、クライアント証明書は使われません。
- `clientCertOnly`: サーバーはクライアント証明書の資格情報を使ってクライアントを識別します。この設定では、DSML クライアントはすべて、セキュリティ保護された HTTPS ポートを使って DSML 要求を送信し、証明書を提示する必要があります。サーバーは、このクライアント証明書がディレクトリ内のエントリと一致するかどうかを確認します。クライアント証明書については、第 11 章「セキュリティの実装」を参照してください。
- `clientCertFirst`: クライアント証明書が提示された場合、サーバーはまずその証明書を使ってクライアントの認証を試みます。それ以外の場合は、`Authorization` ヘッダーの内容を使ってクライアントを認証します。

HTTP 要求に証明書も `Authorization` ヘッダーもない場合は、匿名バインドを使って DSML 要求を実行します。匿名バインドは、次の場合にも使われます。

- `clientCertOnly` が指定されている場合で、クライアントから有効な `Authorization` ヘッダーが提示されたが、証明書は提示されていないとき
- `httpBasicOnly` が指定されている場合で、クライアントから有効な証明書が提示されたが、`Authorization` ヘッダーは提示されていないとき

証明書が提示されていてもエントリと一致しない場合や、HTTP `Authorization` ヘッダーが指定されていてもユーザーのエントリに対応付けることができない場合、`ds-hdsml-clientauthmethod` 属性の値にかかわらず DSML 要求は拒否され、エラーメッセージ 403「Forbidden」が返されます。

DSML のセキュリティ要件をコンソールから設定するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで、設定ツリーのルートノードを選択します。次に、右側のパネルで「暗号化」タブを選択します。

   第 11 章「セキュリティの実装」で説明する方法で、SSL を設定し、有効化している必要があります。

2. 「DSML クライアント認証」フィールドのドロップダウンメニューからいずれかの項目を選択します。

3. 「保存」をクリックし、サーバーを再起動して新しいセキュリティ設定を適用します。

DSML のセキュリティ要件をコマンド行から設定するには、次の手順を実行します。

1. DSML フロントエンドプラグインの属性を編集するには、次の `ldapmodify` コマンドを実行します。

```
% ldapmodify -h host -p LDAPport -D "cn=Directory Manager" -w passwd
dn:cn=DSMLv2-SOAP-HTTP,cn=frontends,cn=plugins,cn=config
changetype: modify
replace: ds-hdsml-clientauthmethod
ds-hdsml-clientauthmethod: httpBasicOnly|
                           clientCertOnly|
                           clientCertFirst
-
^D
```

2. DSML フロントエンドプラグインを変更したら、この新しいセキュリティ設定を有効にするために、サーバーを再起動する必要があります。

DSML の ID マッピング

証明書を使わない基本認証を実行するときは、Directory Server は ID マッピングというメカニズムを使って、DSML 要求を受け入れるときに使うバインド DN を決定します。このメカニズムでは、HTTP 要求の `Authorization` ヘッダーから情報が抽出され、バインドに使う ID を決定します。このメカニズムの詳細については、391 ページの「ID マッピング」を参照してください。

DSML-over-HTTP のデフォルトの ID マッピングは、サーバー設定の次のエントリで指定されます。

```
dn:cn=default,cn=HTTP-BASIC, cn=identity mapping, cn=config
objectclass: top
objectclass: nsContainer
objectclass: dsIdentityMapping
cn: default
dssearchbasedn: ou=People,userRoot
dssearchfilter: (uid=${Authorization})
```

このマッピングでは、`ou=People,`*userRoot* サブツリーで、`Authorization` ヘッダーに指定されたユーザー名と一致する `uid` 属性を持つエントリが検索されます。*userRoot* は、ディレクトリのインストール時に定義したサフィックスです (`dc=example,dc=com` など)。

マッピングエントリの属性には、${*header*} という形式のプレースホルダを使用できます。ここで、*header* は HTTP ヘッダーの名前です。DSML マッピングでよく使われるヘッダーは次のとおりです。

- ${Authorization} : この文字列は、HTTP Authorization ヘッダーに格納されているユーザー名で置き換えられます。Authorization ヘッダーにはユーザー名とパスワードの両方が格納されていますが、このプレースホルダにはユーザー名だけが入ります。
- ${From} : この文字列は、HTTP From ヘッダーに格納されている電子メールアドレスで置き換えられます。
- ${host} : この文字列は、DSML 要求の URL に含まれるホスト名とポート番号 (サーバー自体のホスト名とポート番号) に置き換えられます。

DSML 要求で別の ID マッピングを実行するには、HTTP ヘッダーの ID マッピングを新しく定義します。

1. デフォルトの DSML-over-HTTP ID マッピングを編集するか、このプロトコル用のカスタムマッピングを作成します。ID マッピングエントリの属性の定義については、391 ページの「ID マッピング」を参照してください。これらのマッピングエントリは、次のエントリの下に配置する必要があります。cn=HTTP-BASIC, cn=identity mapping, cn=config

   新しいマッピングエントリは、次の 2 とおりの方法で作成できます。

   - Directory Server コンソールの最上位にある「ディレクトリ」タブを使って、適切なオブジェクトクラスを持つ新しいエントリを作成します。手順については、48 ページの「コンソールからのエントリの管理」を参照してください。
   - ldapmodify ツールを使って、コマンド行からこのエントリを追加します。手順については、66 ページの「ldapmodify によるエントリの追加」を参照してください。

2. 新しいマッピングを有効にするには、Directory Server を再起動します。

   最初にカスタムマッピングが評価され、どのカスタムマッピングも成功しない場合は、デフォルトのマッピングが評価されます。どのマッピングを使っても、DSML 要求の認証に使うバインド DN を特定できない場合、DSML 要求は禁止され拒否されます (エラー 403)。

第 2 章

ディレクトリエントリの作成

この章では、Directory Server コンソールと `ldapmodify` および `ldapdelete` コマンド行ユーティリティを使用して、構造化エントリ、ユーザーエントリ、リフェラルの基本タイプなど、ディレクトリの内容を変更する方法について説明します。また、Directory Server 5.2 の新機能である、オプションの属性暗号化機能を使って属性を保存する方法についても説明します。

ディレクトリの導入を計画する段階で、ディレクトリに格納するデータ形式の特徴を把握しておく必要があります。エントリを作成したり、デフォルトのスキーマを変更する前に、『Sun ONE Directory Server Deployment Guide』の第 2 章「Designing and Accessing Directory Data」を参照してください。

この章は、LDAP スキーマ、オブジェクトクラス、およびオブジェクトクラスに定義される属性について、ある程度の基本知識が読者にあることを前提としています。Directory Server のスキーマと、すべてのオブジェクトクラスと属性の定義に関する基本情報については、『Sun ONE Directory Server Reference Manual』の第 4 部「Directory Server Schema」を参照してください。

注	適切な ACI (アクセス制御命令) が定義されていない場合、ディレクトリは変更できません。詳細については、第 6 章「アクセス制御の管理」を参照してください。

この章は、次の節で構成されています。

- 設定エントリ
- コンソールからのエントリの管理
- コマンド行からのエントリの管理
- リフェラルの設定
- 属性値の暗号化
- 参照整合性の管理

設定エントリ

ディレクトリサーバーは、すべての設定情報を次のファイルに保存します。

ServerRoot/slapd-*serverID*/config/dse.ldif

このファイルの形式は LDIF (LDAP Data Interchange Format) で、LDAP エントリ、属性、およびその値がテキストとして表記されます。このファイルには、ディレクトリサーバーの次の設定が含まれます。

- cn=config エントリの属性と値
- cn=config の下のサブツリーに含まれるすべてのエントリと、その属性および属性値。多くの場合、エントリまたは属性の存在は重要です。
- ルートエントリ ("") と cn=monitor エントリのオブジェクトクラス、および ACI (アクセス制御命令)。これらのエントリのその他の属性は、サーバーによって生成されます。

Directory Server では、LDAP を通じてすべての設定を読み取り、書き込むことができます。デフォルトでは、ディレクトリの cn=config ブランチには、管理サーバーに定義されているディレクトリ管理者と、Directory Manager だけがアクセスできます。これらの管理ユーザーは、他のディレクトリエントリと同様に、設定エントリを表示、変更できます。

cn=config エントリの下のエントリは、通常のエントリのようなスケーラブルなデータベースとは異なる dse.ldif ファイルに格納されるため、cn=config の下にはエントリを作成しないでください。多くのエントリ、特に頻繁に更新されるエントリが cn=config の下に格納されている場合は、パフォーマンスが低下します。ただし、レプリケーションマネージャ (サプライヤバインド DN) などの特別なユーザーエントリを cn=config の下に格納しておくと、設定情報を集中管理できて便利です。

コンソールからの設定の変更

設定を変更するときは、Directory Server コンソールの最上位にある「設定」タブを使用することをお勧めします。このタブのパネルとダイアログには、タスクベースの制御が用意されており、迅速かつ効率的な設定に役立ちます。また、コンソールのインタフェースは、設定の複雑さや相互依存の解決に役立ちます。

このマニュアルで、コンソールのインタフェースを使った設定手順を説明するときは、「コンソールからの〜」という見出しで示されます。これらの手順は、「設定」タブのパネルとダイアログを使って特定の管理タスクを実行する方法を説明します。変更を適用するためにサーバーの再起動が必要な場合は、設定の保存方法がインタフェース自体にも明示されます。

コマンド行からの設定の変更

cn=config サブツリーには LDAP を通じてアクセスできるので、ldapsearch、ldapmodify、ldapdelete コマンドを使用して、サーバーの設定を表示、変更することができます。cn=config エントリとその下のすべてのエントリは、62 ページの「コマンド行からのエントリの管理」で説明する手順と LDIF 形式を使って変更できます。

ただし、これらのエントリの意味、および許容される属性と値の目的を理解しておくことは重要です。このマニュアルでは、「コマンド行からの〜」という見出しがつけられた手順で、重要な注意点を説明します。これらの手順には、設定エントリの例や、設定できる属性が示されます。すべての設定エントリとその属性、および属性に設定できる値の範囲について詳細は、『Sun ONE Directory Server Reference Manual』を参照してください。

コマンド行からの設定の変更は、コンソールを利用した変更ほど単純ではありません。しかし、一部の設定は、コンソールから変更することができず、コマンド行からの操作だけを受け付けます。また、コマンド行からの手順を利用する場合は、コマンド行ツールを使ったスクリプトを記述することで、設定タスクを自動化することができます。

dse.ldif ファイルの変更

dse.ldif ファイルには、サーバーの起動時または再起動時に読み取られ、適用される設定が含まれます。このファイルの LDIF コンテンツは、cn=config エントリとそのサブツリーです。ファイルの読み取りと書き込みが許可されているのは、インストール時に定義されたシステムユーザーだけです。

このファイルの内容を直接編集して設定を変更することは、エラーが生じる可能性が高くなるため、お勧めできません。次の点に注意が必要です。

- dse.ldif ファイルは起動時に一度だけ読み取られる読み取り専用ファイルです。このため、サーバー設定は、設定エントリのメモリ内の LDAP イメージに基づきます。したがって、起動後にファイルの内容を変更しても、次回の再起動時まで反映されません。
- コンソールまたはコマンド行から設定を変更すると、設定の LDAP イメージが変更されます。一部のディレクトリ機能は、呼び出されたときに現在の設定を読み取り、サーバーの再起動を必要としません。
- サーバーは、設定の LDAP イメージが変更されるたびに dse.ldif ファイルに書き込みを行います。一部のディレクトリ機能は、サーバーの起動時にその機能の設定だけを読み取り、変更が適用されるようにファイルに書き込みを行います。

既存の dse.ldif ファイルは dse.ldif.bak にコピーされ、既存の dse.ldif.bak ファイルは上書きされます。このため、dse.ldif ファイルに手動で変更を加えても、サーバーが再起動される前に LDAP によって設定が変更された場合は、手動による変更は失われます。

- ディレクトリの起動が成功すると、dse.ldif ファイルは毎回同じ場所の dse.ldif.startOK にコピーされます。誤った設定変更によってサーバーが起動できない場合は、このファイルから dse.ldif ファイルを復元する必要があります。

コンソールからのエントリの管理

Directory Server コンソールの「ディレクトリ」タブとエントリエディタダイアログを使用して、エントリの追加、変更、または削除を個別に行うことができます。複数のエントリに対して同時に処理を行う方法については、61 ページの「コンソールからの一括処理」を参照してください。

Directory Server コンソールの起動およびユーザーインタフェースの使用方法については、23 ページの「Directory Server コンソールの使用」を参照してください。

ディレクトリエントリの作成

Directory Server コンソールには、ディレクトリエントリの作成に使用できる、カスタムテンプレートがいくつか用意されています。それぞれのテンプレートは、オブジェクトクラスの種類に固有のカスタムエディタです。表 2-1 は、各カスタムエディタで使用されるオブジェクトクラスを示しています。

表 2-1 エントリテンプレートと対応するオブジェクトクラス

テンプレート	オブジェクトクラス
ユーザー	inetOrgPerson (作成および編集用) organizationalPerson (編集用) person (編集用)
グループ	groupOfUniqueNames およびその他のダイナミックグループおよび証明書グループ
組織単位	organizationalUnit
ロール	nsRoleDefinition、および管理されたロール、フィルタリングされたロール、または入れ子のロールのいずれかを選択するかに応じてその他のクラス

表 2-1　エントリテンプレートと対応するオブジェクトクラス (続き)

テンプレート	オブジェクトクラス
サービスクラス	cosSuperDefinition、およびサービスクラスのタイプに応じてその他のクラス
パスワードポリシー	passwordPolicy
リフェラル	referral

これらのカスタムエディタには、対応するオブジェクトクラスのすべての必須属性と、共通して使用される一部のオプション属性を表すフィールドが含まれています。いずれかのテンプレートを使用してエントリを作成する方法については、49 ページの「カスタムエディタを使用したエントリの作成」を参照してください。その他のタイプのエントリを作成する方法については、51 ページの「その他のタイプのエントリの作成」を参照してください。

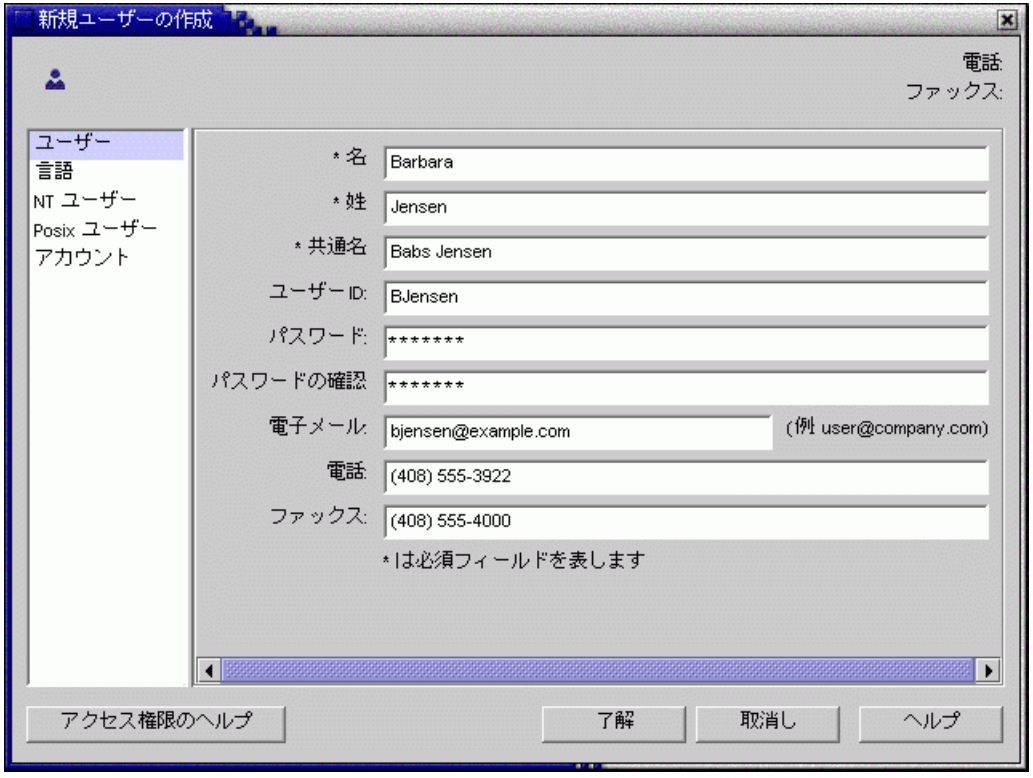
カスタムエディタを使用したエントリの作成

1. Directory Server コンソールの最上位にある「ディレクトリ」タブで、ディレクトリツリーを展開して、新しいエントリの親となるエントリを表示します。
2. 親エントリをマウスの右ボタンでクリックし、「新規」メニューを選び、サブメニューの「ユーザー」、「グループ」、「組織単位」、「ロール」、「サービスクラス」、「パスワードポリシー」、「リフェラル」の中からエントリのタイプを選択します。あるいは、親エントリをマウスの左ボタンでクリックして選択し、「オブジェクト」メニューから「新規」を選びます。選択したエントリタイプのカスタムエディタダイアログが表示されます。

   カスタムエディタの左側の列には複数のタブがあり、各タブのフィールドが右側に表示されます。デフォルトでは、すべてのカスタムエディタは、「ユーザー」タブまたは「一般」タブが選択された状態で表示されます。これらのタブには、新しいエントリの名前と説明を入力するためのフィールドが含まれています。

次の図はユーザーエントリのカスタムエディタを示しています。

図 2-1 Directory Server コンソール : ユーザーエントリのカスタムエディタ



3. カスタムエディタで、設定する属性のフィールドに値を入力します。フィールド名の隣にアスタリスク (*) が表示されたすべての必須属性には値を入力する必要があります。その他のフィールドには何も入力しなくても問題ありません。複数の値を入力できるフィールドでは、Return で値を区切ります。

   指定したエントリタイプのカスタムエディタの各フィールドについて、説明を表示するときは「ヘルプ」ボタンをクリックします。「ユーザー」エディタと「組織単位」エディタの「言語」タブの説明については、54 ページの「言語サポートの属性の設定」を参照してください。

   サービスエントリのグループ、ロール、クラスを作成する方法については、第 5 章「高度なエントリの管理」を参照してください。パスワードポリシーの作成については、第 7 章「ユーザーアカウントの管理」を参照してください。リフェラルの作成については、73 ページの「リフェラルの設定」を参照してください。

4. 「了解」をクリックして新しいエントリを作成し、カスタムエディタダイアログを閉じます。新しいエントリがディレクトリツリーに表示されます。

5. カスタムエディタダイアログには、対応するオブジェクトクラスのすべてのオプション属性のフィールドが表示されるわけではありません。カスタムエディタに表示されないオプション属性を追加する方法については、54 ページの「汎用エディタによるエントリの変更」を参照してください。

その他のタイプのエントリの作成

48 ページの表 2-1 に示されるオブジェクトクラス以外のオブジェクトクラスのエントリを作成するには、次の手順を実行します。この手順を実行して、ディレクトリスキーマに定義したカスタムオブジェクトクラスのエントリを作成することもできます。

1. Directory Server コンソールの最上位にある「ディレクトリ」タブで、ディレクトリツリーを展開して、新しいエントリの親となるエントリを表示します。
2. 親エントリをマウスの右ボタンでクリックして「新規」を選び、サブメニューから「その他」を選択します。あるいは、親エントリをマウスの左ボタンでクリックして選択し、「オブジェクト」メニューから「新規」、「その他」を順に選択することもできます。

   「新規オブジェクト」ダイアログが表示されます。
3. 「新規オブジェクト」ダイアログに表示されるオブジェクトクラスのリストから、新しいエントリを定義するオブジェクトクラスを選び、「了解」をクリックします。

   48 ページの表 2-1 に示されるオブジェクトクラスを選択した場合は、対応するカスタムエディタが表示されます (49 ページの「カスタムエディタを使用したエントリの作成」を参照)。それ以外の場合は、汎用エディタが表示されます。
4. 新規エントリを作成するときは、汎用エディタには選択しているオブジェクトクラスの必須属性に対応するフィールドが表示されます。すべての必須属性に値を設定する必要があります。一部のフィールドには、「新規」などの汎用のプレースホルダが表示されます。これは、作成するエントリに適した値で置き換える必要があります。
5. 選択しているオブジェクトクラスで利用できるその他の属性を定義するには、それを明示的に追加する必要があります。オプション属性の値を設定するには、次の手順を実行します。

   a. 「属性の追加」ボタンをクリックして、利用できる属性のリストを表示します。

   b. 「属性の追加」ダイアログで 1 つまたは複数の属性を選択し、「了解」をクリックします。

   c. 汎用エディタで、新しい属性の名前の隣に値を入力します。

   このダイアログのその他のコントロールについて詳細は、54 ページの「汎用エディタによるエントリの変更」を参照してください。

6. デフォルトでは、必須属性の 1 つがネーミング属性として選択され、汎用エディタにエントリの DN として表示されます。ネーミング属性を変更するには、次の手順を実行します。

   a. 「変更」ボタンをクリックして、「ネーミング属性の変更」ダイアログを表示します。

   b. 属性のテーブルで、新しいエントリの DN として使用する 1 つまたは複数の属性の隣にあるチェックボックスを選択します。

   c. 「了解」をクリックして「ネーミング属性の変更」ダイアログを閉じます。汎用エディタの DN には、選択したネーミング属性による新しい DN が表示されます。

7. 汎用エディタの「了解」をクリックして新しいエントリを保存します。

   新しいエントリは、ディレクトリツリー内の親エントリの子として表示されます。

カスタムエディタによるエントリの変更

48 ページの表 2-1 に示されるオブジェクトクラスでは、対応するカスタムエディタまたは汎用エディタを使ってエントリを編集できます。カスタムエディタを使う場合は、最も一般的なフィールドに簡単にアクセスできます。また、このインタフェースを使えば、ロールやサービスクラスの定義などに関連する複雑な属性も簡単に設定できます。

汎用エディタでは、オブジェクトクラスの追加、許可されている属性の追加、複数値属性の処理など、エントリに対してより高度な設定を行えます。汎用エディタを使ってエントリを編集する方法については、54 ページの「汎用エディタによるエントリの変更」を参照してください。

注　カスタムエディタは、48 ページの表 2-1 に示されるオブジェクトクラスだけの編集に使用できます。たとえば、`inetorgperson` から継承するカスタムクラスなど、その他の構造化オブジェクトクラスを含むエントリの編集には、汎用エディタを使う必要があります。

リストに含まれるオブジェクトクラスのほかに *auxiliary* オブジェクトクラスを含むエントリは、カスタムエディタを使って管理できます。ただし、auxiliary クラスによって定義される属性はカスタムエディタには表示されません。auxiliary オブジェクトクラスの定義については、『Sun ONE Directory Server Reference Manual』の第 9 章にある「Object Classes」を参照してください。

カスタムエディタの起動

オブジェクトクラスが 48 ページの表 2-1 に示されるエントリを編集するには、次の手順を実行します。

1. Directory Server コンソールの最上位にある「ディレクトリ」タブでディレクトリツリーを展開し、編集するエントリを表示します。
2. エントリをダブルクリックします。これ以外の方法でエントリのカスタムエディタを呼び出すこともできます。
   - エントリをマウスの右ボタンでクリックし、「カスタムエディタで編集」を選択する
   - エントリをマウスの左ボタンでクリックして選択し、「オブジェクト」メニューから「カスタムエディタで編集」を選択する
   - エントリをマウスの左ボタンでクリックして選択し、キーボードショートカットの Control-P を使用する

   エントリのオブジェクトクラスに対応するカスタムエディタが表示されます。たとえば、50 ページの図 2-1 はユーザーエントリのカスタムエディタを示しています。
3. デフォルトでは、すべてのカスタムエディタは、「ユーザー」タブまたは「一般」タブが選択された状態で表示されます。これらのタブには、新しいエントリの名前と説明を入力するためのフィールドが含まれます。カスタムエディタで、変更する属性のフィールドに値を入力するか、値を削除します。フィールド名の隣にアスタリスク (*) が表示された必須属性の値は、変更することはできますが、削除することはできません。その他のフィールドは何も入力しなくても問題ありません。複数の値を入力できるフィールドでは、Return で値を区切ります。

   左側の列のその他のタブを選択し、対応するパネルで値を変更します。指定したエントリタイプのカスタムエディタの各フィールドについて、説明を表示するときは「ヘルプ」ボタンをクリックします。

   「ユーザー」エディタと「組織単位」エディタの「言語」タブの説明については、54 ページの「言語サポートの属性の設定」を参照してください。ユーザーエントリまたはグループエントリの「アカウント」タブのフィールドについては、第 7 章「ユーザーアカウントの管理」を参照してください。「NT ユーザー」タブと「Posix ユーザー」タブが、Directory Server Synchronization Service 用に用意されています。詳細については、Sun の担当者までお問い合わせください。

   サービスエントリのグループ、ロール、クラスを変更する方法については、第 5 章「高度なエントリの管理」を参照してください。パスワードポリシーの変更については、第 7 章「ユーザーアカウントの管理」を参照してください。リフェラルの変更については、73 ページの「リフェラルの設定」を参照してください。
4. 「了解」をクリックしてエントリに加えた変更を保存し、カスタムエディタダイアログを閉じます。ユーザーエントリの共通名など、ネーミング属性を変更した場合は、ディレクトリツリーに変更が反映されます。

言語サポートの属性の設定

ユーザーエントリと組織単位エントリのカスタムエディタには、国際化ディレクトリ用に言語サポートが用意されています。

1. 53 ページの「カスタムエディタの起動」で説明している方法で、エントリのカスタムエディタを開きます。
2. 左側の列で「言語」タブをクリックします。
3. ユーザーエントリでは、ドロップダウンリストから適切な言語を選択できます。
4. ユーザーエントリと組織単位エントリのどちらでも、指定のフィールドに、リストに示される任意の言語を使ってローカライズされた値を入力できます。「利用可能な言語」リストから言語を選択し、1 つまたは複数の値をその言語で入力します。ローカライズされた値を定義すると、その言語がリストに太字で表示されます。

   一部の言語では、ローカライズされた値の発音表記のために、発音 (ふりがな) を入力するフィールドが表示されます。
5. 「了解」をクリックしてエントリに加えた変更を保存し、カスタムエディタダイアログを閉じます。

汎用エディタによるエントリの変更

汎用エディタでは、コンソールへのログインに使用したバインド DN に応じて、エントリの読み取り可能なすべての属性を表示し、書き込み可能なすべての属性を編集できます。また、属性の追加と削除、複数値属性の設定、エントリのオブジェクトクラスの管理も行えます。属性を追加するときは、バイナリ属性のサブタイプと言語サポートを定義できます。

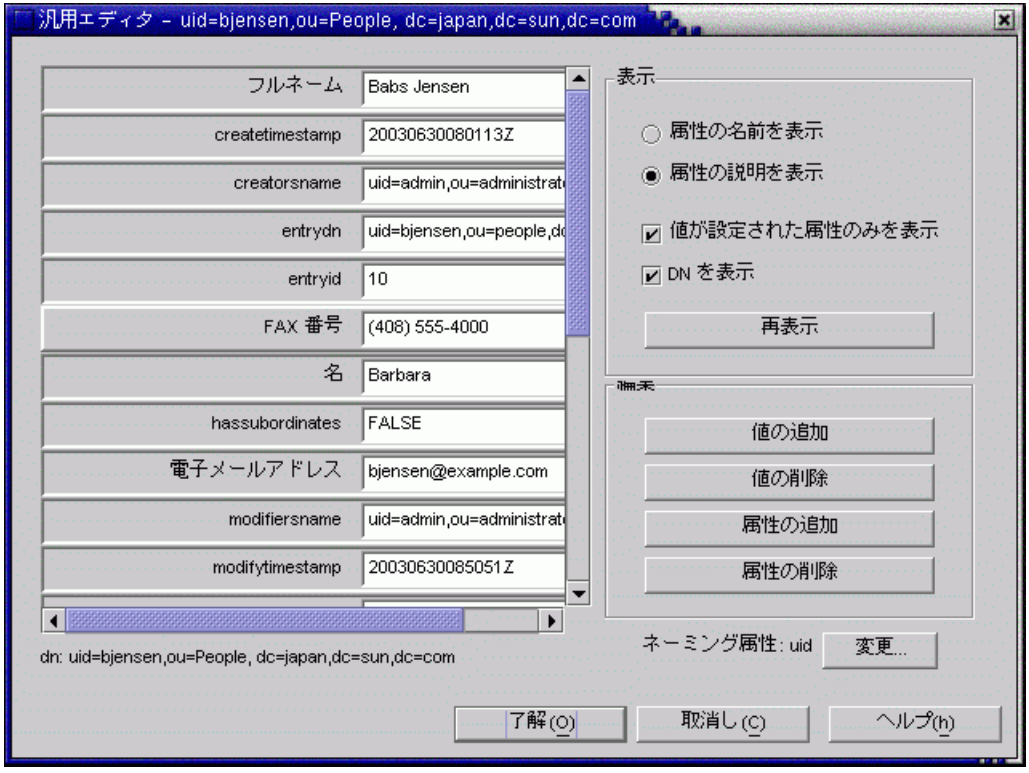
汎用エディタの起動

ディレクトリ内の任意のエントリの汎用エディタを呼び出すには、次の手順を実行します。

1. Directory Server コンソールの最上位にある「ディレクトリ」タブでディレクトリツリーを展開し、編集するエントリを表示します。
2. エントリをマウスの右ボタンでクリックし、「汎用エディタで編集」を選択します。これ以外の方法でも汎用エディタを呼び出すことができます。
   - エントリをマウスの左ボタンでクリックして選択し、「オブジェクト」メニューから「汎用エディタで編集」を選択します。
   - オブジェクトクラスが 48 ページの表 2-1 に示されていない場合は、エントリをダブルクリックします。カスタムエディタを持たないオブジェクトクラスの編集には、デフォルトで汎用エディタが使用されます。

次の図に示すように、汎用エディタが表示されます。

図 2-2　Directory Server コンソール : 汎用エディタ



汎用エディタでは、エントリの属性はアルファベット順に表示され、それぞれの値がテキストボックスに表示されます。読み取り専用属性やオペレーショナル属性などのすべての属性が表示されます。右側のコントロールを使うことで、エディタの表示を変更したり、属性のリストを編集することができます。

3. 必要に応じて、「表示」ボックスのコントロールを使って汎用エディタの表示を変更できます。

   - 属性の名前をスキーマに最初に定義したとおりに表示するときは、「属性の名前を表示」オプションを選択します。属性リストの表示が更新され、属性が名前のアルファベット順に表示されます。
   - スキーマに属性の別名が定義されている場合に、属性を別名順にリスト表示するときは、「属性の説明を表示」オプションを選択します。通常、別名は属性を明示的に説明します。属性リストの表示が更新され、属性が別名 (説明) のアルファベット順に表示されます。

- エントリのオブジェクトクラスのスキーマで明示的に許可されているすべての属性を表示するときは、「値が設定された属性のみを表示」チェックボックスの選択を解除します。エントリに `extensibleObject` オブジェクトクラスが含まれる場合、暗黙的にすべての属性が許可されますが、それは表示されません。デフォルトでは、値が定義されている属性だけが表示されます。
- 属性の下のエントリの識別名の表示と非表示を切り替えるときは、「DN を表示」チェックボックスを選択または選択解除します。
- 「再表示」ボタンをクリックすると、そのエントリの現在の内容に基づいて、すべての属性の値が更新されます。

警告 「再表示」ボタンをクリックすると、保存する前に汎用エディタで行なっていたすべての変更が直ちに削除されます。

属性値の設定、オブジェクトクラスの管理、エントリのネーミング属性の変更に関連するコントロールについては後述します。

属性値の変更

1. 54 ページの「汎用エディタの起動」で説明する方法で、汎用エディタを開きます。
2. 属性のリストをスクロールし、変更する値をクリックします。

   選択した属性が強調表示され、選択した値を含むテキストフィールドに編集カーソルが表示されます。
3. マウスとキーボードを使ってテキストを編集し、適切な値を入力します。システムのクリップボードを使って、このフィールドのテキストのコピー、カット、ペーストを行うことができます。

   テキストフィールドの内容を編集できないときは、その属性が読み取り専用であるか、値の変更に必要な書き込み権限がありません。
4. このエントリのその他の値を編集するか、その他の変更を加え、「了解」をクリックして変更を保存し、汎用エディタを閉じます。

複数値属性の編集

ディレクトリスキーマで複数値として定義されている属性は、汎用エディタの 1 つのフィールドに複数の値を設定できます。詳細は、第 9 章「ディレクトリスキーマの拡張」を参照してください。

複数値属性に新しい値を追加するには、次の手順を実行します。

1. 54 ページの「汎用エディタの起動」で説明する方法で、汎用エディタを開きます。

2. 属性のリストをスクロールし、属性またはその値をクリックします。選択した属性が強調表示され、「値の追加」ボタンが有効になります。このボタンが有効にならないときは、選択している属性が複数値として定義されていないか、読み取り専用である、または値の変更に必要な書き込み権限がありません。
3. 「値の追加」ボタンをクリックします。リスト内の属性名の隣に、新しい空白のテキストフィールドが表示されます。
4. 新しいテキストフィールドに、この属性の新しい値を入力します。システムのクリップボードを使って、このフィールドのテキストのコピー、カット、ペーストを行うことができます。
5. このエントリのその他の値を編集するか、その他の変更を加え、「了解」をクリックして変更を保存し、汎用エディタを閉じます。

複数値属性の値を削除するには、次の手順を実行します。

1. 54 ページの「汎用エディタの起動」で説明する方法で、汎用エディタを開きます。
2. 属性のリストをスクロールし、削除する値をクリックします。選択した属性が強調表示され、「値の削除」ボタンが有効になります。このボタンが有効にならないときは、選択している属性が読み取り専用であるか、または値の変更に必要な書き込み権限がありません。
3. 「値の削除」ボタンをクリックします。選択している値を含むテキストフィールドが削除されます。
4. このエントリのその他の値を編集するか、その他の変更を加え、「了解」をクリックして変更を保存し、汎用エディタを閉じます。

属性の追加

エントリに属性を追加するには、その属性を必須属性または許可された属性として持つオブジェクトクラスが、対象のエントリに含まれていることが必要です。詳細は、59 ページの「オブジェクトクラスの管理」および第 9 章「ディレクトリスキーマの拡張」を参照してください。

エントリに属性を追加するには、次の手順を実行します。

1. 54 ページの「汎用エディタの起動」で説明する方法で、汎用エディタを開きます。
2. 「値が設定された属性のみを表示」オプションが選択されていることを確認します。
3. 「属性の追加」ボタンをクリックして、属性のリストを示すダイアログを表示します。このリストには、そのエントリに定義されているオブジェクトクラスで使用できる属性だけが表示されます。
4. 「属性の追加」ダイアログで、追加する 1 つまたは複数の属性を選択します。

5. 必要に応じて、ダイアログ上部のドロップダウンリストから次のいずれか、または両方のサブタイプを選択できます。

   - 言語サブタイプ : 属性の値に適用される言語を指定するときは、このサブタイプを選択します。異なる言語で 1 つの属性を複数回追加し、ディレクトリにローカライゼーション情報を保存できます。

     オプションとして、言語のほかに「ふりがな」サブタイプを選択し、この属性の値に指定の言語の値に対応する発音表記が含まれていることを示すことができます。

   - バイナリサブタイプ : 属性にバイナリサブタイプを割り当てることによって、その属性値がバイナリデータであることを示します。バイナリサブタイプを選択しなくても属性にバイナリデータを格納することができますが、これを選択することで、クライアントに複数の属性タイプが存在する可能性を示すことができます。

6. 属性とオプションサブタイプの選択が完了したら、「了解」をクリックします。汎用エディタの属性のリストに、属性がアルファベット順に追加されます。

7. 新しい属性の名前の隣にある空白のテキストフィールドに、この属性の新しい値を入力します。システムのクリップボードを使って、このフィールドのテキストのコピー、カット、ペーストを行うことができます。

8. このエントリのその他の値を編集するか、その他の変更を加え、「了解」をクリックして変更を保存し、汎用エディタを閉じます。

属性の削除

属性とすべての値をエントリから削除するには、次の手順を実行します。

1. 54 ページの「汎用エディタの起動」で説明する方法で、汎用エディタを開きます。

2. 属性のリストをスクロールし、削除する属性の名前をクリックします。選択した属性が強調表示され、「属性の削除」ボタンが有効になります。このボタンが有効にならないときは、選択している属性が読み取り専用であるか、値の変更に必要な書き込み権限がありません。

注 汎用エディタでは、この属性に定義されているオブジェクトクラスが必要とする属性も削除できます。必須オブジェクトクラスを含まないエントリを保存しようとすると、サーバーはオブジェクトクラス違反を返します。すべてのオブジェクトクラスの必須属性がエントリに含まれることを確認してください。

---

3. 「属性の削除」ボタンをクリックします。属性と、その属性のすべてのテキストフィールドが削除されます。

4. このエントリのその他の値を編集するか、その他の変更を加え、「了解」をクリックして変更を保存し、汎用エディタを閉じます。

オブジェクトクラスの管理

エントリのオブジェクトクラスは、複数値の objectclass 属性によって定義されます。この属性を変更する場合に、定義されているオブジェクトクラスを管理できるように、汎用エディタには特別なダイアログがあります。

オブジェクトクラスをエントリに追加するには、次の手順を実行します。

1. 54 ページの「汎用エディタの起動」で説明する方法で、汎用エディタを開きます。
2. 属性のリストをスクロールし、オブジェクトクラスまたは objectclass 属性を選択します。「値の追加」ボタンが有効になります。このボタンが有効にならないときは、このエントリのオブジェクトクラスの変更に必要な権限がありません。
3. 「値の追加」ボタンをクリックします。

   「オブジェクトクラスの追加」ダイアログが表示されます。このウィンドウには、エントリに追加できるオブジェクトクラスのリストが表示されます。
4. このエントリに追加するオブジェクトクラスを 1 つまたは複数選択し、「了解」をクリックします。選択したオブジェクトクラスが、objectclass 属性の値のリストに表示されます。
5. 新しいオブジェクトクラスが、エントリに含まれない属性を必要とする場合は、汎用エディタはそれを自動的に追加します。すべての必須属性に値を設定する必要があります。
6. このエントリのその他の値を編集するか、その他の変更を加え、「了解」をクリックして変更を保存し、汎用エディタを閉じます。

エントリからオブジェクトクラスを削除するには、次の手順を実行します。

1. 54 ページの「汎用エディタの起動」で説明する方法で、汎用エディタを開きます。
2. 属性のリストをスクロールし、削除する objectclass 属性の値をクリックします。選択しているオブジェクトクラスの削除がスキーマで許可され、このエントリのオブジェクトクラスを変更する権限がある場合は、「値の削除」ボタンが有効になります。
3. 「値の削除」ボタンをクリックします。指定したオブジェクトクラスが削除されます。

   オブジェクトクラスを削除すると、汎用エディタは残りのオブジェクトクラスが許可しないか、必要としないすべての属性を自動的に削除します。いずれかのネーミング属性が削除されると、別のネーミング属性が自動的に選択されます。コンソールは、この変更を示すメッセージを表示します。

4. このエントリのその他の値を編集するか、その他の変更を加え、「了解」をクリックして変更を保存し、汎用エディタを閉じます。

エントリ名の変更

ネーミング属性は、識別名 (DN) に表示されるエントリの属性値のペアです。ネーミング属性は、エントリの既存の属性から選択されます。ネーミング属性を変更してエントリの名前を変更するには、次の手順を実行します。

1. 54 ページの「汎用エディタの起動」で説明する方法で、汎用エディタを開きます。

   「変更」ボタンの隣のテキストは、このエントリの現在のネーミング属性を示します。「DN を表示」チェックボックスが選択されている場合、これらの属性が属性値リストの下の DN に表示されます。

2. 「変更」ボタンをクリックします。このボタンが有効にならないときは、このエントリの名前を変更する権限がありません。

   「ネーミング属性の変更」ダイアログが表示されます。

3. 属性のリストをスクロールし、このエントリの DN にする属性を選択します。属性の隣のチェックボックスを選択してネーミング属性に追加するか、選択解除して削除します。

   同じ親の下のエントリの DN は、一意なものにする必要があります。このため、値または値の組み合わせが一意となるネーミング属性を選択する必要があります。DN が重複している場合、そのエントリを保存しようとすると、サーバーがこれを拒否します。たとえば、ユーザーを示すすべてのエントリでは、同じネーミング属性を使用する必要があります。

4. 「了解」をクリックして「ネーミング属性の変更」ダイアログを閉じます。汎用エディタには、このエントリの新しい DN が表示されます。

5. このエントリのその他の値を編集するか、その他の変更を加え、「了解」をクリックして変更を保存し、汎用エディタを閉じます。

ディレクトリエントリの削除

Directory Server コンソールを使用してディレクトリエントリを削除するには、次の手順を実行します。

1. Directory Server コンソールの最上位にある「ディレクトリ」タブでディレクトリツリーを展開し、削除するエントリを表示します。

   サブツリーのルートノードを選択することで、ディレクトリのブランチ全体を削除することもできます。

2. エントリをマウスの右ボタンでクリックし、「削除」を選択します。これ以外の方法でエントリを削除することもできます。

   - エントリをマウスの左ボタンでクリックして選択し、「編集」メニューから「削除」を選択します。このエントリをディレクトリ内の別の場所にペーストするときは、「編集」メニューから「カット」を選択することもできます。
   - エントリをマウスの左ボタンでクリックして選択し、キーボードショートカットの Control-D を使用します。

   「表示」メニューから「レイアウト」オプションを選択して Directory Server コンソールの右側のパネルに子を表示しているときは、Control または Shift を押しながらクリックすることで、複数のエントリを選択できます。

3. エントリまたはサブツリーとそのすべての内容を削除することを確認します。

   選択したエントリがただちに削除されます。この処理を元に戻すことはできません。複数のエントリを削除した場合、削除したエントリの数を示すダイアログが表示されます。また、削除時にエラーが発生した場合は、エラーを示すダイアログが表示されます。

コンソールからの一括処理

LDIF ファイルを使用することで、複数エントリの追加、組み合わせ操作の実行、サフィックス全体のインポートを行うことができます。LDIF ファイルと Directory Server コンソールを使用してエントリを追加するには、次の手順を実行します。

1. 前述の項に示される構文を使って LDIF ファイルにエントリまたは操作を定義します。エントリを追加するだけ、またはサフィックスを初期化するだけの処理では、`changetype` キーワードは必要ありません。エントリだけを LDIF ファイルに指定します。組み合わせ操作を実行するときは、すべての DN に `changetype` を続け、必要に応じて特定の処理または属性値を指定します。

2. Directory Server コンソールを使用して、LDIF ファイルをインポートします。詳細は、135 ページの「LDIF ファイルのインポート」を参照してください。

   組み合わせ操作を実行するときは、サーバーがすべての LDIF 処理を実行できるように、「LDIF のインポート」ダイアログで「追加のみ」が選択されていないことを確認します。

コマンド行からのエントリの管理

コマンド行ユーティリティ ldapmodify および ldapdelete には、ディレクトリの内容を追加、編集、削除するための完全な機能が用意されています。これらを使用して、サーバーの設定エントリと、ユーザーエントリに含まれるデータの両方を管理できます。これらのユーティリティは、1 つまたは複数のディレクトリの一括管理を実行するためのスクリプトの作成にも利用できます。

ldapmodify コマンドと ldapdelete コマンドは、このマニュアル全体の手順で使用されます。次に、これらの管理手順の実行に必要なすべての基本操作について説明します。機能の詳細、すべてのコマンド行オプション、これらのコマンドの戻り値については、『Sun ONE Directory Server Resource Kit Tools Reference』の第 4 章「ldapmodify」および第 5 章「ldapdelete」を参照してください。

コマンド行ユーティリティへの入力は、常に LDIF (LDAP Data Interchange Format) 形式で行います。この形式の入力は、コマンド行に直接指定できるだけでなく、入力ファイルからも行うことができます。LDIF は、エントリ、属性、およびその値をテキストで表現します。LDIF は、RFC 2849 (http://www.ietf.org/rfc/rfc2849.txt) に定義されている標準形式です。次の項では、LDIF 入力について説明し、それ以降の項では各種変更処理で使われる LDIF について説明します。

LDIF 入力の供給

コマンド行ユーティリティに LDIF 入力を行うときは、コマンド行入力、特殊文字、スキーマ検査、エントリの順序とサイズについて特別な注意を払う必要があります。

コマンド行での LDIF 入力の終了

ldapmodify ユーティリティと ldapdelete ユーティリティは、ファイルから読み取るのとまったく同様に、ユーザーがコマンドの後に入力した LDIF 文を読み取ります。入力が終了したら、ファイルの最後 (EOF) を示すエスケープシーケンスとしてシェルに認識される文字を入力します。

使用しているオペレーティングシステムに応じて、通常は次のいずれかが EOF エスケープシーケンスとなります。

- UNIX : ほとんどの場合 Control+D (^D)
- Windows NT : 通常は Control+Z の後にキャリッジリターン (^Z<Return>)

次の例は、UNIX システムでの ldapmodify コマンドの入力の終了を示しています。

```
prompt> ldapmodify -h host -p port -D bindDN -w password
dn: cn=Barry Nixon,ou=People,dc=example,dc=com
changetype: modify
delete: telephonenumber
^D
prompt>
```

表記を単純かつわかりやすくするために、このマニュアルの例には EOF シーケンスは表示されません。

特殊文字の使い方

コマンド行にコマンドオプションを指定するときは、コマンド行インタープリタにとって特別な意味を持つエスケープ文字の入力が必要になることがあります。このような文字には、空白 ( )、アスタリスク (*)、円記号 (¥) などが含まれます。たとえば、多くの DN には空白文字が含まれ、ほとんどの UNIX シェルでは値を二重引用符 ("") で囲む必要があります。

```
-D "cn=Barbara Jensen,ou=Product Development,dc=example,dc=com"
```

一重引用符または二重引用符のどちらを使用するかは、コマンド行インタープリタのタイプによって異なります。詳細は、オペレーティングシステムのマニュアルを参照してください。

さらに、コンマを含む DN を使用する場合は、円記号 (¥) でコンマをエスケープする必要があります。次に例を示します。

```
-D "cn=Patricia Fuentes,ou=People,o=example.com Bolivia¥,S.A."
```

`ldapmodify` コマンドの後の LDIF 文はシェルではなく、コマンドによって解釈されるため、特別な注意が必要ないことに注意してください。

スキーマ検査

エントリを追加または変更する場合、使用する属性は、エントリのオブジェクトクラスが必要とするか、許可する属性である必要があり、その属性には、定義されている構文に準拠する値が含まれている必要があります。

エントリを変更すると、Directory Server は変更される属性だけでなく、エントリ全体に対してスキーマ検査を行います。このため、エントリのいずれかのオブジェクトクラスまたは属性がスキーマに準拠していない場合、変更処理は失敗します。詳細は、335 ページの「スキーマ検査」を参照してください。

LDIF エントリの順序

エントリを追加するための LDIF テキストのシーケンスでは、コマンド行に指定する場合も、ファイルに指定する場合も、親エントリを子の前に指定する必要があります。これにより、サーバーが LDIF テキストを処理するときに、子エントリの前に親エントリが作成されます。

たとえば、ディレクトリに存在しない People サブツリーにエントリを作成する場合、サブツリー内のエントリの前に People コンテナを表すエントリを指定します。

```
dn: dc=example,dc=com
dn: ou=People,dc=example,dc=com
...
People サブツリーのエントリ
...
dn: ou=Group,dc=example,dc=com
...
Group サブツリーのエントリ
...
```

`ldapmodify` コマンド行ユーティリティを使ってディレクトリ内にエントリを作成することができますが、サフィックスまたはサブサフィックスのルートは、必要な設定エントリと関連づける必要のある特別なエントリです。新しいルートサフィックスまたはサブサフィックス、およびそれに関連する設定エントリを追加する手順については、94 ページの「コマンド行からのサフィックスの作成」を参照してください。

大規模なエントリの管理

極端に大きな属性値を持つエントリを追加または変更するときは、それを受け入れることができるように、事前にサーバーの設定が必要になることがあります。サーバーのオーバロードを防ぐために、デフォルトでは、クライアントは 2M バイトを超えるデータを送信できないように制限されています。

これを超えるエントリを追加するか、属性をこれ以上の値に変更しようとすると、サーバーはその処理を拒否し、直ちに接続を閉じます。たとえば、1 つのエントリの 1 つまたは複数の属性にマルチメディアコンテンツなどのバイナリデータが含まれると、この制限を超える可能性があります。

また、多数のメンバーを含む大規模なスタティックグループを定義するエントリも、この制限を超える可能性があります。ただし、パフォーマンスを考慮すると、このようなグループはお勧めできません。ディレクトリ構造の再設計を考慮する必要があります。詳細は、154 ページの「グループの管理」を参照してください。

クライアントが送信するデータにサーバーが適用するサイズ制限を変更するには、次の手順を実行します。

1. `cn=config` エントリの `nsslapd-maxbersize` 属性に新しい値を設定します。

- この処理をコンソールから行うには、管理者または Directory Manager としてログオンし、54 ページの「汎用エディタによるエントリの変更」で説明する方法で、cn=config エントリを編集します。nsslapd-maxbersize 属性の値を、クライアントが一度に送信できる最大バイト数に設定します。
- この処理をコマンド行から行うには、次のコマンドを実行します。

  ```
  ldapmodify -h host -p port -D "cn=Directory Manager" -w password
  dn: cn=config
  changetype: modify
  replace: nsslapd-maxbersize
  nsslapd-maxbersize: sizeLimitInBytes
  ```

  詳細については、『Sun ONE Directory Server Reference Manual』の第 4 章にある「nsslapd-maxbersize」を参照してください。

2. 20 ページの「Directory Server の起動と停止」で説明している手順を実行してサーバーを再起動します。

エラーの処理

コマンド行ツールは、LDIF 入力に含まれるすべてのエントリまたは変更を順番に処理します。最初のエラーが発生した場合のデフォルトの対応は、処理の停止です。エラーに関係なくすべての入力の処理を継続するときは、-c オプションを指定します。エラーの状態は、ツールの出力に表示されます。

上記注意点のほかに一般的なエラーには、次のようなものがあります。

- 実行する操作に必要なアクセス権がない
- ディレクトリにすでに存在する DN が指定されたエントリを追加しようとする
- 存在しない親の下にエントリを追加しようとする

エラー状態とそれを回避する方法について詳細は、『Sun ONE Directory Server Resource Kit Tools Reference』の第 4 章「ldapmodify」および第 5 章「ldapdelete」を参照してください。

ldapmodify によるエントリの追加

ldapmodify の -a オプションを使って、ディレクトリに 1 つまたは複数のエントリを追加できます。次の例では、ユーザーを含む構造化エントリを作成し、次にユーザーエントリを作成します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: ou=People,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
ou: People
description: Container for user entries

dn: uid=bjensen,ou=People,dc=example,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgPerson
uid: bjensen
givenName: Barbara
sn: Jensen
cn: Babs Jensen
telephoneNumber: (408) 555-3922
facsimileTelephoneNumber: (408) 555-4000
mail: bjensen@example.com
userPassword: clearPassword
```

-D オプションと -w オプションは、これらのエントリの作成に必要な権限を持つユーザーのバインド DN とパスワードを指定します。-a オプションは、LDIF に指定されているすべてのエントリが追加されることを示します。各エントリには DN と属性値が指定され、エントリとエントリの間には空白行が挿入されます。ldapmodify ユーティリティは、入力されるすべてのエントリを順番に作成し、エラーが発生した場合は、それをレポートします。

慣例により、エントリの LDIF には、次の順序で属性が指定されます。

- オブジェクトクラスのリスト
- 1 つまたは複数のネーミング属性。これは DN で使用される属性で、必須属性である必要はない
- すべてのオブジェクトクラスの必須属性
- エントリに指定する、許可されているその他の属性

userpassword 属性の値を入力するときは、パスワードをクリアテキストで指定します。サーバーはこの値を暗号化し、暗号化された値だけが格納されます。LDIF ファイルに表示されるクリアテキストのパスワードを保護するために、読み取りアクセス権を制限してください。

-a オプションを必要としない、別の形式の LDIF をコマンド行に指定することもできます。この形式の利点は、エントリを追加する文と、次の項で説明するエントリを変更する文を組み合わせて指定できることです。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: ou=People,dc=example,dc=com
changetype: add
objectclass: top
objectclass: organizationalUnit
ou: People
description: Container for user entries

dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: add
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgPerson
uid: bjensen
givenName: Barbara
sn: Jensen
cn: Barbara Jensen
telephoneNumber: (408) 555-3922
facsimileTelephoneNumber: (408) 555-4000
mail: bjensen@example.com
userPassword: clearPassword
```

changetype: add キーワードは、指定の DN を持つエントリが、それ以後のすべての属性を持った状態で作成されることを示します。それ以外のすべてのオプションと LDIF の表記は同じです。

どちらの例でも、-f *filename* オプションを使うことで、端末からの入力の代わりにファイルから LDIF を読み取ることができます。LDIF ファイルには、-a オプションを使用した場合、端末からの入力と同じ形式で情報を指定する必要があります。

ldapmodify によるエントリの変更

既存のエントリの属性と属性値を追加、置換、または削除するときは、changetype: modify キーワードを使います。changetype: modify を指定する場合は、エントリの変更方法を示す、1 つまたは複数の変更操作も指定する必要があります。次の例には、3 種類の LDIF 変更操作が指定されています。

```
dn: entryDN
changetype: modify
add: attribute
attribute: value
...
-
replace: attribute
attribute: newValue
...
-
delete: attribute
[attribute: value]
...
```

同じエントリに対する操作を区切るときはハイフン (-) を使い、異なるエントリに対する操作セットを区切るときは空白文字を使います。各操作の対象となる *attribute*: *value* のペアを複数指定して、それを一度に追加、置換、または削除することもできます。

属性値の追加

次の例は、同じ add LDIF 文を使って、既存の複数値属性と、まだ存在しない属性に値を追加する方法を示しています。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: cn
cn: Babs Jensen
-
add: mobile
mobile: (408) 555-7844
mobile: (408) 555-7845
```

次の場合は、処理が失敗し、エラーが返されることがあります。

- 指定した値がその属性にすでに存在する
- 値が、属性に定義されている構文に準拠していない

- エントリのオブジェクトクラスが、その属性タイプを必要としないか、許可しない
- 属性タイプが複数値ではなく、その属性にすでに値が存在する

バイナリ属性値の追加

バイナリ属性値は、*attribute*;binary サブタイプによって示されます。サブタイプは必要ありませんが、ユーザーとクライアントが属性の内容を知る上で便利です。ldapmodify コマンドで使用するどの LDIF 文でも、属性名に適切なサブタイプを追加できます。

バイナリ値を入力するには、LDIF テキストに直接入力するか、別のファイルから読み取ります。次の例は、ファイルから読み取る LDIF の構文を示しています。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
version: 1
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: jpegphoto;binary
jpegphoto;binary: < file:///path/filename.jpg
```

< の前後の空白文字は重要です。ここに示されるとおりに指定する必要があります。ファイル名の指定に < 構文を利用するには、LDIF 文を version: 1 という行から開始する必要があります。ldapmodify がこの文を処理するときに、このツールは、指定ファイルの内容全体から読み取った値を属性に設定します。

言語サブタイプを持つ属性の追加

属性の言語とふりがなのサブタイプは、ローカライズされた値を特定します。属性に対して言語サブタイプを指定すると、そのサブタイプが属性名に次のように追加されます。

attribute;lang-*CC*

ここで、*attribute* は既存の属性タイプを示し、*CC* は言語を特定する 2 文字の国コードを示します。オプションとして、言語サブタイプにふりがなのサブタイプを追加し、ローカライズされた値の発音表記を指定することもできます。この場合、属性名は次のようになります。

attribute;lang-*CC*;phonetic

サブタイプを持つ属性に対して処理を行うには、そのタイプを明示的に一致させる必要があります。たとえば、lang-ja の言語サブタイプを持つ属性値を変更する場合は、次の例に示すように、変更操作に lang-ja を含める必要があります。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
replace: homePostalAddress;lang-ja
homePostalAddress;lang-fr: 34¥, avenue des Champs-Elysées
```

属性値の変更

次の例は、LDIF の replace 文を使って、単一値の属性と、複数値属性のすべての値を変更する方法を示しています。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
replace: sn
sn: Morris
-
replace: cn
cn: Barbara Morris
cn: Babs Morris
```

replace 文を使うときは、指定した属性のすべての現在値が削除され、指定した値が追加されます。

属性値の削除

次の例は、属性全体、または複数値属性の 1 つの値だけを削除する方法を示しています。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
delete: facsimileTelephoneNumber
-
delete: cn
cn: Babs Morris
```

attribute: value のペアを指定せずに delete 構文を使用すると、属性のすべての値が削除されます。*attribute: value* のペアを指定した場合は、その値だけが削除されます。

複数値属性の 1 つの値の変更

ldapmodify コマンドを使って、複数値属性の 1 つの値を変更するには、次の例に示すように、2 段階の処理が必要です。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
delete: mobile
mobile: (408) 555-7845
-
add: mobile
mobile: (408) 555-5487
```

ldapmodify によるエントリ名の変更

エントリの名前を変更するときは、関連識別名 (RDN) を変更します。RDN は、エントリの DN に含まれる、いちばん左の *attribute=value* のペアです。この属性はネーミング属性と呼ばれ、エントリのすべての属性に同じ値が存在する必要があります。

エントリの名前を変更するときは、エントリが別のサブツリーに移動するような、DN の別の部分を変更することはできません。エントリを異なるブランチに移動するには、別のサブツリー内にそのエントリの属性を使用して新しいエントリを作成してから、元のエントリを削除する必要があります。

また、親の RDN は子の DN で使われており、すべてのエントリに DN が含まれる必要があるため、子を持つエントリの名前を変更することはできません。ツリー全体を移動するには、新しい場所でそのツリーを再作成する必要があります。

LDIF 文を使ってエントリの名前を変更するには、`changetype: modrdn` キーワードを使います。次の例では、Barbara Morris の `uid` ネーミング属性の名前を変更します。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modrdn
newrdn: uid=bmorris
deleteoldrdn: 1
```

`newrdn` 行は、*attribute=value* 構文を使って新しいネーミング属性を指定します。`deleteoldrdn` 行は、同時に以前のネーミング属性を削除するかどうかを指定します (`1` であれば削除、`0` の場合は削除しない)。いずれの場合も、エントリに新しいネーミング属性が追加されます。

ldapdelete によるエントリの削除

ディレクトリからエントリを削除するときは、ldapdelete コマンド行ユーティリティを使います。このユーティリティは、ディレクトリサーバーにバインドし、DN によって指定される 1 つまたは複数のエントリを削除します。指定のエントリを削除する権限を持つバインド DN を指定する必要があります。

親エントリの名前を変更できないのと同じ理由で、子を持つエントリを削除することはできません。LDAP プロトコルでは、親を持たない子エントリが存在する状況を禁止しています。たとえば、組織単位に属するすべてのエントリを先に削除しない限り、組織単位エントリは削除できません。

警告 サフィックス o=NetscapeRoot は削除しないでください。Sun ONE 管理サーバーは、このサフィックスを使用してインストールした Sun ONE サーバーに関する情報を格納します。このサフィックスを削除すると、Directory Server を含むすべての Sun ONE サーバーの再インストールが必要になります。

次の例では、組織単位には 1 つのエントリしか含まれていないため、そのエントリを削除すれば、親エントリを削除できます。

```
ldapdelete -h host -p port -D "cn=Directory Manager" -w password
uid=bjensen,ou=People,dc=example,dc=com
ou=People,dc=example,dc=com
```

ldapmodify によるエントリの削除

changetype: delete キーワードを利用することで、ldapmodify ユーティリティを使ってエントリを削除できます。この場合も、前述の ldapdelete と同じ制限が適用されます。LDIF 構文を使ってエントリを削除する利点は、1 つの LDIF ファイルで複数の処理を組み合わせて実行できることです。

次の例は、前述の例と同じ削除処理を行います。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: delete

dn: ou=People,dc=example,dc=com
changetype: delete
```

リフェラルの設定

情報をローカルに取得できない場合に、どのサーバーに接続すべきかをクライアントアプリケーションに通知するには、リフェラルを使います。リフェラルとは、リモートサフィックスへのポインタ、つまり Directory Server が結果の代わりにクライアントへ返すエントリへのポインタです。クライアントは、リフェラルで指定されたリモートサーバー上で、再度、操作を実行する必要があります。このリダイレクションは、次の 3 つの場合に行われます。

- クライアントアプリケーションがローカルサーバーに存在しないエントリを要求し、サーバーがデフォルトのリフェラルを返す場合
- サフィックス全体がメンテナンスまたはセキュリティ上の理由でオフラインになり、サーバーがサフィックスに定義されているリフェラルを返す場合。サフィックスレベルのリフェラルについては、98 ページの「アクセス権とリフェラルの設定」を参照。クライアントが書き込み処理を要求する場合、サフィックスの読み取り専用レプリカも、マスターサーバーにリフェラルを返す
- スマートリフェラルと呼ばれるエントリを作成できる。クライアントがスマートリフェラルにアクセスすると、サーバーは定義されているリフェラルを返す。Directory Server コンソールは、スマートリフェラルを自動的にたどるため、最上位の「ディレクトリ」タブでローカルエントリのように見える

いずれの場合も、リフェラルは LDAP URL であり、ホスト名、ポート番号、およびオプションとして別のサーバー上の DN を含みます。詳細については、『Sun ONE Directory Server Reference Manual』の付録 D「LDAP URL」を参照してください。ディレクトリの導入におけるリフェラルの使用方法については、『Sun ONE Directory Server Deployment Guide』を参照してください。

次に、ディレクトリのデフォルトリフェラルを定義する手順と、スマートリフェラルを定義する手順について説明します。

デフォルトリフェラルの設定

デフォルトリフェラルは、ディレクトリで管理されているサフィックスのどれにも含まれない DN に対して、操作を送信するクライアントアプリケーションに返されます。デフォルトリフェラルはディレクトリ内のすべてのサフィックスに適用されるため、グローバルリフェラルとも呼ばれます。サーバーは定義されているすべてのリフェラルを返しますが、返す順序は定義されていません。

コンソールからのデフォルトリフェラルの設定

1. Directory Server コンソールの最上位の「設定」タブで、設定ツリーのルートノードを選択し、右側のパネルで「ネットワーク」タブを選択します。

2. 「返すリフェラル」チェックボックスを選択し、テキストフィールドに LDAP URL を入力します。あるいは、「構成」をクリックして、指示に従って LDAP URL を定義します。次に、セキュリティ保護されているポートへの LDAP URL の例を示します。

   ```
   ldaps://east.example.com:636/dc=example,dc=com
   ```

   複数のリフェラル URL を入力するには、次のように空白で区切って、それぞれを引用符で囲みます。

   ```
   "ldap://east.example.com:389/" "ldap://backup.example.com:389/"
   ```

3. 「保存」をクリックして変更を保存します。変更は直ちに適用されます。

コマンド行からのデフォルトリフェラルの設定

ディレクトリ設定ファイルの `cn=config` エントリに 1 つまたは複数のデフォルトリフェラルを追加するか、交換するときは、`ldapmodify` コマンド行ユーティリティを使います。次に例を示します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=config
changetype: modify
replace: nsslapd-referral
nsslapd-referral: ldap://east.example.com:389/
nsslapd-referral: ldap://backup.example.com:389/
```

サーバーを再起動する必要はありません。

スマートリフェラルの作成

スマートリフェラルを使用して、ディレクトリエントリおよびディレクトリツリーを、特定の LDAP URL に割り当てることができます。スマートリフェラルを使用すると、クライアントアプリケーションに、特定のサーバーや特定のサーバーにある特定のエントリを参照させることができます。

多くの場合、スマートリフェラルは別のサーバー上の同じ DN を持つ実際のエントリを指しています。ただし、同じサーバーまたは別のサーバーのあらゆるエントリに対するスマートリフェラルを定義できます。たとえば、次の DN を持つエントリを定義することができます。

```
uid=bjensen,ou=People,dc=example,dc=com
```

この場合、スマートリフェラルは `east.example.com` というサーバー上の次のエントリを指しています。

```
cn=Babs Jensen,ou=Sales,o=east,dc=example,dc=com
```

ディレクトリがスマートリフェラルを使用する方法は、RFC 2251 (`http://www.ietf.org/rfc/rfc2251.txt`) のセクション 4.1.11 に指定されている標準に準拠する必要があります。

コンソールからのスマートリフェラルの作成

1. Directory Server コンソールの最上位にある「ディレクトリ」タブでディレクトリツリーを展開し、スマートリフェラルの親となるエントリを表示します。
2. 親エントリをマウスの右ボタンでクリックし、「新規」メニューの「リフェラル」を選択します。あるいは、親エントリをマウスの左ボタンでクリックして選択し、「オブジェクト」メニューから「新規」、「リフェラル」を順に選択することもできます。リフェラルエントリのカスタムエディタダイアログが表示されます。
3. エディタの「一般」タブで、リフェラルの名前を入力し、ドロップダウンリストからネーミング属性を選択します。名前は、選択したネーミング属性の値となります。必要に応じて、このリフェラルを説明する文字列も入力できます。
4. エディタの「URL」タブで、「構成」ボタンをクリックしてスマートリフェラルの URL を定義します。表示されるダイアログに LDAP URL の要素を入力します。

   URL の要素には、リフェラルエントリを保持するディレクトリサーバーのホスト名と LDAP ポート番号、およびサーバー上のターゲットエントリの DN が含まれています。デフォルトでは、ターゲット DN はスマートリフェラルエントリと同じ DN となります。しかし、ターゲット DN には、任意のサフィックス、サブツリー、または最下位エントリを指定できます。
5. LDAP URL の構成ダイアログで「了解」をクリックします。新しいリフェラルのテキストボックスに URL が表示されます。
6. 新しいリフェラルの隣にある「追加」をクリックして、リフェラルをリストに追加します。
7. このエントリのリフェラルとして返される、複数の URL を定義できます。「リフェラルリスト」を作成および管理するには、「構成」、「追加」、「削除」、「変更」ボタンを使います。
8. 「リフェラル認証」ボタンをクリックして、ダイアログを表示します。このダイアログでは、リモートサーバーを指すリフェラルをたどるときに、Directory Server コンソールがバインドに使用する証明情報を設定できます。サーバーへのアクセス時に使われるバインド DN とパスワードを定義できます。同じサーバーを指すすべてのリフェラルは、同じ証明情報を使います。
9. サーバー、および対応する証明情報のリストを管理するには、「追加」、「編集」、「削除」ボタンを使います。設定が完了したら、「了解」をクリックします。

10. リフェラルのカスタムエディタで「了解」をクリックし、スマートリフェラルエントリを保存します。

    コンソールのディレクトリツリーには、スマートリフェラルエントリのターゲットサブツリーまたはエントリが表示されます。スマートリフェラルエントリに黄色の警告アイコンが表示されるときは、URL または証明情報が無効です。エントリをダブルクリックし、「リフェラルエラー」が表示されていたら「継続」をクリックして、「URL」または「リフェラル認証」を変更してエラーを修正します。

コマンド行からのスマートリフェラルの作成

スマートリフェラルを作成するには、`referral` オブジェクトクラスと `extensibleObject` オブジェクトクラスを持つエントリを作成します。`referral` オブジェクトクラスは、LDAP URL を含むことになる `ref` 属性を許可します。`extensibleObject` オブジェクトクラスは、ターゲットエントリと一致させるために、任意のスキーマ属性をネーミング属性として使用することを許可します。

たとえば、`uid=bjensen` エントリの代わりにスマートリフェラルを返すには、次のエントリを定義します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: referral
uid: bjensen
ref: ldap://east.example.com/cn=Babs%20Jensen,ou=Sales,
 o=east,dc=example,dc=com
```

注 サーバーでは、LDAP URL で空白のあとに続く情報はすべて無視されます。このため、リフェラルとして使用する予定のある LDAP URL では、空白の代わりに `%20` を使用する必要があります。

スマートリフェラルを定義すると、別のサーバー上の `cn=Babs Jensen` エントリで、`uid=bjensen` エントリの修正が実際に行われます。`ldapmodify` コマンドは、たとえば次のように、自動的にリフェラルをたどります。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: replace
replace: telephoneNumber
telephoneNumber: (408) 555-1234
```

スマートリフェラルエントリを変更するには、たとえば次のように、`ldapmodify` の `-M` オプションを使う必要があります。

```
ldapmodify -M -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: replace
replace: ref
ref: ldap://east.example.com/cn=Babs%20Jensen,ou=Marketing,
 o=east,dc=example,dc=com
```

属性値の暗号化

属性の暗号化は Sun ONE Directory Server 5.2 の新機能で、ディレクトリに格納されている機密データを保護します。この機能を使用すると、エントリの特定の属性を暗号化された形式で格納するように指定できます。これにより、データベースファイル、バックアップファイル、およびエクスポートされた LDIF ファイルに格納されているデータが読み取られることを防ぎます。

この機能を使用すると、属性値は Directory Server に格納される前に暗号化され、返される前に復号化されます。ACI などのメカニズムを使用して、アクセスが禁止されているデータに LDAP クライアントがアクセスすることを防ぎ、SSL を使用して通信を暗号化する必要があります。データセキュリティ一般と、属性暗号化のアーキテクチャの概要については、『Sun ONE Directory Server Deployment Guide』の第 7 章「Designing a Secure Directory」を参照してください。

属性の暗号化は、サーバー上で SSL が設定され有効になっている場合だけアクティブになります。ただし、デフォルトでは、どの属性も暗号化されません。属性の暗号化はサフィックスレベルで設定されます。つまり、サフィックス内でその属性が現れるすべてのエントリについて、属性が暗号化されます。ディレクトリ全体で属性を暗号化するには、すべてのサフィックスでその属性の暗号化を有効にする必要があります。

警告

属性を暗号化すると、サフィックスに関連付けられたすべてのデータとインデックスファイルが影響を受けます。既存のサフィックスについて暗号化設定を変更するときは、まずサフィックスの内容をエクスポートし、設定を変更してからその内容をふたたびインポートする必要があります。この手順は、コンソールを使って簡単に実行できます。

さらに、暗号化を有効にするときは、暗号化されていない値を含んでいる可能性のあるデータベースキャッシュファイルを手動で削除する必要があります。

新しいサフィックスにデータを読み込むときや作成するときは、暗号化されているすべての属性をあらかじめ有効にする必要があります。

一部のエントリがネーミング属性として使っている属性を暗号化する場合、DN に表示される値は暗号化されず、エントリに格納される値が暗号化されます。

DIGEST-MD5 SASL 認証の場合と同様に、暗号化のために `userPassword` 属性を選択しても、パスワードがクリアテキストとして格納されている場合を除き、実質的なセキュリティ上の利点はありません。パスワードポリシーにパスワードの暗号化メカニズムが定義されている場合、それをさらに暗号化してもセキュリティの強化にはならず、バインド操作のたびにパフォーマンスが低下するという結果になるだけです。

コンソールからの属性の暗号化設定

1. Directory Server コンソールで「設定」タブを選択し、「データ」ノードを展開します。次に、属性値を暗号化する対象のサフィックスを選択します。右側のパネルで「属性の暗号化」タブを選択します。

   このタブには、このサフィックスで現在暗号化されているすべての属性の名前と暗号化スキームを示すテーブルがあります。

2. 属性の暗号化を有効にするには、次の手順を実行します。

   a. 「属性の追加」ボタンをクリックして、属性のリストを表示します。

   b. 暗号化する属性をリストから選択し、「了解」をクリックします。この属性が、テーブルの「属性名」列に追加されます。

   c. この属性の暗号化スキーマを、属性名の隣にあるドロップダウンリストから選択します。

3. 属性が暗号化されないようにするには、テーブルから属性名を選択し、「属性の削除」ボタンをクリックします。

4. 「保存」をクリックします。暗号化設定を変更する前にサフィックスの内容を LDIF ファイルにエクスポートするように促すメッセージが表示されます。

5. 「サフィックスをエクスポート」をクリックして、エクスポートダイアログを開きます。エクスポートを行わない場合は、「継続」をクリックして、属性の暗号化設定を変更します。新しい設定が保存されます。

   サフィックスをまだエクスポートしていないときは、内容を保存するために、この時点でエクスポートします。暗号化されている属性がサフィックスに含まれてる場合、次の手順でこの LDIF ファイルを使ってサフィックスを再初期化するのであれば、暗号化された状態のままこれを LDIF にエクスポートすることもできます。

   LDIF ファイルを使用してサフィックスを初期化するように促すメッセージが表示されます。

6. 「ただちにサフィックスを初期化」をクリックして初期化ダイアログを開き、ディレクトリに読み込む LDIF ファイルの名前を入力します。

   サフィックスを再初期化すると、暗号化された値を復元できなくなるため、前の手順で、暗号化された属性をそのままサフィックスからエクスポートした場合は、そのファイル使ってこの時点で初期化を行う必要があります。このファイルが読み込まれ、インデックスが作成されるときに、指定した属性の値はすべて暗号化されます。

   この時点でサフィックスの初期化を行わない場合は、「閉じる」をクリックします。データを後からインポートする手順については、134 ページの「データのインポート」を参照してください。

7. 1 つまたは複数の属性を暗号化するように設定を変更し、インポート処理の前にその属性が値を持っていた場合、暗号化されていない一部の値は、データベースキャッシュに判読可能な状態で残ることがあります。データベースキャッシュをクリアするには、次の手順を実行します。

   a. 20 ページの「Directory Server の起動と停止」で説明する方法で、サーバーを停止します。

   b. root または管理者権限を持つユーザーとして、ファイルシステムの次の場所にあるデータベースキャッシュファイルを削除します。

      ServerRoot/slapd-*serverID*/db/__db.*

   c. Directory Server を再起動します。サーバーは、新しいデータベースキャッシュファイルを自動的に作成します。

コマンド行からの属性の暗号化設定

1. 属性の暗号化を設定するサフィックスに何らかのエントリが含まれるときは、最初にそのサフィックスの内容を LDIF ファイルにエクスポートします。詳細は、141 ページの「データのエクスポート」を参照してください。

   暗号化されている属性がサフィックスに含まれる場合、手順 5 でこの LDIF ファイルを使ってサフィックスを再初期化するのであれば、暗号化された状態のままこれを LDIF にエクスポートします。

2. 属性の暗号化を有効にするときは、`ldapmodify` コマンドを使って次の設定エントリを追加します。

```
ldapmodify -a -h host -p port -D cn=Directory Manager -p password
dn: cn=attributeName, cn=encrypted attributes, cn=databaseName,
 cn=ldbm database, cn=plugins, cn=config
objectclass: top
objectclass: dsAttributeEncryption
cn: attributeName
dsEncryptionAlgorithm: cipherName
```

ここで、*attributeName* は暗号化する属性のタイプ名で、*databaseName* はサフィックスに対応するデータベースの識別名です。*cipherName* は次のいずれかです。

- `ckm_des_cbc` : DES ブロック暗号化方式
- `ckm_des3_cbc` : トリプル DES ブロック暗号化方式
- `ckm_rc2_cbc` : RC2 ブロック暗号化方式
- `ckm_rc4` : RC4 ストリーム暗号化方式

3. 属性が暗号化されないようにするには、`ldapmodify` コマンドを使って次の設定エントリを変更します。

```
ldapmodify -h host -p port -D cn=Directory Manager -p password
dn: cn=attributeName, cn=encrypted attributes, cn=databaseName,
 cn=ldbm database, cn=plugins, cn=config
changetype: modify
replace: dsEncryptionAlgorithm
dsEncryptionAlgorithm: clearText
```

ここで、*attributeName* は暗号化する属性のタイプ名で、*databaseName* はサフィックスに対応するデータベースの識別名です。

注 属性暗号化の設定エントリを削除しないでください。これは、サフィックスを次に初期化したときに自動的に削除されます。

4. 1 つまたは複数の属性を暗号化するように設定を変更し、インポート処理の前にその属性が値を持っていた場合、暗号化されていない一部の値は、データベースキャッシュに判読可能な状態で残ることがあります。データベースキャッシュをクリアするには、次の手順を実行します。

   a. 20 ページの「Directory Server の起動と停止」で説明する方法で、サーバーを停止します。

   b. root または管理者権限を持つユーザーとして、ファイルシステムの次の場所にあるデータベースキャッシュファイルを削除します。

   ServerRoot`/slapd-`*serverID*`/db/__db.*`

c. Directory Server を再起動します。サーバーは、新しいデータベースキャッシュファイルを自動的に作成します。再びキャッシュがいっぱいになるまで、このサフィックスでの操作のパフォーマンスは、若干の影響を受ける可能性があります。

5. 134 ページの「データのインポート」で説明する方法で、LDIF ファイルを使ってサフィックスを初期化します。手順 1 でサフィックスをエクスポートした場合は、サフィックスに最新の内容が含まれるように、そのファイルを使用します。サフィックスを再初期化すると、暗号化された値を復元できなくなるため、手順 1 で、暗号化された属性をそのままサフィックスからエクスポートした場合は、そのファイル使ってこの時点で初期化を行う必要があります。

   このファイルが読み込まれ、対応するインデックスが作成されるときに、指定した属性の値はすべて暗号化されます。

参照整合性の管理

参照整合性は、関連するエントリ間の関係を保持するプラグインメカニズムです。グループのメンバーシップなど、一部のタイプの属性には別のエントリの DN が含まれています。参照整合性を利用することで、エントリを削除したときに、そのエントリの DN を含むすべての属性も削除できます。

たとえば、参照整合性が有効になっているときに、あるユーザーのエントリがディレクトリから削除されると、そのユーザーは、所属しているあらゆるグループからも削除されます。参照整合性が無効な状態では、管理者はグループからユーザーを手動で削除する必要があります。Directory Server と、ユーザーとグループの管理をディレクトリに頼っているその他の Sun ONE 製品を統合する場合には、この機能がとても重要です。

参照整合性のしくみ

参照整合性検査プラグインが有効になっているときに削除操作や名前変更の操作を実行すると、指定された属性に対する整合性更新がただちに実行されます。ただし、デフォルトでは、参照整合性検査プラグインは無効になっています。

ディレクトリ内にあるユーザーエントリまたはグループエントリの削除や名前変更のたびに、その操作が次の参照整合性ログファイルに記録されます。

ServerRoot/slapd-*serverID*/logs/referint

更新間隔と呼ばれる指定した時間が経過すると、参照整合性が有効になっているすべての属性が検索され、検索結果のエントリと、ログファイル内に記録された削除または変更されたエントリの DN が照合されます。特定のエントリが削除されたことがログファイルに記録されている場合は、対応する属性が削除されます。特定のエントリが変更されたことがログファイルに記録されている場合は、対応する属性値が記録に従って変更されます。

参照整合性プラグインのデフォルトの設定が有効な場合は、削除操作や名前変更の操作を実行すると、`member`、`uniquemember`、`owner`、`seeAlso`、および `nsroledn` の各属性に対する整合性更新がただちに実行されます。ただし、参照整合性検査プラグインの動作は、次のような用途に合わせてユーザーが自由に設定できます。

- 参照整合性の更新を別のファイルに記録する
- 更新間隔を変更する。参照整合性の更新がシステムに与える影響を軽減するために、更新間隔を長くする
- 参照整合性を適用する属性を選択する。DN 値を含む属性を使用または定義するために、参照整合性プラグインを使ってそれを監視する

参照整合性の設定

Directory Server コンソールから参照整合性を有効化または無効化したり、プラグインを設定するときは、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで「プラグイン」ノードを展開し、「referential integrity postoperation」プラグインを選択します。

   プラグインの設定が右側のパネルに表示されます。

2. プラグインを有効にする場合は、「プラグインを有効に」チェックボックスを選択します。プラグインを無効にする場合は、このチェックボックスの選択を解除します。
3. 更新間隔を秒単位で変更するときは、「引数 1」に値を設定します。一般的な値は次のとおりです。
   - 0 : 処理の終了後、毎回直ちに更新する。変更処理のたびに、その直後に参照整合性検査を行うことは、サーバーのパフォーマンスに重大な影響を及ぼすため、注意する必要がある
   - 90 : 90 秒ごとに更新する
   - 3600 : 1 時間ごとに更新する
   - 10,800 : 3 時間ごとに更新する
   - 28,800 : 8 時間ごとに更新する
   - 86,400 : 1 日に 1 回更新する

- 604,800 : 1 週間に 1 回更新する

4. 使用する参照整合性ログファイルのパスを「引数 2」に設定します。

   「引数 3」は使用されませんが、表示されている必要があります。

5. 参照整合性が監視される属性は、「引数 4」の最初にリスト表示されます。このリストの管理、および独自の属性の追加には、「追加」ボタンと「削除」ボタンを使います。

注 最適なパフォーマンスを得るには、参照整合性プラグインによって更新される属性にもインデックスを設定する必要があります。詳細は、第 10 章「インデックスの管理」を参照してください。

6. 「保存」をクリックして、変更内容を保存します。
7. 変更を適用するには、Directory Server を再起動する必要があります。

レプリケーションにおける参照整合性の使用

レプリケーション環境では、次のようないくつかの参照整合性検査プラグインの使用に関する制限があります。

- マスターレプリカを含むすべてのサーバーで有効化する必要がある
- すべてのマスターで同じ設定で有効化する必要がある
- ハブまたはコンシューマレプリカだけを含むサーバーで有効化しても意味がない

レプリケーショントポロジで参照整合性プラグインを設定するには、次の手順を実行します。

1. すべてのレプリカが設定され、すべてのレプリケーションアグリーメントが定義されていることを確認します。
2. 参照整合性を維持する属性のセットを定義します。また、マスターサーバーに適用する更新間隔を決定します。
3. 同じ属性セットと同じ更新間隔を使用して、すべてのマスターサーバーで参照整合性プラグインを有効化します。この手順については、82 ページの「参照整合性の設定」を参照してください。
4. すべてのコンシューマサーバー上で参照整合性検査プラグインが無効になっていることを確認します。

第 3 章

ディレクトリツリーの作成

ディレクトリツリーはサーバー内のすべてのエントリから構成され、エントリはそれぞれ DN (識別名) によって示されます。DN は階層構造を持つため、ツリー内のデータ構成を決定する分岐のエントリおよび最下位のエントリが作成されます。管理上、ディレクトリツリーは、サフィックス、サブサフィックス、および連鎖サフィックスによって定義されます。Directory Server コンソールでは、これらのすべての要素を作成および管理できます。また、コマンド行ツールを使用して管理することもできます。

ディレクトリデータ構造の概念については、『Sun ONE Directory Server Deployment Guide』の第 4 章にある「ディレクトリツリーの設計」を参照してください。

この章は、次の節で構成されます。

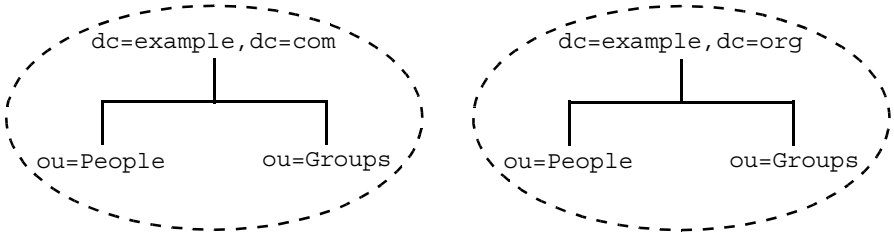
- はじめに
- サフィックスの作成
- サフィックスの管理
- 連鎖サフィックスの作成
- 連鎖サフィックスの管理
- カスケード型連鎖の設定

はじめに

サフィックスとは分岐またはサブツリーであり、サフィックスの内容全体が管理タスクの単位として扱われます。たとえば、サフィックス全体に対してインデックスが定義され、サフィックス全体を 1 回の操作で初期化でき、また、サフィックスはレプリケーションの単位となります。同じ方法でアクセスおよび管理したいデータは、同じサフィックスに格納する必要があります。サフィックスはディレクトリツリーのルートに配置することもでき、その場合はルートサフィックスと呼ばれることがあります。

次の図に、2 種類のルートサフィックスを配置したディレクトリを示します。それぞれ、個別の企業エンティティに対応しています。

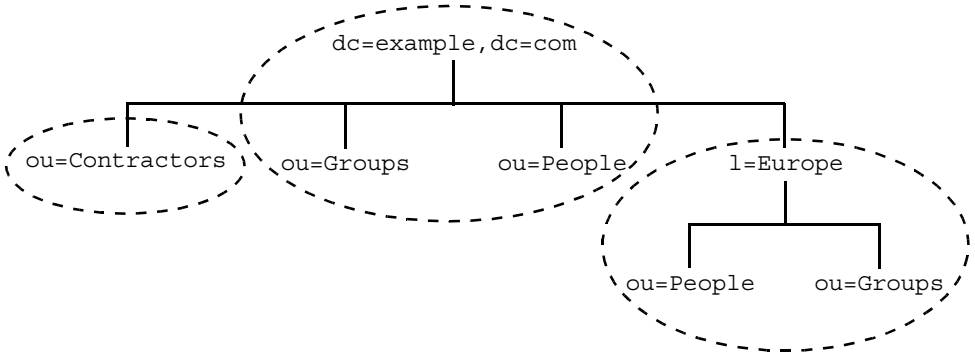
図 3-1 単一 Directory Server 内の 2 種類のルートサフィックス



サフィックスは、他のサフィックスの分岐となることもできます。その場合はサブサフィックスと呼ばれます。親サフィックスを管理する場合、サブサフィックスの内容は操作対象に含まれません。つまり、サブサフィックスは、親から独立して管理されます。ただし、LDAP 操作の結果には、サフィックスに関する情報は含まれません。また、ディレクトリクライアントは、エントリがルートサフィックスの一部であるか、サブサフィックスの一部であるかを認識できません。

次の図に、大規模な企業エンティティに対して、単一のルートサフィックスと複数のサブサフィックスを配置したディレクトリを示します。

図 3-2　複数のサブサフィックスを持つ単一のルートサフィックス



サフィックスは、サーバー内の個々のデータベースに対応します。ただし、データベースおよびそのファイルは、サーバーが内部的に管理するようになったため、Sun ONE Directory Server 5.2 ではデータベース用語は使用されていません。

連鎖サフィックスは、仮想ディレクトリツリーを作成して、別のサーバー上にあるサフィックスを参照します。連鎖サフィックスを使用すると、Directory Server は、リモートサフィックスに対する操作でも、ローカルで実行したかのように結果を返します。サフィックスが連鎖されていて、データがリモートサーバーから取得されることは、クライアントは認識できないため、データの位置が透過的になります。サーバー上のルートサフィックスが、別のサーバーに連鎖されているサブサフィックスを持つ場合もあります。その場合、クライアントからは、単独のツリー構造が作られているように見えます。

カスケード型連鎖の場合、連鎖サフィックスはリモートサーバー上などの別の連鎖サフィックスを参照することがあります。各サーバーは操作を転送し、最終的にクライアントの要求を処理するサーバーへ結果を返します。

連鎖に関する全体的な説明については、『Sun ONE Directory Server Deployment Guide』の第 7 章にある「Designing the Directory Topology」を参照してください。

サフィックスの作成

Directory Server コンソールやコマンド行を使用すると、ルートサフィックスとサブサフィックスのどちらも作成できます。

コンソールを使用した新しいルートサフィックスの作成

1. Directory Server コンソールの最上位レベルにある「設定」タブで、「データ」ノードをマウスの右ボタンでクリックし、ポップアップメニューから「新規サフィックス」を選びます。

   または、「データ」ノードを選択し、「オブジェクト」メニューから「新規サフィックス」を選択します。

   「新規サフィックス」ダイアログボックスが表示されます。

2. 「サフィックス DN」フィールドに、一意のサフィックス名を入力します。名前は、識別名の形式を使用し、コンマで区切られた属性と値の 1 つ以上のペアで構成されている必要があります。

   規則に従い、ルートサフィックスにはドメインコンポーネント (dc) のネーミング属性を使用します。たとえば、`dc=example,dc=org` という新しいサフィックス DN を入力します。

注 サフィックス名には DN 形式の属性と値のペアが含まれますが、サフィックス名は単一の文字列として扱われます。そのため、空白はすべて意味を持ち、サフィックス名の一部となります。

---

3. デフォルトでは、このサフィックスに対するデータベースファイルの位置は、サーバーによって自動的に選択されます。また、サフィックスはデフォルトではシステムインデックスだけを保持し、どの属性も暗号化されず、レプリケーションは設定されません。

   デフォルト値を変更するには、「オプション」ボタンをクリックして、新しいサフィックスのオプションを表示します。

a. データベース名は、データベースファイルを含むディレクトリ名と一致します。デフォルトのデータベース名は、サフィックス DN 内の最初のネーミング属性の値で、一意となるように数字が追加されていることがあります。別の名前を使用するには、「カスタムを使用」ラジオボタンを選択し、新しい一意のデータベース名を入力します。

   データベース名には、ASCII (7 ビット) 英数文字、ハイフン (-)、およびアンダスコア (_) だけを使用できます。たとえば、新しいデータベースに `example_2` という名前を付けることができます。

b. データベースファイルを格納するディレクトリの位置を選択することもできます。デフォルトでは、次のパスを持つサブディレクトリです。

   ServerRoot`/slapd-`*serverID*`/db`

   新しいパスを入力します。または、「参照」をクリックして、データベースディレクトリの新しい位置を選択します。新しいパスには、ディレクトリサーバーホスト上でアクセスできる必要があります。

c. 新しいサフィックスを素速く設定するために、既存のサフィックスのクローンを作成することもできます。「サフィックス設定をクローン」を選択し、ドロップダウンメニューからクローンを作成するサフィックスを選択します。クローンの作成について、次の設定のどれかを選択します。

   - インデックス設定をクローン : 新しいサフィックスは、クローン作成元のサフィックスと同じ属性で同じインデックスを保持します。
   - 属性の暗号化設定をクローン : 新しいサフィックスで、クローン作成元のサフィックスと同じ属性のリストについての暗号化と同じ暗号化スキーマを有効にします。
   - レプリケーション設定をクローン : 新しいサフィックスのレプリカ タイプは、クローン作成元のサフィックスと同じになります。サプライヤの場合は、すべてのレプリケーションアグリーメントが複製され、レプリケーションが有効になります。

d. 新しいサフィックスのオプションをすべて設定したら、「了解」をクリックします。新しいサフィックスのダイアログに、選択したオプションがすべて表示されます。

4. 新しいサフィックスのダイアログで「了解」をクリックして、新しいルートサフィックスを作成します。

   ルートサフィックスが「データ」の分岐の下に自動的に表示されます。新しいサフィックスをさらに詳しく設定するには、97 ページの「サフィックスの管理」を参照してください。

   新しいルートサフィックスには、サフィックス DN に関するエントリも、どのエントリも含まれません。そのため、ディレクトリ内ではアクセスできません。また、初期化されて、適切なアクセス権を与えられるまでは、コンソールの「ディレクトリ」タブに表示されません。

   このサフィックスを LDIF ファイルから初期化する場合は、以下の手順を省略できます。ただし、LDIF ファイル内のルートエントリに、導入に必要な ACI (アクセス制御命令) が含まれていることを確認してください。

5. コンソールで、最上位の「ディレクトリ」タブを選択します。新しいサフィックスは、まだディレクトリツリーに表示されていません。

6. Directory Manager としてログインしていない場合は、「コンソール」>「新規ユーザーとしてログイン」の順に選択して、Directory Manager としてログインします。Directory Manager の DN とパスワードを入力して、ログインします。デフォルトでは、Directory Manager の DN は `cn=Directory Manager` です。

7. ディレクトリツリーのルートノードを右クリックします。ルートノードには、サーバーのホスト名とポートが含まれます。ポップアップメニューから「新規ルートオブジェクト」を選択し、新しいルートサフィックスの DN を選択します。

   または、ディレクトリツリーのルートノードを選択し、「オブジェクト」メニューから「新規ルートオブジェクト」を選択します。

8. 表示された「新規オブジェクト」ダイアログボックスで、ルートオブジェクトにするオブジェクトクラスを 1 つ選択します。このオブジェクトクラスによって、ルートエントリに追加されるその他の属性が決まります。

   規則では、`dc` のネーミング属性を含むサフィックス DN のルートオブジェクトは、`domain` オブジェクトクラスに属します。通常、ルートオブジェクトは単純なオブジェクトで、データはほとんど含まれません。

9. オブジェクトクラスを選択したら、「新規オブジェクト」ダイアログボックスで「了解」をクリックします。

   コンソールには、新しいルートオブジェクトに対して汎用エディタが表示されます。デフォルトの ACI セットが、新しいオブジェクトに自動的に追加されます。詳細は、188 ページの「デフォルト ACI」を参照してください。ACI セットの変更も含めて、トポロジに必要な属性値を追加および編集します。

   新しいサフィックスにユーザーエントリが含まれるときは、「Allow self entry modification except for nsroledn and aci attributes.」というデフォルトの ACI を変更する必要があります。セキュリティを強化するために、これを次の ACI に置き換えます。

```
aci: (targetattr != "nsroledn || aci || nsLookThroughLimit ||
 nsSizeLimit || nsTimeLimit || nsIdleTimeout ||
 passwordPolicySubentry || passwordExpirationTime ||
 passwordExpWarned || passwordRetryCount || retryCountResetTime
 || accountUnlockTime || passwordHistory ||
 passwordAllowChangeTime")(version 3.0; acl "Allow self entry
 modification except for nsroledn, aci, resource limit
 attributes, passwordPolicySubentry and password policy state
 attributes"; allow (write)userdn ="ldap:///self";)
```

10. エントリを編集したら、汎用エディタで「了解」をクリックして、新しいサフィックスのルートオブジェクトを作成します。

    新しいサフィックスがディレクトリツリーに表示され、ACI によって与えられた権限に従って、コンソールで管理できるようになります。

コンソールを使用した新しいサブサフィックスの作成

既存のルートサフィックスまたはサブサフィックスの下に新しいサブサフィックスを作成する方法は次のとおりです。

1. Directory Server コンソールの最上位レベルにある「設定」タブで、「データ」ノードと任意のサフィックスノードを展開して、親サフィックスを表示します。
2. 親サフィックスのノードを右クリックし、ポップアップメニューから「新規サブサフィックス」を選択します。

   または、親サフィックスのノードを選択し、「オブジェクト」メニューから「新規サブサフィックス」を選択します。

   「新規サブサフィックス」ダイアログボックスが表示されます。
3. 「サブサフィックス RDN」フィールドに、一意の名前を入力します。名前は、相対識別名 (RDN) の形式で、コンマで区切られた属性と値の 1 つ以上のペアで構成されている必要があります。たとえば、`ou=Contractors` のようにします。

   テキストボックスの下の行に、このサブサフィックスの完全な DN が表示されます。これは、RDN に追加された親サフィックス DN で構成されています。

注 サブサフィックス名には RDN 形式の属性と値のペアが含まれますが、サブサフィックス名は単一の文字列として扱われます。そのため、空白はすべて意味を持ち、サフィックス DN の一部となります。

4. デフォルトでは、このサフィックスに対するデータベースファイルの位置は、サーバーによって自動的に選択されます。また、サフィックスはデフォルトではシステムインデックスだけを保持し、どの属性も暗号化されず、レプリケーションは設定されません。

   デフォルト値を変更するには、「オプション」ボタンをクリックして、新しいサフィックスのオプションを表示します。

   a. データベース名は、データベースファイルを含むディレクトリ名と一致します。デフォルトのデータベース名は、RDN 内の最初のネーミング属性の値で、一意となるように数字が追加されていることがあります。別の名前を使用するには、「カスタムを使用」ラジオボタンを選択し、新しい一意のデータベース名を入力します。

      データベース名には、ASCII (7 ビット) 英数文字、ハイフン (-)、およびアンダスコア (_) だけを使用できます。たとえば、新しいデータベースに `temps-US` という名前を付けることができます。

   b. データベースファイルを格納するディレクトリの位置を選択することもできます。デフォルトでは、次のパスを持つサブディレクトリです。

      ServerRoot`/slapd-`*serverID*`/db`

      新しいパスを入力します。または、「参照」をクリックして、データベースディレクトリの新しい位置を選択します。新しいパスには、ディレクトリサーバーアプリケーションによってアクセスできる必要があります。

   c. 新しいサブサフィックスを素速く設定するために、既存のサフィックスのクローンを作成することもできます。既存のサフィックスは、親サフィックスでも別のサフィックスでも構いません。「サフィックス設定をクローン」を選択し、ドロップダウンメニューからクローンを作成するサフィックスを選択します。クローンの作成について、次の設定のどれかを選択します。

      - インデックス設定をクローン : 新しいサフィックスは、クローン作成元のサフィックスと同じ属性で同じインデックスを保持します。
      - 属性の暗号化設定をクローン : 新しいサフィックスで、クローン作成元のサフィックスと同じ属性のリストについての暗号化と同じ暗号化スキーマを有効にします。
      - レプリケーション設定をクローン : 新しいサフィックスのレプリカ タイプは、クローン作成元のサフィックスと同じになります。サプライヤの場合は、すべてのレプリケーションアグリーメントが複製され、レプリケーションが有効になります。

   d. 新しいサフィックスのオプションをすべて設定したら、「了解」をクリックします。新しいサブサフィックスのダイアログに、選択したオプションがすべて表示されます。

5. 新しいサブサフィックスのダイアログで「了解」をクリックして、サブサフィックスを作成します。

   作成したサブサフィックスは、「設定」タブで親サフィックスの下に自動的に表示されます。新しいサフィックスをさらに詳しく設定するには、97 ページの「サフィックスの管理」を参照してください。

   新しいサブサフィックスには、RDN に対するエントリも、どのエントリも含まれません。そのため、ディレクトリ内ではアクセスできません。また、初期化されて、適切なアクセス権を与えられるまでは、コンソールの「ディレクトリ」タブに表示されません。

   このサフィックスを LDIF ファイルから初期化する場合は、以下の手順を省略できます。ただし、LDIF ファイル内の親サフィックスと新しいエントリに、導入に必要な ACI (アクセス制御命令) が含まれていることを確認してください。

6. コンソールで、最上位の「ディレクトリ」タブでディレクトリツリーを展開し、サブサフィックスの親を表示します。新しいサブサフィックスは、まだ表示されていません。

7. Directory Manager としてログインしていない場合は、「コンソール」>「新規ユーザーとしてログイン」の順に選択して、Directory Manager としてログインします。Directory Manager の DN とパスワードを入力して、ログインします。デフォルトでは、Directory Manager の DN は `cn=Directory Manager` です。

8. サブサフィックスの親を右クリックし、ポップアップメニューから「新規」を選択します。新しいオブジェクトのリストで、サブサフィックスの RDN に対応するオブジェクトのタイプを選択します。たとえば、`ou=Contractors` サブサフィックスを作成した場合は、「OrganizationalUnit」を選択します。サブサフィックスのオブジェクトクラスがリストにない場合は、「その他」を選択し、表示される「新規オブジェクト」ダイアログボックスでリストからオブジェクトクラスを選択します。

   または、サブサフィックスの親を選択し、「オブジェクト」メニューから「新規」を選択します。

9. コンソールには、新しいオブジェクトに対してカスタムエディタまたは汎用エディタが表示されます。ACI セットの変更も含めて、トポロジに必要な属性値を追加および編集します。

10. エントリを編集したら、エディタで「了解」をクリックして、新しいサブサフィックスのエントリを作成します。

    新しいサブサフィックスがディレクトリツリーに表示され、ACI によって与えられた権限に従って、コンソールで管理できるようになります。

コマンド行からのサフィックスの作成

ldapmodify コマンド行ユーティリティを使用しても、ディレクトリ内にサフィックスを作成できます。ルートサフィックスとサブサフィックスは、サーバーによって内部的に同じ方法で管理されるため、それらをコマンド行から作成する手順はほとんど同じです。

1. 次のコマンドを使用して、ルートサフィックスのサフィックス設定エントリを cn=mapping tree,cn=config の下に作成します。

   ```
   ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
   dn: cn="suffixDN",cn=mapping tree,cn=config
   objectclass: top
   objectclass: extensibleObject
   objectclass: nsMappingTree
   cn: suffixDN
   nsslapd-state: backend
   nsslapd-backend: databaseName
   ^D
   ```

   サブサフィックスについては、上記のコマンドに次の属性を追加します。
   `nsslapd-parent-suffix: "parentSuffixDN"`

   suffixDN は、新しいサフィックスの完全 DN です。ルートサフィックスでは、ドメインコンポーネント (dc) のネーミング属性が使用されます。たとえば、dc=example,dc=org などです。サブサフィックスでは、*suffixDN* にサブサフィックスの RDN と親サフィックスの DN が含まれます。たとえば、ou=Contractors,dc=example,dc=com のようになります。

注 サフィックス名は DN 形式となりますが、単一の文字列として扱われます。そのため、空白はすべて意味を持ち、サフィックス名の一部となります。このサフィックスにアクセスするには、空白文字の使い方を *suffixDN* の文字列で使用される場合と同じにする必要があります。

   databaseName は、このサフィックスに関連付けられ、内部的に管理されるデータベースの名前です。この名前は、すべてのサフィックスの *databaseNames* で一意となる必要があります。規則では、この名前が *suffixDN* の 1 番目のネーミングコンポーネントの値となります。*databaseName* は、サフィックスに対するデータベースファイルを格納するディレクトリの名前でもあるため、使用できる文字は、ASCII (7 ビット) 英数字、ハイフン (-)、およびアンダースコア (_) だけです。

   サブサフィックスでは、*parentSuffixDN* は親サフィックスの正確な DN です。

2. 次のコマンドを使用して、データベース設定エントリを作成します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=databaseName,cn=ldbm database,cn=plugins,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsBackendInstance
cn: databaseName
nsslapd-suffix: suffixDN
^D
```

databaseName と *suffixDN* は、前の手順の値と同じにする必要があります。

このエントリがディレクトリに追加されると、サーバーのデータベースモジュールは、次のディレクトリにデータベースファイルを自動的に作成します。

ServerRoot/slapd-*serverID*/db/*databaseName*

サーバーがデータベースファイルを別の位置に作成するようにするには、次の属性を使用してデータベース設定エントリを作成します。

nsslapd-directory: *path*/*databaseName*

サーバーはデータベースファイルを保存するために、指定の場所に *databaseName* という名前のディレクトリを自動的に作成します。

3. ルートサフィックスまたはサブサフィックスのベースエントリを作成します。

たとえば、次のコマンドを使用すると、dc=example,dc=org ルートサフィックスのベースエントリを作成できます。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: dc=example,dc=org
objectclass: top
objectclass: domain
dc: 例
^D
```

DN の 1 番目のネーミング属性とその値を指定する必要があります。さらに、ベースエントリのオブジェクトクラスのスキーマで必要とされる属性もすべて指定します。規則では、ドメインコンポーネント (dc) を使用するルートサフィックス DN には、domain オブジェクトクラスが含まれます。これは、その他の属性には必要ありません。

また、ルートサフィックスには ACI (アクセス制御命令) 属性を追加して、アクセスポリシーを適用する必要があります。次の aci 属性の値を追加すると、匿名の読み取り、本人によるセキュリティが保護された状態での変更、および完全な管理者アクセスを許可できます。

```
aci: (targetattr != "userPassword") (version 3.0; acl
 "Anonymous access";
 allow (read, search, compare)userdn = "ldap:///anyone";)
aci: (targetattr != "nsroledn || aci || nsLookThroughLimit ||
```

```
 nsSizeLimit || nsTimeLimit || nsIdleTimeout ||
 passwordPolicySubentry || passwordExpirationTime ||
 passwordExpWarned || passwordRetryCount || retryCountResetTime
 || accountUnlockTime || passwordHistory ||
 passwordAllowChangeTime")(version 3.0; acl "Allow self entry
 modification except for nsroledn, aci, resource limit
 attributes, passwordPolicySubentry and password policy state
 attributes"; allow (write)userdn ="ldap:///self";)
aci: (targetattr = "*")(version 3.0; acl
 "Configuration Administrator";
 allow (all) userdn = "ldap:///uid=admin,ou=Administrators,
 ou=TopologyManagement, o=NetscapeRoot";)
aci: (targetattr = "*")(version 3.0; acl
 "Configuration Administrators Group";
 allow (all) (groupdn =
 "ldap:///cn=Configuration Administrators, ou=Groups,
 ou=TopologyManagement, o=NetscapeRoot");)
```

サブサフィックスの例としては、次のコマンドを使用すると、`ou=Contractors,dc=example,dc=com` のベースエントリを作成できます。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: ou=Contractors,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
description: base of separate subsuffix for contractor identities
^D
```

DN のネーミング属性と値を指定する必要があります。さらに、ベースエントリのオブジェクトクラスのスキーマで必要とされる属性もすべて指定します。その他、許可されている属性も追加できます。親の ACI の適用範囲に新しいサブサフィックスが含まれている場合、サブサフィックスには親の ACI で定義されたアクセス制御が適用されます。サブサフィックスに異なるアクセスポリシーを定義するには、ベースエントリの作成時に `aci` 属性を指定します。

サフィックスの管理

サフィックスを作成すると、内容をすべて一括で管理できます。ここでは、サフィックスへのアクセスを管理する方法について説明します。すべての操作を無効にする方法、サフィックスを読み取り専用にする方法、サフィックスレベルのリフェラルを作成する方法などが含まれます。

その他多くのディレクトリ管理業務はサフィックスレベルで設定されますが、このマニュアルでは次に示す個別の章で説明しています。

- 134 ページの「データのインポート」
- 141 ページの「データのエクスポート」
- 351 ページの「インデックスの管理」
- 77 ページの「属性値の暗号化」
- 273 ページの「レプリケーションの管理」

サフィックスの無効化と有効化

保守のためにサフィックスを使用できないようにすることや、セキュリティ上の理由からサフィックスの内容を使用できないようにすることが必要な場合があります。サフィックスを無効にすると、サーバーは、サフィックスへのアクセスを試みたクライアント操作への応答時に、サフィックスの内容を読み書きできなくなります。デフォルトのリフェラルを定義している場合は、無効になっているサフィックスにクライアントがアクセスしようとすると、デフォルトのリフェラルが返されます。

コンソールからのサフィックスの無効化と有効化

1. Directory Server コンソールの最上位の「設定」タブで「データ」ノードを展開し、無効にするサフィックスを選択します。
2. 右側のパネルで、「設定」タブを選択します。デフォルトでは、すべてのサフィックスは、作成した時点で有効になります。

   対象のサフィックスに対してレプリケーションを有効にしている場合は、このタブの内容が自動的に更新される旨のメッセージが表示されます。レプリケートされているサフィックスを無効にすると、そのサフィックスへのレプリケーションもできなくなります。レプリケーションの中断が回復設定より短い限り、サフィックスが有効な状態に戻ったときに、レプリケーションメカニズムによってこのレプリカの更新が再開されます。レプリケーション回復設定には、コンシューマーレプリカのパージ遅延、およびこのサプライヤの更新履歴ログの最大サイズと有効期限があります (280 ページの「コンシューマの詳細設定」を参照)。

3. 「このサフィックスへのアクセスを有効に」チェックボックスの選択を解除して、サフィックスを無効にします。または、このチェックボックスを選択して、サフィックスを有効にします。
4. 「保存」をクリックして、変更内容を適用します。サフィックスは、ただちに無効または有効になります。
5. 必要に応じて、サフィックスが無効である間にサフィックスへのすべての操作に対して返されるグローバルのデフォルトリフェラルを設定できます。これは、最上位の「設定」タブのルートノードの「ネットワーク」タブで設定します。詳細は、73 ページの「コンソールからのデフォルトリフェラルの設定」を参照してください。

コマンド行からのサフィックスの無効化と有効化

1. 次のコマンドを使用して、サフィックスの設定エントリ内の `nsslapd-state` 属性を編集します。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn="suffixDN",cn=mapping tree,cn=config
changetype: modify
replace: nsslapd-state
nsslapd-state: disabled or backend
^D
```

   suffixDN は、定義済みのサフィックス DN の完全な文字列です。空白もすべて含まれます。`nssladp-state` 属性の値を `disabled` に設定するとサフィックスは無効になり、`backend` に設定するとすべてのアクセス権が許可されます。

   コマンドの実行が成功すると、サフィックスはただちに無効になります。

2. 必要に応じて、サフィックスが無効である間にサフィックスへのすべての操作に対して返されるグローバルのデフォルトリフェラルを設定できます。詳細は、74 ページの「コマンド行からのデフォルトリフェラルの設定」を参照してください。

アクセス権とリフェラルの設定

サフィックスを完全に無効にすることなくサフィックスへのアクセスを制限するには、アクセス権を変更して、読み取り専用アクセスを許可することもできます。この場合、書き込み操作に対しては、別のサーバーへのリフェラルを定義する必要があります。また、読み取りアクセスと書き込みアクセスの両方を拒否し、サフィックスへのすべての操作に対するリフェラルを定義することもできます。

さらに、リフェラルを使用して、クライアントアプリケーションが一時的に別のサーバーを使用するように設定することもできます。たとえば、サフィックスにリフェラルを追加しておくことにより、サフィックスの内容のバックアップ中に、クライアントが別のサーバーを使用するように設定できます。

レプリケーションメカニズムは、サフィックスをレプリケーションの対象として設定するために、書き込み権限とリフェラルが必要です。レプリケーションの有効化、レプリカの昇格、またはレプリカの降格によって、リフェラルの設定は変更されます。

警告 サフィックスがレプリケートされている場合にリフェラルを変更すると、そのサフィックスのレプリケートされた動作が影響を受ける可能性があります。

コンソールからのアクセス権とリフェラルの設定

1. Directory Server コンソールの最上位の「設定」タブで「データ」ノードを展開し、リフェラルを設定するサフィックスを選択します。
2. 右側のパネルで、「設定」タブを選択します。連鎖サフィックスが有効である場合は、アクセス権とリフェラルだけを設定できます。対象のサフィックスに対してレプリケーションを有効にしている場合は、このタブの内容が自動的に更新される旨のメッセージが表示されます。
3. 次のラジオボタンのどれかを選択して、該当のサフィックスのエントリに対する書き込み操作への応答を設定します。
   - 書き込みおよび読み取り要求を処理 : このラジオボタンは、デフォルトで選択されています。サフィックスの通常の動作を示します。リフェラルが定義されていることもありますが、リフェラルは返されません。
   - 読み取り要求を処理し、書き込み要求にはリフェラルを返す : サフィックスを読み取り専用にする場合は、このラジオボタンを選択し、書き込み要求に対してリフェラルとして返す 1 つ以上の LDAP URL をリストに入力します。
   - 読み取りおよび書き込み要求の両方にリフェラルを返す : 読み取りアクセスと書き込みアクセスの両方を拒否する場合は、このラジオボタンを選択します。この場合の動作は、サフィックスへのアクセスを無効にした場合と同様ですが、グローバルのデフォルトリフェラルを使用するのではなく、該当のサフィックス専用にリフェラルを定義してもよい点が異なります。
4. 「追加」ボタンまたは「削除」ボタンを使用して、リフェラルのリストを編集します。「追加」ボタンをクリックすると、新しいリフェラルの LDAP URL を作成するためのダイアログが表示されます。リモートサーバー内の任意の分岐 DN にリフェラルを作成できます。LDAP URL の構造については、『Sun ONE Directory Server Getting Started Guide』を参照してください。

   複数のリフェラルを入力できます。ディレクトリは、クライアントアプリケーションからの要求に対応して、このリスト内のすべてのリフェラルを返します。
5. 「保存」をクリックして、変更内容を適用します。新しいアクセス権とリフェラルの設定が、ただちに有効になります。

コマンド行からの権限とリフェラルの設定

次のコマンドで、*suffixDN* は、定義済みのサフィックス DN の完全な文字列です。空白もすべて含まれます。*LDAPURL* は、ターゲットのホスト名、ポート番号、および DN を含む有効な URL です。たとえば、次のように指定します。

```
ldap://phonebook.example.com:389/ou=People,dc=example,dc=com
```

1. 次のコマンドを使用して、サフィックスの設定エントリを編集します。

   ```
   ldapmodify -h host -p port -D "cn=Directory Manager" -w password
   dn: cn="suffixDN",cn=mapping tree,cn=config
   changetype: modify
   replace: nsslapd-state
   nsslapd-state: referral on update or referral
   -
   add: nsslapd-referral
   nsslapd-referral: LDAPURL
   ^D
   ```

   最後の変更文を繰り返すことにより、任意の数の LDAP URL を `nsslapd-referral` 属性に追加できます。

   `nsslapd-state` が `referral on update` のときには、サフィックスは読み取り専用になり、書き込み操作に対してはすべての LDAP URL がリフェラルとして返されます。この値が `referral` のときには、読み取り操作と書き込み操作の両方が拒否され、すべての要求に対してリフェラルが返されます。

2. コマンドの実行が成功すると、サフィックスはただちに読み取り専用になるかアクセスできなくなり、リフェラルを返すことができるようになります。

サフィックスの削除

サフィックスを削除すると、そのサフィックスの分岐全体がディレクトリから削除されます。親サフィックスを削除し、そのサブサフィックスを、ディレクトリで新しいルートサフィックスとして保持することもできます。

警告 サフィックスを削除すると、ディレクトリ内の全エントリが完全に削除されるため、レプリケーション設定も含めてサフィックスの全設定が削除されます。

コンソールからのサフィックスの削除

1. Directory Server コンソールの「設定」タブで、「データ」ノードを展開します。

2. 削除するサフィックスを右クリックし、ポップアップメニューから「削除」を選択します。

   または、サフィックスのノードを選択し、「オブジェクト」メニューから「削除」を選択します。

3. すべてのサフィックスエントリがディレクトリから削除されることを示す確認のダイアログが表示されます。

   親サフィックスに加えて、すべてのサブサフィックスを再帰的に削除することもできます。分岐全体を削除する場合は、「このサフィックスとそのサブサフィックスすべてを削除」を選択します。また、特定のサフィックスだけを削除し、そのサブサフィックスをディレクトリに残しておく場合は、「このサフィックスだけを削除する」を選択します。

4. 「了解」をクリックして、サフィックスを削除します。

   コンソールによって処理されている内容を示すダイアログボックスが表示されます。

コマンド行からのサフィックスの削除

コマンド行からサフィックスを削除するには、`ldapdelete` コマンドを使用して、ディレクトリからサフィックスの設定エントリを削除します。

サブサフィックスも含めて分岐全体を削除する場合は、削除する親サフィックスのサブサフィックスを見つけ出し、サブサフィックスとさらにそれらのサブサフィックスのすべてについて手順を繰り返す必要があります。

1. 次のコマンドを使用して、サフィックスの設定エントリを削除します。

   ```
   ldapdelete -h host -p port -D "cn=Directory Manager" -w password ¥
              -v 'cn="suffixDN",cn=mapping tree,cn=config'
   ```

   このコマンドによって、*suffixDN* のベースエントリで始まるサフィックスがサーバーから削除されます。これで、サフィックスはディレクトリに表示されなくなり、アクセスできなくなります。

2. `cn=`*databaseName*`,cn=ldbm database,cn=plugins,cn=config` にある対応するデータベース設定エントリと、その下位にあるすべてのエントリを削除します。次のコマンドは、『Sun ONE Directory Server Resource Kit (DSRK)』の `ilash` ツールを使用します。DSRK のダウンロードと使用方法については、16 ページの「Directory Server ツールのダウンロード」を参照してください。

```
% ilash -call "http://host:port/" -user "cn=Directory Manager"
[...]
Enter password for "cn=Directory Manager": password
[...]
[example,com]% dcd cn=config
[config]% ddelete -subtree ¥
"cn=databaseName,cn=ldbm database,cn=plugins,cn=config"

Removed cn=aci, cn=index, cn=databaseName, cn=ldbm database,
cn=plugins, cn=config

Removed cn=entrydn, cn=index, cn=databaseName, cn=ldbm database,
cn=plugins, cn=config

[...]

Removed cn=encrypted attributes, cn=databaseName, cn=ldbm database,
cn=plugins, cn=config

Removed cn=index, cn=databaseName, cn=ldbm database, cn=plugins,
cn=config

Removed cn=monitor, cn=databaseName, cn=ldbm database, cn=plugins,
cn=config

Removed cn=databaseName,cn=ldbm database,cn=plugins,cn=config
```

この出力には、データベースに関連付けられていて削除する必要のあるすべてのインデックス設定エントリが示されています。データベース設定がすべて削除されると、サーバーは、このサフィックスに関連付けられているすべてのデータベースファイルとディレクトリを削除します。

連鎖サフィックスの作成

ルートサフィックスとサブサフィックスはどちらも、別のサーバーに連鎖することができます。どの手順も、コンソールを使用しても、コマンド行からでも実行できます。

ただし、連鎖サフィックスを作成する前に、リモートサーバーのプロキシ ID を作成する必要があります。ローカルサーバーは、連鎖サフィックスを通じて操作を転送するときに、プロキシ ID を使用してリモートサーバーをバインドします。

識別パラメータを使用して多くの連鎖サフィックスを設定する場合は、新しい連鎖サフィックスの連鎖パラメータに対するデフォルト値も設定する必要があります。連鎖サフィックスの作成前後の任意の時点で、LDAP 制御とサーバーコンポーネントに関する連鎖ポリシーを設定することもできます。115 ページの「連鎖ポリシーの設定」を参照してください。

プロキシ ID の作成

プロキシ ID は、ローカルサーバーが連鎖操作をバインドおよび転送するために使用するリモートサーバー上のユーザーです。セキュリティ上の理由から、プロキシには Directory Manager または管理ユーザー (`admin`) を使用しないでください。

その代わりに、指定されたサーバーからだけの連鎖操作に使用する新しい ID を作成します。連鎖されるすべてのサーバーと、107 ページの「コンソールからの連鎖サフィックスの作成」または 118 ページの「コンソールからの連鎖ポリシーの変更」で定義されるすべてのフェイルオーバーサーバーで、この ID を作成します。

コンソールからのプロキシ ID の作成

この手順は、連鎖サフィックスのターゲットであるリモートサーバーに接続された Directory Server コンソールに適用されます。

1. Directory Server コンソールの最上位の「ディレクトリ」タブで、ディレクトリツリーを展開します。
2. `cn=config` エントリを右クリックし、ポップアップメニューから「新規」>「ユーザー」の順に選択します。または、`cn=config` エントリを選択し、「オブジェクト」メニューから「新規」>「ユーザー」の順に選択します。
3. 「新規ユーザーの作成」ダイアログボックスの各フィールドに、プロキシ ID の説明となる値を入力します。たとえば、次のように入力します。

名 : proxy
姓 : *host1*
共通名 : *host1* chaining proxy
ユーザー ID: *host1*_proxy
パスワード : *password*
パスワードの確認 : *password*

host1 は、連鎖サフィックスを格納するサーバーの名前です。このサーバーに連鎖されているサフィックスを持つサーバーごとに異なるプロキシ ID を使用する必要があります。

4. 「了解」をクリックして、この新しいプロキシ ID を保存します。

コマンド行からのプロキシ ID の作成

この手順では、連鎖サフィックスを格納するローカルサーバーに *host1* を使用し、連鎖サフィックスのターゲットであるリモートサーバーに *host2* を使用します。

1. 次のコマンドを使用して、*host2* にプロキシ ID を作成します。

```
ldapmodify -a -h host2 -p port2 -D "cn=Directory Manager" -w password2
dn: uid=host1_proxy,cn=config
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgperson
uid: host1_proxy
cn: host1 chaining proxy
sn: host1
userpassword: password
description: proxy entry to be used for chaining from host1
^D
```

デフォルト連鎖パラメータの設定

連鎖パラメータによって、連鎖サーバーへのサーバーの接続方法と連鎖サフィックスに対する操作の処理方法が決まります。これらのパラメータは連鎖サフィックスごとに設定されます。Directory Server は、連鎖サフィックスの作成時に使用されるデフォルト値を提供します。これらのデフォルト値を編集して、すべての新しい連鎖サフィックスに関する連鎖パラメータを変更することもできます。

デフォルトパラメータの変更後に新しく作成されるすべての連鎖サフィックスに、指定した値が設定されます。ただし、作成済みのサフィックスのパラメータは、115 ページの「連鎖サフィックスの管理」で説明されている方法でしか変更できません。

連鎖パラメータの属性値とデフォルト値について、次に説明します。有効な値については、『Sun ONE Directory Server Reference Manual』の第 5 章にある「Chained Suffix Plug-in Attributes」を参照してください。

クライアント戻りパラメータ

- `nsReferralOnScopedSearch` : オン (デフォルト) の場合、連鎖サフィックス全体を適用範囲としたクライアントの検索には、リモートサーバーへのリフェラルが返されます。これにより、検索結果の二重送信が回避されます。オフに設定した場合は、サイズと時間の制限パラメータを設定して、連鎖サフィックスに対して長時間の検索が行われることを回避する必要があります。
- `nsslapd-sizelimit` : このパラメータは、連鎖された検索操作への応答として返されるエントリ数を決定します。デフォルトは 2000 エントリです。連鎖サフィックスを含む広範囲の検索を制限する場合は、このパラメータの値を小さくします。どのような場合でも、操作はリモートサーバー上のサイズ設定によって制限を受けます。
- `nsslapd-timelimit` : このパラメータは、連鎖操作の時間の長さを制御します。デフォルトの制限時間は 3600 秒 (1 時間) です。連鎖サフィックスに対する操作に有効な時間を制限する場合は、このパラメータの値を小さくします。どのような場合でも、操作はリモートサーバー上の時間設定によって制限を受けます。

カスケード型連鎖パラメータ

- `nsCheckLocalACI` : 単一レベルの連鎖では、ローカルサーバーは、連鎖サフィックス上でバインドされたユーザーのアクセス権をチェックしません。これは、リモートサーバーが行うためです。このため、デフォルト値は `off` です。ただし、カスケード型連鎖の中間サーバーでは、このパラメータを `on` に設定して、連鎖操作を転送するサーバーが使用するプロキシ DN のアクセス権をチェックおよび制限する必要があります。
- `nsHopLimit` : ループ検出は、このパラメータを使用して、有効な最大ホップ数を定義します。このホップ数に達した連鎖操作は転送されず、カスケードトポロジに異常なループがあると見なして中断されます。

接続管理パラメータ

- `nsOperationConnectionsLimit` : 連鎖サフィックスがリモートサーバーとの間で同時に確立できる LDAP 接続の最大数を指定します。デフォルトは 10 です。
- `nsBindConnectionsLimit` : 連鎖サフィックスがリモートサーバーとの間で同時に確立できる TCP 接続の最大数を指定します。デフォルトは 3 です。
- `nsConcurrentBindLimit` : LDAP 接続ごとに同時に行うことができるバインド操作の最大数を指定します。デフォルト値は、接続ごとに 10 です。

- nsBindRetryLimit : エラー発生時に連鎖サフィックスがリモートサーバーとの再バインドを試行する回数を指定します。「0」を指定すると、連鎖サフィックスは 1 回だけバインドを試みます。デフォルトは 3 です。
- nsConcurrentOperationsLimit : LDAP 接続ごとに同時に行うことができる操作の最大数を指定します。デフォルト値は 1 接続あたり 10 です。
- nsBindTimeout : 連鎖サフィックスへのバインド操作の試行がタイムアウトになるまでの時間 (秒) を指定します。デフォルトは 15 秒です。
- nsAbandonedSearchCheckInterval : 操作が中断されているかどうかをサーバーがチェックするまでの秒数を指定します。デフォルトは 2 秒です。
- nsConnectionLife : 連鎖サフィックスとリモートサーバー間で確立された接続を再利用可能な接続時間を指定します。接続したままにすると、処理は速くなりますが、リソースが多く使用されます。たとえば、ダイヤルアップ接続を使用している場合は、接続時間を制限する必要があります。デフォルトは 0 で、接続時間に制限はありません。

エラー検出パラメータ

- nsmaxresponsedelay : 連鎖操作に対する LDAP 要求の開始にリモートサーバーが応答するまでにかかる最大時間を指定します。この時間は秒単位で指定します。ローカルサーバーは、この遅延時間が経過したあとに接続をテストします。デフォルトの遅延時間は 60 秒です。
- nsmaxtestresponsedelay : リモートサーバーが応答しているかどうかをチェックするテストの持続時間を指定します。テストは、存在しないエントリに対する簡単な検索要求です。この時間は秒単位で指定します。テストの遅延時間内に応答が受信されない場合、連鎖サフィックスはリモートサーバーが停止していると見なします。デフォルトのテスト応答遅延時間は 15 秒です。

  該当の連鎖サフィックスに対してリモートサーバーを 1 つだけ定義している場合、リモートサーバーへのすべての連鎖操作は、過度の負荷を防ぐために 30 秒間ブロックされます。フェイルオーバーサーバーを定義している場合、連鎖操作は次に定義されている別のサーバーの使用を開始します。

コンソールからのデフォルト連鎖パラメータの設定

1. Directory Server コンソールの最上位の「ディレクトリ」タブで、ディレクトリツリーを展開し、次のエントリを選択します。`cn=default instance config,cn=chaining database,cn=plugins,cn=config`
2. このエントリをダブルクリックするか、「オブジェクト」>「汎用エディタで編集」メニューの順に選択します。上記のリストのうち、目的の属性値を変更します。
3. 「汎用エディタ」ダイアログボックスの「保存」をクリックします。変更はただちに有効になります。

コマンド行からのデフォルト連鎖パラメータの設定

1. `ldapmodify` コマンドを使用して、エントリ `cn=default instance config,cn=chaining database,cn=plugins,cn=config` を編集します。このエントリのすべての属性が、新しい連鎖サフィックスのパラメータのデフォルト値になります。

   たとえば、次のコマンドでは、新しい連鎖サフィックスのデフォルトのサイズ制限が 5000 エントリに増加し、デフォルトの時間制限が 10 分に減少します。

   ```
   ldapmodify -h host -p port -D "cn=Directory Manager" -w password
   dn: cn=default instance config,cn=chaining database,
    cn=plugins,cn=config
   changetype: modify
   replace: nsslapd-sizelimit
   nsslapd-sizelimit: 5000
   -
   replace: nsslapd-timelimit
   nsslapd-timelimit: 600
   ^D
   ```

   このエントリへの変更は、ただちに有効になります。

コンソールからの連鎖サフィックスの作成

次の手順は、連鎖ルートサフィックスと連鎖サブサフィックスを作成する場合とほぼ同じです。

1. Directory Server コンソールの「設定」タブを選択します。

- 連鎖ルートサフィックスについては、「データ」ノードを右クリックし、ポップアップメニューから「新規連鎖サフィックス」を選択します。または、「データ」ノードを選択し、「オブジェクト」メニューから「新規連鎖サフィックス」を選択します。
- 連鎖サブサフィックスについては、「データ」ノードと任意のサフィックスノードを展開して、親サフィックスを表示します。親サフィックスのノードを右クリックし、ポップアップメニューから「新規連鎖サブサフィックス」を選択します。または、親サフィックスのノードを選択し、「オブジェクト」メニューから「新規連鎖サブサフィックス」を選択します。

  「新規連鎖 (サブ) サフィックス」ダイアログボックスが表示されます。

2. 連鎖するリモートサーバー上のエントリの DN を入力します。リモートエントリは、リモートサフィックスのベースエントリである必要はありません。

- ルートサフィックスについては、「サフィックス DN」フィールドにリモートエントリの完全 DN を入力します。リモートディレクトリツリー内のエントリである DN のどれかを入力できます。このエントリは連鎖ルートサフィックスのベースになり、連鎖サフィックスを通じて、その下位にあるすべてのエントリを使用できます。
- サブサフィックスについては、連鎖されるエントリのサブサフィックス RDN を入力します。このエントリは、連鎖サブサフィックスのベースになります。テキストフィールドの下に表示される完全なサブサフィックス名は、リモートサーバー内に存在するエントリである必要があります。

3. サフィックスデータを格納するリモートサーバーのホスト名を入力します。必要に応じてドメインも指定します。
4. リモートサーバーにアクセスするためのポート番号を入力し、セキュリティ保護されたポートである場合はチェックボックスを選択します。セキュリティ保護されたポートを使用する場合、連鎖操作は SSL により暗号化されます。詳細は、114 ページの「SSL を使用した連鎖」を参照してください。

   ダイアログの最下部には、リモートサーバーの完全 URL が表示されます。
5. リモートサーバーのプロキシ ID のバインド DN とパスワードを入力します。ローカルサーバーは、リモートサーバー上のサフィックスの内容にアクセスするときに、この DN をプロキシとして使用します。たとえば、103 ページの「プロキシ ID の作成」で定義されている `uid=`*host1*`_proxy,cn=config` DN を使用します。

   リモートサーバー上の Directory Manager の DN を使用することはできません。連鎖サフィックスを通じて実行される操作は、`creatorsName` 属性と `modifiersName` 属性で、このプロキシ ID を使用します。プロキシ DN を省略することもできますが、その場合、ローカルサーバーはリモートサーバーへのアクセス時に匿名でバインドします。
6. 「了解」をクリックして、連鎖サフィックスを作成します。新しいサフィックスが、設定ツリーに連鎖を示すアイコンとともに表示されます。
7. 新しい連鎖サフィックスをクリックして選択し、右側のパネルで「リモートサーバー」タブを選択します。
8. 必要に応じて、この連鎖サフィックスに対する 1 つ以上のフェイルオーバーサーバーを定義することもできます。サーバーは、リモートサーバーに接続できない場合、応答が得られるまで、定義された順に各フェイルオーバーサーバーへの接続を試行します。フェイルオーバーサーバーには、連鎖サフィックスと同じサフィックスが格納されていて、プロキシに対して同じバインド DN が許可されている必要があります。

   複数のフェイルオーバーサーバーを定義するには、「リモートサーバー URL」フィールドに、ホスト名とポート番号のペアを空白で区切って入力します。このフィールドの書式は次のとおりです。

   ```
   ldap[s]://hostname[:port][ hostname[:port]].../
   ```

9. 「リモートサーバー」タブの最下部のテキストボックスには、プロキシで許可された連鎖操作を行うために必要な ACI が表示されます。この ACI を、リモートサーバーの *suffixDN* とともにエントリに追加する必要があります。フェイルオーバーサーバーを定義している場合は、ACI をすべてのフェイルオーバーサーバーに追加する必要があります。「ACI をコピー」ボタンを使用して、ACI のテキストをペーストできるように、システムのクリップボードにコピーします。

   この ACI がリモートサーバーのベースエントリに追加されると、連鎖サフィックスがローカルサーバーのディレクトリツリーに表示されます。

警告 連鎖によって公開されているリモートサーバーへのアクセスを制限するには、同じエントリで別の ACI を定義することが必要な場合もあります。詳細は、113 ページの「連鎖サフィックスのアクセス制御」を参照してください。

10. サーバーコンポーネントに対して連鎖ポリシーを設定済みである場合は、これらのコンポーネントのリモートサーバーへのアクセスを許可する ACI を追加することも必要です。たとえば、参照整合性検査プラグインの連鎖を許可する場合、手順 2 で指定した DN を持つベースエントリに、次の ACI を追加する必要があります。

```
aci: (targetattr "*")
 (target="ldap:///suffixDN")
 (version 3.0; acl "RefInt Access for chaining"; allow
 (read,write,search,compare) userdn = "ldap:///cn=referential
 integrity postoperation,cn=plugins,cn=config";)
```

コマンド行からの連鎖サフィックスの作成

`ldapmodify` コマンド行ユーティリティを使用しても、ディレクトリ内に連鎖サフィックスを作成できます。連鎖ルートサフィックスと連鎖サブサフィックスは、サーバーによって内部的に同じ方法で管理されるため、それらをコマンド行から作成する手順はほとんど同じです。

1. 次のコマンドを使用して、連鎖ルートサフィックスの連鎖サフィックスエントリを `cn=mapping tree,cn=config` の下に作成します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=suffixDN,cn=mapping tree,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsMappingTree
```

```
cn: suffixDN
nsslapd-state: backend
nsslapd-backend: databaseName
^D
```

連鎖サブサフィックスについては、上記のコマンドに次の属性を追加します。
`nsslapd-parent-suffix: parentSuffixDN`

連鎖サブサフィックスでは、*suffixDN* は、サブサフィックスの RDN と親サフィックスの DN となります。たとえば、`l=Europe,dc=example,dc=com` のようになります。*suffixDN* には、リモートサーバーから使用可能なエントリの DN を指定する必要がありますが、リモートサフィックスのベースエントリである必要はありません。

注 サフィックス名は DN 形式となりますが、単一の文字列として扱われます。そのため、空白はすべて意味を持ち、サフィックス名の一部となります。サーバーがリモートエントリにアクセスするためには、*suffixDN* の文字列にはリモートサフィックスと同じ空白を使用する必要があります。

databaseName は、連鎖プラグインコンポーネントが連鎖サフィックスを識別するために使用する別名 (ニックネーム) です。この名前は、すべてのサフィックスの *databaseNames* で一意となる必要があります。規則では、*suffixDN* の 1 番目のネーミングコンポーネントの値となります。ローカルサフィックスとは異なり、連鎖サフィックスはローカルサーバー上にデータベースファイルを保持しません。

サブサフィックスでは、*parentSuffixDN* は親サフィックスの正確な DN です。親は、ローカルサフィックスまたは連鎖サフィックスのどちらかです。

2. 次のコマンドを使用して、連鎖設定エントリを作成します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=databaseName,cn=chaining database,cn=plugins,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsBackendInstance
cn: databaseName
nsslapd-suffix: suffixDN
nsfarmserverurl: LDAPURL
nsmultiplexorbinddn: proxyDN
nsmultiplexorcredentials: ProxyPassword
^D
```

databaseName と *suffixDN* は、前の手順の値と同じにする必要があります。*LDAPURL* リモートサーバーの URL ですが、サフィックスに関する情報は含まれません。URL には、次の書式で複数のフェイルオーバーサーバーを指定できます。

`ldap[s]://`*hostname*`[:`*port*`][ `*hostname*`[:`*port*`]]...`*/*

LDAP URL 内のすべてのリモートサーバーには、*suffixDN* を指定する必要があります。セキュリティ保護されたポートの指定方法については、114 ページの「SSL を使用した連鎖」を参照してください。

proxyDN には、リモートサーバー上のプロキシ ID の DN を指定します。ローカルサーバーは、リモートサーバー上のサフィックスの内容にアクセスするときに、この DN をプロキシとして使用します。連鎖サフィックスを通じて実行される操作は、`creatorsName` 属性と `modifiersName` 属性で、このプロキシ ID を使用します。プロキシ DN を指定しない場合、ローカルサーバーはリモートサーバーへのアクセス時に匿名でバインドします。

ProxyPassword はプロキシ DN のパスワードで、この値は暗号化されません。パスワードは、設定ファイルに格納されるときに暗号化されます。次に例を示します。

```
nsmultiplexorbinddn: uid=host1_proxy,cn=config
nsmultiplexorcredentials: secret
```

警告 パスワードをそのまま送信しないために、暗号化されたポートから `ldapmodify` コマンドを実行する必要があります。

新しいエントリは、すべての連鎖パラメータに自動的に追加されます。その際、`cn=default instance config,cn=chaining` `database,cn=plugins,cn=config` で定義されたデフォルト値が使用されます。連鎖設定エントリの作成時、属性値に別の値を設定すると、デフォルト値を上書きできます。値を定義できる属性のリストは、104 ページの「デフォルト連鎖パラメータの設定」を参照してください。

3. 次のコマンドを使用して、リモートエントリに ACI を作成します。この ACI は、プロキシで許可された連鎖操作を行うために必要です。ACI については、第 6 章「アクセス制御の管理」を参照してください。

```
ldapmodify -h host2 -p port2 -D "cn=Directory Manager" -w password2
dn: suffixDN
changetype: modify
add: aci
aci: (targetattr=*)(target = "ldap:///suffixDN")(version 3.0;acl
 "Allows use of admin for chaining"; allow (proxy)
 (userdn="ldap:///proxyDN");)
^D
```

警告 このサーバーから公開されているリモートサーバーへのアクセスを制限するには、同じエントリで別の ACI を定義することが必要な場合もあります。詳細は、113 ページの「連鎖サフィックスのアクセス制御」を参照してください。

4. サーバーコンポーネントに対して連鎖ポリシーを設定済みである場合は、これらのコンポーネントのリモートサーバーへのアクセスを許可する ACI を追加することも必要です。たとえば、参照整合性検査プラグインの連鎖を許可する場合、*suffixDN* が指定されているベースエントリに、次の ACI を追加する必要があります。

```
aci: (targetattr "*")
 (target="ldap:///suffixDN")
 (version 3.0; acl "RefInt Access for chaining"; allow
 (read,write,search,compare) userdn = "ldap:///cn=referential
 integrity postoperation,cn=plugins,cn=config";)
```

次のコマンドは、連鎖サブサフィックスの作成例です。DN 内のネーミング属性に指定する場合、suffixDN 内のコンマの前にエスケープ文字の円記号 (¥) を付ける必要があります。

コード例 3-1 コマンド行からの連鎖サフィックスの作成

```
ldapmodify -a -h host1 -p port1 -D "cn=Directory Manager" -w password1
dn: cn=l=Europe¥,dc=example¥,dc=com,cn=mapping tree,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsMappingTree
cn: l=Europe,dc=example,dc=com
nsslapd-state: backend
nsslapd-backend: Europe
nsslapd-parent-suffix: dc=example,dc=com

dn: cn=Europe,cn=chaining database,cn=plugins,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsBackendInstance
cn: Europe
nsslapd-suffix: l=Europe,dc=example,dc=com
nsfarmserverurl: ldap://host2:port2/
nsmultiplexorbinddn: uid=host1_proxy,cn=config
nsmultiplexorcredentials: proxyPassword
^D

ldapmodify -h host2 -p port2 -D "cn=Directory Manager" -w password2
dn: l=Europe,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr=*)(target =
 "ldap:///l=Europe,dc=example,dc=com")(version 3.0;acl
 "Allows use of admin for chaining"; allow (proxy)
 (userdn="ldap:///uid=host1_proxy,cn=config");)
^D
```

連鎖サフィックスのアクセス制御

認証ユーザーが連鎖サフィックスにアクセスすると、サーバーはユーザーの ID をリモートサーバーに送信します。ここでのアクセス制御は、常にリモートサーバーで評価されます。リモートサーバーで評価される LDAP 操作では、プロキシ承認制御により渡されたクライアントアプリケーションのオリジナル ID が使用されます。ユーザーが、リモートサーバーに含まれるサブツリーに対して正しいアクセス制御を持っている場合にだけ、リモートサーバーで操作が成功します。つまり、リモートサーバーには、通常のアクセス制御を追加しておく必要があります。これには次のような制約があります。

- すべてのタイプのアクセス制御を使用できるとは限らない

  たとえば、ロールベースの ACI やフィルタベースの ACI では、ユーザーエントリへのアクセス権が必要です。連鎖サフィックスを経由してデータにアクセスしているので、プロキシ制御にあるデータだけが検証されます。つまり、ユーザーエントリがユーザーのデータと同じサフィックスに必ず置かれるように、ディレクトリを設計しておく必要があります。

- クライアントのオリジナルドメインが連鎖中に失われるので、クライアントの IP アドレスや DNS ドメインに基づくアクセス制御がすべて動作しないことがある

  リモートサーバーでは、クライアントアプリケーションは連鎖サフィックスと同じ IP アドレスにあり、同じ DNS ドメインに存在するものと見なされます。

連鎖サフィックスとともに使用するために作成する ACI には、次の制約があります。

- ACI は、ACI が使用するグループと同じサーバーにある必要がある。ダイナミックグループの場合は、グループのすべてのユーザーが、ACI および ACI が使用するグループと同じ場所にある必要がある。スタティックグループの場合は、リモートユーザーが参照される場合がある
- ACI は、ACI が使用するロール定義と同じサーバー上にあり、これらのロールを持つ予定のユーザーも同じ場所に存在する必要がある
- ユーザーのエントリ値を参照する ACI (たとえば、`userattr` サブジェクトルール) は、ユーザーがリモートの場合に機能する

アクセス制御は常にリモートサーバーで評価されますが、連鎖サフィックスを含むサーバーとリモートサーバーの両方でアクセス制御が評価されるように選択することもできます。これにはいくつかの制約があります。

- アクセス制御の評価中に、ユーザーエントリの内容が使用できるとは限らない (たとえば、連鎖サフィックスが含まれるサーバーでアクセス制御が評価され、エントリがリモートサーバーにある場合)

  パフォーマンス上の理由から、クライアントがリモートの問い合わせやアクセス制御の評価を行うことはできません。

- 連鎖サフィックスに、クライアントアプリケーションによって変更されているエントリへのアクセス権があるとは限らない

  変更操作を実行するときに、リモートサーバーに格納されているすべてのエントリへのアクセス権が、連鎖サフィックスにあるとは限りません。削除操作を実行する場合、連鎖サフィックスはエントリの DN だけを認識します。アクセス制御で特定の属性が指定されている場合は、連鎖サフィックスを通じて削除操作を実行すると失敗します。

デフォルトでは、連鎖サフィックスを含むサーバーで設定されたアクセス制御は評価されません。このデフォルト値を上書きするには、`cn=`*databaseName*`,cn=chaining database,cn=plugins,cn=config` エントリの `nsCheckLocalACI` 属性を使用します。ただし、連鎖サフィックスを含むサーバーでアクセス制御を評価することは、カスケード型連鎖を使用している場合を除き、お勧めできません。詳細は、129 ページの「カスケード型連鎖の設定」を参照してください。

SSL を使用した連鎖

連鎖サフィックス上の操作を実行するときに、SSL を使用してリモートサーバーと通信するように、サーバーを設定できます。連鎖で SSL を使用するには、次の手順を実行します。

1. リモートサーバーで、SSL を有効にします。
2. 連鎖サフィックスが含まれるサーバーで SSL を有効にします。

   SSL の有効化については、第 11 章「セキュリティの実装」を参照してください。
3. 連鎖サフィックスの作成または変更の手順で、リモートサーバーの SSL とセキュリティ保護されたポートを指定します。

   コンソールを使用する場合は、連鎖サフィックスの作成または設定の手順で、セキュリティ保護されたポートのチェックボックスを選択します。詳細は、107 ページの「コンソールからの連鎖サフィックスの作成」または 118 ページの「コンソールからの連鎖ポリシーの変更」を参照してください。

   コマンド行を使用して実行する場合は、リモートサーバーの LDAPS URL とセキュリティ保護されたポートを `ldaps://example.com:636/` のように指定します。詳細は、109 ページの「コマンド行からの連鎖サフィックスの作成」または 119 ページの「コマンド行からの連鎖ポリシーの変更」を参照してください。

SSL を使用して通信するように連鎖サフィックスとリモートサーバーを設定すると、操作を要求するクライアントアプリケーションが SSL を使用して通信する必要はありません。クライアントは、LDAP プロトコルまたは DSML プロトコルのポートを使用できます。

連鎖サフィックスの管理

ここでは、既存の連鎖サフィックスの更新方法と削除方法について説明します。また、連鎖メカニズムの制御方法についても説明します。

連鎖ポリシーの設定

サーバーの連鎖ポリシーによって、連鎖されたサーバーに伝達する LDAP 制御と、連鎖サフィックスへのアクセスを許可するサーバーコンポーネントを決定します。連鎖ポリシーの設定と、連鎖サフィックスに関わる操作への影響については、認識しておく必要があります。連鎖ポリシーは、サーバー上のすべての連鎖サフィックスに適用されます。

デフォルトの設定は、通常の操作を透過的に実行できることを目的としています。ただし、操作に LDAP 制御が含まれる場合、あるいは参照整合性検査プラグインなどのサーバーコンポーネントを使用している場合は、要件を満たす連鎖ポリシーを設定する必要があります。

連鎖サフィックスを作成する前に連鎖ポリシーを設定しておくことが最適です。そのようにすると、連鎖サフィックスを有効にした時点でただちにポリシーが適用されます。ただし、その後いつでもポリシーを変更できます。

LDAP 制御の連鎖ポリシー

LDAP 制御は要求の一部としてクライアントから送信され、操作またはその結果が何らかの方法によって変更されます。サーバーの連鎖ポリシーによって、サーバーが操作とともにどの制御を連鎖サフィックスへ転送するかが決まります。デフォルトでは、次の制御が連鎖サフィックスのリモートサーバーに転送されます。

表 3-1 デフォルトで連鎖に許可される LDAP 制御

制御の OID	制御の名前と説明
1.2.840.113556.1.4.473	サーバー側ソート : 検索と関連付けられて、属性値に従って結果のエントリをソートする *
1.3.6.1.4.1.1466.29539.12	連鎖ループの検出 : サーバーが別のサーバーと連鎖する回数を記録する。この回数が設定した値に達すると、操作は中断され、クライアントアプリケーションに通知される。詳細は、131 ページの「カスケード型連鎖の LDAP 制御の送信」を参照してください。
2.16.840.1.113730.3.4.2	スマートリフェラルの管理 DSA : リフェラルに従わずに、スマートリフェラルをエントリとして返す。スマートリフェラル自体を変更または削除できる

表 3-1　デフォルトで連鎖に許可される LDAP 制御 (続き)

制御の OID	制御の名前と説明
2.16.840.1.113730.3.4.9	VLV (仮想リスト表示) : 検索に対してすべての結果エントリを一度に返すのではなく、部分的に結果を表示する *

(*) サーバー側ソート制御と VLV 制御は、検索範囲が 1 つのサフィックスである場合にだけ、連鎖を通じてサポートされます。クライアントアプリケーションからの要求が複数のサフィックスに対して行われた場合は、連鎖サフィックスでは、VLV 制御はサポートされません。

次の表に、連鎖ポリシーを設定して連鎖を許可できる、その他の LDAP 制御を示します。

表 3-2　連鎖できる LDAP 制御

制御の OID	制御の名前と説明
1.3.6.1.4.1.42.2.27.9.5.2	有効な権限の要求 : 結果のエントリと属性に関連するアクセス権と ACI に関する情報を返すようサーバーに要求する
2.16.840.1.113730.3.4.3	持続検索 : サーバーが操作を有効な状態に維持し、検索フィルタに一致するエントリが追加、削除、または変更されるたびに結果をクライアントに送信することを示す
2.16.840.1.113730.3.4.4	パスワード期限切れの通知 : パスワードが期限切れになったことをクライアントアプリケーションに通知する
2.16.840.1.113730.3.4.5	パスワード期限切れ予告通知 : パスワードが一定期間内に期限切れになることをクライアントアプリケーションに通知する
2.16.840.1.113730.3.4.12	プロキシ承認 (旧仕様) : 要求中に別の識別情報を使用することを、クライアントに許可する *
2.16.840.1.113730.3.4.13	レプリケーション更新情報 : レプリケーション操作の UUID (汎用的な一意の識別子) と CSN (変更シーケンス番号) を送信する
2.16.840.1.113730.3.4.14	特定データベース検索 : 検索操作で使用され、制御で指定されているデータベース上で検索が実行されるように指定する
2.16.840.1.113730.3.4.15	認証応答 : バインド応答とともにクライアントアプリケーションに返され、使われる DN と認証方法を通知する。SASL または証明書を使用する場合に役立つ
2.16.840.1.113730.3.4.16	認証要求 : バインド要求とともに通知され、バインド応答で証明書を返すようにサーバーに要求する

表 3-2　連鎖できる LDAP 制御 (続き)

制御の OID	制御の名前と説明
2.16.840.1.113730.3.4.17	実際の属性だけの要求 : サーバーはエントリ内に実際に含まれる属性だけを返し、仮想属性を解決する必要がないことを示す
2.16.840.1.113730.3.4.18	プロキシ承認 (新仕様) : 要求中に別の識別情報を使用することを、クライアントに許可する *
2.16.840.1.113730.3.4.19	仮想属性だけの要求 : サーバーはサービス機能のロールとクラスによって生成された属性だけを返すことを示す

(*) アプリケーションは、プロキシ承認 にどちらの制御も使用できます。これらの OID には同じ連鎖ポリシーを設定する必要があります。詳細は、131 ページの「カスケード型連鎖の LDAP 制御の送信」を参照してください。

サーバーコンポーネントの連鎖ポリシー

コンポーネントとは、内部操作を使用するサーバーの機能単位です。たとえば、プラグインはコンポーネントです。ほとんどのコンポーネントのタスクを実行するには、コンポーネントはディレクトリの内容にアクセスする必要があります。ディレクトリの内容とは、ディレクトリ内に格納されている設定データまたはユーザーデータです。

デフォルトでは、サーバーコンポーネントの連鎖は禁止されています。コンポーネントから連鎖サフィックスにアクセスする場合は、連鎖を明示的に許可する必要があります。連鎖されたデータにアクセスできるコンポーネントの DN を次に示します。

107 ページの「コンソールからの連鎖サフィックスの作成」で説明されるように、連鎖できるようにするには、リモートサーバー上の ACI で適切な権限を許可する必要があります。サーバーコンポーネントを連鎖するときには、この ACI で検索、読み取り、および比較を許可して、サーバーコンポーネントがこれらの操作を実行できるようにする必要があります。さらに、コンポーネントによっては、次に説明するようにリモートサーバー上での書き込み権限が必要となる場合もあります。

- `cn=ACL Plugin,cn=plugins,cn=config` : この ACL プラグインは、アクセス制御機能を実装する。ローカルとリモートの ACI 属性を混在させると危険なので、ACI 属性を検出および更新するための処理は連鎖されない。ただし、ユーザーエントリへのアクセスに使用する要求は連鎖できる。ACI および連鎖の制限については、188 ページの「ACI の制限事項」を参照
- `cn=old plugin,cn=plugins,cn=config` : このプラグインは、すべての Directory Server 4.x プラグインと、それらに連鎖が許可されているかどうかを表す。4.x プラグインの連鎖ポリシーはすべて同じ。4.x プラグインによって実行される操作によっては、リモートサーバー上に ACI を設定することが必要な場合がある

- `cn=resource limits,cn=components,cn=config`: このコンポーネントは、ユーザーバインド DN に基づいてリソースの使用制限を設定する。このコンポーネントが連鎖を許可された場合、連鎖サフィックスに ID が格納されているユーザーに対してリソースの使用を制限できる
- `cn=certificate-based authentication,cn=components,cn=config`: このコンポーネントは、SASL 外部バインド方法とともに使用される。リモートサーバーからユーザー証明書を取得する

警告 連鎖サフィックスから証明書に基づく認証を許可すると、セキュリティホールが作成される場合があります。信頼されないリモートサーバーに別のサフィックスが連鎖されている場合、信頼されないサーバー上の証明書が認証に使われる可能性があります。

- `cn=referential integrity postoperation,cn=plugins,cn=config`: このプラグインは、エントリが削除されたときに、グループメンバーのリストなど、その DN を参照していた別のエントリに伝達する。連鎖でこのプラグインを使用すると、グループのメンバーが連鎖サフィックスに格納されているときに、スタティックグループの管理が簡単になる。このプラグインが連鎖サフィックスにアクセスするには、リモートサーバー上で書き込み権限が必要となる
- `cn=uid uniqueness,cn=plugins,cn=config`: UID 一意性検査プラグインは、新しく指定された属性のすべての値がサーバー上で一意であることを確認する。このプラグインの連鎖を許可すると、ディレクトリツリー全体での一意性が保持される

注 次のコンポーネントは連鎖できません。

- ロールプラグイン
- パスワードポリシーコンポーネント
- レプリケーションプラグイン

コンソールからの連鎖ポリシーの変更

1. Directory Server コンソールの「設定」タブで、「データ」ノードを選択し、右側のパネルで「連鎖」タブを選択します。
2. 右側のリストから 1 つ以上の LDAP 制御を選択し、「追加」をクリックして、それらの連鎖を許可します。「追加」ボタンと「削除」ボタンを使用して、連鎖を許可する制御のリストを作成します。

   LDAP 制御は OID で示されます。各制御の名前と説明については、115 ページの「LDAP 制御の連鎖ポリシー」を参照してください。

3. 連鎖を許可されたサーバーコンポーネントが、同じタブの下部に一覧表示されます。右側のリストから 1 つ以上のコンポーネント名を選択し、「追加」をクリックして、それらの連鎖を許可します。「追加」ボタンと「削除」ボタンを使用して、連鎖を許可するコンポーネントのリストを作成します。

   各コンポーネントの説明については、117 ページの「サーバーコンポーネントの連鎖ポリシー」を参照してください。

4. 「保存」をクリックして、連鎖ポリシーを保存します。

5. 変更内容を有効にするために、サーバーを再起動します。

コマンド行からの連鎖ポリシーの変更

`cn=config,cn=chaining database,cn=plugins,cn=config` エントリには、連鎖ポリシー設定のための属性が含まれます。`ldapmodify` コマンドを使用して、このエントリを編集します。

1. 複数値属性の `nsTransmittedControls` を変更して、連鎖を許可するすべての LDAP 制御の OID を指定します。連鎖できる制御の OID については、115 ページの「LDAP 制御の連鎖ポリシー」を参照してください。

   たとえば、次のコマンドによって、有効な権限の制御が、連鎖される制御のリストに追加されます。

   ```
   ldapmodify -h host -p port -D "cn=Directory Manager" -w password
   dn: cn=config,cn=chaining database,cn=plugins,cn=config
   changetype: modify
   add: nsTransmittedControls
   nsTransmittedControls: 1.3.6.1.4.1.42.2.27.9.5.2
   ^D
   ```

   クライアントアプリケーションがカスタム制御を使用していて、それらの連鎖を許可する場合には、それらのカスタム制御の OID を `nsTransmittedControls` 属性に追加することもできます。

2. 複数値属性の `nsActiveChainingComponents` を変更して、連鎖を許可するすべてのサーバーコンポーネントの DN を指定します。各コンポーネントの説明については、117 ページの「サーバーコンポーネントの連鎖ポリシー」を参照してください。

   たとえば、次のコマンドによって、参照整合性コンポーネントが、連鎖されるコンポーネントのリストに追加されます。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=config,cn=chaining database,cn=plugins,cn=config
changetype: modify
add: nsActiveChainingComponents
nsActiveChainingComponents: cn=referential integrity
 postoperation,cn=components,cn=config
^D
```

3. 連鎖ポリシーの設定エントリを変更したら、変更内容を有効にするにはサーバーを再起動する必要があります。

連鎖サフィックスの無効化と有効化

保守やセキュリティ上の理由から、連鎖サフィックスを使用できないようにすることが必要な場合があります。サフィックスを無効にすると、サーバーは、サフィックスへのアクセスを試みたクライアント操作への応答時に、リモートサーバーに接続できなくなります。デフォルトのリフェラルを定義している場合は、無効になっているサフィックスにクライアントがアクセスしようとすると、デフォルトのリフェラルが返されます。

コンソールからの連鎖サフィックスの無効化と有効化

1. Directory Server コンソールの最上位の「設定」タブで「データ」ノードを展開し、無効になる連鎖サフィックスを選択します。
2. 右側のパネルで、「設定」タブを選択します。デフォルトでは、すべての連鎖サフィックスは作成した時点で有効になります。
3. 「このサフィックスへのアクセスを有効に」チェックボックスの選択を解除して、サフィックスを無効にします。または、このチェックボックスを選択して、サフィックスを有効にします。
4. 「保存」をクリックして、変更内容を適用します。サフィックスは、ただちに無効または有効になります。
5. 必要に応じて、サフィックスが無効である間にサフィックスへのすべての操作に対して返されるグローバルのデフォルトリフェラルを設定できます。これは、最上位の「設定」タブのルートノードの「ネットワーク」タブで設定します。詳細は、73 ページの「コンソールからのデフォルトリフェラルの設定」を参照してください。

コマンド行からのサフィックスの無効化と有効化

1. 次のコマンドを使用して、連鎖サフィックスエントリ内の `nsslapd-state` 属性を編集します。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=suffixDN,cn=mapping tree,cn=config
changetype: modify
replace: nsslapd-state
nsslapd-state: disabled or backend
^D
```

suffixDN は、定義済みのサフィックス DN の完全な文字列です。値には空白やコンマのエスケープ文字である円記号 (¥) も含まれます。`nssladp-state` 属性の値を `disabled` に設定するとサフィックスは無効になり、`backend` に設定するとすべてのアクセス権が許可されます。

コマンドの実行が成功すると、サフィックスはただちに無効になります。

2. 必要に応じて、サフィックスが無効である間にサフィックスへのすべての操作に対して返されるグローバルのデフォルトリフェラルを設定できます。詳細は、74 ページの「コマンド行からのデフォルトリフェラルの設定」を参照してください。

アクセス権とリフェラルの設定

連鎖サフィックスを完全に無効にすることなくサフィックスへのアクセスを制限するには、アクセス権を変更して読み取り専用アクセスを許可することもできます。この場合、書き込み操作に対しては、別のサーバーへのリフェラルを定義する必要があります。また、読み取りアクセスと書き込みアクセスの両方を拒否し、サフィックスへのすべての操作に対するリフェラルを定義することもできます。

リフェラルの概要については、『Sun ONE Directory Server Deployment Guide』を参照してください。

コンソールからのアクセス権とリフェラルの設定

1. Directory Server コンソールの最上位の「設定」タブで「データ」ノードを展開し、リフェラルを設定する連鎖サフィックスを選択します。
2. 右側のパネルで、「設定」タブを選択します。連鎖サフィックスが有効である場合は、アクセス権とリフェラルだけを設定できます。
3. 次のラジオボタンのどれかを選択して、該当のサフィックスのエントリに対する書き込み操作への応答を設定します。
   - 書き込みおよび読み取り要求を処理 : このラジオボタンは、デフォルトで選択され、通常の動作を示します。読み取り操作と書き込み操作の両方がリモートサーバーに転送され、結果がクライアントに返されます。リフェラルが定義されていることもありますが、クライアントには返されません。

- 読み取り要求を処理し、書き込み要求にはリフェラルを返す : サーバーは読み取り要求だけを転送し、その結果をクライアントに返します。書き込み要求に対するリフェラルとして返す 1 つ以上の LDAP URL をリストに入力します。
- 読み取りおよび書き込み要求の両方にリフェラルを返す : すべての操作に対するリフェラルとして返す 1 つ以上の LDAP URL をリストに入力します。この場合の動作は、サフィックスへのアクセスを無効にした場合と同様ですが、グローバルのデフォルトリフェラルを使用するのではなく、該当のサフィックス専用にリフェラルを定義してもよい点が異なります。

4. 「追加」ボタンまたは「削除」ボタンを使用して、リフェラルのリストを編集します。「追加」ボタンをクリックすると、新しいリフェラルの LDAP URL を作成するためのダイアログが表示されます。リモートサーバー内の任意の分岐 DN にリフェラルを作成できます。LDAP URL の構造については、『Sun ONE Directory Server Getting Started Guide』を参照してください。

   複数のリフェラルを入力できます。ディレクトリは、クライアントアプリケーションからの要求に対応して、このリスト内のすべてのリフェラルを返します。

5. 「保存」をクリックして、変更内容を適用します。新しいアクセス権とリフェラルの設定が、ただちに有効になります。

コマンド行からのアクセス権とリフェラルの設定

次のコマンドで、*suffixDN* は、定義済みの連鎖サフィックスの完全な文字列です。空白も含まれます。*LDAPURL* は、ターゲットのホスト名、ポート番号、および DN を含む有効な URL です。たとえば、次のように指定します。

```
ldap://alternate.example.com:389/ou=People,dc=example,dc=com
```

1. 次のコマンドを使用して、連鎖サフィックスエントリを編集します。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=suffixDN,cn=mapping tree,cn=config
changetype: modify
replace: nsslapd-state
nsslapd-state: referral on update or referral
-
add: nsslapd-referral
nsslapd-referral: LDAPURL
^D
```

   最後の変更文を繰り返すことにより、任意の数の LDAP URL を `nsslapd-referral` 属性に追加できます。

nsslapd-state が referral on update のときには、サフィックスは読み取り専用になり、書き込み操作に対してはすべての LDAP URL がリフェラルとして返されます。この値が referral のときには、読み取り操作と書き込み操作の両方が拒否され、すべての要求に対してリフェラルが返されます。

2. コマンドの実行が成功すると、サフィックスはただちに読み取り専用になるか、アクセスできなくなり、リフェラルを返すことができるようになります。

連鎖パラメータの変更

連鎖サフィックスを定義したあとには、連鎖を制御するパラメータを変更できます。リモートサーバーへのアクセス方法、プロキシに使用する DN の変更方法、あるいはリモートサーバーの変更方法も指定できます。また、サーバーが連鎖サーバーとの接続を確立する方法や維持する方法を制御するパフォーマンスパラメータも変更できます。

コンソールからの連鎖パラメータの変更

1. Directory Server コンソールの最上位の「設定」タブで「データ」ノードを展開し、変更する連鎖サフィックスを選択します。
2. 右側のパネルで、「リモートサーバー」タブを選択します。
3. リモートサーバーの名前とポートを変更するには、「リモートサーバー URL」フィールドを変更します。URL には、次の書式で、1 つ以上のリモートサーバーのホスト名と省略可能なポート番号が指定されています。

   ldap[s]://*hostname*[:*port*][*hostname*[:*port*]]...//

   URL にサフィックス情報は含まれません。セキュリティ保護されたポートの指定方法については、114 ページの「SSL を使用した連鎖」を参照してください。
   URL で 1 番目に指定されたサーバーが、連鎖された要求への応答に失敗すると、各サーバーには指定された順序で接続されます。LDAP URL 内のすべてのリモートサーバーには、連鎖サフィックスのベースエントリである *suffixDN* を指定する必要があります。
4. プロキシユーザーの DN を変更するために、「バインド DN」フィールドに新しい値を入力します。「パスワード」フィールドに、この DN のパスワードを入力し、確認します。

   proxyDN には、リモートサーバー上のユーザーの DN を指定します。ローカルサーバーは、リモートサーバー上のサフィックスの内容にアクセスするときに、この DN をプロキシとして使用します。連鎖サフィックスを通じて実行される操作は、creatorsName 属性と modifiersName 属性で、このプロキシ ID を使用します。プロキシ DN を指定しない場合、ローカルサーバーはリモートサーバーへのアクセス時に匿名でバインドします。

5. タブの最下部にあるテキストボックスには、このサフィックスの連鎖を許可するために必要な ACI が表示されます。リモートサーバー URL を変更した場合は、新しいリモートサーバーまたは新しいサーバーの *suffixDN* とともに ACI をエントリに追加する必要があります。プロキシ DN を変更した場合は、すべての連鎖サーバー上の ACI を更新する必要があります。「ACI をコピー」ボタンを使用して、ACI のテキストをペーストできるように、システムのクリップボードにコピーします。
6. 「制限と制御」タブを選択して、連鎖される要求に関するパラメータを設定します。カスケード型連鎖パラメータについては、129 ページの「カスケード型連鎖の設定」を参照してください。
7. 「クライアント応答の制御」パラメータを設定して、連鎖される操作のサイズと時間を制限します。
   - 「範囲検索でリフェラルを返す」: 連鎖サフィックス内全体を適用範囲とする検索は、結果が 2 回送信されるため役に立ちません。デフォルトでは、その代わりに、サーバーは連鎖サーバーにリフェラルを返すため、クライアントは連鎖サーバー上で直接検索を実行します。このオプションの選択を解除する場合は、次のパラメータを設定して、連鎖される結果のサイズを制限する必要があります。
   - 「サイズ制限」または「制限サイズなし」: このパラメータは、連鎖された検索操作への応答として返されるエントリ数を決定します。デフォルトは 2000 エントリです。連鎖サフィックスを含む広範囲の検索を制限する場合は、このパラメータの値を小さくします。どのような場合でも、操作はリモートサーバー上のサイズ設定によって制限を受けます。
   - 「時間制限」または「制限時間なし」: このパラメータは、連鎖操作可能な時間の長さを制御します。デフォルトの制限時間は 3600 秒 (1 時間) です。連鎖サフィックスに対する操作に有効な時間を制限する場合は、このパラメータの値を小さくします。どのような場合でも、操作はリモートサーバー上の時間設定によって制限を受けます。
8. 「接続管理」パラメータを設定して、サーバーがリモートサーバーとのネットワーク接続とバインドを管理する方法を調整します。
   - 「最大 LDAP 接続数」: 連鎖サフィックスがリモートサーバーとの間で同時に確立できる LDAP 接続の最大数を指定します。デフォルトは 10 です。
   - 「最大 TCP 接続数」: 連鎖サフィックスがリモートサーバーとの間で同時に確立できる TCP 接続の最大数を指定します。デフォルトは 3 です。
   - 「接続ごとの最大バインド数」: LDAP 接続ごとに同時に行うことができるバインド操作の最大数を指定します。デフォルト値は、接続ごとに 10 です。
   - 「最大バインド再試行数」: エラー時に連鎖サフィックスがリモートサーバーとの再バインドを試行する回数を指定します。「0」を指定すると、連鎖サフィックスは 1 回だけバインドを試みます。デフォルトは 3 です。

- 「接続ごとの最大操作数」: LDAP 接続ごとに同時に行うことができる操作の最大数を指定します。デフォルト値は 1 接続あたり 10 です。
- 「バインドタイムアウト」または「バインドタイムアウトなし」: 連鎖サフィックスへのバインド操作の試行がタイムアウトになるまでの時間 (秒) を指定します。デフォルトは 15 秒です。
- 「中断までのタイムアウト時間」または「中断前タイムアウトなし」: 操作が中断されているかどうかをサーバーがチェックするまでの秒数を指定します。デフォルトは 2 秒です。
- 「接続継続時間」または「存続時間無制限」: 連鎖サフィックスとリモートサーバー間で確立された接続を再利用するための接続の持続時間を指定します。接続したままにすると、処理は速くなりますが、リソースが多く使用されます。たとえば、ダイヤルアップ接続を使用している場合は、接続時間を制限する必要があります。デフォルトでは、接続時間は無制限です。

コンソールでは、エラー検出パラメータを使用できません。詳細は、125 ページの「コマンド行からの連鎖パラメータの変更」を参照してください。

9. 連鎖パラメータの設定が終了したら、「保存」をクリックします。

コマンド行からの連鎖パラメータの変更

コマンド行からは、コンソールを使用した場合と同じパラメータをすべて設定できます。また、106 ページの「エラー検出パラメータ」で説明される追加のパラメータも設定できます。

1. 次のコマンドを使用して、変更するサフィックスに対応する連鎖設定エントリを編集します。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=databaseName,cn=chaining database,cn=plugins,cn=config
changetype: modify
replace: attributeName
attributeName: attributeValue
-
changetype: modify
replace: attributeName
attributeName: attributeValue
...
^D
```

設定できる属性の名前と値は、次の手順で説明します。コマンドの中に複数の変更文を含めることによって、一度に複数のパラメータを変更できます。

2. `nsfarmserverURL` 属性を変更して、リモートサーバーの名前またはポートを変更します。値は URL で、次の書式で、1 つ以上のリモートサーバーのホスト名と省略可能なポート番号が含まれています。

```
ldap[s]://hostname[:port][ hostname[:port]].../
```

URL にサフィックス情報は含まれません。セキュリティ保護されたポートの指定方法については、114 ページの「SSL を使用した連鎖」を参照してください。
URL で 1 番目に指定されたサーバーが、連鎖された要求への応答に失敗すると、各サーバーには指定された順序で接続されます。LDAP URL 内のすべてのリモートサーバーには、連鎖サフィックスのベースエントリである *suffixDN* を指定する必要があります。

3. `nsmultiplexorBindDN` 属性と `nsmultiplexorCredentials` 属性を変更して、リモートサーバーへのプロキシアクセスに使用する DN を変更します。

   ローカルサーバーは、リモートサーバー上のサフィックスの内容にアクセスするときに、この DN をプロキシとして使用します。連鎖サフィックスを通じて実行される操作は、`creatorsName` 属性と `modifiersName` 属性で、このプロキシ ID を使用します。プロキシ DN を指定しない場合、ローカルサーバーはリモートサーバーへのアクセス時に匿名でバインドします。

4. プロキシ DN またはその資格を変更する場合は、リモートサーバー上に対応する ACI を作成する必要があります。この ACI は、プロキシで許可された連鎖操作を行うために必要です。

   ```
   ldapmodify -h host2 -p port2 -D "cn=Directory Manager" -w password2
   dn: suffixDN
   changetype: modify
   add: aci
   aci: (targetattr=*)(target = "ldap:///suffixDN")(version 3.0;acl
    "Allows use of admin for chaining"; allow (proxy)
    (userdn="ldap:///proxyDN");)
   ^D
   ```

5. 104 ページの「デフォルト連鎖パラメータの設定」で説明される属性を設定して、リモートサーバーへの接続とリモートサーバー上の操作を制御します。カスケード型パラメータについては、129 ページの「カスケード型連鎖の設定」を参照してください。

スレッド使用の最適化

サーバーがグローバルに使用するスレッド数を設定して、連鎖に使用されるスレッドリソースを考慮することもできます。連鎖操作は、リモートサーバーへ転送される必要があるため、かなり長く時間がかかることがあります。ただし、リモートサーバーが操作を処理している間は、スレッドはアイドル状態になります。連鎖されたサーバーを使用することで長時間の遅延が発生する場合は、スレッド数を増加させて、その間より多くのスレッドがローカルの操作を処理できるようにする必要があります。

デフォルトでは、サーバーが使用できるスレッド数は 30 です。ただし、連鎖サフィックスを使用する場合は、操作の処理で使用可能なスレッドの数を増やすことによって、パフォーマンスを向上させることができます。必要なスレッド数は、連鎖サフィックスの数、連鎖サフィックス上の操作の数とタイプ、およびリモートサーバー上で操作を処理するために必要な平均時間によって変わります。

通常、連鎖サフィックスごとに 5 から 10 のスレッド数を増加させる必要があります。これは、連鎖サフィックスで、ローカルサフィックスと同じ数の操作があることを仮定しています。

コンソールからのスレッドリソースの設定

1. Directory Server コンソールの最上位の「設定」タブで「パフォーマンス」ノードをクリックし、右側のパネルで「その他」タブを選択します。
2. 「スレッドの最大数」フィールドに新しい値を入力します。
3. 「了解」をクリックして変更を保存し、変更内容を有効にするためにサーバーを再起動する必要があるというメッセージを了承します。
4. 新しい数のスレッドを使用するために、Directory Server を再起動します。

コマンド行からのスレッドリソースの設定

1. 次のコマンドを使用して、グローバル設定エントリを編集し、スレッド数を変更します。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=config
changetype: modify
replace: nsslapd-threadnumber
nsslapd-threadnumber: newThreadNumber
^D
```

2. Directory Server を再起動して、変更した数のスレッドを使用できるようにします。

連鎖サフィックスの削除

連鎖サフィックスを削除すると、ローカルのディレクトリツリーからアクセスできなくなりますが、連鎖サーバー上のエントリやサフィックスは削除されません。親サフィックスを削除し、そのサブサフィックスを、ディレクトリで新しいルートサフィックスとして保持することもできます。

コンソールからの連鎖サフィックスの削除

1. Directory Server コンソールの「設定」タブで、「データ」ノードを展開します。
2. 削除するサフィックスを右クリックし、ポップアップメニューから「削除」を選択します。

   または、サフィックスのノードを選択し、「オブジェクト」メニューから「削除」を選択します。
3. 該当の連鎖サフィックスを通じてアクセスできるエントリは、リモートディレクトリから削除されないことを示す確認のダイアログが表示されます。

   親サフィックスに加えて、すべてのサブサフィックスを再帰的に削除することもできます。分岐全体を削除する場合は、「このサフィックスとそのサブサフィックスすべてを削除」を選択します。また、特定のサフィックスだけを削除し、そのサブサフィックスをディレクトリに残しておく場合は、「このサフィックスだけを削除する」を選択します。
4. 「了解」をクリックして、サフィックスを削除します。

   コンソールによって処理されている内容を示すダイアログボックスが表示されます。

コマンド行からのサフィックスの削除

コマンド行からサフィックスを削除するには、`ldapdelete` コマンドを使用して、ディレクトリからサフィックスの設定エントリを削除します。

サブサフィックスも含めて分岐全体を削除する場合は、削除する親サフィックスのサブサフィックスを見つけ出し、サブサフィックスとさらにそれらのサブサフィックスのすべてについて手順を繰り返す必要があります。

1. 次のコマンドを使用して、サフィックスの設定エントリを削除します。

   ```
   ldapdelete -h host -p port -D "cn=Directory Manager" -w password
   cn=suffixDN,cn=mapping tree,cn=config
   ```

   このコマンドによって、連鎖サフィックスとそのリモートエントリがディレクトリに表示されなくなります。
2. `cn=`*databaseName*`,cn=chaining database,cn=plugins,cn=config` にある対応するデータベース設定エントリと、その下位にある監視エントリを削除します。

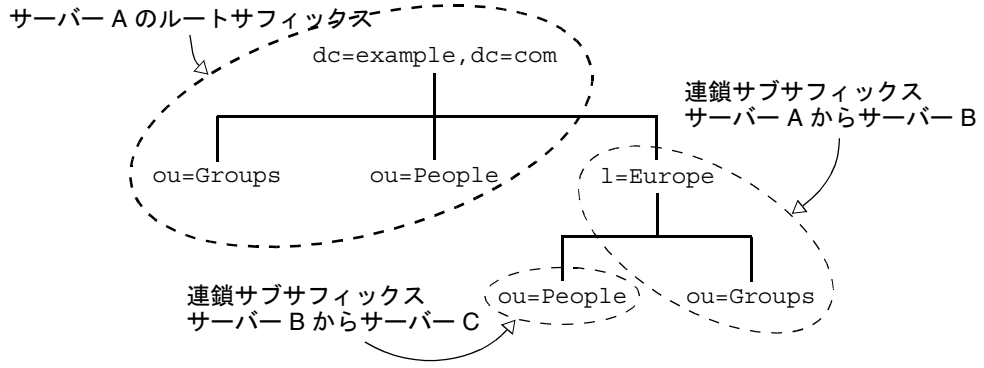
```
ldapdelete -h host -p port -D "cn=Directory Manager" -w password
cn=monitor,cn=dbName,cn=chaining database,cn=plugins,cn=config
cn=dbName,cn=chaining database,cn=plugins,cn=config
```

カスケード型連鎖の設定

カスケード型連鎖では、あるサーバーから連鎖されているサブツリーは、それ自身が連鎖サフィックスであることも、連鎖サブサフィックスを含むこともできます。サーバーの連鎖サフィックスが含まれる操作は、第 3 のサーバーなどに接続する中間サーバーに転送されます。ディレクトリツリーの全データにアクセスするために、サーバー間の複数のホップが必要な場合、カスケード型連鎖が発生します。

たとえば、エントリ ou=People,l=Europe,dc=example,dc=com へのアクセスがサーバー A から サーバー B に、そして最終的にサーバー C に連鎖される様子を、次の図に示します。サーバー A には、ルートサフィックス dc=example,dc=com と分岐 l=Europe,dc=example,dc=com に対するサーバー B への連鎖サブサフィックスがあります。サーバー B には、エントリ l=Europe,dc=example,dc=com がありますが、分岐 ou=People,l=Europe,dc=example,dc=com はサーバー C への連鎖サブサフィックスです。サーバー C には、実際にエントリ ou=People,l=Europe,dc=example,dc=com があります。

図 3-3　3 台のサーバーのカスケード型連鎖



カスケード型パラメータの設定

カスケード型の設定には、次の 2 つの連鎖パラメータがあります。

- すべてのサーバーでは、連鎖トポロジ内の異常なループをすべて検出できるように、ループ検出を設定する必要がある。ループ検出が無効になっていると、ループ内のサーバーは操作の転送を繰り返し、過度の負荷がかかる
- すべての中間連鎖サフィックスでローカル ACI を評価するように設定する必要がある。通常、ローカル ACI の評価は、連鎖サフィックスの第 1 レベルでは行われない

コンソールからのカスケード型パラメータの設定

1. Directory Server コンソールの最上位の「設定」タブで「データ」ノードを展開し、変更する連鎖サフィックスを選択します。
2. 右側のパネルで、カスケード型連鎖パラメータを変更できる「制限と制御」タブを選択します。
3. カスケード型連鎖内のすべての中間サーバーについて、ローカル ACI のチェックを行うために、チェックボックスを選択します。

   ユーザーのアクセス権は最初のサーバーでは評価されず、プロキシを通じた 2 番目のサーバーで評価されるため、単一レベルの連鎖では、このチェックボックスは選択されていません。ただし、カスケード型連鎖の中間サーバーでは、ACI のチェックを有効にして、操作が再度転送される前にアクセス制御の実行を許可する必要があります。
4. カスケード型連鎖内のすべてのサーバーについて、トポロジ内のすべての連鎖操作を許可する最大ホップ数を設定します。操作が別の連鎖サフィックスに転送されるたびにホップとしてカウントされ、この制限に到達していれば、連鎖サフィックスは操作をそれ以上転送しません。

   この数値には、最長のカスケード型連鎖内のホップ数よりも大きい値を設定する必要があります。制限に達した場合は、サーバーがトポロジ内で異常なループがあると判断するため、操作は中断されます。

   また、ループ検出制御を許可するように連鎖を設定する必要があります。131 ページの「カスケード型連鎖の LDAP 制御の送信」を参照してください。
5. カスケード型パラメータの設定が終了したら、「保存」をクリックします。

コマンド行からのカスケード型パラメータの設定

1. 次のコマンドを使用して、すべての中間サーバーで、カスケード型サフィックスに関する連鎖設定エントリを編集します。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=databaseName,cn=chaining database,cn=plugins,cn=config
changetype: modify
replace: nsCheckLocalACI
nsCheckLocalACI: on
-
changetype: modify
replace: nsHopLimit
nsHopLimit: maximumHops
^D
```

maximumHops には、最長のカスケード型連鎖内のホップ数よりも大きい値を設定する必要があります。制限に達した場合は、サーバーがトポロジ内で異常なループがあると判断するため、操作は中断されます。また、ループ検出制御を許可するように連鎖を設定する必要があります。131 ページの「カスケード型連鎖の LDAP 制御の送信」を参照してください。

2. カスケード型連鎖内のその他すべてのサーバーについて、次のコマンドを使用して、カスケード型サフィックスに関する連鎖設定エントリを編集します。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=databaseName,cn=chaining database,cn=plugins,cn=config
changetype: modify
replace: nsHopLimit
nsHopLimit: maximumHops
^D
```

maximumHops の定義は、前の手順と同じです。

カスケード型連鎖の LDAP 制御の送信

デフォルトでは、連鎖サフィックスはプロキシ承認制御を送信しません。ただし、連鎖サフィックスが別の連鎖サフィックスに接続するときには、リモートサーバー上でのアクセス制御に必要なユーザー識別情報を送信するために、プロキシ承認制御が必要となります。中間連鎖サフィックスは、連鎖のためにこの制御を許可する必要があります。

最近、プロキシ承認制御に関して、2 番目のプロトコルが定義されました。さまざまなサーバーのバージョンでは、制御のどれかが使用される可能性があるため、すべてのカスケード型サーバーでは、連鎖に新旧両方のプロキシ承認制御を許可するように設定する必要があります。

また、カスケード型連鎖中のループを防ぐために、ループ検出制御が必要となります。デフォルトでは、連鎖操作とともに転送されるよう許可されますが、この設定を検証する必要があります。サーバーでループ検出制御の連鎖が許可されていない場合、そのサーバーを含むループはすべて検出されません。

115 ページの「連鎖ポリシーの設定」の手順に従って、次の 3 つの制御の連鎖を許可します。

- 2.16.840.1.113730.3.4.12 : プロキシ承認制御 (旧仕様)
- 2.16.840.1.113730.3.4.18 : プロキシ承認制御 (新仕様)
- 1.3.6.1.4.1.1466.29539.12 - ループ検出制御

第 4 章

ディレクトリへのデータの実装

Directory Server で管理されるデータは、まとめてインポートされることがよくあります。Directory Server には、サフィックス全体のインポートとエクスポートを行うツールが用意されています。また、一度にすべてのサフィックスのバックアップを作成したり、すべてのデータをバックアップから復元したりするツールも用意されています。

この章では、次のディレクトリへのデータの実装手順について説明します。

- サフィックスの読み取り専用モードの設定
- データのインポート
- データのエクスポート
- データのバックアップ
- バックアップからのデータの復元

サフィックスの読み取り専用モードの設定

Directory Server 上でエクスポート操作またはバックアップ操作を実行する前に、特定のサフィックスに対して読み取り専用モードを有効にすると、その時点でのサフィックスの内容の正確なイメージを確保できます。また、インポート操作または復元操作を実行する前に、対象となるサフィックスが読み取り専用モードになっていないことを確認する必要があります。

Directory Server コンソールとコマンド行ユーティリティでは、エクスポート操作またはバックアップ操作の前に、ディレクトリが自動的に読み取り専用モードに設定されることはありません。これは、読み取り専用にしてしまうと、ディレクトリの更新ができなくなるためです。ただし、マルチマスター環境では、1 つのサーバーで読み取り専用モードを有効にし、ほかのマスターはデータの書き込みを可能にしておくことができます。

サフィックスを読み取り専用にするには、98 ページの「アクセス権とリフェラルの設定」に記載されている手順を実行します。または、Directory Server 全体を書込み禁止にすることもできます。この手順については、36 ページの「グローバルな読み取り専用モードの設定」を参照してください。

データのインポート

Sun ONE Directory Server では、次の 2 つの方法でデータをインポートできます。

- LDIF ファイルをインポートする方法では、ディレクトリ内の任意のサフィックスについて、そのエントリをまとめて追加、変更、削除できます。
- LDIF ファイルを使用してサフィックスを初期化する方法では、サフィックスに現在含まれているデータは削除され、LDIF ファイルの内容で置き換えられます。

どちらの方法も、Directory Server コンソールまたはコマンド行ユーティリティを使用して実行できます。

注 インポートする LDIF ファイルでは、UTF-8 文字セットエンコードが使用されている必要があります。

LDIF をインポートするときは、ディレクトリ内に親エントリが存在するか、ファイルから親エントリを最初にコピーする必要があります。サフィックスを初期化するときは、ルートエントリと、対応するサフィックスのすべてのディレクトリツリーノードが LDIF ファイルに含まれている必要があります。

次の表は、インポートと初期化の違いを示しています。

表 4-1 データのインポートとサフィックスの初期化の比較

比較ドメイン	データのインポート	サフィックスの初期化
内容の上書き	不可	可
LDAP 処理	追加、変更、削除	追加のみ
パフォーマンス	低速	高速
サーバーの障害への対応	ベストエフォート (障害発生時までの変更内容はそのまま残る)	不可 (障害が発生するとすべての変更内容は失われる)
LDIF ファイルの位置	コンソールマシン上	コンソールまたはサーバーと同じマシン上

表 4-1　データのインポートとサフィックスの初期化の比較 (続き)

比較ドメイン	データのインポート	サフィックスの初期化
設定情報のインポート (cn=config)	可	不可

LDIF ファイルのインポート

インポート操作を実行するとき、Directory Server コンソールでは、新しいエントリをディレクトリに追加するために ldapmodify 処理が実行されます。エントリは LDIF ファイルで指定されます。このファイルには、インポート処理の一部として既存のエントリの変更や削除を行う更新文が含まれている場合もあります。

エントリのインポート先は、Directory Server で管理されている任意のサフィックスか、設定に定義されている任意の連鎖サフィックスまたは連鎖サブサフィックスです。エントリを追加するほかの処理と同様に、インポートされた新しいエントリすべてにインデックスが付けられます。

コンソールからの LDIF のインポート

インポートを実行するには、Directory Manager または管理者としてログインする必要があります。

1. Directory Server コンソールの最上位にある「タスク」タブで一番下までスクロールし、「LDIF からインポート」の隣にあるボタンをクリックします。

   「LDIF のインポート」ダイアログボックスが表示されます。

2. 「LDIF のインポート」ダイアログボックスで、インポートする LDIF ファイルの絶対パスを「LDIF ファイル」フィールドに入力するか、「参照」をクリックしてローカルファイルシステムからファイルを選択します。

   リモートマシン上のディレクトリにアクセスしている場合、このフィールド名は「LDIF ファイル (コンソールマシン上)」と表示されます。このラベルは、リモートの Directory Server マシンのファイルシステムではなく、ローカルファイルシステムを参照していることを示します。

3. 必要に応じて、次のオプションを設定します。

   a. 追加のみ : LDIF ファイルには、デフォルトの追加命令のほかに、変更命令や削除命令が含まれている場合もあります。LDIF ファイル内の追加命令だけを実行し、そのほかの命令はすべて無視するように指定するには、このチェックボックスを選択します。

b. エラー時に継続 : エラーが発生してもインポートを続行するように指定するには、このチェックボックスを選択します。たとえば、すでにサフィックス上に存在するエントリを含む LDIF ファイルをインポートする場合に、このオプションを使用できます。インポート処理の実行中、既存エントリなどのエラーが拒否エントリ用ファイルに記録されます。

このチェックボックスが選択されていない場合、最初のエラー発生後にインポート処理は停止します。それ以前の LDIF ファイルのエントリはすべて正常にインポートされているので、ディレクトリ内に残ります。

4. 「拒否エントリ用ファイル」フィールドには、コンソールがインポートできなかったすべてのエントリを記録するファイルの絶対パスを入力します。あるいは、「参照」をクリックして、ローカルファイルシステムからファイルを選択します。

たとえば、サーバーはディレクトリにすでに存在するエントリや、親オブジェクトのないエントリをインポートできません。コンソールは、サーバーから送られたエラーメッセージを拒否ファイルに書き込みます。

このフィールドに何も書き込まれないと、拒否されたエントリは記録されません。

5. 「了解」をクリックして、インポート処理を開始します。

Directory Server コンソールでは、処理の状態と発生したエラーを示すダイアログボックスが表示されます。「拒否エントリ用ファイル」フィールドに入力した場合は、そこに指定されたファイルにもすべてのエラーメッセージが書き込まれます。

コマンド行からの LDIF のインポート

LDAP から LDIF ファイルをインポートし、そこに含まれるすべての処理を実行するには、`ldif2ldap` コマンド (Solaris パッケージ内の `directoryserver ldif2ldap`) を使用します。このスクリプトを使用すると、すべてのディレクトリサフィックスに対して同時にデータをインポートできます。`ldif2ldap` を使用してインポートを実行するには、サーバーを動作させておく必要があります。

このコマンドの完全パスは次のとおりです。

Solaris パッケージ
```
# /usr/sbin/directoryserver ldif2ldap
```
その他のインストール
```
# ServerRoot/slapd-serverID/ldif2ldap
```

次の例では、`ldif2ldap` コマンドを使ってインポートが実行されます。このコマンドを実行するために root 権限は必要ありませんが、コマンド行で Directory Manager に資格を付与する必要があります。最後のパラメータには、インポートする 1 つ以上の LDIF ファイル名を指定します。

UNIX シェルスクリプト

```
# use directoryserver ldif2ldap on Solaris パッケージ installations
/var/Sun/mps/slapd-example/ldif2ldap ¥
  "cn=Directory Manager" password ¥
  /var/Sun/mps/slapd-example//ldif/demo.ldif
```

Windows バッチファイル

```
C:¥Program Files¥Sun¥MPS¥slapd-example¥ldif2ldap.bat
  "cn=Directory Manager" password
  C:¥Program Files¥Sun¥MPS¥slapd-example¥ldif¥demo.ldif
```

このスクリプトの使用方法の詳細は、『Sun ONE Directory Server Reference Manual』の第 2 章にある「ldif2ldap」を参照してください。

サフィックスの初期化

サフィックスを初期化すると、サフィックスに含まれている既存のデータが、追加するエントリだけを含む LDIF ファイルの内容によって上書きされます。

警告 LDIF ファイルからサフィックスを初期化するときは、データを復元する場合を除いて、o=NetscapeRoot サフィックスを上書きしないように注意してください。このサフィックスを上書きしてしまうと、重要な情報が削除されてしまい、すべての Sun ONE サーバーをインストールし直す必要があります。

サフィックスを初期化するユーザーは、Directory Manager または管理者としての認証を受けている必要があります。セキュリティ上の理由から、サフィックスのルートエントリ (たとえば、dc=example,dc=com) にアクセスできるのは、Directory Manager および管理者だけに限定されます。このため、ルートエントリを含む LDIF ファイルをインポートできるのは、これらの ID でログインしたユーザーだけに限定されます。

コンソールからのサフィックスの初期化

1. Directory Server コンソールの最上位の「設定」タブで「データ」ノードを展開し、初期化するサフィックスを表示します。
2. このサフィックスノードをマウスの右ボタンでクリックし、ポップアップメニューから「初期化」を選択します。または、サフィックスノードを選択し、「オブジェクト」メニューから「初期化」を選択します。

   「サフィックスを初期化」ダイアログが表示されます。
3. 初期化に使用する LDIF ファイルの絶対パスを「LDIF ファイル」フィールドに入力するか、「参照」をクリックしてマシン上のファイルを選択します。

4. インポートするファイルが置かれているローカルマシンからコンソールを操作している場合は、手順 6 に進みます。LDIF ファイルがあるサーバーのリモートマシンからコンソールを操作している場合は、次のどちらかのオプションを選択します。

   ローカルマシンから : LDIF ファイルがローカルマシン上に置かれていることを示します。

   サーバーマシンから : LDIF ファイルがリモートサーバー上にあることを示します。デフォルトでは、コンソールは次のディレクトリ内でファイルを検索します。

   ServerRoot/slapd-*serverID*/ldif

5. 「了解」をクリックします。

警告 スクリプトは、サフィックスのデータを上書きします。

6. サフィックスに含まれるデータを上書きしてよいか確認します。

   サフィックスの初期化が開始され、エラーが発生した場合はダイアログに表示されます。

ldif2db コマンドによるサフィックスの初期化

サフィックスを初期化し、既存のデータを上書きするには、ldif2db コマンド (Solaris パッケージ内の directoryserver ldif2db) を使用します。このスクリプトでは、インポートを開始する前に、サーバーを停止する必要があります。

デフォルトでは、まず既存の o=NetscapeRoot 設定情報すべてが保存され、インポートされるファイル内の o=NetscapeRoot 設定情報と結合されます。

警告 スクリプトは、サフィックスのデータを上書きします。

サーバーを停止して LDIF をインポートするには、次の手順を実行します。

1. root としてコマンド行に次のコマンドを入力し、サーバーを停止させます。

Solaris パッケージ
```
# /usr/sbin/directoryserver stop
```
その他のインストール
```
# ServerRoot/slapd-serverID/stop-slapd
```

2. 次の場所にあるコマンドを実行します。

Solaris パッケージ
```
# /usr/sbin/directoryserver ldif2db
```
その他のインストール
```
# ServerRoot/slapd-serverID/ldif2db
```

3. 適切なコマンドでサーバーを起動します。

Solaris パッケージ
```
# /usr/sbin/directoryserver start
```
その他のインストール
```
# ServerRoot/slapd-serverID/start-slapd
```

次の例では、ldif2db コマンドを使用して、LDIF ファイルを 1 つのサフィックスにインポートします。

UNIX シェルスクリプト

```
# use directoryserver ldif2db on Solaris パッケージ installations
/var/Sun/mps/slapd-example/ldif2db -n Database1 ¥
 -i /var/Sun/mps/slapd-example/ldif/demo.ldif ¥
 -i /var/Sun/mps/slapd-example/ldif/demo2.ldif
```

Windows バッチファイル

```
C:¥Program Files¥Sun¥MPS¥slapd-example¥ldif2db.bat -n Database1
  -i C:¥Program Files¥Sun¥MPS¥slapd-example¥ldif¥demo.ldif
  -i C:¥Program Files¥Sun¥MPS¥slapd-example¥ldif¥demo2.ldif
```

表 4-2 例で使用した ldif2db オプションの説明

オプション	内容
-n	データのインポート先となるデータベースの名前を指定する 注意 : -n オプションで、LDIF ファイルに含まれるサフィックスに対応しないデータベースを指定した場合は、データベースに含まれるすべてのデータが削除され、インポートは失敗します。データベース名を間違えないように注意してください。
-i	インポートする LDIF ファイルの絶対パス名を指定する。このオプションは必須。一度に複数の LDIF ファイルをインポートする場合は、複数の -i 引数を指定できる。複数のファイルをインポートする場合、サーバーはコマンド行で指定された順に LDIF ファイルをインポートする

このコマンドの使用方法の詳細は、『Sun ONE Directory Server Reference Manual』の第 2 章にある「ldif2ldap」を参照してください。

ldif2db.pl Perl スクリプトによるサフィックスの初期化

ldif2db コマンドと同様に ldif2db.pl スクリプト (Solaris パッケージ内の directoryserver ldif2db-task) は、指定したサフィックスのデータを上書きします。このスクリプトを使用してインポートを実行する場合は、サーバーを動作させておく必要があります。

警告　スクリプトは、サフィックスのデータを上書きします。

このスクリプトのコマンドは、プラットフォームごとに異なります。

Solaris パッケージ
```
# /usr/sbin/directoryserver ldif2db-task
```
Windows プラットフォーム
```
cd ServerRoot
bin¥slapd¥admin¥bin¥perl slapd-serverID¥ldif2db.pl
```
その他のインストール
```
# ServerRoot/slapd-serverID/ldif2db.pl
```

次の例では、ldif2db.pl スクリプトを使用して、LDIF ファイルをインポートします。このスクリプトの実行には、root 権限は必要ありませんが、Directory Manager として認証する必要があります。

UNIX シェルスクリプト

```
# use directoryserver ldif2db-task on Solaris パッケージ installations
/var/Sun/mps/slapd-example/ldif2db.pl ¥
  -D "cn=Directory Manager" -w password -n Database1 ¥
  -i /var/Sun/mps/slapd-example/ldif/demo.ldif
```

Windows バッチファイル

```
C:¥Program Files¥Sun¥MPS¥bin¥slapd¥admin¥bin¥perl.exe
  C:¥Program Files¥Sun¥MPS¥slapd-example¥ldif2db.pl
    -D "cn=Directory Manager" -w password -n Database1
    -i C:¥Program Files¥Sun¥MPS¥slapd-example¥ldif¥demo.ldif
```

次の表に、これらの例で使用されている ldif2db.pl のオプションを示します。

表 4-3　例で使用した ldif2db.pl オプションの説明

オプション	内容
-D	Directory Manager の DN を指定する
-w	Directory Manager のパスワードを指定する
-n	データのインポート先となるデータベースの名前を指定する
-i	インポートする LDIF ファイルの絶対パス名を指定する。このオプションは必須。一度に複数の LDIF ファイルをインポートする場合は、複数の -i 引数を指定できる。複数のファイルをインポートする場合、サーバーはコマンド行で指定された順に LDIF ファイルをインポートする

この Perl スクリプトの使用方法の詳細は、『Sun ONE Directory Server Reference Manual』の第 2 章にある「ldif2db.pl」を参照してください。

データのエクスポート

プレーンテキスト形式の LDIF (LDAP Data Interchange Format) を使用して、ディレクトリの内容をエクスポートすることができます。LDIF は、エントリ、属性、およびその値をテキストで表現します。LDIF は、RFC 2849 (`http://www.ietf.org/rfc/rfc2849.txt`) に定義されている標準形式です。

データのエクスポートは、次のような場合に便利です。

- サーバー上のデータのバックアップ
- 他の Directory Server へのデータのコピー
- 他のアプリケーションへのデータのエクスポート
- ディレクトリトポロジ変更後のサフィックスの再生成

エクスポート処理を実行しても、設定情報 (`cn=config`) はエクスポートされません。

警告 エクスポートの処理中には、サーバーを停止しないでください。

コンソールを使用した LDIF へのディレクトリ全体のエクスポート

エクスポートされるファイルの最終的な位置に応じて、ディレクトリデータの一部またはすべてを LDIF にエクスポートできます。LDIF ファイルがサーバー上にある場合は、サーバー上のローカルサフィックスにあるデータしかエクスポートできません。LDIF ファイルがリモートマシン上にある場合は、すべてのサフィックスと連鎖サフィックスをエクスポートできます。

サーバーの動作中に、Directory Server コンソールから LDIF にディレクトリデータをエクスポートするには、次の手順を実行します。

1. Directory Server コンソールの最上位にある「タスク」タブで一番下までスクロールし、「LDIF にエクスポート」の隣にあるボタンをクリックします。

   「エクスポート」ダイアログが表示されます。

2. 「LDIF ファイル」フィールドに LDIF ファイルの絶対パスとファイル名を入力するか、「参照」をクリックしてファイルを選択します。

   リモートサーバー上でコンソールを実行している場合は、「参照」は無効になっています。「参照」ボタンが無効になっている場合は、ファイルがデフォルトで次のディレクトリに格納されています。

   ServerRoot`/slapd-`*serverID*`/ldif`

3. サーバー以外のリモートマシン上でコンソールを実行している場合は、「LDIF ファイル」フィールドの下に 2 つのラジオボタンが表示されます。コンソールの実行マシン上の LDIF ファイルをエクスポート先として指定する場合は、「ローカルマシンへ」を選択します。サーバーのマシン上に置かれている LDIF ファイルをエクスポート先として指定する場合は、「サーバーマシンへ」を選択します。
4. ディレクトリ全体をエクスポートする場合は、「すべてのサフィックス」ラジオボタンを選択します。

   ディレクトリのサブツリーだけをエクスポートするときは、「サブツリー」ラジオボタンを選択し、テキストボックスにサブツリーのベースとして DN を入力します。

   「参照」をクリックしてサブツリーを選択することもできます。
5. 「了解」をクリックして、ディレクトリの内容をファイルにエクスポートします。

コンソールを使用した LDIF への単一サフィックスのエクスポート

サーバーの動作中に、Directory Server コンソールから LDIF に単一のサフィックスをエクスポートするには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで「データ」ノードを展開し、エクスポートするサフィックスを表示します。
2. このサフィックスノードをマウスの右ボタンでクリックし、ポップアップメニューから「エクスポート」を選択します。または、サフィックスノードを選択し、「オブジェクト」メニューから「エクスポート」を選択します。

   「サフィックスをエクスポート」ダイアログが表示されます。
3. LDIF ファイルの絶対パスを「LDIF ファイル」フィールドに入力するか、「参照」をクリックしてマシン上のファイルを選択します。

   「参照」ボタンが有効でない場合、デフォルトではファイルが次のディレクトリに格納されます。

   ServerRoot/slapd-*serverID*/ldif
4. レプリケートされたサフィックスでは、「レプリケーション情報のエクスポート」チェックボックスを選択することもできます。この機能は、エクスポートした LDIF を使ってこのサフィックスの別のレプリカを初期化する場合にだけ必要です。

5. このサフィックスで属性の暗号化が有効に設定されているときは、「属性の復号化」チェックボックスを選択できます。この場合は、サーバーの証明書データベースを保護しているパスワードを指定する必要があります。オプションを選択してパスワードを入力するか、パスワードを記録したファイルの名前を入力します。属性値を復号化するためのパスワードを指定しない場合、暗号化された値がLDIF に出力されます。
6. 「了解」をクリックして、サフィックスの内容をファイルにエクスポートします。

コマンド行からの LDIF へのエクスポート

ディレクトリのサフィックスまたはサブツリーを LDIF にエクスポートするときは、`db2ldif` コマンド (Solaris パッケージ内の `directoryserver db2ldif`) を使用します。このスクリプトは、サーバーが動作中または停止中に、サフィックスの内容のすべてまたは一部を LDIF ファイルにエクスポートします。

データベースの内容を LDIF ファイルにエクスポートするには、次のコマンドを使用します。

Solaris パッケージ
```
# /usr/sbin/directoryserver db2ldif
```
その他のインストール
```
# ServerRoot/slapd-serverID/db2ldif
```

次の例では、2 つのサフィックスが 1 つの LDIF ファイルにエクスポートされます。

```
db2ldif -a output.ldif ¥
        -s "dc=example,dc=com" -s "o=NetscapeRoot"
```

次の表に、これらの例で使用されている `db2ldif` のオプションを示します。

表 4-4 例で使用した `db2ldif` オプションの説明

オプション	内容
`-a`	サーバーがエクスポートした LDIF を保存する出力ファイル名を定義する。デフォルトでは、このファイルは *ServerRoot*`/slapd-`*serverID* ディレクトリに格納される
`-s`	エクスポートに取り込むサフィックスまたはサブツリーを指定する。複数の `-s` 引数を指定すると、複数のサフィックスまたはサブツリーを指定できる

db2ldif コマンドに -r オプションを指定して、レプリケートされたサフィックスを LDIF ファイルにエクスポートすることもできます。結果として作成される LDIF ファイルには、レプリケーションメカニズムで使用される属性サブタイプが含まれています。あとでこの LDIF ファイルをコンシューマサーバーにインポートして、コンシューマレプリカを初期化できます。この手順については、294 ページの「レプリカの初期化」を参照してください。

db2ldif コマンドに -r オプションを指定して実行する場合、サーバーは停止している必要があります。サーバーをあらかじめ停止し、コマンドの終了後に起動してください。または、db2ldif.pl スクリプトに -r オプションを指定します。この場合、サーバーを停止する必要はありません。

このスクリプトの使用方法の詳細は、『Sun ONE Directory Server Reference Manual』の第 2 章にある「db2lidf」を参照してください。

データのバックアップ

データのバックアップでは、データベースファイルが破損したり削除されたりする場合に備えて、ディレクトリの内容のスナップショットを保存できます。Directory Server コンソールやコマンド行スクリプトを使用して、サフィックスのバックアップを行うことができます。

警告 バックアップの処理中には、サーバーを停止しないでください。

ここで説明するバックアップ手順では、サーバーファイルのコピーがデフォルトで同じホスト上に格納されます。セキュリティ強化のために、このバックアップを別のマシンや別のファイルシステムにコピーして格納してください。

注 これらの方法では、リモートサーバー上にある連鎖サフィックスをバックアップすることはできません。独立したサーバーは個別にバックアップする必要があります。

コンソールを使用したサーバーのバックアップ

Directory Server コンソールを使用してサーバーをバックアップする場合は、サーバーのすべての内容と、関連するインデックスファイルがバックアップ位置にコピーされます。バックアップは、サーバーが動作中でも実行できます。

サーバーコンソールを使用してサーバーをバックアップするには、次の手順を実行します。

1. Directory Server コンソールの最上位にある「タスク」タブで、「Directory Server のバックアップ」の隣にあるボタンをクリックします。

   「Directory Server のバックアップ」ダイアログボックスが表示されます。

2. 「ディレクトリ」テキストボックスに、バックアップの格納先ディレクトリへの絶対パスを入力します。ディレクトリと同じマシン上でコンソールを実行している場合は、「参照」をクリックしてローカルディレクトリを選択します。

   または、「デフォルトの使用」をクリックして、バックアップを次のディレクトリに格納します。

   ServerRoot/slapd-*serverID*/bak/*YYYY_MM_DD_hh_mm_ss*

   ここで、*serverID* はディレクトリサーバーの名前です。ディレクトリ名は、バックアップが作成された日時を表わす形式で生成されます。

3. 「了解」をクリックすると、バックアップが作成されます。

コマンド行からのサーバーのバックアップ

db2bak コマンド (Solaris パッケージ内の directoryserver db2bak) を使用すると、コマンド行からサーバーをバックアップできます。このスクリプトは、サーバーが動作中か動作中でないかにかかわらず実行できます。

ただし、この方法では設定情報をバックアップできません。設定情報のバックアップについては、146 ページの「dse.ldif 設定ファイルのバックアップ」を参照してください。

ディレクトリをバックアップするには、次のコマンドを使用します。

Solaris パッケージ
```
# /usr/sbin/directoryserver db2bak backupDir
```
その他のインストール
```
# ServerRoot/slapd-serverID/db2bak backupDir
```

backupDir パラメータには、バックアップを格納するディレクトリを指定します。デフォルトでは、バックアップディレクトリ名は、現在の日付 YYYY_MM_DD_hh_mm_ss で生成されます。このスクリプトの使用方法の詳細は、『Sun ONE Directory Server Reference Manual』の第 2 章にある「db2bak」を参照してください。

dse.ldif 設定ファイルのバックアップ

Directory Server は、自動的に dse.ldif 設定ファイルをバックアップします。Directory Server を起動すると、dse.ldif ファイルのバックアップが、次のディレクトリの dse.ldif.startOK ファイルに自動的に作成されます。

ServerRoot/slapd-*serverID*/config

cn=config ブランチの内容を変更する場合は、サーバーが dse.ldif ファイルに変更を書き込む前に、ファイルが config ディレクトリの dse.ldif.bak ファイルにバックアップされます。設定を保存する必要がある場合には、どちらかのファイルのコピーを作成してください。

バックアップからのデータの復元

次に、Directory Server コンソールまたはコマンド行を使用して、ディレクトリ内のサフィックスを復元する手順について説明します。144 ページの「データのバックアップ」の手順に従って、サーバーのバックアップが作成されている必要があります。レプリケーションアグリーメントに関係しているサフィックスを復元する場合は、その前に 146 ページの「レプリケートされたサフィックスの復元」をお読みください。

警告 バックアップや復元の処理中には、サーバーを停止しないでください。

サーバーを復元すると、既存のデータベースファイルが上書きされます。したがって、バックアップの作成時以降にデータに加えられた変更内容はすべて失われます。

レプリケートされたサフィックスの復元

サプライヤサーバーとコンシューマサーバーの間でレプリケートされるサフィックスを復元する場合は、特別な注意が必要です。可能な場合は、サフィックスをバックアップから復元するのではなく、レプリケーションメカニズムにより更新するようにしてください。ここでは、レプリカを復元すべき場合とその方法、および復元後にほかのレプリカとの同期を確保する方法について説明します。バックアップと復元を使用してレプリカを初期化する方法については、294 ページの「レプリカの初期化」を参照してください。

シングルマスターモデルでのサプライヤの復元

シングルマスターサプライヤであるサフィックスには、レプリケーショントポロジ全体に対して権限のあるデータが含まれています。したがって、このサフィックスを復元することは、トポロジ全体のすべてのデータを初期化し直すことと同じです。シングルマスターを復元するのは、復元するバックアップの内容ですべてのデータを初期化し直す場合に限定してください。

エラーのためにシングルマスターのデータを復旧できない場合は、コンシューマ上のデータを使用することも検討してください。これは、バックアップされたデータより新しい更新がコンシューマ上のデータに含まれている可能性があるためです。この場合は、コンシューマレプリカから LDIF ファイルにデータをエクスポートし、この LDIF ファイルを使用してマスターを初期化し直します。

バックアップから復元する場合でも、LDIF ファイルをインポートする場合でも、このマスターレプリカから更新を受け取るすべてのハブレプリカとコンシューマレプリカをあとで初期化し直す必要があります。コンシューマの再初期化が必要であることを示すメッセージが、サプライヤサーバーのログファイルに記録されます。

マルチマスターモデルでのサプライヤの復元

マルチマスターレプリケーションでは、ほかの各マスターも、レプリケートされるデータに対してコピーする権限を持っています。現在のレプリカの内容が反映されていない可能性があるため、古いバックアップを復元することはできません。可能な場合は、レプリケーションメカニズムにより、ほかのマスターの内容を使用してマスターを更新するようにしてください。

それが不可能な場合は、次のどちらかの方法でマルチマスターレプリカを復元する必要があります。

- もっとも簡単な方法として、バックアップから復元する代わりに、ほかのマスターの 1 つを使用して目的のマスターを初期化し直します。これにより、目的のマスターに最新のデータが送られ、データはすぐにレプリケートできる状態になります。詳細は、299 ページの「コンソールによるレプリカの初期化」または 300 ページの「コマンド行によるレプリカの初期化」を参照してください。
- 数百万のエントリを持つレプリカの場合は、新機能であるバイナリコピーを使用して、ほかのマスターの 1 つから作成したより新しいバックアップから復元することで、所要時間を短縮できます。詳細は、302 ページの「バイナリコピーによるレプリカの初期化」を参照してください。
- このマスターのバックアップが、ほかのどのマスターに対しても更新履歴ログの最長保存期間を過ぎていない場合は、このバックアップを使用してマスターを復元できます。更新履歴ログの保存期間については、287 ページの「マルチマスターの詳細設定」を参照してください。このようにバックアップからマスターを復元すると、ほかのマスターはそれぞれの更新履歴ログを使用して、このマスターを更新します。これにより、バックアップの作成時以降に加えられた変更内容がすべてこのマスターに反映されます。

復元や再初期化の方法にかかわらず、初期化後のマスターレプリカは読み取り専用モードになります。この動作により、このレプリカとほかのマスターとの同期をとったあとに、書込み操作を許可できます。詳細は、296 ページの「マルチマスター初期化後のマスター間の一致」を参照してください。

復元または初期化し直したマスターに書き込み操作を許可する前に、すべてのレプリカを反映させることができるので、ハブサーバーやコンシューマサーバーを初期化し直すことが不要になるという利点があります。

ハブの復元

この節の内容は、レプリケーションメカニズムで自動的にハブレプリカを更新できない場合だけに適用されます。たとえば、データベースファイルが破損した場合や、レプリケーションが長時間にわたって中断された場合などに適用されます。このような場合は、次のどちらかの方法で、ハブレプリカを復元または初期化し直す必要があります。

- もっとも簡単な方法として、バックアップから復元する代わりに、ほかのマスターレプリカの 1 つを使用してハブを初期化し直します。これにより、ハブに最新のデータが送られ、データはすぐにレプリケートできる状態になります。詳細は、299 ページの「コンソールによるレプリカの初期化」または 300 ページの「コマンド行によるレプリカの初期化」を参照してください。
- 数百万のエントリを持つレプリカの場合は、新機能であるバイナリコピーを使用して、別のハブレプリカから作成したより新しいバックアップから復元することで、所要時間を短縮できます。詳細は、302 ページの「バイナリコピーによるレプリカの初期化」を参照してください。コピーできるほかのハブレプリカがない場合は、前述の方法でハブを初期化し直すか、可能な場合は次の方法でハブを復元する必要があります。
- このハブのバックアップが、そのサプライヤ (ハブレプリカまたはマスターレプリカ) のどちらに対しても更新履歴ログの最長保存期間を過ぎていない場合は、このバックアップを使用してハブを復元できます。更新履歴ログの保存期間については、287 ページの「マルチマスターの詳細設定」を参照してください。このようにバックアップからハブを復元すると、そのサプライヤはそれぞれの更新履歴ログを使用して、このハブを更新します。これにより、バックアップの作成時以降に加えられた変更内容が、すべてこのハブに反映されます。

注 ハブレプリカの復元や再初期化の方法にかかわらず、あとでこのハブのコンシューマをすべて初期化し直す必要があります。ほかのレベルのハブもすべて初期化し直す必要があります。

専用コンシューマの復元

この節の内容は、レプリケーションメカニズムで自動的に専用コンシューマレプリカを更新できない場合だけに適用されます。たとえば、データベースファイルが破損した場合や、レプリケーションが長時間にわたって中断された場合などに適用されます。このような場合は、次のどちらかの方法で、コンシューマを復元または初期化し直す必要があります。

- もっとも簡単な方法として、バックアップから復元する代わりに、そのサプライヤの 1 つ (マスターレプリカまたはハブレプリカ) を使用してコンシューマを初期化し直します。これにより、コンシューマに最新のデータが送られ、データはすぐにレプリケートできる状態になります。詳細は、299 ページの「コンソールによるレプリカの初期化」または 300 ページの「コマンド行によるレプリカの初期化」を参照してください。
- 数百万のエントリを持つレプリカの場合は、新機能であるバイナリコピーを使用して、別のコンシューマレプリカから作成したより新しいバックアップから復元することで、所要時間を短縮できます。詳細は、302 ページの「バイナリコピーによるレプリカの初期化」を参照してください。コピーできるほかのコンシューマがない場合は、前述の方法でそのレプリカを初期化し直すか、可能な場合は次の方法でレプリカを復元する必要があります。
- このコンシューマのバックアップが、そのサプライヤ (ハブレプリカまたはマスターレプリカ) のどちらに対しても更新履歴ログの最長保存期間を過ぎていない場合は、このバックアップを使用してコンシューマを復元できます。更新履歴ログの保存期間については、287 ページの「マルチマスターの詳細設定」を参照してください。このようにバックアップからハブを復元すると、そのサプライヤはそれぞれの更新履歴ログを使用して、このハブを更新します。これにより、バックアップの作成時以降に加えられた変更内容が、すべてこのハブに反映されます。

コンソールからのサーバーの復元

ディレクトリデータが壊れた場合、Directory Server コンソールを使用して、以前作成されたバックアップからデータを復元できます。コンソールを使用してデータベースを復元するには、Directory Server を動作させておく必要があります。ただし、復元中に対応するサフィックスの処理を行うことはできません。

以前に作成したバックアップからサーバーを復元するには、次の手順を実行します。

1. Directory Server コンソールの最上位にある「タスク」タブで、「Directory Server の復元」の隣にあるボタンをクリックします。

   「Directory Server の復元」ダイアログボックスが表示されます。

2. 「使用可能なバックアップ」リストからバックアップを選択します。あるいは、「ディレクトリ」テキストボックスに、有効なバックアップファイルの絶対パスを入力します。

   「使用可能なバックアップ」リストには、デフォルトディレクトリに置かれているすべてのバックアップが表示されます

   ServerRoot/slapd-*serverID*/bak

3. 「了解」をクリックすると、サーバーが復元されます。

コマンド行からのサーバーの復元

次に示すスクリプトを使用すると、コマンド行からサーバーを復元できます。

- bak2db コマンド (Solaris パッケージ内の directoryserver bak2db) このスクリプトを使用する場合は、サーバーを停止させる必要がある
- bak2db.pl Perl スクリプト (Solaris パッケージ内の directoryserver bak2db-task) このスクリプトを使用する場合は、サーバーを動作させておく必要がある

bak2db コマンド行スクリプトの使用

サーバーの停止中にコマンド行からディレクトリを復元するには、次の手順を実行します。

1. root としてコマンド行に次のコマンドを入力し、サーバーを停止させます。

Solaris パッケージ
```
# /usr/sbin/directoryserver stop
```
その他のインストール
```
# ServerRoot/slapd-serverID/stop-slapd
```

2. バックアップディレクトリへの絶対パスを指定して bak2db コマンドを使用します。

Solaris パッケージ
```
# /usr/sbin/directoryserver bak2db backupDir
```
その他のインストール
```
# ServerRoot/slapd-serverID/bak2db backupDir
```

3. 適切なコマンドでサーバーを起動します。

Solaris パッケージ
```
# /usr/sbin/directoryserver start
```
その他のインストール
```
# ServerRoot/slapd-serverID/start-slapd
```

次の例では、デフォルトのバックアップディレクトリからバックアップを復元します。

```
# bak2db /var/Sun/mps/slapd-example/bak/2001_07_01_11_34_00
```

詳細については、『Sun ONE Directory Server Reference Manual』の第 2 章にある「insync」を参照してください。

bak2db.pl Perl スクリプトの使用

サーバーの動作中にコマンド行を使ってディレクトリを復元するには、次の Perl スクリプトを使用します。

Solaris パッケージ
Windows プラットフォーム
その他のインストール

```
# /usr/sbin/directoryserver bak2db-task
cd ServerRoot
bin¥slapd¥admin¥bin¥perl slapd-serverID¥bak2db.pl
# ServerRoot/slapd-serverID/bak2db.pl
```

次の例では、ldif2db.pl スクリプトを使用して、LDIF ファイルをインポートします。-a オプションを指定すると、バックアップディレクトリの完全パスが表示されます。

UNIX シェルスクリプト

```
# use directoryserver bak2db-task on Solaris パッケージ installations
/var/Sun/mps/slapd-example/bak2db.pl ¥
  -D "cn=Directory Manager" -w password ¥
  -a /var/Sun/mps/slapd-example/bak/checkpoint
```

Windows バッチファイル

```
C:¥Program Files¥Sun¥MPS¥bin¥slapd¥admin¥bin¥perl.exe
 C:¥Program Files¥Sun¥MPS¥slapd-example¥bak2db.pl
  -D "cn=Directory Manager" -w password
  -a C:¥Program Files¥Sun¥MPS¥slapd-example¥bak¥2001_07_01_11_34_00
```

詳細については、『Sun ONE Directory Server Reference Manual』の第 2 章にある「bak2db.pl」を参照してください。

dse.ldif 設定ファイルの復元

次のディレクトリ内に、dse.ldif ファイルのバックアップコピーが 2 つ作成されます。

ServerRoot/slapd-*serverID*/config

dse.ldif.startOK ファイルには、サーバー起動時に dse.ldif ファイルのコピーが記録されます。dse.ldif.bak ファイルは、dse.ldif ファイルに加えられた最新の変更内容のバックアップが含まれます。最新の変更内容を含むファイルを自分のディレクトリにコピーします。

dse.ldif 設定ファイルを復元するには、次の手順を実行します。

1. root としてコマンド行に次のコマンドを入力し、サーバーを停止させます。

Solaris パッケージ
```
# /usr/sbin/directoryserver stop
```
その他のインストール
```
# ServerRoot/slapd-serverID/stop-slapd
```

2. 次の設定ファイルを含むディレクトリに移動します。
3. 正常であると考えられるバックアップ設定ファイルで dse.ldif ファイルを上書きします。たとえば、次のように入力します。

```
cp dse.ldif.startOK dse.ldif
```

4. 適切なコマンドでサーバーを起動します。

Solaris パッケージ
```
# /usr/sbin/directoryserver start
```
その他のインストール
```
# ServerRoot/slapd-serverID/start-slapd
```

第 5 章

高度なエントリの管理

ユーザーを表すエントリを管理するには、ディレクトリ内のデータの階層構造を超えて、グループを作成したり、共通の属性値を共有したりすることがしばしば必要になります。Sun ONE Directory Server では、グループ、ロール、およびサービスクラス (CoS) を使ってエントリを高度に管理できます。

グループとは、メンバーのリストまたはメンバーに適用するフィルタを使用して、ほかのエントリを指定するエントリです。ロールは、ロールの各メンバーに対して `nsrole` 属性を生成するメカニズムによって、グループと同等またはそれ以上の機能を提供します。CoS も仮想属性を生成します。これにより、エントリは、各エントリに値を格納することなく、共通の属性値を共有できるようになります。

注 Sun ONE Directory Server 5.2 には、ロールと CoS 仮想属性の値に基づく検索を実行する機能があります。どの操作で使用されるフィルタ文字列にも、`nsRole` 属性または CoS 定義によって生成された任意の属性を含めることができ、この属性の値について任意の比較演算を実行できます。ただし、CoS 仮想属性にはインデックスを付けることができないため、CoS で生成された属性を使用する検索は、すべてインデックスを使用しない検索となります。

ロールとサービスクラスが提供する機能を活用するには、ディレクトリの導入を計画する段階で、ディレクトリのトポロジを決定しておく必要があります。これらのメカニズムの詳細と、それによってトポロジを単純化できるしくみについては、『Sun ONE Directory Server Deployment Guide』の第 4 章「Designing the Directory Tree」を参照してください。

この章は、次の節で構成されています。

- グループの管理
- ロールの割り当て
- サービスクラス (CoS) の定義

グループの管理

グループとは、ACI の定義などのように、管理しやすくするためにエントリを相互に関連付けるメカニズムです。グループ定義は特別なエントリで、スタティックなリストにメンバーの名前を指定するか、またはダイナミックなエントリセットを定義するフィルタを指定します。同等のロール定義を作成する手順については、157 ページの「ロールの割り当て」を参照してください。

グループに含めることが可能なメンバーの範囲は、グループ定義エントリの位置に関係なく、ディレクトリ全体となります。管理を簡略化するために、すべてのグループ定義エントリは、通常、1 か所に格納されます。通常は、ルートサフィックスの下の `ou=Groups` に格納されます。

スタティックグループを定義するエントリは、`groupOfUniqueNames` オブジェクトクラスから継承されます。グループのメンバーは、その DN ごとに `uniqueMember` 属性の複数値として一覧表示されます。

ダイナミックグループを定義するエントリは、`groupOfUniqueNames` および `groupOfURLs` オブジェクトクラスから継承されます。グループのメンバーシップは、複数値属性 `memberURL` に指定された、1 つまたは複数のフィルタによって定義されます。フィルタが評価されたときにそのどれかに一致するエントリが、ダイナミックグループのメンバーとなります。

次の節では、コンソールを使用してスタティックグループとダイナミックグループを作成および変更する方法について説明します。

新しいスタティックグループの追加

1. Directory Server コンソールの最上位の「ディレクトリ」タブを選択し、ディレクトリツリーで、新しいグループの追加先エントリを右クリックします。次に、「新規」>「グループ」の順に選択します。

   または、エントリを選択し、「オブジェクト」メニューから「新規」>「グループ」の順に選択します。

2. 「新規グループの作成」ダイアログボックスで、新しいグループの名前を「グループ名」フィールドに入力します。グループの説明を「説明」フィールドに追加することもできます。このグループ名は、新しいグループエントリの `cn` (共通名) 属性の値になり、その DN の中に表示されます。

3. ダイアログボックスの左側のリストで「メンバー」をクリックします。右側のパネルでは、「スタティックグループ」タブがデフォルトで選択されています。

4. 「追加」をクリックして、グループに新しいメンバーを追加します。標準の「ユーザーとグループの検索」ダイアログボックスが表示されます。

5. 「検索」ドロップダウンリストから「ユーザー」を選択し、検索する文字列を入力してから、「検索」をクリックします。特定の属性や属性値を検索するには、「詳細」ボタンをクリックします。

   検索結果からエントリを 1 つ以上選択し、「了解」をクリックします。この手順を繰り返して、このスタティックグループに必要なメンバーをすべて追加します。

注 スタティックグループのメンバーは、連鎖によってリモートに存在する可能性があります。参照整合性検査プラグインを使用すると、削除されたメンバーのエントリをスタティックグループのエントリから自動的に削除できます。連鎖と参照整合性を併用する方法については、115 ページの「連鎖ポリシーの設定」を参照してください。

6. 左側のリストで「言語」をクリックし、グループの名前と説明を別の言語で指定します。これらは、その言語に対応するロケールがコンソールで使用される場合に表示されます。

7. 「了解」をクリックすると、新しいグループが作成されます。グループは、そのグループが作成された場所であるエントリの子の 1 つとして表示されます。

新しいダイナミックグループの追加

1. Directory Server コンソールの最上位の「ディレクトリ」タブを選択し、ディレクトリツリーで、新しいグループの追加先エントリを右クリックします。次に、「新規」>「グループ」の順に選択します。

   または、エントリを選択し、「オブジェクト」メニューから「新規」>「グループ」の順に選択します。

2. 「新規グループの作成」ダイアログボックスで、新しいグループの名前を「グループ名」フィールドに入力します。グループの説明を「説明」フィールドに追加することもできます。このグループ名は、新しいグループエントリの cn (共通名) 属性の値になり、その DN の中に表示されます。

3. ダイアログボックスの左側のリストで「メンバー」をクリックし、右側のパネルで「ダイナミックグループ」タブを選択します。

4. 「追加」をクリックして、グループのメンバーを定義するフィルタ文字列を含んだ LDAP URL を作成します。標準の「LDAP URL の構成とテスト」ダイアログボックスが表示されます。

5. テキストフィールドに LDAP URL を入力するか、または「構成」を選択し、指示に従って、グループに適用するフィルタを含む LDAP URL を作成します。「テスト」をクリックして、このフィルタで取得されるエントリのリストを表示します。

   URL の作成が完了したら、「了解」をクリックします。この手順を繰り返して、ダイナミックグループを定義するフィルタを含んだ URL をすべて追加します。

6. 左側のリストで「言語」をクリックし、グループの名前と説明を別の言語で指定します。これらは、その言語に対応するロケールがコンソールで使用される場合に表示されます。
7. 「了解」をクリックすると、新しいグループが作成されます。グループは、そのグループが作成された場所であるエントリの子の 1 つとして表示されます。

グループ定義の変更

1. Directory Server コンソールの最上位の「ディレクトリ」タブで、変更するグループを表すエントリをダブルクリックします。

   または、エントリを選択し、「オブジェクト」メニューの「開く」を選択します。
2. 「エントリの編集」ダイアログボックスで、「一般」、「メンバー」、「言語」の各カテゴリのグループ情報を変更します。スタティックグループの場合はメンバーの追加と削除、ダイナミックグループの場合はフィルタを含んだ URL の追加、編集、削除を行うことができます。
3. グループ定義の変更が完了したら、「了解」をクリックします。

   変更内容をコンソールで確認するには、「表示」メニューの「再表示」を選択します。

グループ定義の削除

グループのタイプのどれかを削除するには、そのグループを定義するエントリを削除します。

ロールの割り当て

ロールは、アプリケーションでより効率的に簡単に使用できる新しいグループ化メカニズムです。ロールは、グループと同じように定義および管理されますが、それに加えて、メンバーエントリにも、所属するロールを示す属性が生成されます。たとえば、アプリケーションでは、グループを選択してメンバーリストを参照しなくても、エントリのロールを読み取るだけで済みます。

デフォルトでは、ロールの適用範囲は、それが定義されているサブツリーに限定されます。Sun ONE Directory Server 5.2 では、入れ子のロールの適用範囲が拡張されたため、ほかのサブツリーにあるロールも入れ子にでき、ディレクトリ内の任意の位置にメンバーを置くことができます。

ロールについて

各ロールはメンバー、つまりそのロールを所有するエントリを持ちます。ディレクトリからエントリが取得されるとき、ロールメカニズムによって自動的に、なんらかのロールに所属するすべてのエントリに `nsRole` 属性が生成されます。この複数値属性には、そのエントリをメンバーとして持つすべてのロール定義の DN が値として設定されます。`nsRole` 属性は、算出される属性であるためエントリ自体には格納されませんが、処理結果は通常の属性としてクライアントアプリケーションに返されます。

Sun ONE Directory Server は、次の 3 種類のロールをサポートしています。

- 管理されているロール : 管理者は、対象となるメンバーエントリに `nsRoleDN` 属性を追加することにより、管理されているロールを割り当てることができます。この属性の値は、ロール定義エントリの DN です。管理されているロールは、メンバーがロール定義エントリではなく各エントリに定義されていることを除いて、スタティックグループと似ています。
- フィルタを適用したロール : このロールはダイナミックグループと同じです。このロールでは、`nsRoleFilter` 属性にフィルタ文字列を定義します。フィルタを適用したロールの適用範囲は、定義エントリの親をルートとする、ロールが位置するサブツリーです。サーバーが、フィルタ文字列に一致した、フィルタを適用したロールの適用範囲内のエントリを返す場合、そのエントリにはロールを識別する `nsRole` 属性が含まれています。
- 入れ子のロール : ほかの入れ子のロールも含め、ほかのロール定義を指定して使用するロールです。入れ子のロールに含まれているロールのすべてのメンバーが、入れ子のロールのメンバーとなります。入れ子のロールでは、その適用範囲を拡張して、ほかのサブツリーにあるロールのメンバーを含めることもできます。

ロールは、nsRole 属性の内容を直接読み取ることで、エントリのすべてのロールメンバーシップをクライアント アプリケーションに知らせます。これにより、クライアントの処理が簡素化され、ディレクトリの使用が最適化されます。ロールと CoS メカニズムを組み合わせて、ロールメンバーの他の属性を生成することもできます (181 ページの「ロールに基づく属性の作成」を参照)。ロールは、アクセス制御の定義にも使用できます (206 ページの「ロールアクセスの定義 : roledn キーワード」を参照)。またロールには、そのロールのすべてのメンバーを一度に有効化または無効化する機能も用意されています (269 ページの「ユーザーとロールの無効化と有効化」を参照)。

nsRole 属性の検索

Sun ONE Directory Server 5.2 では、任意の検索フィルタで nsRole 属性を使用できるようになりました。任意の比較演算子を使用して、この属性の特定の値を検索できます。ただし、次の事項に注意してください。

- nsRole 属性を使用する検索では、エントリにフィルタを適用する前にすべてのロールを評価する必要があるため、所要時間がかなり長くなります。
- Directory Server は、特に管理されているロールのメンバーシップの等価検索用に最適化されています。たとえば、次の検索は、実際の属性に対する検索とほぼ同じ速さで実行されます。

  ```
  (&(objectclass=person)
    (nsRole=cn=managersRole,ou=People,dc=example,dc=com)
  ```

- 管理されているロールのメンバーシップを定義する nsRoleDN 属性には、デフォルトではすべてのサフィックスでインデックスが付けられます。Directory Server は管理されているロールのメンバーシップの検索用に最適化されていますが、この属性のインデックス付けが無効になっていると、その効果は失われます。
- フィルタが適用されたロールを含むエントリの検索には、ロールフィルタを使用した内部検索が関連します。この内部処理が最も速くなるのは、ロールフィルタ内に表示されるすべての属性に、ロールの適用範囲内にあるすべてのサフィックスでインデックスが付けられている場合です。

nsRole 属性に対するアクセス権

nsRole 属性はロールメカニズムによってだけ割り当てられ、ディレクトリユーザーが書き込みや変更を行うことはできません。ただし、次の事項に注意してください。

- ディレクトリユーザーなら誰でも、基本的には nsRole 属性を読み取ることができますが、管理者はアクセス制御を定義してこれを防止できます。
- nsRoleDN 属性は、管理されているロールのメンバーシップを定義します。自身をロールに追加したり削除したりする許可をユーザーに与えるかどうかは、管理者が決定します。ユーザーが自身のロールを変更できないようにする ACI については、164 ページの「管理されているロール定義の例」を参照してください。

- フィルタを適用したロールでは、ユーザーエントリ内に特定の属性値が存在するかどうかに基づいてメンバーが定義されます。フィルタを適用したロールでは、そのメンバーシップを定義できるユーザーを制限するために、これらの属性のユーザー権限を慎重に定義する必要があります。

ディレクトリでのロールの使用方法については、『Sun ONE Directory Server Deployment Guide』の第 4 章「Designing the Directory Tree」を参照してください。

コンソールを使用したロールの割り当て

ここでは、ロールの作成と変更に関する次の手順について説明します。

管理されているロールの作成

管理されているロールにはロール定義エントリが 1 つあり、メンバーエントリに `nsRoleDN` 属性を追加することでメンバーが定義されます。コンソール を使用して、管理されているロールを作成してメンバーを追加するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「ディレクトリ」タブを選択し、ディレクトリツリーで、新しいロール定義の追加先エントリを右クリックします。次に、「新規」>「ロール」の順に選択します。

   または、エントリを選択し、「オブジェクト」メニューから「新規」>「ロール」の順に選択します。

2. 「新規ロールの作成」ダイアログボックスで、新しいロールの名前を「ロール名」フィールドに入力します。ロールの説明を「説明」フィールドに追加することもできます。このロール名は、新しいロールエントリの cn (共通名) 属性の値になり、その DN の中に表示されます。

3. ダイアログボックスの左側のリストで「メンバー」をクリックします。右側のパネルでは、「管理されているロール」ラジオボタンがデフォルトで選択されています。

4. このロールに新しいメンバーを追加するために、メンバーリストの下の「追加」をクリックします。標準の「ユーザーとグループの検索」ダイアログボックスが表示されます。

5. 「検索」ドロップダウンリストから「ユーザー」を選択し、検索する文字列を入力してから、「検索」をクリックします。特定の属性や属性値を検索するには、「詳細」ボタンをクリックします。

   検索結果からエントリを 1 つ以上選択し、「了解」をクリックします。この手順を繰り返して、このスタティックロールに必要なメンバーをすべて追加します。

6. ロールへのエントリの追加が完了したら、「了解」をクリックします。管理されているロールを表すアイコンとともに、この新しいロールがディレクトリツリーに表示されます。また、すべてのメンバーエントリに nsRoleDN 属性が追加され、この新しいロールエントリの DN の値が設定されます。
7. ロールを作成したら、このロールを任意のエントリに割り当てることができます。それには、目的のエントリに nsRoleDN 属性を追加し、ロールエントリの DN の値を設定します。

フィルタを適用したロールの作成

フィルタを適用したロールでは、ロール定義の LDAP フィルタに選択される属性や属性値を持つエントリが、このロールのメンバーとなります。

注 フィルタを適用したロールのフィルタ文字列には、任意の属性を使用できます。ただし、CoS メカニズムによって生成されるその他の仮想属性は使用できません (168 ページの「CoS について」を参照)。

コンソールを使用して、フィルタを適用したロールを作成してメンバーを追加するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「ディレクトリ」タブを選択し、ディレクトリツリーで、新しいロール定義の追加先エントリを右クリックします。次に、「新規」>「ロール」の順に選択します。

   または、エントリを選択し、「オブジェクト」メニューから「新規」>「ロール」の順に選択します。
2. 「新規ロールの作成」ダイアログボックスで、新しいロールの名前を「ロール名」フィールドに入力します。ロールの説明を「説明」フィールドに追加することもできます。このロール名は、新しいロールエントリの cn (共通名) 属性の値になり、その DN の中に表示されます。
3. ダイアログボックスの左側のリストで「メンバー」をクリックし、右側のパネルで「フィルタを適用したロール」ラジオボタンを選択します。
4. ロールのメンバーを決定するための LDAP フィルタをテキストフィールドに入力します。または、「構成」をクリックし、指示に従って LDAP フィルタを作成します。
5. 「構成」をクリックすると、「LDAP フィルタの構成」ダイアログが表示されます。フィルタを適用したロールの定義では、「LDAP サーバーホスト」、「ポート」、「ベース DN」、および「検索」の各フィールドを指定できないので、これらは無視します。

a. フィルタを適用したロールのユーザーだけを検索します。これにより、(objectclass=person) というコンポーネントがフィルタに追加されます。このコンポーネントを使用しない場合は、「新規ロールの作成」ダイアログボックスのテキストフィールドで LDAP フィルタを編集する必要があります。

b. 「場所」ドロップダウンリストから属性を選択し、一致条件を設定して、このフィルタを詳しく定義します。フィルタを追加するには、「フィルタの追加」をクリックします。不要なフィルタを削除するには、「フィルタの削除」をクリックします。

c. 「了解」をクリックして、フィルタを適用したロールの定義にこのフィルタを追加します。その後、必要に応じてフィルタのコンポーネントをテキストフィールドで編集できます。

6. 「テスト」をクリックして、フィルタをテストします。「フィルタテスト結果」ダイアログボックスに、その時点でフィルタに一致するエントリが表示されます。

7. 「了解」をクリックして、この新しいロールエントリを作成します。フィルタを適用したロールを表すアイコンとともに、この新しいロールがディレクトリツリーに表示されます。

入れ子のロールの作成

入れ子のロールを使用すると、別のロールを含むロールを作成でき、既存のロールの適用範囲を拡張できます。入れ子のロールを作成する前に、別のロールを作成しておく必要があります。入れ子のロールを作成する場合は、入れ子にできるロールのリストが表示されます。入れ子のロールには、最大で 30 の段階まで、さらに入れ子のロールが含まれている可能性があります。この制限を超えた入れ子構造が含まれると、ロールの評価時にサーバーがエラーを記録します。

コンソールを使用して、入れ子のロールを作成してメンバーを追加するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「ディレクトリ」タブを選択し、ディレクトリツリーで、新しいロール定義の追加先エントリを右クリックします。次に、「新規」>「ロール」の順に選択します。

   または、エントリを選択し、「オブジェクト」メニューから「新規」>「ロール」の順に選択します。

2. 「新規ロールの作成」ダイアログボックスで、新しいロールの名前を「ロール名」フィールドに入力します。ロールの説明を「説明」フィールドに追加することもできます。このロール名は、新しいロールエントリの cn (共通名) 属性の値になり、その DN の中に表示されます。

3. ダイアログボックスの左側のリストで「メンバー」をクリックし、右側のパネルで「入れ子になっているロール」ラジオボタンを選択します。

4. 「追加」をクリックして、既存のロールを入れ子のロールのリストに追加します。「ロールセレクタ」ダイアログボックスで、利用可能なロールのリストから 1 つまたは複数のロールを選択し、「了解」をクリックします。
5. 「了解」をクリックして、入れ子のロールエントリを作成します。ディレクトリに新しいロールと入れ子のロールのアイコンが表示されます。
6. 入れ子のロールの適用範囲を変更するには、166 ページの「入れ子のロール定義の例」の手順をコマンド行から実行します。

エントリのロールの表示と編集

1. Directory Server コンソールの最上位レベルにある「ディレクトリ」タブでディレクトリツリーを表示し、表示または編集するエントリを探します。
2. このエントリをマウスの右ボタンでクリックし、ポップアップメニューから「ロールを設定」を選択します。あるいは、エントリをクリックして選択し、「オブジェクト」メニューから「ロールを設定」を選択します。

   「ロールを設定」ダイアログが表示されます。
3. 「管理されているロール」タブを選択すると、このエントリが所属する管理されているロールが表示されます。次の処理を実行できます。

- 新しい管理されているロールを追加するには、「追加」をクリックし、「ロールセレクタ」ウィンドウから使用可能なロールを選択します。「ロールセレクタ」ウィンドウで「了解」をクリックします。
- 管理されているロールを削除するには、削除するロールを選択し、「削除」をクリックします。
- エントリに関連付けられた管理されているロールを編集するには、テーブルからロールを選び、「編集」をクリックします。ロールのカスタムエディタにロールが表示されます。ロールに変更を加え、「了解」をクリックして新しいロール定義を保存します。

4. 「その他のロール」タブを選択すると、このエントリが所属する、フィルタを適用したロールや入れ子のロールが表示されます。フィルタを適用したロールまたは入れ子のロールのロールのメンバーシップを変更するときは、ロール定義を編集する必要があります。

- ロールを選択して「編集」をクリックし、そのロールのカスタムエディタを表示します。ロールに変更を加え、「了解」をクリックして新しいロール定義を保存します。

5. ロールの変更が完了したら、「了解」をクリックして、変更を保存します。

ロールのエントリの変更

1. Directory Server コンソールで、「ディレクトリ」タブを選択します。

2. ナビゲーションツリーを参照して、既存のロールの定義エントリを検索します。ロールは、そのロールを作成した位置の子エントリになります。ロールをダブルクリックします。

   「ロールの編集」ダイアログボックスが表示されます。

3. ロールの名前と説明を変更するには、左側のパネルで「一般」をクリックします。
4. 管理されているロールと入れ子のロールのメンバーを変更するか、またはフィルタを適用したロールのフィルタを変更する場合は、左側のパネルで「メンバー」をクリックします。
5. 「了解」をクリックして、変更を保存します。

ロールの削除

ロールを削除すると、ロール定義のエントリだけが削除されます。ロールのメンバーが削除されることはありません。

ロールを削除するには、次の手順を実行します。

1. Directory Server コンソールで、「ディレクトリ」タブを選択します。
2. ナビゲーションツリーを参照して、ロールの定義エントリを検出します。ロールは、そのロールを作成した位置の子エントリになります。
3. ロールを右クリックし、「削除」を選択します。

   削除の確認を求めるダイアログボックスが表示されます。「はい」をクリックします。

4. ロールが正しく削除されたことを通知する「削除されたエントリ」ダイアログボックスが表示されます。「了解」をクリックします。

注 ロールを削除すると、ロールエントリは削除されますが、各ロールメンバーの `nsRoleDN` 属性は削除されません。この属性を削除するには、参照整合性検査プラグインを有効にし、`nsRoleDN` 属性を管理します。詳細は、81 ページの「参照整合性の管理」を参照してください。

コマンド行からのロールの管理

ロールは、ディレクトリ管理者がコマンド行ユーティリティを使用してアクセスできるようにエントリに定義されます。ロールの作成が完了したら、次のようにロールにメンバーを割り当てます。

- 管理されているロールのメンバーのエントリに、nsRoleDN 属性を含める
- フィルタを適用したロールのメンバーは、nsRoleFilter 属性で指定したフィルタに一致するエントリとなる
- 入れ子のロールのメンバーは、入れ子のロール定義エントリの nsRoleDN 属性で指定したロールのメンバーとなる

すべてのロール定義は、LDAPsubentry および nsRoleDefinition オブジェクトクラスから継承されます。次の表に、各ロールタイプに固有のその他のオブジェクトクラスと関連付けられた属性を示します。

表 5-1 ロールを定義するオブジェクトクラスと属性

ロールタイプ	オブジェクトクラス	属性
管理されているロール	nsSimpleRoleDefinition nsManagedRoleDefinition	Description (省略可能)
フィルタを適用したロール	nsComplexRoleDefinition nsFilteredRoleDefinition	nsRoleFilter Description (省略可能)
入れ子のロール	nsComplexRoleDefinition nsNestedRoleDefinition	nsRoleDN Description (省略可能)

管理されているロール定義の例

すべてのマーケティングスタッフに割り当てるロールを作成するには、次の ldapmodify コマンドを実行します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=Marketing,ou=marketing,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsSimpleRoleDefinition
objectclass: nsManagedRoleDefinition
cn: Marketing
description: managed role for marketing staff
```

nsManagedRoleDefinition オブジェクトクラスは、LDAPsubentry、nsRoleDefinition、および nsSimpleRoleDefinition の各オブジェクトクラスから継承されることに注意してください。

次のように ldapmodify コマンドを実行して、Bob のエントリを更新することによって、Bob というマーケティングスタッフメンバーにロールを割り当てます。

```
ldapmodify -D "cn=Directory Manager" -w secret -h host -p 389
dn: cn=Bob Arnold,ou=marketing,ou=People,dc=example,dc=com
changetype: modify
add: nsRoleDN
nsRoleDN: cn=Marketing,ou=marketing,ou=People,dc=example,dc=com
```

エントリ内の nsRoleDN 属性は、そのエントリが管理されているロールのメンバーであることを示します。これは、そのロール定義の DN で判別されます。ユーザーが nsRoleDN 属性を変更できると、管理されているロールに自身を追加したり削除したりできます。これを防ぐには、次の ACI (アクセス制御命令) を追加します。

```
aci: (targetattr="nsRoleDN")
     (targattrfilters="
add=nsRoleDN:(!(nsRoleDN=cn=AdministratorRole,dc=example,dc=com)),
del=nsRoleDN:(!(nsRoleDN=cn=nsManagedDisabledRole,dc=example,dc=com
)
     ")
      (version3.0;aci "allow mod of nsRoleDN by self
        except for critical values";
        allow(write)
          userdn="ldap:///self";)
```

フィルタを適用したロール定義の例

セールスマネージャ用にフィルタを適用したロールを設定するには、全員が isManager 属性を持っていると仮定し、次の ldapmodify コマンドを実行します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=ManagerFilter,ou=sales,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: ManagerFilter
nsRoleFilter: (isManager=True)
Description: filtered role for sales managers
```

nsFilteredRoleDefinition オブジェクトクラスは、LDAPsubentry、nsRoleDefinition、および nsComplexRoleDefinition の各オブジェクトクラスから継承されることに注意してください。たとえば次のように、nsRoleFilter 属性は、ou=sales という組織に含まれ、下位組織を持つすべての従業員を検索するフィルタを指定します。

```
ldapsearch -h host -p port -D "cn=Directory Manager" -w password ¥
-b "ou=People,dc=example,dc=com" -s sub "(cn=*Fuentes)"

dn: cn=Carla Fuentes,ou=sales,ou=People,dc=example,dc=com
cn: Carla Fuentes
isManager: TRUE
...
nsRole: cn=ManagerFilter,ou=sales,ou=People,dc=example,dc=com
```

注 フィルタを適用したロールのフィルタ文字列には、任意の属性を使用できます。ただし、CoS メカニズムによって生成されるその他の仮想属性は使用できません (168 ページの「CoS について」を参照)。

フィルタを適用したロールのメンバーがユーザーエントリである場合、ユーザーが自身をロールに追加したり削除したりできないように制限するには、フィルタを適用した属性を ACI (アクセス制御命令) で保護します。

入れ子のロール定義の例

入れ子のロール内に含めるロールを指定するには、nsRoleDN 属性を使用します。前述の例で作成したロールに含まれるマーケティングスタッフとセールスマネージャの両方を含むロールを作成するには、次のコマンドを使用します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=MarketingSales,ou=marketing,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsNestedRoleDefinition
cn: MarketingSales
nsRoleDN: cn=ManagerFilter,ou=sales,ou=People,dc=example,dc=com
nsRoleDN: cn=Marketing,ou=marketing,ou=People,dc=example,dc=com
nsRoleScopeDN: ou=sales,ou=People,dc=example,dc=com
```

nsNestedRoleDefinition オブジェクトクラスは、LDAPsubentry、nsRoleDefinition、および nsComplexRoleDefinition の各オブジェクトクラスから継承されることに注意してください。nsRoleDN 属性は、マーケティングの管理されているロールの DN とセールスマネージャのフィルタを適用したロールの DN を含みます。前述の例のユーザー Bob と Carla は、どちらもこの新しい入れ子のロールのメンバーになります。

このフィルタの適用範囲としては、デフォルトの適用範囲 (フィルタが置かれているサブツリー) に加え、nsRoleScopeDN 属性のすべての値の下にあるサブツリーが含まれます この例では、ManagerFilter は ou=sales,ou=People,dc=example,dc=com サブツリーに置かれているため、このサブツリーも適用範囲に追加されます。

サービスクラス (CoS) の定義

クライアントアプリケーション用にエントリが取得されるとき、サービスクラス (CoS) メカニズムによって仮想属性が生成されます。CoS を使用すると、エントリの管理が簡素化され、必要な格納領域が減少します。

グループやロールと同じように、CoS はディレクトリのヘルパーエントリに依存し、コンソールまたはコマンド行を使用して設定できます。次の節では、CoS について詳しく説明し、コンソールおよびコマンド行を使用して CoS を管理するための手順について説明します。

注 Directory Server 5.2 の新機能として、すべての検索操作で、CoS で生成された属性の有無を調べたり、その値を比較したりできます。クライアントの検索操作で使用されるフィルタや、フィルタを適用したロールで使用される内部フィルタなど、どのフィルタ文字列にも仮想属性の名前を使用できます。Directory Server 5.2 では、VLV (仮想リスト表示) 操作やサーバー側ソート制御でも、実際の属性と同様に仮想属性がサポートされています。

CoS について

CoS は、CoS の適用範囲内にあるすべてのエントリ、すなわちターゲットエントリに対して、仮想属性とその値を定義します。各 CoS は、ディレクトリ内の次のエントリから構成されています。

- CoS 定義のエントリ : 使用中の CoS のタイプおよび生成される CoS 属性の名前を特定する。このエントリは、ロール定義のエントリと同様に、`LDAPsubentry` オブジェクトクラスから継承される。CoS の適用範囲は、CoS 定義のエントリの親の下のサブツリー全体である。同じ CoS 属性に複数の定義が存在する場合は、複数の値が含まれることがある
- テンプレートエントリ : 1 つ以上の仮想属性の値が含まれる。CoS の適用範囲内のすべてのエントリに、ここで定義された値が使用される。複数のテンプレートエントリがある場合は、生成された属性も複数の値を持つことがある

CoS には次の 3 つのタイプがあり、それぞれが CoS 定義のエントリとテンプレートエントリ間のさまざまな相互作用に対応しています。

- ポインタ CoS : CoS 定義のエントリは、テンプレート DN を使用してテンプレートエントリを直接識別する。すべてのターゲットエントリに、テンプレートで指定されているものと同じ CoS 属性値が設定される
- 間接 CoS : CoS 定義は、間接的な指示子と呼ばれる属性を識別する。ターゲットエントリのこの属性の値には、テンプレートの DN を指定する必要がある。間接 CoS を使うと、各ターゲットエントリで異なるテンプレートを使用できるため、CoS 属性に異なる値を指定できる
- クラシック CoS : CoS 定義は、テンプレートのベース DN と指示子 (ターゲットエントリの属性名) を識別する。指示子属性には RDN (相対識別名) を指定する必要がある。この RDN とテンプレートのベース DN との組み合わせによって、CoS 値を含むテンプレートが決定される

CoS 定義のエントリは、`cosSuperDefinition` オブジェクトクラスのインスタンスです。また、CoS のタイプを指定する、次のオブジェクトクラスのうちどれかから継承されます。

- `cosPointerDefinition`
- `cosIndirectDefinition`
- `cosClassicDefinition`

CoS 定義のエントリには、必要に応じて、仮想 CoS 属性、テンプレート DN、およびターゲットエントリの指示子属性を指定できるように、CoS のそれぞれのタイプに固有の属性が含まれています。デフォルトでは、CoS メカニズムは、CoS 属性と同じ名前を持つ既存の属性の値を上書きしません。ただし、CoS 定義のエントリの構文を使用すると、この処理を制御できます。

CoS テンプレートエントリは、cosTemplate オブジェクトクラスのインスタンスです。CoS テンプレートエントリには、CoS メカニズムによって生成された 1 つ以上の属性値があります。特定の CoS 用のテンプレートエントリは、その CoS 定義と同じレベルのディレクトリツリー内に格納されます。

管理を容易にするため、可能なかぎり、定義エントリ、およびテンプレートエントリを同じ場所に置いてください。また、それらが提供する機能を説明するような名前を付けてください。たとえば、定義エントリ DN に cn=ClCosGenerateEmployeeType,ou=People,dc=example,dc=com などの名前を付けると、cn=ClassicCos1,ou=People,dc=example,dc=com よりもわかりやすくなります。

『Sun ONE Directory Server Deployment Guide』の第 4 章にある「Managing Attributes with Class of Service」は、CoS の各タイプを詳細に説明し、例と導入上の注意点を記載しています。各 CoS タイプに関連するオブジェクトクラスと属性については、173 ページの「コマンド行からの CoS の管理」を参照してください。

CoS の制限事項

CoS 定義エントリとテンプレートエントリの作成と管理には、次のような制限事項があります。CoS 仮想属性の導入に関する制限事項の詳細については、『Sun ONE Directory Server Deployment Guide』の第 4 章にある「CoS Limitations」を参照してください。

CoS で生成された属性を使用する検索は、インデックスを使用しない検索となる: どの検索フィルタでも、仮想属性の有無を調べたり、その値を比較したりできます。ただし、仮想属性にはインデックスを付けることができないため、CoS で生成された属性をフィルタコンポーネントで使用するとインデックスを使用しない検索となり、パフォーマンスがかなり低下します。

サブツリーの制限: cn=config または cn=schema サブツリーでは、CoS 定義を作成できません。したがって、これらのエントリには仮想属性を含めることができません。

属性タイプの制限: 次の属性タイプは、同じ名前の実際の属性と動作が異なるため、CoS メカニズムでは生成しないでください。

- userPassword: CoS で生成されたパスワード値は、ディレクトリサーバーへのバインドに使用できない
- aci: Directory Server では、CoS によって定義された仮想 ACI 値の内容に基づいてアクセス制御を適用しない
- objectclass: Directory Server では、CoS によって定義された仮想オブジェクトクラスの値を検査するスキーマが実行されない

- `nsRoleDN` : CoS によって生成された `nsRoleDN` 値は、サーバーによるロールの生成に使用されない

サポートしていない属性サブタイプ。CoS メカニズムは、言語やバイナリなど、サブタイプを持つ属性を生成しません。

実際の属性値と仮想の属性値 : CoS メカニズムで複数値属性が生成される場合、エントリに定義されている「実際」の値と、CoS テンプレートで定義されている「仮想」の値とが混在することはありません。1 つの属性が持つ値は、エントリに格納されている値か、CoS メカニズムで生成された値のどちらかになります。詳細は、「実際の属性値の上書き」および 176 ページの「複数の値を持つ CoS 属性」を参照してください。

すべてのテンプレートをローカルに配置する必要がある : CoS 定義またはターゲットエントリの指示子に指定されているテンプレートエントリの DN は、Directory Server のローカルエントリを参照する必要があります。テンプレートとそこに含まれる値は、ディレクトリ連鎖またはリフェラルからは取得できません。

コンソールを使用した CoS の管理

ここでは、Directory Server コンソールを使った CoS 定義の作成および編集方法について説明します。

また、CoS 値を保護する必要がある場合は、CoS 定義エントリとテンプレートエントリ、およびターゲットエントリの指示子属性について ACI (アクセス制御命令) を定義してください。CoS セキュリティに関する注意事項については『Sun ONE Directory Server Deployment Guide』、コンソールを使用して ACI を作成する手順については本書の第 6 章「アクセス制御の管理」を参照してください。

新しい CoS の作成

ポインタ CoS およびクラシック CoS の場合は、定義エントリの前にテンプレートエントリを作成する必要があります。

1. Directory Server コンソールの最上位の「ディレクトリ」タブを選択し、ディレクトリツリーで、新しいテンプレートエントリの追加先エントリを右クリックして、ポップアップメニューから「新規」、「その他」を順に選択します。

   または、親エントリを選択し、「オブジェクト」メニューから「新規」、「その他」を順に選択します。

2. 「新規オブジェクト」ダイアログが表示されるので、オブジェクトクラスのリストから「`costemplate`」を選択します。「汎用エディタ」ダイアログボックスが表示され、新しいテンプレートのいくつかの属性にデフォルト値が表示されます。

3. 次の手順で新しいテンプレートオブジェクトを編集します。

a. objectclass 属性に ldapsubentry 値および extensibleobject 値を追加する

b. cn 属性を追加し、この属性にテンプレートを識別する値 (例 : cosTemplateForHeadquartersFax) を指定する

c. ネーミング属性を新しい cn 属性に変更する

ほかの属性を追加して、それをネーミング属性として使用することもできるが、通常は cn を使用する

d. 整数の値を設定することにより cosPriority 属性を変更するか、必要がない場合は優先順位属性を削除する。詳細は、177 ページの「CoS 属性の優先順位」を参照してください。

e. CoS メカニズムを使ってターゲットエントリに生成する属性とその値を追加する

4. 「汎用エディタ」ダイアログの「了解」をクリックして、テンプレートエントリを作成します。

5. このテンプレートにポインタ CoS を定義する場合は、ディレクトリツリーで新しいテンプレートエントリを選択し、メニューから「編集」>「DN のコピー」の順に選択します。

定義エントリの作成手順は、すべてのタイプの CoS の作成手順と同じです。

1. Directory Server コンソールの最上位の「ディレクトリ」タブを選択し、ディレクトリツリーで、新しい CoS 定義の追加先エントリを右クリックして、ポップアップメニューから「新規」、「サービスクラス」を順に選択します。

または、親エントリを選択し、「オブジェクト」メニューから「新規」、「サービスクラス」を順に選択します。

サービスクラスエントリのカスタムエディタが表示されます。

2. 新しいサービスクラスの名前と、必要に応じてその説明を入力します。CoS 定義のエントリの cn ネーミング属性に名前が表示されます。

3. 左側のリストで「属性」タブをクリックします。ダイアログに、CoS メカニズムによりターゲットエントリに生成される属性のリストが表示されます。

利用可能な属性のリストを表示し、属性をリストに追加するには、「追加」をクリックします。

4. リストに属性を追加すると、「サービスクラスの動作」列にドロップダウンリストが表示されます。このセルをクリックし、次の上書き動作のうちどれかを選択します。

- **ターゲットエントリ属性をオーバーライドしない** : ターゲットエントリの同じ属性に対応する属性値が格納されていない場合にだけ、CoS 属性値が生成される

- **ターゲットエントリ属性をオーバーライド**: CoS によって生成された属性値によって、ターゲットエントリ内の対応する属性値がすべて上書きされる
- **ターゲットエントリ属性をオーバーライド。(操作可能)**: 明示的に要求した場合を除きクライアントアプリケーションに表示されないようにするため、CoS 属性値をターゲットの値に上書きし、属性を operational にする

注	属性を operational にすることができるのは、その属性がスキーマ内でも operational と定義されている場合だけです。

5. 左側のリストで「テンプレート」タブをクリックします。テンプレートエントリの識別方法を選択し、対応するフィールドに必要事項を入力します。これにより、定義する CoS のタイプを決定できます。
   - **DN による**: これを選択すると、ポインタ CoS を定義できます。「テンプレート DN」フィールドにテンプレートエントリの DN を入力します。「参照」をクリックして、ディレクトリからテンプレート DN を選択するか、または Ctrl + V キーを押して、テンプレートエントリの作成後にコピーした DN をペーストします。
   - **ターゲットエントリの属性のうちの 1 つを使用する**: これを選択すると、間接 CoS を定義できます。「属性名」フィールドに指示子属性の名前を入力します。DN 値を含む属性を選択してください。リストから属性を選択するには、「変更」をクリックします。
   - **DN およびターゲットエントリの属性のうちの 1 つ、の両方を使用する**: これを選択すると、クラシック CoS を定義できます。テンプレートのベース DN と属性名の両方を入力します。「参照」をクリックして、ターゲットエントリの親エントリを選択します。次に「変更」をクリックして、リストから属性を選択します。
6. 「了解」をクリックして、CoS 定義のエントリを作成します。

既存の CoS の編集

1. Directory Server コンソールの最上位にある「ディレクトリ」タブで、CoS 定義のエントリをダブルクリックするか、そのエントリをマウスの右ボタンでクリックして、ポップアップメニューから「カスタムエディタで編集」を選択します。

   サービスクラスエントリのカスタムエディタが表示されます。
2. 必要に応じて名前と説明のフィールドを編集します。
3. CoS メカニズムによって生成される仮想属性を追加または削除するには、左側のリストで「属性」タブをクリックします。

4. テンプレートの指示子属性またはテンプレートエントリ DN の名前を定義し直すには、左側のリストで「テンプレート」タブをクリックします。このダイアログボックスを使うと、CoS 定義のタイプを定義し直すことができます。
5. 「了解」をクリックして、変更を保存します。

CoS の削除

1. Directory Server コンソールの最上位の「ディレクトリ」タブで、ディレクトリツリーを展開し、CoS 定義のエントリを選択します。
2. この CoS エントリをマウスの右ボタンでクリックし、ポップアップメニューから「削除」を選択します。削除の確認を求めるダイアログボックスが表示されます。「はい」をクリックします。

コマンド行からの CoS の管理

設定情報とテンプレートデータはすべてディレクトリ内にエントリとして格納されるので、LDAP コマンド行ツールを使って CoS 定義を設定、管理できます。ここでは、コマンド行を使用して CoS 定義エントリとテンプレートエントリを作成する方法について説明します。

また、CoS 値を保護する必要がある場合は、CoS 定義エントリとテンプレートエントリ、およびターゲットエントリの指示子属性について ACI (アクセス制御命令) を定義してください。コマンド行から ACI を作成する手順については、第 6 章「アクセス制御の管理」を参照してください。

コマンド行からの CoS 定義のエントリの作成

すべての CoS 定義のエントリは、`LDAPsubentry` オブジェクトクラスを持ち、`cosSuperDefinition` オブジェクトクラスから継承されます。さらに、CoS の各タイプは、特定のオブジェクトクラスから継承され、対応する属性を含みます。次の表に、各タイプの CoS 定義エントリ に関連付けられたオブジェクトクラスと属性を一覧表示します。

表 5-2 CoS 定義エントリのオブジェクトクラスと属性

CoS のタイプ	CoS 定義のエントリ

ポインタ CoS	`objectclass: top`
	`objectclass: LDAPsubentry`
	`objectclass: cosSuperDefinition`
	`objectclass: cosPointerDefinition`
	`cosTemplateDN:` *DN*
	`cosAttribute:` *attributeName override merge*

表 5-2　CoS 定義エントリのオブジェクトクラスと属性 (続き)

CoS のタイプ	CoS 定義のエントリ

間接 CoS	`objectclass: top`
	`objectclass: LDAPsubentry`
	`objectclass: cosSuperDefinition`
	`objectclass: cosIndirectDefinition`
	`cosIndirectSpecifier:` *attributeName*
	`cosAttribute:` *attributeName override merge*

クラシック CoS	`objectclass: top`
	`bbjectclass: LDAPsubentry`
	`objectclass: cosSuperDefinition`
	`objectclass: cosClassicDefinition`
	`cosTemplateDN:` *DN*
	`cosSpecifier:` *attributeName*
	`cosAttribute:` *attributeName override merge*

すべての `cosAttribute` は複数値を持ち、それぞれの値が CoS メカニズムによって生成される属性を定義します。

次の属性が CoS 定義のエントリ内で使用できます (属性については、『Sun ONE Directory Server Reference Manual』を参照)。

表 5-3　CoS 定義のエントリの属性

属性	CoS 定義のエントリ内の目的
`cosAttribute:` *attributeName override merge*	値を生成する対象となる仮想属性の名前を定義する。この属性には複数の値を指定できる。それぞれの値には属性の名前が指定され、この属性値はテンプレートから生成される。*override* 修飾子と *merge* 修飾子により、次の表に示す特殊な場合での CoS 属性値の算出方法を指定する *attributeName* にはサブタイプが含まれない可能性がある。サブタイプを持つ属性値は無視されるが、`cosAttribute` のその他の値は処理される
`cosIndirectSpecifier:` *attributeName*	ターゲットエントリの属性名を定義する。間接 CoS は、この属性の値を使ってテンプレートエントリを識別する。名前が指定された属性は指示子と呼ばれ、各ターゲットエントリに完全 DN 文字列を含める必要がある。この属性には値を 1 つしか指定できないが、指示子属性には複数の値を指定して複数のテンプレートを指定できる

表 5-3　CoS 定義のエントリの属性 (続き)

属性	CoS 定義のエントリ内の目的
cosSpecifier: *attributeName*	ターゲットエントリの属性名を定義する。クラシック CoS は、この属性の値を使ってテンプレートエントリを識別する。名前が指定された属性は指示子と呼ばれ、ターゲットエントリの RDN になる文字列を含める必要がある。この属性には値を 1 つしか指定できないが、指示子属性には複数の値を指定して複数のテンプレートを指定できる
cosTemplateDN: *DN*	ポインタ CoS 定義用にテンプレートエントリの完全 DN、またはクラシック CoS 用にテンプレートエントリのベース DN を指定する

cosAttribute 属性を使用すると、CoS 属性名のあとに修飾子を 2 つ付けることができます。*override* 修飾子では、次のいずれかの値を使用できます。

- default (または修飾子なし) : エントリに仮想属性と同じタイプの実際の属性が存在する場合、サーバーはエントリに格納されている実際の属性値を上書きしない
- override : 属性値がエントリとともに格納されている場合も含め、サーバーは常に CoS によって生成された値を返す
- operational : 検索要求内で明示的に属性が要求された場合にだけ、属性が返される。operational 属性の場合は、この属性を取得するために、スキーマ検査を渡す必要はない。override 修飾子と同じ動作もする

  属性を operational にすることができるのは、その属性がスキーマ内でも operational と定義されている場合だけです。たとえば、description 属性は、スキーマ内で operational としてマークされていないので、CoS を使用してこの属性の値を生成する場合は、operational 修飾子を使用できません。

merge 修飾子は指定しないか、または次の値を指定します。

- merge-schemes : 複数テンプレートまたは複数 CoS 定義から、仮想 CoS 属性に複数の値を指定できる。詳細は、176 ページの「複数の値を持つ CoS 属性」を参照してください。

実際の属性値の上書き

override 修飾子を含むポインタ CoS 定義のエントリの作成例を次に示します。

```
dn: cn=pointerCoS,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=exampleUS,cn=data
cosAttribute: postalCode override
```

このポインタ CoS 定義のエントリでは、このポインタ CoS が、postalCode 属性の値を生成するテンプレートエントリ cn=exampleUS,cn=data に関連付けられています。override 修飾子が指定されているので、この値がターゲットエントリに存在する場合は、その postalCode 属性値よりも、この値が優先されます。

注	CoS 属性に operational または override 修飾子を定義すると、CoS 適用範囲内のエントリでは、その属性の「実際」の値に対して書込み操作を行うことはできなくなります。

複数の値を持つ CoS 属性

merge-schemes 修飾子を指定すると、生成された CoS 属性に複数の値を指定できます。CoS 属性に複数の値を指定するには、次の 2 つの方法があります。

- 間接 CoS またはクラシック CoS では、ターゲットエントリの指示子属性に複数の値を指定できる。この場合、それぞれの値によってテンプレートが決定され、各テンプレートの値は生成された値の一部になる
- cosAttribute に同じ属性名を持つ任意のタイプの CoS 定義のエントリが複数存在することが可能である。この場合、すべての定義に merge-schemes 修飾子が含まれているときは、各定義によって算出されたすべての値が生成された属性に含まれる

2 つの状況が同時に発生したり、さらに多くの値を定義する場合もあります。ただし、どの場合でも、重複した値が生成された属性に返されるのは 1 度だけです。

merge-schemes 修飾子を指定しない場合は、次の節で説明するように、テンプレートエントリの cosPriority 属性を使用して、生成された属性のすべてのテンプレートの中から 1 つの値を決定します。

merge-schemes 修飾子は、ターゲットに定義された「実際」の値とテンプレートから生成された値をマージしません。*merge* 修飾子は、*override* 修飾子に依存しません。すべての組み合わせが可能で、それぞれの組み合わせが示す動作は有効です。また、修飾子は属性名のあとに任意の順序で指定できます。

注 同じ属性に複数の CoS 定義が存在する場合は、そのすべてに同じ *override* 修飾子および *merge* 修飾子を指定する必要があります。CoS 定義に指定された修飾子の組み合わせが異なる場合は、すべての定義から任意の 1 つの組み合わせが選択されます。

CoS 属性の優先順位

複数の CoS 定義または複数値を持つ指示子があるが、`merge-schemes` 修飾子が指定されていない場合、Directory Server では優先順位属性を使用して、仮想属性の 1 つの値を定義する 1 つのテンプレートを選択します。

`cosPriority` 属性は、対象となるすべてのテンプレートの中の特定のテンプレートのグローバルな優先順位を表します。優先順位 0 は、優先順位がもっとも高いことを示します。`cosPriority` 属性を含まないテンプレートは、もっとも優先順位が低いとみなされます。2 つ以上のテンプレートによって属性値が指定されているが、優先順位が同じまたは設定されていない場合は、任意の値が選択されます。

`merge-schemes` 修飾子を使用する場合は、テンプレートの優先順位は考慮されません。マージするときに、定義する優先順位に関係なく、対象となるすべてのテンプレートが値を定義します。次の節で説明するように、`cosPriority` 属性は CoS テンプレートエントリに対して定義されます。

注 `cosPriority` 属性は、負の値を持つ必要があります。また、間接 CoS が生成する属性は優先順位をサポートしていません。間接 CoS 定義のテンプレートエントリでは、`cosPriority` を使用しないでください。

コマンド行からの CoS テンプレートエントリの作成

ポインタ CoS またはクラシック CoS を使用するときは、`LDAPsubentry`、`cosTemplate` の各オブジェクトクラスがテンプレートエントリに含まれます。このエントリは、特に CoS 定義用に作成する必要があります。CoS テンプレートエントリを `LDAPsubentry` オブジェクトクラスのインスタンスにすることで、設定エントリの影響を受けずに、通常の検索を実行できるようになります。

間接 CoS メカニズムのテンプレートは、ディレクトリ内の任意の既存テンプレートエントリです。事前にターゲットを指定する必要はなく、`LDAPsubentry` オブジェクトクラスを指定する必要もありませんが、任意の `cosTemplate` オブジェクトクラスが含まれている必要があります。間接 CoS テンプレートには、CoS を評価して仮想属性とその値を生成する場合にだけアクセスします。

どのような場合でも CoS テンプレートエントリには、ターゲットエントリ上の CoS によって生成された属性と値を含める必要があります。属性名は、CoS 定義のエントリの cosAttribute 属性に指定されています。

次の例は、postalCode 属性を生成するポインタ CoS の優先順位がもっとも高いテンプレートエントリを示します。

```
dn: cn=ZipTemplate,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
postalCode: 95054
cosPriority: 0
```

次の節では、テンプレートエントリの例と CoS 定義のエントリの各タイプの例を紹介します。

ポインタ CoS の例

次のコマンドは、cosPointerDefinition オブジェクトクラスを持つ ポインタ CoS 定義エントリを作成します。この定義エントリは、上の CoS テンプレートエントリを使って、ou=People,dc=example,dc=com ツリーのすべてのエントリに共通する郵便番号を共有します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
```

```
dn: cn=pointerCoS,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=ZipTemplate,ou=People,dc=example,dc=com
cosAttribute: postalCode
```

ここで作成した CoS テンプレートエントリ cn=ZipTemplate,ou=People,dc=example,dc=com は、ou=People,dc=example,dc=com サフィックスの下に置かれているすべてのエントリに対して、その postalCode 属性に格納されている値を提供します。同じサブツリーで郵便番号を持たないエントリを検索すると、生成される属性の値は次のようになります。

```
ldapsearch -h host -p port -D "cn=Directory Manager" -w password ¥
-b "ou=People,dc=example,dc=com" -s sub "(cn=*Jensen)"
```

```
dn: cn=Babs Jensen,ou=People,dc=example,dc=com
cn: Babs Jensen
...
postalCode: 95054
```

間接 CoS の例

間接 CoS は、各ターゲットに固有のテンプレートを特定するために、cosIndirectSpecifier 属性内の属性に名前をつけます。ここで説明する間接 CoS は、ターゲットエントリの manager 属性を使用して、CoS テンプレートエントリを識別するものです。テンプレートエントリはマネージャのユーザーエントリで、生成する属性の値を含んでいる必要があります。

次のコマンドは、cosIndirectDefinition オブジェクトクラスを含む間接 CoS 定義エントリを作成します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
```

```
dn: cn=generateDeptNum,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosIndirectDefinition
cosIndirectSpecifier: manager
cosAttribute: departmentNumber
```

次に、テンプレートエントリに cosTemplate オブジェクトクラスを追加し、生成する属性が定義されていることを確認します。この例では、すべてのマネージャエントリはテンプレートです。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
```

```
dn: cn=Carla Fuentes,ou=People,dc=example,dc=com
changetype: modify
add: objectclass
objectclass: cosTemplate
-
add: departmentNumber
departmentNumber: 318842
```

この CoS では、manager 属性を含むターゲットエントリ (ou=People,dc=example,dc=com の下のエントリ) は、自動的にマネージャの部署番号を持ちます。departmentNumber 属性は、サーバー上に存在せず、ターゲットエントリの一部として返されるだけなので、ターゲットエントリの仮想属性となります。たとえば、Babs Jensen のマネージャを Carla Fuentes として定義した場合、このマネージャの部署番号は次のように表示されます。

```
ldapsearch -h host -p port -D "cn=Directory Manager" -w password ¥
-b "ou=People,dc=example,dc=com" -s sub "(cn=*Jensen)"
```

```
dn: cn=Babs Jensen,ou=People,dc=example,dc=com
cn: Babs Jensen
...
manager: cn=Carla Fuentes,ou=People,dc=example,dc=com
departmentNumber: 318842
```

クラシック CoS の例

この例は、クラシック CoS を使用して住所を生成する方法を示しています。生成される値は、CoS 定義内の cosTemplateDN と、ターゲットエントリ内の cosSpecifier 属性の値の組み合わせで、テンプレートエントリの形式で表示されます。次のコマンドは、cosClassicDefinition オブジェクトクラスを使用して定義エントリを作成します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
```

```
dn: cn=classicCoS,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosClassicDefinition
cosTemplateDn: ou=People,dc=example,dc=com
cosSpecifier: building
cosAttribute: postalAddress
```

同じコマンドを使って、各ビルの住所を持つテンプレートエントリを作成します。

```
dn: cn=B07,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
postalAddres: 7 Old Oak Street$Anytown, CA 95054
```

この CoS では、building 属性を含むターゲットエントリ (ou=People,dc=example,dc=com の下のエントリ) は、自動的に対応する住所を持ちます。CoS メカニズムは、RDN 内に指示子属性値を持つテンプレートエントリを検索します。この例では、Babs Jensen に B07 ビルが割り当てられていれば、住所は次のように表示されます。

```
ldapsearch -h host -p port -D "cn=Directory Manager" -w password ¥
-b "ou=People,dc=example,dc=com" -s sub "(cn=*Jensen)"
```

```
dn: cn=Babs Jensen,ou=People,dc=example,dc=com
cn: Babs Jensen
...
building: B07
postalAddres: 7 Old Oak Street$Anytown, CA 95054
```

ロールに基づく属性の作成

クラシック CoS スキーマとして、エントリが持つロールに基づいてエントリの属性値を生成するものも作成できます。たとえば、ロールに基づく属性を使用して、サーバーの検索制限をエントリごとに設定できます。

ロールに基づく属性を作成するには、クラシック CoS の CoS 定義のエントリ内で cosSpecifier として nsRole 属性を使用します。nsRole 属性には複数の値を指定できるので、複数の使用可能なテンプレートエントリを含む CoS スキーマを定義できます。使用するテンプレートエントリを明確に決定するには、cosPriority 属性を CoS テンプレートエントリに追加します。

たとえば、マネージャロールのメンバーであれば、標準のメールボックス容量の割り当てを超えて使用できるようにする CoS を作成できます。次のようなマネージャロールが存在するとします。

```
dn: cn=ManagerRole,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: ManagerRole
nsRoleFilter: (isManager=True)
Description: filtered role for managers
```

次のようなクラシック CoS 定義のエントリが作成されます。

```
dn: cn=generateManagerQuota,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectlass: cosClassicDefinition
cosTemplateDn: cn=managerCOS,dc=example,dc=com
cosSpecifier: nsRole
cosAttribute: mailboxquota override
```

CoS テンプレートの名前は、cosTemplateDn と、nsRole の値 (ロールの DN) の組み合わせである必要があります。次に例を示します。

```
dn:cn="cn=ManagerRole,ou=People,dc=example,dc=com",ou=People,
 dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectlass: cosTemplate
mailboxquota: 1000000
```

CoS テンプレートエントリは、mailboxquota 属性値を提供します。追加で指定した override 修飾子は、CoS がターゲットエントリ内にある既存のすべての mailboxquota 属性値を上書きするように指定します。ロールのメンバーであるターゲットエントリは、たとえば次のような、ロールと CoS が生成する仮想属性を持ちます。

```
ldapsearch -h host -p port -D "cn=Directory Manager" -w password ¥
-b "ou=People,dc=example,dc=com" -s sub "(cn=*Fuentes)"

dn: cn=Carla Fuentes,ou=People,dc=example,dc=com
cn: Carla Fuentes
isManager: TRUE
...
nsRole: cn=ManagerRole,ou=People,dc=example,dc=com
mailboxquota: 1000000
```

注 ロールエントリおよび CoS 定義のエントリは、適用範囲内に同じターゲットエントリを指定できるように、ディレクトリツリーの同じ位置に置く必要があります。CoS ターゲットエントリも、検索や管理を簡単に実行できるように、同じ位置に置く必要があります。

第 6 章

アクセス制御の管理

安全なディレクトリを作成する上で、ディレクトリの内容へのアクセスを制御することは最も重要です。この章では、ディレクトリに対してどのようなアクセス権をユーザーに許可するかを決定する ACI (アクセス制御命令) について説明します。Sun ONE Directory Server 5.2 には、特定のユーザーが特定のエントリに対して持っている実効権限を表示する機能があります。この機能を使用すると、複雑で強力なアクセス制御メカニズムを簡単に管理できます。

ディレクトリ導入の計画段階では、全体的なセキュリティポリシーとして利用できるアクセス制御戦略を定義する必要があります。アクセス制御戦略を計画するためのヒントについては、『Sun ONE Directory Server Deployment Guide』の第 7 章にある「Designing Access Control」を参照してください。

この章は、次の節で構成されています。

- アクセス制御の原則
- デフォルト ACI
- ACI の構文
- バインドルール
- コマンド行からの ACI の作成
- コンソールを使用した ACI の作成
- アクセス制御の使用例
- 実効権限の表示
- 高度なアクセス制御 : マクロ ACI の使用
- アクセス制御とレプリケーション
- アクセス制御情報のログ
- 以前のリリースとの互換性

アクセス制御の原則

アクセスを定義するためのメカニズムをアクセス制御と呼びます。サーバーが要求を受け取ると、バインド操作でユーザーが提供する認証情報、およびサーバー内で定義された ACI (アクセス制御命令) を使用して、ディレクトリ情報へのアクセスが許可または拒否されます。サーバーは、読み取り、書き込み、検索、比較などのアクセス権を許可または拒否できます。ユーザーに与えられるアクセス権のレベルは、そのユーザーの認証情報によって決まります。

アクセス制御を使用すると、ディレクトリ全体、ディレクトリのサブツリー、ディレクトリ内の特定エントリ (設定タスクを定義するエントリを含む)、エントリ属性の特別なセットなどに対するアクセスを制御できます。アクセス権は、特定ユーザー、特定のグループまたはロールに属するすべてのユーザー、またはそのディレクトリのすべてのユーザーに対して設定できます。また、IP アドレスや DNS 名などによって特定されるクライアントに対してもアクセス権を定義できます。

ACI の構造

ACI は、エントリの属性としてディレクトリ内に格納されます。`aci` 属性はオペレーショナル属性です。この属性は、そのエントリのオブジェクトクラス用に定義されたものであるかどうかにかかわらず、ディレクトリ内のすべてのエントリで使用できます。`aci` 属性は、ディレクトリサーバーがクライアントから LDAP 要求を受け取るときに、どのアクセス権が与えられ、どのアクセス権が拒否されるかを判定するために使用されます。`aci` 属性が `ldapsearch` 処理で返されるように指定できます。

ACI 文は 3 つの主要部分から構成されます。

- ターゲット - 権限が適用されるエントリまたは属性を決定します。
- 権限 - 許可または拒否される処理を定義します。
- バインドルール - バインド DN に基づいて、ACI の対象を定義します。

ACI のアクセス権およびバインドルール部分はペアで設定され、ACR (アクセス制御規則) とも呼ばれます。指定されたターゲットにアクセスする権限が与えられるか拒否されるかは、これに付随するルールが true であると判定されるかどうかによって決まります。詳細は、189 ページの「ACI の構文」を参照してください。

ACI の配置

ACI を含むエントリが子エントリを持たない場合は、ACI はそのエントリだけに適用されます。そのエントリが子エントリを持つ場合は、ACI はそのエントリと、そのエントリよりも下位にあるすべてのエントリに適用されます。結果的に、サーバーが任意のエントリに対するアクセス権を評価するときは、要求されたエントリとルートサフィックスのベースの間にあるすべてのエントリの ACI を確認します。

`aci` 属性には複数の値を設定できます。つまり、同じエントリまたは同じサブツリーに対して、複数の ACI を定義できます。

あるエントリに対して ACI を設定する場合は、そのエントリ自体には ACI を適用せず、そのエントリの下位にある一部またはすべてのエントリに対してだけ適用するように定義することもできます。このように ACI を定義すると、ディレクトリツリーの高いレベルに汎用的な ACI を置き、ツリーの下位に置かれる可能性の高いエントリに対してこの ACI を効果的に適用できます。たとえば、`organizationalUnit` エントリまたは `locality` エントリのレベルで、`inetorgperson` オブジェクトクラスを含むエントリをターゲットとする ACI を作成できます。

この機能を使用すると、汎用的なルールを分岐点のできるだけ高いレベルに置くことによって、ディレクトリツリー内の ACI の数を最小限にできます。より限定的なルールの適用範囲を制限するには、できるだけ最下位のエントリに近い位置にそのルールを置きます。

注 ルート DSE エントリ (`""` という DN を持ちます) に置かれた ACI は、そのエントリだけに適用されます。

ACI の評価

特定のエントリに対するアクセス権限を評価する場合は、サーバーによって、そのエントリ上と、エントリのルートサフィックスのベースにバックアップされる親エントリの ACI のリストが作成されます。評価中に、この順番でサーバーにより ACI が処理されます。ACI は、エントリとそのルートサフィックスのベースの間にあるすべてのサフィックスとサブサフィックスで評価され、他のサーバー上の連鎖サフィックスでは評価されません。

注 Directory Manager は、アクセス制御の制限を受けない権限を持つ唯一のユーザーです。クライアントが Directory Manager としてディレクトリにバインドされると、サーバーは処理の実行前に ACI を評価しません。

このため、Directory Manager としての LDAP 操作は、他のユーザーによる操作とは異なる結果を生じます。ディレクトリのパフォーマンスをテストするときは、常に一般的なユーザーとして実行する必要があります。

デフォルトでは、エントリにどの ACI も適用されない場合、Directory Manager 以外のすべてのユーザーはアクセスを拒否されます。ユーザーがサーバー上のエントリにアクセスするには、ACI によってアクセスが明示的に許可されている必要があります。デフォルト ACI は匿名の読み取りアクセス権を定義し、ユーザーによる各自のエントリの修正を許可します。ただし、セキュリティに必要な属性を変更することはできません。詳細は、188 ページの「デフォルト ACI」を参照してください。

サーバーは、ターゲットエントリに最も近い ACI から処理を開始しますが、エントリに適用されるすべての ACI が有効です。いずれかの ACI によって許可されるアクセスは、他の ACI が拒否しない場合に限り有効です。アクセスを拒否する ACI は、リストに含まれているかどうかに関係なく、同じリソースへのアクセスを許可する ACI に優先して適用されます。

たとえば、ディレクトリのルートレベルで書き込みアクセス権を拒否すると、ユーザーに特定のアクセス権を与えても、どのユーザーもディレクトリに書き込めなくなります。特定ユーザーにそのディレクトリへの書き込みアクセス権を与えるには、書き込みアクセス権の元の拒否対象を制限し、書き込みアクセス権を付与するユーザーを除外しておく必要があります。

ACI の制限事項

ディレクトリサービスに対するアクセス制御ポリシーを決定するときは、次の制限事項に注意してください。

- ディレクトリツリーが連鎖機能によって複数のサーバー上に分散されている場合は、アクセス制御文で使用できるキーワードにいくつかの制約がある
  - グループエントリ (`groupdn` キーワード) に依存する ACI は、グループエントリと同じサーバー上に置く必要がある。そのグループがダイナミックである場合は、そのメンバーすべても同じサーバー上にエントリを持つ必要がある。グループがスタティックである場合は、リモートサーバー上にメンバーのエントリを置くことができる
  - ロール定義 (`roledn` キーワード) に依存する ACI は、ロール定義エントリと同じサーバー上に置く必要がある。ロールを持たせる予定のエントリも、すべて同じサーバー上に置く必要がある

  ただし、ターゲットエントリに格納された値と、バインドユーザーのエントリに格納された値のマッチングは可能です (`userattr` キーワードなどを使用)。ACI を持つサーバー上にバインドユーザーがエントリを持っていない場合も、通常どおりにアクセスに対する評価が行われます。

  アクセス制御の評価を連続して行う方法については、113 ページの「連鎖サフィックスのアクセス制御」を参照してください。

- CoS によって作成された属性を、すべての ACI キーワードで使用できるわけではない。特に、アクセス制御規則が機能しないため、`userattr` および `userdnattr` キーワードによって CoS で作成した属性を使用しないこと。詳細は、207 ページの「userattr キーワードの使用」を参照してください。CoS については、第 5 章「高度なエントリの管理」を参照してください。
- アクセス制御規則の評価は、常にローカルサーバー上で行われる。このため、ACI キーワードで使用される LDAP URL で、サーバーのホスト名やポート番号を指定してはならない。指定しても、LDAP URL は無視される。詳細については、『Sun ONE Directory Server Reference Manual』の付録 D、「LDAP URLs」を参照。
- プロキシ権限を与える場合、ユーザーに Directory Manager となるプロキシ権限を与えたり、Directory Manager にプロキシ権限を与えたりすることはできない

デフォルト ACI

ディレクトリサーバーをインストールすると、設定時に指定したルートサフィックスに次のデフォルト ACI が定義されます。

- すべてのユーザーは、ディレクトリに匿名でアクセスして、検索、比較、および読み取り操作を行うことができる
- バインドユーザーは、ディレクトリ内にある個人のエントリを変更できるが、削除はできない。`aci`、`nsroledn`,、`passwordPolicySubentry` 属性を変更することはできず、各自のリソース制限属性、パスワードポリシー状態属性、アカウントロックアウト状態属性を変更することもできない
- 構成管理者 (デフォルトでは `uid=admin,ou=Administrators, ou=TopologyManagement,o=NetscapeRoot`) には、プロキシ権限以外のすべての権限が与えられる
- 構成管理者グループのすべてのメンバーには、プロキシ権限以外のすべての権限が与えられる
- ディレクトリ管理者グループのすべてのメンバーには、プロキシ権限以外のすべての権限が与えられる
- SIE グループのすべてのメンバーには、プロキシ権限以外のすべての権限が与えられる。SIE グループは、管理サーバー内の、このディレクトリのサーバーグループの管理者のグループである。

ディレクトリに新しいルートサフィックスを作成するたびに、ベースエントリに自己変更 ACI を除く上記デフォルト ACI が設定されます。セキュリティを強化するには、88 ページの「コンソールを使用した新しいルートサフィックスの作成」で説明している方法でこの ACI を追加します。

管理サーバー用の NetscapeRoot サブツリーには、専用のデフォルト ACI が置かれます。

- 構成管理者グループのすべてのメンバーには、プロキシ権限以外のすべての NetscapeRoot サブツリーでの権限が与えられる。これにより、このメンバーは設定管理者グループに新しいメンバーを追加できる
- すべてのユーザーは NetscapeRoot サブツリーに匿名でアクセスして、検索および読み取り操作を行うことができる
- グループ拡張 ACI は、管理グループのメンバーにグループ定義へのアクセスを許可する

次の節では、ユーザーの必要に応じてこれらのデフォルト設定を変更する方法を説明します。

ACI の構文

ACI は、バリエーションに富む複雑な構造をしています。ACI の作成と変更にコンソールを使うか、コマンド行を使うかに関係なく、LDIF 内の ACI の構文を理解しておくことは重要です。次の各項では、ACI の構文について詳細に説明します。

ヒント ACI はたいへん複雑であるため、Directory Server コンソールはすべての ACI の視覚的な編集には対応していません。ただし、多数のディレクトリエントリに対してアクセス制御を設定する場合は、コマンド行を使用するよりも時間を大幅に短縮できます。このため、効果的なアクセス制御が設定されたソースディレクトリを作成するには、ACI の構文を理解することが重要になります。

`aci` 属性の構文は次のとおりです。

```
aci: (target)(version 3.0;acl "name";permission  bindRules;)
```

各オプションは、次のように指定します。

- *target* は、アクセス制御の対象となるエントリ、属性、またはエントリと属性のセットを指定する。ターゲットには、識別名、1 つ以上の属性、または 1 つの LDAP フィルタを指定できる。ターゲットは省略できる。ターゲットを指定しないときは、そのエントリに定義されている ACI がエントリ全体とその子に対して適用される
- `version 3.0` は、ACI バージョンの識別に必要な文字列
- `"`*name*`"` は、ACI の名前。名前には、ACI を識別する任意の文字列を適用できる。ACI の名前は必須であり、その ACI の機能を説明するものが望ましい。
- *permission* は、許可または拒否する権限 (読み取り権限や検索権限など) を指定する
- *bindRules* は、ユーザーがアクセス権を許可されるために必要な資格およびバインドパラメータを指定する。バインドルールは、ユーザーまたはグループのメンバーシップ、またはクライアントの接続プロパティに基づいて指定することもできる

複数のターゲットと、権限とバインドルールのペアを利用できます。これにより、対象となるエントリと属性の両方を詳細に指定し、特定のターゲットに対して複数のアクセス制御を効率的に設定できます。たとえば、次のようにします。

```
aci: (target)...(target)(version 3.0;acl "name"; permission  bindRule;
 permission  bindRule; ...; permission  bindRule;)
```

LDIF ACI の例を次に示します。

```
aci: (target="ldap:///uid=bjensen,dc=example,dc=com")(targetattr=*)
 (version 3.0; acl "aci1"; allow (write) userdn="ldap:///self";)
```

この ACI では、bjensen というユーザーに対して、自身のディレクトリ内にあるすべての属性を変更できる書き込み権限を与えています。

次の節では、ACI の各部の構文について詳しく説明します。

ターゲットの定義

ターゲットは、ACI の適用対象を指定します。クライアントがエントリに含まれる属性に対する処理を要求すると、サーバーはターゲットを評価し、その処理の許可または拒否のために ACI の評価が必要であるかどうかを確認します。ターゲットを指定しないと、ACI は `aci` 属性を含むエントリ内のすべての属性、およびその下位のエントリに適用されます。

ターゲットの一般的な構文は、次のいずれかです。

(*keyword* = "*expression*")

(*keyword* != "*expression*")

各オプションは、次のように指定します。

- *keyword* は、ターゲットのタイプを示す。190 ページの表 6-1 のキーワードによって、次のタイプのターゲットが定義される
  - ディレクトリエントリ、またはそのサブツリー
  - エントリの属性
  - LDAP フィルタと一致するエントリまたは属性のセット
  - LDAP フィルタと一致する属性値、または値の組み合わせ
- 等号 (=) は、ターゲットが *expression* で指定されたオブジェクトであることを示し、不等号 (!=) は、ターゲットが *expression* で指定されたオブジェクトではないことを示す
- *expression* はキーワードに依存し、ターゲットを識別する。*expression* を囲む引用符 ("") は省略できない

次の表に、各キーワードとそれに対応する式を示します。

表 6-1　LDIF ターゲットキーワード

キーワード	有効な式	ワイルドカード使用の可否
`target`	ldap:///*distinguished_name*	可
`targetattr`	属性	可

表 6-1　LDIF ターゲットキーワード (続き)

キーワード	有効な式	ワイルドカード使用の可否
targetfilter	*LDAP_filter*	可
targattrfilters	*LDAP_operation:LDAP_filter*	可

ディレクトリエントリのターゲット指定

特定のディレクトリ、およびその下のエントリを指定するときは、target キーワードと、LDAP URL に含まれる DN を使います。ターゲット DN は、ACI が定義されるエントリの下のサブツリーに指定する必要があります。ターゲットは、次の構文で表記されます。

```
(target = "ldap:///distinguished_name")
(target != "ldap:///distinguished_name")
```

識別名は、ACI が定義されるエントリをルートとするサブツリーに指定する必要があります。たとえば、ou=People,dc=example,dc=com の ACI では、次のターゲットを使用します。

```
(target = "ldap:///uid=bjensen,ou=People,dc=example,dc=com")
```

注　アクセス制御規則を適用するエントリの DN にコンマが含まれる場合は、1 つの円記号 (¥) を使用して、コンマをエスケープする必要があります。たとえば、次のようにします。

```
(target="ldap:///uid=cfuentes,o=Example Bolivia¥, S.A.")
```

DN にワイルドカードを使用して、LDAP URL と一致する複数のエントリをターゲットにすることもできます。次に、ワイルドカードの正しい使用例を示します。

- `(target="ldap:///uid=*,dc=example,dc=com")`

  example.com ツリー内のエントリで、その RDN 内に uid 属性を含むすべてのエントリを示します。次の例に示すように、このターゲットは、ツリーの下のすべての階層のエントリと一致します。

  ```
  uid=tmorris,ou=sales,dc=example,dc=com
  uid=yyorgens,ou=marketing,dc=example,dc=com
  uid=bjensen,ou=eng,ou=east,dc=example,dc=com
  ```

- `(target="ldap:///uid=*Anderson,ou=People,dc=example,dc=com")`

  ou=People 分岐内で、uid が Anderson で終わるすべてのエントリと一致します。

- (target="ldap:///*Anderson,ou=People,dc=example,dc=com")

  ou=People 内で、ネーミング属性に関係なく RDN が Anderson で終わるすべてのエントリと一致します。

uid=*,ou=*,dc=example,dc=com のように、複数のワイルドカードを使用できます。この例は、識別名に uid と ou の属性だけを含む、example.com ツリーのすべてのエントリと一致します。

注 識別名のサフィックス部分には、ワイルドカードを使用できません。つまり、ディレクトリのサフィックスが c=US と c=GB である場合に、両方のサフィックスを参照させる次のようなターゲットは使用できません。

(target="ldap:///dc=example,c=*")

また、uid=bjensen,o=*.com のようなターゲットも使用できません。

属性のターゲット指定

ディレクトリエントリをターゲットとして指定できるだけでなく、ターゲットとして指定したエントリに含まれる 1 つ以上の属性 (または、1 つ以上の属性を除くすべての属性) をターゲットとすることもできます。これは、エントリに関する部分的な情報へのアクセスを許可または拒否するときに便利です。たとえば、あるエントリの共通名、名字、および電話番号の属性に限ってアクセスを限定できます。あるいは、個人データなど、取り扱いに注意を要する情報へのアクセスを一括して拒否することもできます。

ターゲットエントリ、またはそのサブツリーにターゲット属性が存在する必要はありません。ただし、指定されている ACI は常に適用されます。ターゲット属性は、スキーマで定義されている必要はありません。スキーマ検査が行われなければ、データとスキーマをインポートする前にアクセス制御ポリシーを実装できます。

属性をターゲットとして指定するには、targetattr キーワードを使用して、属性名を指定します。targetattr キーワードの構文は次のとおりです。

```
(targetattr = "attribute")
(targetattr != "attribute")
```

targetattr キーワードにより、複数の属性をターゲットとして指定できます。構文は次のとおりです。

```
(targetattr = "attribute1 || attribute2 ... || attributen")
(targetattr != "attribute1 || attribute2 ... || attributen")
```

たとえば、エントリの共通名、名字、および uid 属性をターゲットとして指定するには、次のように入力します。

```
(targetattr = "cn || sn || uid")
```

ターゲットに指定された属性には、名前が付けられた属性のすべてのサブタイプが含まれます。たとえば、(targetattr = "locality") と指定すると、locality;fr もターゲットに指定できます。また、(targetattr = "locality;fr;quebec") のように、サブタイプをターゲットに指定することもできます。

属性とエントリ両方によるターゲット指定

デフォルトでは、targetattr キーワードを含む ACI のターゲットに指定されたエントリに ACI が置かれます。

```
aci: (targetattr = "uid")(accessControlRules;)
```

という ACI を ou=Marketing,dc=example,dc=com エントリに置いた場合は、ACI は Marketing サブツリー全体に適用されます。ただし、次のように target キーワードを使用して、ターゲットを明示的に指定することもできます。

```
aci: (target="ldap:///uid=*,ou=Marketing,dc=example,dc=com")
 (targetattr="uid") (accessControlRules;)
```

target および targetattr キーワードを指定する順番は、特に重要ではありません。

LDAP フィルタを使用したエントリまたは属性のターゲット指定

LDAP フィルタを使用して、一定の基準に一致するエントリのセットをターゲットとして指定できます。このためには、LDAP フィルタとともに targetfilter キーワードを使用する必要があります。ACI を含むエントリの下のサブツリーに含まれるフィルタと一致するすべてのエントリに ACI が適用されます。

targetfilter キーワードの構文は次のとおりです。

```
(targetfilter = "LDAPfilter")
```

ここで、*LDAPfilter* は、標準的な LDAP 検索フィルタです。フィルター構文の詳細については、『Sun ONE Directory Server Getting Started Guide』の第 4 章にある「LDAP Search Filters」を参照してください。

たとえば、従業員を表すすべてのエントリに、正社員または契約社員の状態と、勤務時間数の全就業時間に対する割合を表す属性が設定されているとします。契約社員またはパート社員を表すすべてのエントリをターゲットとして指定するには、次のフィルタを使用できます。

```
(targetfilter = "(|(status=contractor)(fulltime<=79))")
```

注 ACI では、国際化値のマッチングルールに対応したフィルタ構文はサポートされていません。たとえば、次のターゲットフィルタは指定できません。

```
(targetfilter = "(locality:fr:=<= Quebec)")
```

ターゲットフィルタでは、ACI のターゲットとしてエントリ全体が選択されます。targetfilter および targetattr キーワードを組み合わせて、ターゲットエントリの属性のサブセットに適用される ACI を作成できます。

次の例に示す LDIF では、Engineering Admins グループのメンバーは、engineering 部門のすべてのエントリの departmentNumber 属性と manager 属性を変更できます。この例では、LDAP フィルタを使用して、businessCategory 属性の値が Engineering に設定されたすべてのエントリを選択しています。

```
dn: dc=example,dc=com
objectClass: top
objectClass: organization
aci: (targetattr="departmentNumber || manager")
 (targetfilter="(businessCategory=Engineering)")
 (version 3.0; acl "eng-admins-write"; allow (write)
 groupdn ="ldap:///cn=Engineering Admins, dc=example,dc=com";)
```

ヒント ディレクトリ内に分散したエントリおよび属性をターゲットとして指定する場合に LDAP フィルタを使用すると便利ですが、アクセス管理の対象となるオブジェクトを直接フィルタが指定するわけではないため、思わぬ結果を招くことがあります。フィルタを適用した ACI のターゲットとなるエントリセットは、属性の追加や削除に応じて変化することがあります。したがって、ACI で LDAP フィルタを使用する場合は、ldapsearch 操作で同じフィルタを使用して、適切なエントリと属性がターゲットとして指定されていることを確認する必要があります。

LDAP フィルタを使用した属性値のターゲット指定

アクセス制御を使用すると、特定の属性値をターゲットとして指定できます。つまり、属性値と ACI 内で定義された基準が一致する場合は、その属性に対するアクセス権を許可または拒否できます。属性値に基づいてアクセスを許可または拒否する ACI は、値に基づく ACI と呼ばれます。

たとえば、組織内のすべてのユーザーに、ユーザー自身のエントリ内の nsRoleDN 属性を変更できるアクセス権を与えるとします。ただし、同時に、これらのユーザーが、自身に対して「最上位レベルの管理者」のような重要なロールを割り当てることができないようにします。LDAP フィルタは、このような場合に属性値の条件が満たされているかどうかを確認するために使用されます。

値に基づく ACI を作成するには、targattrfilters キーワードを使用する必要があります。構文は次のとおりです。

```
(targattrfilters="add=attr1:F1 && attr2:F2... && attrn:Fn,
                  del=attr1:F1 && attr2:F2 ... && attrn:Fn")
```

各オプションは、次のように指定します。

- `add` は、属性を作成する操作を示す
- `del` は、属性を削除する操作を示す
- *attrn* は、ターゲットの属性を示す
- *Fn* は、対応する属性だけに適用されるフィルタを示す

エントリを作成するときに、新しいエントリ内の属性に対してフィルタを適用する場合は、その属性の各インスタンスはすべてフィルタの条件を満たす必要があります。エントリを削除するときに、エントリ内の属性に対してフィルタを適用する場合も、その属性の各インスタンスはすべてフィルタの条件を満たす必要があります。

エントリを変更するときに、属性を追加する場合は、その属性に適用される追加フィルタの条件を満たす必要があります。属性を削除する場合は、その属性に適用される削除フィルタの条件を満たす必要があります。すでにエントリ内にある属性の個々の値を置き換える場合は、追加フィルタと削除フィルタの両方の条件を満たす必要があります。

たとえば、次のような属性フィルタがあるとします。

```
(targattrfilters="add=nsroleDN:(!(nsRoleDN=cn=superAdmin)) &&
telephoneNumber:(telephoneNumber=123*)")
```

このフィルタを使用すると、ユーザーは、個人のエントリに `superAdmin` 以外の任意のロール (`nsRoleDN` 属性) を追加できます。また、先頭に 123 が付く電話番号を追加することもできます。

注 サーバーコンソールを使用して値に基づく ACI を作成することはできません。

単一のディレクトリエントリのターゲット指定

単一のエントリを明示的にターゲットとすることはできません。ただし、次の方法で指定することは可能です。

- ターゲットエントリ内に格納された属性値を使用して、バインド要求時のユーザー入力に一致するバインドルールを作成する。詳細は、207 ページの「値マッチングに基づくアクセスの定義」を参照してください。
- `targetfilter` キーワードを使用する

`targetfilter` キーワードを使うことで、目的のエントリだけに含まれる属性値を指定できます。たとえば、ディレクトリサーバーをインストールすると、次の ACI が作成されます。

```
aci: (targetattr="*")(targetfilter=(o=NetscapeRoot))(version 3.0;
 acl "Default anonymous access"; allow (read, search)
 userdn="ldap:///anyone";)
```

o 属性の値が NetscapeRoot のエントリは o=NetscapeRoot だけなので、ACI はこのエントリだけに適用されます。

これらの方法に関する問題点は、今後ディレクトリツリーを変更するときに、この ACI の変更が必要なことを憶えておき、手動で変更しなければならないことです。

アクセス権の定義

アクセス権は、許可または拒否するアクセスのタイプを指定します。ディレクトリ内で特定の操作を実行するためのアクセス権を許可または拒否できます。割り当てることのできる各操作は、アクセス権と呼ばれます。

アクセス権の設定には、2 つの手順が必要です。

- アクセスの許可または拒否
- 権限の割り当て

アクセスの許可または拒否

ディレクトリツリーに対するアクセス権は、明示的に許可または拒否できます。いつアクセスを許可し、いつアクセスを拒否するかについて、詳しいガイドラインは『Sun ONE Directory Server Deployment Guide』の第 7 章にある「Designing Access Control」を参照してください。

注	サーバーコンソールを使用して明示的にアクセスを拒否することはできませんが、アクセス権を与えることはできます。

権限の割り当て

権限は、ディレクトリデータに対してユーザーが実行できる特定の操作を詳細に定義します。すべての権限を許可または拒否するか、次に示す 1 つ以上の権限を割り当てることができます。

読み取り : ユーザーがディレクトリデータを読み込めるかどうかを示します。このアクセス権は、検索操作だけに適用されます。

書き込み : ユーザーが属性を追加、変更、または削除することによって、エントリを変更できるかどうかを示します。このアクセス権は、変更および modrdn 操作に適用されます。

追加 : ユーザーがエントリを追加できるかどうかを示します。このアクセス権は、追加操作だけに適用されます。

削除 : ユーザーがエントリを削除できるかどうかを示します。このアクセス権は、削除操作だけに適用されます。

検索 : ユーザーがディレクトリデータを検索できるかどうかを示します。ユーザーが検索結果の一部として返されたデータを参照するには、検索権限および読み取り権限が必要です。このアクセス権は、検索操作だけに適用されます。

比較 : ユーザーが入力したデータと、ディレクトリに格納されているデータを比較できるかどうかを示します。比較権限を持っている場合は、照会に対して成功または失敗を示すメッセージが返されますが、エントリまたは属性の値を表示することはできません。このアクセス権は、比較操作だけに適用されます。

本人による書き込み : ユーザーが、ターゲットエントリの属性に含まれる本人の DN を追加または削除できるかどうかを示します。このアクセス権は、グループ管理専用です。「本人による書き込み」は、プロキシ承認で使用できます。グループエントリからプロキシ DN を追加または削除するアクセス権を与えます (バインドユーザーの DN とは異なる)。

プロキシ承認 : 指定された DN が、他のエントリの権限でターゲットにアクセスできるかどうかを示します。Directory Manager DN を除く、ディレクトリ内の任意のユーザーの DN を使用して、プロキシ権限を与えることができます。なお、Directory Manager には、プロキシ権限を与えることはできません。参考例については、242 ページの「プロキシ承認を使用した ACI の例」を参照してください。プロキシ権限の概要については、『Sun ONE Directory Server Deployment Guide』を参照してください。

すべて : 指定された DN が、ターゲットエントリに対して、プロキシ権限以外のすべての権限 (読み取り、書き込み、検索、削除、比較、本人による書き込み) を持つことを示します。

これらの権限は個別に与えられます。たとえば、追加権限を与えられたユーザーがエントリを作成しても、そのユーザーに削除権限が与えられていなければ、そのエントリを削除できません。したがって、ディレクトリのアクセス制御ポリシーを決定するときは、ユーザーに対して理にかなった権限を与える必要があります。たとえば、読み取りおよび検索アクセス権を与えずに書き込みアクセス権だけを与えても、通常は意味がありません。

LDAP 操作に必要な権限

この節では、ユーザーに許可する LDAP 操作のタイプに応じて、そのユーザーに与える必要がある権限について説明します。

エントリを追加する場合

- 追加されるエントリに対する追加アクセス権
- エントリ内の各属性値に対する書き込みアクセス権。このアクセス権はデフォルトで与えられているが、`targattrfilters` キーワードを使用して制限できる

エントリを削除する場合

- 削除されるエントリに対する削除アクセス権
- エントリ内の各属性値に対する書き込みアクセス権。このアクセス権はデフォルトで与えられているが、targattrfilters キーワードを使用して制限できる

エントリの属性を変更する場合

- 目的の属性タイプに対する書き込みアクセス権
- 各属性タイプの値に対する書き込みアクセス権。このアクセス権はデフォルトで与えられているが、targattrfilters キーワードを使用して制限できる

エントリの RDN を変更する場合

- そのエントリに対する書き込みアクセス権
- 新しい RDN で使用される属性タイプに対する書き込みアクセス権
- 古い RDN を削除する書き込みアクセス権を与える場合は、古い RDN 属性タイプに対する書き込みアクセス権
- 新しい RDN で使用される属性タイプの値に対する書き込みアクセス権。このアクセス権はデフォルトで与えられているが、targattrfilters キーワードを使用して制限できる

属性値を比較する場合

- 目的の属性タイプに対する比較アクセス権

エントリを検索する場合

- 検索フィルタで使用される各属性タイプに対する検索アクセス権
- エントリで使用される属性タイプに対する読み取りアクセス権

ユーザーにディレクトリを検索させるために設定する必要があるアクセス権について理解するには、次の例を参照してください。次のような ldapsearch 操作を実行するとします。

```
% ldapsearch -h host -s suffix -b "uid=bjensen,dc=example,dc=com" ¥
              objectclass=* mail
```

bkolics というユーザーにアクセス権を与えるかどうかは、次に示す ACI を使用して決定します。

```
aci: (targetattr = "mail")(version 3.0; acl "self access to mail";
 allow (read, search) userdn = "ldap:///self";)
```

この ACI は objectclass 属性へのアクセス権を与えないので、検索結果のリストには何も表示されません。前述した検索操作を正常に実行するには、次のように ACI を変更する必要があります。

```
aci: (targetattr = "mail || objectclass")(version 3.0; acl "self
 access to mail"; allow (read, search) userdn = "ldap:///self";)
```

アクセス権の構文

ACI 文におけるアクセス権の構文は、次のとおりです。

`allow|deny (`*rights*`)`

ここで、*rights* は 1 ～ 8 個のキーワードのリストです。キーワードは、コンマで区切り、カッコでくくります。使用できるキーワードは、**read**、**write**、**add**、**delete**、**search**、**compare**、**selfwrite**、**proxy**、または **all** です。

次の例では、バインドルールが true であると判定された場合は、読み取り、検索、および比較アクセスが許可されます。

```
aci:  (target="ldap:///dc=example,dc=com") (version 3.0;acl
"example";
 allow (read, search, compare) bindRule;)
```

バインドルール

ディレクトリに対して定義された ACI に応じて、一部の操作では、ディレクトリに対するバインドが必要です。バインドとは、バインド DN とパスワードを入力して、ディレクトリに対して、ログインまたは自身の認証を行うことです。SSL を使用する場合は、証明書が必要です。ディレクトリに対するアクセスが許可されるか拒否されるかは、バインド操作で与えられる資格とバインドの状況によって決まります。

ACI 内のすべてのアクセス権のセットには、対応するバインドルールが存在します。

バインドルールは単純なものです。たとえば、バインドルールで、ディレクトリにアクセスするユーザーが特定のグループに属している必要があることだけを指定できます。また、より複雑なバインドルールを設定することもできます。たとえば、バインドルールで、ユーザーが特定のグループに属し、特定の IP アドレスを持つコンピュータで、午前 8 時から午後 5 時の間にログインする必要があることを指定できます。

バインドルールにより、誰が、いつ、どこからディレクトリにアクセスできるかを定義できます。具体的には、バインドルールで次の内容を指定できます。

- アクセス権が与えられたユーザー、グループ、およびロール
- エンティティがバインドを開始する位置
- バインドを実行できる時刻または日付
- バインド時に使用する認証のタイプ

さらに、ブール演算子を使用してこれらの条件を組み合わせると、複雑なバインドルールを定義できます。詳細は、217 ページの「ブール型バインドルールの使用」を参照してください。

サーバーでは、LDAP フィルタの評価で使用したものと似た 3 値論理に従って、ACI で使用する論理式が評価されます (「RFC 2251 Lightweight Directory Access Protocol (v3)」を参照)。つまり、式の構成要素が未定義と評価された場合 (リソースの制約によって、式の評価が中断された場合など) は、サーバーは適切に対応します。複雑なブール式で未定義値が発生しても、間違ってアクセス権を与えることはありません。

バインドルールの構文

アクセスが許可されるか拒否されるかは、ACI のバインドルールが true であると判定されるかどうかによって決まります。バインドルールには、次の 2 つのパターンのうちどちらかが使用されます。

keyword = "*expression*";

keyword != "*expression*";

等号 (=) は、バインドルールを true とするには *keyword* と *expression* が一致しなければならないことを示し、不等号 (!=) は、バインドルールを true とするには *keyword* と *expression* が一致してはならないことを示します。

注 timeofday キーワードでは、不等式表現 (<、<=、>、>=) もサポートしています。timeofday は、これらの表現をサポートする唯一のキーワードです。

expression の両側の引用符 ("") と区切りを示すセミコロン (;) は省略できません。使用できる式は、対応する *keyword* によって決まります。

次の表に、各キーワードとそれに対応する式を示します。式でワイルドカードが使用できるかどうかについても示します。

表 6-2 LDIF バインドルールキーワード

キーワード	有効な式	ワイルドカード使用
userdn	ldap:///*distinguished_name* ldap:///all ldap:///anyone ldap:///self ldap:///parent ldap:///*suffix*??sub?(*filter*)	可 (ただし DN に限る)

表 6-2　LDIF バインドルールキーワード (続き)

キーワード	有効な式	ワイルドカード使用
`groupdn`	ldap:///*DN* | | *DN*	不可
roledn	ldap:///*DN* | | *DN*	不可
`userattr`	*attribute#bindType* or *attribute# value*	不可
`ip`	*IP_address*	可
`dns`	*DNS_host_name*	可
`dayofweek`	sun mon tue wed thu fri sat	不可
`timeofday`	0 - 2359	不可
`authmethod`	none simple ssl sasl *authentication_method*	不可

次の節では、各キーワードのバインドルールの構文を詳しく説明します。

ユーザーアクセスの定義 : userdn キーワード

ユーザーアクセスは userdn キーワードを使用して定義します。userdn キーワードには、1 つ以上の有効な識別名が必要です。書式は次のとおりです。

```
userdn = "ldap:///dn [|| ldap:///dn]...[||ldap:///dn]"
```

ここで、*dn* は DN、つまり anyone、all、self、parent の 1 つです。これらの式は、次のユーザーを示します。

- `userdn = "ldap:///anyone"` : 匿名ユーザーと認証ユーザーの両方
- `userdn = "ldap:///all"` : 認証ユーザーのみ
- `userdn = "ldap:///self"` : ACI のターゲットエントリと同じユーザーのみ
- `userdn = "ldap:///parent"` : ACI ターゲットの親エントリのみ

userdn キーワードは、LDAP フィルタとして表すこともできます。書式は次のとおりです。

`ldap:///`*suffix*`??sub?(`*filter*`)`

注	DN にコンマが含まれる場合は、コンマの前にエスケープ文字の円記号 (¥) を付けて区別する必要があります。

匿名アクセス (anyone キーワード)

ディレクトリへの匿名アクセス権を与えると、バインド状況にかかわりなく、バインド DN やパスワードなしで、誰でもそのディレクトリにアクセスできます。匿名アクセスは、特定タイプのアクセス (たとえば、読み取りのためのアクセスや検索のためのアクセス)、あるいは特定のサブツリーやディレクトリ内の個々のエントリに、アクセスの対象を制限できます。anyone キーワードを使った匿名アクセスは、すべての認証ユーザーによるアクセスも許可します。

汎用アクセス (all キーワード)

バインドルールを使用すると、ディレクトリに対して正常にバインドしたすべてのユーザーに対してアクセス権を許可できます。このため、all キーワードは、すべての認証ユーザーによるアクセスを許可します。これは、匿名アクセスを防ぐ一方、汎用アクセスを許可します。

自己アクセス (self キーワード)

ユーザー自身が所有するエントリに対して、アクセス権を許可または拒否します。つまり、バインド DN がターゲットエントリの DN と一致するかどうかで、エントリへのアクセス権が許可または拒否されます。

親アクセス (parent キーワード)

ユーザーのバインド DN がターゲットエントリの親である場合に限り、ユーザーはエントリに対するアクセスを許可または拒否されます。parent キーワードを使うには、サーバーコンソールで ACI を手動で編集する必要があります。

LDAP URL

フィルタ付きの URL を使用すると、次のように ACI 内でダイナミックにターゲットユーザーを指定できます。

```
userdn = "ldap:///<suffix>??sub?(filter)"
```

たとえば、example.com ツリーの accounting および engineering の分岐に含まれる、すべてのユーザーのターゲットリソースに対するアクセスを、次の URL に基づいてダイナミックに許可または拒否できます。

```
userdn = "ldap:///dc=example,dc=com??sub?(|(ou=engineering)(ou=accounting))"
```

注	LDAP URL では、ホスト名またはポート番号を指定しないでください。LDAP URL は、常にローカルサーバーに適用されます。

LDAP URL については、『Sun ONE Directory Server Getting Started Guide』の該当する章を参照してください。

ワイルドカード

ワイルドカード文字 (*) を使用して、ユーザーのセットを指定することもできます。たとえば、uid=u*,dc=example,dc=com というユーザー DN を指定すると、設定したアクセス権に基づいて、u という文字で始まるバインド DN を持つユーザーのアクセスだけを許可または拒否できます。

ユーザーアクセスは、アクセス制御エディタを使用してサーバーコンソールから設定します。詳細は、219 ページの「コンソールを使用した ACI の作成」を参照してください。

例

この節では、userdn 構文の例を示します。

LDAP URL を含む userdn キーワード

```
userdn = "ldap:///uid=*,dc=example,dc=com";
```

ユーザーが、指定されたパターンの任意の識別名を使用してディレクトリにバインドすると、バインドルールは true であると判定されます。たとえば、次に示すバインド DN は、両方とも true と判定されます。

```
uid=ssarette,dc=example,dc=com
uid=tjaz,ou=Accounting,dc=example,dc=com
```

一方、次に示すバインド DN は、false と判定されます。

```
cn=Babs Jensen,dc=example,dc=com
```

LDAP URL の論理 OR を含む userdn キーワード

```
userdn="ldap:///uid=bj,c=example.com ||
ldap:///uid=kc,dc=example,dc=com";
```

クライアントが、与えられた 2 つの識別名のどちらかとしてバインドすると、バインドルールは true と判定されます。

特定の LDAP URL を含まない userdn キーワード

```
userdn != "ldap:///uid=*,ou=Accounting,dc=example,dc=com";
```

accounting サブツリーで、クライアントが UID を基にした識別名としてバインドしない場合に、バインドルールは true と判定されます。このバインドルールは、ターゲットエントリがディレクトリツリーの accounting 分岐の下にはない場合に限り意味を持ちます。

self キーワードを含む userdn キーワード

```
userdn = "ldap:///self";
```

ユーザーが、ディレクトリにバインドするための DN で表されるエントリにアクセスすれば、バインドルールは true と判定されます。つまり、ユーザーが `uid=ssarette,dc=example,dc=com` としてバインドし、`uid=ssarette,dc=example,dc=com` エントリで操作を試行すれば、バインドルールは true と判定されます。

たとえば、example.com ツリー内のすべてのユーザーに対して、`userPassword` 属性への書き込みアクセス権を与える場合は、`dc=example,dc=com` ノード上に次の ACI を作成します。

```
aci: (targetattr = "userPassword") (version 3.0;
 acl "write-self"; allow (write) userdn = "ldap:///self";)
```

all キーワードを含む Userdn キーワード

```
userdn = "ldap:///all";
```

バインド DN が有効なものであれば、バインドルールは true であると判定されます。true と判定されるには、バインド操作中にユーザーが有効な識別名とパスワードを入力する必要があります。

たとえば、認証されたすべてのユーザーに対してツリー全体の読み取りアクセス権を与える場合は、次に示す ACI を dc=example,dc=com ノードに作成します。

```
aci: (version 3.0; acl "all-read"; allow (read)
 userdn="ldap:///all";)
```

anyone キーワードを含む userdn キーワード

```
userdn = "ldap:///anyone";
```

すべてのユーザーに対して、バインドルールは true と判定されます。ディレクトリへの匿名アクセスを許可する場合は、このキーワードを使用します。

たとえば、example.com ツリー全体への匿名による読み取りアクセスと検索アクセスを許可する場合は、次に示す ACI を dc=example,dc=com ノードに作成します。

```
aci: (version 3.0; acl "anonymous-read-search";
 allow (read, search) userdn = "ldap:///anyone";)
```

parent キーワードを含む userdn キーワード

```
userdn = "ldap:///parent";
```

バインド DN がターゲットエントリの親であれば、バインドルールは true と判定されます。

たとえば、すべてのユーザーの子エントリに書き込みアクセス権を与える場合は、次に示す ACI を dc=example,dc=com ノードに作成します。

```
aci: (version 3.0; acl "parent access";
 allow (write) userdn="ldap:///parent";)
```

ユーザーが engineering または sales サブツリーに属していれば、バインドルールは true と判定されます。

グループアクセスの定義 : groupdn キーワード

指定されたグループのメンバーは、ターゲットリソースにアクセスできます。これは、グループアクセスと呼ばれます。グループアクセスは groupdn キーワードを使用して定義され、ユーザーが特定のグループに属する DN を使用してバインドすれば、ターゲットエントリへのアクセスが許可または拒否されます。

groupdn キーワードには、1 つ以上の有効な識別名が必要です。書式は次のとおりです。

```
groupdn="ldap:///dn [|| ldap:///dn]...[|| ldap:///dn]"
```

バインド DN が指定されたグループに属していれば、バインドルールは true と判定されます。

注	DN にコンマが含まれる場合、1 つの円記号 (¥) を使用してコンマをエスケープする必要があります。

特定のグループの定義には、サーバーコンソールとしてアクセス制御エディタを使用します。詳細は、219 ページの「コンソールを使用した ACI の作成」を参照してください。

例

この節では、groupdn 構文の例を示します。

LDAP URL を含む groupdn キーワード

```
groupdn = "ldap:///cn=Administrators,dc=example,dc=com";
```

バインド DN が Administrators グループに属していれば、バインドルールは true と判定されます。Administrators グループに対してディレクトリツリー全体への書き込みアクセス権を与える場合は、次の ACI を dc=example,dc=com ノードに作成します。

```
aci: (version 3.0; acl "Administrators-write"; allow (write)
 groupdn="ldap:///cn=Administrators,dc=example,dc=com";)
```

LDAP URL の論理 OR を含む groupdn キーワード

```
groupdn = "ldap:///cn=Administrators,dc=example,dc=com ||
ldap:///cn=Mail Administrators,dc=example,dc=com";
```

バインド DN が Administrators グループまたは Mail Administrators グループに属していれば、バインドルールは true と判定されます。

ロールアクセスの定義 : roledn キーワード

指定されたロールのメンバーは、ターゲットリソースにアクセスできます。これは、ロールアクセス と呼ばれます。ロールアクセスは roledn キーワードを使用して定義され、ユーザーが特定のロールに属する DN を使用してバインドすれば、ターゲットエントリへのアクセスが許可または拒否されます。

roledn キーワードには、1 つ以上の有効な識別名が必要です。書式は次のとおりです。

```
roledn = "ldap:///dn [|| ldap:///dn]... [|| ldap:///dn]"
```

バインド DN が指定されたロールに属していれば、バインドルールは true と判定されます。

注	DN にコンマが含まれる場合、1 つの円記号 (¥) を使用してコンマをエスケープする必要があります。

roledn キーワードの構文と使い方は、groupdn キーワードと同じです。

値マッチングに基づくアクセスの定義

バインドルールを設定することによって、ディレクトリへのバインドに使用するエントリの属性値が、ターゲットエントリの属性値に一致するように指定できます。

たとえば、ACI が適用されるように、バインド DN がユーザーエントリの manager 属性の DN に一致するように指定できます。この場合は、ユーザーのマネージャだけがエントリにアクセスできます。

この例は、DN マッチングに基づいています。したがって、ターゲットエントリとのバインドに使用されるエントリの任意の属性を一致させることができます。たとえば、favoriteDrink 属性に「Beer」という値を持つユーザーに対し、同じ値の favoriteDrink 属性を持つほかのユーザーのすべてのエントリの読み取りを許可する ACI を作成できます。

userattr キーワードの使用

userattr キーワードは、バインド操作に使用するエントリとターゲットエントリの間で、どの属性値が一致する必要があるかを指定するのに使用できます。

次のタイプを指定できます。

- ユーザー DN
- グループ DN
- ロール DN
- LDAP フィルタ (LDAP URL 内)
- 任意の属性タイプ

userattr キーワードの LDIF 構文は次のとおりです。

```
userattr = "attrName#bindType"
```

ユーザー DN、グループ DN、ロール DN、または LDAP フィルタ以外の値が必要な属性タイプを使用する場合は、次の構文になります。

```
userattr = "attrName#attrValue"
```

各オプションは、次のように指定します。

- *attrName* は、値マッチングに使用される属性の名前
- *bindType* は、USERDN,GROUPDN,LDAPURL の 1 つ
- *attrValue* は、属性値を表す任意の文字列

注 CoS (サービスクラス) 定義で作成された属性は、userattr キーワードと一緒に使用できません。Cos によって作成された属性値に依存するバインドルールを含む ACI は機能しません。

次の節では、考えられるさまざまなバインドタイプを指定した userattr キーワードの例を示します。

USERDN バインドタイプを指定した例

次に、ユーザー DN に基づくバインドに関連する userattr キーワードの例を示します。

```
userattr = "manager#USERDN"
```

バインド DN がターゲットエントリの manager 属性値と一致すれば、バインドルールは true と判定されます。これによって、ユーザーのマネージャが社員の属性を変更できるようになります。このメカニズムは、ターゲットエントリの manager 属性が、絶対 DN として指定されている場合にだけ機能します。

次の例では、マネージャは社員のエントリに対してすべてのアクセス権が許可されています。

```
aci: (target="ldap:///dc=example,dc=com")(targetattr=*) (version
3.0;
 acl "manager-write"; allow (all) userattr = "manager#USERDN";)
```

GROUPDN バインドタイプを指定した例

次に、グループ DN に基づくバインドに関連する userattr キーワードの例を示します。

```
userattr = "owner#GROUPDN"
```

バインド DN がターゲットエントリの owner 属性で指定されたグループのメンバーであれば、バインドルールは true と判定されます。たとえば、このメカニズムを使用して、社員の役職に関する情報の管理アクセス権を、あるグループに許可できます。使用する属性にグループエントリの DN が含まれていれば、owner 以外の属性も使用できます。

指定するグループをダイナミックグループにすることも、グループの DN をディレクトリ内の任意のサフィックスの下に置くこともできます。ただし、サーバーでこのタイプの ACI を評価するには、多くのリソースを必要とします。

ターゲットエントリと同じサフィックスの下にあるスタティックグループを使用する場合は、次の式を使用します。

```
userattr = "ldap:///dc=example,dc=com?owner#GROUPDN"
```

この例では、グループエントリは `dc=example,dc=com` というサフィックスの下にあります。サーバーによるこのタイプの構文の処理時間は、前述の例の処理時間よりも短くなります。

ROLEDN バインドタイプを指定した例

次に、ロール DN に基づくバインドに関連する `userattr` キーワードの例を示します。

```
userattr = "exampleEmployeeReportsTo#ROLEDN"
```

バインド DN がターゲットエントリの `exampleEmployeeReportsTo` 属性で指定されたロールに属していれば、バインドルールは true と判定されます。たとえば、社内のすべてのマネージャに対して階層化されたロールを作成する場合は、このメカニズムを使用して、マネージャよりも下の役職にある社員に関する情報へのすべてのレベルのアクセス権を、マネージャに与えることができます。

ロールの DN を、ディレクトリ内の任意のサフィックスの下に置くことができます。さらに、フィルタを適用したロールを使用する場合は、サーバーがこのタイプの ACI を評価するためには、多くのリソースを必要とします。

LDAPURL バインドタイプを指定した例

次に、LDAP フィルタに基づくバインドに関連する `userattr` キーワードの例を示します。

```
userattr = "myfilter#LDAPURL"
```

バインド DN とターゲットエントリの *myfilter* 属性で指定されたフィルタが一致すれば、バインドルールは true と判定されます。*myfilter* 属性は、LDAP フィルタを含む任意の属性に置き換えることができます。

任意の属性値を指定した例

次に、任意の属性値に基づくバインドに関連する `userattr` キーワードの例を示します。

```
userattr = "favoriteDrink#Beer"
```

バインド DN とターゲット DN の両方に `favoriteDrink` 属性が含まれ、その値がともに **`Beer`** であれば、バインドルールは true と判定されます。

継承を含む userattr キーワードの使用

userattr キーワードを使用して、バインド操作に使用されるエントリをターゲットエントリと関連付けると、ACI は指定されたターゲットだけに適用され、下位のエントリには適用されません。ただし、状況によっては、ターゲットエントリよりも下位のエントリにも、ACI の適用が必要になることもあります。このためには、parent キーワードを使用して、ターゲットのいくつ下のレベルまで ACI を継承するかを指定します。

userattr キーワードとともに parent キーワードを使用する場合の構文は次のとおりです。

```
userattr = "parent[inheritance_level].attribute#bindType"
```

各オプションは、次のように指定します。

- *inheritance_level* は、ターゲットのいくつ下のレベルまで ACI を継承するかを示すリストで、各レベルはコンマで区切る。レベルはターゲットエントリの 5 レベル [0,1,2,3,4] 下まで指定できる。0 はターゲットエントリを示す
- *attribute* は、userattr または groupattr キーワードのターゲットとなる属性
- *bindType* には、USERDN、GROUPDN のいずれかを指定する。LDAPURL および ROLEDN のバインドタイプは、継承ではサポートされない

次に例を示します。

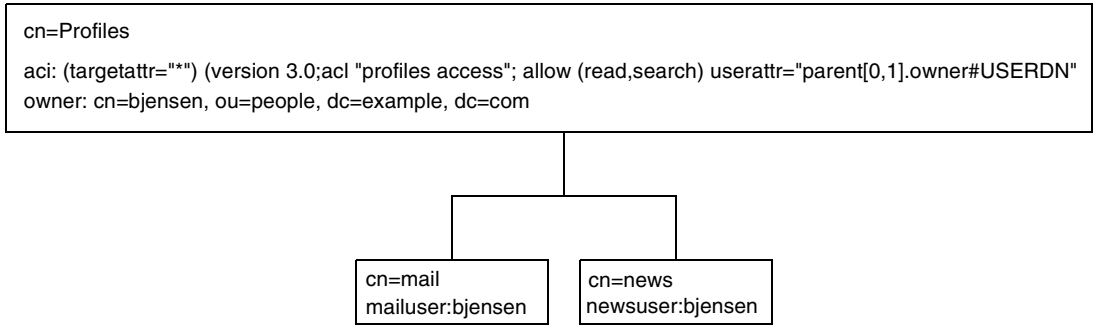
```
userattr = "parent[0,1].manager#USERDN"
```

バインド DN とターゲットエントリの manager 属性が一致すれば、このバインドルールは true と判定されます。バインドルールが true と判定されると、アクセス権が与えられます。このアクセス権は、ターゲットエントリおよびその直下にあるすべてのエントリに適用されます。

userattr の継承を含む例

次の図は、bjensen というユーザーが、cn=Profiles エントリ、および cn=mail と cn=news を含む 1 レベル下の子エントリに対して、読み取りと検索を許可された例を示しています。つまり、このユーザーは、自身のメールとニュース ID をすべて検索できます。

図 6-1　userattr キーワードでの継承の使用



この例において、継承を使用せずに同じ結果を得るには、次のどちらかの操作を実行する必要があります。

- ディレクトリ内の cn=Profiles、cn=mail、および cn=news エントリに対するユーザー bjensen の読み取りアクセスと検索アクセスを明示的に設定する
- cn=mail および cn=news エントリに対して owner 属性を追加し、その値を bjensen にする。さらに cn=mail および cn=news エントリに次の ACI を追加する

  ```
  aci: (targetattr="*") (version 3.0; acl "profiles access"; allow
   (read,search) userattr="owner#USERDN";)
  ```

userattr キーワードによる追加アクセス権の許可

all または **add** アクセス権とともに userattr キーワードを使用すると、サーバーが期待どおりに動作しないことがあります。通常、ディレクトリ内に新しいエントリを作成すると、Directory Server は作成されているエントリのアクセス権限を確認しますが、親エントリのアクセス権限は確認されません。ただし、userattr キーワードを使用する ACI の場合は、この動作によってセキュリティホールが生じる可能性があるので、これを避けるためにサーバーの通常動作は変更されます。

次のような例を想定します。

```
aci: (target="ldap:///dc=example,dc=com")(targetattr=*) (version
3.0;
 acl "manager-write"; allow (all) userattr = "manager#USERDN";)
```

この ACI は、部下のエントリに対するすべての権限をマネージャに与えます。ただし、新しく作成されるエントリについてもアクセス権限が確認されるので、このタイプの ACI では、すべての社員がエントリを作成でき、そのエントリについては manager 属性を社員自身の DN に設定できます。たとえば、会社に不満を持つ Joe という社員 (cn=Joe,ou=eng,dc=example,dc=com) がツリーの Human Resources 分岐にエントリを作成した場合、Human Resources の社員に与えられているアクセス権を所有し、そのアクセス権を使用する (あるいは悪用する) ことが可能になります。

このような行為は、次のようなエントリを作成することで実現できてしまいます。

```
dn: cn= Trojan Horse,ou=Human Resources,dc=example,dc=com
objectclass: top
...
cn: Trojan Horse
manager: cn=Joe,ou=eng,dc=example,dc=com
```

このようなセキュリティ上の危険を回避するために、ACI の評価プロセスでは、レベル 0 の追加権限、つまりエントリ自身に対する追加権限を与えません。ただし、既存エントリの下位にあるエントリには、parent キーワードを使用して追加アクセス権を与えることができます。親のいくつ下のレベルまで追加アクセス権を許可するかを指定する必要があります。たとえば、次の ACI によって、dc=example,dc=com 内にあってバインド DN に一致する manager 属性を持つ任意のエントリに、子エントリを追加できます。

```
aci: (target="ldap:///dc=example,dc=com")(targetattr=*)
 (version 3.0; acl "parent-access"; allow (add)
 userattr = "parent[0,1].manager#USERDN";)
```

この ACI は、バインド DN と親エントリの manager 属性が一致するユーザーだけに追加アクセス権を与えます。

特定 IP アドレスからのアクセスの定義

バインドルールを使用して、特定の IP アドレスからバインドするように指定できます。これは、ディレクトリへのすべての更新が、特定のマシンまたはネットワークドメインから行われるように強制する場合によく使用されます。

IP アドレスに基づくバインドルールを設定するための LDIF 構文は、次のとおりです。

```
ip = "IPaddressList" または ip != "IPaddressList"
```

IPaddressList は、次の項目からなる 1 つまたは複数の要素です (複数の要素はコンマで区切られます)。

- 特定の IPv4 アドレス (123.45.6.7 など)

- ワイルドカードを使ってサブネットワークを指定した IPv4 アドレス (`12.3.45.*` など)
- サブネットワークマスクを持つ IPv4 アドレスまたはサブネットワーク (`123.45.6.*+255.255.255.115` など)
- RFC 2373 (`http://www.ietf.org/rfc/rfc2373.txt`) の定義に従った、有効な形式で指定された IPv6 アドレス。次のアドレスは、どれもが同等と見なされる
  - `12AB:0000:0000:CD30:0000:0000:0000:0000`
  - `12AB::CD30:0:0:0:0`
  - `12AB:0:0:CD30::`
- サブネットの接尾辞を持つ IPv6 アドレス (`12AB::CD30:0:0:0:0/60` など)

ディレクトリにアクセスするクライアントが指定された IP アドレスを持っていれば、バインドルールは true と判定されます。この方法は、一部のディレクトリへのアクセス元を、特定のサブネットまたはマシンに制限する場合に有効です。

ACI の適用対象を特定のコンピュータに制限するには、アクセス制御エディタを使用してサーバーコンソールから定義します。詳細は、219 ページの「コンソールを使用した ACI の作成」を参照してください。

特定ドメインからのアクセスの定義

バインドルールを使用して、特定のドメインまたは特定のホストマシンだけからバインドできるように指定できます。これは、ディレクトリへのすべての更新が、特定のマシンまたはネットワークドメインから行われるように強制する場合によく使用されます。

DNS ホスト名に基づくバインドルール設定のための LDIF 構文は、次のとおりです。

```
dns = "DNS_Hostname" or dns != "DNS_Hostname"
```

警告 `dns` キーワードを使用するためには、マシンで使用されるネームサービスは DNS である必要があります。ネームサービスが DNS でない場合、`dns` キーワードの代わりに `ip` キーワードを使用します。

`dns` キーワードには、完全修飾による DNS ドメイン名が必要です。ドメインを指定せずにホストへのアクセス権を与えると、セキュリティ上の問題が発生する可能性があります。たとえば、次のような式を使用することもできますが、このような方法はできるだけ避けてください。

```
dns = "legend.eng";
```

名前は、絶対パスで指定します。

```
dns = "legend.eng.example.com";
```

dns キーワードではワイルドカードを使用できます。たとえば、次のようにします。

```
dns = "*.example.com";
```

この例では、ディレクトリにアクセスするクライアントが指定されたドメインにあれば、バインドルールは true と判定されます。これは、アクセスを特定ドメインに制限する場合に有効です。使用しているシステムのネームサービスが DNS でなければ、ワイルドカードは使用できません。ネームサービスが DNS でない場合、アクセスを特定ドメインからのアクセスに制限するには、212 ページの「特定 IP アドレスからのアクセスの定義」の説明に従って、ip キーワードを使用します。

特定の時刻または曜日におけるアクセスの定義

バインドルールを使用して、特定の時刻または曜日だけにバインドするように制限できます。たとえば、月曜日から金曜日の朝 8 時から午後 5 時までの間にアクセスを制限するようなルールを設定できます。アクセス権限の評価に使用される時刻はディレクトリサーバー上の時刻で、クライアント上の時刻ではありません。

時刻に基づくバインドルールを設定するための LDIF 構文は、次のとおりです。

```
timeofday operator "time"
```

operator には、次のどれかを指定できます。等号 (=)、不等号 (!=)、大なり記号 (>)、大きいまたは等しい (>=)、小なり記号 (<)、小さいまたは等しい (<=)。

timeofday キーワードでは、24 時間法による「時」と「分」で、時刻を表します (0 ～ 2359)。

注	評価にはクライアント上の時刻ではなく、サーバー上の時刻が使用されます。

曜日に基づくバインドルールを設定するための LDIF 構文は、次のとおりです。

```
dayofweek = "day1, day2 ..."
```

dayofweek キーワードの値には、アルファベット 3 文字で示される sun、mon、tue、wed、thu、fri、sat の曜日の略号が使用されます。

例

次に、timeofday および dayofweek 構文の例を示します。

```
timeofday = "1200";
```

クライアントが正午ちょうどにディレクトリにアクセスすると、バインドルールは true と判定されます。

```
timeofday != "0100";
```

クライアントが午前 1 時以外の任意の時刻にディレクトリにアクセスすると、バインドルールは true と判定されます。

```
timeofday > "0800";
```

クライアントが午前 8 時を過ぎてからディレクトリにアクセスすると、バインドルールは true と判定されます。

```
timeofday < "1800";
```

クライアントが午後 6 時前にディレクトリにアクセスすると、バインドルールは true と判定されます。

```
timeofday >= "0800";
```

クライアントが午前 8 時以後にディレクトリにアクセスすると、バインドルールは true と判定されます。

```
timeofday <= "1800";
```

クライアントが午後 6 時以前にディレクトリにアクセスすると、バインドルールは true と判定されます。

```
dayofweek = "Sun, Mon, Tue";
```

クライアントが日曜日、月曜日、または火曜日にディレクトリにアクセスすると、バインドルールは true と判定されます。

認証方法に基づくアクセスの定義

クライアントが特定の認証方法でディレクトリにバインドするように、バインドルールを設定できます。次に示す認証方法を使用できます。

- None : 認証は不要です。この値がデフォルトで、匿名アクセスを表します。
- Simple : クライアントはユーザー名とパスワードを入力し、ディレクトリにバインドする必要があります。
- SSL : クライアントは、SSL (Secure Socket Layer) または TLS (Transport Layer Security) 接続により、ディレクトリにバインドする必要があります。

  SSL の場合は、接続は LDAPS の 2 番目のポートに確立されます。TLS の場合は、Start TLS 操作によって接続が確立されます。どちらの場合も証明書が必要です。SSL 設定については、第 11 章「セキュリティの実装」を参照してください。

- SASL : クライアントは、SASL (Simple Authentication and Security Layer) 接続により、ディレクトリにバインドする必要があります。Sun ONE Directory Server には SASL モジュールはありません。

認証方法に基づくバインドルールは、アクセス制御エディタでは設定できません。

認証方法に基づくバインドルールを設定するための LDIF 構文は、次のとおりです。

```
authmethod = "authentication_method"
```

ここで、*authentication_method* は、**none**、**simple**、**ssl**、または "**sasl** *sasl_mechanism*" です。

例

次に、authmethod キーワードの例を示します。

```
authmethod = "none";
```

バインドルールの評価時に認証検査は行われません。

```
authmethod = "simple";
```

クライアントがユーザー名とパスワードを使用してディレクトリにアクセスすると、バインドルールは true と判定されます。

```
authmethod = "ssl";
```

クライアントが LDAPS を経由した証明書を使用してディレクトリ に対する認証を行うと、バインドルールは true と判定されます。クライアントが LDAPS を経由した単純認証 (バインド DN とパスワード) を行うと、バインドルールは false と判定されます。

```
authmethod = "sasl DIGEST-MD5";
```

クライアントが SASL DIGEST-MD5 メカニズムを使用してディレクトリにアクセスすると、バインドルールは true と判定されます。サポートされている SASL メカニズムには、これ以外に EXTERNAL と GSSAPI があります (Solaris システムのみ)。

ブール型バインドルールの使用

AND、OR、NOT のブール式を使用して細かいアクセスルールを設定すると、複雑なバインドルールを作成できます。ブール型バインドルールは、サーバーコンソールでは作成できません。LDIF 文を作成する必要があります。

ブール型バインドルールの LDIF 構文は、次のとおりです。

bindRule [*boolean*] [*bindRule*] [*boolean*] [*bindRule*] . . . ;)

たとえば、バインド DN が管理者のグループまたはメール管理者のグループのメンバーで、クライアントが example.com ドメイン内から実行されていれば、次のバインドルールは true と判定されます。

```
(groupdn = "ldap:///cn=administrators,dc=example,dc=com" or
groupdn = "ldap:///cn=mail administrators,dc=example,dc=com" and
dns = "*.example.com";)
```

最後のセミコロン (;) は省略できません。

ブール式は、次の順序で評価されます。

- 内側のカッコでくくられた式から外側のカッコでくくられた式へ
- すべての式を左から右へ
- AND または OR 演算子の前に NOT ブール演算子

OR と AND の優先順位はありません。

次のようなブール型バインドルールがあるとします。

(*bindRule_A*) OR (*bindRule_B*)

(*bindRule_B*) OR (*bindRule_A*)

ブール式は左から右へ評価されるので、上の例ではバインドルール B の前にバインドルール A が評価され、下の例ではバインドルール A の前にバインドルール B が評価されます。

ただし、ブール演算子 NOT は、OR または AND よりも先に評価されます。たとえば、次のような式があるとします。

(*bindRule_A*) AND NOT (*bindRule_B*)

ここでは、「左から右へ」の原則は適用されず、バインドルール A よりも先にバインドルール B が評価されます。

コマンド行からの ACI の作成

LDIF 文を使用してアクセス制御命令を手動で作成し、ldapmodify コマンドを使用して、その命令をディレクトリツリーに追加できます。ACI の値は複雑であるため、新しい ACI を作成するときは、既存の値を表示してコピーすると便利です。

aci 属性値の表示

ACI は aci 属性の値として、エントリに格納されます。aci は複数の値を持つオペレーショナル属性であり、ディレクトリユーザーはこの属性の読み取りや変更を行うことができます。この属性自体が ACI で保護される必要があります。通常、管理ユーザーには aci 属性へのすべてのアクセス権が許可されるので、管理ユーザーは次の方法のどれかでこの属性の値を表示できます。

汎用エディタを使用して、他の属性値と同様に aci 属性値を表示できます。Directory Server コンソールの最上位の「ディレクトリ」タブで、ACI が格納されているエントリを右クリックし、メニューから「汎用エディタで編集」を選択します。ただし、aci の値は長い文字列であることが多く、このダイアログでは表示や編集が困難な場合があります。

代わりに、ディレクトリツリー内でエントリを右クリックし、「アクセス権を設定」を選択すると、アクセス制御エディタを起動できます。ACI を選択して「編集」をクリックし、「手動での編集」をクリックして、対応する aci 値を表示します。ACI エディタの手動モードとビジュアルモードを切り替えることで、aci 値の構文とその設定を比較できます。

また、オペレーティングシステムが対応している場合は、汎用エディタまたは手動モードのアクセス制御エディタから aci 値をコピーし、作成中の LDIF ファイルにペーストすることもできます。管理ユーザーは次の ldapsearch コマンドを実行して、エントリの aci 属性を表示することもできます。

```
% ldapsearch -h host -p port -D "cn=Directory Manager" -w password ¥
              -b entryDN -s base aci
```

このコマンドで得られた LDIF テキストを、新しい LDIF ACI 定義にコピーして編集できます。

注 aci の値によってアクセス権限がどのように許可または拒否されるかを確認するには、244 ページの「実効権限の表示」を参照してください。

コンソールを使用した ACI の作成

Directory Server コンソールを設定して、ディレクトリのどのエントリが aci 属性を持っているかを表示できます。この表示のオンとオフを切り替えるには、「表示」>「表示」>「ACI カウント」の順に選択または選択解除します。最上位の「ディレクトリ」タブに一覧表示されるエントリの末尾に、その aci 属性に定義されている ACI の数が表示されます。Directory Server コンソールを使用して、ディレクトリの ACI の表示、作成、編集、削除を行うことができます。

Directory Server のセキュリティポリシーに使用される一般的なアクセス制御規則と、その規則を作成するための Directory Server コンソールの使い方の手順については、224 ページの「アクセス制御の使用例」を参照してください。

アクセス制御エディタがビジュアル編集モードになっている場合は、複雑な ACI を作成できません。特に、アクセス制御エディタから次の操作を実行できません。

- アクセスの拒否 (199 ページの「アクセス権の構文」を参照)
- 値に基づいた ACI の作成 (194 ページの「LDAP フィルタを使用した属性値のターゲット指定」を参照)
- 親アクセスの定義 (203 ページの「親アクセス (parent キーワード)」を参照)
- ブール型バインドルールを含む ACI の作成 (217 ページの「ブール型バインドルールの使用」を参照)
- 一般的に、次のキーワードを使用する ACI の作成 : `roledn`、`userattr`、`authmethod`

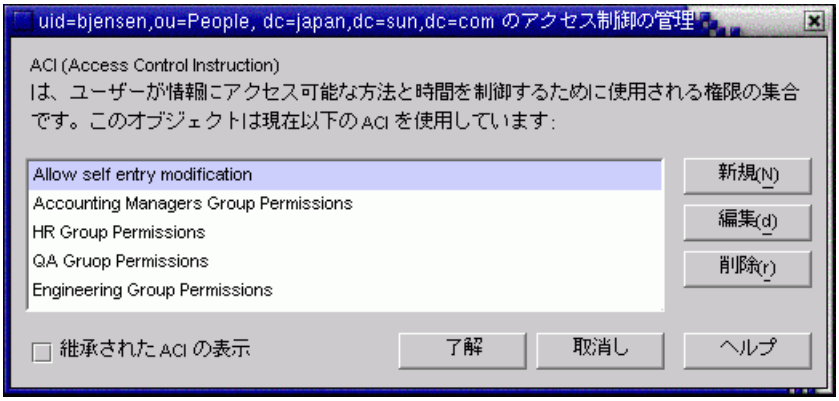
ヒント　アクセス制御エディタで「手動での編集」ボタンをクリックすると、グラフィカルインタフェースで変更した内容をいつでも LDIF で確認できます。

エントリの ACI の表示

1. Directory Server コンソールの最上位レベルにある「ディレクトリ」タブでディレクトリツリーを表示し、アクセス制御を設定するエントリを探します。ACI を編集するには、ディレクトリ管理者または Directory Manager としての権限が必要です。
2. このエントリをマウスの右ボタンでクリックし、ポップアップメニューから「アクセス権を設定」を選択します。あるいは、エントリをクリックして選択し、「オブジェクト」メニューから「アクセス権を設定」を選択します。

   次の図に示すように、「アクセス制御の管理」ダイアログが表示されます。このダイアログボックスには、選択したエントリで定義されたすべての ACI についての説明が一覧表示され、ACI を修正、削除、および新しく作成できます。

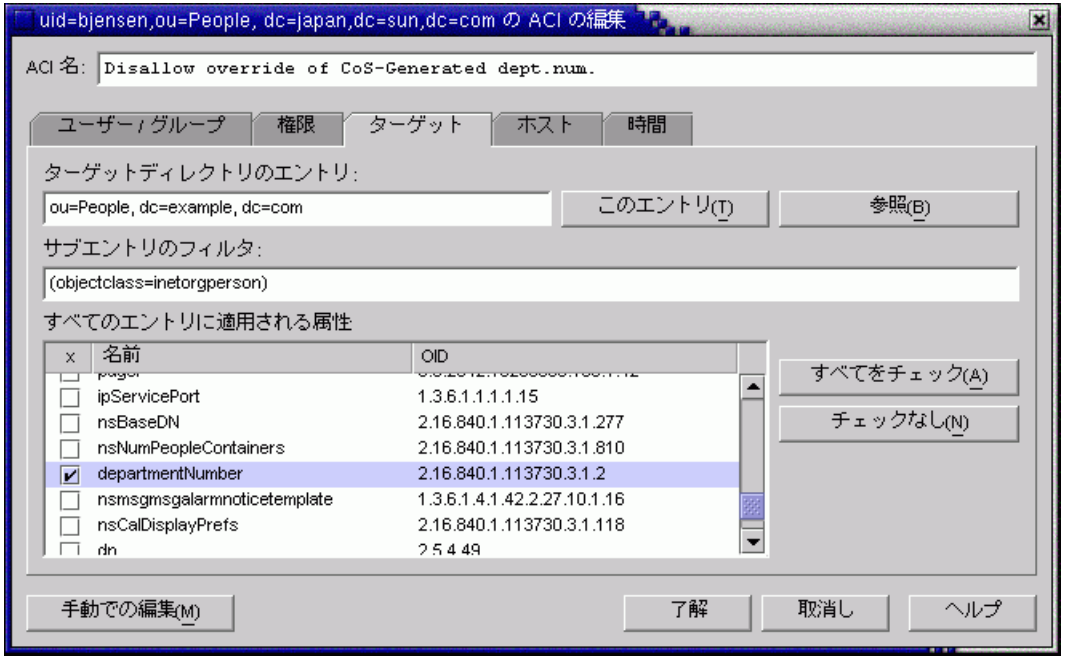
図 6-2　「アクセス制御の管理」ダイアログボックス



「継承された ACI の表示」チェックボックスを選択すると、選択したエントリの親によって定義され、エントリに適用されるすべての ACI も一覧表示されます。ただし、継承された ACI を修正または削除することはできません。エントリは定義された場所で管理する必要があります。

3. 「新規」をクリックし、選択したオブジェクトとそのサブツリー全体に対する新しいアクセス権を定義します。次の図に示すように、ACI エディタが表示されます。

図 6-3　ACI エディタダイアログ



ダイアログボックス最上部の「ACI 名」には、「アクセス制御の管理」ダイアログボックスに表示される ACI の記述が表示されます。ACI にわかりやすい名前を付けると、ディレクトリで ACI を管理しやすくなります。最下位のエントリ上の継承された ACI を表示する場合には特にそうです。

「アクセス制御エディタ」のタブを使うと、アクセスを受け入れまたは拒否されたユーザー、アクセス中またはアクセス制限中のターゲット、許可されたホスト名および操作時間などの詳細なパラメータを指定できます。「アクセス制御」タブの各フィールドについては、オンラインヘルプを参照してください。

ACI エディタのタブには、ACI 値の内容がグラフィック表示されます。ACI 値を表示してテキストとして編集するには、「手動での編集」ボタンをクリックします。テキストエディタを使って、タブでは定義できない高度な ACI を定義できます。ただし、ACI 値を編集すると、高度な機能を利用するかどうかに関係なく、それ以後は ACI を視覚的に編集できなくなることがあります。

新しい ACI の作成

1. アクセス制御エディタを表示します。

   この手順については、219 ページの「エントリの ACI の表示」を参照してください。

   表示画面が 221 ページの図 6-3 と異なる場合は、「ビジュアル編集」ボタンをクリックします。

2. 「ACI 名」テキストボックスに ACI の名前を入力します。

   ACI 名には任意の文字列を指定できます。ほかの ACI と重複しない名前を付けてください。名前を指定しない場合は、自動的に「名前のない ACI」という名前が付けられます。

3. 「ユーザー / グループ」タブで「すべてのユーザー」を強調表示してアクセス権を与えるユーザーを選択するか、「追加」ボタンをクリックして追加するユーザーのディレクトリを検索します。

   「ユーザーおよびグループの追加」ウィンドウで、次の手順を実行します。

   a. ドロップダウンリストから検索領域を選択し、「検索」フィールドに検索文字列を入力してから、「検索」ボタンをクリックします。

      下のリストに検索結果が表示されます。

   b. 検索結果リストで必要なエントリを選択し、「追加」ボタンをクリックして、アクセス権が与えられたエントリのリストにそれらを追加します。

   c. 「了解」をクリックして、「ユーザーおよびグループの追加」ウィンドウを閉じます。

      選択したエントリが ACI エディタの「ユーザー / グループ」タブに一覧表示されます。

4. アクセス制御エディタで「権限」タブをクリックし、チェックボックスを使用して与える権限を選択します。

5. 「ターゲット」タブをクリックし、「このエントリ」をクリックして、ACI のターゲットとして指定されているノードを表示します。

   ターゲット DN の値は変更できますが、新しい DN は、選択したエントリの直接的または間接的な子である必要があります。

   このノードの下にあるサブツリー内の一部のエントリを ACI のターゲットから外す場合は、「サブエントリのフィルタ」フィールドにフィルタを入力する必要があります。

   さらに、ターゲットとして指定する属性を属性リストから選択することによって、ACI の範囲を特定の属性だけに制限できます。

6. 「ホスト」タブをクリックしてから「追加」ボタンをクリックして、「ホストフィルタの追加」ダイアログボックスを表示します。

   ホスト名または IP アドレスを指定できます。IP アドレスを指定する場合は、ワイルドカード文字 (*) を使用できます。

7. 「時間」タブをクリックして、アクセスが許可される時刻のテーブルを表示します。

   デフォルトでは、常時アクセスが許可されています。テーブル上でカーソルを操作し、時刻をクリックしてドラッグすることによって、アクセス時間を変更できます。連続していない時間帯を選択することはできません。

8. ACI の修正が完了したら、「了解」をクリックします。

   ACI エディタが閉じられ、ACI マネージャのウィンドウに新しい ACI のリストが表示されます。

注 ACI の作成中に「手動での編集」ボタンをクリックすると、入力した内容に対応する LDIF 文をいつでも表示できます。この文は変更できますが、加えた変更は必ずしもグラフィカルインタフェースに反映されません。

ACI の編集

ACI を編集するには、次の手順を実行します。

1. 「ディレクトリ」タブで、サブツリーの一番上のエントリを右クリックし、ポップアップメニューから「アクセス権を設定」を選択します。

   「アクセス制御の管理」ウィンドウが表示されます。このウィンドウには、そのエントリに属する ACI のリストが表示されます。

2. 「アクセス制御の管理」ウィンドウで、編集する ACI を選択し、「編集」をクリックします。

   アクセス制御エディタが表示されます。このダイアログボックスで編集できる情報については、オンラインヘルプを参照してください。

3. アクセス制御エディタの各種タブを使用して、必要な変更を加えます。

4. ACI の修正が完了したら、「了解」をクリックします。

   ACI エディタが閉じられ、ACI マネージャのウィンドウに変更された ACI のリストが表示されます。

ACI の削除

ACI を削除するには、次の手順を実行します。

1. 「ディレクトリ」タブで、サブツリーの一番上のエントリを右クリックし、ポップアップメニューから「アクセス権を設定」を選択します。

   「アクセス制御の管理」ウィンドウが表示されます。このウィンドウには、そのエントリに属する ACI のリストが表示されます。

2. 「アクセス制御の管理」ウィンドウで、削除する ACI を選択します。
3. 「削除」をクリックします。

   削除した ACI は、アクセス制御の管理に表示されなくなります。

アクセス制御の使用例

この節で示す例では、架空の ISP である example.com 社が、アクセス制御ポリシーを決定していきます。すべての例では、コンソールまたは LDIF ファイルを使用して、与えられたタスクをどのように処理するかを説明しています。

Example.com 社の業務は、Web ホスティングサービスとインターネットアクセスの提供です。Example.com の Web ホスト サービスには、クライアント企業のディレクトリのホスティングが含まれます。Example.com は実際に 2 つの中規模企業のディレクトリ Company333 と Company999 をホストし、部分的に管理を行なっています。また、多数の個人契約者にインターネットへのアクセスを提供しています。

現在、example.com 社は、次のようなアクセス制御規則を設定しようとしています。

- example.com 社の社員に、example.com ツリー全体を対象とした読み取り、検索、および比較のための匿名アクセス権を与える (225 ページの「匿名アクセスの許可」を参照)
- example.com 社の社員に、`homeTelephoneNumber`、`homeAddress` などの個人情報への書き込みアクセス権を与える (227 ページの「個人のエントリへの書き込みアクセス権の許可」を参照)
- example.com 社の社員が個人のエントリにロールを追加するアクセス権を与える。ただし、一部の重要なロールは除く (230 ページの「重要なロールに対するアクセスの制限」を参照)
- example.com 社の Human Resources グループに、People 分岐のエントリを対象としたすべての権限を与える (232 ページの「サフィックスに対するすべてのアクセス権のグループへの許可」を参照)

- example.com 社のすべての社員に対し、ディレクトリの Social Committee 分岐の下にグループエントリを作成し、自身が所有するグループエントリを削除するアクセス権を与える (233 ページの「グループエントリの追加および削除権限の許可」を参照)
- example.com 社のすべての社員に対し、Social Committee 分岐の下のグループエントリに、自身を追加するアクセス権を与える (240 ページの「ユーザー自身の操作によるグループへの参加と不参加」を参照)
- SSL 認証、日時の制約、位置の指定などの一定の条件付きで、ディレクトリツリーのそれぞれの分岐へのアクセス権を Company333 および Company999 のディレクトリ管理者 (ロール) に与える (235 ページの「グループまたはロールへの条件付きアクセスの許可」を参照)
- 個人契約者に対し、個人のエントリへのアクセス権を与える (227 ページの「個人のエントリへの書き込みアクセス権の許可」を参照)
- 個人契約者が個人のエントリ内の課金情報にアクセスできないようにする (238 ページの「アクセスの拒否」を参照)
- 世界のユーザーに対し、個人契約者のサブツリーへの匿名アクセス権を与える。ただし、特に非公開を希望している契約者は除く。ディレクトリのこの部分は、ファイアウォール外部のスレーブサーバーとなることがあり、毎日 1 回更新される (225 ページの「匿名アクセスの許可」および 240 ページの「フィルタを使用したターゲットの設定」を参照)

匿名アクセスの許可

ほとんどのディレクトリは、読み取り、検索、または比較を行うために、少なくとも 1 つのサフィックスに匿名でアクセスできるように設定されています。たとえば、電話帳のように、企業内の個人情報を収めたディレクトリを管理している場合に、社員がその内容を検索できるようにするには、そのためのアクセス権の設定が必要になることがあります。これは example.com 社内のケースであり、「ACI「Anonymous example.com」」にその例が示されています。

example.com 社では、ISP として、世界中からアクセス可能な公開電話帳を作成し、契約者全員の連絡先情報を公開することも計画しています。これについては、「ACI「Anonymous World」」で例を示しています。

ACI「Anonymous example.com」

example.com 社の社員に example.com ツリー全体を対象とした読み取り、検索、および比較アクセス権を与えるには、LDIF で次のような文を作成します。

```
aci: (targetattr !="userPassword")(version 3.0; acl "Anonymous
 example"; allow (read, search, compare) userdn= "ldap:///anyone"
and
 dns="*.example.com";)
```

この例では、aci を dc=example,dc=com エントリに追加することを仮定しています。userPassword 属性は ACI の対象に含まれていません。

このアクセス権を設定するには、コンソールを使用して次の手順を実行します。

1. 「ディレクトリ」タブの左側のナビゲーションツリーで example.com ノードを右クリックし、ポップアップメニューから「アクセス権を設定」を選択して、アクセス制御の管理を表示します。
2. 「新規」をクリックして、アクセス制御エディタを表示します。
3. 「ユーザー / グループ」タブの ACI 名フィールドに、「Anonymous example.com」と入力します。アクセス権が与えられたユーザーのリストに、「すべてのユーザー」と表示されていることを確認します。
4. 「権限」タブで、読み取り (read)、比較 (compare)、および検索 (search) の各権限のチェックボックスを選択します。これ以外のチェックボックスは、選択が解除されていることを確認してください。
5. 「ターゲット」タブで「このエントリ」をクリックすると、ターゲットディレクトリの入力フィールドにサフィックス dc=example,dc=com が表示されます。属性テーブルで userPassword 属性を検索し、対応するチェックボックスの選択を解除します。

   これ以外のチェックボックスは選択されている必要があります。「名前」ヘッダーをクリックして属性リストをアルファベット順に並べ替えると、userPassword 属性の検索が簡単になります。
6. 「ホスト」タブの「追加」をクリックし、「DNS ホストフィルタ」フィールドに「*.example.com」と入力します。「了解」をクリックして、ダイアログボックスを閉じます。
7. 「アクセス制御エディタ」ウィンドウの「了解」ボタンをクリックします。

   「アクセス制御の管理」ウィンドウの ACI リストに、新しい ACI が追加されます。

ACI「Anonymous World」

個人契約者サブツリーの読み取りおよび検索アクセス権を世界中に与え、非公開契約者の情報へのアクセスを拒否するには、LDIF で次のような文を作成します。

```
aci: (targetfilter= "(!(unlistedSubscriber=yes))")
 (targetattr="homePostalAddress || homePhone || mail") (version 3.0;
 acl "Anonymous World"; allow (read, search) userdn=
 "ldap:///anyone";)
```

この例では、ACI を ou=subscribers,dc=example, dc=com エントリに追加することを仮定しています。また、各契約者のエントリには、yes または no の値を持つ unlistedSubscriber 属性が設定されているものとします。非公開契約者は、この属性値に基づいて、ターゲット定義のフィルタによって除外されます。フィルタ定義については、240 ページの「フィルタを使用したターゲットの設定」を参照してください。

このアクセス権を設定するには、コンソールを使用して次の手順を実行します。

1. 「ディレクトリ」タブの左側のナビゲーションツリーで example.com ノードの下にある Subscribers エントリを右クリックし、ポップアップメニューから「アクセス権を設定」を選択して、アクセス制御の管理を表示します。
2. 「新規」をクリックして、アクセス制御エディタを表示します。
3. 「ユーザーおよびグループ」タブの ACI 名フィールドに、「Anonymous World」と入力します。アクセス権が与えられたユーザーのリストに、「すべてのユーザー」と表示されていることを確認します。
4. 「権限」タブで、読み取り (read) と検索 (search) の各権限のチェックボックスを選択します。これ以外のチェックボックスは、選択が解除されていることを確認してください。
5. 「ターゲット」タブで「このエントリ」をクリックすると、ターゲットディレクトリの入力フィールドにサフィックス dc=subscribers, dc=example,dc=com が表示されます。

   a. 「サブエントリのフィルタ」フィールドに、次のフィルタを入力します。

      (!(unlistedSubscriber=yes))

   b. 属性テーブルで、homePhone、homePostalAddress、および mail 属性のチェックボックスを選択します。

      ただし、これ以外のチェックボックスの選択は、解除されている必要があります。「チェックしない」ボタンをクリックしてテーブル内のすべての属性のチェックボックスの選択を解除し、次に「名前」ヘッダーをクリックしてアルファベット順に属性を並べ替えると、この作業が簡単になります。

6. 「了解」をクリックします。

   「アクセス制御の管理」ウィンドウの ACI リストに、新しい ACI が追加されます。

個人のエントリへの書き込みアクセス権の許可

多くの場合、内部ユーザーが個人で変更できるエントリの属性は、ディレクトリ管理者によって一部だけに制限されています。example.com 社のディレクトリ管理者は、ユーザーが変更できる対象を、パスワード、自宅の電話番号、自宅住所だけに制限しようとしています。これについては、「ACI「Write example.com」」で例を示しています。

また、契約者がディレクトリに対して SSL 接続を確立することを条件に、example.com ツリー内にある個人情報を更新できるようにするというポリシーもあります。これについては、「ACI「Write Subscribers」」で例を示しています。

ACI「Write example.com」

注	このアクセス権を設定することによって、ユーザーは属性値の削除アクセス権も与えられます。

example.com 社の社員が、個人のパスワード、自宅の電話番号、自宅住所を変更できるようにするには、LDIF で次のような文を作成します。

```
aci: (targetattr="userPassword || homePhone || homePostalAddress")
 (version 3.0; acl "Write example.com"; allow (write) userdn=
 "ldap:///self" and dns="*.example.com";)
```

この例では、ACI を `ou=example-people,dc=example, dc=com` エントリに追加しています。

このアクセス権を設定するには、コンソールを使用して次の手順を実行します。

1. 「ディレクトリ」タブの左側のナビゲーションツリーで example.com ノードを右クリックし、ポップアップメニューから「アクセス権を設定」を選択して、アクセス制御の管理を表示します。
2. 「新規」をクリックして、アクセス制御エディタを表示します。
3. 「ユーザー / グループ」タブの ACI 名フィールドに、「Write example.com」と入力します。アクセス権が与えられたユーザーのリストで、次の手順を実行します。
   a. 「すべてのユーザー」を選択して削除し、「追加」をクリックします。

      「ユーザーおよびグループの追加」ダイアログボックスが表示されます。

   b. 「検索領域」を「特殊権限」に設定し、「検索結果」リストで「自己」を選択します。
   c. 「追加」ボタンをクリックすると、アクセス権が与えられたユーザーのリストに「自己」が追加されます。
   d. 「了解」をクリックして、「ユーザーおよびグループの追加」ダイアログボックスを閉じます。
4. 「権限」タブで、書き込みアクセス権 (write) のチェックボックスを選択します。これ以外のチェックボックスは、選択が解除されていることを確認してください。

5. 「ターゲット」タブで「このエントリ」をクリックすると、ターゲットディレクトリの入力フィールドにサフィックス dc=example,dc=com が表示されます。属性テーブルで、homePhone、homePostalAddress、および userPassword 属性のチェックボックスを選択します。

   ただし、これ以外のチェックボックスの選択は、解除されている必要があります。「チェックしない」ボタンをクリックしてテーブル内のすべての属性のチェックボックスの選択を解除し、次に「名前」ヘッダーをクリックしてアルファベット順に属性を並べ替えると、この作業が簡単になります。

6. 「ホスト」タブの「追加」ボタンをクリックして、「ホストフィルタの追加」ダイアログボックスを表示します。「DNS ホストフィルタ」フィールドに「*.example.com」と入力します。「了解」をクリックして、ダイアログボックスを閉じます。

7. 「アクセス制御エディタ」ウィンドウの「了解」ボタンをクリックします。

   「アクセス制御の管理」ウィンドウの ACI リストに、新しい ACI が追加されます。

ACI「Write Subscribers」

注	このアクセス権を設定することによって、ユーザーは属性値の削除アクセス権も与えられます。

example.com 社の契約者が個人のパスワードと自宅の電話番号を変更できるようにするには、LDIF で次のような文を作成します。

```
aci: (targetattr="userPassword || homePhone") (version 3.0; acl
 "Write Subscribers"; allow (write) userdn= "ldap://self" and
 authmethod="ssl";)
```

この例では、aci を ou=subscribers,dc=example, dc=com エントリに追加することを仮定しています。

住所は example.com 社からの請求に必要な情報で、この情報を削除する可能性があるので、契約者にはこの属性への書き込みアクセス権は与えられていません。つまり、自宅住所はビジネス的に重要な情報なのです。

このアクセス権を設定するには、コンソールを使用して次の手順を実行します。

1. 「ディレクトリ」タブの左側のナビゲーションツリーで example.com ノードの下にある Subscribers エントリを右クリックし、ポップアップメニューから「アクセス権を設定」を選択して、アクセス制御の管理を表示します。

2. 「新規」をクリックして、アクセス制御エディタを表示します。

3. 「ユーザー / グループ」タブの ACI 名フィールドに、「Write Subscribers」と入力します。アクセス権が与えられたユーザーのリストで、次の手順を実行します。

a. 「すべてのユーザー」を選択して削除し、「追加」をクリックします。

「ユーザーおよびグループの追加」ダイアログボックスが表示されます。

b. 「検索領域」を「特殊権限」に設定し、「検索結果」リストで「自己」を選択します。

c. 「追加」ボタンをクリックすると、アクセス権が与えられたユーザーのリストに「自己」が追加されます。

d. 「了解」をクリックして、「ユーザーおよびグループの追加」ダイアログボックスを閉じます。

4. 「権限」タブで、書き込みアクセス権 (write) のチェックボックスを選択します。これ以外のチェックボックスは、選択が解除されていることを確認してください。

5. 「ターゲット」タブで「このエントリ」をクリックすると、ターゲットディレクトリの入力フィールドにサフィックス `dc=subscribers, dc=example,dc=com` が表示されます。

a. 「サブエントリのフィルタ」フィールドに、次のフィルタを入力します。

(!(unlistedSubscriber=yes))

b. 属性テーブルで、`homePhone`、`homePostalAddress`、および `mail` 属性のチェックボックスを選択します。

ただし、これ以外のチェックボックスの選択は、解除されている必要があります。「チェックしない」ボタンをクリックしてテーブル内のすべての属性のチェックボックスの選択を解除し、次に「名前」ヘッダーをクリックしてアルファベット順に属性を並べ替えると、この作業が簡単になります。

6. ユーザーが SSL を使用して認証するように設定する場合は、「手動での編集」ボタンをクリックして、手動による編集に切り替え、次のように LDIF 文に `authmethod=ssl` を追加します。

```
(targetattr="homePostalAddress || homePhone || mail") (version
3.0; acl "Write Subscribers"; allow (write) (userdn=
"ldap:///self") and authmethod="ssl";)
```

7. 「了解」をクリックします。

「アクセス制御の管理」ウィンドウの ACI リストに、新しい ACI が追加されます。

重要なロールに対するアクセスの制限

ディレクトリ内のロール定義を使用して、業務やネットワーク、ディレクトリの管理などに含まれている重要な機能を特定できます。

たとえば、国際的な企業のサイトで特定の時間と曜日に有効なシステム管理者のサブセットを指定する superAdmin ロールを作成する必要があるかもしれません。あるいは、特定のサイト上に、応急手当のトレーニングを受けたすべてのスタッフを含む First Aid ロールの作成が必要になることもあるかもしれません。ロール定義を作成する方法については、157 ページの「ロールの割り当て」を参照してください。

ロールによって、業務上あるいはビジネス上重要な機能に関するユーザー特権を与える場合は、そのロールに対するアクセス制限を考慮する必要があります。たとえば、example.com の社員は、superAdmin ロール以外の任意のロールを個人のエントリに追加できます。これについては、「ACI「Roles」」で例を示しています。

ACI「Roles」

example.com の社員が、superAdmin 以外の任意のロールを個人のエントリに追加できるようにするには、LDIF で次のような文を作成します。

```
aci: (targetattr="*") (targattrfilters="add=nsRoleDN:(nsRoleDN !=
 "cn=superAdmin, dc=example, dc=com")") (version 3.0; acl "Roles";
 allow (write) userdn= "ldap:///self" and dns="*.example.com";)
```

この例では、ACI を ou=example-people,dc=example, dc=com エントリに追加しています。

このアクセス権を設定するには、コンソールを使用して次の手順を実行します。

1. 「ディレクトリ」タブの左側のナビゲーションツリーで example.com ノードを右クリックし、ポップアップメニューから「アクセス権を設定」を選択して、アクセス制御の管理を表示します。
2. 「新規」をクリックして、アクセス制御エディタを表示します。
3. 「ユーザー / グループ」タブの ACI 名フィールドに、「Roles」と入力します。アクセス権が与えられたユーザーのリストで、次の手順を実行します。
   a. 「すべてのユーザー」を選択して削除し、「追加」をクリックします。

      「ユーザーおよびグループの追加」ダイアログボックスが表示されます。

   b. 「ユーザーおよびグループの追加」ダイアログボックスの「検索領域」を「特殊権限」に設定し、「検索結果」リストで「自己」を選択します。
   c. 「追加」ボタンをクリックすると、アクセス権が与えられたユーザーのリストに「自己」が追加されます。
   d. 「了解」をクリックして、「ユーザーおよびグループの追加」ダイアログボックスを閉じます。
4. 「権限」タブで、書き込みアクセス権 (write) のチェックボックスを選択します。これ以外のチェックボックスは、選択が解除されていることを確認してください。

5. 「ホスト」タブの「追加」ボタンをクリックして、「ホストフィルタの追加」ダイアログボックスを表示します。「DNS ホストフィルタ」フィールドに「*.example.com」と入力します。「了解」をクリックして、ダイアログボックスを閉じます。
6. ロール用に値に基づくフィルタを作成するには、「手動での編集」ボタンをクリックして、手動による編集に切り替えます。LDIF 文の先頭に、次の文を追加します。

   ```
   (targattrfilters="add=nsRoleDN:(nsRoleDN != "cn=superAdmin,
   dc=example,dc=com")")
   ```

   追加後の LDIF 文は次のようになります。

   ```
   (targetattr="*") (targattrfilters="add=nsRoleDN:(nsRoleDN !=
   "cn=superAdmin, dc=example,dc=com")") (target =
   "ldap:///dc=example,dc=com") (version 3.0; acl "Roles"; allow
   (write) (userdn = "ldap:///self") and (dns="*.example.com");)
   ```

7. 「了解」をクリックします。

   「アクセス制御の管理」ウィンドウの ACI リストに、新しい ACI が追加されます。

サフィックスに対するすべてのアクセス権のグループへの許可

ほとんどのディレクトリには、業務上の固有の職務を特定するためのグループがあります。このグループには、ディレクトリのすべてまたは一部に対してすべてのアクセス権を与えることができます。グループにアクセス権限を与えることにより、グループメンバーに個別にアクセス権限を設定する必要がなくなります。また、グループにメンバーを追加するだけで、グループに認められたアクセス権限をそのメンバーに与えることができます。

たとえば、標準インストールプロセスを使用して Directory Server をインストールすると、ディレクトリへのすべてのアクセス権を持つ Administrators グループがデフォルトで作成されます。

example.com 社の Human Resources のグループには、ディレクトリの ou=example-people 分岐へのすべてのアクセス権が許可されています。これによって、このグループのメンバーは社員のディレクトリを更新できます。これについては、「ACI「HR」」で例を示しています。

ACI「HR」

ディレクトリの employee 分岐に対するすべての権限を HR のグループに与えるには、LDIF で次のような文を作成します。

```
aci: (targetattr="*") (version 3.0; acl "HR"; allow (all)
 userdn= "ldap:///cn=HRgroup,ou=example-people,dc=example,dc=com";)
```

この例では、ACI を ou=example-people,dc=example, dc=com エントリに追加しています。

このアクセス権を設定するには、コンソールを使用して次の手順を実行します。

1. 「ディレクトリ」タブの左側のナビゲーションツリーで example.com ノードの下にある example.com-people エントリを右クリックし、ポップアップメニューから「アクセス権を設定」を選択して、アクセス制御の管理を表示します。
2. 「新規」をクリックして、アクセス制御エディタを表示します。
3. 「ユーザー / グループ」タブの ACI 名フィールドに、「HR」と入力します。アクセス権が与えられたユーザーのリストで、次の手順を実行します。
   a. 「すべてのユーザー」を選択して削除し、「追加」をクリックします。

      「ユーザーおよびグループの追加」ダイアログボックスが表示されます。

   b. 「検索領域」を「ユーザーおよびグループ」に設定し、「検索」フィールドに「HRgroup」と入力します。

      この例は、HR のグループまたはロールがすでに作成されていることを前提としています。グループおよびロールについては、第 5 章「高度なエントリの管理」を参照してください。

   c. 「追加」ボタンをクリックすると、アクセス権が与えられたユーザーのリストに HR のグループが追加されます。
   d. 「了解」をクリックして、「ユーザーおよびグループの追加」ダイアログボックスを閉じます。
4. 「権限」タブで、「すべてチェック」ボタンをクリックします。

   プロキシ権限 (proxy) 以外のすべてのチェックボックスが選択されます。
5. 「了解」をクリックします。

   「アクセス制御の管理」ウィンドウの ACI リストに、新しい ACI が追加されます。

グループエントリの追加および削除権限の許可

一部の企業では、業務の効率化や企業全体の活力向上につながる場合、社員自身がツリー内にエントリを作成できるようにしています。

たとえば、example.com 社には、活発に活動している社内委員会があり、テニス、水泳、スキー、演劇などのさまざまなクラブが組織されています。example.com の社員は、誰でも新しいクラブのグループエントリを作成できます。これについては、「ACI「Create Group」」で例を示しています。example.com 社の社員であれば、これらのグ

ループのどれか 1 つのメンバーになることができます。これについては、240 ページの「ユーザー自身の操作によるグループへの参加と不参加」の「ACI「Group Members」」に例を示します。グループエントリの変更や削除ができるのは、グループの所有者だけです。これについては、「ACI「Delete Group」」で例を示しています。

ACI「Create Group」

example.com 社の社員が `ou=Social Committee` 分岐の下にグループエントリを作成できるようにするには、LDIF で次のような文を作成します。

```
aci: (target="ldap:///ou=social committee,dc=example,dc=com)
 (targetattr="*")(targattrfilters="add=objectClass:
 (objectClass=groupOfNames)") (version 3.0; acl "Create Group";
 allow (read,search,add) (userdn= "ldap:///uid=*,ou=example-people,
 dc=example,dc=com") and dns="*.example.com";)
```

注	この ACI は、書き込みアクセス権を与えないので、エントリを変更できません。

この例では、ACI を `ou=social committee, dc=example,dc=com` エントリに追加しています。

このアクセス権を設定するには、コンソールを使用して次の手順を実行します。

1. 「ディレクトリ」タブの左側のナビゲーションツリーで example.com ノードの下にある Social Committee エントリを右クリックし、ポップアップメニューから「アクセス権を設定」を選択して、アクセス制御の管理を表示します。
2. 「新規」をクリックして、アクセス制御エディタを表示します。
3. 「ユーザー / グループ」タブの ACI 名フィールドに、「Create Group」と入力します。アクセス権が与えられたユーザーのリストで、次の手順を実行します。
   a. 「すべてのユーザー」を選択して削除し、「追加」をクリックします。

      「ユーザーおよびグループの追加」ダイアログボックスが表示されます。

   b. 「検索領域」を「特殊権限」に設定し、「検索結果」リストで「すべての認証ユーザー」を選択します。
   c. 「追加」ボタンをクリックすると、アクセス権が与えられたユーザーのリストに「すべての認証ユーザー」が追加されます。
   d. 「了解」をクリックして、「ユーザーおよびグループの追加」ダイアログボックスを閉じます。
4. 「権限」タブで、読み取り (read)、検索 (search)、および追加 (add) のチェックボックスを選択します。これ以外のチェックボックスは、選択が解除されていることを確認してください。

5. 「ターゲット」タブで「このエントリ」をクリックすると、ターゲットディレクトリの入力フィールドにサフィックス ou=social committee, dc=example,dc=com が表示されます。
6. 「ホスト」タブの「追加」ボタンをクリックして、「ホストフィルタの追加」ダイアログボックスを表示します。「DNS ホストフィルタ」フィールドに「*.example.com」と入力します。「了解」をクリックして、ダイアログボックスを閉じます。
7. 値に基づくフィルタを作成して、社員がこのサブツリーにグループエントリだけを追加できるようにするには、「手動での編集」ボタンをクリックして、手動による編集に切り替えます。LDIF 文の先頭に、次の文を追加します。

   ```
   (targattrfilters="add=objectClass:(objectClass=groupOfNames)")
   ```

   追加後の LDIF 文は次のようになります。

   ```
   (targetattr = "*")
   (targattrfilters="add=objectClass:(objectClass=groupOfNames)")
   (target="ldap:///ou=social committee,dc=example,dc=com) (version
   3.0; acl "Create Group"; allow (read,search,add) (userdn=
   "ldap:///all") and (dns="*.example.com"); )
   ```

8. 「了解」をクリックします。

   「アクセス制御の管理」ウィンドウの ACI リストに、新しい ACI が追加されます。

ACI「Delete Group」

example.com 社の社員が ou=Social Comittee 分岐の下に所有しているグループエントリを変更または削除できるようにするには、LDIF で次のような文を作成します。

```
aci: (target="ou=social committee,dc=example,dc=com)(targetattr =
"*")
 (targattrfilters="del=objectClass:(objectClass=groupOfNames)")
 (version 3.0; acl "Delete Group"; allow (write,delete) userattr=
 "owner#GROUPDN";)
```

この例では、aci を ou=social committee, dc=example,dc=com エントリに追加しています。

コンソールを使用してこの ACI を作成すると、手動編集モードでのターゲットフィルタの作成とグループ所有権の確認が必要なので、あまり効率的ではありません。

グループまたはロールへの条件付きアクセスの許可

多くの場合、ディレクトリへのアクセス特権をグループやロールに与える場合、それらの特権が、特権ユーザーになりすました侵入者から保護されていることを確認する必要があります。したがって、多くの場合、グループまたはロールへの重要なアクセス権を与えるようなアクセス制御規則には、数多くの条件が付けられます。

たとえば、example.com 社では、ホスティングサービスの提供先企業である Company333 および Company999 に対して、それぞれ Directory Administrator ロールを作成しました。example.com 社では、侵入者からデータを保護するために、それぞれの企業が各自でデータを管理し、独自のアクセス制御規則を決定することが求められています。このため、Company333 と Company999 は、ディレクトリツリーのそれぞれの分岐に関してすべての権限を持っていますが、このアクセス権を行使するには次の条件を満たす必要があります。

- 証明書を使用して、SSL 経由の接続が認証されること
- アクセス要求は月曜日から木曜日の午前 8 時から午後 6 時までの間に限ること
- それぞれの企業に割り当てられた特定の IP アドレスからアクセスが要求されること

これらの条件は、各社の ACI である「Company333」と「Company999」に示されています。これらの ACI の内容は同等なので、「Company333」という ACI だけを次に示します。

ACI「Company333」

Company333 に対して、前述した条件に従ったディレクトリの自社の分岐へのすべてのアクセス権を与えるには、LDIF で次のような文を作成します。

```
aci:
(target="ou=Company333,ou=corporate-clients,dc=example,dc=com")
 (targetattr = "*") (version 3.0; acl "Company333"; allow (all)
 (roledn="ldap:///cn=DirectoryAdmin,ou=Company333,
 ou=corporate-clients,dc=example,dc=com") and (authmethod="ssl") and
 (dayofweek="Mon,Tues,Wed,Thu") and (timeofday >= "0800" and
 timeofday <= "1800") and (ip="255.255.123.234"); )
```

この例では、ACI を `ou=Company333, ou=corporate-clients,dc=example,dc=com` エントリに追加しています。

このアクセス権を設定するには、コンソールを使用して次の手順を実行します。

1. 「ディレクトリ」タブの左側のナビゲーションツリーで example.com ノードの下にある Company333 エントリを右クリックし、ポップアップメニューから「アクセス権を設定」を選択して、アクセス制御の管理を表示します。
2. 「新規」をクリックして、アクセス制御エディタを表示します。
3. 「ユーザー / グループ」タブの ACI 名フィールドに、「Company333」と入力します。アクセス権が与えられたユーザーのリストで、次の手順を実行します。
   a. 「すべてのユーザー」を選択して削除し、「追加」をクリックします。

      「ユーザーおよびグループの追加」ダイアログボックスが表示されます。

b. 「検索領域」を「ユーザーおよびグループ」に設定し、「検索」フィールドに「DirectoryAdmin」と入力します。

この例では、cn を DirectoryAdmin とした管理者ロールがすでに作成されていることを前提としています。

c. 「追加」ボタンをクリックすると、アクセス権が与えられたユーザーのリストに管理者ロールが追加されます。

d. 「了解」をクリックして、「ユーザーおよびグループの追加」ダイアログボックスを閉じます。

4. 「権限」タブで、「すべてチェック」ボタンをクリックします。

5. 「ターゲット」タブで「このエントリ」をクリックすると、ターゲットディレクトリの入力フィールドにサフィックス ou=Company333,ou=corporate-clients,dc=example,dc=com が表示されます。

6. 「ホスト」タブの「追加」ボタンをクリックして、「ホストフィルタの追加」ダイアログボックスを表示します。「IP アドレスホストフィルタ」フィールドに「255.255.123.234」と入力します。「了解」をクリックして、ダイアログボックスを閉じます。

ここで入力する IP アドレスは、Company333 の管理者が example.com ディレクトリに接続するために使用するホストマシンの有効な IP アドレスである必要があります。

7. 「時間」タブで、月曜日から木曜日の午前 8 時から午後 6 時に対応する時間ブロックを選択します。

テーブルの下に、選択した時間ブロックを示すメッセージが表示されます。

8. Company333 の管理者が SSL 認証を行うようにするには、「手動での編集」ボタンをクリックして、手動による編集に切り替えます。LDIF 文の末尾に次の内容を追加します。

and (authmethod="ssl")

追加後の LDIF 文は次のようになります。

```
aci: (targetattr = "*")(target="ou=Company333,
 ou=corporate-clients,dc=example,dc=com") (version 3.0; acl
 "Company333"; allow (all) (roledn="ldap:///cn=DirectoryAdmin,
 ou=Company333,ou=corporate-clients, dc=example,dc=com") and
 (dayofweek="Mon,Tues,Wed,Thu") and (timeofday >= "0800" and
 timeofday <= "1800") and (ip="255.255.123.234") and
 (authmethod="ssl"); )
```

9. 「了解」をクリックします。

「アクセス制御の管理」ウィンドウの ACI リストに、新しい ACI が追加されます。

アクセスの拒否

ディレクトリ内に業務上重要な情報が含まれている場合は、その情報へのアクセスを拒否する必要があります。

たとえば、example.com 社では、すべての契約者に対し、契約者自身のエントリにある接続時間や料金内訳などの課金情報の読み取りアクセス権を与え、書き込みアクセス権を拒否する必要があります。これについては、それぞれ「ACI「Billing Info Read」」と「ACI「Billing Info Deny」」で説明しています。

ACI「Billing Info Read」

個人のエントリ内にある課金情報の読み取りアクセス権を契約者に与えるには、LDIF で次のような文を作成します。

```
aci: (targetattr="connectionTime || accountBalance") (version 3.0;
 acl "Billing Info Read"; allow (search,read)
 userdn="ldap:///self";)
```

この例は、関連する属性がスキーマ内で作成済みであり、ACI を `ou=subscribers,dc=example,dc=com` エントリに追加しています。

このアクセス権を設定するには、コンソールを使用して次の手順を実行します。

1. 「ディレクトリ」タブの左側のナビゲーションツリーで example.com ノードの下にある Subscribers エントリを右クリックし、ポップアップメニューから「アクセス権を設定」を選択して、アクセス制御の管理を表示します。
2. 「新規」をクリックして、アクセス制御エディタを表示します。
3. 「ユーザー / グループ」タブの ACI 名フィールドに、「Billing Info Read」と入力します。アクセス権が与えられたユーザーのリストで、次の手順を実行します。
   a. 「すべてのユーザー」を選択して削除し、「追加」をクリックします。

      「ユーザーおよびグループの追加」ダイアログボックスが表示されます。

   b. 「ユーザーおよびグループの追加」ダイアログボックスの「検索領域」を「特殊権限」に設定し、「検索結果」リストで「自己」を選択します。
   c. 「追加」ボタンをクリックすると、アクセス権が与えられたユーザーのリストに「自己」が追加されます。
   d. 「了解」をクリックして、「ユーザーおよびグループの追加」ダイアログボックスを閉じます。
4. 「権限」タブで、検索 (search) と読み取り (read) の各権限のチェックボックスを選択します。これ以外のチェックボックスは、選択が解除されていることを確認してください。

5. 「ターゲット」タブで「このエントリ」をクリックすると、ターゲットディレクトリの入力フィールドにサフィックス ou=subscribers, dc=example,dc=com が表示されます。属性テーブルで、connectionTime および accountBalance 属性のチェックボックスを選択します。

   ただし、これ以外のチェックボックスの選択は、解除されている必要があります。「チェックしない」ボタンをクリックしてテーブル内のすべての属性のチェックボックスの選択を解除し、次に「名前」ヘッダーをクリックしてアルファベット順に属性を並べ替えると、この作業が簡単になります。

   この例は、スキーマに connectionTime および accountBalance 属性が追加されていることを前提としています。

6. 「了解」をクリックします。

   「アクセス制御の管理」ウィンドウの ACI リストに、新しい ACI が追加されます。

ACI「Billing Info Deny」

各契約者に対し、契約者個人のエントリ内にある課金情報の変更アクセス権を拒否するには、LDIF で次のような文を作成します。

```
aci: (targetattr="connectionTime || accountBalance") (version 3.0;
 acl "Billing Info Deny"; deny (write) userdn= "ldap:///self";)
```

この例は、関連する属性がスキーマ内で作成済みであり、ACI を ou=subscribers,dc=example,dc=com エントリに追加しています。

このアクセス権を設定するには、コンソールを使用して次の手順を実行します。

1. 「ディレクトリ」タブの左側のナビゲーションツリーで example.com ノードの下にある Subscribers エントリを右クリックし、ポップアップメニューから「アクセス権を設定」を選択して、アクセス制御の管理を表示します。
2. 「新規」をクリックして、アクセス制御エディタを表示します。
3. 「ユーザー / グループ」タブの ACI 名フィールドに、「Billing Info Deny」と入力します。アクセス権が与えられたユーザーのリストで、次の手順を実行します。
   a. 「すべてのユーザー」を選択して削除し、「追加」をクリックします。

      「ユーザーおよびグループの追加」ダイアログボックスが表示されます。

   b. 「ユーザーおよびグループの追加」ダイアログボックスの「検索領域」を「特殊権限」に設定し、「検索結果」リストで「自己」を選択します。
   c. 「追加」ボタンをクリックすると、アクセス権が与えられたユーザーのリストに「自己」が追加されます。
   d. 「了解」をクリックして、「ユーザーおよびグループの追加」ダイアログボックスを閉じます。

4. 「権限」タブで、書き込みアクセス権 (write) のチェックボックスを選択します。これ以外のチェックボックスは、選択が解除されていることを確認してください。
5. 「手動での編集」ボタンをクリックし、表示された LDIF 文の中の **allow** を **deny** に変更します。
6. 「ターゲット」タブで「このエントリ」をクリックすると、ターゲットディレクトリの入力フィールドにサフィックス ou=subscribers, dc=example,dc=com が表示されます。属性テーブルで、connectionTime および accountBalance 属性のチェックボックスを選択します。

   ただし、これ以外のチェックボックスの選択は、解除されている必要があります。「チェックしない」ボタンをクリックしてテーブル内のすべての属性のチェックボックスの選択を解除し、次に「名前」ヘッダーをクリックしてアルファベット順に属性を並べ替えると、この作業が簡単になります。

   この例は、スキーマに connectionTime および accountBalance 属性が追加されていることを前提としています。
7. 「了解」をクリックします。

   「アクセス制御の管理」ウィンドウの ACI リストに、新しい ACI が追加されます。

フィルタを使用したターゲットの設定

ディレクトリ内に分散した多数のエントリに対して、アクセス制御の設定が必要な場合は、フィルタを使用してターゲットを設定できます。ただし、検索フィルタは、アクセス管理の対象となるオブジェクトを直接指定するわけではないので、予想外のオブジェクトへのアクセスを許可または拒否してしまうことがあります。ディレクトリ構造が複雑になるほど、この問題は発生しやすくなります。さらに、フィルタによって、ディレクトリ内のアクセス制御に関する問題解決が難しくなる場合もあります。

ユーザー自身の操作によるグループへの参加と不参加

多くのディレクトリの ACI は、ユーザーが自分でグループへの参加と不参加を設定できるようになっています。これは、メーリングリストへの参加や不参加を許可する場合に便利です。

example.com 社では、社員であれば ou=social committee サブツリーの下のどのグループエントリにも参加できます。これについては、「ACI「Group Members」」で例を示しています。

ACI「Group Members」

example.com 社の社員が自分でグループへの参加や不参加を設定できるようにするには、LDIF で次のような文を作成します。

```
aci: (targettattr="member")(version 3.0; acl "Group Members";
 allow (selfwrite)
 (userdn= "ldap:///uid=*,ou=example-people,dc=example,dc=com") ;)
```

この例では、ACI を ou=social committee, dc=example,dc=com エントリに追加しています。

このアクセス権を設定するには、コンソールを使用して次の手順を実行します。

1. 「ディレクトリ」タブの左側のナビゲーションツリーで example.com ノードの下にある example-people エントリを右クリックし、ポップアップメニューから「アクセス権を設定」を選択して、アクセス制御の管理を表示します。
2. 「新規」をクリックして、アクセス制御エディタを表示します。
3. 「ユーザー / グループ」タブの ACI 名フィールドに、「Group Members」と入力します。アクセス権が与えられたユーザーのリストで、次の手順を実行します。
   a. 「すべてのユーザー」を選択して削除し、「追加」をクリックします。

      「ユーザーおよびグループの追加」ダイアログボックスが表示されます。

   b. 「ユーザーおよびグループの追加」ダイアログボックスの「検索領域」を「特殊権限」に設定し、「検索結果」リストで「すべての認証ユーザー」を選択します。
   c. 「追加」ボタンをクリックすると、アクセス権が与えられたユーザーのリストに「すべての認証ユーザー」が追加されます。
   d. 「了解」をクリックして、「ユーザーおよびグループの追加」ダイアログボックスを閉じます。
4. 「権限」タブで、書き込み (write) アクセス権のチェックボックスを選択します。これ以外のチェックボックスは、選択が解除されていることを確認してください。
5. 「ターゲット」タブのターゲットディレクトリ入力フィールドに、「dc=example,dc=com」というサフィックスを入力します。属性テーブルで、member 属性のチェックボックスを選択します。

   ただし、これ以外のチェックボックスの選択は、解除されている必要があります。「チェックしない」ボタンをクリックしてテーブル内のすべての属性のチェックボックスの選択を解除し、次に「名前」ヘッダーをクリックしてアルファベット順に属性を並べ替えると、この作業が簡単になります。
6. 「了解」をクリックします。

   「アクセス制御の管理」ウィンドウの ACI リストに、新しい ACI が追加されます。

コンマを含む DN のアクセス権の定義

DN にコンマが含まれている場合、LDIF ACI 文内で特別な処理が必要です。ACI 文のターゲット部分とバインドルール部分で、1 つの円記号 (¥) を使用して、コンマをエスケープする必要があります。次に、この構文の例を示します。

```
dn: dc=example.com Bolivia¥, S.A.,dc=com
objectClass: top
objectClass: organization
aci: (target="ldap:///dc=example.com Bolivia¥,
 S.A.,dc=com")(targetattr="*") (version 3.0; acl "aci 2"; allow
 (all) groupdn = "ldap:///cn=Directory Administrators,dc=example.com
 Bolivia¥, S.A.,dc=com";)
```

プロキシ承認を使用した ACI の例

プロキシ承認方式は、特殊な形式の認証です。自分のユーザー ID を使用してディレクトリにバインドするユーザーには、プロキシ承認を使って他のユーザーの権限が与えられます。

この例では、次の条件が満たされているものとします。

- クライアントアプリケーションのバインド DN は `"uid=MoneyWizAcctSoftware, ou=Applications,dc=example,dc=com"`
- クライアントアプリケーションがアクセスを要求するターゲットサブツリーは `ou=Accounting,dc=example,dc=com`
- ディレクトリ内に、`ou=Accounting,dc=example,dc=com` サブツリーへのアクセス権を持つ Accounting Administrator が存在する

クライアントアプリケーションが Accounting サブツリーへのアクセス権を取得するには、次の条件が満たされている必要があります (Accounting Administrator と同じアクセス権を使用)。

- Accounting Administrator は、`ou=Accounting,dc=example,dc=com` サブツリーへのアクセス権を持っている必要がある。たとえば、次の ACI は Accounting Administrator エントリに対するすべての権限を与える

  ```
  aci: (target="ldap:///ou=Accounting,dc=example,dc=com")
   (targetattr="*") (version 3.0; acl "allowAll-AcctAdmin"; allow
   (all) userdn="uid=AcctAdministrator,ou=Administrators,
   dc=example,dc=com")
  ```

- クライアントアプリケーションに対するプロキシ権限を与える次の ACI が、ディレクトリ内に存在する必要がある

```
aci: (target="ldap:///ou=Accounting,dc=example,dc=com")
 (targetattr="*") (version 3.0; acl "allowproxy-
 accountingsoftware"; allow (proxy) userdn=
 "uid=MoneyWizAcctSoftware,ou=Applications,dc=example,dc=com")
```

この ACI が設定されていれば、MoneyWizAcctSoftware クライアントアプリケーションがディレクトリにバインドし、プロキシ DN のアクセス権限を要求する `ldapsearch` や `ldapmodify` などの LDAP コマンドを送信できます。

前述の例で、クライアントが `ldapsearch` コマンドを実行する場合は、このコマンドに次の制御が含まれます。

```
# ldapsearch -w password ¥
-D "uid=MoneyWizAcctSoftware,ou=Applications,dc=example,dc=com" ¥
-y "uid=AcctAdministrator,ou=Administrators,dc=example,dc=com" ...
```

クライアントはそのままバインドしますが、プロキシエントリの特権が与えられます。クライアントには、プロキシエントリのパスワードは必要ありません。

注 Directory Manager の DN をプロキシ DN として使用することはできません。また、Directory Manager にプロキシ権限を与えることはできません。同じバインド操作中に Directory Server が複数のプロキシ認証を受け取った場合は、クライアントアプリケーションにエラーが返され、バインド試行は失敗します。

実効権限の表示

ディレクトリのエントリに対するアクセスポリシーを管理するとき、定義した ACI がセキュリティに与える影響を知ることができれば非常に役立ちます。Sun ONE Directory Server 5.2 には、既存の ACI を評価して、特定のユーザーが特定のエントリに対して持つ実効権限について報告するという新しいメカニズムがあります。

この新しい実行権限の取得制御は検索操作で使用され、Directory Server はそれを処理します。この制御に対する応答として、エントリと属性に対する実効権限の情報が検索結果の中で返されます。この追加情報としては、各エントリとその中の各属性に対する読み取り権限および書き込み権限などがあります。検索に使用されるバインド DN や任意の DN では権限が必要とされることがあるので、管理者はディレクトリユーザーの権限を検査できます。

警告 実効権限を表示することはそれ自体がディレクトリ操作なので、適切に保護し、制限する必要があります。`aclRights` 属性と `aclRightsInfo` 属性に対する ACI を追加で作成して、ディレクトリユーザーによるこの情報へのアクセスを制限します。

実効権限を表示する機能は、LDAP 制御を利用しています。連鎖サフィックスに対する実効権限を表示するには、連鎖ポリシーの中でこの制御を有効にする必要があります。詳細は、115 ページの「連鎖ポリシーの設定」を参照してください。また、リモートサーバーとのバインドに使用されるプロキシ ID にも、実効権限の属性へのアクセスが許可されていることを確認してください。

実行権限の取得制御の使用

実行権限の取得制御を指定するには、`ldapsearch` コマンドに `-J "1.3.6.1.4.1.42.2.27.9.5.2"` オプションを指定して実行します。デフォルトでは、エントリと属性に対してバインド DN エントリが持っている実効権限が検索結果の中で返されます。デフォルトの動作を変更するには、次のオプションを使用します。

- `-c "dn: `*DN*`"` : 検索結果には、指定された *DN* にバインドされているユーザーの実効権限が表示されます。管理者はこのオプションを使用して、別のユーザーの実効権限を確認できます。`-c "dn:"` オプションを指定すると、匿名認証用の実効権限が表示されます。

- `-X "`*attributeName ...*`"` : 検索結果には、指定された属性に対する実効権限も表示されます。このオプションは、検索結果に表示されない属性を指定する場合に使用します。たとえば、このオプションを使用すると、現在はエントリに存在していない属性について、ユーザーがその属性を追加する権限を持っているかどうかを調べることができます。

`-c` 属性または `-X` 属性、あるいはその両方を使用するときは、`-J` オプションに実行権限の取得制御の OID が暗黙的に指定されるため、このオプションを指定する必要はありません。

次に、表示する情報の種類を選択する必要があります。権限だけを表示するか、権限がどのように許可または拒否されているかを示す詳細なログ情報を表示できます。情報の種類を指定するには、検索結果で返す属性として `aclRights` または `aclRightsInfo;logs` を追加します。両方の属性を要求すると、実効権限の情報をすべて取得できます。ただし、単純な権限情報は詳細なログ情報にも含まれています。

注 `aclRights` 属性と `aclRightsInfo;logs` 属性は、仮想オペレーショナル属性として動作します。これらの属性はディレクトリには格納されず、これらを取得するには明示的に要求する必要があります。これらの属性は、実行権限の取得制御に対する応答として Directory Server で生成されます。

このため、どちらの属性も、フィルタや何らかの検索操作に使用することはできません。

次の例は、ユーザーがディレクトリでの自身の権限を確認する方法を示しています。結果の中で、`1` は権限が与えられていることを示し、`0` は拒否されていることを示します。

```
% ldapsearch -J "1.3.6.1.4.1.42.2.27.9.5.2" ¥
             -h rousseau.example.com -p 389 ¥
             -D "uid=cfuente,ou=People,dc=example,dc=com" ¥
             -w password -b "dc=example,dc=com" ¥
             "(objectclass=*)" aclRights

dn: dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: ou=Groups, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: cn=Accounting Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
```

```
dn: cn=HR Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: uid=bjensen,ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: uid=cfuente, ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:1,proxy:0
```

この結果は、Carla Fuente にはディレクトリ内のエントリに少なくとも読み取り権限が与えられていて、自分のエントリを変更できることを示しています。実効権限制御で指定されていないと通常のアクセス権は有効ではないため、ユーザーは読み取り権限が与えられていないエントリを見ることはできません。次の例で、Directory Manager は、Carla Fuente に読み取り権限が与えられていないエントリを確認できます。

```
% ldapsearch -h rousseau.example.com -p 389 ¥
             -D "cn=Directory Manager" -w password ¥
             -c "dn: uid=cfuente,ou=People,dc=example,dc=com" ¥
             -b "dc=example,dc=com" ¥
             "(objectclass=*)" aclRights

dn: dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: ou=Groups, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: cn=Directory Administrators, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:0,write:0,proxy:0

dn: ou=Special Users,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:0,write:0,proxy:0

dn: ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: cn=Accounting Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: cn=HR Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: uid=bjensen,ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0

dn: uid=cfuente, ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:1,proxy:0
```

上記の出力で、Directory Manager は、Carla Fuente がディレクトリツリーの Special Users 分岐と Directory Administrators 分岐のどちらも表示できないことを確認できます。次の例では、Directory Manager は、Carla Fuente が自身のエントリの `mail` 属性と `manager` 属性を変更できないことを確認できます。

```
% ldapsearch -h rousseau.example.com -p 389 ¥
              -D "cn=Directory Manager" -w password ¥
              -c "dn: uid=cfuente,ou=People,dc=example,dc=com" ¥
              -b "dc=example,dc=com" ¥
              "(uid=cfuente)" aclRights "*"

version: 1
dn: uid=cfuente, ou=People, dc=example,dc=com

aclRights;attributeLevel;mail: search:1,read:1,compare:1,
 write:0,selfwrite_add:0,selfwrite_delete:0,proxy:0
mail: cfuente@example.com

aclRights;attributeLevel;uid: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
uid: cfuente

aclRights;attributeLevel;givenName: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
givenName: Carla

aclRights;attributeLevel;sn: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
sn: Fuente

aclRights;attributeLevel;cn: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
cn: Carla Fuente

aclRights;attributeLevel;userPassword: search:0,read:0,
 compare:0,write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
userPassword: {SSHA}wnbWHIq2HPiY/5ECwe6MWBGx2KMiZ8JmjF80Ow==

aclRights;attributeLevel;manager: search:1,read:1,compare:1,
 write:0,selfwrite_add:0,selfwrite_delete:0,proxy:0
manager: uid=bjensen,ou=People,dc=example,dc=com

aclRights;attributeLevel;telephoneNumber:
search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
telephoneNumber: (234) 555-7898
```

```
aclRights;attributeLevel;objectClass: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson

aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
```

aclRights 属性と aclRightsInfo;logs 属性の形式については、『Sun ONE Directory Server Deployment Guide』の第 7 章にある「Understanding the Effective Rights Results」を参照してください。

高度なアクセス制御 : マクロ ACI の使用

同じようなディレクトリツリー構造をいくつも持つ組織では、マクロによってディレクトリ内で使用する ACI の数を最適化できます。ディレクトリツリー内の ACI の数を減らすことによって、アクセス制御ポリシーの管理が簡単になり、ACI によるメモリ使用の効率が向上します。

マクロは、ACI の中で DN、または DN の一部を表現するために使用される可変部分です。マクロを使用すると、ACI のターゲット部分またはバインドルール部分、あるいはその両方の DN を表すことができます。実際の処理では、Directory Server が LDAP 操作を受け取ると、一致する部分文字列の存在を確認するために、LDAP 操作のターゲットとなるリソースに対して ACI マクロのマッチングが行われます。一致が検出された場合は、一致した部分文字列を使ってバインドルール側のマクロが展開され、その展開バインドルールを評価してリソースにアクセスします。

マクロ ACI の例

マクロ ACI の利点ともっとも効果的に機能させる方法を例を示しながら説明します。250 ページの図 6-4 は、全体的な ACI の数を減らすために、マクロ ACI を効果的に利用しているディレクトリツリーです。

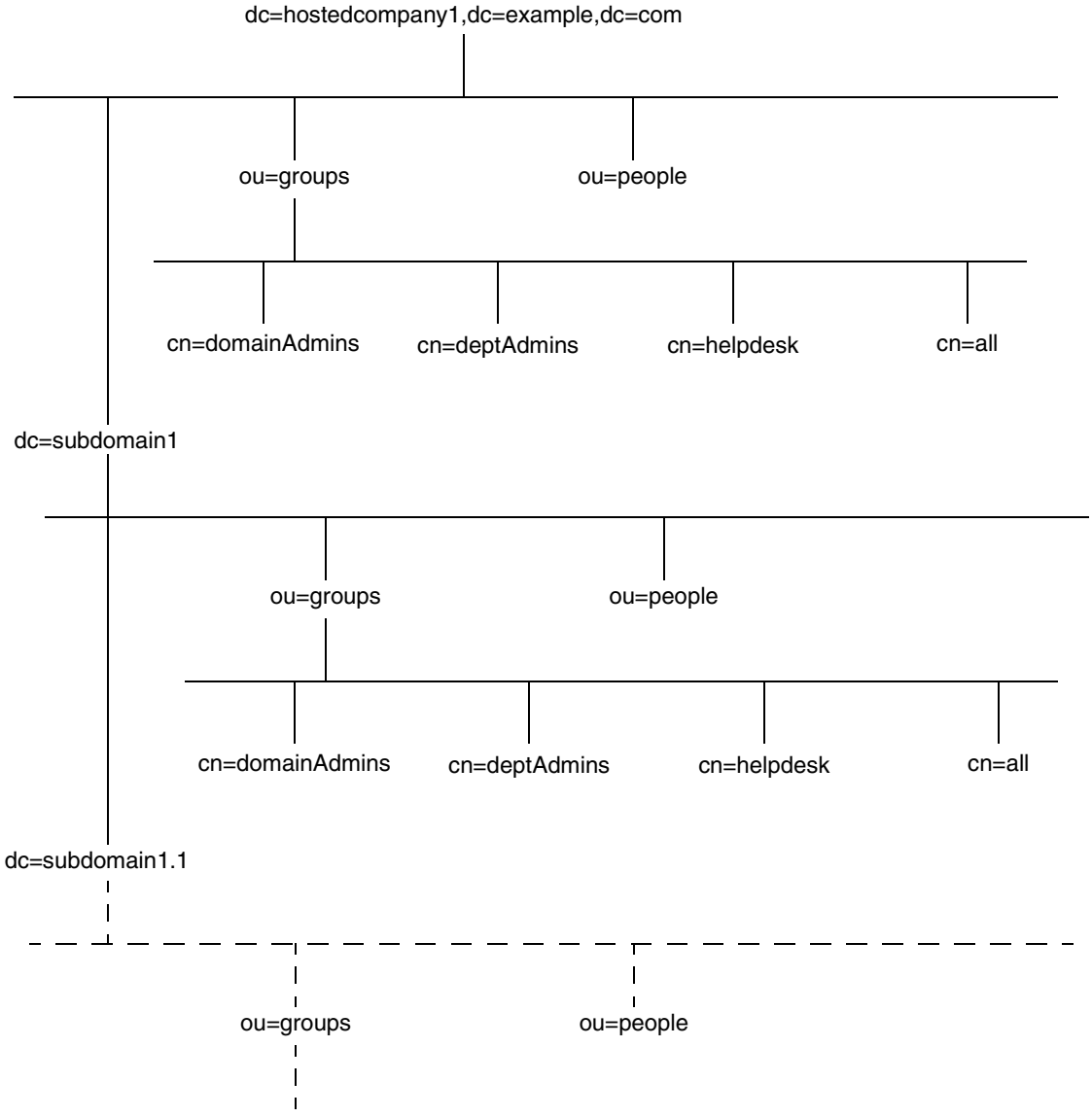
この例では、同じツリー構造のサブドメインが同じパターンで繰り返されています (ou=groups, ou=people)。example.com ディレクトリツリーには、サフィックス dc=hostedCompany2,dc=example,dc=com および dc=hostedCompany3,dc=example,dc=com が格納されているので、このパターンはツリー内でも繰り返されています。

ディレクトリツリーに適用される ACI でも、同じパターンが繰り返されています。たとえば、次の ACI は dc=hostedCompany1,dc=example,dc=com ノード上に置かれています。

```
aci: (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search) groupdn=
 "ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany1,dc=example,
 dc=com";)
```

この ACI は、dc=hostedCompany1,dc=example,dc=com ツリー内のすべてのエントリに対する読み取り権限および書き込み権限を DomainAdmins グループに与えます。

図 6-4　マクロ ACI のディレクトリツリーの例



次の ACI は、`dc=hostedCompany1,dc=example,dc=com` ノード上に置かれています。

```
aci: (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany1,
 dc=example,dc=com";)
```

次の ACI は、dc=subdomain1,dc=hostedCompany1, dc=example,dc=com ノード上に置かれています。

```
aci: (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=subdomain1,
 dc=hostedCompany1,dc=example,dc=com";)
```

次の ACI は、dc=hostedCompany2,dc=example,dc=com ノード上に置かれています。

```
aci: (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany2,
 dc=example,dc=com";)
```

次の ACI は、dc=subdomain1,dc=hostedCompany2, dc=example,dc=com ノード上に置かれています。

```
aci: (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups, dc=subdomain1,
 dc=hostedCompany2,dc=example,dc=com";)
```

前述の 4 つの ACI の違いは、groupdn キーワード内で指定されている DN だけです。DN 用のマクロを使用することによって、これらの ACI を、ルートツリーの dc=example,dc=com ノードに置かれた 1 つの ACI で置き換えることができます。この ACI は次のようになります。

```
aci: (target="ldap:///ou=Groups,($dn),dc=example,dc=com")
 (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search) groupdn=
 "ldap:///cn=DomainAdmins,ou=Groups,[$dn],dc=example,dc=com";)
```

ターゲットキーワードが未使用の場合は、これを設定する必要があります。

前述の例では、ACI の数が 4 つから 1 つに減っています。ただし、本当の利点は、ディレクトリツリー全体に複数の繰り返しパターンを含めることができることです。

マクロ ACI の構文

ここでは、わかりやすくするために、userdn、roledn、groupdn、userattr などのバインド資格を与えるために使用される ACI キーワードをまとめてサブジェクトと呼びます。サブジェクトは、ACI の適用対象を決定します。

マクロ ACI では、次のような式を使用して、DN または DN の一部を置き換えることができます。

- (`$dn`) - ターゲット内のマッチングと、サブジェクト内の直接置換
- [`$dn`] - サブジェクトのサブツリーで機能する複数の RDN の置換
- (`$attr.`*attributeName*) - ターゲットエントリの *attributeName* 属性からサブジェクトへの置換

DN マクロを使用できる ACI の場所を表 6-3 に示します。

表 6-3　ACI キーワード中のマクロ

マクロ	ACI キーワード
(`$dn`)	target、targetfilter、userdn、roledn、groupdn、userattr
[`$dn`]	targetfilter、userdn、roledn、groupdn、userattr
(`$attr.`*attrName*)	userdn、roledn、groupdn、userattr

この場合、次のような制限があります。

- サブジェクトで (`$dn`) マクロおよび [`$dn`] マクロを使用するときは、(`$dn`) マクロを含むターゲットを定義する必要があります。
- サブジェクトで (`$attr.`*attrName*) マクロと (`$dn`) マクロを組み合わせることができますが、[`$dn`] マクロを組み合わせることはできません。

ターゲットでの ($dn) との一致

ACI のターゲットに含まれる (`$dn`) マクロは、LDAP 要求のターゲットとなるエントリとの比較によって、置換値を決定します。たとえば、`cn=all,ou=groups,dc=subdomain1, dc=hostedCompany1,dc=example,dc=com` エントリをターゲットとする LDAP 要求がある場合は、ターゲットを定義する ACI は次のようになります。

```
(target="ldap:///ou=Groups,($dn),dc=example,dc=com")
```

この場合、(`$dn`) マクロは `dc=subdomain1, dc=hostedCompany1` と一致します。この部分文字列は、ACI のサブジェクト内で置換値として使用されます。

サブジェクト内での ($dn) の置換

ACI のサブジェクト内では、(`$dn`) マクロはターゲット内で一致する部分文字列全体に置き換えられます。次に例を示します。

```
groupdn="ldap:///cn=DomainAdmins,ou=Groups,($dn),dc=example,dc=com"
```

これは次のようになります。

```
groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=subdomain1,
 dc=hostedCompany1,dc=example,dc=com"
```

マクロが展開されると、通常のプロセスに続いて Directory Server が ACI を評価し、アクセス権が与えられるかどうかを決定します。

注 標準の ACI とは異なり、マクロ置換を使った ACI はターゲットエントリの子へのアクセスを許可する必要はありません。これは、子の DN がターゲットとなった場合に、置換によってサブジェクト文字列内に有効な DN が作成されない可能性があるためです。

サブジェクト内での [$dn] の置換

`[$dn]` の置換メカニズムは (`$dn`) のものと少し異なります。ターゲットリソースの DN は数回にわたって確認されますが、一致する対象が見つかるまで、一番左にある RDN コンポーネントは外されます。

たとえば、`cn=all,ou=groups,dc=subdomain1,dc=hostedCompany1,dc=example,dc=com` サブツリーをターゲットとする LDAP 要求で、次のような ACI があるとします。

```
aci: (targetattr="*") (target="ldap:///ou=Groups,($dn),dc=example,
 dc=com") (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,[$dn],dc=example,dc=com
";)
```

サーバーは次のように処理を続け、この ACI を展開します。

1. ターゲットの (`$dn`) が `dc=subdomain1,dc=hostedCompany1` に一致します。
2. サブジェクトの `[$dn]` を `dc=subdomain1,dc=hostedCompany1` で置き換えます。

   この結果、サブジェクトは `groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=subdomain1,dc=hostedCompany1,dc=example,dc=com"` になります。バインド DN がそのグループのメンバーであるためにアクセスが許可される場合は、マクロの展開は中止され、ACI が評価されます。メンバーでない場合は、プロセスが続行されます。

3. サブジェクトの [$dn] を dc=hostedCompany1 で置き換えます。

   この結果、サブジェクトは groupdn="ldap:///cn=DomainAdmins,ou=Groups, dc=hostedCompany1,dc=example,dc=com" になります。バインド DN がこのグループのメンバーであるかどうかが再び検証され、メンバーである場合は ACI が完全に評価されます。メンバーでない場合は、マクロの展開は一致した値の最後の RDN で中止され、この ACI の ACI 評価は完了します。

[$dn] マクロの利点は、ドメインレベルの管理者に対して、ディレクトリツリー内のすべてのサブドメインへのアクセス権を柔軟な方法で与えることができることです。したがって、このマクロは、ドメイン間の階層的な関係を表す場合に便利です。

たとえば、次のような ACI があるとします。

```
aci: (target="ldap:///ou=*, ($dn),dc=example,dc=com")
 (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search) groupdn=
 "ldap:///cn=DomainAdmins,ou=Groups,[$dn],dc=example,dc=com";)
```

この ACI は、cn=DomainAdmins,ou=Groups, dc=hostedCompany1,dc=example,dc=com のすべてのメンバーに対して、dc=hostedCompany1 の下にあるすべてのサブドメインへのアクセス権を与えます。したがって、たとえばそのグループに属する管理者は、サブツリー ou=people,dc=subdomain1.1,dc=subdomain1 にアクセスできます。

ただし、同時に、cn=DomainAdmins,ou=Groups, dc=subdomain1.1 のメンバーの ou=people,dc=subdomain1, dc=hostedCompany1 および ou=people,dc=hostedCompany1 ノードに対するアクセスは拒否されます。

($attr.*attrName*) に対するマクロマッチング

($attr.*attrname*) マクロは、常に DN のサブジェクト部分で使用されます。たとえば、次のような roledn を定義できます。

```
roledn = "ldap:///cn=DomainAdmins,($attr.ou),dc=HostedCompany1,
dc=example,dc=com"
```

ここで、サーバーが次のエントリをターゲットとする LDAP 操作を受け取ったとします。

```
dn: cn=Babs Jensen,ou=People,dc=HostedCompany1, dc=example,dc=com
cn: Babs Jensen
sn: Jensen
ou: Sales
...
```

ACI の roledn の部分を評価するために、サーバーはターゲットエントリの ou 属性の値を読み取り、サブジェクト内でこの値を置換してマクロを展開します。この例では、roledn は次のように展開されます。

```
roledn = "ldap:///cn=DomainAdmins,ou=Sales,dc=HostedCompany1,
dc=example,dc=com"
```

続いて、通常の ACI 評価アルゴリズムに従って、Directory Server が ACI を評価します。

マクロ内で名前が指定された属性が複数の値を持つ場合は、それぞれの値を使用してマクロが展開され、最初にマッチングに成功した値が使用されます。

アクセス制御とレプリケーション

ACI は、エントリの属性として格納されます。したがって、レプリケートされるサフィックスの一部に ACI を含むエントリがあれば、ほかの属性と同じように ACI もレプリケートされます。

ACI の評価は、着信 LDAP 要求を実行する Directory Server 上で行われます。つまり、コンシューマサーバーが更新要求を受け取ると、その要求がマスター上で実行されるかどうかを評価する前に、コンシューマサーバーがマスターサーバーにリフェラルを返します。

アクセス制御情報のログ

エラーログに記録されているアクセス制御に関する情報を取得するには、適切なログレベルを設定する必要があります。

コンソールからエラーログレベルを設定するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「ディレクトリ」タブで cn=config ノードをマウスの右ボタンでクリックし、ポップアップメニューから「汎用エディタで編集」を選択します。

   cn=config エントリの内容を表示した状態で汎用エディタが起動されます。

2. 属性値の組み合わせリストをスクロールして、nsslapd-errorlog-level 属性を探します。

3. nsslapd-errorlog-level フィールドに表示されている値に 128 を加えます。

   たとえば、8192 (レプリケーションデバッグ) という値が表示されている場合は、8320 に変更します。エラーログレベルについては、『Sun ONE Directory Server Reference Manual』を参照してください。

4. 「了解」をクリックして変更を保存し、汎用エディタを閉じます。

以前のリリースとの互換性

Directory Server の以前のリリースで使用されていた一部の ACI キーワードは、Sun ONE Directory Server 5.2 ではお勧めできません。ただし、下位互換の観点から、これらのキーワードも引き続きサポートされています。対象となるキーワードを次に示します。

- `userdnattr`
- `groupdnattr`

このため、旧バージョンのサプライヤサーバーと Directory Server 5.2 のコンシューマの間にレプリケーションアグリーメントを設定する場合でも、ACI のレプリケーションに関する問題が発生することはありません。

ただし、207 ページの「値マッチングに基づくアクセスの定義」で説明している手順に従って、これらのキーワードを `userattr` キーワードの機能に置き換えることをお勧めします。

第 7 章

ユーザーアカウントの管理

ユーザーが Directory Server に接続すると、ユーザー認証が行われ、ディレクトリは認証中に確立された識別情報に基づいて、アクセス権限とリソース制限をユーザーに設定します。

この章では、ユーザーアカウントを管理するための作業について説明します。これらのタスクは、ディレクトリのパスワードおよびアカウントのロックアウトポリシーの設定、ディレクトリに対するアクセスを防止するためのアカウントまたはグループの無効化、およびバインド DN に応じたユーザーのシステムリソースの使用制限などを行います。

Directory Server 5.2 では、個別のパスワードポリシーに対応できるようになりました。異なるパスワードポリシーをいくつでも定義し、特定のユーザーやユーザーグループに個別に適用できます。これにより、異なるタイプのユーザーによるディレクトリへのアクセスを簡単に制御できます。

この章は、次の節で構成されています。

- パスワードポリシーの概要
- グローバルパスワードポリシーの設定
- 個別パスワードポリシーの管理
- ユーザーパスワードのリセット
- ユーザーとロールの無効化と有効化
- 個別のリソース制限の設定

パスワードポリシーの概要

安全なパスワードポリシーを使用して、次の項目を義務付けることによって、簡単に推測されそうなパスワードに関するリスクを最小限に抑えることができます。

- スケジュールに従ったパスワードの変更
- 推測しにくいパスワードの使用
- 不正なパスワードで複数回バインドしようとした場合のアカウントのロック

Directory Server 5.2 では、個別パスワードポリシーとグローバルパスワードポリシーの両方を利用できます。個別パスワードポリシーはディレクトリツリーのサブエントリによって定義され、そのポリシーが設定されたユーザーエントリによって参照されます。ユーザーエントリが個別ポリシーを参照しない場合、`cn=PasswordPolicy,cn=config` のグローバルパスワードポリシーが適用されます。

次の節では、パスワードポリシーの実装方法と、ユーザーおよびグループへの割り当て方法について説明します。詳細については、『Sun ONE Directory Server Deployment Guide』の第 7 章にある「Designing Your Password Policies」を参照してください。

辞書攻撃の防止

辞書攻撃では、侵入者は認証が得られるまで繰り返しパスワードを推測して解読しようとします。このような攻撃に対応するために、サーバーには 3 つのツールが用意されています。

- パスワード構文検査では、ユーザーエントリの `uid`、`cn`、`sn`、`givenName`、`ou`、または `mail` 属性の一致を検証します。いずれかの値と一致する場合、サーバーはユーザーによるパスワードの設定を拒否します。ただしパスワード構文検査では、`/usr/dict/words` の単語をすべて試すような実際の辞書攻撃は阻止されません。
- パスワードに最小限必要な文字数を設定すると、ユーザーは短いパスワードを設定できません。パスワードが長くなるほど、すべての値を想像したり、組み合わせることが指数関数的に困難になります。Directory Server では、パスワード構文検査と最小限必要な文字数の設定の両方を有効にする必要があります。
- アカウントロックアウトメカニズムは、認証の試みが数回失敗した後にバインドを拒否します。パスワードポリシーの厳密度に応じて、一時的なロックアウトと永続的なロックアウトのいずれかが適用されます。

いずれも、自動的なパスワードの推測を効果的に防止します。たとえば、5 回までの試行を許可してその後にユーザーアカウントを 5 分間ロックアウトした場合、平均すると侵入者は 1 分間に 1 回の推測しか行えず、正規のユーザーが入力ミスなどでロックアウトされた場合も短時間で再試行できます。永続的なロックアウトの場合は、Directory Manager から手動でパスワードをリセットする必要があります。

レプリケーション環境でのパスワードポリシー

個別のパスワードポリシー、グローバルパスワードポリシーのどちらもレプリケーションが可能です。これにより、パスワードポリシーをマスターで定義し、レプリケーションによってレプリケートされたサーバーにポリシーを伝達させることができます。設定したすべての属性は、パスワード履歴 (すでに使用された古いパスワード) とパスワードの有効期限を含むオペレーショナル属性としてレプリケートされます。

ただし、レプリケートされた環境では、パスワードポリシーによる次の影響を考慮する必要があります。

- パスワードの期限切れが近づいたユーザーは、パスワードを変更するまで、バインドするすべてのレプリカから警告を受信する
- ユーザーがパスワードを変更すると、すべてのレプリカでパスワード変更の情報が更新されるまでに時間がかかる。ユーザーがパスワードを変更し、すぐに新しいパスワードでコンシューマレプリカのどれかに再度バインドしようとすると、レプリカが更新されたパスワードを受信するまでは、バインドに失敗する
- 各レプリカには、レプリケートされない個別のアカウントロックアウトカウンタが保持されている。その結果、ロックアウトポリシーはどれか 1 つのレプリカで適用されるが、ユーザーが複数のレプリカでバインドを試みるとポリシーが適用されないことがある。たとえば、レプリケーショントポロジに 10 のサーバーがあり、3 回の試行後にロックアウトが有効になる場合、侵入者はパスワードの推測を 30 回行える計算になる

レプリケーションによって侵入者が推測できるパスワードの数は増えますが、推測される無数の組み合わせと比較すれば、影響はほとんどありません。それ以上に、パスワード検査を有効にし、6 文字以上のパスワードを設定して、強度の高いパスワードをユーザーに強制することのほうが重要です。また、辞書に登場するような一般的な単語ではないパスワードを選び、記憶する方法について、ユーザーにガイドラインを示すことも必要です。最後に、すべてのディレクトリ管理者ユーザーが強力なパスワードを持たなければならないことは言うまでもありません。

グローバルパスワードポリシーの設定

グローバルパスワードポリシーは、個別ポリシーが定義されていない、ディレクトリのすべてのユーザーに適用されます。ただし、グローバルパスワードポリシシーは Directory Manager には適用されません。

コンソールを使用したパスワードポリシーの設定

Directory Server のグローバルパスワードポリシーを設定または変更するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで「データ」ノードを選び、右側のパネルで「パスワード」タブを選びます。
2. 「パスワード」タブで、ポリシーに関する次の項目を設定します。

- 「リセット後、ユーザーにパスワード変更を要求」チェックボックスを選択して、初回ログオン時に、ユーザーがパスワードを変更しなければならないように指定します。

  このチェックボックスを選択した場合、ユーザーのパスワードのリセットは Directory Manager だけができます。一般の管理者は、パスワードの更新をユーザーに義務付けることはできません。

- 各ユーザーが、自分のパスワードを変更できるようにするには、「ユーザーによるパスワードの変更可」チェックボックスを選択します。
- ユーザーが本人のパスワードを変更できる頻度を制限するには、「再変更を許可」テキストボックスに日数を指定します。制限を設けずにユーザーが自由に本人のパスワードを変更できるようにするには、「制限なし」チェックボックスを選択します。
- ユーザーが同じパスワードを何度も繰り返して利用することを禁じるには、「パスワードの履歴を保持」チェックボックスを選択し、ユーザーごとにサーバーがいくつのパスワードを保持するかを指定します。ユーザーは、リストに残されているパスワードを設定することができません。効率的に機能させるには、ユーザーがパスワードを変更する頻度を制限する必要があります。
- ユーザーのパスワードを無期限にする場合は、「パスワードは永久に失効しない」ラジオボタンを選択します。
- ユーザーにパスワードを定期的に変更させる場合は、「パスワードの失効まで」ラジオボタンを選択し、パスワードの有効日数を入力します。

- 有効期限が近づいたパスワードを選択した場合に、期限が切れる何日前に警告をユーザーに送信するかを指定するには、「警告を送信」フィールドに日数を指定します。

  ユーザーが警告を受信すると、パスワードは当初の期日に失効します。警告の送信後、警告期間の経過後に有効期限を延長するには、「失効の警告を行わない」チェックボックスの選択を解除します。警告と延長は、それぞれ 1 回ずつ行われます。パスワードが失効した後にユーザーがバインドしたときは、猶予ログインは認められなくなります。

- ユーザーパスワードの構文を検査して、パスワードポリシーで設定した要件を満たしていることをサーバー側で確認する場合は、「パスワード構文を検査」チェックボックスを選択します。次に、「パスワードの最低長」テキストボックスに、最小限必要なパスワードの文字数を指定します。

- デフォルトでは、Directory Manager はパスワードポリシーに違反するパスワード (履歴に残されているパスワードの再使用など) をリセットできません。これを許可するには、「Directory Manager がパスワードポリシーをバイパスする」チェックボックスを選択します。

- 「パスワードの暗号化」プルダウンメニューで、パスワードの格納時にサーバーで使用する暗号化方式を指定します。

3. アカウントのロックアウトポリシーを定義するときは、「アカウントのロックアウト」タブをクリックし、「アカウントのロックアウト機能を利用」チェックボックスを選択します。

- どれだけの時間に何回のログインが失敗した場合にロックアウトを有効にするか、回数と時間を指定します。

- Directory Manager がユーザーパスワードをリセットするまで永続的にロックアウトするには、「無期限にロックアウト」ラジオボタンを選択します。

- それ以外の場合は「ロックアウト時間」ラジオボタンを選択し、ユーザーアカウントが一時的にロックアウトされる時間を分単位で指定します。

4. パスワードポリシーの変更が完了したら、「保存」をクリックします。新しいグローバルパスワードポリシーは、直ちに適用されます。

コマンド行からのパスワードポリシーの設定

グローバルパスワードポリシーは、`cn=Password Policy, cn=config` エントリの属性によって定義されます。このエントリ内のグローバルポリシーを変更するには、`ldapmodify` ユーティリティを使います。

パスワードポリシーに関連するすべての属性の説明については、『Sun ONE Directory Server Reference Manual』の第 4 章にある「cn=Password Policy」を参照してください。

たとえば、デフォルトではパスワードの構文の長さの検査は行われず、アカウントロックアウトも無効になっています。構文検査を有効にして最小文字数を 8 に設定し、5 回のパスワード入力に失敗した場合に 5 分間の一時的なロックアウトを設定するには、次のコマンドを実行します。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=Password Policy,cn=config
changetype: modify
replace: passwordCheckSyntax
passwordCheckSyntax: on
-
replace: passwordMinLength
passwordMinLength: 8
-
replace: passwordLockout
passwordLockout: on
-
replace: passwordMaxFailure
passwordMaxFailure: 5
-
replace: passwordLockoutDuration
passwordLockoutDuration: 300
-
replace: passwordUnlock
passwordUnlock: on
```

個別パスワードポリシーの管理

個別パスワードポリシーは、passwordPolicy オブジェクトクラスを使ってサブエントリに定義されます。ポリシーは、cn=*policy name*,*subtree* という形式の DN でディレクトリツリー内の任意の場所に定義できます。Directory Server コンソールまたはコマンド行ユーティリティを使ってパスワードポリシーを定義したら、該当するユーザーエントリの passwordPolicySubentry 属性を設定してパスワードポリシーを割り当てます。

ここでは、サブツリールートが dc=example,dc=com の Example.com の一時従業員にパスワードポリシーを実装する例を使います。

コンソールからのポリシーの定義

1. Directory Server コンソールの最上位にある「ディレクトリ」タブで、個別パスワードポリシーのサブエントリを定義するエントリを表示します。
2. エントリをマウスの右ボタンでクリックし、「新規」、「パスワードポリシー」を順に選択します。あるいは、エントリをクリックして選択し、「オブジェクト」メニューから「新規」、「パスワードポリシー」を順に選択します。

   パスワードポリシーエントリのカスタムエディタが表示されます。
3. このポリシーの名前と、オプションとしてポリシーの説明を「一般」の「ポリシー名」と「説明」フィールドに入力します。この名前は、ポリシーを定義するサブエントリの cn ネーミング属性の値となります。
4. 「パスワード」タブをクリックし、ポリシーに関する次の項目を設定します。

- 「リセット後、ユーザーにパスワード変更を要求」チェックボックスを選択して、初回ログオン時に、ユーザーがパスワードを変更しなければならないように指定します。

  このチェックボックスを選択した場合、ユーザーのパスワードのリセットは Directory Manager だけができます。一般の管理者は、パスワードの更新をユーザーに義務付けることはできません。
- 各ユーザーが、自分のパスワードを変更できるようにするには、「ユーザーによるパスワードの変更可」チェックボックスを選択します。
- ユーザーが本人のパスワードを変更できる頻度を制限するには、「再変更を許可」テキストボックスに日数を指定します。制限を設けずにユーザーが自由に本人のパスワードを変更できるようにするには、「制限なし」チェックボックスを選択します。

- ユーザーが同じパスワードを何度も繰り返して利用することを禁じるには、「パスワードの履歴を保持」チェックボックスを選択し、ユーザーごとにサーバーがいくつのパスワードを保持するかを指定します。ユーザーは、リストに残されているパスワードを設定することができません。効率的に機能させるには、ユーザーがパスワードを変更する頻度を制限する必要があります。
- ユーザーのパスワードを無期限にする場合は、「パスワードは永久に失効しない」ラジオボタンを選択します。
- ユーザーにパスワードを定期的に変更させる場合は、「パスワードの失効まで」ラジオボタンを選択し、パスワードの有効日数を入力します。
- 有効期限が近づいたパスワードを選択した場合に、期限が切れる何日前に警告をユーザーに送信するかを指定します。パスワードの有効期限の何日前に警告を送信するかを「警告を送信」テキストボックスに入力します。

  ユーザーが警告を受信すると、パスワードは当初の期日に失効します。警告の送信後に完全な警告期間を設けて有効期限を延長するには、「失効の警告を行わない」チェックボックスの選択を解除します。警告と延長は、それぞれ 1 回ずつ行われます。パスワードが失効した後にユーザーがバインドしたときは、猶予ログインは認められなくなります。
- ユーザーパスワードの構文を検査して、パスワードポリシーで設定した要件を満たしていることをサーバー側で確認する場合は、「パスワード構文を検査」チェックボックスを選択します。次に、「パスワードの最低長」テキストボックスに、最小限必要なパスワードの文字数を指定します。
- デフォルトでは、Directory Manager はパスワードポリシーに違反するパスワード (履歴に残されているパスワードの再使用など) をリセットできません。これを許可するには、「Directory Manager がパスワードポリシーをバイパスする」チェックボックスを選択します。
- 「パスワードの暗号化」プルダウンメニューで、パスワードの格納時にサーバーで使用する暗号化方式を指定します。

5. アカウントのロックアウトポリシーを定義するときは、「ロックアウト」タブをクリックし、「アカウントのロックアウト機能を利用」チェックボックスを選択します。

- どれだけの時間に何回のログインが失敗した場合にロックアウトを有効にするか、回数と時間を指定します。
- Directory Manager がユーザーパスワードをリセットするまで永続的にロックアウトするには、「無期限にロックアウト」ラジオボタンを選択します。
- それ以外の場合は「ロックアウト時間」ラジオボタンを選択し、ユーザーアカウントが一時的にロックアウトされる時間を分単位で指定します。

6. カスタムエディタの「了解」をクリックして、ポリシーを保存し、サブエントリを作成します。

コマンド行からのポリシーの定義

このパスワードポリシーでは、一時従業員のパスワードの有効期限を 100 日 (8,640,000 秒) とします。また、パスワードの有効期限が切れる 3 日 (259,200 秒) 前から、バインド時にユーザーへ有効期限切れの警告を開始します。構文検査を有効にして、パスワードセキュリティのための最小限必要な文字数の検査を行います。ロックアウトによって、辞書攻撃を使用してパスワードを解読しようとする侵入者を阻止します。その他のポリシー項目については、デフォルト値を適用します。

`dc=example,dc=com` の下に次のサブエントリを追加して、Example.com のサブツリーにこのパスワードポリシーを定義します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=TempPolicy,dc=example,dc=com
objectClass: top
objectClass: passwordPolicy
objectClass: LDAPsubentry
cn: TempPolicy
passwordStorageScheme: SSHA
passwordChange: on
passwordMustChange: on
passwordCheckSyntax: on
passwordExp: on
passwordExp: on
passwordMinLength: 6
passwordMaxAge: 8640000
passwordMinAge: 0
passwordWarning: 259200
passwordInHistory: 6
passwordLockout: on
passwordMaxFailure: 3
passwordUnlock: on
passwordLockoutDuration: 3600
passwordResetFailureCount: 600
```

パスワードポリシーに関連するすべての属性の説明については、『Sun ONE Directory Server Reference Manual』の第 4 章にある「cn=Password Policy」を参照してください。

パスワードポリシーの割り当て

個別パスワードポリシーの割り当ては、適切なポリシーサブエントリを指定することによって行います。passwordPolicySubentry の値として単一エントリにポリシーを追加するか、CoS とロールを使用してポリシーを管理します。また、アクセス制御を設定して、ユーザーが自分に設定されているパスワードポリシーを変更することを禁止する必要があります。

コンソールから

Directory Server コンソールには、ユーザーまたはグループに割り当てられているパスワードポリシーを管理するためのインタフェースが用意されています。

1. Directory Server コンソールの最上位にある「ディレクトリ」タブで、個別パスワードポリシーを割り当てるか、変更するユーザーまたはグループのエントリを表示します。
2. このエントリをマウスの右ボタンでクリックし、ポップアップメニューから「パスワードポリシーを設定」を選択します。あるいは、エントリをクリックして選択し、「オブジェクト」メニューから「パスワードポリシーを設定」を順に選択します。
3. 「パスワードポリシー」ダイアログが表示され、このエントリに適用されているパスワードポリシーを示します。

- グローバルポリシーが適用されているときは、「割り当て」をクリックして、ディレクトリツリー内の任意の場所にあるパスワードポリシーのサブエントリを選択します。
- すでに個別ポリシーが定義されている場合は、それを置換、削除、編集することができます。「ポリシーを編集」をクリックすると、そのポリシーサブエントリのカスタムエディタが起動されます。

  パスワードポリシーの割り当てまたは置換を行うときは、ディレクトリブラウザダイアログが表示されます。このダイアログには、パスワードポリシーのサブエントリが小さなアイコンで表示されます。

4. ポリシーの変更が完了したら、「パスワードポリシー」ダイアログの「了解」ボタンをクリックします。新しいポリシーは直ちに適用されます。

コマンド行から

ユーザーまたはグループのエントリにパスワードポリシーを割り当てるには、パスワードポリシーの DN を passwordPolicySubentry 属性の値として追加します。たとえば、次のコマンドは cn=TempPolicy,dc=example,dc=com を Barbara Jensen に割り当てます。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: passwordPolicySubentry
passwordPolicySubentry: cn=TempPolicy,dc=example,dc=com
```

ロールと CoS の使い方

ロールを使用してユーザーをグループ化するときには、CoS を使用して適切なポリシーサブエントリに指定できます。ロールと CoS の使用方法については、第 5 章「高度なエントリの管理」を参照してください。

たとえば次のコマンドは、Example.com の一時従業員のフィルタリングされたロールを作成し、このロールを持つ従業員に `cn=TempPolicy,dc=example,dc=com` を割り当てます。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=TempFilter,ou=people,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: TempFilter
nsRoleFilter: (&(objectclass=person)(status=contractor))
description: filtered role for temporary employees

dn: cn=PolTempl,dc=example,dc=com
objectclass: top
objectclass: nsContainer

dn: cn="cn=TempFilter,ou=people,dc=example,dc=com",
 cn=PolTempl,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: LDAPsubentry
objectclass: costemplate
cosPriority: 1
passwordPolicySubentry: cn=TempPolicy,dc=example,dc=com

dn: cn=PolCoS,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
```

```
objectclass: cosClassicDefinition
cosTemplateDN: cn=PolTempl,dc=example,dc=com
cosSpecifier: nsRole
cosAttribute: passwordPolicySubentry operational
```

これにより、契約社員ステータスを持つユーザーは、cn=TempPolicy,dc=example,dc=com パスワードポリシーの対象となります。

個別パスワードポリシーの保護

ユーザーが自分に割り当てられるパスワードポリシーを変更できないようにするには、ルートエントリに次のような ACI を追加する必要があります。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr != "passwordPolicySubentry")(version 3.0; acl
 "Allow self entry modification except for passwordPolicySubentry";
 allow (write) (userdn ="ldap:///self");)
```

ユーザーパスワードのリセット

ディレクトリは、ユーザーエントリの userPassword 属性にパスワード値を格納しています。サーバーのアクセス制御設定によっては、指定したパスワードポリシーに基づいて、ldapmodify などの標準ツールを使用して、userPassword 値を設定することもできます。

永久的なアカウントロックアウトが発生した場合 (ユーザーのオペレーショナル属性 accountUnlockTime が 0 で、パスワードポリシーの passwordUnlock が off の場合) は、Directory Manager としてパスワードをリセットして、ユーザーアカウントのロックを解除できます。たとえば、Example.com のディレクトリユーザーである Barbara Jensen がパスワードを思い出せないために、永続的なロックアウトが適用されたと仮定します。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
replace: userPassword
userPassword: ChAnGeMe
```

パスワードポリシーで passwordMustChange がオンの場合、このユーザーは次回のバインド後にパスワードを変更する必要があります。このユーザーのパスワードが ChAnGeMe に変更されたことを、できるだけ安全な方法でこのユーザーに知らせる必要があります。

ユーザーとロールの無効化と有効化

1 つのユーザーアカウントまたはアカウントのセットを、一時的に無効にすることができます。アカウントが無効になると、ユーザーはディレクトリにバインドできません。そのため、このユーザーの認証操作は失敗します。

ここで説明する手順は、ユーザーとロールの両方の無効化に利用できます。ただし、ロールを無効化すると、ロールのメンバーは無効になりますが、ロールのエントリ自体は無効になりません。ロールの概要、およびロールとアクセス制御が相互に及ぼす影響については、第 5 章「高度なエントリの管理」を参照してください。

コンソールからのユーザーとロールの有効化設定

1. Directory Server コンソールの最上位レベルにある「ディレクトリ」タブでディレクトリツリーを表示し、有効化または無効化するユーザーまたはロールのエントリを探します。
2. このエントリをダブルクリックしてカスタムエディタを表示し、左側の列の「アカウント」タブをクリックします。

   右側のパネルには、このエントリの有効化状態が表示されます。
3. ボタンをクリックして、このエントリに対応するユーザーまたはロールを無効化または有効化します。ユーザーまたはロールのアイコンに表示される赤いボックスとバーは、そのエントリが無効化されることを示しています。
4. 「了解」をクリックしてダイアログボックスを閉じ、このエントリの新しい有効化状態を保存します。

   有効、無効を簡単に設定するには、エントリを選択して「オブジェクト」メニューから「無効」または「有効」を選びます。

「表示」メニューから「表示」をクリックし、「アクティブでない状態」を選択すると、任意のディレクトリオブジェクトの有効化状態を確認できます。無効化されているエントリのすべてのアイコンには、赤いバーが表示されます。ユーザーエントリの有効化状態は、そのエントリが直接無効化されたか、ロールメンバーシップとして無効化されたかに関係なく、正しく表示されます。

コマンド行からのユーザーとロールの有効化設定

コマンド行からユーザーアカウントを無効にするには、ns-inactivate.pl スクリプト (Solaris パッケージでは directoryserver account-inactivate) を使用します。ユーザーまたはロールを有効化、または再有効化するには、ns-activate.pl スクリプト (Solaris パッケージでは directoryserver account-activate) を使用します。このスクリプトのコマンドは、プラットフォームごとに異なります。

Solaris パッケージ

```
# /usr/sbin/directoryserver account-inactivate
# /usr/sbin/directoryserver account-activate
```

Windows プラットフォーム

```
cd ServerRoot
bin¥slapd¥admin¥bin¥perl slapd-serverID¥ns-inactivate.pl
bin¥slapd¥admin¥bin¥perl slapd-serverID¥ns-activate.pl
```

その他のインストール

```
# ServerRoot/slapd-serverID/ns-inactivate.pl
# ServerRoot/slapd-serverID/ns-activate.pl
```

次のコマンドは、perl スクリプトを使って Barbara Jensen のユーザーアカウントを無効化および再有効化する方法を示しています。

```
ns-inactivate.pl -h host -p port -D "cn=Directory Manager" -w password ¥
                 -I "uid=bjensen,ou=People,dc=example,dc=com"

ns-activate.pl -h host -p port -D "cn=Directory Manager" -w password ¥
               -I "uid=bjensen,ou=People,dc=example,dc=com"
```

どちらのコマンドでも、-I オプションは、有効化状態を設定するユーザーまたはロールの DN を指定します。

詳細については、『Sun ONE Directory Server Reference Manual』の第 2 章にある「ns-inactivate.pl」および「ns-activate.pl」を参照してください。

個別のリソース制限の設定

ディレクトリにバインドするクライアントアプリケーションでは、特別なオペレーショナル属性値を使用して、検索操作に関するサーバーの制限を制御できます。検索操作に関しては、次の制限を設定できます。

- 検索制限は、検索処理で参照されるエントリの最大数を指定する
- サイズ制限は、検索処理に応答してクライアントアプリケーションに返されるエントリの最大数を指定する
- 時間制限は、サーバーが検索処理のために使用できる最大時間を指定する
- アイドルタイムアウトは、サーバーが接続を切断するまでに、サーバーへのクライアント接続がアイドル状態でいられる時間を指定する

注	デフォルトでは、Directory Manager は無制限にリソースを利用できます。

特定のユーザーに対して設定したリソース制限は、グローバルサーバー設定で設定したデフォルトのリソース制限より優先されます。個別リソース制限を格納する属性がユーザー自身によって変更されないように、ユーザーエントリを含むサフィックスに次の ACI を追加する必要があります。

```
(targetattr != "nsroledn || aci || nsLookThroughLimit ||
nsSizeLimit || nsTimeLimit || nsIdleTimeout ||
passwordPolicySubentry || passwordExpirationTime ||
passwordExpWarned || passwordRetryCount || retryCountResetTime ||
accountUnlockTime || passwordHistory ||
passwordAllowChangeTime")(version 3.0; acl "Allow self entry
modification except for nsroledn, aci, resource limit attributes,
passwordPolicySubentry and password policy state attributes";
allow (write)userdn ="ldap:///self";)
```

コンソールを使用したリソース制限の設定

1. Directory Server コンソールの最上位レベルにある「ディレクトリ」タブでディレクトリツリーを表示し、リソース制限を設定するエントリを探します。
2. このエントリをダブルクリックしてカスタムエディタを表示し、左側の列の「アカウント」タブをクリックします。右側のパネルには、このエントリの現在の制限セットが表示されます。
3. 上で説明したリソース制限について、4 つのテキストフィールドに値を指定します。-1 を指定すると、そのリソースの制限はなくなります。

4. 設定が完了したら「了解」をクリックし、新しい制限を保存します。

コマンド行からのリソース制限の設定

ldapmodify コマンドを使って次の属性をユーザーエントリに設定することで、ユーザーによるリソース使用を制限できます。

属性	内容
nsLookThroughLimit	検索操作で検査できるエントリの数を指定します。エントリの数として指定します。この属性値を -1 に設定すると、無制限になります。
nsSizeLimit	サーバーが検索操作に対してクライアントアプリケーションに返す最大エントリ数を指定します。この属性値を -1 に設定すると、無制限になります。
nsTimeLimit	サーバーが検索操作を処理するために使用できる最大時間を指定します。この属性値を -1 に設定すると時間が無制限になる
nsIdleTimeout	サーバーがアイドル状態になってから接続が切断されるまでの時間を指定します。値の単位は秒数です。この属性値を -1 に設定すると、無制限になります。

たとえば、次のように ldapmodify を実行することによって、エントリのサイズの制限を設定できます。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: nsSizeLimit
nsSizeLimit: 500
```

この ldapmodify 文は、Barbara Jensen のエントリに nsSizeLimit 属性を追加し、検索結果のサイズの制限を 500 エントリに設定します。

第 8 章

レプリケーションの管理

レプリケーションとは、1 つの Directory Server から別のもう 1 つ、または複数の Directory Server にディレクトリの内容を自動的にコピーするメカニズムです。エントリの追加、変更、削除など、あらゆる書き込み処理はコピー先の Directory Server に自動的にミラー化されます。レプリケーションの概念、レプリケーションの導入例、ディレクトリ導入時にレプリケーションを計画する方法についての詳細は、『Sun ONE Directory Server Deployment Guide』の第 6 章「Designing the Replication Process」を参照してください。

Sun ONE Directory Server 5.2 には、レプリケーションの新機能が数多く用意されています。

- 広域ネットワーク (WAN) 上のマルチマスターレプリケーション (MMR) では、地理的に離れたマスターとの間でレプリケーションアグリーメントを確立し、データをより効率的に配信できます。
- MMR は、完全に接続されたマスターを同時に 4 つサポートできるようになりました。これにより、一層のフェイルオーバ保護を提供できます。
- バイナリコピーにより、大容量レプリカの初期化がより高速になりました。
- 部分レプリケーションにより、レプリケートする属性セットを指定できるので、データをより効率的に配信できます。
- レプリケーション導入の監視には、新しいコマンド行ツールも利用できます。

この章では、レプリケーションのすべての導入例の設定について、マスター、ハブ、コンシューマサーバーで実行するタスクについて説明します。この章は、次の節で構成されています。

- はじめに
- レプリケーションの設定手順のまとめ
- レプリケーションマネージャの選択
- 専用コンシューマの設定

- ハブの設定
- マスターレプリカの設定
- レプリケーションアグリーメントの作成
- 部分レプリケーションの設定
- レプリカの初期化
- 参照整合性プラグインの有効化
- SSL を経由するレプリケーション
- WAN を経由するレプリケーション
- レプリケーショントポロジの変更
- 旧バージョンからのレプリケーション
- 旧バージョン形式の更新履歴ログプラグインの使用
- レプリケーション状態の監視
- よく発生するレプリケーションの競合の解決

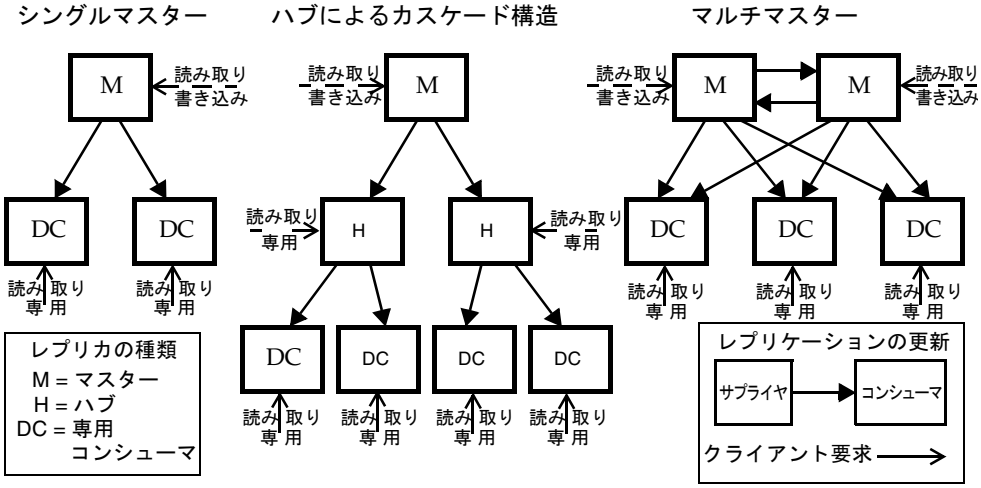
はじめに

レプリケーションの設定は複雑な作業です。設定を始める前に、シングルマスターとマルチマスターのどちらを導入するのか、またはハブを利用したカスケード型レプリケーションを導入するのかなど、組織に導入するレプリケーションの種類を明確にしておく必要があります。レプリケーションの単位はサフィックスまたはサブサフィックスです。1 つのサフィックスに含まれるすべてのエントリは、まとめてレプリケートされます。導入を意図したように行うには、各サフィックスをマスター、ハブ、または含まれるデータの専用コンシューマとして識別する必要があります。

サーバーにレプリケートされるサフィックスをレプリカと呼びます。マスターは、クライアントからの読み取り処理と書き込み処理の両方を受け付けるレプリカです。ハブと専用コンシューマは、レプリケーションメカニズムを使って行われる更新だけを受け付ける、読み取り専用のレプリカです。ハブは、マスターまたは別のハブからの更新を受信し、それを別のハブまたは専用コンシューマに転送します。専用コンシューマは、マスターまたはハブから更新を受信するだけです。

次の図は、レプリケーションの一般的な導入例でのレプリカ間の関係を示しています。

図 8-1　一般的なレプリケーションの例



このマニュアルでは、サプライヤとコンシューマという用語も使います。これは、レプリケーションアグリーメントに関与する 2 種類のサーバーの役割を意味します。サプライヤはレプリケーション更新を送信するサーバーで、コンシューマはそれを受信するサーバーです。上の図は、次の関係を示しています。

- シングルマスターはサプライヤであり、コンシューマではない
- マルチマスターレプリケーションのマスターには、サプライヤとしての役割だけでなく、他のマスターのコンシューマとしての役割もある
- ハブには常にサプライヤとコンシューマの役割がある
- 専用コンシューマは常にコンシューマである

レプリカの種類に関係なく、アグリーメントにはサプライヤとコンシューマの役割として、数多くのレプリケーション設定がレプリカに適用されます。

レプリケーションの設定手順のまとめ

次の手順は、シングル サフィックスのレプリケーションを前提としています。複数のサフィックスをレプリケートする場合は、各サーバーでそれぞれを並行して設定する必要があります。つまり、複数サフィックスのレプリケーションを設定するには、各手順を繰り返す必要があります。

レプリケーションのトポロジを設定する手順は、次のとおりです。

1. シングルマスターを除くすべてのサーバーでレプリケーションマネージャのエントリを定義します。または、すべてのサーバーでデフォルトのレプリケーションマネージャを使用します。
2. 専用コンシューマのレプリカが作成されるすべてのサーバーでは、次の処理を行います。
   a. コンシューマレプリカ用の空のサフィックスを作成します。
   b. レプリケーションウィザードを使って、サフィックスに含まれるコンシューマレプリカを有効にします。
   c. 必要に応じて、詳細なレプリカ設定を行います。
3. ハブを利用する場合は、ハブのレプリカが作成されるすべてのサーバーで次の処理を行います。
   a. ハブレプリカ用の空のサフィックスを作成します。
   b. レプリケーションウィザードを使って、サフィックスに含まれるハブレプリカを有効にします。
   c. 必要に応じて、詳細なレプリカ設定を行います。
4. マスターレプリカが作成されるすべてのサーバーでは、次の処理を行います。
   a. マスターレプリカとなるマスターで、サフィックスを 1 つ選択するか、作成します。
   b. レプリケーションウィザードを使って、サフィックスに含まれるマスターレプリカを有効にします。
   c. 必要に応じて、詳細なレプリカ設定を行います。
5. すべてのサプライヤレプリカで、次の順序でレプリケーションアグリーメントを設定します。
   a. マルチマスターセットのマスター間
   b. マスターと専用コンシューマの間

c. マスターとハブレプリカの間

必要に応じて、部分レプリケーションを設定し、この時点でコンシューマレプリカとハブレプリカを初期化することもできます。マルチマスターレプリケーションでは、データのオリジナルコピーを含むマスターレプリカから順にすべてのマスターを初期化します。

6. マスターからデータを直接受け取るすべてのハブレプリカで、レプリケーションアグリーメントを設定します。これは、ハブレプリカとそのコンシューマとの間のアグリーメントです。必要に応じて、この時点でコンシューマレプリカを初期化します。カスケード型のレプリケーションでは、ハブのすべての階層でこの手順を繰り返します。

注 以上の手順で重要なことは、レプリケーションアグリーメントを作成する前に、レプリカをすべて作成し、設定しておくことです。こうすることで、レプリケーションアグリーメントの作成後、ただちにコンシューマレプリカを初期化できます。コンシューマの初期化は、常にレプリケーションの設定の最後の段階で実行します。

レプリケーションマネージャの選択

レプリケーションの設定で重要なのは、レプリケーションマネージャというエントリを選択することです。レプリケーションマネージャは、サプライヤがレプリケーションの更新を送信するときに、コンシューマサーバーとのバインドに使われます。専用コンシューマ、ハブ、マルチマスターレプリケーションに組み込まれているマスターなど、更新を受け取るサフィックスを持つすべてのサーバーでは、少なくとも 1 つのレプリケーションマネージャエントリが必要です。

Directory Server には、すべてのサーバーで利用できるデフォルトのレプリケーションマネージャエントリが用意されています。このエントリの DN は `cn=Replication Manager,cn=replication,cn=config` です。

注 単純なレプリケーションでは、すべての導入例でデフォルトのレプリケーションマネージャを利用することをお勧めします。レプリケーションウィザードは、このエントリを使ってコンシューマレプリカを自動的に設定するので、レプリカを簡単に導入できます。

デフォルトレプリケーションマネージャのパスワードが定義されていない場合、レプリケーションウィザードはパスワードの入力を要求します。デフォルトレプリケーションマネージャのパスワードをあとから変更する手順は、次のとおりです。

1. Directory Server コンソールの最上位の「設定」タブで「データ」ノードを選び、右側のパネルで「レプリケーション」タブを選びます。
2. 「レプリケーションマネージャ」という見出しの下にある 2 つのテキストフィールドに新しいパスワードを入力します。
3. パスワードの確認への入力が完了したら、「保存」をクリックします。パスワードが確認のために入力したパスワードと一致しない場合、「保存」ボタンは有効になりません。

レプリケーションマネージャとして機能するエントリを新たに作成することもできます。たとえば、レプリケートされるすべてのサフィックスで、複数のレプリケーションマネージャエントリに異なるパスワードを持たせることができます。また、たとえば SSL 経由の証明書など、異なる認証モデルをレプリケーションに適用するには、独自のレプリケーションマネージャを作成する必要があります。

レプリケーションマネージャエントリには、レプリケーションアグリーメントを定義するときに、選択した認証方法に適した属性が含まれている必要があります。たとえば、デフォルトのレプリケーションマネージャのオブジェクトクラスは `person` です。このクラスは、`userPassword` 属性を使った簡単な認証に対応しています。証明書を使ってレプリケーションマネージャをバインドする方法の詳細は、305 ページの「SSL を経由するレプリケーション」を参照してください。

このレプリケーションマネージャのエントリは、コンシューマサーバーのレプリケートされたサフィックスの中に保存しておくことはできません。レプリケーションマネージャの定義場所としてふさわしいのは、`cn=replication,cn=config` です。

警告 レプリケーションマネージャエントリの DN とパスワードを使って、バインドを実行したり、サーバー上で処理を行うことはできません。レプリケーションマネージャはレプリケーションメカニズムだけが使用するものであり、その他の使用ではレプリカの再初期化が必要です。

各コンシューマのレプリケーションマネージャを選択したら、次の処理を行います。

1. 選択または作成したレプリケーションマネージャの DN を書き留めるか、覚えておきます。この DN とそのパスワードは、あとからこのコンシューマのサプライヤとの間でレプリケーションアグリーメントを作成するときに必要です。
2. パスワードに有効期限ポリシーを定義したときは、レプリケーションマネージャを除外しておく必要があります。除外しない場合、パスワードの有効期限が切れるとレプリケーションが行われなくなります。レプリケーションマネージャエントリでパスワードの有効期限を無効にするには、パスワードの有効期限が切れないパスワードポリシーを作成し、それをレプリケーションマネージャエントリに割り当てます。詳細は、263 ページの「個別パスワードポリシーの管理」を参照してください。

専用コンシューマの設定

専用コンシューマは、レプリケートされたサフィックスの読み取り専用コピーです。これは、特別なレプリケーションマネージャとしてバインドされたマスターサーバーから更新を受け取り、変更を行います。コンシューマサーバーを設定するには、レプリカを保持する空のサフィックスを準備し、レプリケーションウィザードを使ってそのサフィックスのレプリケーションを有効にします。必要に応じて、異なるレプリケーションマネージャの選択、リフェラルの設定、パージ遅延の設定など、詳細な設定を行うこともできます。

次に、1 つの専用コンシューマレプリカをそのレプリカのサーバーに設定する手順について説明します。特定のサフィックスの専用コンシューマレプリカを含むすべてのサーバーで、同じ手順を繰り返してください。

コンシューマレプリカのサフィックスの作成

サフィックスをまだ作成していない場合は、レプリケーションの対象となるマスターレプリカと同じ DN を使ってコンシューマに空のサフィックスを作成します。手順については、88 ページの「サフィックスの作成」を参照してください。

すでにサフィックスが存在し、それが空でない場合は、マスターからレプリカが初期化されたときにそのサフィックスの内容は失われます。

コンシューマレプリカの有効化

レプリケーションウィザードを使うことで、専用コンシューマのレプリカを簡単に有効にできます。

1. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、コンシューマレプリカとして設定するサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。

   右側のパネルにレプリカのステータス情報が表示されます。

2. 「レプリケーションを有効に」ボタンをクリックすると、レプリケーションウィザードが起動されます。

3. デフォルトでは、「コンシューマレプリカ」ラジオボタンが選択されています。「次へ」をクリックして処理を続けます。

4. デフォルトレプリケーションマネージャのパスワードが定義されていない場合、パスワードの入力と確認のための再入力が求められます。各フィールドに同じパスワードを入力し、「次へ」をクリックします。

   デフォルトレプリケーションマネージャにすでにパスワードが定義されている場合、この手順は省略されます。

5. レプリケーションウィザードはレプリケーションの設定更新を開始します。ウィザードには、ステータスメッセージが表示されます。処理が完了したら、「閉じる」をクリックします。

レプリケーションのステータスは、レプリカが更新の受信準備が整っていることを示し、それに対応して左側のペインのアイコンも変更されます。

コンシューマの詳細設定

デフォルトでは、ウィザードはデフォルトのレプリケーションマネージャを使うようにレプリカを設定します。レプリケーションマネージャエントリを独自に作成した場合、それを使うには詳細な設定が必要です。また、このダイアログを使って変更とパージ遅延のリフェラルも設定できます。

1. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、設定するサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。
2. 右側のパネルで「詳細」ボタンをクリックし、「詳細レプリカ設定」ダイアログを開きます。
3. 「バインド DN」タブで、「追加」ボタンと「削除」ボタンを使って有効なレプリケーションマネージャの DN リストを作成します。このリストは、サプライヤとこのレプリカとの間のアグリーメントに含まれるため、サプライヤはリスト内のいずれの DN も利用できるようになります。新しい DN を追加するときは、DN 名を入力するか、ディレクトリを参照して選択します。

   SSL 経由の証明書を使うようにレプリケーションを設定するときは、レプリケーションマネージャの 1 つのエントリとして証明書の DN を指定します。
4. 処理が完了したら、「了解」をクリックします。「オプション」タブを選んで、さらに詳細な設定を行うこともできます。

5. 「詳細レプリカ設定」ダイアログの「オプション」タブには、このコンシューマに送信される変更要求の追加リフェラルを指定する LDAP URL のリストが表示されます。LDAP URL のリストを作成するには、「追加」ボタンと「削除」ボタンを使います。

   レプリケーションのメカニズムでは、レプリケーショントポロジに含まれるすべての既知のマスターのリフェラルを返すようにコンシューマを自動的に設定します。これらのデフォルトリフェラルは、クライアントが標準的な接続で簡単な認証を使うことを前提としています。安全な接続のために SSL を使ってマスターにバインドするオプションをクライアントに提供するには、`ldaps://`*servername*`:`*port* という形式でリフェラルを追加します。*port* にはセキュリティ保護された接続に使うポート番号を指定します。

   リフェラルとして 1 つまたは複数の LDAP URL を追加したときは、リストの下に表示されるチェックボックスを選択して、コンシューマがマスターレプリカのリフェラルではなく、これらの LDAP URL のリフェラルを送信するようにします。たとえば、クライアントがデフォルトのポートではなく、常にマスターサーバーのセキュリティ保護されたポートにアクセスするように設定するには、これらのセキュリティ保護されたポートの LDAP URL リストを作成し、このチェックボックスを選択します。すべての更新を処理する特定のマスターまたは Directory Server プロキシを指定する場合も、排他的なリフェラルを用意する必要があります。

6. 「オプション」タブでは、パージ遅延も変更できます。

   コンシューマサーバーは、レプリカの内容に加えられる変更に関する内部情報を格納する必要があり、パージ遅延はこの情報を保持する期間を決定します。この値は、サプライヤサーバーの更新履歴ログの MaxAge パラメータと関連づけられています。これらの 2 つのパラメータのうち、短いほうの設定が、2 つのサーバー間のレプリケーションが無効になった、またはダウンした場合でも正常な状態に復元できる最長期間を決定します。ほとんどの場合には、デフォルトの 7 日間が適当です。

7. 「了解」をクリックして、このレプリカの詳細設定を保存します。

ハブの設定

ハブレプリカは、コンシューマとしてだけではなく、マスターとしても機能し、レプリケートされたデータをより多くのコンシューマに配信します。このため、レプリケーションの更新をそれぞれのサプライヤから受信するだけでなく、レプリケーションの更新をそれぞれのコンシューマに送信する必要があります。ハブレプリカは変更を受け付けませんが、マスターにリフェラルを返します。

ハブサーバーを設定するには、レプリカを保持する空のサフィックスを準備し、レプリケーションウィザードを使ってそのサフィックスのレプリケーションを有効にします。必要に応じて、異なるレプリケーションマネージャの選択、リフェラルの設定、パージ遅延の設定、ログパラメータの設定と変更など、詳細な設定も行うことができます。

次に、1 つのハブサーバーを設定する手順について説明します。特定のサフィックスのハブレプリカを含むすべてのサーバーで、同じ手順を繰り返してください。

ハブレプリカのサフィックスの作成

サフィックスをまだ作成していない場合は、レプリケーションの対象となるマスターレプリカと同じ DN を使ってハブサーバーに空のサフィックスを作成します。手順については、88 ページの「サフィックスの作成」を参照してください。

すでにサフィックスが存在し、それが空でない場合は、マスターからレプリカが初期化されたときにそのサフィックスの内容は失われます。

ハブレプリカの有効化

レプリケーションウィザードを使うことで、ハブレプリカを簡単に有効にできます。

1. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、ハブレプリカとして設定するサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。

   右側のパネルにレプリカのステータス情報が表示されます。

2. 「レプリケーションを有効に」ボタンをクリックすると、レプリケーションウィザードが起動されます。
3. 「ハブレプリカ」ラジオボタンを選択し、「次へ」をクリックします。

4. 更新履歴ログファイルを選択していない場合は、ログファイルの選択が求められます。テキストフィールドには、デフォルトの更新履歴ログファイルが表示されます。デフォルトの更新履歴ログファイルを使わないときは、ファイル名を入力するか、「参照」をクリックしてファイルを選択します。

   更新履歴ログファイルがすでに指定されている場合、この手順は省略されます。

5. 「次へ」をクリックします。デフォルトレプリケーションマネージャのパスワードが定義されていない場合、パスワードの入力と確認のための再入力が求められます。各フィールドに同じパスワードを入力し、「次へ」をクリックします。

   デフォルトレプリケーションマネージャにすでにパスワードが定義されている場合、この手順は省略されます。

6. レプリケーションウィザードはレプリケーションの設定更新を開始します。ウィザードには、ステータスメッセージが表示されます。処理が完了したら、「閉じる」をクリックします。

レプリケーションのステータスは、レプリカが更新の受信準備が整っていることを示し、それに対応して左側のペインのアイコンも変更されます。

ハブの詳細設定

ハブはサプライヤとして更新履歴ログファイルを必要とします。ウィザードを使った場合、デフォルトの更新履歴ログの設定を使ってハブレプリカが設定されます。この設定を変更する手順は、次のとおりです。

1. Directory Server コンソールの最上位の「設定」タブで「データ」ノードを選び、右側のパネルで「レプリケーション」タブを選びます。
2. このタブの内容の更新が必要な場合には、「更新履歴ログを有効に」チェックボックスを選択して「リセット」ボタンをクリックします。これにより、レプリケーションウィザードで選択した更新履歴ログファイルが表示されます。
3. 更新履歴ログファイルの名前を変更したり、ログパラメータを変更したりできます。
   a. 「更新履歴ログの最大レコード数」- コンシューマに更新を送信するために格納しておく変更の最大数を決定します。デフォルトでは、無制限です。レプリカが数多くの大容量の変更を受信するときは、レコード数を少なめに設定して、ディスク容量を節約できます。
   b. 「更新履歴ログの最長保存期間」- コンシューマに送信する必要のある更新をどれだけの期間ハブが保持しているかを決定します。デフォルトでは、無制限です。更新履歴ログのサイズを制限するときは、最長保存期間パラメータを利用することをお勧めします。

レプリケーションウィザードは、デフォルトのレプリケーションマネージャを使用します。レプリケーションマネージャエントリを独自に作成した場合、それを使うには詳細な設定が必要です。また、このダイアログを使って変更とパージ遅延のリフェラルも設定できます。

1. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、設定するサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。
2. 右側のパネルで「詳細」ボタンをクリックし、「詳細レプリカ設定」ダイアログを開きます。
3. 「バインド DN」タブで、「追加」ボタンと「削除」ボタンを使って有効なレプリケーションマネージャの DN リストを作成します。このリストは、サプライヤとこのレプリカとの間のアグリーメントに含まれるため、サプライヤはリスト内のいずれの DN も利用できるようになります。新しい DN を追加するときは、DN 名を入力するか、ディレクトリを参照して選択します。

   SSL 経由の証明書を使うようにレプリケーションを設定するときは、レプリケーションマネージャの 1 つのエントリとして証明書の DN を指定します。
4. 処理が完了したら、「了解」をクリックします。「オプション」タブを選んで、さらに詳細な設定を行うこともできます。
5. 「詳細レプリカ設定」ダイアログの「オプション」タブには、このハブに送信される変更要求の追加リフェラルを指定する LDAP URL のリストが表示されます。LDAP URL のリストを作成するには、「追加」ボタンと「削除」ボタンを使います。

   レプリケーションのメカニズムでは、レプリケーショントポロジに含まれるすべての既知のマスターのリフェラルを返すようにハブを自動的に設定します。これらのデフォルトリフェラルは、クライアントが標準的な接続で簡単な認証を使うことを前提としています。安全な接続のために SSL を使ってマスターにバインドするオプションをクライアントに提供するには、`ldaps://`*servername*`:`*port* という形式でリフェラルを追加します。*port* にはセキュリティ保護された接続に使うポート番号を指定します。

   リフェラルとして 1 つまたは複数の LDAP URL を追加したときは、リストの下に表示されるチェックボックスを選択して、サーバーがマスターレプリカのリフェラルではなく、これらの LDAP URL のリフェラルを送信するようにします。たとえば、クライアントがデフォルトのポートではなく、常にマスターサーバーのセキュリティ保護されたポートにアクセスするように設定するには、これらのセキュリティ保護されたポートの LDAP URL リストを作成し、このチェックボックスを選択します。すべての更新を処理する特定のマスターまたは Directory Server プロキシを指定する場合も、排他的なリフェラルを用意する必要があります。

6. 「オプション」タブでは、パージ遅延も変更できます。

   ハブサーバーは、レプリカの内容に加えられる変更に関する内部情報を格納する必要があり、パージ遅延はこの情報を保持する期間を決定します。この値は、更新を提供するサーバーの更新履歴ログ (このサーバー自体の更新履歴ログではない) の MaxAge パラメータと関連づけられています。これらの 2 つのパラメータのうち、短いほうの設定が、2 つのサーバー間のレプリケーションが無効になった、またはダウンした場合でも正常な状態に復元できる最長期間を決定します。ほとんどの場合には、デフォルトの 7 日間が適当です。

7. 「了解」をクリックして、このレプリカの詳細設定を保存します。

マスターレプリカの設定

マスターレプリカにはデータのマスターコピーが含まれ、更新を他のすべてのレプリカに配信する前に、すべての変更を集中的に管理します。マスターはすべての変更を記録し、関連する各コンシューマの状態を確認して、必要に応じて更新を送信します。マルチマスターレプリケーションでは、マスターレプリカが他のマスターから更新を受け取ることもあります。

マスターサーバーを設定するときは、マスターレプリカを含むサフィックスを決定し、レプリケーションウィザードを使ってマスターレプリカを有効にします。また、必要に応じてレプリケーションの詳細設定を行います。

次に、1 つのマスターサーバーを設定する手順について説明します。特定のサフィックスのマスターレプリカを含むすべてのサーバーで、同じ手順を繰り返してください。

マスターレプリカのサフィックスの定義

レプリケートするエントリを保存するマスターサーバー上でサフィックスを選択、または作成します。手順については、88 ページの「サフィックスの作成」を参照してください。

サフィックスには、レプリケーションアグリーメントを作成する前にすべての初期データを含めておく必要があります。こうすることで、このデータからただちにコンシューマレプリカを初期化できます。マルチマスター設定と初期化を正しく確実に行うために、すべての初期データを 1 つのマスターだけに含め、他のマスターのサフィックスは空にしておきます。

マスターレプリカの有効化

レプリケーションウィザードを使うことで、マスターレプリカを簡単に有効にできます。

1. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、マスターレプリカとして設定するサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。

   右側のパネルにレプリカのステータス情報が表示されます。

2. 「レプリケーションを有効に」ボタンをクリックすると、レプリケーションウィザードが起動されます。

3. 「マスターレプリカ」ラジオボタンを選択し、「次へ」をクリックします。

4. レプリカ ID を入力します。指定できる値は、1 ～ 65534 までの間で他の ID と重複しない整数です。

   あるサフィックスのすべてのマスターレプリカでは、それぞれのレプリカ ID が一意である必要があります。同一サーバー上であってもサフィックスが異なる場合は、マスターレプリカのレプリカ ID は同じでも問題ありません。

5. 「次へ」をクリックします。更新履歴ログファイルを選択していない場合は、ログファイルの選択が求められます。テキストフィールドには、デフォルトの更新履歴ログファイルが表示されます。デフォルトの更新履歴ログファイルを使わないときは、ファイル名を入力するか、「参照」をクリックしてファイルを選択します。

   更新履歴ログファイルがすでに指定されている場合、この手順は省略されます。

6. 「次へ」をクリックします。デフォルトのレプリケーションマネージャのパスワードが定義されていない場合、パスワードの入力と確認のための再入力が求められます。シングルマスターレプリケーションではレプリケーションマネージャは使われませんが、設定を続けるにはパスワードを入力する必要があります。各フィールドに同じパスワードを入力し、「次へ」をクリックします。

   デフォルトレプリケーションマネージャにすでにパスワードが定義されている場合、この手順は省略されます。

7. レプリケーションウィザードはレプリケーションの設定更新を開始します。ウィザードには、ステータスメッセージが表示されます。処理が完了したら、「閉じる」をクリックします。

レプリケーションのステータスにはこのマスターのレプリカ ID が表示され、左側のペインにはこのサフィックスのレプリケーションが有効であることを示すアイコンが表示されます。

マルチマスターの詳細設定

デフォルトでは、ウィザードはデフォルトの更新履歴ログを使うようにマスターレプリカを設定します。更新履歴ログの設定を変更する手順は、次のとおりです。

1. Directory Server コンソールの最上位の「設定」タブで「データ」ノードを選び、右側のパネルで「レプリケーション」タブを選びます。
2. このタブの内容の更新が必要な場合には、「更新履歴ログを有効に」チェックボックスを選択して「リセット」ボタンをクリックします。これにより、レプリケーションウィザードで選択した更新履歴ログファイルが表示されます。
3. 更新履歴ログファイルの名前を変更したり、ログパラメータを変更したりできます。
   a. 「更新履歴ログの最大レコード数」- コンシューマに更新を送信するために格納しておく変更の最大数を決定します。デフォルトでは、無制限です。レプリカが数多くの大容量の変更を受信するときは、レコード数を少なめに設定して、ディスク容量を節約できます。
   b. 「更新履歴ログの最長保存期間」- コンシューマに送信する必要のある更新をどれだけの期間ハブが保持しているかを決定します。デフォルトでは、無制限です。更新履歴ログのサイズを制限するときは、最長保存期間パラメータを利用することをお勧めします。

レプリケーションウィザードは、デフォルトのレプリケーションマネージャを使用します。レプリケーションマネージャエントリを独自に作成した場合、それを使うには詳細な設定が必要です。また、このダイアログを使って変更とパージ遅延のリフェラルも設定できます。シングルマスターの設定では、この手順を省略します。

1. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、設定するサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。
2. 右側のパネルで「詳細」ボタンをクリックし、「詳細レプリカ設定」ダイアログを開きます。
3. 「バインド DN」タブで、「追加」ボタンと「削除」ボタンを使って有効なレプリケーションマネージャの DN リストを作成します。このリストは、サプライヤとこのレプリカとの間のアグリーメントに含まれるため、サプライヤはリスト内のいずれの DN も利用できるようになります。新しい DN を追加するときは、DN 名を入力するか、ディレクトリを参照して選択します。

   SSL 経由の証明書を使うようにレプリケーションを設定するときは、レプリケーションマネージャの 1 つのエントリとして証明書の DN を指定します。
4. 処理が完了したら、「了解」をクリックします。「オプション」タブを選んで、さらに詳細な設定を行うこともできます。

5. 「詳細レプリカ設定」ダイアログの「オプション」タブには、このマスターに送信される変更要求の追加リフェラルを指定する LDAP URL のリストが表示されます。296 ページの「マルチマスター初期化後のマスター間の一致」でも説明しますが、初期化が完了すると、マスターはただちにリフェラルを返します。LDAP URL のリストを作成するには、「追加」ボタンと「削除」ボタンを使います。

   レプリケーションのメカニズムでは、レプリケーショントポロジに含まれるすべての既知のマスターのリフェラルを返すようにハブを自動的に設定します。これらのデフォルトリフェラルは、クライアントが標準的な接続で簡単な認証を使うことを前提としています。安全な接続のために SSL を使ってマスターにバインドするオプションをクライアントに提供するには、`ldaps://`*servername*`:`*port* という形式でリフェラルを追加します。*port* にはセキュリティ保護された接続に使うポート番号を指定します。

   リフェラルとして 1 つまたは複数の LDAP URL を追加したときは、リストの下に表示されるチェックボックスを選択して、サーバーがマスターレプリカのリフェラルではなく、これらの LDAP URL のリフェラルを送信するようにします。たとえば、クライアントがデフォルトのポートではなく、常にマスターサーバーのセキュリティ保護されたポートにアクセスするように設定するには、これらのセキュリティ保護されたポートの LDAP URL リストを作成し、このチェックボックスを選択します。

6. 「オプション」タブでは、パージ遅延も変更できます。

   マスターサーバーは、レプリカの内容に加えられる変更に関する内部情報を格納する必要があり、パージ遅延はこの情報を保持する期間を決定します。この値は、更新を提供するマスターサーバーの更新履歴ログ (このサーバー自体の更新履歴ログではない) の MaxAge パラメータと関連づけられています。これらの 2 つのパラメータのうち、短いほうの設定が、2 つのサーバー間のレプリケーションが無効になった、またはダウンした場合でも正常な状態に復元できる最長期間を決定します。ほとんどの場合には、デフォルトの 7 日間が適当です。

7. 「了解」をクリックして、このレプリカの詳細設定を保存します。

レプリケーションアグリーメントの作成

レプリケーションアグリーメントはサプライヤ側に設定されるパラメータセットで、更新をどのようにコンシューマに送信するかを制御します。コンシューマに更新を送信するサプライヤレプリカには、レプリケーションアグリーメントを作成する必要があります。レプリケーションアグリーメントは、レプリケーションメカニズムを使って更新するすべてのコンシューマに 1 つずつ作成する必要があります。

レプリケーションアグリーメントを作成する順序は、次のとおりです。

1. マルチマスターセットのマスターの間 (レプリケートするサフィックスのオリジナルコピーを含むマスターを最初に作成)
2. マスターと、レプリケーションにハブを使わない専用コンシューマの間
3. マスターとハブレプリカの間
4. ハブレプリカと、ハブを使うコンシューマの間

たとえば 275 ページの図 8-1 は、2 つのマスターと 3 つの専用コンシューマのレプリケーショントポロジを持つマルチマスターレプリケーションを示しています。この場合、次の順序で 8 つのレプリケーションアグリーメントを作成します。

- 1 つのマスターともう一方のマスターの間
- もう一方のマスターと最初のマスターの間
- 1 つのマスターと 3 つの専用コンシューマの間
- もう一方のマスターと 3 つの専用コンシューマの間

レプリケーションアグリーメントを作成する手順は、次のとおりです。

1. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、サプライヤサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。

   右側のパネルにレプリカのステータス情報が表示されます。

2. 定義されているレプリケーションアグリーメントの隣の「新規」ボタンをクリックします。
3. 「レプリケーションアグリーメント」ダイアログが表示されるので、コンシューマレプリカを含む既存のサーバーをメニューから選択するか、「その他」ボタンをクリックして新たに定義します。

   「その他」ボタンをクリックしたときは、コンシューマサーバーの完全修飾名と LDAP ポート番号を入力します。そのポートで SSL を使っているときは、セキュリティ保護されたポートのチェックボックスを選択し、安全な接続によるレプリケーションの更新を有効にします。

4. コンシューマサーバー上のレプリケーションマネージャエントリの DN とパスワードを入力します。デフォルトでは、デフォルトレプリケーションマネージャの DN が指定されます。

   セキュリティ保護されたポートを持つコンシューマを選択したときは、「オプション」ボタンをクリックして、「DN」フィールドの内容が示す意味を指定できます。パスワードを使った接続では、サプライヤは簡単な認証と、暗号化された SSL 接続を経由する通信を利用します。証明書を使った接続では、「DN」フィールドの内容は証明書を含むエントリの DN を意味し、パスワードは必要ありません。

5. 必要に応じて、このアグリーメントの説明文を入力します。コンシューマサーバーの名前とポート番号、説明文は、このマスターレプリカのレプリケーションアグリーメントリストに表示されます。

6. 設定が完了したら「了解」をクリックします。設定した接続パラメータをテストするかどうかを確認するダイアログが表示されます。

7. 指定したレプリケーションマネージャとパスワードを使って指定のサーバーとポート番号への接続をテストするときは、「はい」をクリックします。接続に失敗した場合でも、そのアグリーメントを使う設定を維持できます。これは、たとえばパラメータに問題はないが、サーバーがオフラインである場合などに有効です。

   設定が完了すると、このマスターレプリカのレプリケーションアグリーメントリストにアグリーメントが表示されるようになります。

レプリケーションアグリーメントを編集して、コンシューマサーバー上のレプリケーションマネージャの DN とパスワードをあとから変更できます。

1. リストからレプリケーションアグリーメントを選択し、「編集」ボタンをクリックします。

2. 「レプリケーションアグリーメント」ダイアログが表示されるので、「接続」タブを選びます。

3. コンシューマサーバー上のレプリケーションマネージャの DN またはパスワードを編集します。

4. 必要に応じて、このアグリーメントの説明文を編集します。

5. 「了解」をクリックして新しい設定を保存し、このコンシューマに更新を送信すると、ただちにそのアグリーメントが使用されます。

   その他のタブの設定パラメータについては、293 ページの「部分レプリケーションの有効化」と 306 ページの「WAN を経由するレプリケーション」を参照してください。

6. 必要なレプリケーションアグリーメントを作成したら、このサフィックスの部分レプリケーションを設定し、レプリカをただちに初期化することもできます。詳細は、294 ページの「レプリカの初期化」を参照してください。

部分レプリケーションの設定

デフォルトでは、レプリケートされるサフィックスに含まれるすべてのエントリがコンシューマレプリカにコピーされます。Sun ONE Directory Server 5.2 に新たに追加された部分レプリケーション機能を使うことで、レプリケーション時にコピーの対象に含める、またはコピーの対象から除外する属性のサブセットを指定できます。部分レプリケーションはレプリケーションアグリーメントに設定されるので、マスターのコンシューマレプリカごとに属性セットを定義できます。これにより、配信するデータを制御し、レプリケーションの帯域幅とコンシューマリソースをより効率的に利用できます。

たとえば、photo、jpegPhoto、audio のように一般に値が大きい属性のレプリケーションを除外することで、レプリケーションの帯域幅を節約できます。この場合、コンシューマではこれらの属性を利用できなくなります。また、認証に必要な uid 属性と userpassword 属性だけをコンシューマサーバーにレプリケートすることもできます。

部分レプリケーションに関する注意点

属性の部分的なセットを有効化または変更するには、コンシューマレプリカを初期化し直す必要があります。このため、配備前に部分レプリケーションの必要性を検討し、レプリケーションを初期化する前に属性セットを設定しておく必要があります。

ACI、ロール、CoS などの複雑な機能が特定の属性に依存する小規模な属性セットをレプリケートするときは、慎重な対応が必要です。ACI、ロール、CoS の各メカニズムの指定子やフィルタで参照されるその他の属性がレプリケートされない場合、データのセキュリティが損なわれたり、検索時に異なる属性セットが返されることがあります。レプリケーションの対象に含める属性のリストを管理するよりも、除外する属性のリストを管理する方法が安全であり、人的なミスも少なくなります。

レプリケートするすべてのエントリがスキーマに準拠しない属性セットをレプリケートするときは、コンシューマサーバーのスキーマチェックを無効にする必要があります。スキーマに準拠しないエントリをレプリケートしても、レプリケーションメカニズムはコンシューマ上でのスキーマチェックを行わないため、エラーは発生しません。しかし、スキーマに準拠しないエントリがコンシューマに含まれるようになるので、クライアントにとって一貫した状態にする必要があるためスキーマチェックを無効にする必要があります。

部分レプリケーションは、ハブと専用コンシューマに関連するマスターレプリカのレプリケーションアグリーメントに設定します。マルチマスターレプリケーション環境の 2 つのマスターレプリカ間での部分レプリケーションは設定できません。また、複数のマスターが同じレプリカとの間でレプリケーションアグリーメントを持つ場合、同じ属性セットをレプリケートするようにすべてのアグリーメントを設定する必要があります。

Sun ONE Directory Server 5.2 が提供する部分レプリケーション機能は、旧バージョンの Directory Server との間に下位互換性を持ちません。部分レプリケーションアグリーメントを設定するときは、マスターレプリカとコンシューマレプリカの両方が Directory Server 5.2 インスタンス上に存在する必要があります。

属性セットの定義

属性セットは、部分レプリケーションがレプリカで有効になった場合にレプリケートされる属性のリストで、リストに含まれない属性はレプリケートされません。マスターサーバーには任意の数の属性セットを設定することができ、その属性セットのどれかをレプリケーションアグリーメントに関連づけます。

1. Directory Server コンソールの最上位の「設定」タブで「データ」ノードを選び、右側のパネルで「レプリケーション」タブを選びます。
2. 「レプリケーション」タブの下部にある「レプリケートされた属性のセットを管理」ボタンをクリックします。このボタンを表示するには、スクロールが必要なこともあります。
3. 「追加」をクリックして新しい属性セットを定義するか、リストから既存の属性セットを選び、「編集」をクリックして内容を変更します。「属性セット」ダイアログが表示されるので、レプリケートする列のチェックボックスを使って属性をセットに加えたり、セットから除外したりします。名前の隣にチェックマークが表示された属性がレプリケートされます。

   デフォルトでは、すべての属性が選択されています。レプリケートを避ける具体的な理由のある属性だけの選択を解除することをお勧めします。最初から設定し直すときは、「すべてをチェック」ボタンをクリックすると、すべての属性が選択されます。多数の属性の選択を解除した場合、Directory Server は選択が解除された属性以外のすべての属性をレプリケートします。あとから新しい属性をスキーマに定義し、レプリケートされたエントリでそれを使用する場合、属性セットを編集して選択を解除するまで、これらの新しい属性はレプリケートされます。

   「チェックしない」ボタンをクリックすると、すべての属性の選択が解除されるので、セットに含める属性 (レプリケートする属性) だけを選択できます。「チェックしない」ボタンをクリックしてから属性セットを定義した場合、選択した属性だけがレプリケートされます。あとから新しい属性をスキーマに定義し、レプリケートされたエントリでそれを使用する場合、属性セットを編集してその属性を選択するまで、これらの新しい属性はレプリケートされません。

注 objectClass、nsUniqueId、nsDS50ruv の各属性と RDN ネーミング属性は、属性セットの定義に関係なくレプリケートされます。これは、objectClass 属性とネーミング属性は LDAP の修正に必要であり、nsUniqueId 属性と nsDS50ruv 属性はレプリケーションを正常に行うために必要なためです。

ACI 属性をレプリケーションの対象から除外すると、コンシューマレプリカでのアクセス制御に影響が生じます。userPassword 属性を除外した場合、コンシューマレプリカにアクセスするすべてのユーザーを認証できなくなります。

4. 必要に応じて、属性セットの説明文を入力または変更します。この文章は、このセットを使用するレプリケーションアグリーメントの編集時に、定義されているセットとともにリスト表示されます。説明文を入力しない場合、含まれるか、除外される属性に基づいてサーバーが説明を自動的に生成します。
5. 処理が終了したら、「保存」をクリックします。

部分レプリケーションの有効化

部分レプリケーションは、既存のレプリケーションアグリーメントだけで有効にできます。

1. 289 ページの「レプリケーションアグリーメントの作成」の説明に従ってレプリケーションアグリーメントを作成するか、すでに定義されているアグリーメントを選択して変更します。
2. 312 ページの「レプリケーションアグリーメントの無効化」の説明に従って、レプリケーションアグリーメントを無効にします。部分レプリケーションの設定を変更するときは、アグリーメントを無効にする必要があります。
3. 無効にしたアグリーメントを選び、「編集」をクリックします。「レプリケーションアグリーメント」ダイアログが表示されるので、「レプリケートされた属性」タブを選びます。
4. 「属性のセットのみレプリケート」チェックボックスを選びます。
5. ドロップダウンリストから既存の属性セットを選ぶか、「新規」をクリックし、292 ページの「属性セットの定義」の説明に従って新しい属性セットを定義します。「レプリケートされた属性のセットを管理」をクリックして既存のセットの定義を表示し、それを編集することもできます。

   部分レプリケーションでは、レプリケーションアグリーメントに関連づけることができる属性セットは 1 つだけです。このセットには、レプリケートする属性だけで構成されたリストを含める必要があります。

6. 属性セットを選んだら、「了解」をクリックします。部分レプリケーションの設定を変更したため、コンシューマレプリカを初期化し直す必要があることを示すメッセージが表示されます。「了解」をクリックすると、メッセージは消えます。
7. 「有効」をクリックして、レプリケーションアグリーメントを有効な状態に戻します。
8. レプリケートする属性によっては、コンシューマサーバーで実行されるスキーマチェックを無効にする必要があります。
9. 他のマスターがこのレプリカとの間にレプリケーションアグリーメントを持つ場合、そのすべてにおいて同じ手順を繰り返し、同じ属性セットによる部分レプリケーションを有効にする必要があります。
10. 次に、コンシューマレプリカを初期化するか、すでにレプリケートされていた場合は再初期化します。次の「レプリカの初期化」を参照してください。

レプリカの初期化

レプリケーションアグリーメントを作成したら、レプリケーションを実際に開始する前にコンシューマレプリカを初期化する必要があります。初期化時は、サプライヤレプリカからコンシューマレプリカにデータが物理的にコピーされます。

特定のエラーが発生した場合、または設定を変更した場合は、レプリカを初期化し直す必要があります。初期化し直したときは、コンシューマ側のレプリケートされたサフィックスは削除され、マスター側のサフィックスの内容に置き換えられます。これにより、レプリカの同期が確保され、レプリケーションの更新が再開されます。また、ここで説明するどの方法で初期化を行なっても、コンシューマレプリカのインデックスは自動的にふたたび作成されるため、クライアントからの読み取り要求にもただちに正しく対応できます。

初期化のタイミング

レプリカの初期化は、関連する両方のレプリカの設定が完了したあとで、レプリケーションを開始する前に行う必要があります。サフィックスに含まれるデータ全体がコンシューマにコピーされると、サプライヤはコンシューマに対する更新処理を開始します。

通常の運用では、コンシューマを初期化し直す必要はありません。しかし、何らかの理由で 1 つのマスターレプリカをバックアップから復元した場合、そのレプリカが更新するすべてのレプリカを初期化し直す必要があります。マルチマスターレプリケーションでは、他のマスターによって更新されたコンシューマであれば、初期化し直す必要がない場合もあります。

コンソールを使ってレプリカをオンラインで初期化するか、コマンド行を使って手動で初期化できます。コンソールを使ったオンライン初期化は、少数のコンシューマを初期化する場合に便利です。レプリケーションアグリーメントからレプリカを直接オンラインで初期化できます。ただし、この方法ではレプリカは 1 つずつ初期化されるため、多数のレプリカを処理する場合には適していません。コマンド行を使った手動による初期化は、1 つの LDIF ファイルから多数のコンシューマを同時に初期化できるので、多数のコンシューマを初期化する場合に効果的です。

最後に、経験が豊富な管理者であれば、Directory Server 5.2 に新たに搭載されたバイナリコピー機能を利用することで、マスターレプリカまたはコンシューマレプリカのクローンを作成できます。この機能にはある種の制限が適用されるため、処理時間の短縮を見込めるのは、たとえば百万件単位のエントリを含むレプリカなど、大容量のデータベースファイルを持つレプリカだけです。

マルチマスターレプリケーションにおけるレプリカの初期化

マルチマスターレプリケーションの場合、次の順序でレプリカを初期化する必要があります。

1. 1 つのマスターが、レプリケーション対象の完全なデータセットを保持していることを確認します。その他の各マスターのレプリカを初期化するには、このマスターを使います。
2. それぞれのマスターから、またはいずれかのマスターの LDIF ファイルからコンシューマレプリカを初期化します。

カスケード型レプリケーションでのレプリカの初期化

カスケード型レプリケーションの場合、常に次の順序でレプリカを初期化する必要があります。

1. マルチマスターレプリケーションとの組み合わせでは、1 つのマスターが、レプリケーション対象の完全なデータセットを保持していることを確認します。その他の各マスターのレプリカを初期化するには、このマスターを使います。
2. それぞれのマスターレプリカから、最初の階層のハブレプリカに属するレプリカを初期化します。
3. ハブの構成が複数の階層に分かれている場合、各階層を上から順に初期化していきます。
4. 最後の階層のハブレプリカから専用コンシューマのレプリカを初期化します。

マルチマスター初期化後のマスター間の一致

マルチマスターレプリケーションでは、あるマスターの初期化中に他のマスターが変更を処理することもあります。このため、初期化が完了した時点で、新しいマスターは初期化データに含まれていなかった新しい更新を受け取る必要があります。初期化には時間がかかるため、その間に発生する未適用の更新の数も問題となります。

これらの未適用更新が適用されるように、新たに初期化されたマスターは、初期化後、クライアント側からの操作に対して自動的に読み取り専用モードに設定されます。この設定は、コンソールを使ったオンライン初期化、コマンド行から LDIF ファイルを使った初期化、バイナリコピーを使ったバックアップの実行など、初期化の種類に関係なく行われます。これは、Sun ONE Directory Server 5.2 の新機能です。

したがって、マルチマスター設定の初期化後のマスターは、レプリケーションの更新を処理し、クライアントからの読み取り操作を受け付けますが、すべての書き込み操作に対してはリフェラルを返します。リフェラルは、287 ページの「マルチマスターの詳細設定」で説明されている手順に従って定義できます。マスターのモードは、次の場合に読み書きモードに変わります。

- `ds5BeginReplicaAcceptUpdates` 設定属性を `start` に設定し、更新の受け付けを明示的に可能にします。更新の受け付けを有効にする前に、新しいマスターレプリカの内容が他のマスターと一致していることを確認する必要があります。確認には、Directory Server コンソールのレプリケーション設定パネル、またはコマンド行 (後述する手順を参照) を使います。

  更新の受け付けを有効にする前に、新しいマスターが他のマスターと完全に同期していることを確認できるので、新たに初期化されたマスターの更新を有効にする方法としては、手動による処理をお勧めします。

- 事前に `ds5referralDelayAfterInit` 属性が設定されている場合、所定の時間が経過したあとにマスターレプリカは自動的に読み書きモードに切り替わります。この属性は、サーバー情報のマスターレプリカごとに設定できます。

  この属性を設定するときは、初期化後に新しいマスターレプリカが内容を他のマスターと一致するために必要な時間を考慮する必要があります。この処理に要する時間は、予定される初期化の規模と所要時間、およびその他のマスターで並行して行われる更新の頻度によって異なります。初期化後、マスターが更新をレプリケートしている最中に新たな更新を受け付けた場合、予期せぬエラーが発生することがあります。レプリケーションエラーが発生したときは、『Sun ONE Directory Server Reference Manual』の付録 A 「エラーコード」を参照してください。

注 この新しい対応方法によってマスターレプリカがリフェラルを送信する場合、書き込み処理を待機しているクライアントのホップ回数が、制限回数に達してしまうことも考えられます。利用可能なマスターにアクセスできるように、クライアントのホップ制限の設定を変更する必要があるかもしれません。すべてのマスターレプリカを初期化または再初期化するときは、どのレプリカもクライアントからの更新を受け付けられないため、すべての書き込み処理が失敗します。

サーバーの応答を最大化するには、いかなる場合も初期化したマスターを注意深く監視し、リフェラルの属性を適切に設定する必要があります。

コンソールによる更新の受け付け開始

マルチマスターレプリカの初期化後に、更新の受け付けを明示的に開始する手順は、次のとおりです。

1. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、レプリケートされたサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。

   レプリカが初期化され、現時点では更新処理に対してリフェラルを返していることを示すメッセージが右側のパネルに表示されます。自動的なリフェラル遅延が有効であることを示すメッセージが表示される場合でも、この手順を使って自動遅延の代わりに受け付けの開始を明示的に指定できます。

2. `insync` ツールを使って、レプリカの状態が他のすべてのマスターと一致していることを確認します。すべてのサーバーで変更の遅れがゼロである場合、またはそのレプリカに適用する更新がなかった場合 (遅れが `-1` となる場合) は、すべてのレプリカが同期しています。詳細については、『Sun ONE Directory Server Reference Manual』の第 1 章にある「insync」を参照してください。

3. メッセージの右にあるボタンをクリックして、ただちに更新の受け付けを開始します。

コマンド行による更新の受け付け開始

マルチマスターレプリカの初期化プロセスを自動化するスクリプトで、次のコマンドを実行することができます。このコマンドは、マスターレプリカ間の一致を確認し、更新の受け付けを明示的に有効にします。

1. `insync` ツールを使って、レプリカの状態が他のすべてのマスターと一致していることを確認します。すべてのサーバーで変更の遅れがゼロである場合、またはそのレプリカに適用する更新がなかった場合 (遅れが `-1` となる場合) は、すべてのレプリカが同期しています。詳細については、『Sun ONE Directory Server Reference Manual』の第 1 章にある「insync」を参照してください。

2. 次のコマンドを使って ds5BeginReplicaAcceptUpdates 設定属性を変更します。

```
% ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn:cn=replica, cn=suffixName, cn=mapping tree, cn=config
changetype: modify
add: ds5BeginReplicaAcceptUpdates
ds5BeginReplicaAcceptUpdates: start
^D
```

レプリカの初期化が完了すると ds5BeginReplicaAcceptUpdates は自動的に削除されるので、次回の初期化後は更新処理の受け付けは拒否されます。

自動リフェラル遅延の設定

ds5referralDelayAfterInit 設定属性は、初期化後に何秒間レプリカがリフェラルを返すかを決定します。この時間が経過すると、レプリカは自動的にクライアントからの更新処理を受け付けるようになります。この属性は各レプリカに固有の設定です。296 ページの「マルチマスター初期化後のマスター間の一致」で説明されている条件に基づいて値を設定してください。

この属性の値の変更は、初期化され、まだ更新を受け付けていないレプリカにダイナミックに適用されます。この値を変更することで、遅延時間を延長または縮小できます。遅延時間が経過し、レプリカが更新処理の受け付けを開始したあとに属性の設定を変更しても、レプリカは影響を受けません。

この属性のデフォルト値は -1 で、レプリカは無期限に更新処理を拒否します。この場合、遅延時間を設定することで、初期化からこの時間が経過した時点で自動的に更新処理を許可できます。すでに経過している時間を遅延時間に設定すると、レプリカは更新をただちに受け付けます。

1. ds5referralDelayAfterInit 属性に次のコマンドを設定します。

```
% ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn:cn=replica, cn=suffixName, cn=mapping tree, cn=config
changetype: modify
replace: ds5referralDelayAfterInit
ds5referralDelayAfterInit: seconds
^D
```

コンソールによるレプリカの初期化

コンソールを使ったオンラインでのレプリカの初期化は、コンシューマの初期化と再初期化でもっとも簡単な方法です。ただし、多数のエントリ (100 ～ 200 万) を初期化する場合、処理にかなり時間がかかるため、コマンド行を使用した手動によるコンシューマの初期化の方がより効果的なこともあります。詳細は、300 ページの「コマンド行によるレプリカの初期化」を参照してください。

注 コンソールを使ってコンシューマレプリカを初期化している最中は、レプリカに対するすべての処理 (検索を含む) は初期化のプロセスが完了するまでマスターサーバーを参照します。

Directory Server コンソールを使った場合、部分レプリケーションが設定されたレプリカの初期化は透過的に行われます。初期化時に、選択されている属性だけがコンシューマに送られます。

オンラインでのレプリカ初期化の実行

コンソールを使ってレプリカを初期化または再初期化する手順は、次のとおりです。

1. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、マスターレプリカサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。

   右側のパネルにレプリカのステータス情報が表示されます。

2. 定義されているレプリケーションアグリーメントのリストから、初期化するコンシューマに対応するアグリーメントを選び、「アクション」>「リモートレプリカを初期化」の順にクリックします。

   コンシューマ上のレプリカに格納されている情報がすべて削除されるという確認メッセージが表示されます。

3. 確認ボックスで「はい」をクリックします。

   オンラインコンシューマの初期化がただちに開始されます。レプリケーションアグリーメントのアイコンが赤いギアとなり、初期化プロセスの状態を示します。

4. 「再表示」>「ただちに再表示」の順にクリックするか、「再表示」>「継続して再表示」の順にクリックして、コンシューマ初期化の状態を監視します。

   強調表示されているアグリーメントに関するメッセージが、リストの下のテキストボックスに表示されます。

レプリケーションおよび初期化の状態の監視については、327 ページの「レプリケーション状態の監視」を参照してください。

コマンド行によるレプリカの初期化

コマンド行を使って手動でレプリカを初期化する方法は、多数のエントリのレプリケーションが必要な導入で、コンシューマを初期化するにはもっとも速い方法です。パフォーマンス上の問題からオンラインプロセスが適切でないと判断する場合は、手動プロセスを使用するように提案します。ただし、手動によるコンシューマの初期化は、オンラインでのコンシューマ初期化と比べてプロセスが複雑です。

レプリカを手動で初期化または再初期化するには、まず、オリジナルのサフィックスデータのコピーを LDIF ファイルにエクスポートする必要があります。部分レプリカを初期化するときは、ファイルをフィルタリングして、レプリケートされる属性だけを確保する必要があります。次に、そのファイルをすべてのコンシューマサーバーに転送し、それをインポートします。マルチマスターレプリケーションの導入では、オリジナルマスターからエクスポートした LDIF ファイルを使って他のマスターとコンシューマの両方を初期化できます。カスケード型のレプリケーションでは、同じファイルを使ってハブレプリカとそのコンシューマを初期化できます。

どの場合にも、設定が完了しているマスターレプリカからエクスポートした LDIF ファイルから開始する必要があります。これ以外の任意の LDIF ファイルにはレプリケーションデータが含まれないため、これを使ってすべてのレプリカを初期化することはできません。最初に LDIF ファイルをマスターレプリカにインポートし、次の手順でそれをエクスポートする必要があります。

LDIF ファイルへのレプリカのエクスポート

レプリカの内容を LDIF ファイルに格納するには、`db2ldif -r` コマンドまたは `db2ldif.pl -r` コマンドを使います。詳細は、143 ページの「コマンド行からの LDIF へのエクスポート」を参照してください。これらのコマンドを使ってレプリカをエクスポートするときは、`-r` オプションを指定する必要があります。

次の例は、`dc=example,dc=com` レプリカ全体を `example_master.ldif` というファイルにエクスポートします。

Solaris パッケージ

```
# /usr/sbin/directoryserver stop
# /usr/sbin/directoryserver db2ldif -r -s "dc=example,dc=com" ¥
  -a /var/ds5/slapd-serverID/ldif/example_master.ldif
# /usr/sbin/directoryserver start
```

その他のインストール

```
# ServerRoot/slapd-serverID/stop-slapd
# ServerRoot/slapd-serverID/db2ldif -r -s "dc=example,dc=com" ¥
  -a ServerRoot/slapd-serverID/ldif/example_master.ldif
# ServerRoot/slapd-serverID/start-slapd
```

次に、必要に応じて LDIF ファイルをフィルタリングし、それをコンシューマホストに転送して、コンシューマレプリカを初期化します。

部分レプリケーションのための LDIF ファイルのフィルタリング

部分レプリケーションを設定したときは、エクスポートした LDIF ファイルをコンシューマサーバーにコピーする前に、不要な属性をフィルタリングする必要があります。この処理を行うために、Directory Server には `fildif` というツールが用意されています。このツールは、指定した LDIF ファイルをフィルタリングし、レプリケーションアグリーメントに定義されている属性セットが許可する属性だけを残します。

このツールはサーバーの設定を読み取り、属性セットの定義を決定します。設定ファイルの読み取りが必要になるため、`fildif` ツールはルートとして実行する必要があります。たとえば、次のコマンドは、前の例で `dc=example,dc=com` サフィックスからエクスポートされたファイルをフィルタリングします。

```
# CAMUS=/var/Sun/mps/slapd-camus
# /var/Sun/mps/shared/bin/fildif ¥
-i $CAMUS/ldif/example_master.ldif ¥
-o $CAMUS/ldif/filtered.ldif -c $CAMUS/config/dse.ldif ¥
-b "cn=rousseau.example.com:389, cn=replica, ¥
cn=dc=example¥, c=com, cn=mapping tree, cn=config"
```

`-i` オプションと `-o` オプションは、それぞれ入力ファイルと出力ファイルです。`-c` オプションは、レプリケーションアグリーメントと属性セットの定義を含む設定ファイルです。サーバーは、`cn=config` エントリの内容を `dse.ldif` ファイルに格納します。レプリケーションアグリーメントと属性セットもこれに含まれます。

`-b` オプションは、部分レプリケーションが定義されているレプリケーションアグリーメントの DN です。このエントリを見つけるには、Directory Server コンソールで Directory Manager として `cn=config` サフィックスを参照します。サフィックスの `cn=replica` エントリの下のエントリを選び、メニューから「編集」>「DN のコピー」の順に選んで、この DN をクリップボードにコピーします。コマンドを入力するときは、この情報を使います。

`fildif` ツールの完全なコマンド行構文は、『Sun ONE Directory Server Reference Manual』の第 1 章にある「LDIF Command-Line Utilities」で説明しています。

`fildif` ツールを使って作成した `filtered.ldif` ファイルを使って、このレプリケーションアグリーメントの対象となるコンシューマを初期化できます。ファイルをコンシューマサーバーに転送し、次に説明する手順に従ってインポートします。

コンシューマレプリカへの LDIF ファイルのインポート

マスターレプリカの内容が含まれている LDIF ファイルをコンシューマレプリカにインポートするには、Directory Server コンソールのインポート機能を使用するか、`ldif2db` コマンドまたは `ldif2db.pl` スクリプト (Solaris パッケージでは `directoryserver ldif2db` または `directoryserver ldif2db-task`) を使用しま

す。その他すべてのインポート処理と同様に、インポートを行うには、Directory Manager のバインド DN とパスワードをスクリプトに指定する必要があります。これらのインポート方法については、136 ページの「コマンド行からの LDIF のインポート」を参照してください。

次の例は、LDIF ファイルをインポートして、`dc=example,dc=com` コンシューマレプリカを初期化する方法を示しています。

Solaris パッケージ

```
# /usr/sbin/directoryserver stop
# /usr/sbin/directoryserver ldif2db -s "dc=example,dc=com" ¥
  -i example_master.ldif
# /usr/sbin/directoryserver start
```

その他のインストール

```
# ServerRoot/slapd-serverID/stop-slapd
# ServerRoot/slapd-serverID/ldif2db -s "dc=example,dc=com" ¥
  -i example_master.ldif
# ServerRoot/slapd-serverID/start-slapd
```

`ldif2db.pl` スクリプトを使う場合、事前にサーバーを停止する必要はありません。詳細については、『Sun ONE Directory Server Reference Manual』の第 2 章にある「ldif2db.pl」を参照してください。

バイナリコピーによるレプリカの初期化

Directory Server 5.2 に新たに搭載されたバイナリコピー機能は、1 つのサーバーのバイナリバックアップファイルを使って、別のサーバー上の同じディレクトリの内容を復元することで、サーバー全体のクローンを作成します。この高度な機能では、Directory Server 上のデータベースファイルとの間で情報をやり取りします。この機能は、経験が豊富な管理者以外は使用しないでください。

バイナリコピーの制限

バイナリコピー機能は、あるマシンから別のマシンにデータベースファイルを移動するため、次の制限が厳密に適用されます。

- どちらのマシンも同じハードウェアと同じオペレーティングシステム (サービスパック、パッチも含む) を使用する
- どちらのマシンにも同じバージョンの Directory Server をインストールする。バイナリ形式 (32 ビットまたは 64 ビット)、サービスパックとパッチのレベルも同一である必要がある
- どちらのサーバーも同じディレクトリツリーを持ち、同じサフィックスに分割されている。すべてのサフィックスのデータベースファイルを一度にコピーすることが必要で、サフィックスを個別にコピーすることはできない

- どちらのサーバーでも各サフィックスに VLV (仮想リスト表示) インデックスも含めて同じインデックスを設定する。サフィックスのデータベースには、同じ名前をつける必要がある
- コピーする Directory Server には `o=NetscapeRoot` サフィックスを含めることはできない。つまり、Directory Server を Sun ONE 管理サーバーの設定ディレクトリにすることはできない
- 各サーバーには、レプリカとして同じサフィックスが設定され、両方のサーバーで各レプリカは同じ役割 (マスター、ハブ、またはコンシューマ) を持つ必要がある。部分レプリケーションを設定する場合は、すべてのマスターサーバーで同様に設定する必要がある
- どちらのサーバーでも属性の暗号化を使用できない
- 属性値の一意性プラグインが有効な場合は、両方のサーバーで設定を共通させる。また、次に説明する手順で、新しいコピーを設定し直す必要がある

上の条件を満たす環境では、別のマスターサーバーのバイナリコピーからマスターを初期化または再初期化したり、別のコンシューマサーバーのバイナリコピーからコンシューマを初期化または再初期化したりできます。次の 2 つの手順は、バイナリコピーの実行方法を示しています。一つはサーバーを停止せずに行う方法で、もう一つはディスクスペースの消費を最小限に抑える方法です。

サーバーの停止を必要としないバイナリコピー

次の手順は、通常のバックアップ機能を使ってサーバーのデータベースファイルのコピーを作成するので、バイナリコピーを実行するときは、この方法をお勧めします。通常のバックアップを実行することで、サーバーを停止しなくても、すべてのデータベースファイルを一定の状態に維持できます。

ただし、この手順には注意を要する制限があります。バックアップと復元の処理によって、同じマシンにデータベースファイルのコピーが作成されるため、各マシンでこれらのファイルが占有するディスクスペースの容量が 2 倍になります。また、これらのファイルに対する実際のコピー処理は、ディレクトリに G バイト単位のデータが含まれる場合、時間がかかります。ディスクスペースが限られていたり、データベースファイルのサイズが極端に大きい場合の対応については、304 ページの「ディスクスペースの消費量を最小限に抑えるバイナリコピー」を参照してください。

1. 新しいレプリカのターゲットマシンに Directory Server をインストールし、必要に応じてサーバーの新しいインスタンスを作成します。次に、302 ページの「バイナリコピーの制限」に従ってインスタンスを設定します。
2. このレプリカに関連するレプリケーショントポロジにすべてのレプリケーションアグリーメントを作成します。これには、サプライヤからこのレプリカへのアグリーメントも含まれ、専用コンシューマ以外では、このレプリカから各コンシューマへのアグリーメントも含まれます。

3. 初期化するレプリカと同じ種類 (マスター、ハブ、コンシューマのどれか) の、完全に設定され、初期化されたサフィックスを選択し、145 ページの「コンソールを使用したサーバーのバックアップ」の手順に従って通常のバックアップ処理を行います。
4. バックアップディレクトリからターゲットマシンのディレクトリにファイルをコピーまたは転送します。この操作には、`ftp` コマンドなどを使います。
5. 146 ページの「バックアップからのデータの復元」の手順に従って、ファイルをターゲットサーバーにロードします。
6. マルチマスターレプリケーションの新しいマスターを初期化したときは、296 ページの「マルチマスター初期化後のマスター間の一致」の手順に従って、新しいレプリカがクライアントからの更新処理を受け付けるように設定します。

ディスクスペースの消費量を最小限に抑えるバイナリコピー

次の手順では、データベースファイルのバックアップコピーを作成しないため、ディスクスペースの消費が少なく、処理に要する時間も少なくなります。ただし、データベースファイルを一貫した状態に保つため、クローン作成の対象となるサーバーを停止する必要があります。

警告 マルチマスターレプリケーションにすでに組み込まれているマスターの再初期化に、この手順を使うことはできません。この手順を利用できるのは、コンシューマサーバーの再初期化、または新しいマスターサーバーの初期化だけです。既存のマスターレプリカを再初期化するときは、オンラインによる初期化を行い、LDIF ファイルをインポートするか、303 ページの「サーバーの停止を必要としないバイナリコピー」の手順を実行します。

1. 新しいレプリカのターゲットマシンに Directory Server をインストールし、必要に応じてサーバーの新しいインスタンスを作成します。次に、302 ページの「バイナリコピーの制限」に従ってインスタンスを設定します。
2. このレプリカに関連するレプリケーショントポロジにすべてのレプリケーションアグリーメントを作成します。これには、サプライヤからこのレプリカへのアグリーメントも含まれ、専用コンシューマ以外では、このレプリカから各コンシューマへのアグリーメントも含まれます。
3. 20 ページの「Directory Server の起動と停止」の説明に従って、初期化または再初期化するサーバーを停止します。
4. 初期化するレプリカと同じ種類 (マスター、ハブ、コンシューマのどれか) の、完全に設定され、初期化されたレプリカを選択し、このサーバーも停止します。マルチマスター設定に組み込まれているマスターレプリカのクローンを作成するときは、それを停止する前に、その他のマスターから最新のすべての変更が完全に反映されていることを確認する必要があります。

5. トランザクションログを含むすべてのデータベースファイルをソースレプリカマシンからターゲットマシンにコピーまたは転送します。この処理には、`ftp` コマンドなどを使います。ファイルの位置を変更していない限り、データベースファイルとトランザクションログは *ServerRoot*`/slapd-`*serverID*`/db` ディレクトリに保存されています。

   マスターレプリカまたはハブレプリカを初期化するときは、更新履歴ログに記録されているすべてのファイルもコピーする必要があります。更新履歴ログは、デフォルトでは *ServerRoot*`/slapd-`*serverID*`/changelog` ディレクトリにあります。

6. ソースサーバーとターゲットサーバーの両方を再起動します。

参照整合性プラグインの有効化

参照整合性プラグインを使っている場合、すべてのマスターサーバーでそれを有効にする必要があります。ハブサーバーまたはコンシューマサーバー上のプラグインを有効にする必要はありません。詳細は、83 ページの「レプリケーションにおける参照整合性の使用」を参照してください。

SSL を経由するレプリケーション

すべてのレプリケーション操作が SSL 接続を経由するように、レプリケーションに関連する Directory Server を設定できます。この設定を行う手順は、次のとおりです。

1. サプライヤサーバーとコンシューマサーバーの両方を、SSL を使用するように設定します。

   詳細は、第 11 章「セキュリティの実装」を参照してください。

注 次のサプライヤサーバー証明書では、SSL を経由するレプリケーションは失敗します。

- 自己署名の証明書
- SSL ハンドシェイク時にクライアントとして機能できない SSL サーバー専用証明書

---

2. コンシューマサーバー上のサフィックスにレプリケーションが設定されていない場合、279 ページの「コンシューマレプリカの有効化」の説明に従って、そのサフィックスを有効にします。
3. 280 ページの「コンシューマの詳細設定」の手順に従って、コンシューマの証明書エントリの DN を別のレプリケーションマネージャとして定義します。

4. サプライヤサーバー上のサフィックスにレプリケーションが設定されていない場合 282 ページの「ハブレプリカの有効化」または 286 ページの「マスターレプリカの有効化」の説明に従って、そのサフィックスを有効にします。
5. サプライヤサーバーで新しいレプリケーションアグリーメントを作成し、セキュリティ保護された SSL ポート上のコンシューマに更新を送信します。詳細な方法については、289 ページの「レプリケーションアグリーメントの作成」を参照してください。セキュリティ保護されたポートをコンシューマサーバーに設定し、パスワードまたは証明書のどちらを使うかについて、SSL オプションを選択します。選択した SSL オプションの DN (レプリケーションマネージャまたは証明書) を入力します。

レプリケーションアグリーメントの設定が完了すると、サプライヤはすべてのレプリケーション更新メッセージを SSL 経由でコンシューマに送信します。証明書を使用するオプションを選んだ場合は、証明書が利用されます。SSL のアグリーメント設定を使ってコンソールからカスタマーの初期化を行う場合も、セキュリティ保護された接続が使われます。

WAN を経由するレプリケーション

Sun ONE Directory Server 5.2 では、広域ネットワーク (WAN) に接続されたマシン間のマルチマスターレプリケーション (MMR) を含め、あらゆる種類のレプリケーションを行えます。レプリケーションメカニズムを内部的に見直したことで、応答時間が遅く、帯域幅の狭いネットワークでも、妥当な遅延でサプライヤサーバーがコンシューマを初期化、更新できるようになりました。

注 レプリケーションの実際の遅延と更新パフォーマンスは、修正頻度、エントリのサイズ、サーバーハードウェア、平均応答時間、平均帯域幅など、多くの要因に左右されます。利用環境でのレプリケーションについて疑問がある場合は、Sun Professional Service の担当者に連絡してください。

デフォルトでは、レプリケーションメカニズムの内部パラメータは WAN に合わせて最適化されています。ただし、前述の要因などが原因でレプリケーションが遅くなるときは、ウィンドウサイズとグループサイズのパラメータを調節してみてください。また、ネットワークのピーク時を避けてレプリケーションをスケジュールすることで、ネットワークの全体的な利用率を高めることができます。最後に、Solaris、Linux の各プラットフォームで稼動する Directory Server は、帯域幅の使用を最適化するためにレプリケーションデータの圧縮に対応しています。

ネットワークパラメータの設定

ネットワーク経由でエントリをより効率的に送信するために、レプリケーションメカニズムがエントリをグループ化する方法は、次の 2 つのパラメータによって決定されます。これらのパラメータは、サプライヤとコンシューマがレプリケーション更新メッセージと、その確認応答を交換する方法に影響します。

- ウィンドウサイズ (デフォルト値は 10) : コンシューマからの即時の確認応答なしに送信できる更新メッセージの最大数を表します。WAN 環境では、メッセージごとに確認応答を待つよりも、多数のメッセージを一度に送信したほうが効率的です。
- グループサイズ (デフォルト値は 1) : 1 つの更新メッセージに入れることのできるデータ修正の最大数を表します。データのサイズとネットワークのプロパティによっては、より大きなメッセージ、つまりグループサイズがより大きいメッセージを送信するほうが効率的になります。

ほとんどの状況では、デフォルト値の利用が最適です。しかし、ディレクトリエントリの数が極端に多い、または少ない場合、またはレプリケートが必要な修正の頻度が極端に高い場合は、これらのパラメータを変更して、WAN 経由のレプリケーションパフォーマンスに与える影響をテストできます。

この 2 つのネットワークパラメータは、すべてのレプリケーションアグリーメントで設定できます。これにより、各コンシューマのネットワーク状態に応じてレプリケーションパフォーマンスを調節できます。

ウィンドウとグループのサイズパラメータを変更するときに、レプリケーションを中断する必要はありません。

1. Directory Server コンソールで「設定」タブを選び、「データ」ノードを展開し、レプリケートするサフィックスのノードを展開します。
2. サフィックスの下の「レプリケーション」ノードを選び、右側のペインで設定するレプリケーションアグリーメントを選んで、「編集」をクリックします。
3. 「レプリケーションアグリーメント」ダイアログの「ネットワーク」タブを選び、ウィンドウサイズの新しい値 (1 ～ 1000) とグループサイズの新しい値 (1 ～ 100) を入力します。グループサイズは、ウィンドウサイズ以下に設定する必要があります。
4. 「了解」をクリックして新しい値を保存し、「レプリケーションアグリーメント」ダイアログを閉じます。

   新しいパラメータ値はただちに有効となり、対応するコンシューマに次にレプリケーション更新を送信するときに適用されます。

レプリケーションアクティビティのスケジュール

レプリカ間の即時同期が重要でない場合は、ネットワーク利用率が低い時間帯に更新をスケジュールして、WAN 経由のデータレプリケーションを実行できます。より多くのネットワーク資源を利用できれば、更新はより高速で処理され、ネットワークの利用率がすでに高い場合は、レプリケーション メッセージによってネットワークにそれ以上の負荷がかかることはありません。

レプリケーションアグリーメントを設定することで、コンシューマごとに日次または週次の更新をスケジュールできます。

1. Directory Server コンソールで最上位の「設定」タブを選び、「データ」ノードを展開し、レプリケートするサフィックスのノードを展開します。
2. サフィックスの下の「レプリケーション」ノードを選び、右側のペインで設定するレプリケーションアグリーメントを選んで、「編集」をクリックします。
3. 「レプリケーションアグリーメント」ダイアログの「スケジュール」タブを選び、週次スケジュールの隣のラジオボタンを選びます。
4. スケジュールを定義します。
   a. 週次更新では、レプリケーションを行う曜日 (複数を指定できる) の隣のチェックボックスを選択します。各曜日のレプリケーションをさらに細かく指定するときは、時間帯を 24 時間表記で指定します。
   b. 日時更新では、「すべて」をクリックしてすべての曜日を選択し、レプリケーションを行う時間帯を 24 時間表記で指定します。

      深夜 0 時をまたがる時間帯は設定できません。
5. 「了解」をクリックして新しい値を保存し、「レプリケーションアグリーメント」ダイアログを閉じます。

   新しいスケジュールはただちに有効になり、対応するコンシューマに対する次回のレプリケーション更新は、スケジュールされた次回の更新まで行われなくなります。

データの圧縮

レプリケーションで使われる帯域幅を節約するために、コンシューマの更新時に送信されるデータを圧縮するようにレプリケーションを設定できます。レプリケーションメカニズムは、Zlib 圧縮ライブラリを使用します。これは、対応している Solaris および Linux プラットフォームだけで利用できます。圧縮を利用するには、Solaris または Linux プラットフォームでサプライヤとコンシューマの両方が稼動している必要があります。

レプリケーションの圧縮の設定は、マスターサーバー側のレプリケーションアグリーメントエントリに含まれる ds5ReplicaTransportCompressionLevel 属性だけを使って行われます。この属性には、次のいずれかの値を指定できます。

0 - 圧縮を行いません。これは、ds5ReplicaTransportCompressionLevel 属性が定義されていない場合のデフォルトの設定です。

1 - Zlib ライブラリのデフォルトの圧縮レベルを使用します。

2 - Zlib ライブラリの最適サイズの圧縮レベルを使用します。

3 - Zlib ライブラリの最速の圧縮レベルを使用します。

各圧縮レベルを実験的に使用し、レプリケーションの用途に合わせて WAN 環境で最適な結果を得られるオプションを選択します。圧縮と解凍のプロセスによってレプリケーション速度が低下するため、遅延があまり問題とならない LAN (ローカルエリアネットワーク) ではこのパラメータを使用しないでください。

たとえば、east.example.com にあるコンシューマに最速圧縮オプションを使ってレプリケーション更新を送信するには、次の ldapmodify コマンドを実行します。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=east.example.com:389,cn=replica,cn="suffixDN",
 cn=mapping tree,cn=config
changetype: modify
add: ds5ReplicaTransportCompressionLevel
ds5ReplicaTransportCompressionLevel: 3
^D
```

レプリケーショントポロジの変更

ここでは、レプリケーションアグリーメントの編集と削除、レプリカの昇格と降格、無効化、コンシューマの強制更新、更新履歴ログの管理など、既存のレプリケーショントポロジを管理する手順について説明します。

レプリケーションアグリーメントの管理

マスターサフィックスのレプリケーションパネルでは、レプリケーションアグリーメントの設定を変更して、アグリーメントの認証情報の変更、特定のコンシューマに対するレプリケーションの中断、トポロジからのコンシューマの削除を実行できます。

レプリケーションマネージャの変更

レプリケーションアグリーメントを編集して、コンシューマサーバーへのバインドに使われるレプリケーションマネージャの識別情報を変更できます。レプリケーションが中断されないように、レプリケーションアグリーメントを変更する前に、新しいレプリケーションマネージャエントリまたはコンシューマの証明書エントリを定義する必要があります。ただし、バインドの失敗によってレプリケーションが中断された場合、レプリケーション回復設定の制限内でエラーを修正したときは、レプリケーションメカニズムによって必要なすべての更新が自動的に送信されます (280 ページの「コンシューマの詳細設定」を参照)。

コンシューマの認証に使うレプリケーションマネージャを変更する手順は、次のとおりです。

1. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、レプリケートされたサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。
2. 右側のパネルで変更するレプリケーションアグリーメントを選び、「編集」をクリックします。
3. 「レプリケーションアグリーメント」ダイアログで、「接続」タブを選びます。

   ステータス行には、コンシューマサーバーのホスト名とポート番号が表示されます。
4. DN とパスワードのフィールドを変更します。別のレプリケーションマネージャエントリの DN とパスワードを入力するか、コンシューマサーバー上の証明書エントリの DN を入力します。

5. このレプリケーションアグリーメントが、セキュリティ保護されたポートで SSL を使用するときは、「オプション」ボタンをクリックして、セキュリティ保護された認証の種類を選択することもできます。パスワードを使った接続では、サプライヤは簡単な認証と指定の DN、および暗号化された SSL 接続による通信を利用します。証明書を使った接続では、「DN」フィールドの内容は証明書エントリの DN を意味し、パスワードは必要ありません。

   既存のレプリケーションアグリーメントでセキュリティ保護ありをセキュリティ保護なしに切り替えたり、その反対に切り替えることはできません。別のセキュリティ設定でレプリケーションを有効にするには、別のレプリケーションアグリーメントを作成する必要があります。

6. 「了解」をクリックして、変更を保存します。

レプリケーションアグリーメントの複製

レプリケーションアグリーメントを複製することで、大規模なレプリケーショントポロジでサプライヤレプリカの多数のコンシューマを簡単に設定できます。

1. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、レプリケートされたサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。

2. レプリケーションアグリーメントのリストから、複製するアグリーメントを 1 つ選択します。コンシューマとの接続をセキュリティ保護する新しいアグリーメントを作成するときは、セキュリティ保護されたポートを使用する既存のアグリーメントを選択する必要があります。セキュリティ保護されていないアグリーメントを新たに作成する場合は、セキュリティ保護されていないアグリーメントを選択する必要があります。

   「編集」をクリックして「レプリケーションアグリーメント」ダイアログのタブを表示し、このアグリーメントの設定を確認します。これらのタブで行う設定については、次の各項で説明しています。

   - 「接続」タブについては、310 ページの「レプリケーションマネージャの変更」を参照してください。
   - 「スケジュール」タブと「ネットワーク」タブについては、306 ページの「WAN を経由するレプリケーション」を参照してください。
   - 「レプリケートされた属性」タブについては、291 ページの「部分レプリケーションの設定」を参照してください。

3. 同じレプリケーションアグリーメントを選択した状態で、「複製」ボタンをクリックします。

4. 新しいコンシューマのホスト名とポート番号をリストから選択するか、「ホストを追加」ボタンをクリックして、別のホストとポートを指定します。リストと「ホストを追加」ダイアログでは、複製するコンシューマアグリーメントと同じ種類のセキュリティ (セキュリティ保護あり、またはなし) が設定されたコンシューマだけを選択できます。
5. リストのホスト名が選択されていることを確認し、「了解」をクリックして、そのコンシューマサーバーの新しいレプリケーションアグリーメントを作成します。
6. 新しいアグリーメントには、既存のアグリーメントのすべての設定情報が複製されます。つまり、2 つのサーバーにまったく同じレプリケーションマネージャエントリを定義し、同じパスワードを使用する必要があります。レプリケーションマネージャの DN の変更など、新しいアグリーメントの設定を変更するときは、リストからそのアグリーメントを選択し、「編集」をクリックします。

レプリケーションアグリーメントの無効化

レプリケーションアグリーメントを無効にすると、そのアグリーメントに指定されているコンシューマに対してマスターが更新を送信しなくなります。そのサーバーのレプリケーションは停止されますが、アグリーメントに記録されているすべての設定は残されます。あとからまたアグリーメントを有効にすることで、レプリケーションを再開できます。中断後のレプリケーションメカニズムの再開については、次の「レプリケーションアグリーメントの有効化」を参照してください。

レプリケーションアグリーメントを無効にする手順は、次のとおりです。

1. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、レプリケートされたサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。
2. 右側のパネルで、無効にするレプリケーションアグリーメントを選びます。
3. アグリーメントのリストの下で、「アクション」>「アグリーメントを無効に」の順に選択します。
4. 「はい」をクリックして、レプリケーションアグリーメントが無効になったことを確認します。

リスト上のアグリーメントのアイコンが変化し、無効になったことを示します。

レプリケーションアグリーメントの有効化

レプリケーションアグリーメントを有効にすると、指定のコンシューマのレプリケーションが再開されます。ただし、レプリケーションの回復設定で許容される時間より長くレプリケーションを中断していた場合は、別のサプライヤによるコンシューマの更新が行われないため、コンシューマを初期化し直す必要があります。レプリケーション回復設定は、このサプライヤの更新履歴ログの最大サイズと有効期限、およびコンシューマのパージ遅延です (280 ページの「コンシューマの詳細設定」を参照)。

中断時間が短く、レプリケーションが回復された場合は、アグリーメントがふたたび有効になったときに、マスターが自動的にそのコンシューマを更新します。

レプリケーションアグリーメントを有効にする手順は、次のとおりです。

1. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、レプリケートされたサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。
2. 右側のパネルで、有効にするレプリケーションアグリーメントを選びます。
3. アグリーメントのリストの下にある「有効」ボタンをクリックします。
4. 必要に応じて、コンシューマレプリカを初期化し直します。

レプリケーションアグリーメントの削除

レプリケーションアグリーメントを削除すると、対応するコンシューマのレプリケーションは停止され、アグリーメントに関するすべての設定情報が失われます。あとからレプリケーションを再開する必要があるときは、312 ページの「レプリケーションアグリーメントの無効化」の説明に従って、アグリーメントを削除せずに無効にします。

レプリケーションアグリーメントを削除する手順は、次のとおりです。

1. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、レプリケートされたサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。
2. 右側のパネルで、削除するレプリケーションアグリーメントを選びます。
3. アグリーメントのリストの右にある「削除」ボタンをクリックします。
4. 「はい」をクリックして、レプリケーションアグリーメントの削除を確認します。

レプリカの昇格と降格

レプリカの昇格と降格は、レプリケーショントポロジで、レプリカの役割を変更することを意味します。専用コンシューマをハブに変更したり、ハブをマスターに変更したりできます。また、マスターをハブに変更したり、ハブを専用コンシューマに変更したりすることもできます。ただし、マスターを直接コンシューマに格下げしたり、コンシューマを直接マスターに格上げすることはできません。

マルチマスターレプリケーションのメカニズムでレプリカの役割を変更できることで、トポロジがとても柔軟になります。コンシューマレプリカが処理を担当していたサイトの負荷が増え、複数のレプリカを持つハブによる処理が必要になることもあります。レプリカの内容に対して多数の変更が含まれるときは、ハブをマスターに昇格させることで、ローカルな変更に迅速に対応し、その変更を他のサイトの他のマスターにレプリケートできます。

レプリカの役割を変更する手順は、次のとおりです。

1. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、レプリケートされたサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。
2. 右側のパネルで、メニューから「変更」>「レプリカの昇格と降格」の順に選択します。
3. レプリケーションウィザードでは、設定できる新しい役割だけを選択できます。次に、そのレプリカの新しい役割について、順に設定を行います。このとき、次の点に注意が必要です。
   - マスターをハブに降格させると、レプリカは読み取り専用となり、残りのマスターに対してはリフェラルを送信するように設定されます。新しいハブは、設定されているすべてのコンシューマをハブまたは専用コンシューマとして維持します。
   - シングルマスターレプリケーションでマスターをハブに降格させると、マスターレプリカの存在しないトポロジが作成されます。新しいマスターを定義することを前提として、ウィザードでもこのような変更が可能です。ただし、マスターを降格させる前にマルチマスターとして新しいマスターを追加し、初期化できるようにしておくことをお勧めします。
   - ハブをコンシューマに降格させると、すべてのレプリケーションアグリーメントは削除されます。そのハブのコンシューマが他のハブまたはマスターによって更新されるように設定されていない場合、そのコンシューマは更新されなくなります。これらのコンシューマが更新されるように、残りのハブまたはマスターに新しいアグリーメントを作成する必要があります。
   - コンシューマをハブに昇格させると、更新履歴ログが有効になり、コンシューマとの間に新しいアグリーメントを定義できるようになります。
   - ハブをマスターに昇格させると、レプリカは更新要求を受け付けるようになり、他のマスター、ハブ、または専用コンシューマとの間に新しいアグリーメントを定義できるようになります。

レプリカの無効化

レプリカを無効にすると、そのレプリカはレプリケーショントポロジから除外されます。設定されている役割 (マスター、ハブ、またはコンシューマ) に応じて、そのレプリカは更新されなくなり、更新を送信しなくなります。サプライヤを無効にすると、すべてのレプリケーションアグリーメントが削除されます。そのレプリカをふたたび有効にするときは、これらのアグリーメントを作成し直す必要があります。

レプリカを無効にする手順は、次のとおりです。

1. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、レプリケートされたサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。
2. 右側のパネルで、メニューから「変更」>「レプリケーションを無効に」の順に選択します。
3. 確認ダイアログが表示されるので、「はい」をクリックします。
4. 必要に応じて、このサフィックスの書き込み権限とリフェラルをリセットします。これらの設定は、レプリカを無効にした時点の状態で残されます。たとえば、無効になったコンシューマは、無効になる以前のマスターレプリカに対して更新要求を送信します。

   書き込み権限とリフェラルを変更するには、「設定」タブでこのサフィックスのノードを選択し、右側のパネルの「設定」タブで変更を加えます。詳細は、121 ページの「アクセス権とリフェラルの設定」を参照してください。

更新履歴ログの移動

更新履歴ログは、特定のサプライヤレプリカに加えられるすべての変更を記録した内部レコードで、サーバーが他のレプリカに修正を加えるときに使われます。更新履歴ログの内容はサーバーによって自動的に管理され、サーバーを再起動したあとでも、マルチマスター更新によって更新されます。

旧バージョンの Directory Server では、LDAP から更新履歴ログにアクセスできました。しかし今回、更新履歴ログの形式が変更され、サーバーによる内部処理専用になりました。使用しているアプリケーションで更新履歴ログを読み取る必要がある場合は、旧バージョン形式の更新履歴ログプラグインを使用して、下位互換性を保つことができます。詳細は、324 ページの「旧バージョン形式の更新履歴ログプラグインの使用」を参照してください。

管理者が更新履歴ログの内容を変更する必要があるのは、ファイルが記録されるディスクがいっぱいになった場合など、このファイルを別の場所に移動するときだけです。

警告 更新履歴ログは、無効化したり、別の場所に移動したりすると、ふたたび初期化されます。どちらの場合も、このサーバーのレプリカのすべてのコンシューマを初期化し直す必要があります。

更新履歴ログの移動には、Directory Server コンソールを使用する必要があります。オペレーティングシステムの `rename` コマンドまたは `mv` コマンドを使用することはできません。

1. Directory Server コンソールの最上位の「設定」タブで「データ」ノードを選び、右側のパネルで「レプリケーション」タブを選びます。

2. 新しい場所をテキストフィールドに入力します。これは、更新履歴ログを格納する新しいパスとディレクトリ名です。たとえば、更新履歴ログをデフォルト位置の *ServerRoot*/slapd-*serverID*/changelogdb から *ServerRoot*/slapd-*serverID*/newchangelog に移動できます。

   古い場所にある既存の更新履歴ログは削除され、新しい場所に新しいログが作成されます。

3. 「レプリケーション」タブの「保存」をクリックします。
4. Directory Server を再起動します。
5. 294 ページの「レプリカの初期化」の説明に従って、コンシューマを初期化し直します。

レプリカの同期の維持

定期保守のためにレプリケーションに関連する Directory Server の停止後、オンライン状態に復帰させたときは、レプリケーションを介してそれが更新されていることをただちに確認する必要があります。特に、マルチマスター環境のマスターサーバーでは、マルチマスターセットのもう一つのサーバーからディレクトリ情報を更新する必要があります。マルチマスター以外の環境でも、ハブレプリカや専用コンシューマが保守のためにオフラインになった場合、オンラインに復帰したときは、マスターレプリカ側から更新を行う必要があります。

ここでは、レプリケーションの再試行アルゴリズムおよび次の実行まで待たずに、強制的にレプリケーション更新を行う方法について説明します。

注 ここで説明されている手順を利用できるのは、レプリケーションの設定が完了し、さらにコンシューマを初期化した直後だけです。

レプリケーションの再試行アルゴリズム

サプライヤがコンシューマのレプリケーションに失敗した場合は、時間間隔を大きくしながら定期的にレプリケーションを再試行します。再試行のパターンは、20、40、80、160 秒後です。その後、サプライヤは 160 秒おきに再試行を繰り返します。

サプライヤレプリカとコンシューマレプリカの間で、常に同期をとるレプリケーションアグリーメントを設定していても、オフライン状態の時間が 5 分を超えたレプリカをただちに最新の状態に戻すには、この方法では不十分です。

サーバーがオンライン状態に復帰した直後にディレクトリ情報を確実に同期させるには、Directory Server コンソールまたはカスタマイズ可能なスクリプトのどれかを利用できます。

コンソールによるレプリケーションの強制的な更新

コンシューマまたはマルチマスターレプリケーション設定のマスターが一定の時間を経過してオンライン状態に復帰したとき、レプリケーション更新をただちに送信させるためには、最新のディレクトリデータを保持しているサプライヤ上で次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、マスターレプリカのサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。

   右側のパネルにレプリカのステータス情報が表示されます。

2. 更新するコンシューマに対応するレプリケーションアグリーメントをリストから選択し、「アクション」>「ただちに更新を送信」の順にクリックします。

   これにより、更新が必要な情報を保持しているレプリカに対してレプリケーションが開始されます。

コマンド行によるレプリケーションの強制的な更新

サプライヤに対してレプリケーションの更新がただちに転送されるように要求するスクリプトを、更新が必要なコンシューマから実行することもできます。このスクリプトを、318 ページのコード例 8-1 に示します。

このスクリプト例をコピーして、適切な名前 (`replicate_now.sh` など) をつけてください。なお、コード例 8-1 のリストに含まれている変数には、実際の値を設定する必要があります。

注　このスクリプトは、サーバーがオフラインからオンラインに復帰したあとすぐに自動的に実行できるように設定できないため、管理者がこのスクリプトを実行する必要があります。

コード例 8-1 Replicate_Now スクリプトの使用例

```
#!/bin/sh
SUP_HOST=supplier_hostname
SUP_PORT=supplier_portnumber
SUP_MGRDN=supplier_directoryManager
SUP_MGRPW=supplier_directoryManager_passwd
MY_HOST=consumer_hostname
MY_PORT=consumer_portnumber

ldapsearch -1 -h ${SUP_HOST} -p ${SUP_PORT} -D "${SUP_MGRDN}" ¥
-w ${SUP_MGRPW} -b "cn=mapping tree, cn=config" ¥
"(&(objectclass=nsds5replicationagreement) ¥
(nsDS5ReplicaHost=${MY_HOST})(nsDS5ReplicaPort=${MY_PORT}))" ¥
dn nsds5ReplicaUpdateSchedule > /tmp/$$

cat /tmp/$$ |
awk '
BEGIN { s = 0 }
/^dn: / { print $0;
    print "changetype: modify";
    print "replace: nsds5ReplicaUpdateSchedule";
    print "nsds5ReplicaUpdateSchedule: 0000-2359 0123456";
    print "-";
    print "";
    print $0;
    print "changetype: modify";
    print "replace: nsds5ReplicaUpdateSchedule";
}

/^nsds5ReplicaUpdateSchedule: / { s = 1; print $0; }

/^$/ {
  if ( $s == 1 )
        { print "-" ; print ""; }
  else
    { print "nsds5ReplicaUpdateSchedule: 0000-2359 0123456";
      print "-" ; print ""; };
  s = 0; }

' > /tmp/ldif.$$

echo "Ldif is in /tmp/ldif.$$"
echo

ldapmodify -c -h ${SUP_HOST} -p ${SUP_PORT} -D "${SUP_MGRDN}" ¥
-w ${SUP_MGRPW} -f /tmp/ldif.$$
```

このスクリプトを使用する場合は、スクリプトに含まれている次の変数を、レプリケーション環境の実際の値に置き換える必要があります。

表 8-1 Replicate_Now 変数

変数	定義
supplier_hostname	現在のコンシューマとのレプリケーションアグリーメントに関する情報を取得するための、問い合わせ先サプライヤサーバーのホスト名
supplier_portnumber	サプライヤで使用中の LDAP ポート
supplier_directoryManager	サプライヤ上の Directory Manager 特権ユーザーの DN、または `cn=config` で書き込み権限を持つ `admin` ユーザーの DN
supplier_directoryManager_passwd	サプライヤ上の Directory Manager 特権ユーザーまたは `admin` ユーザーのパスワード
consumer_hostname	現在のコンシューマのホスト名
consumer_portnumber	コンシューマで使用中の LDAP ポート

SSL 接続を経由して更新処理を実行する場合、スクリプト中の `ldapmodify` コマンドを適切なパラメータと値で置き換える必要があります。詳細は、393 ページの「LDAP クライアントでセキュリティを使用するための設定」を参照してください。

旧バージョンからのレプリケーション

ここでは、Sun ONE Directory Server の旧バージョンからのレプリケーションを設定する方法について説明します。

Sun ONE Directory Server 5.1 および 5.2 は、次の例外を除いてレプリケーションの設定については完全に互換性があります。

- Directory Server 5.2 マスターレプリカと 5.1 コンシューマレプリカの間では、部分レプリケーションは行うことはできないので設定できません。
- 5.2 マスターと 5.1 コンシューマの間のアグリーメントを設定する前に、`cn=config` で `nsslapd-schema-repl-useronly` を `on` に設定する必要があります。この設定を行わないと、5.1 にレプリケートするときに、5.2 のスキーマによって競合が発生します。この設定により、`99user.ldif` ファイルに格納されているユーザー定義のスキーマエレメントだけがレプリケートされます。詳細は、348 ページの「スキーマ定義のレプリケーション」を参照してください。
- Directory Server 5.2 では、RFC 2307 に合わせてスキーマ ファイル 11rfc2307.ldif が変更されています。322 ページの「Directory Server 5.1 スキーマの更新」で説明する手順に従って、5.1 サーバーで対応するファイルを更新する必要があります。
- ハブに降格された 5.2 マスターは、5.1 コンシューマのリフェラルのリストに残ります。ただし、降格の内部メカニズムにより、降格されたレプリカのポート番号はゼロになります。このリフェラル URL は使用できないため、ほとんどのクライアントは他のマスターへのリフェラルを自動的に試行します。しかし、これらの 5.1 レプリカにアクセスするクライアントのリフェラルでは、ホップ制限を引き上げる必要があります。5.2 コンシューマレプリカは、降格されたマスターを表示しません。また、そのマスターへの有効なリフェラル URL を返しません。

Sun ONE Directory Server 5.2 がリリース 4.x の Directory Server と組み合わせたレプリケーションに関与できるのは、次の場合です。

- Directory Server 5.2 がマスターとして設定され、Directory Server 4.x サプライヤのコンシューマとしてだけレプリケートされる場合
- コンシューマレプリカは、4.x サプライヤと 5.2 サプライヤの両方のコンシューマとなることはできない。ただし、5.2 サーバーは異なる種類のレプリカ (旧バージョンの Directory Server から提供されるレプリカと 5.2 Directory Server から提供されレプリカ) を保持できる
- 4.x サプライヤのコンシューマとして設定された Directory Server 5.2 レプリカは、トポロジのこのサフィックスではハブレプリカとして機能できない

旧バージョンの Directory Server のコンシューマとして Directory Server 5.2 を使用できる利点は、レプリケートされた環境を簡単に移行できることです。レプリケートされた環境を移行するための手順について詳細は、『Sun ONE Directory Server インストールおよびチューニングガイド』の第 2 章にある「旧バージョンからのアップデート」を参照してください。

Directory Server 4.x のコンシューマとしての Directory Server 5.2 の設定

リリース 4.x の Directory Server のコンシューマとして Directory Server 5.2 を使うときは、次のように設定する必要があります。

1. 286 ページの「マスターレプリカの有効化」の説明に従って、レプリカをマスターレプリカとして有効にします。レプリカは 4.x サプライヤのコンシューマとなりますが、マスターレプリカとして設定する必要があります。
2. Directory Server コンソールの最上位の「設定」タブで、「データ」ノードを展開し、レプリケートされたサフィックスのノードを展開し、サフィックスの下の「レプリケーション」ノードを選択します。
3. このレプリカについて、右側のパネルで「変更」>「4.x との互換性を有効に」の順に選択します。または、「オブジェクト」メニューから「4.x との互換性を有効に」を選びます。
4. 「4.x との互換性を有効に」ウィンドウが表示されるので、旧バージョンのサプライヤサーバーがバインドに使用するバインド DN とパスワードを指定します。バインド DN には、デフォルトのレプリケーションマネージャを含め、どのような管理エントリでも使用できます。バインド DN については、277 ページの「レプリケーションマネージャの選択」を参照してください。

   サプライヤがレプリケーション更新にサーバーのセキュリティ保護されたポートを使うときは、セキュリティ保護された認証を行えるように、サーバーの証明書エントリの DN を指定できます。
5. 「了解」をクリックします。これで、このコンシューマレプリカは旧バージョンのサプライヤから更新を受信できます。
6. 5.2 レプリカサーバーのスキーマが、4.x マスターからレプリケートされる内容が使用するすべての属性とオブジェクト クラスを定義していることを確認してください。
7. 4.x マスターで作成された LDIF レプリカファイルをインポートして、5.2 レプリカを初期化します。このファイルの最初のエントリには、4.x レプリケーションメカニズムが必要とする `copiedfrom` 属性が含まれます。

サーバーで 4.x との互換性を有効にすると、デフォルトでインストールされている旧バージョンとのレプリケーションのプラグインが設定されます。このプラグインは、旧バージョンのサプライヤからの更新を処理し、レプリケートされるサフィックスの内容を更新します。

注 4.x との互換性を有効にしている限り、このレプリカはクライアントからすべての変更要求に対してリフェラルを返します。Directory Server 5.2 をマスターレプリカとして設定しても、このサフィックスでは変更要求は行われません。その代わりに、4.x サプライヤサーバーへのリフェラルが返されます。

旧バージョンのレプリケーションの設定を完了するには、Directory Server 5.2 にレプリケートする旧バージョンのサプライヤを設定する必要があります。4.x Directory Server にレプリケーションアグリーメントを設定する手順については、旧バージョンの Directory Server のマニュアルを参照してください。

Directory Server 5.1 スキーマの更新

Directory Server 5.2 では、RFC 2307 に合わせてスキーマ ファイル `11rfc2307.ldif` が変更されています (`http://www.ietf.org/rfc/rfc2307.txt`)。5.2 サーバーと 5.1 サーバーの間でレプリケーションを設定または有効化する前に、5.1 サーバー側のスキーマを更新する必要があります。どちらのバージョンのサーバーでも、スキーマ ファイルは *ServerRoot*`/slapd-`*serverID*`/config/schema/` に保存されています。

1. 5.2 サーバーから 5.1 サーバーに `11rfc2307.ldif` ファイルをコピーします。

- 5.1 サーバーの Solaris パッケージインストールを利用しているときは、古くなった `10rfc2307.ldif` ファイルを削除します。
- その他のプラットフォームで zip ファイルのインストールを利用しているときは、既存の `11rfc2307.ldif` ファイルを上書きします。

2. この変更の影響を受けるスキーマ ファイルは次のとおりです。これらのファイルを 5.2 サーバーから 5.1 サーバーにコピーして、既存のファイルを上書きする必要があります。
   - `20subscriber.ldif`
   - `30ns-common.ldif`
   - `50ns-admin.ldif`
   - `50ns-certificate.ldif`
   - `50ns-directory.ldif`

- 50ns-legacy.ldif
- 50ns-mail.ldif
- 50ns-mlm.ldif
- 50ns-msg.ldif
- 50ns-netshare.ldif

3. 5.1 サーバーを再起動し、レプリケーションの設定とレプリカの初期化に進みます。他のスキーマエレメントとの同期によって、サーバー間でレプリケートされるスキーマ属性もありますが、これはレプリケーションメカニズムによる正常な動作です。
4. 古いバージョンのスキーマに依存するアプリケーションでは、更新が必要になる可能性があります。新しい 11rfc2307.ldif ファイルには、次の変更が加えられています。
   - automount 属性と automountInformation 属性は削除されました。
   - ipHost オブジェクトクラスの使用可能属性リストから o $ ou $ owner $ seeAlso $ serialNumer が削除されました。
   - ieee802Device オブジェクトクラスの必須属性リストから cn が削除されました。
   - ieee802Device オブジェクトクラスの使用可能属性リストから description $ l $ o $ ou $ owner $ seeAlso $ serialNumber が削除されました。
   - bootableDevice オブジェクトクラスの必須属性リストから cn が削除されました。
   - bootableDevice オブジェクトクラスの使用可能属性リストから description $ l $ o $ ou $ owner $ seeAlso $ serialNumber が削除されました。
   - nisMap オブジェクトクラスの OID が 1.3.6.1.1.1.2.9 に変更されました。

旧バージョン形式の更新履歴ログプラグインの使用

旧バージョン形式の更新履歴ログプラグインを使用すると、Directory Server 5.2 マスターレプリカで 4.x スタイルの更新履歴ログを維持できます。このプラグインは、Directory Server 4.x の形式の更新履歴ログに依存している、Sun ONE Meta Directory などのアプリケーションで必要になる場合があります。これは、アプリケーションが更新履歴ログからデータを読み取るためです。

旧バージョン形式の更新履歴ログプラグインでは、Directory Server 5.2 を 4.x コンシューマレプリカのサプライヤとすることはできません。320 ページの「旧バージョンからのレプリケーション」で説明したように、4.x サプライヤの Directory Server 5.2 コンシューマだけがサポートされます。旧バージョン形式の更新履歴ログプラグインは、レプリケーションプロトコルとは無関係に動作し、レプリケーショントポロジに影響しません。旧バージョン形式の更新履歴ログプラグインは、シングルマスター構成のどのサーバーでも有効にできます。これは、マルチマスター環境では正しく動作しないので、この環境では有効にしないでください。

旧バージョン形式の更新履歴ログは、サーバーの 5.2 更新履歴ログとは別に維持されます。旧バージョン形式の更新履歴ログは、`cn=changelog` という特別なサフィックスの下で、独立したデータベースに格納されます。旧バージョン形式の更新履歴ログは、単一レベルの複数のエントリから構成されます。更新履歴ログの各エントリには `changeLogEntry` というオブジェクトクラスがあり、次の表に一覧表示されている属性を含めることができます。

表 8-2 旧バージョン形式の更新履歴ログエントリの属性

属性	定義
`changeNumber`	1 つの値からなる属性で、常に存在する。各変更を識別する一意の整数を含む。この値は、変更の順序を示す。値が大きいほど、変更は新しい
`targetDN`	この属性には、LDAP 処理の影響を受けるエントリの DN が含まれる。`modrdn` 処理の場合、`targetDN` 属性には、変更または移動される前のエントリの DN が含まれる
changeTime	この属性は、変更操作が行われた時刻を特定する
`changeType`	LDAP 処理の種類を特定する。この属性の値には、`add`、`delete`、`modify`、`modrdn` のいずれかを指定できます。
`changes`	`add` または `modify` 処理の場合、エントリに対する更新が LDIF 形式で記録される
`newRDN`	`modrdn` 処理の場合、エントリの新しい RDN が特定される
`deleteOldRdn`	`modrdn` 処理の場合、古い RDN が削除されたかどうかを示す

表 8-2　旧バージョン形式の更新履歴ログエントリの属性 (続き)

属性	定義
newSuperior	modrdn 処理の場合、エントリの newSuperior 属性を示す

旧バージョン形式の更新履歴ログプラグインの有効化

旧バージョン形式の更新履歴ログプラグインの設定情報は、dse.ldif の cn=Retro Changelog Plugin,cn=plugins,cn=config エントリに保持されます。

Directory Server コンソールから旧バージョン形式の更新履歴ログプラグインを有効にする手順は、次のとおりです。

1. Directory Server コンソールの最上位の「設定」タブで、「プラグイン」ノードを展開し、下にスクロールして「旧バージョン形式の更新履歴ログプラグイン (Retro Changelog Plugin)」を選択します。
2. 右側のパネルで、「プラグインを有効に」チェックボックスを選択し、「保存」をクリックします。プラグインを無効にするには、このチェックボックスの選択を解除します。
3. プラグインを有効または無効にしたときは、Directory Server を再起動する必要があります。

コマンド行から旧バージョン形式の更新履歴ログプラグインを有効にする手順は、次のとおりです。

1. 次のコマンドを実行して、旧バージョン形式の更新履歴ログプラグインの設定エントリを変更します。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=Retro Changelog Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginenabled
nsslapd-pluginenabled: on
```

2. サーバーを再起動します。サーバーの再起動については、20 ページの「Directory Server の起動と停止」を参照してください。

旧バージョン形式の更新履歴ログの削除

更新履歴ログのエントリは、指定した一定時間後に自動的に削除されます。更新履歴ログからエントリを自動的に削除する期間を指定するには、cn=Retro Changelog Plugin、cn=plugins、cn=config エントリで nsslapd-changelogmaxage 設定属性を設定する必要があります。この属性を設定するには、次のようにコマンド行だけで設定できます。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -p password
dn: cn=Retro Changelog Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-changelogmaxage
nsslapd-changelogmaxage: IntegerTimeunit
```

nsslapd-changelogmaxage 属性は、次の形式の単一値属性です。

```
nsslapd-changelogmaxage: IntegerTimeunit
```

ここで、*Integer* は数字を表し、*TimeUnit* の s は秒、m は分、h は時間、d は日数、および w は週を表します。次の例のように、*Integer* 変数と *Timeunit* 変数の間には空白を挿入しません。

```
nsslapd-changelogmaxage: 2d
```

旧バージョン形式の更新履歴ログは、更新記録ログに対する次回の処理時に削除されます。

旧バージョン形式の更新履歴ログへのアクセス

更新履歴ログは検索処理をサポートし、次の形式のフィルタを含む検索用に最適化されています。

```
(&(changeNumber>=X)(changeNumber<=Y))
```

一般的な規則として、更新履歴ログのサイズを小さくするためにエントリを削除するとしても、旧バージョン形式の更新履歴ログでは追加または変更処理は実行すべきではありません。旧バージョン形式の更新履歴ログで修正処理を実行する必要があるのは、デフォルトのアクセス制御ポリシーを修正する場合だけです。

旧バージョン形式の更新履歴ログが作成されると、次のアクセス制御規則がデフォルトで適用されます。

- 旧バージョン形式の更新履歴ログのトップエントリ cn=changelog に対する読み取り、検索、および比較の権限は、すべての認証ユーザー (userdn=anyone のユーザー。userdn=all で指定された匿名アクセスとは異なる) に付与される
- Directory Manager に対する暗黙の了承を除き、書き込みおよび削除アクセスは付与されない

更新履歴ログのエントリにはパスワードなどの重要な情報が含まれている場合があるので、読み取りアクセス権を匿名ユーザーに付与しないでください。認証されたユーザーにも内容の表示が許可されない場合でも、旧バージョン形式の更新履歴ログの内容へのアクセスをさらに制限することが必要なことがあります。

旧バージョン形式の更新履歴ログに対するデフォルトのアクセス制御ポリシーを変更するには、`cn=changelog` エントリの `aci` 属性を変更する必要があります。`aci` 属性の設定については、第 6 章「アクセス制御の管理」を参照してください。

レプリケーション状態の監視

新しいコマンド行ツールと Directory Server コンソールを使って、レプリケーションの状態を監視できます。

コマンド行ツール

レプリケーションの状態の監視には、次の 3 種類の新しいコマンド行ツールを利用できます。

- `repldisc` : レプリケーションに含まれるすべての既知のサーバーを検出し、テーブルを作成する
- `insync` : サプライヤレプリカと、1 つまたは複数のコンシューマレプリカの間の同期状態を示す
- `entrycmp` : 複数のレプリカに含まれる同じエントリを比較する

これらのツールは、次のディレクトリにあります。

ServerRoot`/shared/bin`

これらのツールの完全なコマンド行構文と使用例については、『Sun ONE Directory Server Reference Manual』の第 1 章にある「Replication Monitoring Tools」を参照してください。

レプリケーション状態タブ

Directory Server コンソールでレプリケーション状態の概要を確認する手順は、次のとおりです。

1. Directory Server コンソールの最上位の「状態」タブで、「レプリケーション」ノードを選択します。

   このサーバーに設定されている各レプリケーションアグリーメントに関する情報が、右側のパネルに表示されます。

2. レプリケーションの状態を監視するには、「継続して再表示」チェックボックスを選択します。たとえば、レプリカの初期化がいつ完了したかを確認できます。

3. コンシューマにまだレプリケートされていないマスター側の最後の変更を調べるには、「保留中の変更数」ボタンをクリックします。この処理は時間がかかることがあるため、実行するかどうかを確認するメッセージが表示されます。保留中の変更数を調べるには、コンシューマ側の更新レコードをダウンロードし、それをマスター側の更新履歴ログと比較する必要があります。これらのログのサイズが大きい場合、多くの時間とサーバーリソースが使われます。

4. 列の見出しをクリックしてサイズを変更することで、テーブルのレイアウトを変更できます。また、「表示オプション」ボタンをクリックして、表示する列だけを指定することもできます。次の表 8-3 は、このサーバーの各アグリーメントのテーブルの表示について指定できるレプリケーションパラメータを示しています。

表 8-3 Directory Server コンソールの状態タブのレプリケーションパラメータ

テーブルの見出し	内容
サフィックス	レプリケートされるサフィックスまたはサブサフィックスの名前
リモートレプリカ	コンシューマサーバーのホスト名とポート
内容	このレプリケーションアグリーメントに設定されている説明文
状態	アグリーメントが無効になっている、コンシューマを初期化している、通常の増分更新でレプリケーションを行なっている、のどれか
概要	最後のイベント (初期化または更新の開始または終了) と、最後に受信したメッセージ
送信済みの更新	レプリケーションが有効になってから、またはサーバーが再起動されてからコンシューマに更新が送信された合計回数
最終更新開始時間	最後のレプリケーションの更新開始日時

表 8-3　Directory Server コンソールの状態タブのレプリケーションパラメータ (続き)

テーブルの見出し	内容
最終更新終了時間	最後のレプリケーションの更新終了日時
最終更新メッセージ	最後のレプリケーション更新の状態
最終初期化メッセージ	最後の初期化後のコンシューマの状態
最終初期化開始時間	最後のコンシューマレプリカの初期化開始日時
最終初期化終了時間	最後のコンシューマレプリカの初期化終了日時

よく発生するレプリケーションの競合の解決

マルチマスターレプリケーションでは、疎整合型レプリケーションモデル (Loose Consistency Replication Model) を使用します。つまり、同一のエントリを別々のサーバーから同時に変更できます。2 つのサーバー間で更新が送信された場合、更新の競合を解決する必要があります。ほとんどの場合、各サーバーの変更に関連するタイムスタンプに基づいて競合は自動的に解決されます。時刻の新しい更新が優先されます。

ただし、更新の競合を解決するためにユーザーの介入が必要となる場合もあります。レプリケーションプロセスで自動的に解決できない更新の競合があるエントリには、競合マーカーとしてのオペレーショナル属性 `nsds5ReplConflict` が含まれます。

この属性を含むエントリを定期的に検索することで、競合が生じているエントリを探すことができます。たとえば、次の `ldapsearch` コマンドを使用できます。

```
% ldapsearch -h host -p port -D "cn=Directory Manager" -w password ¥
-b "dc=example,dc=com" "(nsds5ReplConflict=*)"
```

`nsds5ReplConflict` 属性には、デフォルトでインデックスが設定されています。

ネーミングの競合の解決

同じ DN で異なるサーバーに 2 つのエントリを作成すると、レプリケーションの競合解決メカニズムによって、2 番目に作成されたエントリの名前が自動的に変更されます。すべてのディレクトリエントリには、`nsuniqueid` オペレーショナル属性が指定する一意の識別子があり、ネーミングの競合が発生すると、一意でない DN の名前に一意の ID が追加されます。

最初のサーバーが 2 番目のサーバーに変更をレプリケートする前に 2 番目のサーバーでエントリが作成された場合、同じ DN を持つエントリが作成される可能性があります。たとえば、2 つのマスターで uid=bjensen,ou=People,dc=example,dc=com というエントリが同時に作成されると、レプリケーション後の 2 つのエントリは、次のようになります。

- uid=bjensen,ou=People,dc=example,dc=com
- nsuniqueid=66446001-1dd211b2+uid=bjensen,dc=example,dc=com

2 番目のエントリは、DN が一意になるように名前を変更する必要があります。競合しているエントリを削除し、競合しない名前で追加し直すことができます。ただし、作成時に名前を変更する方法が最も確実です。名前変更の手順は、ネーミング属性が 1 つの値を持つか複数の値を持つかによって異なります。各手順は次のとおりです。

複数の値からなるネーミング属性を持つエントリの名前変更

複数の値からなるネーミング属性を持つ競合エントリの名前を変更する手順は、次のとおりです。

1. ネーミング属性の新しい値を使用してエントリの名前を変更し、古い RDN を保持しておきます。たとえば、次のようにします。

   ldapmodify -h *host* -p *port* -D "cn=Directory Manager" -w *password*

   ```
   dn: nsuniqueid=66446001-1dd211b2+uid=bjensen,dc=example,dc=com
   changetype: modrdn
   newrdn: uid=NewValue
   deleteoldrdn: 0
   ^D
   ```

2. ネーミング属性の古い RDN 値と競合マーカー属性を削除します。たとえば、次のようにします。

   ldapmodify -h *host* -p *port* -D "cn=Directory Manager" -w *password*

   ```
   dn: uid=NewValue,dc=example,dc=com
   changetype: modify
   delete: uid
   uid: bjensen
   -
   delete: nsds5ReplConflict
   ^D
   ```

注　一意の識別子属性 nsuniqueid は削除できないため、RDN の変更は 2 段階で実行されます。

1 つの値からなるネーミング属性を持つエントリの名前変更

ネーミング属性が 1 つの値の場合は、エントリの名前を単に同じ属性の別の値に変更することはできません。その代わりに、一時的に次の処理を行う必要があります。

1. 別のネーミング属性を使用してエントリの名前を変更し、古い RDN を保持しておきます。たとえば、次のようにします。

   `ldapmodify` -h *host* -p *port* `-D` "cn=Directory Manager" `-w` *password*

   ```
   dn: nsuniqueid=66446001-1dd211b2+dc=HR,dc=example,dc=com
   changetype: modrdn
   newrdn: o=TempName
   deleteoldrdn: 0
   ^D
   ```

2. ネーミング属性の古い RDN 値と競合マーカー属性を削除します。たとえば、次のようにします。

   `ldapmodify` -h *host* -p *port* `-D` "cn=Directory Manager" `-w` *password*

   ```
   dn: o=TempName,dc=example,dc=com
   changetype: modify
   replace: dc
   dc: uniqueValue
   -
   delete: nsds5ReplConflict
   ^D
   ```

注 一意の識別子属性 `nsuniqueid` は削除できないため、RDN の変更は 2 段階で実行されます。

3. 新しい競合しない値をネーミング属性に指定して、エントリの名前を変更します。たとえば、次のようにします。

   `ldapmodify` -h *host* -p *port* `-D` "cn=Directory Manager" `-w` *password*

   ```
   dn: o=TempName,dc=example,dc=com
   changetype: modrdn
   newrdn: dc=uniqueValue
   deleteoldrdn: 1
   ^D
   ```

`deleteoldrdn` 属性の値に `1` を設定すると、一時的な属性と値のペアである `o=`*TempName* が削除されます。この属性を保持する場合は、`deleteoldrdn` 属性の値に `0` を設定します。

親のないエントリの競合の解決

エントリの削除操作がレプリケートされたとき、コンシューマサーバーが削除されるエントリが子エントリを持つことを検出した場合、競合解決処理によって *glue* エントリが作成され、親のないエントリをディレクトリに持つことを回避します。

同様に、エントリの追加後にレプリケーションが実行され、コンシューマサーバーが追加されたエントリの親エントリを検出できなかった場合も、競合解決処理は親を表す glue エントリを作成し、親のないエントリが追加されることを回避します。

glue エントリは、`glue` および `extensibleObject` というオブジェクトクラスを持つ一時的なエントリです。glue エントリは、次のいくつかの方法で作成されます。

- 競合解決処理が、一致する一意の識別子をともなう削除されるエントリを検出した場合、glue エントリは、`glue` オブジェクトクラスと `nsds5ReplConflict` 属性を加えて、そのエントリを復元する

  この場合は、glue エントリを修正して `glue` オブジェクトクラスと `nsds5ReplConflict` 属性を削除し、通常のエントリに戻すか、または glue エントリとその子エントリを削除します。

- サーバーによって、`glue` および `extensibleObject` オブジェクトクラスを持つ必要最小限のエントリが作成される

  このような場合は、意味のあるエントリになるようにエントリを修正するか、またはエントリとその子エントリをすべて削除します。

潜在的な相互運用性の問題の解決

メールサーバーのように属性の一意性に依存するアプリケーションとの相互運用性の理由から、`nsds5ReplConflict` 属性を持つエントリへのアクセスを制限する必要がある場合があります。これらのエントリへのアクセスを制限しない場合は、1 つの属性だけを要求するアプリケーションが元のエントリと `nsds5ReplConflict` を含む競合解決エントリの両方を取得し、処理が失敗します。

アクセスを制限するには、次のコマンドを使用して、匿名の読み取りアクセスを許可するデフォルトの ACI を変更する必要があります。

`ldapmodify -h` *hostname* `-D "cn=Directory Manager" -w` *password*

```
dn: dc=example,dc=com
changetype: modify
delete: aci
aci: (target ="ldap:///dc=example,dc=com")
 (targetattr !="userPassword"
 (version 3.0;acl "Anonymous read-search  access";
 allow (read, search, compare)(userdn = "ldap:///anyone");)
```

```
-
add: aci
aci: (target="ldap:///dc=example,dc=com")
 (targetattr!="userPassword")
 (targetfilter="(!(nsds5ReplConflict=*))")(version 3.0;acl
 "Anonymous read-search access";allow (read, search, compare)
 (userdn="ldap:///anyone");)
^D
```

新しい ACI フィルタは、検索結果から `nsds5ReplConflict` 属性を持つすべてのエントリを除外します。

第 9 章

ディレクトリスキーマの拡張

Sun ONE Directory Server には、数多くのオブジェクトクラスおよび属性を持つ標準のスキーマが付属しています。通常の作業では標準のオブジェクトクラスと属性で十分ですが、新しいオブジェクトクラスや属性の作成など、スキーマの拡張が必要となることもあります。

この章では、スキーマの拡張方法について、次の項目ごとに説明します。

- スキーマ検査
- スキーマ拡張の概要
- 属性定義の管理
- オブジェクトクラス定義の管理
- スキーマ定義のレプリケーション

スキーマ検査

スキーマ検査を有効にすると、インポート、追加、変更のすべての処理が Directory Server によって、次のように現在定義されているディレクトリスキーマに準拠するようになります。

- 各エントリのオブジェクトクラスと属性は、スキーマに準拠する
- エントリには、そのエントリに定義されているすべてのオブジェクトクラスに必要なすべての属性が含まれる
- エントリには、そのエントリのオブジェクトクラスに許可されている属性だけが含まれる

注 エントリを変更すると、Directory Server は変更される属性だけでなく、エントリ全体に対してスキーマ検査を行います。このため、エントリのいずれかのオブジェクトクラスまたは属性がスキーマに準拠していない場合、変更処理は失敗します。

Directory Server では、デフォルトでスキーマ検査がオンになっています。Directory Server の起動中は、常にスキーマ検査をオンにしておくべきです。多くのクライアントアプリケーションでは、スキーマ検査をオンにしておくことは、すべてのエントリがスキーマに準拠しているものと見なされます。

しかし、スキーマ検査をオンにしても、ディレクトリ内の既存のコンテンツが検証されるわけではありません。ディレクトリのすべての内容がスキーマに確実に準拠するようにするには、エントリを追加する前、またはすべてのエントリを再初期化する前にスキーマ検査をオンにする以外に方法はありません。

スキーマ検査をオフにするのは、スキーマに準拠していることが確実な LDIF ファイルのインポートを高速で処理するときだけです。しかし、その場合、スキーマに準拠しないエントリがインポートされるリスクがあり、この違反は検出されません。

エントリがスキーマに準拠していない場合は、このエントリを検索することはできず、そのエントリに対する変更処理も失敗します。エントリがスキーマに準拠するようにするには、次の処理を行う必要があります。

1. サーバーが運用環境にある場合は、サーバー全体を読み取り専用にして、スキーマ検査がオフの状態で変更が行われないようにします。詳細は、36 ページの「グローバルな読み取り専用モードの設定」を参照してください。
2. 後述する方法でスキーマ検査をオフにします。
3. エントリを検索し、そのエントリと現在定義されているスキーマを手動で比較して、準拠していない原因を特定します。340 ページの「属性の表示」および 344 ページの「オブジェクトクラスの表示」を参照してください。
4. スキーマに準拠するようにエントリを修正します。

   準拠していないエントリが多い場合に、それらのエントリにパターンがあったり、新しいデータ形式が採用されているときは、エントリではなくスキーマの変更を検討します。ただし、スキーマへの変更を最小限にするために、配備前にスキーマを計画する必要があります。詳細については、『Sun ONE Directory Server Deployment Guide』の第 3 章「Designing the Schema」を参照してください。
5. 後述する方法でスキーマ検査をオンにします。
6. グローバルな読み取り専用モードを有効にしていた場合は、これをオフにします。

コンソールからのスキーマ検査の設定

1. Directory Server コンソールの最上位にある「設定」タブで、設定ツリーからスキーマのノードを選択します。

   右側のパネルにスキーマの定義が表示されます。

2. パネル上部のステータスメッセージには、スキーマ検査が現在有効であるか、無効であるかが示されます。その右のボタンをクリックすると、スキーマ検査のオン、オフが切り替わります。

   - スキーマ検査をオフにするときは、ボタン名は「無効」と表示されています。
   - スキーマ検査をオンにするときは、ボタン名は「有効」と表示されています。

   新しいスキーマ検査ポリシーは、直ちに適用されます。

コマンド行からのスキーマ検査の設定

`cn=config` エントリの `nsslapd-schemacheck` 属性を使用して、スキーマ検査のオン / オフを切り替えることもできます。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=config
changetype: modify
replace: nsslapd-schemacheck
nsslapd-schemacheck: on or off
```

新しいスキーマ検査ポリシーは、サーバーによって直ちに適用されます。

スキーマ拡張の概要

スキーマに新しい属性を追加する場合は、それらの属性を持つオブジェクトクラスを新しく作成する必要があります。必要な属性のほとんどが含まれている既存のオブジェクトクラスに対して、新たに必要となった属性を追加すると、LDAP クライアントとの相互運用性が低下するためです。

Directory Server と既存の LDAP クライアントとの相互運用性は、標準の LDAP スキーマに依存しています。標準スキーマを変更すると、サーバーのアップグレード時にも問題が発生します。同様の理由から、標準スキーマの要素を削除することはできません。

オブジェクトクラス、属性、およびディレクトリスキーマと、スキーマ拡張のガイドラインについては、『Sun ONE Directory Server Deployment Guide』の第 3 章「Designing the Schema」を参照してください。標準の属性とオブジェクトクラスについては、『Sun ONE Directory Server Reference Manual』の第 4 部「Directory Server Schema」を参照してください。

Directory Server スキーマは、ディレクトリの `cn=schema` エントリの属性内に格納されます。設定エントリと同様に、これは、サーバーの起動中にファイルから読み取られる、スキーマの LDAP ビューです。スキーマ ファイルは、次の場所にある LDIF ファイルです。

ServerRoot`/slapd-`*serverID*`/config/schema`

このディレクトリには、Directory Server が使用する標準のスキーマと、Directory Server に依存するその他の Sun ONE サーバーのファイルが含まれます。これらのファイルについて詳細は、『Sun ONE Directory Server Reference Manual』の第 9 章にある「Schema Supported by Directory Server 5.2」を参照してください。標準のスキーマ自体は、『Sun ONE Directory Server Reference Manual』の第 10 章「Object Class Reference」および第 11 章「Attribute Reference」で説明しています。

スキーマファイルの変更

サーバーは、起動時に 1 回だけスキーマファイルを読み取ります。スキーマのメモリ内の LDAP ビューの `cn=schema` 内にファイルの LDIF の内容が追加されます。スキーマ定義の順序には意味があるため、スキーマファイルの名前の先頭には番号がつけられ、英数字順に読み込まれます。このディレクトリに含まれるスキーマファイルには、インストール時に定義されたシステムユーザーだけが書き込み処理を実行できます。

ファイル内のスキーマ定義を変更するには、該当するファイルを作成または変更し、サーバーを再起動する必要があります。スキーマファイル内の定義の構文については、RFC 2252 (`http://www.ietf.org/rfc/rfc2252.txt`) を参照してください。

スキーマを LDIF ファイルに直接定義するときは、X-ORIGIN フィールドの値として 'user defined' を指定することはできません。この値は、cn=schema の LDAP ビューで定義されるスキーマエレメント用に予約されており、これは 99user.ldif に表示されます。

99user.ldif ファイルには、cn=schema エントリと、コマンド行またはコンソールから追加されたすべてのスキーマ定義の追加 ACI が含まれます。新しいスキーマ定義を追加すると、99user.ldif ファイルは上書きされます。このファイルを変更するときは、変更が永続的に適用されるように、サーバーを直ちに再起動する必要があります。

他のスキーマファイルに定義されている標準のスキーマを変更することはできません。ただし、新しいファイルを追加して、新しい属性やオブジェクトクラスを定義することはできます。たとえば、複数のサーバーに新しいスキーマ要素を定義するには、98mySchema.ldif という名前をファイルにその要素を定義し、このファイルをすべてのサーバーのスキーマディレクトリにコピーします。次にすべてのサーバーを再起動して、新しいスキーマファイルを読み込みます。

コマンド行からのスキーマの変更

スキーマは cn=schema 内の LDAP ビューによって定義されるため、ldapsearch ユーティリティおよび ldapmodify ユーティリティを使ってスキーマをオンラインで表示、変更することができます。しかし、変更できるスキーマ要素は、X-ORIGIN フィールドに 'user defined' という値が設定されている要素だけです。サーバーは、その他の定義に対するすべての変更処理を拒否します。

attributeTypes 属性と objectClasses 属性の値を個別に追加、削除するには、ldapmodify を使います。これらの属性は複数の値をとるので、1 つの値を変更するには、その値を削除してから新しい値を追加する必要があります (70 ページの「複数値属性の 1 つの値の変更」を参照)。スキーマ要素は、RFC 2252 (http://www.ietf.org/rfc/rfc2252.txt) で説明されている構文で定義します。

新しい要素の定義とユーザー定義の要素に対する変更は、99user.ldif ファイルに保存されます。

コマンド行からのスキーマ定義の変更は、正確な入力が必要な値が長いため、エラーを生じがちです。しかし、ディレクトリスキーマの更新が必要なスクリプトにこの機能を指定することができます。

コンソールからのスキーマの変更

ディレクトリスキーマをカスタマイズするときは、ここで説明する Directory Server コンソールインタフェースを使う方法をお勧めします。コンソールには標準スキーマが表示され、グラフィカルインタフェースを使って新しい属性やオブジェクトクラスを定義したり、定義した要素を編集することができます。

この場合も、新しい要素の定義とユーザー定義の要素に対する変更は、`99user.ldif` ファイルに保存されます。

ディレクトリスキーマを拡張するには、次の手順を実行します。

1. 342 ページの「属性の作成」で説明している手順に従って新しい属性を作成します。
2. 次に、オブジェクトクラスを作成し、そのオブジェクトクラスに新しい属性を追加します。詳細は、345 ページの「オブジェクトクラスの作成」を参照してください。

属性定義の管理

Directory Server コンソールでは、スキーマに含まれるすべての属性を表示し、ユーザー独自の属性定義を作成、編集、削除できます。

属性の表示

現在ディレクトリスキーマにあるすべての属性に対して、その関連情報を表示するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで、設定ツリーの「スキーマ」ノードを選択します。次に、右側のパネルで「属性」タブを選択します。

   このタブには、スキーマ内のすべての標準属性 (読み取り専用) およびユーザー定義属性のテーブルが含まれています。テーブルの行の上にマウスを置くと、属性についての説明が表示されます。

   次の表は、属性テーブルのフィールドを示しています。

表 9-1 「属性」タブのテーブルの列

列の見出し	内容
名前	属性の名前。属性のタイプと呼ぶ場合もある

表 9-1 「属性」タブのテーブルの列 (続き)

列の見出し	内容
OID	属性のオブジェクト識別子。OID はスキーマオブジェクトを一意に識別する文字列で、通常は小数点で区切られた数値 OID や、企業の接頭辞の取得依頼については、IANA (Internet Assigned Number Authority) のアドレス `iana@iana.org` 宛てにメールを送るか、または IANA の Web サイト `http://www.iana.org/` を参照
構文	構文はこの属性値に使用できる形式を示す。属性の構文は、341 ページの表 9-2 に示す
複数値	この列のチェックボックスで、属性に複数の値を指定できるかどうかを指定する。複数値属性は、エントリ内に何回でも現れるが、単一値属性は 1 回しか現れない

表 9-2 属性の構文の定義

構文名	定義
Binary (以前は `bin`)	属性値がバイナリデータとして扱われることを示す
Boolean	属性値が `True` または `False` のどちらか一方であることを示す
Country String	属性値が、ISO 3166 に定められている 2 文字の国コード (たとえば `FR`) に限定されていることを示す
DN (以前は `dn`)	属性値が DN (識別名) であることを示す
DirectoryString (以前は `cis`)	属性値が UTF-8 方式で符号化された文字を含み、大文字と小文字が区別されないことを示す
GeneralizedTime	属性値が印刷可能な文字列として符号化されることを示す。タイムゾーンを指定する必要がある。GMT を使用するのが望ましい
IA5String (以前は `ces`)	属性値が ASCII 文字のサブセットだけを含み、大文字と小文字が区別されることを示す
INTEGER (以前は `int`)	有効な属性値が数字であることを示す
OctetString	Binary と同じ

表 9-2　属性の構文の定義 (続き)

構文名	定義
Postal Address	属性値が次のように符号化されていることを示す *dstring*[$ *dstring*]* 各 *dstring* コンポーネントは DirectoryString 構文の値と同様に符号化される。*dstring* 内の円記号とドル記号は、行区切り文字と間違えられることがないように、引用符で囲む。多くのサーバーで、postal address は最大 30 文字の 6 行に制限されている。次に例を示します。 `1234 Main St.$Anytown, CA 12345$USA`
TelephoneNumber (以前は tel)	属性値が電話番号の形式であることを示す。国際形式の電話番号を使用することを推奨する
URI	属性値が URL を含み、オプションとして http://、https://、ftp://、ldap://、ldaps:// などの接頭辞を持つことを示す。URI 値は、IA5String と同じように動作する。RFC 2396 (http://www.ietf.org/rfc/rfc2396.txt) を参照

属性の作成

ユーザー独自の属性定義をスキーマに追加するときは、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで、設定ツリーの「スキーマ」ノードを選択します。次に、右側のパネルで「属性」タブを選択します。
2. 「作成」をクリックして「属性の作成」ダイアログを表示します。
3. テキストフィールドに次の情報を指定して、新しい属性を定義します。必須項目は、属性名と構文だけです。
   - 属性名 : 属性を一意に識別する名前を入力します。属性タイプとも呼ばれます。属性名はアルファベットから始まる必要があり、ASCII 文字、数字、ハイフンだけが有効です。

注　属性名に大文字を使うこともできますが、LDAP クライアントでは区別されません。RFC 2251 (http://www.ietf.org/rfc/rfc2251.txt) のセクション 4.1.4 にも定められているように、属性名の大文字と小文字は区別されません。

- 属性の OID (オプション) : 属性のオブジェクト ID を入力します。OID については、340 ページの表 9-1 を参照してください。OID を指定しない場合、Directory Server は自動的に *attributeName*-oid を使用します。LDAP v3 に厳密に準拠するには、有効な数値 OID を指定する必要があります。
- 属性のエイリアス (オプション) : 属性の別名をコンマで区切って入力します。
- 属性の説明 (オプション) : 属性の目的を示す短い説明を入力します。
- 構文 : 属性に保持させるデータを記述するための構文を、ドロップダウンメニューから選択します。使用可能な構文については、341 ページの表 9-2 を参照してください。
- 複数値 : デフォルトでは、属性は複数の値をとります。属性がエントリごとに 1 つの値だけを持つように設定するときは、チェックボックスの選択を解除します。

4. 「属性の作成」ダイアログの「了解」をクリックして、新しい属性を定義します。この属性は、ユーザー定義の属性のテーブルに表示されます。

   ディレクトリエントリ内のこの属性に値を定義する前に、344 ページの「オブジェクトクラス定義の管理」で説明している手順に従って、その属性を必要とする、またはその属性を許可するオブジェクトクラスを作成または編集する必要があります。

属性の編集

編集できる属性は、ユーザー定義の属性だけで、コンソールを使う必要があります。属性の名前、構文、複数値の設定を変更する前に、ディレクトリ内のどのエントリも現在この属性を使っていないことを確認します。使われている属性を変更すると、クライアントはそのエントリにアクセスできなくなります。

属性のスキーマ定義を変更するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで、設定ツリーの「スキーマ」ノードを選択します。次に、右側のパネルで「属性」タブを選択します。
2. 「ユーザー定義属性」で編集する属性を選択し、「編集」をクリックします。
3. 「属性の編集」ダイアログのフィールドを修正し、属性を再定義します。

   属性の名前に基づく OID 文字列を使用している場合は、属性の名前を変更したら OID も必ず変更するようにしてください。OID については、340 ページの表 9-1 を参照してください。使用可能な構文については、341 ページの表 9-2 を参照してください。
4. 属性の編集が完了したら「了解」をクリックして変更を保存します。

属性の削除

削除できる属性は、ユーザー定義の属性だけで、コンソールを使う必要があります。属性の定義を削除する前に、ディレクトリ内のどのエントリも現在この属性を使っていないことを確認します。使われている属性を削除すると、クライアントはそのエントリにアクセスできなくなります。

属性のスキーマ定義を削除するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで、設定ツリーの「スキーマ」ノードを選択します。次に、右側のパネルで「属性」タブを選択します。
2. ユーザー定義属性のテーブルで属性を選択し、「削除」をクリックします。
3. 確認メッセージが表示されたら、削除を承認します。

   ただちに属性定義は削除されます。この処理を元に戻すことはできません。

オブジェクトクラス定義の管理

Directory Server コンソールでは、スキーマに含まれるすべてのオブジェクトクラスを表示し、ユーザー独自のオブジェクトクラス定義を作成、編集、削除できます。

オブジェクトクラスの表示

現在ディレクトリスキーマに定義されているすべてのオブジェクトクラスに関する情報を表示するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで、設定ツリーの「スキーマ」ノードを選択します。次に、右側のパネルで「オブジェクトクラス」タブを選択します。

   このタブには、スキーマ内のすべての標準 (読み取り専用) およびユーザー定義のオブジェクトクラスがリスト表示されます。
2. 表示するオブジェクトクラスをリストから選択します。

   このタブのその他のフィールドには、選択しているオブジェクトクラスに関する次の情報が表示されます。

表 9-3 「オブジェクトクラス」タブのフィールド

フィールド	内容
必須属性	このオブジェクトクラスを使用するエントリ内の必須属性のリスト。リストには継承された属性が含まれる

表 9-3　「オブジェクトクラス」タブのフィールド (続き)

フィールド	内容
許可された属性	このオブジェクトクラスを使用するエントリ内の許可された属性のリスト。リストには継承された属性が含まれる
親	親オブジェクトは、あるオブジェクトクラスの属性と構造の継承元であるオブジェクトクラスを識別する。オブジェクトクラスは、必要な属性と許可される属性を自動的に親オブジェクトクラスから継承する
OID	オブジェクトクラスのオブジェクト識別子。OID はスキーマオブジェクトを一意に識別する文字列で、通常は小数点で区切られた数値 OID や、企業の接頭辞の取得依頼については、IANA (Internet Assigned Number Authority) のアドレス `iana@iana.org` 宛てにメールを送るか、または IANA の Web サイト `http://www.iana.org/` を参照

オブジェクトクラスの作成

別のオブジェクトクラスから継承する複数のオブジェクトクラスを作成するときは、最初に親オブジェクトクラスを作成する必要があります。新しいオブジェクトクラスがカスタム属性を使用するときは、その属性も事前に定義しておく必要があります。

注　コンソールからは、構造化オブジェクトクラスだけを作成できます。これらのオブジェクトクラスは、親から継承する必要があります。auxiliary および abstract オブジェクトクラスを定義するには、コマンド行ユーティリティを使用します。

ユーザー独自のオブジェクトクラスの定義をスキーマに追加するときは、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで、設定ツリーの「スキーマ」ノードを選択します。次に、右側のパネルで「オブジェクトクラス」タブを選択します。
2. 「作成」をクリックして「オブジェクトクラスの作成」ダイアログを表示します。
3. テキストフィールドに次の情報を指定して、新しいオブジェクトクラスを定義します。
   - 名前 : オブジェクトクラスの一意の名前を入力します。

- 親 : 親となる既存のオブジェクトクラスを選択します。デフォルトでは、「トップ」が選択されます。他のオブジェクトクラスから継承しないオブジェクトクラスでは、これを選択する必要があります。親から継承される必要属性と許可される属性、および親は、対応するリストに表示されます。

  一般に、ユーザーエントリに対して属性を追加する場合、親オブジェクトは `inetOrgPerson` オブジェクトクラスになります。企業エントリに対して属性を追加する場合、親オブジェクトは通常 `organization` または `organizationalUnit` になります。グループエントリに対して属性を追加する場合、親オブジェクトは通常 `groupOfNames` または `groupOfUniqueNames` になります。

- OID (オプション) : 属性のオブジェクト ID を入力します。OID については、344 ページの表 9-3 を参照してください。OID を指定しない場合、Directory Server は自動的に *objectClassName*`-oid` を使用します。LDAP v3 に厳密に準拠するには、有効な数値 OID を指定する必要があります。

4. 新しいオブジェクトクラスを使用するエントリに含まれる属性を定義します。

   - 必須属性を定義するときは、「利用可能な属性」リストから属性 (1 つまたは複数) を選択し、「必須属性」ボックスの左にある「追加」ボタンをクリックします。

   - 許可された属性を定義するときは、「利用可能な属性」リストから属性 (1 つまたは複数) を選択し、「許可された属性」ボックスの左にある「追加」ボタンをクリックします。

   - すでに追加されている属性を削除するときは、いずれかのリストで属性を選択して強調表示し、対応する「削除」ボタンをクリックします。親オブジェクトクラスから継承された必須属性および許可された属性は、どちらも削除できません。

5. 「オブジェクトクラスの作成」ダイアログの「了解」をクリックして、新しいオブジェクトクラスを定義します。定義したオブジェクトクラスは、ユーザー定義のオブジェクトクラスのテーブルに表示され、これを使ってエントリを定義できるようになります。

オブジェクトクラスの編集

編集できるオブジェクトクラスは、ユーザー定義のオブジェクトクラスだけで、コンソールを使う必要があります。オブジェクトクラスの定義を変更する前に、ディレクトリ内のどのエントリも現在このオブジェクトクラスを使っていないことを確認します。使われている属性を変更すると、クライアントはそのエントリにアクセスできなくなります。

オブジェクトクラスのスキーマ定義を変更するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで、設定ツリーの「スキーマ」ノードを選択します。次に、右側のパネルで「オブジェクトクラス」タブを選択します。
2. 「ユーザー定義のオブジェクトクラス」リストから編集するオブジェクトクラスを選択し、「編集」をクリックします。
3. 「オブジェクトクラスの編集」ダイアログのフィールドを修正し、オブジェクトクラスを再定義します。

   オブジェクトクラスの名前や OID を変更することはできません。これらの情報を変更するには、そのオブジェクトクラスを削除し、新に作成する必要があります。

   - 親 : 親となる既存のオブジェクトクラスを選択します。親から継承される必要属性と許可される属性、および親は、対応するリストに表示されます。
   - 必須属性を定義するときは、「利用可能な属性」リストから属性 (1 つまたは複数) を選択し、「必須属性」ボックスの左にある「追加」ボタンをクリックします。
   - 許可された属性を定義するときは、「利用可能な属性」リストから属性 (1 つまたは複数) を選択し、「許可された属性」ボックスの左にある「追加」ボタンをクリックします。
   - すでに追加されている属性を削除するときは、いずれかのリストで属性を選択して強調表示し、対応する「削除」ボタンをクリックします。親オブジェクトクラスから継承された必須属性および許可された属性は、どちらも削除できません。
4. オブジェクトクラスの編集が完了したら「了解」をクリックして変更を保存します。

オブジェクトクラスの削除

削除できるオブジェクトクラスは、ユーザー定義のオブジェクトクラスだけで、コンソールを使う必要があります。オブジェクトクラスの定義を削除する前に、ディレクトリ内のどのエントリも現在このオブジェクトクラスを使っていないことを確認します。使われている属性を削除すると、クライアントはそのエントリにアクセスできなくなります。

オブジェクトクラスのスキーマ定義を削除するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで、設定ツリーの「スキーマ」ノードを選択します。次に、右側のパネルで「オブジェクトクラス」タブを選択します。
2. ユーザー定義のオブジェクトクラスのリストから削除するオブジェクトクラスを選択し、「削除」をクリックします。
3. 確認メッセージが表示されたら、削除を承認します。

   ただちにオブジェクトクラス定義が削除されます。この処理を元に戻すことはできません。

スキーマ定義のレプリケーション

2 つのサーバーの間で 1 つまたは複数のサフィックスのレプリケーションを設定するたびに、スキーマも自動的にレプリケートされます。これにより、コンシューマにレプリケートされる可能性のあるすべてのオブジェクトクラスと属性を定義する、完全で同一のスキーマがすべてのレプリカに提供されます。このため、マスターサーバーもマスタースキーマを持ちます。

すべてのレプリカにスキーマを適用するには、すべてのマスターでスキーマ検査を有効にする必要があります。スキーマは、LDAP 処理が行われるマスターで検査されるため、コンシューマの更新時は検査の必要はありません。パフォーマンスを向上させるために、レプリケーションメカニズムではコンシューマレプリカでのスキーマ検査を行いません。

注 ハブと専用コンシューマでは、スキーマ検査をオフにすべきではありません。コンシューマでは、スキーマ検査を行なってもパフォーマンスには影響しないので、レプリカの内容がスキーマに準拠していることを確認するため、常にオンにしておく必要があります。

マスターサーバーは、コンシューマの初期化時と、コンソールまたはコマンド行ツールを使ってスキーマを変更するたびに、スキーマを自動的にコンシューマにレプリケートします。デフォルトでは、スキーマ全体がレプリケートされ、コンシューマ側にまだ存在しないスキーマ要素があれば、コンシューマ側に作成され、99user.ldif ファイルに保存されます。

たとえば、マスターサーバーの起動時に 98mySchema.ldif ファイルにスキーマ定義が含まれ、その後でマスター、ハブ、専用コンシューマのいずれかのサーバーとのレプリケーションアグリーメントを定義したと仮定します。このマスターからレプリカを初期化すると、レプリケートされたスキーマには 98mySchema.ldif からの定義が含まれますが、レプリカサーバー側の 99user.ldif にもこの定義が格納されます。

コンシューマの初期化時にスキーマがレプリケートされた後で、マスター側の cn=schema でスキーマを変更すると、マスターはスキーマ全体をコンシューマにもレプリケートします。このように、コマンド行ユーティリティまたはコンソールからマスタースキーマに加えた変更は、コンシューマにレプリケートされます。これらの変更はマスター側の 99user.ldif に保存され、上記メカニズムによって、コンシューマ側の 99user.ldif にも格納されます。

レプリケートされたスキーマファイルの変更

レプリケーションメカニズムでは、スキーマを含む LDIF ファイルに直接加えた変更は検出されません。このため、338 ページの「スキーマファイルの変更」で説明した方法でスキーマを更新した場合は、マスターの再起動後もコンシューマにはレプリケートされません。

スキーマファイル内の変更をコンシューマに強制的に適用するために、Directory Server 5.2 には次のスクリプトが用意されています。

Windows プラットフォーム
```
cd ServerRoot
bin¥slapd¥admin¥bin¥perl slapd-serverID¥schema_push.pl
```

その他のインストール
```
# ServerRoot/slapd-serverID/schema_push.pl
```

マスターサーバー上でスキーマファイルを修正するときは、次の手順を実行します。

1. 次のスキーマディレクトリに新しいスキーマファイルを追加するか、既存のスキーマファイルを編集します。

   ServerRoot/slapd-*serverID*/config/schema

   このディレクトリに含まれるスキーマファイルには、インストール時に定義されたシステムユーザーだけが書き込み処理を実行できます。詳細は、338 ページの「スキーマファイルの変更」を参照してください。

2. 前述の適切なコマンドを指定した schema_push.pl スクリプトを実行します。このスクリプトは、実際にレプリカにスキーマを送信するわけではありません。このスクリプトを実行すると、スキーマファイルに特別な属性が書き込まれ、ロードとほぼ同時にスキーマファイルがレプリケートされます。
3. サーバーを再起動します。サーバーはすべてのスキーマファイルをロードし、レプリケーションメカニズムによって、新しいスキーマが各コンシューマにレプリケートされます。

スキーマレプリケーションの制限

デフォルトでは、レプリケーションメカニズムによってスキーマがレプリケートされるたびに、スキーマ全体がコンシューマに送信されます。次の 2 つの状況では、この処理は望ましくありません。

- コンソールまたはコマンド行から cn=schema に加える変更は、ユーザー定義のスキーマエレメントだけに対象が限定され、すべての標準スキーマは変更されません。スキーマを頻繁に変更する場合、未変更のスキーマ要素を含む大規模な要素セットを毎回送信することはパフォーマンスに影響します。ユーザー定義のスキーマ要素だけをレプリケートすることで、レプリケーションとサーバーのパフォーマンスを向上できます。
- Directory Server 5.2 のマスターから Directory Server 5.1 のコンシューマにレプリケートする場合、これらのバージョンの設定属性のスキーマに違いがあるため、競合が発生します。この場合、次の方法でユーザー定義のスキーマ要素だけをレプリケートする必要があります。

ユーザー定義のスキーマだけがレプリケートされるようにスキーマレプリケーションを制限するには、次のコマンドを使います。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=config
changetype: modify
replace: nsslapd-schema-repl-useronly
nsslapd-schema-repl-useronly: on
```

必要に応じてデフォルト値の off を使うことで、スキーマ全体をレプリケートできます。

第 10 章

インデックスの管理

書籍の索引と同様に、Directory Server のインデックスを利用することで、検索文字列とディレクトリの内容への参照を関連づけ、検索を速く行うことができます。インデックスは、独立したデータベースファイルに格納される属性値のテーブルです。インデックスの作成と管理は、ディレクトリ内のサフィックスごとに行われます。サフィックスの設定にインデックスを作成すると、サーバーは自動的にインデックスを管理します。

インデックスの基本概念、コストと利点、`nsslapd-allidsthreshold` 属性の説明、Directory Server のパフォーマンスを向上する方法については、『Sun ONE Directory Server インストールおよびチューニングガイド』の第 7 章「インデックスのチューニング」を参照してください。

この章は、次の節で構成されています。

- インデックスの概要
- インデックスの管理
- ブラウズインデックスの管理

インデックスの概要

インデックスは、サフィックスごとに対応するデータベースディレクトリ内のファイルに格納されます。各インデックスファイルには、指定の属性についてサフィックスに定義されているすべてのインデックスが含まれます。たとえば、共通名属性 (`cn`) で維持されているすべてのインデックスは、*databaseName*`_cn.db3` ファイルに格納されます。

インデックスファイルは、サフィックスを初期化したときに作成されます。また、この章で説明するコマンドを使って作成することもできます。クライアントによる検索操作および内部操作の最中に、サーバーはインデックスにアクセスし、ディレクトリ内のエントリをより速く検出します。修正操作を行うときは、ディレクトリはディレクトリの内容を更新し、インデックスファイルを更新してインデックスを最新の状態に維持する必要があります。

Directory Server でサポートされているインデックスのタイプは次のとおりです。

- 実在インデックス (pres) : このインデックスには、特定の属性を含むエントリのリストが含まれます。属性の値には依存しません。
- 等価インデックス (eq) : このインデックスでは、特定の属性値を含むエントリを効率的に検索できます。
- 近似インデックス (approx) : このインデックスでは、フィルタ用演算子 ~= を使って、似た音のエントリを効率的に検索できます。たとえば、近似インデックスを利用すると、名前の一部分やスペルの間違った名前でも検索できます。Directory Server は、Metaphone 音声アルゴリズムのバリエーションを使って、近似インデックスの検索を行います。

  注 Directory Server 5.2 の Metaphone 音声アルゴリズムでは、US-ASCII 文字だけがサポートされています。したがって、近似インデックスは英語の値だけで使ってください。

- 部分文字列インデックス (sub) : このインデックスでは、たとえば cn=*john* のように、属性値の部分文字列を効率的に検索できます。値ごとに多数の部分文字列があるため、このインデックスの維持にはコストがかかります。

  部分文字列インデックスとして、各エントリの 2 文字以上を指定する必要があります。

- マッチングルールインデックス : 地域に対応したマッチングルールの OID (照合順序) とインデックス登録される属性を関連づけることで、国際化ディレクトリの検索を速く行います。
- ブラウズインデックス : VLV (仮想リスト表示) 制御で実行される検索を高速化します。たとえば ou=People,dc=example,dc=com のように、各サブツリーの表示パフォーマンスを向上させるために、ディレクトリツリーのすべての分岐点でブラウズインデックスを作成できます。

システムインデックス

システムインデックスは、削除や修正ができないインデックスです。これは、Directory Server が正常かつ効率的に機能する上で必要なインデックスです。次の表は、すべてのサフィックスに自動的に作成されるシステムインデックスを示しています。

表 10-1 すべてのサフィックスのシステムインデックス

属性	等価	実在	目的
aci		X	ディレクトリサーバーが、ディレクトリに維持されているアクセス制御情報を速く取得できるようにする
entrydn	X		DN 検索に基づくエントリの取得を速くする
nsUniqueId	X		特定のエントリの検索に使われる
nscpEntryDN	X		レプリケーションのために Directory Server で内部的に使われる
nsds5ReplConflict	X	X	レプリケーションの競合の検出に利用される
numsubordinates		X	「ディレクトリ」タブの表示パフォーマンスを強化するために、Directory Server コンソールで使われる
objectClass	X		ディレクトリのサブツリー検索を速くするために使われる
parentID	X		1 レベル検索におけるディレクトリのパフォーマンスを強化する

デフォルトインデックス

ディレクトリに新しいサフィックスを作成すると、サーバーは対応するデータベースディレクトリにデフォルトのインデックスセットを設定します。インデックス作成の要件に応じて、デフォルトインデックスを変更できますが、インデックスの設定を変更する前に、企業内のサーバープラグインやその他のサーバーが、インデックス生成されている属性に依存していないことを確認する必要があります。

サフィックスの新規作成時に使われるデフォルトのインデックスセットを変更する方法については、365 ページの「デフォルトのインデックスセットの変更」を参照してください。

次の表は、Directory Server の事前に設定されているデフォルトインデックスを示しています。

表 10-2 すべての新規サフィックスのデフォルトインデックス

属性	等価	実在	部分文字列	目的
cn	X	X	X	もっとも一般的なタイプのユーザーディレクトリ検索のパフォーマンスを向上させる
givenName	X	X	X	もっとも一般的なタイプのユーザーディレクトリ検索のパフォーマンスを向上させる
mail	X	X	X	もっとも一般的なタイプのユーザーディレクトリ検索のパフォーマンスを向上させる
mailAlternateAddress	X			Sun ONE Messaging Server で使われる
mailHost	X			Sun ONE Messaging Server で使われる
member	X			Sun ONE サーバーのパフォーマンスを向上させる。このインデックスは、参照整合性検査プラグインでも使われる。詳細は、81 ページの「参照整合性の管理」を参照
nsCalXItemId	X	X	X	Sun ONE Calendar Server で使われる
nsLIProfileName	X			Sun ONE Messaging Server のローミング機能で使われる
nsRoleDN	X			ロールベースの操作のパフォーマンスを向上させる
nswcalCALID	X			Sun ONE Calendar Server で使われる

表 10-2　すべての新規サフィックスのデフォルトインデックス (続き)

属性	等価	実在	部分文字列	目的
owner	X			Sun ONE サーバーのパフォーマンスを向上させる。このインデックスは、参照整合性検査プラグインでも使われる。詳細は、『Sun ONE Directory Server Administration Guide』を参照
pipstatus	X			Sun ONE Servers で使われる
pipuid		X		Sun ONE Servers で使われる
seeAlso	X			Sun ONE サーバーのパフォーマンスを向上させる。このインデックスは、参照整合性検査プラグインでも使われる。詳細は、81 ページの「参照整合性の管理」を参照
sn	X	X	X	もっとも一般的なタイプのユーザーディレクトリ検索のパフォーマンスを向上させる
telephoneNumber	X	X	X	もっとも一般的なタイプのユーザーディレクトリ検索のパフォーマンスを向上させる
uid	X			Sun ONE サーバーのパフォーマンスを向上させる。
uniquemember	X			Sun ONE サーバーのパフォーマンスを向上させる。このインデックスは、参照整合性検査プラグインでも使われる。詳細は、81 ページの「参照整合性の管理」を参照

データベース内の標準インデックスファイル

デフォルトインデックスとその他の内部インデックス作成メカニズムを維持する必要があるため、Directory Server では、いくつかの標準インデックスファイルも維持します。デフォルトで提供される標準インデックスは次のとおりです。ユーザーがこれらのファイルを作成する必要はありません。

- *databaseName*_id2entry.db3：ディレクトリエントリの実際のデータベースが含まれる。その他のすべてのデータベースファイルは、このファイルから作成し直すことができる
- *databaseName*_id2children.db3：1 レベル検索、つまりあるエントリのすぐ下の子だけを調べるように、検索の範囲を制限する
- *databaseName*_dn.db3：サブツリー検索、つまりあるエントリと、そのエントリの下にあるサブツリーのすべてのエントリを調べるように、検索の範囲を制御する
- *databaseName*_dn2id.db3：エントリの識別名を ID 番号に割り当てることにより、すべての検索を効果的に開始する

属性名のクイックリファレンス

次の表に、基本名 (実際の名前) とエイリアス名の両方を持つ属性のリストを示します。インデックスを作成する場合は、必ず基本名を使ってください。

表 10-3 属性の基本名とエイリアス

属性の基本名	属性のエイリアス名
authorCn	documentAuthorCommonName
authorSn	documentAuthorSurname
c	countryName
cn	commonName
co	friendlyCountryName
dc	domainComponent
dn	distinguishedName
drink	favoriteDrink
facsimileTelephoneNumber	fax
l	localityName
labeledUri	labeledUrl
mail	rfc822mailbox

表 10-3　属性の基本名とエイリアス

属性の基本名	属性のエイリアス名
mobile	mobileTelephoneNumber
o	organizationName
ou	organizationalUnitName
pager	pagerTelephoneNumber
sn	surname
st	stateOrProvinceName
street	streetAddress
ttl	timeToLive
uid	userId

インデックスの管理

ここでは、Directory Server コンソールとコマンド行を使って、特定の属性の実在インデックス、等価インデックス、近似インデックス、部分文字列インデックス、および国際化インデックスを作成および削除する方法を説明します。VLV (仮想リスト表示) 操作の前に必要な手順については、366 ページの「ブラウズインデックスの管理」を参照してください。

注　インデックスはサフィックスごとに異なるため、すべてのサフィックスの設定に新しいインデックスを作成する必要があります。

コンソールからサフィックスを新規作成するときに、既存のサフィックスのインデックス設定のクローンを作成するオプションを選択できます。

新しいインデックスを作成する前に、インデックスを維持する利点とコストのバランスを検証します。次の点に注意してください。

- 電話番号のように、一般に数字が含まれる属性については、近似インデックスは効果的ではないので使わないこと
- バイナリ属性については、部分文字列インデックスは機能しない。たとえば、jpegPhoto などのバイナリデータを格納するための属性などのように、大きい値には等価インデックスを使わないこと

- インデックスの維持にはリソースが必要なので、検索対象となることの多い属性についてだけインデックスを作成すること。サーバーでエントリを作成するには、インデックスの付けられた属性をすべて調べたり、新しいエントリに含まれる各属性について新規エントリを生成したりする必要があるため、CPU 時間が必要となる
- 各インデックスファイルのサイズは、ディレクトリの内容に比例する
- インデックスが付いていない属性も検索要求で指定できるが、検索のタイプによっては、検索のパフォーマンスが低下する

コンソールからのインデックスの管理

多くの属性でインデックスを変更または追加するときは、まず、サフィックスを読み取り専用に変更し、その内容を LDIF にエクスポートします。LDIF ファイルからサフィックスを再初期化することで、サフィックスのインデックスの再生成が速くなります。

1. Directory Server コンソールの最上位の「設定」タブで「データ」ノードを展開し、インデックスを生成するサフィックスを選択します。右側のパネルで「インデックス」タブを選択します。

   システムインデックスのテーブルは変更できません。「追加インデックス」テーブルの属性でインデックスを追加、変更、または削除します。

2. インデックスが生成されていない属性のインデックスを追加するときは、「属性の追加」ボタンをクリックします。ダイアログが表示されるので、インデックスを生成する 1 つまたは複数の属性を選択し、「了解」をクリックします。

   「追加インデックス」テーブルに新しい属性が表示されます。

3. 属性のインデックスを変更するときは、「追加インデックス」テーブルで、その属性で維持するインデックスのタイプのチェックボックスを選択または選択解除します。

4. 英語以外の言語の値を含む属性のインデックスを作成する場合は、「マッチングルール」フィールドで使う照合順序の OID を入力します。

   複数の OID をスペースではなくコンマで区切って指定することにより、属性に複数の言語を使ったインデックスを付けることができます。サポートされているロケールと、それに関連する照合順序の OID のリストについては、『Sun ONE Directory Server Reference Manual』の付録 C 「Directory Internationalization」を参照してください。

5. 属性のすべてのインデックスを削除するときは、テーブルでその行を選択し、「属性の削除」ボタンをクリックします。

6. 「保存」をクリックして、新しいインデックス設定を保存します。

   属性のすべてのインデックスを削除すると、サーバーはその属性のインデックスファイルを削除し、設定が完了します。属性のインデックスを変更または追加した場合は、次の手順に進みます。

7. 新しいインデックスを利用するには、データベースファイルの更新が必要であることを示す警告ダイアログが表示されます。サフィックスのインデックスの再生成を行うか、サフィックスを再初期化できます。

- 1 つまたは 2 つのインデックスを追加または変更した場合、またはサフィックスを利用不可にできない場合は、サフィックスのインデックスの再生成を行います。「サフィックスを再インデックス」ボタンをクリックし、インデックスを再生成するためのダイアログを表示します。デフォルトでは、変更した属性、またはインデックス設定に追加した属性が選択されています。「了解」をクリックして、これらの属性のインデックスの再生成を開始します。数百万のエントリを持つディレクトリで多数の属性のインデックスの再生成を行うには、数時間が必要ですが、インデックスの再生成の最中も、サフィックスは常にオンラインで維持されます。
- いくつかの属性でインデックスを追加または変更した場合、このサフィックスからエクスポートした最新の LDIF ファイルが用意されていれば、「サフィックスを初期化」ボタンをクリックします。「サフィックスを初期化」ボタンが表示されるので、LDIF ファイルの名前を入力するか、名前とパスを表示して選択し、「了解」をクリックします。サーバーは LDIF ファイルに基づいてサフィックスを再初期化し、新しい設定ですべてのインデックスを作成します。ディレクトリのサイズにもよりますが、サフィックスの再初期化は通常は 2 つ以上の属性のインデックスの再生成よりも速く行われます。ただし、初期化中はサフィックスを利用できません。
- サフィックスのインデックスの再生成または再初期化を行わない場合、すべてのデータを続けて利用することはできますが、新しいインデックスは作成されず、ディレクトリのアクセスパフォーマンスは向上しません。

サフィックスのインデックスの再生成または再初期化を行うと、追加されたすべての新規データおよびディレクトリ内の既存データに対して、新しいインデックスが直ちに有効になります。サーバーを再起動する必要はありません。

コマンド行からのインデックスの管理

コマンド行からのインデックスの作成または変更は、次の 2 つの手順で行われます。

- `ldapmodify` コマンド行ユーティリティを使用して、インデックス設定エントリを追加または変更します。インデックスはサフィックスごとに設定され、対応するデータベース設定にインデックス設定エントリが格納されます。

- db2index.pl Perl スクリプト (Solaris パッケージでは directoryserver db2index-task) を実行して、サーバーに保持される新しいインデックスのセットを生成します。

警告

システムインデックスを削除すると、Directory Server のパフォーマンスに重大な影響を及ぼすため、このインデックスは削除しないでください。システムインデックスは、cn=index,cn=*databaseName*,cn=ldbm database,cn=plugins,cn=config エントリと、cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config エントリの下にあります。

デフォルトインデックスを削除すると、Directory Server の動作にも影響を及ぼすので、慎重に行なってください。

インデックス設定エントリの作成

インデックスが設定されていない属性のインデックスを作成するには、対応するデータベースの設定内にその属性の新しいエントリを作成する必要があります。

インデックス設定エントリの DN は次のとおりです。

```
cn=attributeName,cn=index,cn=databaseName,cn=ldbm database,
 cn=plugins,cn=config
```

ここで、*databaseName* はインデックスを作成するサフィックスに対応したデータベースの名前です。たとえば次のコマンドは、フランス語の値で sn (姓) 属性の実在、等価、部分文字列、近似インデックスを作成します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=sn,cn=index,cn=databaseName,cn=ldbm database,
 cn=plugins,cn=config
objectClass: top
objectClass: nsIndex
cn: sn
nsSystemIndex: false
nsIndexType: pres
nsIndexType: eq
nsIndexType: sub
nsIndexType: approx
nsMatchingRule: 1.3.6.1.4.1.42.2.27.9.4.76.1
```

インデックス設定エントリは nsIndex オブジェクトクラスを持ち、nsSystemIndex 属性が必ず存在して、その値が false である必要があります。新しいシステムインデックスを作成することはできません。システムインデックスは、Directory Server によって内部的に定義されているものだけが保持されます。

nsIndexType 属性の値には、指定の属性で維持されるインデックスのリストが含まれます。対応するインデックスを定義するには、前述のいずれかの値を使います。

単一の値 none を使ってその属性のインデックスを明示的に無効にすることもできるので、たとえば、属性のインデックスを一時的に無効にできます。インデックス設定エントリに nsIndexType 属性を含めない場合、すべてのインデックスがデフォルトで維持されます。

オプションの nsMatchingRule 属性には、国際化インデックスの言語照合順序の OID が含まれます。サポートされているロケールと、それに関連する照合順序の OID のリストについては、『Sun ONE Directory Server Reference Manual』の付録 C 「Directory Internationalization」を参照してください。

インデックス設定属性の詳細については、『Sun ONE Directory Server Reference Manual』の第 5 章にある「Default Index Attributes」を参照してください。

注 インデックスを作成する場合は、属性のエイリアスではなく、基本名を常に使う必要があります。属性の基本名は、スキーマでその属性に一覧表示された最初の名前です。たとえば、userid 属性では uid が基本名になります。属性の基本名とエイリアス名のリストについては、356 ページの表 10-3 を参照してください。

インデックス設定エントリの変更

属性にすでに定義されているインデックスの設定を変更するには、対応するインデックスエントリを修正します。たとえば、前述の例で定義した sn インデックス設定に対して次のコマンドを実行すると、近似インデックスが削除され、言語がカナダ系のフランス語 (Canadian French) に変更されます。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=sn,cn=index,cn=databaseName,cn=ldbm database,
 cn=plugins,cn=config
changetype: modify
delete: nsIndexType
nsIndexType: approx
-
replace: nsMatchingRule
nsMatchingRule: 1.3.6.1.4.1.42.2.27.9.4.78.1
^D
```

db2index.pl スクリプトの実行

インデックスエントリを作成する、既存のインデックスエントリにインデックスタイプを追加するか、照合順序を変更する処理が完了したら、`db2index.pl` スクリプト (Solaris パッケージでは `directoryserver db2index-task`) を実行して、新しいインデックスを生成します。このスクリプトはサフィックスの内容を読み取り、設定エントリに基づいて指定の属性のインデックスの再生成を行います。

このコマンドの実行中もサーバーを通じてサフィックスの内容を利用できますが、スクリプトの実行が完了するまで検索のインデックス対応は行われません。インデックスの再生成は多くのリソースを消費するタスクであるため、サーバー上のその他の処理のパフォーマンスに影響を生じることがあります。ディレクトリのサイズにもよりますが、サフィックスの再初期化は通常は 2 つ以上の属性のインデックスの再生成より速く行われます。ただし、初期化中はサフィックスを利用できません。詳細は、364 ページの「サフィックスの再初期化」を参照してください。

このスクリプトのコマンドは、プラットフォームごとに異なります。

Solaris パッケージ
```
# /usr/sbin/directoryserver db2index-task
```
Windows プラットフォーム
```
cd ServerRoot
bin¥slapd¥admin¥bin¥perl slapd-serverID¥db2index.pl
```
その他のインストール
```
# ServerRoot/slapd-serverID/db2index.pl
```

次の例は、*databaseName* に対応するサフィックスに *sn* インデックスを再生成します。

UNIX シェルスクリプト

```
# Solaris パッケージ内の Directory Server db2index タスクを使用します
/var/Sun/mps/slapd-example/db2index.pl ¥
  -D "cn=Directory Manager" -w password -n databaseName -t sn
```

Windows バッチファイル

```
C:¥Program Files¥Sun¥MPS¥bin¥slapd¥admin¥bin¥perl.exe
  C:¥Program Files¥Sun¥MPS¥slapd-example¥db2index.pl
    -D "cn=Directory Manager" -w password -n databaseName -t sn
```

詳細については、『Sun ONE Directory Server Reference Manual』の第 2 章にある「`db2index.pl`」を参照してください。

属性のすべてのインデックスの削除

属性に設定されているすべてのインデックスを削除するときは、設定エントリとデータベースファイルを削除します。たとえば次のコマンドは、*databaseName* というデータベースに含まれる `sn` 属性のすべてのインデックスの設定を解除します。

```
ldapdelete -h host -p port -D "cn=Directory Manager" -w password ¥
"cn=sn,cn=index,cn=databaseName,cn=ldbm database,cn=plugins, ¥
 cn=config"
```

このエントリを削除すると、sn 属性のインデックスは *databaseName* データベースに対応するサフィックスで維持されなくなります。ディスクスペースを節約するために、サーバーが使わなくなったインデックスファイルを削除することもできます。この例では、次のファイルを削除できます。

ServerRoot/slapd-*serverID*/db/*databaseName*/*databaseName*_sn.db3

サフィックスのインデックスの再生成

インデックスファイルが破損した場合は、サフィックスのインデックスの再生成を行なって、対応するデータベースディレクトリにインデックスファイルを作成し直す必要があります。Directory Server コンソールを使ってサフィックスのインデックスの再生成を行うには、インデックスの再生成と再初期化の 2 つの方法があります。

サフィックスのインデックスの再生成

サフィックスのインデックスの再生成を行うと、サーバーはサフィックスに含まれるすべてのエントリを調べ、インデックスファイルを再作成します。インデックスの再生成時は、サフィックスの内容に対して読み取りと書き込みを実行できます。ただし、インデックスを再生成するすべての属性についてサフィックス全体をサーバーがスキャンするため、設定するインデックスによっては、数百万のエントリを持つサフィックスでは完了までに最大で数時間かかります。また、インデックスの再生成中はインデックスを利用できず、サーバーのパフォーマンスに影響が生じることもあります。

コンソールからサフィックスのインデックスの再生成を行うには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで「データ」ノードを展開し、インデックスを再生成するサフィックスを表示します。
2. このサフィックスの設定ノードをマウスの右ボタンでクリックし、ポップアップメニューから「再インデックス」を選択します。あるいは、ノードをマウスの左ボタンでクリックして選択し、「オブジェクト」メニューから「再インデックス」を選択します。

   選択しているサフィックスにインデックスが設定されているすべての属性をリスト表示した「サフィックスを再インデックス」ダイアログが表示されます。
3. インデックスを再生成する属性の隣にあるチェックボックスを選択します。選択には、「すべてをチェック」、「チェックしない」ボタンが役立ちます。属性のすべてのインデックスが同じデータベースファイルに格納されているので、すべてのインデックスを同時に再生成する必要があります。
4. 「了解」をクリックします。インデックスの再生成の処理中に予期せぬ検索結果が得られたり、パフォーマンスに影響が生じたりする可能性についてコンソールに確認メッセージが表示されます。

5. 「はい」をクリックしてインデックスの再生成を開始します。

   インデックスの再生成に関するメッセージはダイアログに表示されます。処理が完了したら、ダイアログを閉じます。

コマンド行からサフィックスのインデックスの再生成を行うには、362 ページの「db2index.pl スクリプトの実行」で説明している手順に従って、インデックスファイルを再作成するすべての属性を指定します。

サフィックスの再初期化

サフィックスを再初期化すると、新しい内容がインポートされてサフィックスの内容が置き換えられ、新しいインデックスファイルが作成されます。サフィックスの再初期化は、エントリのロード時にすべての属性が 1 つのパスでインデックスが作成されるので、通常は複数の属性のインデックスの再生成よりも速く行われます。ただし、再初期化中はサフィックスを利用できません。

Directory Server コンソールまたはコマンド行から次のすべての手順を実行できます。

1. 98 ページの「アクセス権とリフェラルの設定」で説明している方法で、サフィックスを読み取り専用に設定します。内容をエクスポートした後に変更が加えられないように、最初にサフィックスを書き込み不可に設定する必要があります。
2. 142 ページの「コンソールを使用した LDIF への単一サフィックスのエクスポート」で説明している方法で、サフィックス全体を LDIF ファイルにエクスポートします。
3. 137 ページの「サフィックスの初期化」で説明している方法で、同じ LDIF ファイルをインポートしてサフィックスを再初期化します。

   初期化中は、サフィックスを利用することはできません。初期化が完了すると、設定されたすべてのインデックスを利用できるようになります。
4. 98 ページの「アクセス権とリフェラルの設定」で説明している方法で、サフィックスを書き込み可能に戻します。

デフォルトのインデックスセットの変更

新規サフィックスの作成時に使われるデフォルトのインデックスセットは、次のエントリの下に定義されます。

```
cn=default indexes,cn=config,cn=ldbm database,
 cn=plugins,cn=config
```

コンソールまたはコマンド行からサフィックスを作成するたびに、対応するデータベースの初期インデックス設定としてデフォルトのインデックス定義のエントリがコピーされます。

デフォルトのインデックスセットの設定は、コマンド行ユーティリティだけを使って行われます。デフォルトのインデックスエントリの構文は、359 ページの「コマンド行からのインデックスの管理」で説明したインデックス設定エントリの構文とまったく同じです。たとえば、デフォルトのインデックス設定エントリを追加するには、次の `ldapmodify` コマンドを実行します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=drink,cn=default indexes,cn=config,cn=ldbm database,
 cn=plugins,cn=config
objectClass: top
objectClass: nsIndex
cn: drink
nsSystemIndex: false
nsIndexType: eq
nsIndexType: sub
nsMatchingRule: 1.3.6.1.4.1.42.2.27.9.4.76.1
```

このエントリを追加すると、すべての新しいサフィックスの `drink` 属性の値には、等価検索と部分文字列の検索がフランス語でインデックス設定されます。

デフォルトのインデックスエントリを変更または削除するには、`ldapmodify` コマンドまたは `ldapdelete` コマンドを使って `cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config` 内のインデックスセットを編集します。

ブラウズインデックスの管理

ブラウズインデックスは、サーバー側のソートまたは VLV (仮想リスト表示) 結果を要求する検索操作のための特別なインデックスです。ブラウズインデックスを利用することで、多数の検索結果をサーバー側でソートするように要求する検索のパフォーマンスが向上します。ディレクトリの設定によっては、ブラウズインデックスが定義されていない場合にソートを要求する検索の実行をサーバーが拒否することもあります。これにより、大規模なソート処理によってサーバーリソースがオーバロードすることを防止できます。

ブラウズインデックスは、検索のベースとなるエントリに適用され、ソート要求で使用する検索フィルタごとに専用のインデックスを作成する必要があります。たとえば、クライアントアプリケーションがすべてのユーザーのソートされたリストを頻繁に要求する場合は、クライアントが使用するフィルタ文字列用に `ou=People` のブラウズインデックスを作成します。

その他のインデックスと同様に、ブラウズインデックスの維持に必要な更新処理の最中は、パフォーマンスに影響が生じます。ブラウズインデックスの導入は慎重に計画し、テストする必要があります。

コンソール用のブラウズインデックス

Directory Server コンソールは、パネルの内容を更新するためにディレクトリ全体を対象に検索処理を頻繁に行います。32 ページの「ディレクトリツリーの表示オプション」で説明した方法で、ディレクトリツリーに表示されるエントリをソートするようにコンソールを設定したときは、コンソール用のブラウズインデックスを作成する必要があります。

コンソール用のブラウズインデックスは、コンソールが実行する検索に特化されています。これは、コンソールから作成できます。コンソール用のブラウズインデックスを作成するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「ディレクトリ」タブで、たとえば数千のユーザーエントリを持つ `ou=People,dc=example,dc=com` のように、ソートが必要な大規模なサブツリーの親を表示します。
2. この親エントリをマウスの右ボタンでクリックし、ポップアップメニューから「ブラウズインデックスの作成」を選択します。あるいは、エントリをマウスの左ボタンでクリックして選択し、「オブジェクト」メニューから「ブラウズインデックスの作成」を選択します。

   「ブラウズインデックスの作成」ダイアログボックスが開き、インデックス作成の状態が表示されます。コンソールは後述するブラウズインデックス設定エントリを作成し、インデックスファイルの内容を生成します。

3. 「閉じる」をクリックして、「ブラウズインデックスの作成」ダイアログボックスを閉じます。

   新しいインデックスは、コンソールの更新処理に直ちに適用され、ディレクトリに追加する新しいデータにも対応して維持されます。サーバーを再起動する必要はありません。

コンソール用のブラウズインデックスの設定は、次のエントリから構成されます。`vlvSearch` エントリは、インデックスが設定される検索のベース、対象、フィルタを定義します。`vlvIndex` エントリの `vlvSort` 属性は、「ディレクトリ」タブでソート可能な属性をソートされた順序で示します。

```
dn: cn=MCC entryDN,cn=databaseName,cn=ldbm database,
 cn=plugins,cn=config
objectClass: top
objectClass: vlvSearch
cn: MCC entryDN
vlvBase: "entryDN"
vlvScope: 1
vlvFilter: (|(objectclass=*)(objectclass=ldapsubentry))

dn: cn=by MCC entryDN, cn=MCC entryDN,cn=databaseName,
 cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvindex
cn: by MCC entryDN
vlvSort: cn givenname o ou sn uid
```

Directory Server コンソール用のブラウズインデックスを削除するには、次の手順を実行します。

1. Directory Server コンソールの最上位レベルにある「ディレクトリ」タブでディレクトリツリーを表示し、ブラウズインデックスを作成したエントリを探します。
2. このエントリをマウスの右ボタンでクリックし、ポップアップメニューから「ブラウズインデックスの削除」を選択します。あるいは、エントリをマウスの左ボタンでクリックして選択し、「オブジェクト」メニューから「ブラウズインデックスの削除」を選択します。このメニュー項目は、コンソール用のブラウズインデックスのエントリを選択している場合にだけ選択できます。
3. 「ブラウズインデックスの削除」ダイアログボックスが表示され、インデックスを削除するかどうかの確認が求められます。「はい」をクリックして、ブラウズインデックスを削除します。

クライアント検索用のブラウズインデックス

クライアント検索結果をソートするカスタムブラウズインデックスは、手動で定義する必要があります。コマンド行を使ったブラウズインデックス、または仮想リスト表示 (VLV) の作成は、次の 2 つの手順で行われます。

- `ldapmodify` ユーティリティまたは Directory Server コンソールの「ディレクトリ」タブを使って、新しいブラウズインデックスエントリを追加するか、既存のブラウズインデックスエントリを編集します。
- `vlvindex` スクリプト (Solaris パッケージ内の `directoryserver vlvindex`) を実行して、サーバーに保持される新しいブラウズインデックスのセットを生成します。

ブラウズインデックスエントリの指定

ブラウズインデックスは、特定のベースエントリとサブツリーに対して指定された検索ごとに異なります。ブラウズインデックスの設定は、エントリを含むサフィックスのデータベース設定に定義されます。

注	連鎖サフィックスにブラウズインデックスを作成することはできません。ローカルのサフィックスとサブサフィックスだけに作成できます。

ブラウズインデックスは、2 つのエントリによって設定されます。一つは `vlvSearch` オブジェクトクラスを使うエントリで、検索結果にインデックスを作成する検索のベース、対象、フィルタを指定します。もう一つのエントリは最初のエントリの子エントリで、`vlvIndex` オブジェクトクラスを使ってソートする属性とソート順序を指定します。

次の例は、`ldapmodify` ユーティリティを使用して、2 つのブラウズインデックス設定エントリを作成します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=Browsing ou=People, cn=databaseName,
 cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvSearch
cn: Browsing ou=People
vlvbase: ou=People,dc=example,dc=com
vlvscope: 1
vlvfilter: (objectclass=inetOrgPerson)

dn: cn=Sort rev employeenumber, cn=Browsing ou=People,
 cn=databaseName,cn=ldbm database,cn=plugins,cn=config
objectClass: top
```

```
objectClass: vlvindex
cn: Sort rev employeenumber
vlvSort: -employeenumber
^D
```

vlvscope は、0 (ベースエントリだけの場合)、1 (ベースのすぐ下の子の場合)、または 2 (ベースをルートとするサブツリー全体) のいずれかです。vlvfilter は、クライアント検索操作で使われる LDAP フィルタと同じフィルタです。すべてのブラウズインデックスエントリは同じ場所に配置されるため、cn の値にはブラウズインデックスの名前を指定しておくことをお勧めします。

vlvSearch エントリは、それぞれが少なくとも 1 つの vlvIndex エントリを持つ必要があります。vlvSort 属性は、ソートする属性とソート順序を定義する属性名のリストです。属性名の前につけられたダッシュ (-) は、順序を逆にすることを意味します。複数の vlvIndex エントリを定義することで、検索に複数のインデックスを定義できます。前述の例では、次のエントリを追加できます。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
dn: cn=Sort sn givenname uid, cn=Browsing ou=People,
 cn=databaseName,cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvindex
cn: Sort sn givenname uid
vlvSort: sn givenname uid
^D
```

ブラウズインデックスの設定を変更するには、対応する vlvSearch エントリまたは vlvIndex エントリを編集します。ブラウズインデックスを削除してサーバーがそれを維持しないようにするには、それぞれの vlvIndex エントリを削除するか、インデックスが 1 つしかない場合は、vlvSearch エントリと vlvIndex エントリの両方を削除します。vlvIndex エントリを削除すると、たとえば次のような対応するデータベース ファイルも削除できるようになります。

ServerRoot/slapd-*serverID*/db/*dbName*/*dbName*_vlv#Sortsngivennameuid.db3

vlvindex コマンドの実行

ブラウズインデックスのエントリの作成、または既存エントリの変更が完了したら、vlvindex コマンド (Solaris パッケージでは directoryserver vlvindex) を実行して、新しいブラウズインデックスセットを生成する必要があります。このコマンドは、ディレクトリの内容をスキャンし、ブラウズインデックス用のデータベースファイルを作成します。

ブラウズインデックスを生成するには、次のコマンドを使います。

Solaris パッケージ
```
# /usr/sbin/directoryserver vlvindex
```
その他のインストール
```
# installDir/slapd-serverID/vlvindex
```

次の例は、前の項で定義したブラウズインデックスを生成します。

```
# vlvindex -n databaseName -T "Browsing ou=People"
```

表 10-4 例で使った vlvindex オプションの説明

オプション	内容
-n	インデックスを作成するエントリを含んだ、データベースの名前を指定する
-T	対応するブラウズインデックスの vlvSearch エントリのネーミング属性の値を指定する。指定した vlvSearch エントリの vlvIndex エントリに対応するすべてのインデックスが生成される

詳細については、『Sun ONE Directory Server Reference Manual』の第 2 章にある「vlvindex」を参照してください。

第 11 章

セキュリティの実装

Sun ONE Directory Server には、ネットワーク上でセキュリティ保護され、かつ信頼できる通信を可能にするためのメカニズムが用意されています。LDAPS は標準の LDAP プロトコルのうち、データを暗号化したり認証のために証明書を使用したりするために SSL (Secure Sockets Layer) 上で実行されるものです。

Sun ONE Directory Server は Start TLS (Start Transport Layer Security) 拡張処理もサポートしてるため、元は暗号化されていない LDAP 接続でも TLS を有効にすることができます。Directory Server 5.2 からは、Windows プラットフォームと UNIX プラットフォームの両方で StartTLS がサポートされています。

Directory Server 5.2 では、SASL (Simple Authentication and Security Layer) を介した GSSAPI (Generic Security Services API) にも対応するようになりました。これにより、Solaris 環境で Kerberos Version 5 セキュリティプロトコルを利用できます。ID マッピングメカニズムによって、Kerberos 主体とディレクトリ内の ID が関連づけられます。

この章は、次の節で構成されています。

- Directory Server への SSL の導入
- SSL を有効化する手順の概要
- サーバー証明書の入手とインストール
- SSL の有効化
- クライアント認証の設定
- ID マッピング
- LDAP クライアントでセキュリティを使用するための設定

Directory Server への SSL の導入

SSL (Secure Sockets Layer) は暗号化された通信と、オプションとして Directory Server とクライアントの間の認証を提供します。LDAP プロトコルと DSML-over-HTTP プロトコルの両方で SSL を有効化し、サーバーとのすべての接続にセキュリティを提供することができます。また、レプリケーションと連鎖サフィックスのメカニズムが SSL を利用するように設定して、サーバー間の通信をセキュリティ保護することもできます。

簡易認証 (バインド DN とパスワード) の SSL を使用することで、サーバーとやり取りされるすべてのデータが暗号化され、データの機密と整合性が保証されます。必要に応じて、クライアントは証明書を使用して Directory Server への接続を認証したり、SASL (Simple Authentication and Security Layer) を利用したサードパーティ製のセキュリティメカニズムへの接続を認証できます。証明書ベースの認証では、公開鍵暗号方式を使ってクライアントまたはサーバーを偽装したり、認証されているユーザーになりすますことはできなくなります。

Directory Server では、SSL による通信と SSL を使用しない通信を別々のポートで同時に実行できます。また、セキュリティのためにすべての通信をセキュリティ保護されたポートに限定することもできます。クライアント認証を設定して、指定したセキュリティレベルを必須にするか、単にそのレベルを許可するかを定義できます。

SSL を有効にすることで、通常の LDAP 接続をセキュリティ保護する Start TLS 拡張処理への対応も有効になります。クライアントが SSL 以外のポートにバインドされても、TLS (Transport Layer Security) プロトコルを使って SSL 接続を開始できます。Start TLS 処理では、クライアントに一層の柔軟性が与えられ、ポートの割り当ても簡素化されます。

SSL が提供する暗号化メカニズムは、属性の暗号化にも使用されます。SSL を有効にすることで、サフィックスでの属性の暗号化を設定し、ディレクトリに格納するときにデータを保護することができます。詳細は、77 ページの「属性値の暗号化」を参照してください。

これ以外のセキュリティとして、クライアント側の SSL と証明書の使用状況に応じてディレクトリの内容へのアクセス制御を設定できます。特定の認証メソッドを必要とする ACI (アクセス制御命令) を定義することで、セキュリティ保護されたチャンネルだけを通じてデータを転送することができます。詳細は、199 ページの「バインドルール」を参照してください。

管理サーバーに SSL を設定する方法など、SSL、インターネットセキュリティ、証明書の詳細な説明については、『Sun ONE Server Console Server Management Guide』の第 10 章「Using SSL and TLS with Sun ONE Servers」を参照してください。

SSL を有効化する手順の概要

この章の以後の各項では、次の各手順について説明します。

1. Directory Server で使用する証明書を入手してインストールし、証明機関の証明書 (CA) を信頼するように Directory Server を設定します。この処理には、次の手順が含まれます。
   a. 必要に応じて証明書データベースを作成する
   b. 証明書要求を作成し、サーバーからサーバー証明書を提供する認証局 (CA) に対して送信する
   c. 新しい証明書をサーバーにインストールする
   d. CA、およびその CA が発行するすべての証明書を信頼する
2. LDAP 処理および DSML 処理用のセキュリティ保護されたポートの指定を含め、ディレクトリで SSL を有効化し、設定します。サーバーへのアクセスに SSL を使用するように Directory Server コンソールを設定することもできます。
3. 必要に応じて、サーバーに次のクライアント認証メカニズムを 1 つまたは複数設定します。
   a. デフォルトの証明書ベースの認証
   b. SASL を利用した DIGES_MD5 認証メカニズム
   c. Kerberos V5 セキュリティメカニズムを利用できる、SASL を利用した GSSAPI 認証
4. 使用するオプションの認証メカニズムの指定も含め、ディレクトリサーバーとの通信に SSL を使用するようにクライアントを設定します。

上の一部の手順は、コマンド行から証明書を管理するための `certutil` ツールを使って実行することもできます。このツールは、Sun ONE Directory Server Resource Kit に用意されています。詳細については、『Sun ONE Directory Server Resource Kit Tools Reference』の第 30 章「Security Tools」を参照してください。

サーバー証明書の入手とインストール

ここでは、証明書データベースの作成、Directory Server で使用する証明書の入手とインストール、および認証局 (CA) の証明書を信頼するように Directory Server を設定するそれぞれのプロセスについて説明します。

証明書データベースの作成

サーバーに初めて SSL を設定するときは、セキュリティデバイスのパスワードを設定する必要があります。既存のハードウェアセキュリティデバイスを使っていない場合は、次のファイルに格納されている証明書と鍵データベースが内部セキュリティデバイスとなります。

ServerRoot/alias/slapd-*serverID*-cert7.db
ServerRoot/alias/slapd-*serverID*-key3.db

serverID に大文字が含まれている場合は、後述するコマンド行の手順を実行して証明書データベースを作成する必要があります。

コンソールを使用

コンソールを使う場合、証明書マネージャダイアログを初めて呼び出すときに、サーバーが証明書データベースファイルを自動的に作成します。

1. Directory Server コンソールの最上位にある「タスク」タブで、「証明書の管理」ボタンをクリックします。あるいは、「タスク」タブを表示した状態でコンソールの「セキュリティ」メニューから「証明書の管理」を選択します。
2. サーバーは、証明書と鍵データベースを自動的に作成します。このとき、セキュリティデバイスのパスワードの設定が求められます。このパスワードは、サーバーに格納される、証明書の非公開鍵を保護します。確認のためにパスワードをもう一度入力し、「了解」をクリックします。

コマンド行を使用

コマンド行から証明書データベースファイルを作成するときは、サーバーが検出できるように、パスとファイル名の接頭辞を次の手順のように指定する必要があります。

1. サーバーのホストマシンで、次のコマンドを実行して証明書データベースを作成します。

   certutil -N -d *ServerRoot*/alias -P slapd-*LCserverID*-

   ここで、*LCserverID* は、すべて小文字で表記したサーバー名です。

   証明書の鍵を保護するためのパスワードの入力が求められます。

証明書要求の生成

PKCS #10 証明書要求を PEM 形式で生成するときは、次のいずれかの手順を実行します。PEM (Privacy Enhanced Mail) は、RFC 1421 ～ 1424 (`http://www.ietf.org/rfc/rfc1421.txt`) で指定されている形式で、US-ASCII 文字を使用した base64 形式で符号化されます。要求の内容は、次の例のようになります。

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBrjCCARcCAQAwbjELMAkGA1UBhMCVXMxEzARBgNVBAgTCkNBElGT1JOSUExLD
AqBgVBAoTI25ldHNjYXBlIGNvb11bmljYXRpb25zIGNvcnBvcmF0aWuMRwwGgYDV
QQDExNtZWxsb24umV0c2NhcGUuY29tMIGfMA0GCSqGSIb3DQEBAUAA4GNADCBiQK
BgCwAbskGh6SKYOgHy+UCSLnm3ok3X3u83Us7u0EfgSLR0f+K41eNqqWRftGR83e
mqPLDOf0ZLTLjVGJaHJn4l1gG+JDf/n/zMyahxtV7+T8GOFFigFfuxJaxMjr2j7I
vELlxQ4IfZgwqCm4qQecv3G+N9YdbjveMVXW0v4XwIDAQABAAwDQYJKoZIhvcNAQ
EEBQADgYEAZyZAm8UmP9PQYwNy4Pmypk79t2nvzKbwKVb97G+MT/gw1pLRsuBoKi
nMfLgKp1Q38K5Py2VGW1E47/rhm3yVQrIiwV+Z8Lcc=
-----END NEW CERTIFICATE REQUEST-----
```

コンソールを使用

1. Directory Server コンソールの最上位にある「タスク」タブで、「証明書の管理」ボタンをクリックします。あるいは、「タスク」タブを表示した状態でコンソールの「セキュリティ」メニューから「証明書の管理」を選択します。

   「証明書の管理」ダイアログが表示されます。

2. 「サーバー証明書」タブを選択し、「要求」ボタンをクリックします。

   証明書リクエストウィザードが表示されます。

3. サーバーが CA と直接通信するためのプラグインがインストールされている場合は、ここでそのプラグインを選択します。プラグインがインストールされていない場合は、生成した要求を電子メールまたは Web サイト経由で送信し、手動で証明書を要求する必要があります。「次へ」をクリックして処理を続けます。

4. 何も入力されていないテキストフィールドに要求者の情報を入力します。

   サーバー名：DNS 検索で使用される、Directory Server の完全修飾ホスト名を入力します (たとえば、`east.example.com`)。

   組織：企業または組織の正式名称を入力します。CA の多くは、ここに入力された情報を、営業許可証の複写などの法的文書で確認することを要求します。

   組織単位:(省略可能)。企業内の部署名またはビジネス単位の名称を入力します。

   都市 / 地域:(省略可能)。会社の所在地 (市町村名) を入力します。

   都道府県：会社の所在地 (都道府県) を完全名で入力します (省略形は不可)。

   国 / 地域：国名を表す 2 文字の略号 (ISO 形式) を選択します。日本の国コードは「JP」です。ISO 国コードのリストは、『Sun ONE Directory Server Reference Manual』の付録 C 「Directory Internationalization」に記載されています。

   「次へ」をクリックして処理を続けます。

5. セキュリティデバイスのパスワードを入力し、「次へ」をクリックします。これは、374 ページの「証明書データベースの作成」で設定したパスワードです。

6. 「クリップボードにコピー」または「ファイルに保存」を選択し、CA に送る必要のある証明書要求情報を保存します。

7. 「完了」をクリックして、証明書リクエストウィザードを終了します。

コマンド行を使用

1. 次のコマンドを実行して、サーバー証明書の要求を作成します。

   ```
   certutil -R ¥
   -s "cn=serverName,ou=division,o=company,l=city,st=state,c=country" ¥
   -a -d ServerRoot/alias -P slapd-serverID-
   ```

   `-s` オプションは、要求するサーバー証明書の DN を指定します。通常、CA はサーバーを完全に識別するために、この例に含まれるすべての属性を必要とします。各属性の説明については、手順 4 を参照してください。

2. `certutil` ツールは、サーバーの鍵データベースを保護しているパスワードの入力を要求します。これは、374 ページの「証明書データベースの作成」で設定したパスワードです。これにより、PEM で符号化されたテキスト形式の PKCS #10 証明書要求が生成されます。

サーバー証明書のインストール

ここで説明する手順に従って、前の項で生成した要求を CA に送信します。証明書要求は、電子メールとしての送信が求められることもありますが、CA の Web サイトに入力できる場合もあります。

証明書要求を送信したら、証明書に関する CA からの回答を待つ必要があります。要求に対する回答が届くまでの時間は、状況によって異なります。たとえば、CA が社内にある場合は、要求に対する回答は 1 ～ 2 日しかかからないこともありますが、CA が社外にある場合は、数週間かかることもあります。

CA から回答が届いたら、その情報をテキストファイルに確実に保存してください。次に、PEM 形式の PKCS #11 証明書の例を示します。PEM (Privacy Enhanced Mail) は、RFC 1421 ～ 1424 (`http://www.ietf.org/rfc/rfc1421.txt`) で指定されている形式で、US-ASCII 文字を使用した base64 形式で符号化されます。

```
-----BEGIN CERTIFICATE-----
MIICjCCAZugAwIBAgICCEEwDQYJKoZIhKqvcNAQFBQAwfDELMAkGA1UEBhMCVVMx
IzAhBgNVBAoGlBhbG9a2FWaWxsZGwSBXaWRnZXRzLCBJbmMuMR0wGwYDVQQLExRX
aWRnZXQgTW3FrZXJzICdSJyBVczEpMCcGAx1UEAxgVGVzdCBUXN0IFRlc3QgVGVz
dCBUZXN0IFlc3QgQ0EswHhcNOTgwMzEyMDIzMzUWhcNOTgwMzI2MDIzMpzU3WjBP
MQswCYDDVQQGEwJVUzEoMCYGA1UEChMfTmV0c2NhcGUgRGlyZN0b3J5VIFB1Ymxp
Y2F0aW9uczEWMB4QGA1UEAxMNZHVgh49dq2tLNvbjTBaMA0GCSqGSIb3DQEBAQUA
A0kAMEYkCQCksMR/aLGdfp4m0OiGgijG5KgOsyRNvwGYW7kfW+8mmijDtZaRjYNj
jcgpF3Vnlbxbc1X9LVjjNLC5737XZdAgEDozYwpNDARBglghkgBhvhCEAQEEBAMC
APAwHkwYDVR0jBBgwFAU67URjwCaGqZHUpSpdLxlzwJKiMwDQYJKoZIhQvcNAQEF
BQADgYEAJ+BfVem3vBOPBveNdLGfjlb9hucgmaMcQa9FA/db8qimKT/ue9UGOJqL
bwbMKBBopsDn56p2yV3PLIsBgrcuSoBCuFFnxBnqSiTS7YiYgCWqWaUA0ExJFmD6
6hBLseqkSWulk+hXHN7L/NrViO+7zNtKcaZLlFPf7d7j2MgX4Bo=
-----END CERTIFICATE-----
```

また、証明書データのバックアップを安全な場所に置く必要もあります。これにより、システムに保存された証明書データが失われても、バックアップファイルから証明書をインストールし直すことができます。

サーバー証明書を受け取ったら、それをサーバーの証明書データベースにインストールできます。

コンソールを使用

1. Directory Server コンソールの最上位にある「タスク」タブで、「証明書の管理」ボタンをクリックします。あるいは、「タスク」タブを表示した状態でコンソールの「セキュリティ」メニューから「証明書の管理」を選択します。

   「証明書の管理」ウィンドウが表示されます。

2. 「サーバー証明書」タブを選択し、「インストール」をクリックします。

   証明書インストールウィザードが表示されます。

3. 次のオプションから証明書の保存場所を 1 つ選択します。

   このファイル内：証明書の絶対パスをこのフィールドに入力します。

   次の符号化されたテキストブロック中：CA から、または作成したテキストファイルからテキストをコピーし、このフィールドにペーストします。

   「次へ」をクリックして処理を続けます。

4. 表示された証明書情報が正しいことを確認し、「次へ」をクリックします。

5. 証明書の名前を指定し、「次へ」をクリックします。これは、証明書のテーブルに表示される名前です。

6. 非公開鍵を保護するパスワードを入力して、証明書を検証します。このパスワードは、374 ページの「証明書データベースの作成」の手順 2 で入力したものと同じです。処理が終了したら、「完了」をクリックします。

   「サーバー証明書」タブのリストに新しい証明書が表示されます。これで、サーバーで SSL を有効にする準備が整いました。

コマンド行を使用

1. 次のコマンドを実行して、証明書データベースに新しいサーバー証明書をインストールします。

   ```
   certutil -A -n "certificateName" -t "u,," -a -i certFile ¥
            -d ServerRoot/alias -P slapd-serverID-
   ```

   ここで、*certificateName* は証明書を識別するために指定する名前です。*certFile* は、PEM 形式の PKCS #11 証明書を含むテキストファイルです。`-t "u,,"` オプションは、この証明書が SSL 通信用のサーバー証明書であることを示します。

2. 必要に応じて次の `certutil` コマンドを実行して、インストールした証明書を検証することができます。

   ```
   certutil -L -d ServerRoot/alias -P slapd-serverID-
   ```

   `u,,` という信頼属性が表示される証明書は、サーバー証明書です。

CA の信頼設定

CA を信頼するように Directory Server を設定するには、証明書を入手し、それをサーバーの証明書データベースにインストールします。このプロセスは、使用する CA によって異なります。一部の商用 CA では、証明書を自動的にダウンロードできる Web サイトを用意しています。それ以外の CA からは、要求に応じて電子メールで証明書が送られます。

コンソールを使用

CA の証明書を入手したら、証明書インストールウィザードを使って、CA を信頼するように Directory Server を設定できます。

1. Directory Server コンソールの最上位にある「タスク」タブで、「証明書の管理」ボタンをクリックします。あるいは、「タスク」タブを表示した状態でコンソールの「セキュリティ」メニューから「証明書の管理」を選択します。

   「証明書の管理」ウィンドウが表示されます。

2. 「CA 証明書」タブを選択し、「インストール」をクリックします。

   証明書インストールウィザードが表示されます。

3. CA の証明書をファイルに保存した場合は、ファイルのパスを該当のフィールドに入力します。CA の証明書を電子メールで受け取った場合は、ヘッダーを含む証明書をコピーし、該当のテキストフィールドにペーストします。「次へ」をクリックします。

4. 表示された証明書情報が正しい CA からの証明書に関するものであることを確認し、「次へ」をクリックします。

5. 証明書の名前を指定し、「次へ」をクリックします。

6. この CA を信頼する目的を選択します。次のいずれか、または両方を選択できます。

   クライアントからの接続を受け入れる (クライアント認証) : LDAP クライアントが、この CA が発行する証明書を示すことで証明書ベースのクライアント認証を行う場合は、このチェックボックスを選択します。

   ほかのサーバーに接続する (サーバー認証) : サーバーが、この CA が発行する証明書を持つ他のサーバーに対して SSL 経由のレプリケーションサプライヤーまたは連鎖マルチプレクサとして機能する場合は、このチェックボックスを選択します。

7. 「完了」をクリックして、ウィザードを終了します。

コマンド行を使用

1. 次のコマンドを実行して、CA 証明書を信頼することもできます。

```
certutil -A -n "CAcertificateName" -t "trust,," -a -i certFile ¥
         -d ServerRoot/alias -P slapd-serverID-
```

ここで、*CAcertificateName* は信頼した CA を識別するために指定する名前です。*certFile* は、PEM で符号化されたテキスト形式の PKCS #11 CA 証明書を含むテキストファイルです。*trust* には、次のいずれかのコードを指定します。

- `T`: クライアント証明書の発行について、この証明書を信頼する。LDAP クライアントが、この CA が発行する証明書を示すことで証明書ベースのクライアント認証を行う場合は、このコードを選択する
- `C`: サーバー証明書の発行について、この証明書を信頼する。サーバーが、この CA が発行する証明書を持つ他のサーバーに対して SSL 経由のレプリケーションサプライヤーまたは連鎖マルチプレクサとして機能する場合は、このコードを選択する
- `CT`: クライアント証明書とサーバー証明書の両方の発行について、この証明書を信頼する。上記の両方にこの CA を適用するときは、このコードを選択する

2. 必要に応じて、次の `certutil` コマンドを実行して、インストールした証明書を検証することができます。

   ```
   certutil -L -d ServerRoot/alias -P slapd-serverID-
   ```

   `u,,` という信頼属性が表示される証明書はサーバー証明書で、`CT,,` が表示される証明書は CA 証明書です。

SSL の有効化

サーバー証明書をインストールし、CA の証明書を信頼すると、SSL を有効にすることができます。通常、サーバーは SSL を有効にした状態で動作させます。SSL を一時的に無効にする場合は、機密性、認証、またはデータの整合性を必要とする操作を処理する前に、SSL を必ず有効にしてください。

SSL を有効にする前に、374 ページの「サーバー証明書の入手とインストール」の説明に従って、証明書データベースを作成し、サーバーの証明書を入手およびインストールして、CA の証明書を信頼する必要があります。

次に、次の手順を実行して SSL 通信を有効にし、ディレクトリサーバー内の暗号化メカニズムを有効化します。

1. Directory Server コンソールの最上位の「設定」タブで、サーバー名のルートノードを選び、右側のパネルで「暗号化」タブを選びます。

   このタブには、サーバーの現在の暗号化設定が表示されます。

2. 「このサーバーの SSL を有効にする」チェックボックスを選択して、暗号化を有効にするよう指定します。
3. 「この暗号化方式ファミリを使用」チェックボックスを選択します。
4. ドロップダウンメニューから使用する証明書を選択します。
5. 「暗号化方式」の「設定」ボタンをクリックし、「暗号化方式のプリファレンス」ダイアログで使用する暗号化方式を選択します。暗号化方式の詳細については、382 ページの「暗号化方式の選択」を参照してください。
6. クライアント認証を設定します。

   クライアント認証を許可しない：このオプションを選択すると、サーバーはクライアントの証明書または SASL セキュリティメカニズムを無視し、バインド DN とパスワードを要求します。

   クライアント認証を許可する：これはデフォルトの設定です。このオプションを選択すると、クライアントの要求に対して認証が実行されます。証明書に基づく認証については、385 ページの「クライアント認証の設定」を参照してください。

注	証明書に基づく認証をレプリケーションに使用する場合は、クライアント認証を許可するか、または要求するようにコンシューマサーバーを設定する必要があります。

   クライアント認証を要求する：このオプションを選択すると、サーバーからの認証の要求にクライアントが応答しない場合に、クライアント接続が拒否されます。

注	Sun ONE サーバーコンソールが SSL 経由で Directory Server に接続する場合に「クライアント認証を要求する」を選択すると、Sun ONE サーバーコンソールにはクライアント認証に使用する証明書が用意されていないため、接続が無効になります。この属性をコマンド行から変更する方法については、384 ページの「クライアント認証の許可」を参照してください。

7. 必要に応じて、Directory Server との通信にコンソールが SSL を使うときは、「Sun ONE サーバーコンソールで SSL を使用」を選択します。
8. 設定が完了したら「保存」をクリックします。

9. 必要に応じて、サーバーが LDAP プロトコルと DSML-over-HTTP プロトコルの両方で SSL 通信に使うセキュリティ保護されたポートを設定します。詳細は、35 ページの「Directory Server のポート番号の変更」を参照してください。

   セキュリティ保護されたポートへのすべての接続は、SSL を使用する必要があります。SSL を有効にした場合は、セキュリティ保護されたポートを設定するかどうかに関係なく、クライアントは Start TLS 処理を使用して、セキュリティ保護されていないポートを通じて SSL 暗号化を行えます。

10. Directory Server を再起動します。

    詳細は、22 ページの「SSL が有効になった状態でのサーバーの起動」を参照してください。

暗号化方式の選択

暗号化方式は、データの暗号化と復号化に使用されるアルゴリズムです。一般に、暗号化に使用するビット数が多いほど、強度と安全性は高まります。SSL の暗号化方式は、使用するメッセージ認証のタイプによっても識別されます。メッセージ認証は、データの整合性を保証するチェックサムを計算する別のアルゴリズムです。暗号化アルゴリズムとその強度の詳細については、『Sun ONE Server Console Server Management Guide』の付録 B にある「Cipher Settings」を参照してください。

クライアントがサーバーとの SSL 接続を開始するときは、情報の暗号化にどの暗号を使用するかをについて、クライアントとサーバーが合意する必要があります。双方向の暗号化プロセスでは、サーバーとクライアントで同じ暗号化方式を使用する必要があり、通常は、両者が共通して対応している最強の方式が選択されます。

Sun ONE Directory Server には、SSL3.0 と TLS 用に次の暗号化方式が用意されています。

表 11-1 Sun ONE Directory Server が提供する暗号化方式

暗号化方式名	説明
なし	暗号化なし。MD5 メッセージ認証だけを使用する (rsa_null_md5)
RC4 (128 ビット)	128 ビットの暗号化と MD5 メッセージ認証を使用した RC4 暗号化方式 (rsa_rc4_128_md5)
RC4 (エクスポート)	40 ビットの暗号化と MD5 メッセージ認証を使用した RC4 暗号化方式 (rsa_rc4_40_md5)
RC2 (エクスポート)	40 ビットの暗号化と MD5 メッセージ認証を使用した RC2 暗号化方式 (rsa_rc2_40_md5)

表 11-1　Sun ONE Directory Server が提供する暗号化方式 (続き)

暗号化方式名	説明
DES または DES (エクスポート)	56 ビットの暗号化と SHA メッセージ認証を使用した DES (rsa_des_sha)
DES (FIPS)	56 ビットの暗号化と SHA メッセージ認証を使用した FIPS DES。この暗号化方式は、暗号化モジュール (rsa_fips_des_sha) の実装用 FIPS 140-1 米国政府規格に準拠する
トリプル DES	168 ビットの暗号化と SHA メッセージ認証を使用したトリプル DES (rsa_3des_sha)
トリプル DES (FIPS)	168 ビットの暗号化と SHA メッセージ認証を使用した FIPS トリプル DES。この暗号化方式は、暗号化モジュール (rsa_fips_3des_sha) の実装用 FIPS 140-1 米国政府規格に準拠する
Fortezza	80 ビットの暗号化と SHA メッセージ認証を使用する Fortezza 暗号化方式
RC4 (Fortezza)	128 ビットの暗号化と SHA メッセージ認証を使用する Fortezza RC4 暗号化方式
なし (Fortezza)	暗号化は行われず、Fortezza SHA メッセージ認証のみ

Sun ONE サーバーコンソールで常に SSL を使用するには、次の暗号の中から少なくとも 1 つの暗号を選択する必要があります。

- 40 ビットの暗号化と MD5 メッセージ認証を使用する RC4 暗号
- 暗号化は行われず、MD5 メッセージ認証のみ (お勧めできません)
- 56 ビットの暗号化と SHA メッセージ認証を使用する DES
- 128 ビットの暗号化と MD5 メッセージ認証を使用する RC4 暗号
- 168 ビットの暗号化と SHA メッセージ認証を使用するトリプル DES

サーバーが使用する暗号化方式を選択するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで、サーバー名のルートノードを選び、右側のパネルで「暗号化」タブを選びます。

   このタブには、サーバーの現在の暗号化設定が表示されます。380 ページの「SSL の有効化」で説明している方法で、SSL がサーバーで有効になっていることを確認します。

2. 「暗号化方式のプリファレンス」をクリックします。

   「暗号化方式のプリファレンス」ダイアログボックスが表示されます。

3. 「暗号化方式のプリファレンス」ダイアログボックスで、暗号化方式の名前の隣にあるチェックボックスを選択または選択解除して、サーバーで使用する暗号化方式を選択します。

   セキュリティ上の理由で特定の暗号を使用できない場合を除き、none,MD5 以外のすべての暗号を選択します。

警告 暗号化を行わずに MD5 メッセージ認証だけを行うことは避けてください。クライアントで使用可能な暗号がほかにない場合に、サーバーはこのオプションを使用します。この方式は暗号化が行われないため、セキュリティ保護されません。

4. 「暗号化方式のプリファレンス」ダイアログの「了解」をクリックし、「暗号化」タブの「保存」をクリックします。

クライアント認証の許可

Directory Server にクライアント認証を要求し、SSL を使用して Sun ONE サーバーコンソールに接続するように設定した場合は、Sun ONE サーバーコンソールを使って Sun ONE サーバーを管理することができなくなります。その代わりに、該当するコマンド行ユーティリティを使用する必要があります。

ただし、Sun ONE サーバーコンソールを使用できるようにディレクトリ設定を変更するときは、次の手順を実行して、クライアント認証を要求するのではなく許可するように設定します。

1. 次のコマンドを実行して cn=encryption,cn=config エントリを変更します。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=encryption,cn=config
changetype: modify
replace: nsSSLClientAuth
nsSSLClientAuth: allowed
```

2. 21 ページの「コマンド行からのサーバーの起動と停止 (UNIX)」で説明している方法で、Directory Server を再起動します。

   これで、Sun ONE サーバーコンソールを起動できるようになりました。

クライアント認証の設定

クライアント認証は、サーバーがクライアントの識別情報を検証するメカニズムです。クライアント認証は、クライアントが定義する証明書を使用するか、DIGEST-MD5 などの SASL ベースのメカニズムを利用して行われます。Solaris オペレーティングシステムでは、Directory Server は SASL による GSSAPI メカニズムをサポートするようになりました。これにより、Kerberos V5 によるクライアント認証が可能になりました。

証明書ベースの認証では、SSL プロトコルを介して入手したクライアント証明書を使って、ユーザーエントリの識別情報が検出されます。このエントリには、認証されるユーザーと同じ証明書が含まれている必要があります。このメカニズムは SASL メカニズムの外部で処理されるため、EXTERNAL とも呼ばれます。証明書ベースの認証の詳細については、『Sun ONE Server Console Server Management Guide』の第 10 章にある「Using Client Authentication」を参照してください。

次に、2 つの SASL メカニズムをディレクトリサーバーに設定する方法について説明します。393 ページの「LDAP クライアントでセキュリティを使用するための設定」も参照してください。

DIGEST-MD5 を利用した SASL 認証

DIGEST-MD5 メカニズムは、クライアントが送信したハッシュ値とユーザーのパスワードのハッシュ値を比較してクライアントを認証します。ただし、メカニズムがユーザーパスワードを読み取る必要があるため、DIGEST-MD5 による認証を受けるすべてのユーザーは、ディレクトリ内に {CLEAR} パスワードを持っている必要があります。

DIGEST-MD5 メカニズムの設定

DIGEST-MD5 を使用するように Directory Server を設定するときは、次の手順を実行します。

1. コンソールまたは ldapsearch コマンドを使用して、ルートエントリの supportedSASLMechanisms 属性の値が DIGEST-MD5 であることを確認します。たとえば、次のコマンドはどの SASL メカニズムが有効であるかを表示します。

   ```
   ldapsearch -h host -p port -D "cn=Directory Manager" -w password ¥
   -s base -b "" "(objectclass=*)" supportedSASLMechanisms
   ```

   ```
   dn:
   supportedSASLMechanisms: EXTERNAL
   supportedSASLMechanisms: DIGEST-MD5
   supportedSASLMechanisms: GSSAPI
   ```

2. DIGEST-MD5 が有効でない場合は、次の ldapmodify コマンドを実行して有効化します。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=SASL, cn=security, cn=config
changetype: modify
add: dsSaslPluginsEnable
dsSaslPluginsEnable: DIGEST-MD5
-
replace: dsSaslPluginsPath
dsSaslPluginsPath: ServerRoot/lib/sasl
```

3. DIGEST-MD5 のデフォルトの ID マッピングを使用するか、386 ページの「DIGEST-MD5 ID マッピング」で説明している方法で新規作成します。

4. DIGEST-MD5 を使用する SSL 経由でサーバーにアクセスするすべてのユーザーのパスワードが {CLEAR} に含まれていることを確認します。パスワードの保存スキームを設定する方法については、第 7 章「ユーザーアカウントの管理」を参照してください。

警告 ディレクトリに {CLEAR} パスワードを格納するときは、パスワード値へのアクセスが ACI によって適切に制限されていることを確認する必要があります。アクセス制御については、第 6 章「アクセス制御の管理」を参照してください。サフィックスに属性の暗号化を設定して {CLEAR} パスワードをさらに保護する方法については、77 ページの「属性値の暗号化」を参照してください。

5. SASL 設定エントリ、または DIGEST-MD5 ID マッピングエントリの 1 つを変更した場合は、ディレクトリサーバーを再起動します。

DIGEST-MD5 ID マッピング

SASL メカニズムの ID マッピングでは、SASL ID の証明情報と、ディレクトリ内のユーザーエントリの一致が確認されます。このメカニズムの詳細については、391 ページの「ID マッピング」を参照してください。マッピングによって、SASL ID に対応する DN が見つからなかった場合は、認証は失敗します。

SASL ID は、*Principal* という文字列です。これは、各メカニズムに固有の形式でユーザーを表します。DIGEST-MD5 では、クライアントは、dn: 接頭辞と LDAP DN、または u: 接頭辞の後にクライアントが決定するテキストを続けた情報のいずれかが含まれる主体を作成することをお勧めします。マッピング時に、クライアントが送信した主体は、${Principal} プレースホルダで使用されます。

DIGEST-MD5 のデフォルトの ID マッピングは、サーバー設定の次のエントリで指定されます。

```
dn: cn=default,cn=DIGEST-MD5,cn=identity mapping,cn=config
objectClass: top
objectClass: nsContainer
objectClass: dsIdentityMapping
objectClass: dsPatternMatching
cn: デフォルト
dsMatching-pattern: ${Principal}
dsMatching-regexp: dn:(.*)
dsMappedDN: $1
```

この ID マッピングは、主体の dn フィールドに、ディレクトリ内の既存ユーザーの正確な DN が含まれていることを前提としています。

DIGEST-MD5 用の独自の ID マッピングを定義するには、次の手順を実行します。

1. cn=DIGEST-MD5,cn=identity mapping,cn=config の下でデフォルトのマッピングエントリを編集するか、新しいマッピングエントリを作成します。ID マッピングエントリの属性の定義については、391 ページの「ID マッピング」を参照してください。DIGEST-MD5 用のマッピングの例は、次のファイルに保存されています。

   ServerRoot/slapd-*serverID*/ldif/identityMapping_Examples.ldif

   この例は、主体の修飾されていないテキストフィールドに、指定の ID のユーザー名が含まれることを前提としています。次のコマンドは、このマッピングを定義する方法を示しています。

   ```
   ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
   dn: cn=unqualified-username,cn=DIGEST-MD5,cn=identity mapping,
    cn=config
   objectclass: dsIdentityMapping
   objectclass: dsPatternMatching
   objectclass: nsContainer
   objectclass: top
   cn: unqualified-username
   dsMatching-pattern: ${Principal}
   dsMatching-regexp: u:(.*)@(.*)¥.com
   dsSearchBaseDN: dc=$2
   dsSearchFilter: (uid=$1)
   ```

2. 新しいマッピングを有効にするには、Directory Server を再起動します。

GSSAPI を利用した SASL 認証 (Solaris のみ)

SASL を介した GSSAPI (Generic Security Services API) では、クライアントの認証に Kerberos V5 などのサードパーティ製セキュリティメカニズムを利用できます。GSSAPI ライブラリは、Solaris プラットフォームだけで利用できます。SEAM (Sun Enterprise Authentication Mechanism) 1.0.1 サーバーに Kerberos V5 実装をインストールすることをお勧めします。

サーバーは、この API を使ってユーザーの識別情報を検証します。次に、SASL メカニズムは GSSAPI マッピングルールを適用して、この接続中のすべての操作のバインド DN となる DN を取得します。

Keberos システムの設定

製造元の指示に従って、Kerberos ソフトウェアを設定します。SEAM 1.0.1 サーバーを利用している場合は、次の手順を実行します。

1. `/etc/krb5` 内のファイルを設定します。
2. ユーザーとサービスを格納する Kerberos データベースを作成し、その中に LDAP サービスの主体を作成します。LDAP サービスの主体は次のとおりです。

   `ldap/`*`serverFQDN@REALM`*

   ここで、*serverFQDN* はサーバーの完全修飾ドメイン名です。
3. LDAP サービスの鍵を含む、サービス鍵を格納する鍵タブを作成します。
4. Kerberos デーモンプロセスを開始します。

各手順の詳細については、ソフトウェアのマニュアルを参照してください。

GSSAPI メカニズムの設定

Solaris プラットフォームで GSSAPI を使用するように Directory Server を設定するときは、次の手順を実行します。

1. コンソールまたは `ldapsearch` コマンドを使用して、ルートエントリの `supportedSASLMechanisms` 属性の値が `GSSAPI` であることを確認します。たとえば、次のコマンドはどの SASL メカニズムが有効であるかを表示します。

   ```
   ldapsearch -h host -p port -D "cn=Directory Manager" -w password ¥
   -s base -b "" "(objectclass=*)" supportedSASLMechanisms

   dn:
   supportedSASLMechanisms: EXTERNAL
   supportedSASLMechanisms: DIGEST-MD5
   ```
2. デフォルトでは、GSSAPI は無効化されているので、次の `ldapmodify` コマンドを実行して有効化します。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=SASL, cn=security, cn=config
changetype: modify
add: dsSaslPluginsEnable
dsSaslPluginsEnable: GSSAPI
-
replace: dsSaslPluginsPath
dsSaslPluginsPath: ServerRoot/lib/sasl
```

3. 389 ページの「GSSAPI ID マッピング」で説明する方法で、GSSAPI のデフォルトの ID マッピングと、必要に応じてカスタムマッピングを作成します。
4. ホストマシン上で、サーバーの Kerberos を設定します。
   a. セッション鍵を使って、LDAP サービス主体 (`ldap/`*serverHostname*`@`*Realm*) を Kerberos に作成します。
      - *serverHostname* は、サーバーホストマシンの完全修飾ドメイン名です。この値は、`cn=config` の `nsslapd-localhost` 属性と同じ値である必要があります。ただし、こちらはすべて小文字で指定します。
      - *Realm* は、サーバーの Kerberos Realm です。
   b. LDAP サービスは、次のファイルに保存されている鍵データベースに対する読み取りアクセス権をもっている必要があります。
      `/etc/krbs/krb5.keytab`
   c. DNS は、ホストマシンに設定されている必要があります。
5. SASL 設定エントリ、または GSSAPI ID マッピングエントリの 1 つを変更したときは、ディレクトリサーバーを再起動します。

GSSAPI ID マッピング

SASL メカニズムの ID マッピングでは、SASL ID の証明情報と、ディレクトリ内のユーザーエントリの一致が確認されます。このメカニズムの詳細については、391 ページの「ID マッピング」を参照してください。マッピングによって、SASL ID に対応する DN が見つからなかったときは、認証は失敗します。

SASL ID は、*Principal* という文字列です。これは、各メカニズムに固有の形式でユーザーを表します。GSSAPI を利用する Kerberos の主体は、*uid*`[/`*instance*`] [@`*realm*`]` という形式の ID です。ここで、*uid* には、オプションの *instance* 識別子と、それに続けてオプションの *realm* (多くの場合はドメイン名) が含まれることがあります。次の例は、いずれも有効なユーザー主体です。

```
bjensen
bjensen/Sales
bjensen@EXAMPLE.COM
bjensen/Sales@EXAMPLE.COM
```

最初は、ディレクトリ内にはGSSAPIマッピングは定義されていません。デフォルトのマッピングを定義し、使用する主体をクライアントがどのように定義するかに応じてカスタムマッピングを定義する必要があります。

GSSAPI用のIDマッピングを定義するには、次の手順を実行します。

1. `cn=GSSAPI,cn=identity mapping, cn=config` の下に新しいマッピングエントリを作成します。IDマッピングエントリの属性の定義については、391ページの「IDマッピング」を参照してください。

   GSSAPIマッピングの例は、次のファイルに保存されています。

   ServerRoot`/slapd-`*serverID*`/ldif/identityMapping_Examples.ldif`

   このファイルに含まれるデフォルトGSSAPIマッピングは、主体にユーザーIDだけが含まれ、これがディレクトリの固定ブランチ内のユーザーを決定することを前提としています。

   ```
   dn: cn=default,cn=GSSAPI,cn=identity mapping,cn=config
   objectclass: dsIdentityMapping
   objectclass: nsContainer
   objectclass: top
   cn: default
   dsMappedDN: uid=${Principal},ou=people,dc=example,dc=com
   ```

   このファイルに含まれるもう一つの例は、既知のRealmを含む主体にユーザーIDが記録されている場合に、ユーザーIDを決定する方法を示しています。

   ```
   dn: cn=same_realm,cn=GSSAPI,cn=identity mapping,cn=config
   objectclass: dsIdentityMapping
   objectclass: dsPatternMatching
   objectclass: nsContainer
   objectclass: top
   cn: same_realm
   dsMatching-pattern: ${Principal}
   dsMatching-regexp: (.*)@example.com
   dsMappedDN: uid=$1,ou=people,dc=example,dc=com
   ```

2. 新しいマッピングを有効にするには、Directory Serverを再起動します。

ID マッピング

Directory Server のいくつかの認証メカニズムでは、別のプロトコルの証明情報をディレクトリ内の DN にマッピングする必要があります。現時点では、DSML-over-HTTP プロトコルと、DIGEST-MD5 メカニズムおよび GSSAPI SASL メカニズムの場合がこれにあたります。いずれの場合も ID マッピングを使用して、クライアントが提示するプロトコル固有の証明情報に基づいてバインド ID を決定しています。

ID マッピングには、`cn=identity mapping, cn=config` 設定ブランチに含まれるエントリが利用されます。この分岐には、ID マッピングを必要とするプロトコルごとにコンテナが含まれます。

- `cn=HTTP-BASIC, cn=identity mapping, cn=config` : DSML-over-HTTP 接続用のマッピングを含む
- `cn=DIGEST-MD5, cn=identity mapping, cn=config` : DIGEST-MD5 SASL メカニズムによるクライアント認証に必要なマッピングを含む
- `cn=GSSAPI, cn=identity mapping, cn=config` : GSSAPI SASL メカニズムによるクライアント認証に必要なマッピングを含む

マッピングエントリは、ディレクトリの検索に利用できるように、プロトコルに固有の証明情報の要素を展開する方法を定義します。検索が 1 つのユーザーエントリを返す場合、マッピングは成功し、接続ではすべての操作のバインド DN としてこのエントリが使われます。検索結果がゼロ、または複数のエントリである場合、マッピングは失敗し、他のマッピングが適用されます。

各分岐には、そのプロトコルのデフォルトのマッピングと、任意の数のカスタムマッピングが含まれます。デフォルトのマッピングには `cn=default` という RDN が含まれ、カスタムマッピングには `cn` をネーミング属性として使用した別の RDN が含まれることがあります。最初に、いずれかのマッピングが成功するまで、すべてのカスタムマッピングが順不同に評価されます。すべてのカスタムマッピングが失敗した場合は、最後にデフォルトのマッピングが適用されます。デフォルトマッピングも失敗した場合は、そのクライアントの認証は失敗します。

マッピングエントリには、`top`、`Container`、`dsIdentityMapping` オブジェクトクラスが含まれている必要があります。エントリには、次の属性を含めることができます。

- `dsMappedDN:` *DN* : ディレクトリ内の DN を定義するリテラル文字列。マッピングの実行時にこの DN が存在する場合、この DN はバインドに使われます。この DN が存在しない場合に検索を実行するために、次の属性を定義することもできます。
- `dsSearchBaseDN:` *DN* : 検索のベース DN。この属性を省略すると、マッピングではディレクトリツリー全体ですべてのルートサフィックスが検索されます。

- dsSearchScope: base|one|sub : 検索の適用範囲。検索ベース、ベースの 1 つ下のレベル、またはベースの下にあるサブツリー全体を指定できます。この属性を省略すると、マッピング検索の適用範囲は、デフォルトでサブツリー全体となります。
- dsSearchFilter: *filterString* : マッピング検索を実行するためのフィルタ文字列。LDAP 検索フィルタは、RFC 2254 (http://www.ietf.org/rfc/rfc2254.txt) に定義されています。

また、マッピングエントリには dsPatternMatching オブジェクトクラスを含めることができます。このオブジェクトクラスでは、次の属性を利用できます。

- dsMatching-pattern: *patternString* : パターンマッチングを実行する文字列を指定します。
- dsMatching-regexp: *regularExpression* : パターン文字列に適用する正規表現を指定します。

これらのすべての属性値は、${*keyword*} という形式のプレースホルダを持つことができます (dsSearchScope を除く)。*keyword* は、プロトコル固有の証明情報に含まれる要素名です。プレースホルダは、マッピング時にクライアントから提供される要素の実際の値に置き換えられます。

すべてのプレースホルダの置換が完了すると、定義されているパターンマッチングが行われます。一致するパターンは、正規表現と比較されます。正規表現がパターン文字列と一致しない場合、そのマッピングは失敗です。一致する場合、正規表現中でカッコで囲まれている値は、番号つきのプレースホルダとして他の属性値で利用できます。たとえば、SASL 用に次のマッピングを定義できます。

```
dsMatching-pattern: ${Principal}
dsMatching-regexp: (.*)@(.*)¥.(.*)
dsMappedDN: uid=$1,ou=people,dc=$2,dc=$3
```

bjensen@example.com という主体でクライアント認証を行う場合、このマッピングは uid=bjensen,ou=people,dc=example,dc=com というバインド DN を定義します。この DN がディレクトリに存在すれば、マッピングは成功し、クライアントは認証されます。また、この接続で実行されるすべての操作には、このバインド DN が使われます。

dsMatching-pattern は、Posix regexec(3C) および regcomp(3C) 関数呼び出しを使用して、dsMatching-regexp と比較されます。Directory Server では拡張正規表現が使用され、すべての比較で大文字と小文字が区別されません。詳細については、これらの関数の man ページを参照してください。

これらのプレースホルダを使う可能性のある属性値では、プレースホルダを使わない場合でも、プレースホルダ部分以外の $、{、および } の各文字を符号化する必要があります。$ は ¥24、{ は ¥7B、および } は ¥7D という値にそれぞれ符号化します。

プレースホルダにより値の置き換えを利用することで、プロトコル固有の証明情報からユーザー名などの値を抽出し、その値を使ってマップ先の DN を定義したり、ディレクトリ内の任意の場所にある対応する DN を検索するようなマッピングを作成できます。ディレクトリクライアントが提供する証明情報に含まれていることが予想される値を抽出し、特定のディレクトリ構造にそれをマップするマッピングを作成する必要があります。

警告 マッピングの定義が不十分だと、セキュリティホールが生じます。たとえば、パターンマッチングを利用しないハードコードされた DN へのマッピングは常に成功します。このため、ディレクトリユーザー以外のクライアントを認証してしまいます。

形式の異なるクライアント証明情報に対応するには、汎用的に、より許容傾向のある 1 つのマッピングを作成するよりも、複数のマッピングを定義するほうが安全です。クライアントの証明情報に基づいて、クライアント接続を特定のユーザーにマップするように心がける必要があります。

LDAP クライアントでセキュリティを使用するための設定

次に、ディレクトリサーバーとのセキュリティ保護された接続を確立するために、LDAP クライアントが SSL を使用できるように設定する方法について説明します。SSL 接続では、サーバーがクライアントに証明書を送信します。クライアントは、証明書を信頼することで、最初にサーバーを認証する必要があります。次に、必要に応じてクライアントが独自の証明書、または DIGEST-MD5 または Kerberos V5 による GSSAPI のいずれかの SASL メカニズムの情報を送信することで、いずれかのクライアント認証メカニズムを開始できます。

次の各項では、SSL が有効な LDAP クライアントの例として、`ldapsearch` ツールを使用します。ディレクトリサーバーが提供する `ldapmodify`、`ldapdelete`、および `ldapcompare` ツールは、同じ方法で設定できます。これらのディレクトリアクセスツールは、Sun ONE LDAP SDK for C に基づいています。詳細については、『Sun ONE Directory Server Resource Kit Tools Reference』を参照してください。

他の LDAP クライアントに SSL 接続を設定する方法については、アプリケーションに付属するマニュアルを参照してください。

注 クライアントアプリケーションによっては、SSL を実装しても、信頼された証明書がサーバーにあるかどうかを検証しません。これらのアプリケーションは SSL プロトコルを使用してデータの暗号化を行いますが、機密の保護を保証することも第 3 者がユーザーとして認証されることを防止することもできません。

クライアントでのサーバー認証の設定

クライアントがサーバーとの SSL 接続を確立するときは、サーバーが定義する証明書を信頼する必要があります。このとき、クライアントは次の処理を行う必要があります。

- 証明書データベースを用意する
- サーバー証明書を発行する CA を信頼する
- LDAP クライアントの SSL オプションを指定する

Netscape Communicator は、Web サーバーとの通信に HTTP プロトコルを介して SSL を使用するクライアントアプリケーションです。Communicator を使用して、LDAP クライアントが使用する証明書を管理することもできます。また、`certutil` コマンド行ツールを使用して、証明書データベースを管理することもできます。

Communicator によるクライアント証明書の管理

次に、Netscape Communicator を使ってクライアントマシン上の証明書データベースを管理する方法について説明します。

1. Netscape Communicator は、起動時に証明書データベースが存在することを確認し、存在しない場合は証明書データベースを作成します。証明書データベースは、Communicator の設定情報とともにファイルとして保存されます。たとえば、UNIX システムでは、`/home/`*`username`*`/.netscape/cert7.db` のようなファイルに保存されます。

   この手順を利用する場合、Communicator が作成する証明書データベースを特定し、クライアントアプリケーションが利用できるように、そのパスを記録しておきます。

2. Communicator を使って、アクセスするディレクトリサーバーの証明書を発行した CA の Web サイトを表示します。Communicator は、自動的に CA の証明書を取得し、それを信頼するかどうかを確認するメッセージを表示します。

   たとえば、内部に導入された Sun ONE Certificate Server を使用している場合は、`https://`*`hostname`*`:444` という URL にアクセスします。

3. Communicator が CA の証明書を信頼するかどうかを確認するメッセージを表示したら、それを信頼します。サーバー認証のためには、CA の証明書を信頼する必要があります。

   CA の Web サイトによっては、この手順を実行することはできません。Communicator が CA 証明書の信頼を確認するメッセージを表示しない場合は、次の手順を実行して手動で処理します。

コマンド行からのクライアント証明書の管理

コマンド行から証明書を管理するときは、`certutil` ツールを使います。このツールは、Sun ONE Directory Server Resource Kit に用意されています。詳細については、『Sun ONE Directory Server Resource Kit Tools Reference』の第 30 章「Security Tools」を参照してください。

1. クライアントのホストマシンで、次のコマンドを実行して証明書データベースを作成します。

   `certutil -N -d` *path* `-P` *prefix*

   証明書の保護のためにパスワードを入力するように求めるメッセージが表示されます。次に、ツールは *path*/*prefix*`cert7.db` ファイル、および *path*/*prefix*`key3.db` ファイルを作成します。

   証明書データベースは、LDAP クライアントアプリケーションのユーザーごとに、そのユーザーだけがアクセスできる場所に個別に作成する必要があります。たとえば、ユーザーのホームディレクトリ内の保護されたサブディレクトリに作成します。

2. アクセスするディレクトリサーバーの証明書を発行した CA に連絡し、CA 証明書を要求します。電子メールを送信するか、Web サイトにアクセスして PEM 方式で暗号化されたテキスト形式の PKCS #11 証明書を取得します。この証明書をファイルとして保存します。

   たとえば、内部に導入された Sun ONE Certificate Server を使用している場合は、`https://`*hostname*`:444` という URL にアクセスします。最上位の「Retrieval」タブで、「Import CA Certificate Chain」を選択して、符号化された証明書をコピーします。

   クライアント証明書とサーバー証明書の両方を同じ CA から取得した場合は、379 ページの「CA の信頼設定」で説明した手順で取得した CA 証明書を再利用することもできます。

3. SSL 接続に使用するサーバー証明書を発行する信頼できる CA として、CA 証明書をインポートします。次のコマンドを実行します。

```
certutil -A -n "certificateName" -t "C,," -a -i certFile -d path -P prefix
```

ここで、*certificateName* はこの証明書を識別するために指定する値です。*certFile* は、CA の PEM 方式で符号化されたテキスト形式の PKCS #11 証明書を含むテキストファイル、*path* と *prefix* は、手順 1 と同様です。

LDAP クライアントアプリケーションのすべてのユーザーは、各自の証明書データベースに CA 証明書をインポートする必要があります。すべてのユーザーが、*certFile* に保存されている同じ証明書をインポートできます。

サーバー認証の SSL オプションの設定

`ldapsearch` ツールを使って SSL を利用したサーバー認証を行う場合、ユーザーは各自の証明書データベースのパスを指定するだけで設定が完了します。セキュリティ保護されたポートを通じて SSL 接続を確立するときに、サーバーは証明書を送信します。`ldapsearch` ツールは、ユーザーの証明書データベースを検索し、サーバー証明書を発行した CA の信頼されている CA 証明書を検出します。

次のコマンドは、ユーザーが、Netscape Communicator によって作成された各自の証明書データベースを指定する方法を示しています。

```
ldapsearch -h host -p securePort ¥
           -D "uid=bjensen,dc=example,dc=com" -w bindPassword ¥
           -Z -P /home/bjensen/.netscape/cert7.db ¥
           -b "dc=example,dc=com" "(givenname=Richard)"
```

クライアントでの証明書ベースの認証の設定

クライアント認証のデフォルトのメカニズムでは、ディレクトリサーバーにアクセスするユーザーを、証明書を使って安全に識別しています。証明書ベースのクライアント認証を行うには、次の処理を行う必要があります。

- すべてのディレクトリユーザーの証明書を取得し、クライアントアプリケーションがアクセスする場所にインストールします。
- 同じ証明書のバイナリコピーを使って、ユーザーのディレクトリエントリを設定します。認証時に、サーバーはクライアントアプリケーションが提示する証明書と、このコピーを比較してユーザーを識別します。
- サーバーに証明書ベースの認証を設定します。詳細については、『Sun ONE Server Console Server Management Guide』の第 10 章にある「Using Client Authentication」を参照してください。
- 証明書ベースの認証用に、LDAP クライアントの SSL オプションを指定します。

これらの手順を実行するには、コマンド行から証明書を管理するための certutil ツールを使います。このツールは、Sun ONE Directory Server Resource Kit に用意されています。詳細については、『Sun ONE Directory Server Resource Kit Tools Reference』の第 30 章「Security Tools」を参照してください。

ユーザー証明書の取得とインストール

証明書ベースの認証を使ってディレクトリにアクセスするユーザーごとに、クライアント証明書を要求し、それをインストールする必要があります。この手順は、394 ページの「クライアントでのサーバー認証の設定」で説明した方法で、ユーザーがすでに証明書データベースを設定していることを前提としています。

1. 次のコマンドを実行して、ユーザー証明書の要求を作成します。

   ```
   certutil -R ¥
   -s "cn=Babs Jensen,ou=Sales,o=example.com,l=city,st=state,c=country"¥
   -a -d path -P prefix
   ```

   -s オプションは、要求する証明書の DN を指定します。通常、CA は証明書の所有者を完全に識別するために、この例に含まれるすべての属性を必要とします。手順 9 の証明書マッピングメカニズムによって、証明書の DN はユーザーのディレクトリ DN にマッピングされます。

   path と *prefix* は、ユーザーの証明書データベースと鍵データベースの場所を特定します。certutil ツールは、鍵データベースを保護しているパスワードを要求します。これにより、PEM で符号化されたテキスト形式の PKCS #10 証明書要求が生成されます。

2. 符号化された要求をファイルとして保存し、所定の方法でそれを CA に送信します。証明書要求は、電子メールとしての送信が求められることもありますが、CA の Web サイトに入力できる場合もあります。

3. 証明書要求を送信したら、証明書に関する CA からの回答を待つ必要があります。要求に対する回答が届くまでの時間は、状況によって異なります。たとえば、CA が社内にある場合は、要求に対する回答は 1 ～ 2 日しかかからないこともありますが、CA が社外にある場合は、数週間かかることもあります。

4. CA が応答を送信したら、新しい証明書の PEM で符号化されたテキストをダウンロードするか、テキストファイルにコピーします。また、符号化された証明書のバックアップを安全な場所に置く必要もあります。これにより、システムに保存された証明書データが失われても、バックアップファイルから証明書をインストールし直すことができます。

5. 次のコマンドを実行して、証明書データベースに新しいユーザー証明書をインストールします。

```
certutil -A -n "certificateName" -t "u,," -a -i certFile -d path -P prefix
```

ここで、*certificateName* はこの証明書を識別するために指定する値です。*certFile* は、PEM 形式の PKCS #11 証明書を含むテキストファイル、*path* と *prefix* は、手順 1 と同様です。

Netscape Communicator を使って証明書データベースを管理している場合は、証明書を直接インストールできるように、CA の Web サイトのリンクが含まれていることがあります。このリンクをクリックして、Communicator によって表示されるダイアログボックスの指示に従って処理を行います。

6. 次のコマンドを実行して、証明書のバイナリコピーを作成します。

```
certutil -L -n "certificateName" -d path -r > userCert.bin
```

ここで、*certificateName* はインストール時に指定した証明書の名前です。*path* は証明書データベースの場所、*userCert.bin* はバイナリ形式の証明書の出力先となるファイルの名前です。

7. Directory Server で、クライアント証明書を所有するユーザーのディレクトリエントリに `userCertificate` 属性を追加します。

- コンソールから証明書を追加するには、次の手順を実行します。

  a. Directory Server コンソールの最上位にある「ディレクトリ」タブで、ディレクトリツリーを表示してユーザーエントリを探し、そのエントリをマウスの右ボタンでクリックして、ポップアップメニューから「汎用エディタで編集」を選択します。

  b. 汎用エディタで「属性の追加」をクリックし、ポップアップダイアログから `userCertificate` 属性を選択します。

  c. 汎用エディタで新しい `userCertificate` フィールドを探します。対応する「値の設定」ボタンをクリックして、この属性のバイナリ値を設定します。

  d. 「値の設定」ダイアログで、手順 6 で作成した *userCert.bin* ファイルの名前を入力するか、「参照」をクリックしてファイルを選択します。

  e. 「値の設定」ダイアログの「了解」をクリックし、汎用エディタの「保存」をクリックします。

- コマンド行から証明書を追加するには、次の例のように、`ldapmodify` コマンドを実行します。このコマンドは、SSL を利用してセキュリティ保護された接続を通じて証明書を送信します。

```
ldapmodify -h host -p securePort ¥
           -D "uid=bjensen,dc=example,dc=com" -w bindPassword ¥
           -Z -P /home/bjensen/.netscape/cert7.db
version: 1
```

```
dn: uid=bjensen,dc=example,dc=com
changetype: modify
add: userCertificate
userCertificate: < file:///path/userCert.bin
```

< の前後の空白文字は重要です。ここに示されるとおりに指定する必要があります。ファイル名の指定に < 構文を利用するには、LDIF 文を `version: 1` という行から開始する必要があります。`ldapmodify` がこの文を処理するときに、このツールは、指定ファイルの内容全体から読み取った値を属性に設定します。

8. ディレクトリサーバーで、必要に応じてユーザー証明書を発行した CA の証明書をインストールし、それを信頼します。この CA は、クライアントからの接続の許可について信頼されている必要があります。詳細は、379 ページの「CA の信頼設定」を参照してください。

9. ディレクトリサーバーに証明書ベースの認証を設定します。詳細については、『Sun ONE Server Console Server Management Guide』の第 10 章にある「Using Client Authentication」を参照してください。この処理では、LDAP クライアントから提示されるユーザー証明書をサーバーが対応するユーザー DN にマッピングできるように、`certmap.conf` ファイルを編集します。

   `certmap.conf` ファイルで、`verifyCert` パラメータに `on` が設定されていることを確認します。サーバーは、ユーザーエントリに同じ証明書が含まれることを確認し、ユーザーを識別します。

証明書ベースのクライアント認証の SSL オプションの設定

`ldapsearch` ツールを使って SSL による証明書ベースのクライアント認証を行うには、証明書が利用するいくつかのコマンド行オプションを指定する必要があります。セキュリティ保護されたポートを通じて SSL 接続を確立する場合、このツールはサーバーの証明書を認証し、それからユーザー証明書をサーバーに送信します。

次のコマンドは、ユーザーが、Netscape Communicator によって作成された各自の証明書データベースへのアクセスに適用されるオプションを指定する方法を示しています。

```
ldapsearch -h host -p securePort ¥
           -Z -P /home/bjensen/.netscape/cert7.db ¥
           -N "certificateName" ¥
           -K /home/bjensen/.netscape/key3.db -W keyPassword ¥
           -b "dc=example,dc=com" "(givenname=Richard)"
```

`-Z` オプションは証明書ベースの認証を示します。*certificateName* は送信する証明書を指定します。`-K` オプションと `-W` オプションは、クライアントアプリケーションが送信のために証明書にアクセスすることを許可します。`-D` オプションと `-w` オプションを指定しない場合、バインド DN は証明書マッピングに基づいて決定されます。

クライアントでの SASL DIGEST-MD5 の使用

クライアントで DIGEST-MD5 メカニズムを使用するときは、ユーザー証明書をインストールする必要はありません。ただし、暗号化された SSL 通信を利用するには、394 ページの「クライアントでのサーバー認証の設定」で説明した方法で、サーバー証明書を信頼する必要があります。

レルムの指定

レルムは、認証 ID が選択されるネームスペースを定義します。DIGEST-MD5 認証では、特定のレルムに対して認証を行う必要があります。

Directory Server は、DIGEST-MD5 のデフォルトレルムとして、マシンの完全修飾ホスト名を使います。サーバーは、`nsslapd-localhost` 設定属性に含まれる小文字のホスト名を使用します。

レルムを指定しない場合、サーバーが提供するデフォルトのレルムが適用されます。

環境変数の指定

UNIX 環境では、LDAP ツールが DIGEST-MD5 ライブラリを見つけることができるように、`SASL_PATH` 環境変数を設定する必要があります。DIGEST-MD5 ライブラリは、SASL プラグインによってダイナミックにロードされる共有ライブラリであるため、SASL_PATH 変数を次のように設定する必要があります (Korn シェルでの例)

```
export SASL_PATH=ServerRoot/lib/sasl
```

このパスは、LDAP ツールを呼び出したホストと同じホストに Directory Server がインストールされていることを前提としています。

Windows では、SASL ライブラリのパスは次のレジストリキーに指定されます。`[HKEY_LOCAL_MACHINE¥SOFTWARE¥Carnegie Mellon¥Project Cyrus¥SASL Library¥Available Plugins]`。同じホストに Directory Server をインストールした場合は、このキーは自動的に *ServerRoot*`/lib/sasl` に設定され、変更の必要はありません。

ldapsearch コマンドの例

SSL を使用せずに DIGEST-MD5 クライアント認証を実行することができます。次の例は、デフォルトの DIGEST-MD5 ID マッピングを使用してバインド DN を決定します。

```
ldapsearch -h host -p nonSecurePort -D "" -w bindPassword ¥
           -o mech=DIGEST-MD5 [-o realm="hostFQDN"] ¥
           -o authid="dn:uid=bjensen,dc=example,dc=com" ¥
           -o authzid="dn:uid=bjensen,dc=example,dc=com" ¥
           -b "dc=example,dc=com" "(givenname=Richard)"
```

上の例は、-o (小文字の o) オプションを使って SASL オプションを指定しています。レルムの指定は省略できますが、指定する場合は、サーバーホストマシンの完全修飾ドメイン名を指定する必要があります。authzid はプロキシ操作が利用されないことを示しますが、authid と authzid には、どちらにも同じ値を指定する必要があります。

authid の値は、ID マッピングで使用される主体です。authid には、dn: に続けてディレクトリ内の有効なユーザー DN を含めるか、または u: 接頭辞に続けてクライアントが決定する任意の文字列を含めることをお勧めします。これにより、386 ページの「DIGEST-MD5 ID マッピング」に示されるマッピングを利用できるようになります。

通常は、SSL 接続を利用することで、セキュリティ保護されたポートを通じて暗号化された情報をやり取りし、DIGEST-MD5 がクライアント認証を行うことが求められます。次の例は、同じ処理を SSL 経由で実行します。

```
ldapsearch -h host -p securePort ¥
           -Z -P /home/bjensen/.netscape/cert7.db ¥
           -N "certificateName" -W keyPassword ¥
           -o mech=DIGEST-MD5 [-o realm="hostFQDN"] ¥
           -o authid="dn:uid=bjensen,dc=example,dc=com" ¥
           -o authzid="dn:uid=bjensen,dc=example,dc=com" ¥
           -b "dc=example,dc=com" "(givenname=Richard)"
```

この例では、ldapsearch コマンドに -N オプションと -W オプションを指定する必要がありますが、これはクライアント認証には使われません。その代わりに、サーバーは authid の値に含まれる主体の DIGEST-MD5 ID マッピングを行います。

クライアントでの Kerberos SASL GSSAPI の使用

クライアントで GSSAPI メカニズムを使用するときは、ユーザー証明書をインストールする必要はありません。ただし、Kerberos V5 セキュリティシステムを設定する必要があります。また、暗号化された SSL 通信を利用するには、394 ページの「クライアントでのサーバー認証の設定」で説明した方法で、サーバー証明書を信頼する必要があります。

クライアントホストでの Kerberos V5 の設定

LDAP クライアントを実行するホストマシンで Kerberos V5 を設定する必要があります。

1. インストール手順に従って Kerberos V5 をインストールします。SEAM (Sun Enterprise Authentication Mechanism) 1.0.1 クライアントソフトウェアをインストールすることをお勧めします。

2. Kerberos ソフトウェアを設定します。SEAM では、/etc/krb5 の下にあるファイルを設定して kdc サーバーを設定し、デフォルトレルムを定義して、Kerberos システムに必要なその他の設定を行います。
3. 最初の値が kerberos_v5 となるように、必要に応じて /etc/gss/mech ファイルを編集します。

Kerberos 認証の SASL オプションの設定

1. GSSAPI が有効なクライアントアプリケーションを使用する前に次のコマンドを実行し、Kerberos セキュリティシステムをユーザー主体で初期化する必要があります。

   ```
   kinit userPrincipal
   ```

   userPrincipal は、たとえば bjensen@example.com のような SASL ID です。

2. 次に示す ldapsearch ツールの例は、-o (小文字の o) オプションを使って Kerberos の使用を設定する SASL オプションを指定する方法を示しています。

   ```
   ldapsearch -h host -p securePort ¥
              -Z -P /home/bjensen/.netscape/cert7.db ¥
              -N "certificateName" -W keyPassword ¥
              -o mech=GSSAPI [-o realm="example.com" ¥
              -o authid="bjensen@example.com" ¥
              -o authzid="bjensen@example.com"] ¥
              -b "dc=example,dc=com" "(givenname=Richard)"
   ```

   この例では、ldapsearch コマンドに -N オプションと -W オプションを指定する必要がありますが、これはクライアント認証には使われません。realm、authid、authzid は、kinit コマンドによって初期化された Kerberos キャッシュに含まれるので、省略することができます。指定する場合は、authzid はプロキシ操作が利用されないことを示しますが、authid と authzid には、どちらにも同じ値を指定する必要があります。authid の値は、ID マッピングで使用される主体です。詳細は、389 ページの「GSSAPI ID マッピング」を参照してください。

第 12 章

ログファイルの管理

この章では、ログポリシーを設定し、サーバーが管理する状態情報を分析して Directory Server を監視する方法について説明します。

Sun ONE Directory Server には、次の 3 種類のログが用意されています。

- アクセスログ : サーバーに接続するクライアントのリスト
- エラーログ : サーバーエラーに関する情報
- 監査ログ : サフィックスへのアクセスと設定へのアクセスの詳細

サーバーの状態情報には、接続とキャッシュ アクティビティに関する統計が含まれます。この情報には、Directory Server コンソールから得られるものと、LDAP コマンド行ツールを使って監視エントリから得られるものがあります。SNMP を使用してサーバーを監視する方法については、第 13 章「SNMP を使用した Directory Server の監視」を参照してください。

この章は、次の節で構成されています。

- ログファイルポリシーの定義
- アクセスログ
- エラーログ
- 監査ログ
- サーバーアクティビティの監視

ログファイルポリシーの定義

ここでは、ログファイルの作成ポリシーと削除ポリシーを定義する方法について説明します。

ログファイルのローテーションポリシーの定義

ディレクトリの最新ログを定期的にアーカイブして、新しいログへの記録を開始する場合は、Directory Server コンソールを使用してログファイルのローテーションポリシーを定義できます。次のパラメータを設定します。

- ディレクトリに保持するログの総数。ディレクトリ内のログがこの数に達すると、新しいログを作成する前に、フォルダ内のもっとも古いログが削除される。デフォルトは 10。この値に 1 を設定してはならい。設定した場合はログのローテーションが行われず、ログのサイズが無制限に大きくなる。
- 各ログファイルの最大サイズ (M バイト)。最大サイズを設定しない場合は、このフィールドに -1 を入力する。デフォルトは 100 M バイト。ログファイルがこの最大サイズ (あるいは次の手順で定義する最大維持期間) に達すると、そのファイルがアーカイブされ、新しいファイルへの記録が開始される。ログの最大数を 1 に設定すると、この属性は無視される
- 現在のログファイルをアーカイブして、新しいログへの記録を開始する間隔。分、時間、日、週、または月単位で指定する。デフォルトでは、「毎日」に設定されている。ログの最大数を 1 に設定すると、この属性は無視される

ログファイルの削除ポリシーの定義

アーカイブ済みの古いログを自動的に削除する場合には、Directory Server コンソールを使用してログファイル削除ポリシーを定義します。

注	ログファイルのローテーションポリシーが事前に定義されていないと、ログ削除ポリシーを定義しても意味がありません。ログファイルが 1 つしかない場合は、ログファイル削除ポリシーは機能しないからです。 ログのローテーション時に、ログファイル削除ポリシーがサーバーによって評価、適用されます。

次のパラメータを設定します。

- アーカイブされたログの最大合計サイズ。最大サイズに達すると、アーカイブ済みのもっとも古いログが自動的に削除される。最大サイズを設定しない場合は、このフィールドに -1 を入力する。デフォルトは 500 M バイト。ログファイル数が 1 に設定されていると、このパラメータは無視される
- ディスクの最小空き容量。ディスクの空き容量が最小値に達すると、アーカイブ済みのもっとも古いログが自動的に削除される。デフォルトは 5 M バイト。ログファイル数が 1 に設定されていると、このパラメータは無視される
- ログファイルの最大維持期間。ログファイルが作成されてからこの期間が経過すると、ファイルは自動的に削除される。デフォルトは 1 か月。ログファイル数が 1 に設定されていると、このパラメータは無視される

手動によるログファイルのローテーション

ログファイルの自動作成ポリシーや自動削除ポリシーを設定しなかった場合は、手動でログファイルをローテーションさせることもできます。デフォルトでは、アクセスログ、エラーログ、監査ログファイルは、次のディレクトリに置かれます。

ServerRoot/slapd-*serverID*/logs

手動でログファイルをローテーションさせるには、次の手順を実行します。

1. サーバーを停止します。手順については、20 ページの「Directory Server の起動と停止」を参照してください。
2. 古いログファイルをあとで参照できるように、ローテーションさせるログファイルを移動するか、ファイル名を変更します。
3. サーバーを再起動します。手順については、20 ページの「Directory Server の起動と停止」を参照してください。

   サーバーは、各ログ設定に基づいて新規ファイルを自動的に作成します。

アクセスログ

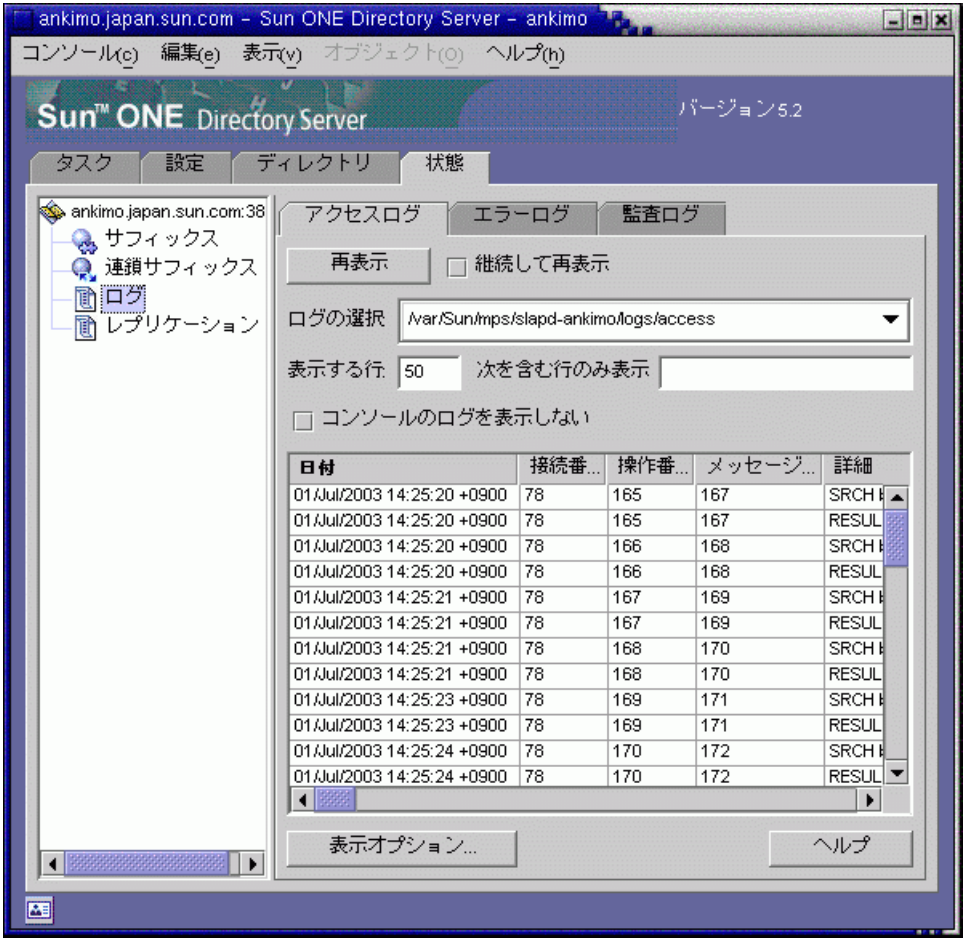
アクセスログには、ディレクトリへのクライアントの接続に関する詳しい情報が記録されます。

アクセスログの表示

1. Directory Server コンソールの最上位の「状態」タブで「ログ」アイコンを選び、右側のパネルで「アクセスログ」タブを選びます。

   次の図に示すように、このタブには、選択しているアクセスログの最新エントリの内容が表形式で表示されます。アクセスメッセージの詳細については、『Sun ONE Directory Server Reference Manual』の第 8 章「Access Logs and Connection Codes」を参照してください。

図 12-1　ログの内容表示



2. 表示を更新するには、「再表示」をクリックします。「連続して再表示」チェックボックスを選択すると、10 秒ごとに自動的に表示が更新されます。
3. 別のアクセスログを表示するには、「ログの選択」ドロップダウンメニューからログを選択します。
4. 表示するメッセージの数を指定するには、表示する数を「表示する行」テキストボックスに入力して、「再表示」をクリックします。
5. ログメッセージをフィルタリングするには、「次を含む行のみ表示」テキストボックスに文字列を入力して、「再表示」をクリックします。また、「コンソールのログを表示しない」チェックボックスを選択して、コンソールからサーバーへの接続によって発生するメッセージを表示対象から外すことができます。
6. ログエントリの表の列を変更するときは、「表示オプション」をクリックします。「表示オプション」ダイアログのコントロールを使用して、列の順序変更、列の追加・削除、表のソートに使用する列の選択を実行できます。

アクセスログの設定

アクセスログは、格納場所、作成ポリシーまたは削除ポリシーなど、さまざまな項目を設定することによってカスタマイズできます。

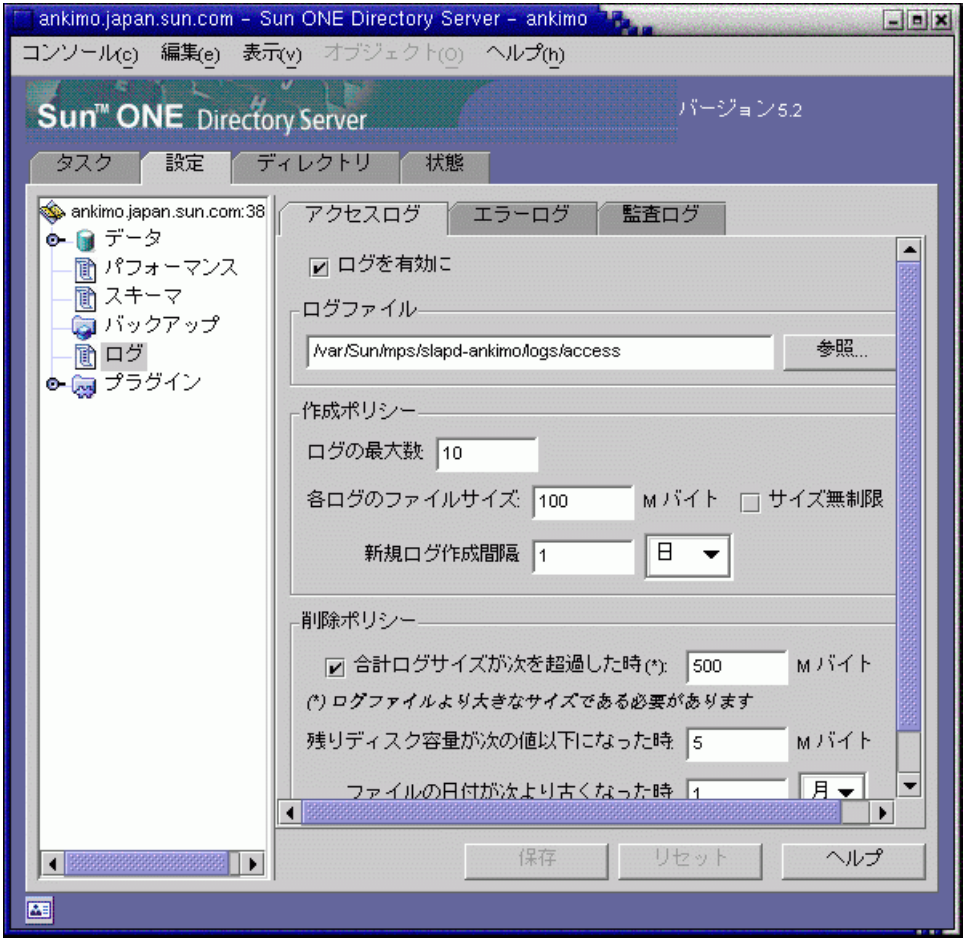
また、ディレクトリのアクセスログ機能を無効にすることもできます。アクセスログはすぐに大きくなるので、この設定が必要になることもあります。ディレクトリへのアクセスが 2000 回に達するごとに、アクセスログは約 1 M バイトずつ大きくなります。ただし、アクセスログにはトラブルシューティングに関する有益な情報が記録されるので、アクセスログをオフにする前に、この点を十分に考慮してください。

ディレクトリのアクセスログを設定するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで「ログ」アイコンを選び、右側のパネルで「アクセスログ」タブを選びます。

次の図に示すように、このタブにはアクセスログの設定が表示されます。

図 12-2　ログファイルの作成と削除の設定パネル



2. アクセスログを有効にするには、「ログを有効に」チェックボックスを選択します。

   アクセスログを使用しない場合は、このチェックボックスの選択を解除します。

   アクセスログはデフォルトで有効に設定されています。

3. 「ログファイル」フィールドに、そのディレクトリのアクセスログの絶対パスとファイル名を入力します。デフォルトでは、次のファイルが使用されます。

   ServerRoot/slapd-*serverID*/logs/access

4. ログの最大数、ログサイズ、およびアーカイブ間隔を設定します。

   これらのパラメータについては、404 ページの「ログファイルのローテーションポリシーの定義」を参照してください。

5. アーカイブされたログの最大合計サイズ、ディスクの最小空き容量、およびログファイルの最大維持期間を設定します。

   これらのパラメータについては、404 ページの「ログファイルの削除ポリシーの定義」を参照してください。

6. 変更が完了したら、「保存」をクリックします。

エラーログ

エラーログには、エラーの詳細メッセージと、通常の操作中にディレクトリに発生したイベントが記録されます。

エラーログの表示

1. Directory Server コンソールの最上位の「状態」タブで「ログ」アイコンを選び、右側のパネルで「エラーログ」タブを選びます。

   406 ページの図 12-1 と同様に、このタブには、選択しているエラーログの最新エントリの内容が表形式で表示されます。エラーメッセージの説明については、『Sun ONE Directory Server Reference Manual』の付録 A「Error Codes」を参照してください。

2. 表示を更新するには、「再表示」をクリックします。「連続して再表示」チェックボックスを選択すると、10 秒ごとに自動的に表示が更新されます。

3. アーカイブ済みのエラーログを表示するには、「ログの選択」プルダウンメニューからログを選択します。

4. 表示するメッセージの数を指定するには、表示する数を「表示する行」テキストボックスに入力して、「再表示」をクリックします。

5. ログメッセージをフィルタリングするには、「次を含む行のみ表示」テキストボックスに文字列を入力して、「再表示」をクリックします。また、「コンソールのログを表示しない」チェックボックスを選択して、コンソールからサーバーへの接続によって発生するメッセージを表示対象から外すことができます。

6. ログエントリの表の列を変更するときは、「表示オプション」をクリックします。「表示オプション」ダイアログのコントロールを使用して、列の順序変更、列の追加・削除、表のソートに使用する列の選択を実行できます。

エラーログの設定

ログの格納場所やログに記録する内容など、エラーログのいくつかの設定は変更できます。

エラーログを設定するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで「ログ」アイコンを選び、右側のパネルで「エラーログ」タブを選びます。

   408 ページの図 12-2 と同様に、このタブにはエラーログの設定が表示されます。

2. エラーログを有効にするには、「ログを有効に」チェックボックスを選択します。

   エラーログを使用しない場合は、このチェックボックスの選択を解除します。

   エラーログはデフォルトで有効に設定されています。

3. エラーログの詳細度を設定するには、「ログレベル」ボタンをクリックして、「エラーログレベル」ダイアログを開きます。エラーとデバッグについてより多くの情報を必要とする、製品の内部コンポーネントを 1 つまたは複数選択します。オプションとして、「冗長モード」チェックボックスを選択すると、些細なメッセージも含め、最大量の実行時出力が得られます。

   これらの設定をデフォルトから変更すると、ログが急激に大きくなることがあるため、十分なディスクスペースを用意する必要があります。Sun ONE カスタマサポートから指示のない限り、ログレベルを変更しないことをお勧めします。

4. 「ログファイル」フィールドに、そのディレクトリのエラーログの絶対パスとファイル名を入力します。デフォルトでは、次のファイルが使用されます。

   ServerRoot/slapd-*serverID*/logs/error

5. ログの最大数、ログサイズ、およびアーカイブ間隔を設定します。

   これらのパラメータについては、404 ページの「ログファイルのローテーションポリシーの定義」を参照してください。

6. アーカイブされたログの最大合計サイズ、ディスクの最小空き容量、およびログファイルの最大維持期間を設定します。

   これらのパラメータについては、404 ページの「ログファイルの削除ポリシーの定義」を参照してください。

7. 変更が完了したら、「保存」をクリックします。

監査ログ

監査ログには、サーバーの設定だけでなく、各サフィックスに対する変更に関する詳細情報が記録されます。アクセスログやエラーログとは異なり、監査ログはデフォルトでは無効化されています。ログを表示するには、最初にログを有効化する必要があります。

監査ログの設定

監査ログ機能の有効または無効の設定や、監査ログファイルの格納場所の指定は、Directory Server コンソールを使用して行います。

監査ログを設定するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで「ログ」アイコンを選び、右側のパネルで「監査ログ」タブを選びます。

   408 ページの図 12-2 と同様に、このタブには監査ログの設定が表示されます。

2. 監査ログを有効にするには、「ログを有効に」チェックボックスを選択します。

   監査ログを無効にするには、このチェックボックスの選択を解除します。監査ログは、デフォルトで無効に設定されています。

3. 「ログファイル」フィールドに、そのディレクトリの監査ログの絶対パスとファイル名を入力します。デフォルトでは、次のファイルが使用されます。

   ServerRoot/slapd-*serverID*/logs/audit

4. ログの最大数、ログサイズ、およびアーカイブ間隔を設定します。

   これらのパラメータについては、404 ページの「ログファイルのローテーションポリシーの定義」を参照してください。

5. アーカイブされたログの最大合計サイズ、ディスクの最小空き容量、およびログファイルの最大維持期間を設定します。

   これらのパラメータについては、404 ページの「ログファイルの削除ポリシーの定義」を参照してください。

6. 変更が完了したら、「保存」をクリックします。

監査ログの表示

1. Directory Server コンソールの最上位の「状態」タブで「ログ」アイコンを選び、右側のパネルで「監査ログ」タブを選びます。

   406 ページの図 12-1 と同様に、このタブには、選択している監査ログの最新エントリの内容が表形式で表示されます。

2. 表示を更新するには、「再表示」をクリックします。「連続して再表示」チェックボックスを選択すると、10 秒ごとに自動的に表示が更新されます。

3. アーカイブ済みの監査ログを表示するには、「ログの選択」プルダウンメニューからログを選択します。
4. 表示するメッセージの数を指定するには、表示する数を「表示する行」テキストボックスに入力して、「再表示」をクリックします。
5. ログメッセージをフィルタリングするには、「次を含む行のみ表示」テキストボックスに文字列を入力して、「再表示」をクリックします。

サーバーアクティビティの監視

接続数や操作の回数、すべてのサフィックスのキャッシュアクティビティなど、サーバーはアクティビティのカウンタと統計を常に維持しています。この情報は、エラーのトラブルシューティングや、サーバーパフォーマンスの監視に役立ちます。Directory サーバーコンソールまたはコマンド行から、Directory Server の現在のアクティビティを監視できます。

監視できるパラメータの多くは、Directory Server のパフォーマンスを反映するので、設定や調整によって影響を受けることがあります。設定可能な属性とその調整方法については、『Sun ONE Directory Server インストールおよびチューニングガイド』を参照してください。

コンソールを使用したサーバの監視

1. Directory Server コンソールの最上位にある「状態」タブで、状態ツリーのルートにあるサーバーのアイコンを選択します。

   右側のパネルには、サーバーアクティビティの現在の状態が示されます。サーバーが実行されていない場合は、このタブにはパフォーマンス監視情報は表示されません。
2. 「再表示」をクリックすると、現在の表示が更新されます。表示される情報を継続して更新するには、「連続して再表示」チェックボックスを選択します。

   このサーバー状態パネルには、次の情報が表示されます。

- サーバーが起動した日付と時刻
- サーバー上の現在の日付と時刻。レプリケーションを有効化するときは、各サーバーで日時に違いが生じないように、定期的に確認する必要がある
- リソースの概要テーブル。この表には、起動以来の総数と、1 分あたりの平均数が次のリソースごとに示される

表 12-1 リソースの概要テーブル

リソース	起動以来の総数と 1 分あたりの平均数
接続回数	確立されたクライアント接続の数
開始した処理	クライアントが要求した処理の数
完了した処理	クライアントが中断しなかった処理の数
クライアントに送信されたエントリ数	検索結果として返されたエントリの数
クライアントに送信されたバイト数 (K バイト)	クライアント要求へのすべての応答の合計バイト数

- 現在のリソース使用状況テーブル。この表には、前回の再表示時に使用されていた次のリソースが表示される

表 12-2 現在のリソース使用状況

リソース	最新のリアルタイム使用状況
有効なスレッド数	要求の処理に使われているスレッドの数。レプリケーションや連鎖などのサーバーの内部メカニズムによって、追加のスレッドが生成されることもある
開いている接続数	各接続は複数の操作、つまり複数のスレッドを扱うことができる
利用可能な接続数	同時に接続できる残りの接続の合計数が表示される。この数は、現在開いている接続の数と、サーバーに許可される最大接続数に基づいて決められる。ほとんどの場合、サーバーに許可される接続数はオペレーティングシステムによって決まり、タスクに割り当てることのできるファイル記述子の数で示される Windows と AIX では、可能な同時接続の数はオペレーティングシステムによって決まる。ファイル記述子に基づくものではない。詳細は、オペレーティングシステムのマニュアルを参照
クライアントから読み取り待機中のスレッド数	サーバーがクライアントからの要求受信を開始したあとに、その要求の送信が何らかの理由で中断された場合は、スレッドは読み出しを待機することがある。一般に、スレッドの読み取り待機状態は、ネットワークの速度やクライアントの処理速度が遅いことを示す
使用中のデータベース	このサーバーがホストするサフィックスの数。この値には連鎖サフィックスは含まれない

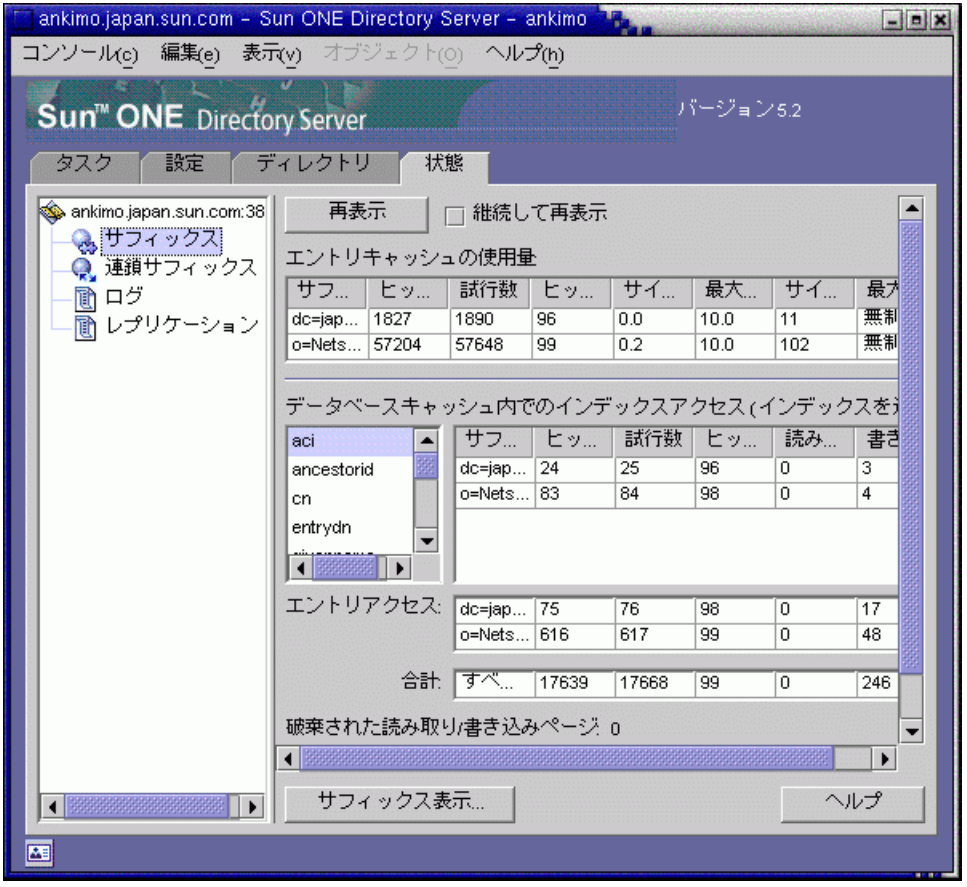
- 接続状態テーブル。この表には、現在開いている接続のそれぞれについて、次の情報が表示される。

表 12-3 接続状態テーブル

列の見出し	内容
開いた時刻	接続が確立した時点のサーバー側時刻
初期化済み	この接続中に要求された処理の数
完了済み	この接続中にクライアントが中断せずに、サーバーが完了した処理の数
バインド DN	クライアントがサーバーへのバインド処理に使用した識別名。クライアントがサーバーに対して認証していない場合は、この列には「(バインドなし)」と表示される
状態	• ブロックされない : サーバーがアイドル状態にある、または接続を通じてデータをアクティブに送信または受信している最中である • ブロックされた : サーバーが接続を通じたデータの読み取りまたは書き込みを待機している。原因としては、ネットワークまたはクライアントが低速であることが考えられる
種別	接続が LDAP であるか、または DSML-over-HTTP であるかを示す

3. 左側の状態ツリーで「サフィックス」ノードをクリックします。次の図に示すように、このパネルには、各サフィックスのデータベースキャッシュのエントリキャッシュとインデックスの使用状況に関する監視情報が表示されます。

図 12-3　サフィックス監視パネル



必要に応じて再表示のモードを設定します。表に表示するサフィックスを選択するときは、パネルの下部にある「サフィックス表示」をクリックします。

- 最初の表には、各エントリキャッシュに関する次の情報が表示されます。

表 12-4　エントリキャッシュの使用状況

列の見出し	内容
サフィックス	サフィックスのベース DN
ヒット数	ディスクではなくキャッシュから読み取られたエントリの数
試行数	キャッシュから要求されたエントリの数
ヒット率 (%)	試行数に対するヒット数の割合 (%)
サイズ (M バイト)	指定のサフィックスのエントリキャッシュの現在の内容のサイズ
最大サイズ (M バイト)	現在の設定のキャッシュの最大サイズ
サイズ (エントリ)	指定されたサフィックスのキャッシュに含まれる現在のエントリの数
最大サイズ (エントリ)	現在の設定のキャッシュエントリの最大数

次の一連のテーブルには、各サフィックスのデータベースキャッシュへのアクセスが示されます。

- 最初の表は、設定されているインデックスを経由したデータベースキャッシュへのアクセスを示す。属性名のリストから、インデックスの統計を表示する属性を選択する。テーブルには、選択した属性がインデックスされているサフィックスのデータだけが表示される
- エントリアクセステーブルは、エントリ検索のためのデータベースキャッシュへのアクセスを示す
- 最後のテーブルの「合計」は、すべてのデータベースキャッシュの総アクセス数を示す

これら 3 種類のすべてのテーブルには、次の見出しがあります。

表 12-5　データベースキャッシュへのアクセス

列の見出し	内容
サフィックス	サフィックスのベース DN
ヒット数	インデックス経由で読み取られたエントリの数
試行数	インデックス経由で要求されたエントリの数
ヒット率 (%)	試行数に対するヒット数の割合 (%)
読み取りページ	ディスクからサフィックスキャッシュに読み取られたページの数

表 12-5　データベースキャッシュへのアクセス (続き)

列の見出し	内容
書き込みページ	キャッシュからディスクに書き込まれたページの数。読み書き可能ページに変更が加えられると、新しいページの領域を確保するために、サフィックスページはディスクに書き込まれ、キャッシュから削除される。

- テーブルの下の次の破棄ページ数は、すべてのデータベースキャッシュの累計を示す。ページがキャッシュから破棄されると、サーバーのパフォーマンスに影響することがあるので、そのページをディスクに書き込む必要がある。この数値が小さいほど、パフォーマンスは高くなる
  - 新しいページ用のスペースを確保するためにキャッシュから破棄された読み取り / 書き込みページの数を示す。この値は、変更されていない読み取りおよび書き込みページであるという点で、「書き込みページ」の値とは異なる
  - 破棄された読み取り専用ページ : 新しいページ用のスペースを確保するためにキャッシュから破棄された読み取り専用ページの数を示す

4. 必要に応じて、左側の状態ツリーで「連鎖サフィックス」ノードをクリックします。このパネルには、ディレクトリに設定されている連鎖サフィックスへのアクセスに関する情報が表示されます。必要に応じて再表示のモードを設定します。

   統計を表示する連鎖サフィックスの DN をリストから選択します。右側の表には、連鎖サフィックスで実行されたすべての種類の処理の回数が表示されます。

コマンド行からのサーバーの監視

次のエントリで検索操作を実行することによって、任意の LDAP クライアントから、Directory Server の現在のアクティビティを監視できます。

- `cn=monitor`
- `cn=monitor, cn=ldbm database, cn=plugins, cn=config`
- `cn=monitor,cn=`*`dbName`*`,cn=ldbm database,cn=plugins,cn=config`
- `cn=monitor,cn=`*`dbName`*`,cn=chaining database,cn=plugins,cn=config`

dbName は、監視するサフィックスのデータベース名です。匿名でバインドされているクライアントを含め、デフォルトではすべてのユーザーが各接続に関する情報を除き `cn=monitor` エントリを読み取れることに注意してください。

次の例は、サーバーの一般的な統計情報を表示する方法を示しています。

```
ldapsearch -h host -p port -D "cn=Directory Manager" -w password ¥
           -s base -b "cn=monitor" "(objectclass=*)"
```

これらのエントリで使用できるすべての監視属性の説明については、次に示す『Sun ONE Directory Server Reference Manual』の対応する項を参照してください。

- 第 4 章の「Monitoring Attributes」
- 第 5 章の「Database Monitoring Attributes」
- 第 5 章の「Database Monitoring Attributes under cn=*dbName*」
- 第 5 章の「Chained Suffix Monitoring Attributes」

第 13 章

SNMP を使用した Directory Server の監視

SNMP (Simple Network Management Protocol) は、リアルタイムでデバイスやアプリケーションを監視および管理するための標準の管理プロトコルです。Directory Server には、SNMP マネージャアプリケーションを使って監視するためのサブエージェントインタフェースが用意されています。これを利用することで、ディレクトリサーバーの状態をネットワークアプリケーションに知らせ、アクティビティに関する情報を取得できます。

しかし、Directory Server の SNMP サブエージェントに含まれる値は読み取り専用であるため、SNMP 管理アプリケーションはサーバーに対して処理を実行できません。また、サブエージェントはイベントをレポートするメッセージである SNMP トラップを送信できません。

一般に、第 12 章「ログファイルの管理」で説明したアクティビティログとエラーログにはサーバーに関するより詳細な情報が含まれており、サーバーの設定に安全にアクセスし、それを変更するには、プロトコルとして LDAP を選択するべきです。それでも Directory Server インスタンスは、SNMP サブエージェントを使って既存のネットワーク管理システムに参加できます。

この章では、次の項目について説明します。

- Sun ONE サーバーでの SNMP
- Directory Server MIB の概要
- SNMP の設定
- Directory Server 側の SNMP の設定
- SNMP サブエージェントの起動と停止

Sun ONE サーバーでの SNMP

SNMP を利用する管理アプリケーションは、エージェントまたはサブエージェントアプリケーションを実行するアプリケーションおよびデバイスに照会を行います。SNMP エージェントまたはサブエージェントは、SNMP マネージャからの照会への応答として、アプリケーションまたはデバイスからの情報を収集します。この情報は変数のテーブルとして構築され、このテーブルはエージェントの MIB (管理情報ベース) によって定義されます。

通常、ネットワーク管理者はサブエージェント内の SNMP 変数を照会し、サブエージェントは要求された値を返します。また SNMP は、すべてのネットワーク管理者にトラップメッセージを送信してエージェントがイベントをレポートするためのメカニズムを定義します。しかし、Directory Server はトラップを実装しておらず、そのサブエージェントもトラップメッセージを送信することはありません。

ホストマシンには、複数のサブエージェントをインストールできます。たとえば、Directory Server、Enterprise Server、および Messaging Server をすべて同じホスト上にインストールした場合、これらの各サーバーのサブエージェントは、同一のマスターエージェントと通信します。Windows 環境では、マスターエージェントは Windows オペレーティングシステムによって提供される SNMP サービスです。UNIX 環境では、マスターエージェントは Sun ONE 管理サーバーと一緒にインストールされます。

詳細については、『Sun ONE Server Console Server Management Guide』の第 11 章「Using SNMP to Monitor Servers」を参照してください。

SNMP 経由で監視できるようにサーバーを設定する一般的な手順は、次のとおりです。

1. Directory Server MIB をコンパイルし、SNMP 管理システムに統合します。使用しているシステムのマニュアルを参照してください。
2. 使用マシンに SNMP を設定し、プラットフォームに合わせて管理サーバーコンソールから SNMP マスターエージェントを設定、起動します。
3. Directory Server コンソールから SNMP サブエージェントを設定します。
4. プラットフォームによっては、Directory Server コンソールから SNMP サブエージェントを起動します。
5. MIB によって定義され、エージェントにより公開される SNMP 管理対象オブジェクトにアクセスします。この手順は、全体的に SNMP 管理システムに依存します。

Directory Server の設定に固有の手順については後述します。

Directory Server MIB の概要

Directory Server の MIB には、次のようなオブジェクト識別子があります。

```
iso.org.dod.internet.private.enterprises.netscape.nsldap
(nsldapd OBJECT IDENTIFIER ::= { 1.3.6.1.4.1.1450.7 })
```

これは、次のファイルに定義されています。

ServerRoot/plugins/snmp/netscape-ldap.mib

MIB は、SNMP 経由で監視できる変数と、各変数に含まれる値のタイプを定義します。Directory Server の MIB は、次の 4 つの管理対象オブジェクトテーブルに分類されます。

- Operations テーブル : ディレクトリサーバーのバインド、処理、リフェラル、エラーに関する統計情報を含みます。これらの変数の値は、ディレクトリの cn=snmp,cn=monitor エントリの属性にも指定されます。『Sun ONE Directory Server Reference Manual』の第 4 章にある「Monitoring Attributes」を参照してください。
- Entries テーブル : ディレクトリ内のエントリ数とエントリキャッシュのヒット数を含みます。これらの変数の値は、ディレクトリの cn=snmp,cn=monitor エントリの属性の操作変数とも一緒に使用されます。『Sun ONE Directory Server Reference Manual』の第 4 章にある「Monitoring Attributes」を参照してください。
- Interation テーブル : このディレクトリサーバーが最後に通信した 5 つのディレクトリサーバーに関する統計情報を含みます。このテーブルのオブジェクトについては、『Sun ONE Directory Server Deployment Guide』の第 8 章にある「SNMP Monitoring」を参照してください。
- Entity テーブル : サーバー ID やバージョンなど、Directory Server のこのインスタンスを説明する変数を含みます。このテーブルのオブジェクトについては、『Sun ONE Directory Server Deployment Guide』の第 8 章にある「SNMP Monitoring」を参照してください。

Directory Server MIB を使うときは、事前に次のディレクトリにある MIB とともにコンパイルする必要があります。

ServerRoot/plugins/snmp/mibs

MIB のコンパイル方法については、SNMP 製品のマニュアルを参照してください。

SNMP の設定

ディレクトリに SNMP 監視を設定する手順は、ホストプラットフォームの種類 (UNIX、AIX、Windows) によって異なります。

1. 次の各項で説明する方法で、使用プラットフォームに SNMP を設定してください。
   - 422 ページの「UNIX プラットフォームでの操作」
   - 423 ページの「AIX プラットフォームでの操作」
   - 423 ページの「Windows プラットフォームでの操作」
2. 424 ページの「Directory Server 側の SNMP の設定」で説明している手順を実行します。
3. 425 ページの「SNMP サブエージェントの起動と停止」で説明している手順を実行して SNMP を再起動します。

UNIX プラットフォームでの操作

AIX 以外の UNIX マシンに Directory Server の SNMP サポートを設定するには、管理サーバーコンソールを使ってマスターエージェントを設定、起動する必要があります。

デフォルトのポート設定 (SNMP は 161) では、管理サーバーと Directory Server を root ユーザーとして実行する必要があります。マスターエージェントを設定し直して 1000 より大きい番号のポートを使用している場合は、root ユーザーとして実行する必要はありません。

デフォルトでは、マスターエージェントはポート 161 を使用します。これは、ほとんどのプラットフォームのネイティブ SNMP エージェントのデフォルトポートと競合します。マスターエージェントを起動する前にネイティブ SNMP エージェントを無効化するか、マスターエージェントが別のポートを使用するように設定します。ネイティブ SNMP エージェントを無効化する方法については、使用プラットフォームのマニュアルを参照してください。マスターエージェントを設定、起動する方法については、『Sun ONE Server Console Server Management Guide』の第 11 章にある「Configuring the Master Agent on UNIX Systems」を参照してください。

AIX プラットフォームでの操作

AIX プラットフォームでは、マスターエージェントを設定する必要はありません。AIX 上で SNMP デーモンが稼動していれば、それが SMUX をサポートし、SMUX がマスターエージェントの代わりに機能します。ただし、AIX SNMP デーモンの設定は変更する必要があります。

デフォルトのポート設定 (SMUX は 199) では、管理サーバーと Directory Server を root ユーザーとして実行する必要があります。マスターエージェントを設定し直して 1000 より大きい番号のポートを使用している場合は、root ユーザーとして実行する必要はありません。

AIX は、複数の設定ファイルを使用して通信をフィルタリングします。この設定ファイルのうち、`snmpd.conf` は、SMUX サブエージェントから送られるメッセージを SNMP デーモンが受け入れるように変更する必要があります。詳細は、オンラインマニュアルの `snmpd.conf` に関するページを参照してください。各サブエージェントに対応する行を追加して、メッセージを受け入れるサブエージェントを定義します。

たとえば、次の行を `snmpd.conf` に追加します。

```
smux 1.3.6.1.4.1.1.1450.7 "" IP_address net_mask
```

ここで、*IP_address* はサブエージェントが動作しているホストの IP アドレスを表し、*net_mask* はホストのネットワークマスクを表します。

注 ループバックアドレスとして 127.0.0.1 を使うことはできません。常にホストの実際の IP アドレスを使用してください。

詳細は、AIX プラットフォームのマニュアルを参照してください。

Windows プラットフォームでの操作

ほかのプラットフォームの場合と同様に、Windows 上のマスターエージェントは SNMP サービスであり、SNMP エージェントではない点に注意してください。SNMP サービスは、Windows レジストリに格納されている情報を使って DLL を呼び出し、ディレクトリサーバー上の監視情報にアクセスします。

Windows マシンに Directory Server の SNMP サポートを設定するには、まず、Windows のコントロールパネルから SNMP サービスをインストール、設定する必要があります。インストールの手順については、Windows オペレーティングシステムのマニュアルを参照してください。

Directory Server 側の SNMP の設定

プラットフォーム側の SNMP エージェントまたはサービスの設定が完了したら、Directory Server インスタンス側の SNMP パラメータを設定する必要があります。Directory Server コンソールから SNMP 設定を設定するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで、設定ツリーのルートノードを選択し、右側のパネルで「SNMP」タブを選択します。
2. 「統計収集を有効に」チェックボックスを選択します。リソースの利用率を改善するために、デフォルトでは SNMP 変数の統計情報は収集されません。SNMP を使用せず、LDAP 経由で `cn=snmp,cn=monitor` エントリの属性を監視しないときは、このチェックボックスの選択は解除した状態で残します。
3. UNIX サーバーでは、マスターエージェントのホスト名とポート番号を対応するテキストフィールドに入力します。

   デフォルトの設定は、それぞれ `localhost` とポート `199` です。
4. 「説明のプロパティ」ボックスのテキストフィールドに情報を入力します。ここに指定した値は、このサーバーが開示する SNMP Entity テーブルに反映されます。
   - 説明 : ディレクトリサーバーの説明を入力します。Sun ONE サーバーコンソールのトポロジツリーにある、このインスタンスの説明フィールドに似ています。
   - 組織 : ディレクトリサーバーが所属する会社または部署の名前を入力します。
   - 場所 : ディレクトリサーバーのホストが設置されている場所の住所を入力します。
   - 連絡先 : ディレクトリサーバー管理者の電子メールアドレスまたは連絡先情報を入力します。
5. 「保存」をクリックして、変更内容を保存します。
6. UNIX プラットフォームでは SNMP サブエージェント、Windows プラットフォームでは SNMP サービスを起動または再起動します。方法については、次の項で説明します。

SNMP サブエージェントの起動と停止

次に、AIX を含む UNIX プラットフォームで SNMP サブエージェントを、Windows プラットフォームで SNMP サービスを起動、再起動、停止する方法について説明します。

注 同じホストに別のサーバーインスタンスを追加し、そのインスタンスを SNMP ネットワークの一部として使用するときは、SNMP サブエージェント (UNIX、AIX 環境) または SNMP サービス (Windows 環境) を再起動する必要があります。

UNIX および AIX プラットフォームでの操作

UNIX 上で動作している Directory Server の SNMP サブエージェントを起動、停止、および再起動するには、次の手順を実行します。

1. Directory Server コンソールの最上位の「設定」タブで、設定ツリーのルートノードを選択し、右側のパネルで「SNMP」タブを選択します。
2. サブエージェントを起動、停止、または再起動するには、「説明のプロパティ」ボックスの下にあるサブエージェント制御ボタンを使用します。

   Directory Server を停止しても、Directory Server のサブエージェントは停止しません。サブエージェントを停止するには、「SNMP」タブで停止する必要があります。

Windows プラットフォームでの操作

Windows 上で動作している Directory Server の SNMP サービスを起動、停止、および再起動するには、次の手順を実行します。

1. Windows の「コントロールパネル」を開き、「サービス」を選択します。
2. サービスのリストから「SNMP」を選択します。
3. SNMP サービスを起動するには「開始」、停止するには「停止」、再起動するには「続行」をクリックします。

   Directory Server を停止しても、Windows SNMP サービスは停止されません。停止するには、コントロールパネルから明示的に指定する必要があります。

第 14 章

パススルー認証プラグインの使用

PTA (パススルー認証) は、1 つのディレクトリサーバーが別のディレクトリサーバーに問い合わせて、バインド要求を認証するメカニズムです。この機能は PTA プラグインによって提供されます。この機能によって、ローカルサフィックスに格納されていないエントリに対するパスワードに基づく単純なバインド操作を、ディレクトリサーバーで受け入れることができるようになります。

Sun ONE Directory Server 5.2 で PTA を使用することによって、管理者は、Directory Server の別のインスタンス上のユーザディレクトリと設定ディレクトリを管理できます。

注 ユーザーディレクトリと設定ディレクトリを同じサーバー上に置いた場合は、Directory Server コンソール内に PTA プラグインは一覧表示されません。ただし、プラグインを作成して、パススルー認証を利用することはできます。

この章では、PTA プラグインについて、次の項目ごとに説明します。

- Directory Server での PTA の使用
- PTA プラグインの設定

Directory Server での PTA の使用

設定ディレクトリとユーザーディレクトリを Directory Server の別のインスタンスにインストールした場合は、構成管理者 (通常は admin) が管理業務を実行できるように、インストールプログラムによって自動的に PTA が設定されます。

このような場合に PTA が必要になるのは、admin ユーザーのエントリが設定ディレクトリ内の o=NetscapeRoot の下に格納されるためです。このため、admin としてユーザーディレクトリにバインドしようとしても、通常は失敗します。PTA を使用すると、ユーザーディレクトリが、資格情報を設定ディレクトリに転送できるようになります。続けて、設定ディレクトリで、資格が検証されます。検証が完了すると、ユーザーディレクトリは、admin ユーザーによるバインドを許可します。

この例のユーザーディレクトリは、PTA サーバー として機能します。つまり、バインド要求をほかのディレクトリサーバーにパススルーするサーバーです。設定ディレクトリは、認証サーバーとして機能します。つまり、エントリを格納し、要求元クライアントのバインド資格を検証するサーバーです。

この章では、PTA サブツリーという用語も使用します。パススルーサブツリーは、PTA サーバー上に存在しないサブツリーです。ユーザーのバインド DN にこのサブツリーが含まれている場合は、ユーザーの資格情報が認証ディレクトリに渡されます。

パススルー認証は、次のような流れで機能します。

1. パススルーサブツリー o=NetscapeRoot を含む設定ディレクトリサーバー (認証ディレクトリ) を configdir.example.com ホストにインストールします。
2. dc=example,dc=com サフィックスにデータを含むユーザーディレクトリサーバー (PTA ディレクトリ) を userdir.example.com ホストにインストールします。
3. ユーザーディレクトリのインストール時に、たとえば次のような設定ディレクトリサーバーの場所を示す LDAP URL の指定が要求されます。

   ```
   ldap://configdir.example.com/o=NetscapeRoot
   ```

4. インストールプログラムは、指定された LDAP URL を使ってユーザーディレクトリ内で PTA プラグインを設定し、有効化します。

   これでユーザーディレクトリが PTA ディレクトリとして設定されました。これは、DN に o=NetscapeRoot が含まれるエントリのすべてのバインド要求を configdir.example.com 設定ディレクトリに送信します。

5. インストールが完了したら、admin ユーザーとしてユーザーディレクトリへのバインドを試み、ユーザーデータの作成を開始します。

   admin エントリは、uid=admin, ou=Administrators,ou=TopologyManagement,o=NetscapeRoot として設定ディレクトリに格納されます。これにより、ユーザーディレクトリは、PTA プラグインの設定で定義されたとおりに、バインド要求を設定ディレクトリにパススルーします。

6. 設定ディレクトリは、パスワードなどのバインド証明情報を認証し、確認結果をユーザディレクトリに返します。

7. ユーザーディレクトリは、admin ユーザーのバインドを許可します。

PTA プラグインの設定

PTA プラグインの設定情報は、PTA サーバー上の cn=Pass Through Authentication,cn=plugins,cn=config エントリに指定されます。

ユーザーディレクトリと設定ディレクトリを異なるサーバーインスタンスにインストールした場合は、PTA プラグインのエントリが自動的にユーザーディレクトリの設定に追加されます。両方のディレクトリを同じインスタンスにインストールし、他のディレクトリとのパススルー認証を行う場合は、事前にプラグイン設定エントリを作成する必要があります。

プラグイン設定エントリの作成

1. プラグイン設定エントリを作成するには、次のコマンドを実行します。

```
ldapmodify -a -h PTAhost -p port -D "cn=Directory Manager" -w password
dn: cn=Pass Through Authentication,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: Pass Through Authentication
nsslapd-pluginPath: ServerRoot/lib/passthru-plugin.extension
nsslapd-pluginInitfunc: passthruauth_init
nsslapd-pluginType: preoperation
nsslapd-plugin-depends-on-type: database
nsslapd-pluginId: passthruauth
nsslapd-pluginVersion: 5.2
```

```
nsslapd-pluginVendor: Sun Microsystems, Inc.
nsslapd-pluginDescription: pass through authentication plugin
nsslapd-pluginEnabled: on or off
nsslapd-pluginarg0: ldap[s]://authenticatingHost[:port]/PTAsubtree  options
```

ここで、*ServerRoot* はインストールによって異なり、HP-UX の場合は *extension* は `.sl`、その他すべての UNIX プラットフォームの場合は `.so`、Windows の場合は `.dll` となります。

プラグインの引数は、LDAP URL を指定し、認証ディレクトリサーバーのホスト名、オプションポート、PTA サブツリーを識別します。ポートを指定しない場合のデフォルトポートは、LDAP が 389、LDAPS が 636 です。また、後述するオプションの接続パラメータを指定することもできます。*PTAhost* に *PTAsubtree* が存在する場合、プラグインはバインド要求を *authenticatingHost* に渡さず、バインドはパススルーなしでローカルに処理されます。

2. 20 ページの「Directory Server の起動と停止」で説明している手順を実行してサーバーを再起動します。

セキュリティ保護された接続を使用するための PTA の設定

PTA プラグインは、パスワードを含むバインド証明情報を認証ディレクトリに送信するため、セキュリティ保護された接続を使用することをお勧めします。PTA ディレクトリが SSL を経由して認証ディレクトリと通信するように設定するには、次の手順を実行します。

- 第 11 章「セキュリティの実装」で説明している方法で、PTA ディレクトリと認証ディレクトリの両方で SSL を設定し、有効化します。
- PTA プラグインの設定を、たとえば次のように作成または変更し、LDAP URL 中の LDAPS とセキュリティ保護されたポートを使用できるようにします。

  ```
  ldaps://configdir.example.com:636/o=NetscapeRoot
  ```

オプションの接続パラメータの設定

PTA プラグインの引数には、LDAP URL の後にオプションの接続パラメータのセットを指定することができます。

```
ldap[s]://host[:port]/subtree [maxconns,maxops,timeout,ldapver,connlife]
```

パラメータは、上記の順序で指定する必要があります。これらのパラメータはオプションですが、いずれか 1 つを指定するときは、すべてを指定する必要があります。すべてのパラメータをカスタマイズする必要がない場合は、次のデフォルト値を指定します。*subtree* パラメータとオプションパラメータの間には、必ず空白文字を挿入してください。

各 LDAP URL プラグインに対して、次のオプションパラメータを設定できます。

- *maxconns* : PTA サーバーが認証サーバーに対して同時に開くことができる接続の最大数。このパラメータは、認証サーバーにパススルーできる同時バインドの最大数を制限する。デフォルト値は 3
- *maxops* : 単一の接続中に、PTA ディレクトリサーバーが認証ディレクトリサーバーに同時に送信できるバインド要求の最大数。このパラメータは、同時パススルー認証の数をさらに制限する。デフォルト値は 5
- *timeout* : PTA サーバーが認証サーバーからの応答を待つ最大遅延時間 (秒単位)。デフォルト値は 300 秒 (5 分)
- *ldapver* : PTA サーバーが認証ディレクトリサーバーへの接続に使用する LDAP プロトコルのバージョン。指定できる値は、LDAPv2 の場合に 2、LDAPv3 の場合に 3。デフォルト値は 3
- *connlife* : PTA サーバーが認証サーバーとの接続を再利用できる制限時間 (秒単位)。この制限時間が過ぎてからクライアントが PTA サブツリー内のバインドを要求した場合は、サーバーは PTA 接続を閉じて新たに開き直す。バインド要求が開始され、サーバーによってタイムアウトが超過していると判断されない限り、サーバーは接続を切断しない。このオプションを指定しない場合、または LDAP URL に指定されている認証サーバーが 1 つだけの場合、制限時間は適用されない。複数のホストが指定されている場合は、デフォルトで 300 秒 (5 分) に設定される。

PTA プラグイン引数の次の例では、接続の最大数を 10 に増やしていますが、タイムアウトの設定を 1 分 (60 秒) に減らしています。その他のパラメータには、デフォルト値が指定されています。

```
ldaps://configdir.example.com:636/o=NetscapeRoot 10,5,60,3,300
```

複数のサーバーとサブツリーの指定

PTA プラグインに複数の引数を設定することで、複数の認証サーバー、複数の PTA サブツリー、またはその両方を指定できます。各引数には 1 つの LDAP URL が含まれ、それぞれに接続オプションを設定できます。

同じ PTA サブツリーに対して複数の認証サーバーが存在するときは、認証サーバーはフェイルオーバサーバーとして機能します。PTA 接続のタイムアウト制限に達すると、プラグインはリストに指定されている順序でサーバーに接続します。すべての接続がタイムアウトになった場合は認証が失敗します。

複数の PTA サブツリーが定義されている場合、プラグインはバインド DN に基づいて、対応するサーバーに認証要求をパススルーします。次の例は、2 つの PTA サブツリーを定義する 4 つの PTA プラグイン引数を示しています。それぞれのサブツリーには、認証のためのフェイルオーバサーバーと、サーバー固有の接続パラメータが指定されています。

```
nsslapd-pluginarg0: ldaps://configdir.example.com/o=NetscapeRoot
 10,10,60,3,300
nsslapd-pluginarg1: ldaps://configbak.example.com/o=NetscapeRoot
 3,5,300,3,300
nsslapd-pluginarg2: ldaps://east.example.com/ou=East,ou=People,
 dc=example,dc=com 10,10,300,3,300
nsslapd-pluginarg3: ldaps://eastbak.example.com/ou=East,ou=People,
 dc=example,dc=com 3.5,300,3,300
```

PTA プラグイン設定の変更

PTA プラグインの設定を変更して、有効と無効の切り替え、認証ホストまたは PTA サブツリーの変更をいつでも実行できます。

1. PTA プラグイン設定エントリ (`cn=Pass Through Authentication,cn=plugins,cn=config`) を編集して、`nsslapd-pluginenabled` 属性と `nsslapd-pluginarg`*N* 属性を変更します。設定の編集には、コンソールと `ldapmodify` ユーティリティのどちらも使用できます。

   たとえば次のコマンドは、SSL と前述の接続パラメータを使用する PTA プラグインを有効化します。

   ```
   dn: cn=Pass Through Authentication,cn=plugins, cn=config
   changetype: modify
   replace: nsslapd-pluginenabled
   nsslapd-pluginenabled: on
   -
   replace: nsslapd-pluginarg0
   ```

```
nsslapd-pluginarg0: ldaps://configdir.example.com:636/
 o=NetscapeRoot 10,10,60,3,300
-
replace: nsslapd-pluginarg1
nsslapd-pluginarg1: ldaps://configbak.example.com:636/
 o=NetscapeRoot 3,5,300,3,300
^D
```

2. 20 ページの「Directory Server の起動と停止」で説明している手順を実行してサーバーを再起動します。

第 15 章

UID 一意性検査プラグインの使用

UID 一意性検査プラグインは、指定された属性の値が、ディレクトリまたはサブツリーのすべてのエントリで一意であることを確認します。このプラグインは、指定された属性に既存の値を含むエントリを追加しようとする操作や、ディレクトリ内の既存の値に属性を追加または変更する操作を停止させます。

デフォルトでは、このプラグインは UID 属性の一意性を確認します。ただし、このプラグインはデフォルトで有効になっていません。プラグインの新しいインスタンスを作成して、その他の属性値を一意にすることができます。UID 一意性検査プラグインが属性値の一意性を確認できるのは、1 つのサーバー上だけです。

この章は、次の節で構成されています。

- 概要
- UID 属性の一意性の適用
- その他の属性の一意性の適用
- レプリケーション使用時の一意性検査プラグインの使用

概要

UID 一意性検査プラグインは、前処理用のプラグインです。サーバーでディレクトリの更新が実行される前に、すべての LDAP 操作が検査されます。このプラグインでは、操作によって同じ属性値を持つエントリが 2 つ発生してしまうことを判別します。そのような場合、サーバーは操作を中断させて、エラー 19 (LDAP_CONSTRAINT_VIOLATION) をクライアントへ返します。

このプラグインは、ディレクトリ内の 1 つ以上のサブツリーや、特定のオブジェクトクラスのエントリ間で、一意性を確保するように設定できます。この設定により、属性値を一意にするエントリのセットが決まります。このエントリセットがターゲットとなった場合、また属性値がこのセットのエントリ全体の中で一意ではない場合、操作は中断されることがあります。

他の属性の一意性を確保する必要がある場合は、UID 一意性検査プラグインの複数のインスタンスを定義します。属性ごとに 1 つのプラグインインスタンスを定義し、値を一意にするエントリのセットを指定します。同じ属性に複数のプラグインインスタンスを用意することで、複数のエントリセットでその属性の一意性を個別に確保できます。指定した属性値は、各セットでは 1 つだけに限定されます。

既存のディレクトリで属性値の一意性を有効にしても、サーバーは既存のエントリ間での一意性をチェックしません。一意性が適用されるのは、エントリを追加する時点、あるいは属性が追加または変更される時点です。

デフォルトでは、UID 一意性検査プラグインは無効になっています。これは、このプラグインがマルチマスターレプリケーションに影響を与えるためです。レプリケーションを使用する場合に UID 一意性検査プラグインを有効にすることはできますが、441 ページの「レプリケーション使用時の一意性検査プラグインの使用」で説明される動作を理解しておく必要があります。

UID 属性の一意性の適用

ここでは、ディレクトリ内で `UID` 属性のデフォルトの一意性検査プラグインを有効にする方法と設定する方法について説明します。その他の属性に一意性を適用する方法については、439 ページの「その他の属性の一意性の適用」を参照してください。

コンソールを使用したプラグインの設定

コンソールの使用時には、デフォルトの `UID` 一意性検査プラグインを変更して、別の属性の一意性を適用しないでください。`UID` 一意性検査プラグインを使用しない場合は、このプラグインを無効にし、別の属性に対して新しいプラグインインスタンスを作成します。詳細は、439 ページの「その他の属性の一意性の適用」を参照してください。

1. Directory Server コンソールの最上位の「設定」タブで「プラグイン」ノードを展開し、`uid uniqueness` プラグインを選択します。
2. 右側のパネルで、「プラグインを有効に」チェックボックスを選択します。

   初期化関数とプラグインモジュールパスに関するフィールドを変更しないでください。
3. 一意性を適用するサブツリーの指定方法に従って、プラグイン引数を変更します。

- 1 つのサブツリーのベース DN を指定するには、引数 2 の値を編集します。複数のサブツリーを指定するには、「追加」をクリックして引数を追加し、新しいテキストフィールドにそれぞれのサブツリーのベース DN を入力します。

- ベースエントリのオブジェクトクラスによってサブツリーを指定するには、引数に次の値を設定します。

  引数 1 : attribute=UID
  引数 2 : markerObjectClass=*baseObjectClass*

  プラグインは、指定された *baseObjectClass* を持つディレクトリ内の各エントリの下位にあるサブツリーに対して、UID の一意性を適用します。たとえば、ou=Employees や ou=Contractors など多くの分岐にユーザーエントリがある場合は、markerObjectClass=organizationalUnit を指定します。

  marker オブジェクトの下位にある分岐の適用範囲は非常に広い可能性があるため、属性の一意性の適用を、さらにそれらのオブジェクトクラスの特定のエントリごとに制限することもできます。第 3 のプラグイン引数を追加するために「追加」をクリックし、次の値を設定します。

  引数 3 : requiredObjectClass=*entryObjectClass*

  baseObjectClass を持つエントリのサブツリー内で、*entryObjectClass* を持つエントリをターゲットとする操作だけで、一意性を適用できます。たとえば、従来からのユーザーエントリがある場合は、requiredObjectClass=inetorgperson を指定します。

4. UID 一意性検査プラグインの編集が終了したら、「保存」をクリックします。変更内容を有効にするには、サーバーを再起動する必要があります。
5. サーバーを再起動して、UID 属性に対して一意の値の適用を開始します。

コマンド行からのプラグインの設定

ldapmodify コマンドを使用して UID 一意性検査プラグインを有効にし、設定する方法について、次の手順で説明します。プラグイン設定エントリの DN は、cn=UID uniqueness,cn=plugins,cn=config です。

1. nsslapd-pluginEnabled 属性を on または off に設定して、プラグインを有効または無効にします。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=UID uniqueness,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on or off
^D
```

2. 一意性を適用するサブツリーの指定方法に従って、プラグイン引数を変更します。

- 1 つのサブツリーのベース DN を指定するには、nsslapd-pluginarg1 の値を変更します。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=UID uniqueness,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginArg1
nsslapd-pluginArg1: subtreeBaseDN
^D
```

複数のサブツリーを指定するには、各サブツリーの完全ベース DN を値として指定した引数を追加します。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=UID uniqueness,cn=plugins,cn=config
changetype: modify
add: nsslapd-pluginArg2
nsslapd-pluginArg2: subtreeBaseDN
-
add: nsslapd-pluginArg3
nsslapd-pluginArg3: subtreeBaseDN
-
...
^D
```

- ベースエントリのオブジェクトクラスに従ってサブツリーを指定するには、引数に次の値を設定します。*baseObjectClass* を持つ各エントリの下位にあるサブツリーに対して、UID 属性の一意性が適用されます。オプションとして、3 番目の引数に *entryObjectClass* を指定すると、このオブジェクトクラスを持つエントリをターゲットとする操作だけで、一意性を適用することもできます。

```
ldapmodify -h host -p port -D "cn=Directory Manager" -w password
dn: cn=UID uniqueness,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginArg0
nsslapd-pluginArg0: attribute=UID
-
replace: nsslapd-pluginArg1
nsslapd-pluginArg1: markerObjectClass=baseObjectClass
-
replace: nsslapd-pluginArg2
nsslapd-pluginArg2: requiredObjectClass=entryObjectClass
^D
```

3. 変更内容を有効にするために、サーバーを再起動します。

その他の属性の一意性の適用

UID 一意性検査プラグインを使用すると、すべての属性の一意性を適用できます。ディレクトリの cn=plugins,cn=config の下に新しいエントリを作成することによって、プラグインの新しいインスタンスを作成する必要があります。

1. ldapmodify コマンドを使用して、新しいプラグインインスタンスの設定エントリを追加します。コマンドの最初の部分は、次のとおりです。残りの部分については、以下の手順で説明します。

   ```
   ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
   dn: cn=plug-in_name,cn=plugins,cn=config
   objectClass: top
   objectClass: nsSlapdPlugin
   objectClass: extensibleObject
   cn: plug-in_name
   nsslapd-pluginDescription: Enforce unique attribute values
   nsslapd-pluginType: preoperation
   nsslapd-plugin-depends-on-type: データベース
   nsslapd-pluginPath: ServerRoot/lib/UID-plugin.extension
   nsslapd-pluginVersion: 5.2
   nsslapd-pluginVendor: Sun Microsystems, Inc.
   nsslapd-pluginId: NSUniqueAttr
   nsslapd-pluginInitfunc: NSUniqueAttr_Init
   nsslapd-pluginEnabled: state
   ...
   ```

   コマンドのこの部分では、*plug-in_name* に、cn=mail uniqueness など属性の名前を含んだ、短くわかりやすい名前を付ける必要があります。*ServerRoot* とライブラリの *extension* は、プラットフォームによって異なります。最後に、サーバーの再起動時に、新しいインスタンスの *state* を on または off に指定します。

2. サーバーでプラグイン署名を確認する場合は、新しい一意性検査プラグインの設定に署名を含める必要があります。一意性検査プラグインは UID 一意性検査プラグインの新しいインスタンスであるため、次のファイルに保存されている同じ署名情報を使用する必要があります。

   ServerRoot/plugins/signatures/plugin.signatures

   このファイルは、root などサーバーのインストールに使用したユーザー ID だけで読み取り可能です。このファイル内で、dn: cn=UID uniqueness,cn=plugins,cn=config エントリの下位の情報を確認します。ファイルで指定されている値と同じものを使用して、次の属性を新しいプラグインインスタンスに追加します。また、ds-signedPlugin オブジェクトクラスを指定することも必要です。

```
objectClass: ds-signedPlugin
ds-pluginDigest:: 02Q7yVLYsC8FInPrvbAKYq7Rj0o=
ds-pluginSignature:: MIIBjwYJKoZIhvcNAQcCoIIBgDCCAXwCAQExCzAJBg
 UrDgMCGgUAMAsGCSqGSIb3DQEHATGCAVswggFXAgEBMFYwTTELMAkGA1UEBhMC
 VVMxGTAXBgNVBAoTEFN1biBNaWNyb3N5c3RlbXMxIzAhBgNVBAMTGlBsdWdpbi
 BTaWduaW5nIENlcnRpZmljYXRlAgUA5X1ATjAJBgUrDgMCGgUAoF0wGAYJKoZI
 A2WjAjBgkqhkiG9w0BCQQxFgQU77mUWWJWttkH89eLwTr/fQtz+BswDQYJKoZI
 hvcNAQEBBQAEgYAzZwvgwo+OdKNkXWxlP+pUNpHesL6UQcvXcm37mEQyikRvLs
 hy3X0JutFhEXaCfU4UX76A3Zzedr2Iy0YEGkiPCu3g8jnkFEG/ux0ZMeOPiulF
 f9PUfqpnz6phq19eBZxZ/MBFLxtlzJHG42Ext/un4ZzQIg==
...
```

プラグインの署名については、38 ページの「プラグインの署名の検証」を参照してください。プラグイン署名を確認しない場合は、これらの属性は必要ありません。設定では新しいプラグインインスタンスが署名されていないことが示されますが、プラグインは正常に機能します。

3. コマンドの残りの部分では、一意性を適用するサブツリーの指定方法に従って、プラグイン引数を指定します。

- 1 つ以上のサブツリーをベース DN に基づいて定義するには、1 番目の引数に一意の値を持つ属性の名前を指定し、2 番目以降の引数にサブツリーのベースエントリの完全 DN を指定する必要があります。

```
nsslapd-pluginarg0: attribute_name
nsslapd-pluginarg1: subtreeBaseDN
nsslapd-pluginarg2: subtreeBaseDN
...
^D
```

- ベースエントリのオブジェクトクラスに基づいてサブツリーを定義するには、1 番目の引数に `attribute=`*attribute_name* を追加して、一意の値を持つ属性の名前を指定する必要があります。2 番目の引数には、一意性を適用するサブツリーのベースエントリを決める *baseObjectClass* を指定する必要があります。オプションとして、3 番目の引数に *entryObjectClass* を指定すると、このオブジェクトクラスを持つエントリをターゲットとする操作だけで、一意性を適用することもできます。

```
nsslapd-pluginarg0: attribute=attribute_name
nsslapd-pluginarg1: markerObjectClass=baseObjectClass
nsslapd-pluginarg2: requiredObjectClass=entryObjectClass
^D
```

すべてのプラグイン引数に、= 記号の前後に空白を挿入することはできません。

4. サーバーを再起動して、一意性検査プラグインの新しいインスタンスをサーバーに読み込みます。

レプリケーション使用時の一意性検査プラグインの使用

UID 一意性検査プラグインでは、レプリケーションの一部として更新処理が行われた場合は、属性値の検査は一切行われません。これはシングルマスターレプリケーションには影響を与えませんが、プラグインはマルチマスターレプリケーションに対する属性の一意性を自動的に適用できません。

シングルマスターレプリケーションモデル

クライアントアプリケーションによる変更処理はすべてマスターレプリカ上で行われるので、UID 一意性検査プラグインをマスターサーバー上で有効にする必要があります。レプリケートされたサフィックスで一意性を適用するように、プラグインを設定する必要があります。マスターが該当の属性値が一意であることを確認するため、コンシューマサーバー上でプラグインを有効にする必要はありません。

シングルマスターのコンシューマ上で UID 一意性検査プラグインを有効にしても、レプリケーションや通常のサーバー操作と干渉することはありません。ただし、パフォーマンスがわずかに低下することがあります。

マルチマスターレプリケーションモデル

UID 一意性検査プラグインは、マルチマスターレプリケーションモデルでの使用を想定して設計されていません。マルチマスターレプリケーションは疎整合型のレプリケーションモデルを使用するので、両方のサーバーでプラグインが有効になっていても、同じ属性値が両方のサーバーに同時に追加された場合は検出されません。

ただし、次の条件では、UID 一意性検査プラグインを使用できます。

- 一意性検査の実行対象となる属性がネーミング属性である
- 一意性検査プラグインが、すべてのマスター上の同じサブツリーの同じ属性で有効になっている

これらの条件を満たしている場合、一意性に関する競合は、レプリケーション時のネーミング競合として報告されます。ただし、レプリケーション時のネーミング競合は、手動で解決する必要があります。レプリケーション時の競合の解決方法については、329 ページの「よく発生するレプリケーションの競合の解決」を参照してください。

サードパーティ製品のライセンス

この製品には以下の著作権が適用されるソフトウェアが含まれています。ここで言及されているすべての商標もしくは登録商標の所有権はそれぞれの所有者に帰属します。

Copyright (c) 1990-2000 Sleepycat Software. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Redistributions in any form must be accompanied by information on how to obtain complete source code for the DB software and any accompanying software that uses the DB software. The source code must either be included in the distribution or be available for no more than the cost of distribution plus a nominal fee, and must be freely redistributable under reasonable conditions. For an executable file, complete source code means the source code for all modules it contains. It does not include source code for modules or files that typically accompany the major components of the operating system on which the executable file runs.
THIS SOFTWARE IS PROVIDED BY SLEEPYCAT SOFTWARE "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT, ARE DISCLAIMED. IN NO EVENT SHALL SLEEPYCAT SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 1990, 1993, 1994, 1995 The Regents of the University of California. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR

CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 1995, 1996 The President and Fellows of Harvard University. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY HARVARD AND ITS CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL HARVARD OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2001 Carnegie Mellon University. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name "Carnegie Mellon University" must not be used to endorse or promote products derived from this software without prior written permission. For permission or any other legal details, please contact Office of Technology Transfer Carnegie Mellon University 5000 Forbes Avenue Pittsburgh, PA 15213-3890 (412) 268-4387, fax: (412) 268-7395 tech-transfer@andrew.cmu.edu
4. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by Computing Services at Carnegie Mellon University (http://www.cmu.edu/computing/)."
CARNEGIE MELLON UNIVERSITY DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL CARNEGIE MELLON UNIVERSITY BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 1997, 1998 Kungliga Tekniska Högskolan (Royal Institute of Technology, Stockholm, Sweden). All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by Kungliga Tekniska Högskolan and its contributors.

4. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 1987, 1988 Student Information Processing Board of the Massachusetts Institute of Technology. Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of M.I.T. and the M.I.T. S.I.P.B. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T. and the M.I.T. S.I.P.B. make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Copyright (c) 1996 Regents of the University of Michigan. All rights reserved. Redistribution and use in source and binary forms are permitted provided that this notice is preserved and that due credit is given to the University of Michigan at Ann Arbor. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission. This software is provided "as is" without express or implied warranty.

Copyright 1992 Network Computing Devices, Inc. Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Network Computing Devices may not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Network Computing Devices makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.
NETWORK COMPUTING DEVICES DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL NETWORK COMPUTING DEVICES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 2001-2002 The Apache Software Foundation. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
4.The names "Xerces" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.
THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

COPYRIGHT Copyright (c) 1997-2000 Messaging Direct Ltd. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
THIS SOFTWARE IS PROVIDED BY MESSAGING DIRECT LTD. "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL MESSAGING DIRECT LTD. OR ITS EMPLOYEES OR AGENTS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. END COPYRIGHT

COPYRIGHT Copyright (c) 2000 Fabian Knittel. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Redistributions in binary form must reproduce all prior and current copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. END COPYRIGHT

The source code to the Standard Version of Perl can be obtained from CPAN sites, including http://www.perl.com/.

This product incorporates compression code by the Info-ZIP group. There are no extra charges or costs due to the use of this code, and the original compression sources are freely available from ftp://ftp.cdrom.com/pub/infozip/ on the Internet.

索引

A

B

C

D

F

G

I

K

L

M

N

P

R

S

T

U

V

W

あ

い

え

お

か

き

く

け

こ

さ

し

つ

て

と

に

は

ひ

ふ

ほ

ま

ゆ

よ

り

る

れ

ろ

わ