管理者ガイド

iPlanet Directory Server

Version 5.1

816-4123-10
2002 年 2 月

目 次

表目次

本書について

iPlanet Directory Server 5.1 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です。iPlanet 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。

本書、『管理者ガイド』では、iPlanet Directory Server に基づくディレクトリサービスの保守に必要なすべての管理作業について説明します。

iPlanet Directory Server リリースの新機能および拡張機能の最新情報については、次のオンラインリリースノートを参照してください。

```
http://docs.iplanet.com/docs/manuals/directory.html
```

お読みになる前に

このマニュアルでは、Directory Server とその内容を管理する方法について説明していますが、ディレクトリサービスの導入、インストール、および管理を正しく行うために必要な、ディレクトリおよび構造に関する基本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、『iPlanet Directory Server 導入ガイド』で説明しています。

Solaris 9 オペレーティング環境版の iPlanet Directory Server 5.1 を構成する方法については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「iPlanet Directory Server 5.1 の構成」の章を参照してください。

『Managing Servers with iPlanet Console』には、iPlanet サーバ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。このマニュアルをお読みになり、一般的な管理業務の概念を理解してから、iPlanet Directory Server の管理を始めてください。

表記上の規則

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。

イタリック体 (*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、一連のメニュー項目選択におけるセパレータとして使用します。たとえば、「オブジェクト」> 「新規作成」> 「ユーザ」は、「オブジェクト」メニューをプルダウンし、マウスをドラッグダウンして「新規作成」を強調表示し、「新規作成」のサブメニューの「ユーザ」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。

本書では、パスおよびファイル名に次の形式を使用しています。

```
/var/ds5/slapd-serverID/...
```

serverID は、構成時にサーバに指定したサーバの識別子を表します。たとえば、Directory Server の名前を `phonebook` にした場合、実際のパスは次のようになります。

```
/var/ds5/slapd-phonebook/...
```

関連情報

iPlanet Directory Server のマニュアルセットには、次のマニュアルも含まれています。

『iPlanet Directory Server 導入ガイド』Directory Server の導入計画の概要について説明し、導入の事例を提供します。

『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』Directory Server に付属するコマンド行スクリプトの使用方法について説明します。

『iPlanet Directory Server スキーマリファレンス』Directory Server に同梱されている LDAP スキーマおよびクライアントアプリケーションに役立つ情報を提供します。

iPlanet に関するその他の情報については、次のサイトを参照してください。

- iPlanet の製品マニュアルオンライン :
  `http://docs.iplanet.com/docs/manuals/`
- iPlanet 製品の技術情報 :
  `http://www.iplanet.com/support/technical_resources/`
- iPlanet プロフェッショナルサービスに関する情報 :
  `http://www.iplanet.com/services/professional_services_3_3.html`
- Solaris パッチおよびサポート用の Sun Enterprise Services :
  `http://www.sun.com/service/`
- iPlanet の開発者向け情報 :
  `http://developer.iplanet.com/`
- iPlanet のトレーニング情報 :
  `http://www.iplanet.com/learning/index.html`
- iPlanet 製品のデータシート :
  `http://www.iplanet.com/products/index.html`

第 1 章

iPlanet Directory Server の概要

iPlanet 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、および両方のサーバを管理するグラフィカルインタフェースを提供する iPlanet Console が含まれています。この章では、Directory Server の概要と、Console 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。

この章は、次の節で構成されています。

- iPlanet Directory Server の管理の概要
- iPlanet Directory Server Console の使用
- iPlanet Directory Server の起動と停止
- LDAP パラメタの構成
- SSL が有効になった状態でのサーバの起動

iPlanet Directory Server の管理の概要

iPlanet Directory Server は、企業全体のユーザおよび資源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です。iPlanet Directory Server は、LDAP (Lightweight Directory Access Protocol) というオープンシステムサーバ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。このサーバは、ディレクトリデータベースを管理し、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、Administration Server を経由して実行できます。Administration Server は、Directory Server (およびその他の iPlanet サーバすべて) を管理できるようにするために iPlanet が提供するもう 1 つのサーバです。iPlanet Console は、Administration Server のグラフィカルインタフェースです。iPlanet Directory Server Console は iPlanet Console の一部であり、特に iPlanet Directory Server で使用する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、iPlanet Directory Server Console から実行できます。構成ファイルを編集するか、コマンド行ユーティリティを使用して、手動で管理タスクを実行することもできます。iPlanet Console については、『Managing Servers with iPlanet Console』を参照してください。

iPlanet Directory Server Console の使用

iPlanet Directory Server Console は、iPlanet Console 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 iPlanet Console から iPlanet Directory Server Console を起動します。

iPlanet Directory Server Console の起動

1. Directory Server 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 ユーザとして次のコマンドを入力し、デーモンを起動します。

   ```
   # /usr/sbin/directoryserver start
   ```

2. Administration Server 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 ユーザとして次のコマンドを入力し、デーモンを起動します。

   ```
   # /usr/sbin/directoryserver start-admin
   ```

3. 次のコマンドを入力して iPlanet Console を起動します。

   ```
   # /usr/sbin/directoryserver startconsole
   ```

   Console のログインウィンドウが表示されます。構成ディレクトリ (`o=NetscapeRoot` 接尾辞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 DN、パスワード、およびその Directory Server の Administration Server の URL を入力する必要があります。

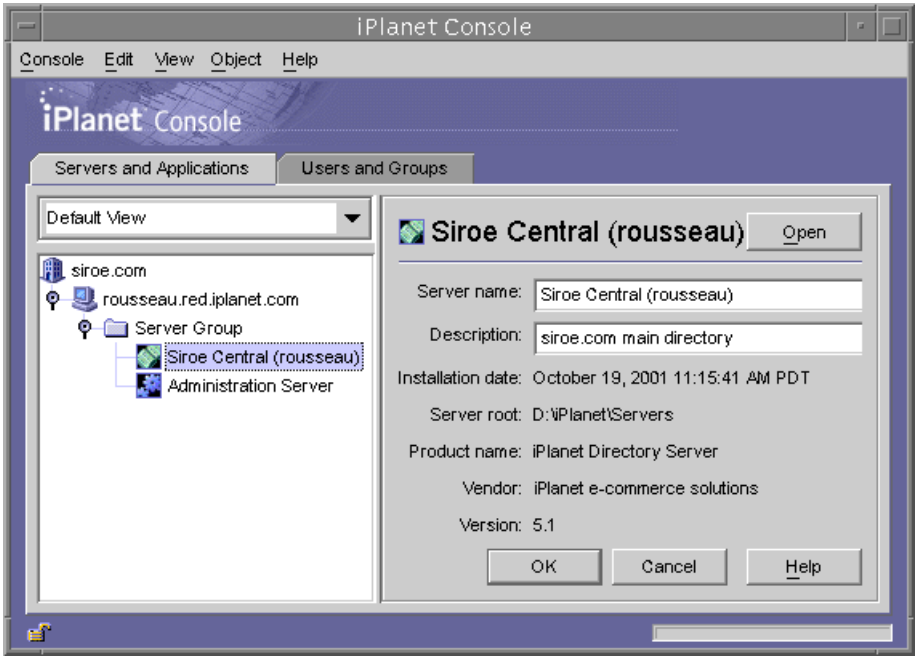
4. 目的の操作を実行するために必要なアクセス権を持つユーザのバインド DN とパスワードを使用してログインします。たとえば、`cn=Directory Manager` とこれに対応する適切なパスワードを使用します。

   iPlanet Console が表示されます。

5. 左側の区画のツリーを使用して 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。

   Directory Server の名前および説明を編集するには、「Edit (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。図のように左のツリーに名前が表示されます。

図 1-1　iPlanet Console



「OK」をクリックして新しい名前と説明を設定します。

6. ツリー内の Directory Server 名をダブルクリックまたは「Open (開く)」ボタンをクリックして、この Directory Server を管理する iPlanet Directory Server Console を表示します。

Directory Server Console の操作

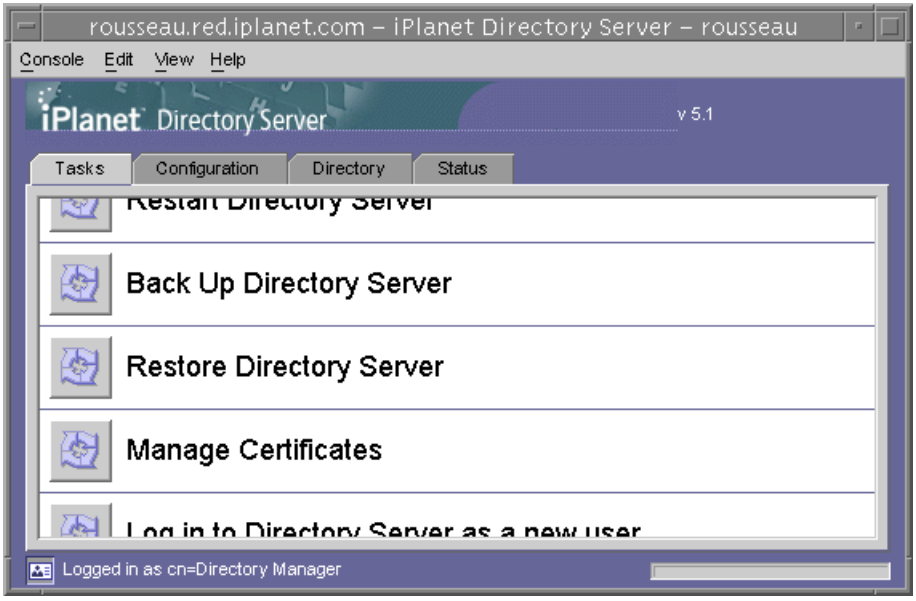
Directory Server Console では、Directory Server インスタンスでブラウズおよび管理操作を実行するためのインタフェースが提供されます。常時表示されている 4 つのタブを使用すると、Directory Server のすべての機能にアクセスできます。

- 「Tasks (タスク)」タブ
- 「Configuration (構成)」タブ
- 「Directory (ディレクトリ)」タブ
- 「Status (状態)」タブ

「Tasks (タスク)」タブ

「Tasks (タスク)」タブは、Directory Server Console 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、Console ウィンドウのサイズを変更する必要があります。

図 1-2　Directory Server Console の「タスク」タブ



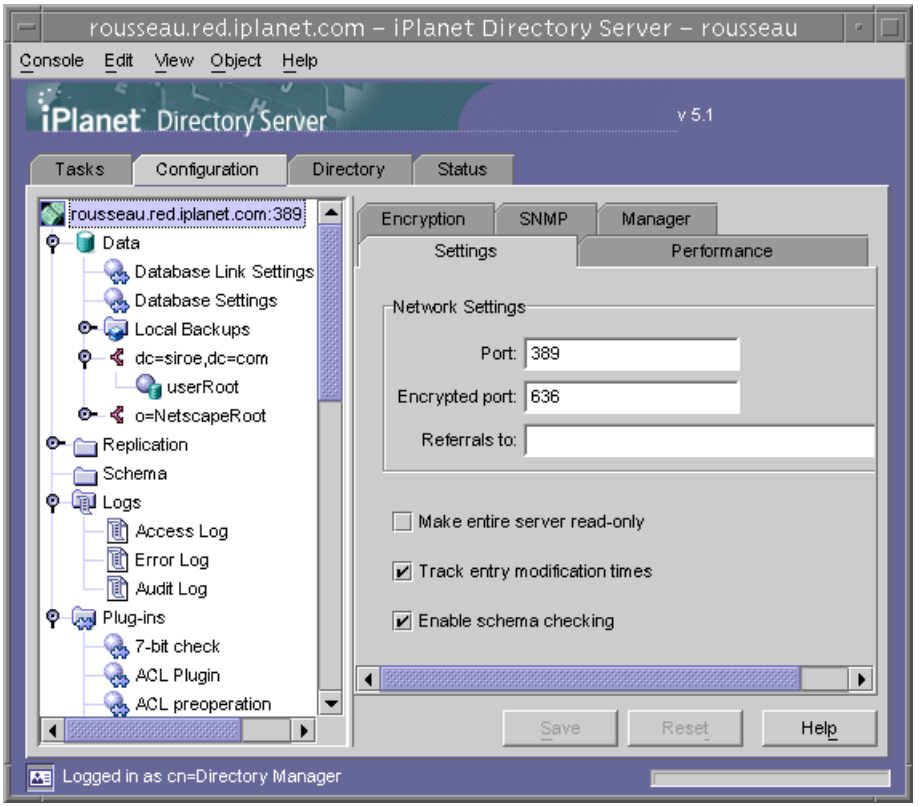
これらのタスクを実行するには、ディレクトリマネージャの権限をもつユーザとしてログインする必要があります。不十分な権限でタスクを実行しようとすると、ディレクトリマネージャの DN およびパスワードの入力を促すメッセージが表示されます。

「Configuration (構成)」タブ

Directory Server Console の「Configuration (構成)」タブでは、データベース、接尾辞、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、ディレクトリマネージャの権限をもつユーザとしてログインした場合だけ有効となります。

このタブの左側にはすべての構成機能のツリー、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。たとえば、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 Console の「Configuration (構成)」タブ



左側のツリーから構成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区画の 1 つ以上のタブに表示されます。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、サーバが再起動されるまで反映されない場合があります。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。

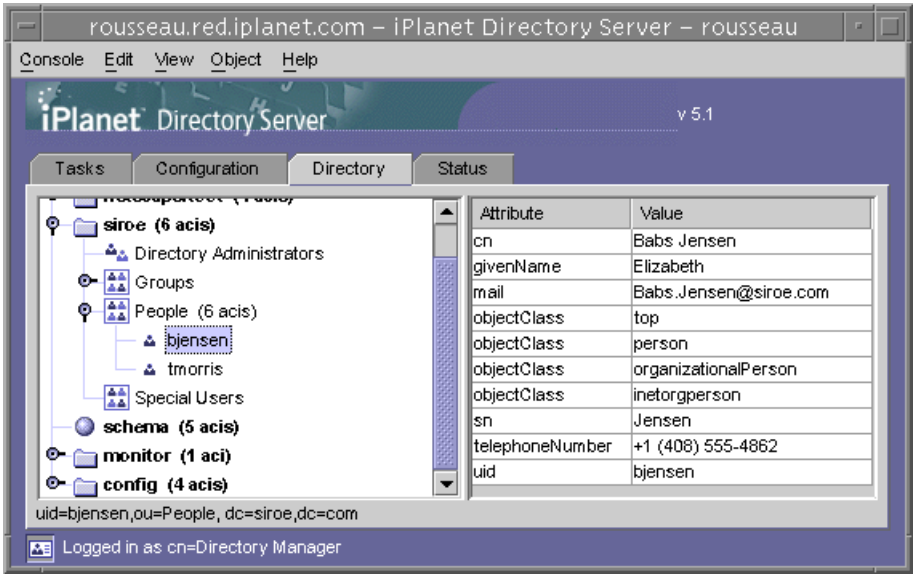
変更が保存されていないタブでは、タブ名の隣に赤のマークが表示されます。変更を保存しないで別の項目を設定したり、他の主要なタブに移動したりしても、行なった変更はタブに残ります。「Save (保存)」および「Reset (リセット)」ボタンは、指定された構成項目のすべてのタブに適用されますが、他の項目の未保存の設定には影響を与えません。

ほとんどのテキストフィールドには、正しい構文でのみ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、「Save (保存)」ボタンは無効になります。「Edit (編集)」>「Configuration (設定)」ダイアログボックスの「Misc. (その他)」タブを使用すると、不正な値をハイライトするためにイタリックフォントを選択したり、まったく強調しないように設定することができます。

「Directory (ディレクトリ)」タブ

Console の「Directory (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 Console の「Directory (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、構成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、構成設定を安全に変更するには、「Configuration (構成)」タブから利用できるダイアログボックスを使用する必要があります。

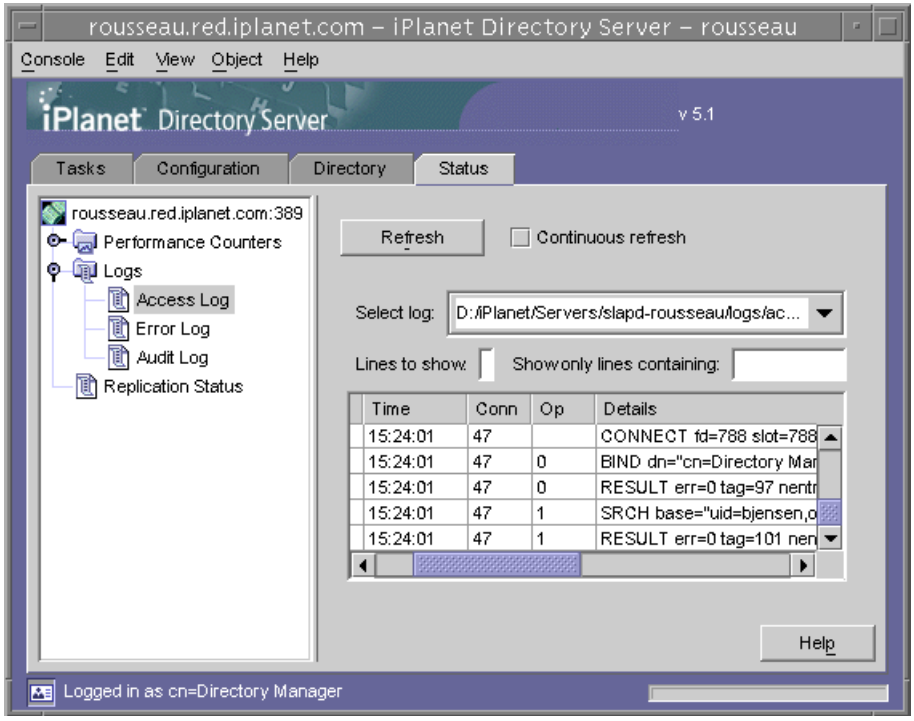
「Directory (ディレクトリ)」タブのレイアウトおよび内容を変更する場合は、「Display (表示)」メニューのオプションを使用できます。新しいレイアウトオプションを使用すると、最下位のエントリを含むすべてのエントリを 1 つのツリーに表示したり、右側の区画に属性を表示したりすることができます。デフォルトでは、最下位のエントリは、左側のツリーではなく、右側に表示されます。

「View (表示)」>「Display (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左側の区画に表示されています。

「Status (状態)」タブ

「Status (状態)」タブには、サーバ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、その内容が右側の区画に表示されます。たとえば、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 Console の「Status (状態)」タブ



Console 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、iPlanet Directory Server Console へのログインに使用したバインド 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

iPlanet Directory Server Console からエントリの作成や管理を行う場合や、はじめて iPlanet Console 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 iPlanet Console を起動するときは、ディレクトリマネージャ DN を使用してログインできます。ログイン後は、Console を停止して再起動しなくても、いつでも別のユーザとしてログインできます。

iPlanet Console でログイン名を変更するには、次の手順を実行します。

1. iPlanet Directory Server Console の「Tasks (タスク)」タブを選択し、「Log on to the Directory Server as a New User (Directory Server に新しいユーザとしてログイン)」というラベルの隣のボタンをクリックします。Console の別のタブでは、「Console (コンソール)」>「Log in as New User (新しいユーザとしてログイン)」を選択します。

   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、「OK」をクリックします。

   サーバ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、ディレクトリマネージャとしてバインドする場合は、「Distinguished Name (識別名)」テキストボックスに次のように入力します。

   cn=Directory Manager

ディレクトリマネージャ DN およびパスワードについては、後の節で説明します。

ディレクトリマネージャの構成

ディレクトリマネージャとは、特権を持つデータベース管理者のことで、UNIX の root ユーザにあたります。このため、ディレクトリマネージャ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。デフォルトは `cn=Directory Manager` です。

このユーザのパスワードは `nsslapd-rootdn` 属性で定義します。

Directory Server Console を使用して、ディレクトリマネージャ 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. ディレクトリマネージャとして Directory Server Console にログインします。

   すでに Console にログインしている場合に、別のユーザ名でログインする方法については、32 ページの「ログイン ID の変更」を参照してください。

2. iPlanet Directory Server Console で「構成」タブを選択し、左側の区画のナビゲーションツリーで一番上にあるエントリを選択します。
3. 右側の区画で「マネージャ」タブを選択します。
4. 「ディレクトリマネージャ DN」フィールドに、ディレクトリマネージャの新しい識別名を入力します。

   デフォルト値は `cn=Directory Manager` です。

5. 「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、サーバ上でディレクトリマネージャのパスワードを保存するために使用する保存スキーマを選択します。
6. 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。
7. 「保存」をクリックします。

ヘルプシステムの起動

iPlanet Directory Server のヘルプシステムは、iPlanet Administration Server に依存しています。Administration Server からはリモートとなるマシンで iPlanet Directory Server Console 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。

Administration Server 上で承認されたクライアント IP アドレス :iPlanet Directory Server Console を実行しているマシンは、Administration Server にアクセスする必要があります。Administration Server がクライアントマシンの IP アドレスを受け入れるように構成するには、Administration Server 上で次の操作を実行します。

1. iPlanet Administration Server Console を起動します。Console は Administration Server と同じマシン上で実行している必要があります。
2. 「構成」タブをクリックしてから、「ネットワーク」タブをクリックします。
3. 「接続制限の設定」プルダウンメニューから「許可する IP アドレス」を選択します。「編集」をクリックします。
4. 「IP アドレス」フィールドを次のように編集します。*.*.*.*

   これで、すべてのクライアントが Administration Server にアクセスできるようになります。
5. Administration Server を再起動します。これで、Directory Server Console の「ヘルプ」ボタンをクリックして、オンラインヘルプを起動することができます。

Administration Server 上で承認されたプロキシ :Directory Server Console を実行しているクライアントマシン上の HTTP 接続でプロキシを使用する場合は、次のいずれかの操作を実行する必要があります。

- Directory Server Console を実行しているマシンからプロキシを削除する。これによって、クライアントマシンが直接 Administration Server にアクセスできるようになる

  Directory Server Console を実行しているコンピュータからプロキシを削除するには、ヘルプの実行に使用するブラウザのプロキシ構成を変更する必要があります。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構成を表示します。Internet Explorer の場合は、「インターネットオプション」の「ツール」メニューを選択します。
- Administration Server の使用可能な IP アドレスのリストに、クライアントマシンのプロキシ IP アドレスを追加する

警告 Administration Server にクライアントマシンの IP アドレスを追加すると、システムに潜在的なセキュリティホールが発生する可能性があります。

Console クリップボード

Directory Server Console では、システムのクリップボードを使用してテキストのコピー、切り取り、および貼り付けを行います。またタイプする手間を省く便利な機能もあります。「ディレクトリ」タブ内で操作中、クリップボードにエントリの DN または URL を生成できます。

1. Directory Server Console で、「ディレクトリ」タブを選択します。
2. ツリーから、DN または URL をコピーしたいエントリを選択 (左クリック) します。

3. 次にメニューの「編集」>「DN のコピー」または「編集」>「URL のコピー」を選択します。

DN または URL をテキストフィールドに貼り付けることができるように、ダイアログボックスまたは他のタブを開く前にこの操作を実行してください。

iPlanet Directory Server の起動と停止

SSL (Secure Sockets Layer) を使用していない場合は、次の方法で Directory Server を起動および停止します。SSL を使用している場合は、39 ページの「SSL が有効になった状態でのサーバの起動」を参照してください。

注	UNIX システムでは、システムを再起動しても `slapd` プロセスは自動的に起動しません。これは、Directory Server ではスタートアップやランコマンド (`rc`) の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は、オペレーティングシステムのマニュアルを確認してください。

Console からのサーバの起動と停止

1. iPlanet Directory Server Console を起動します。

   手順については、26 ページの「iPlanet Directory Server Console の起動」を参照してください。

2. 必要に応じて、「タスク」タブで「Directory Server を起動する」または「Directory Server を停止する」をクリックします。

iPlanet Directory Server Console による Directory Server の起動または停止が正常に完了すると、サーバが正常に起動または停止したことを示すメッセージボックスが表示されます。

コマンド行からのサーバの起動と停止

root 権限で、次のスクリプトの 1 つを実行します。

```
# /usr/sbin/directoryserver start
```

または

```
# /usr/sbin/directoryserver stop
```

スクリプトは Directory Server と同じ UID と GID を使用し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、ユーティリティを nobody として実行する必要があります。

LDAP パラメタの構成

iPlanet Directory Server Console を使用して、サーバのネットワークと LDAP 設定に関連するパラメタを表示および変更できます。ここでは、次の項目について説明します。

- Directory Server のポート番号の変更
- Directory Server 全体を読み取り専用モードへ設定
- ディレクトリエントリへの変更の記録

スキーマの検査については、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。

Directory Server のポート番号の変更

iPlanet Directory Server Console を使用するか、cn=config エントリの下にある nsslapd-port 属性の値を変更して、ユーザディレクトリサーバ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。

iPlanet の構成情報 (o=NetscapeRoot サブツリー) を含む iPlanet 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、iPlanet Directory Server Console を使用します。

構成ディレクトリ、またはユーザ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。

- Administration Server の構成ディレクトリ、またはユーザ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る。詳細は、『Managing Servers with iPlanet Console』を参照してください。
- 構成ディレクトリまたはユーザディレクトリをポイントするほかの iPlanet サーバがインストールされている場合は、これらのサーバを更新して、新しいポート番号をポイントさせる必要があります。

ユーザディレクトリまたは構成ディレクトリが要求を待機する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するには、次の手順を実行します。

1. iPlanet Directory Server Console で「構成」タブを選択し、左側の区画のナビゲーションツリーの一番上のエントリを選択します。

2. 右側の区画で「設定」タブを選択します。
3. 「ポート」テキストボックスに、サーバが 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。
4. 「暗号化ポート」テキストボックスに、サーバが SSL 通信に使用するポート番号を入力します。

   指定する暗号化ポート番号は、通常の LDAP 通信に使用するポート番号とは異なるものである必要があります。デフォルトは 636 です。
5. 「保存」をクリックして、サーバを再起動します。

   詳細は、35 ページの「iPlanet Directory Server の起動と停止」を参照してください。

Directory Server 全体を読み取り専用モードへ設定

1 回の操作だけで Directory Server 内にある複数のデータベースすべてを読み取り専用モードに設定することができます。ただし、Directory Server にレプリカが含まれる場合は、レプリケーションが無効になるので、読み取り専用モードを使用しないように注意してください。

Directory Server を読み取り専用モードに設定するには、次の手順を実行します。

1. iPlanet Directory Server Console で「構成」タブを選択し、左側の区画のナビゲーションツリーの一番上のエントリを選択します。
2. 右側の区画で「設定」タブを選択します。
3. 「サーバ全体を読み取り専用にする」チェックボックスを選択します。
4. 「保存」をクリックして、サーバを再起動します。

注	この操作を実行すると、Directory Server の構成も読み取り専用になります。Directory Server が読み取り専用モードになっている間は、サーバ構成の更新、プラグインの有効化と無効化、または Directory Server の再起動を実行できません。

1 つのデータベースだけを読み取り専用モードに設定する方法については、135 ページの「読み取り専用モードの有効化」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構成できます。

- `creatorsName` : エントリを最初に作成したユーザの識別名
- `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データベースリンクを使用して、エントリの作成や変更をした場合、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、リモートサーバでのプロキシ認証権限を持つ管理者名が反映されます。プロキシ認証については、98 ページの「バインド資格の指定」を参照してください。

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

1. iPlanet Directory Server Console で「構成」タブを選択し、左側の区画のナビゲーションツリーの一番上のエントリを選択します。
2. 右側の区画で「設定」タブを選択します。
3. 「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。
4. 「保存」をクリックして、サーバを再起動します。

   詳細は、35 ページの「iPlanet Directory Server の起動と停止」を参照してください。

SSL が有効になった状態でのサーバの起動

コマンド行からサーバを起動する必要があり、パスワードファイルを作成して証明書のパスワードを保存できます。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、Server Console からサーバを起動できます。さらに、無人でサーバを実行している場合も、サーバ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保存されます。したがって、この方法を使用すると、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用しないでください。

パスワードファイルは、次の位置に置く必要があります。

`/usr/iplanet/ds5/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の識別子を示します。

ファイルには、次のようにトークン名とパスワードを含める必要があります。

```
Token:Password
```

たとえば、次のようにします。

```
Internal (Software) Token:mypassword
```

証明書データベースを作成するには、Administration Server と証明書設定ウィザードを使用する必要があります。証明書データベース、証明書エイリアス、SSL、およびサーバ証明書の取得方法については、『Managing Servers with iPlanet Console』を参照してください。Directory Server 上での SSL の使い方については、第 11 章「SSL の管理」を参照してください。

第 2 章

ディレクトリエントリの作成

この章では、Directory Server Console および `ldapmodify` と `ldapdelete` コマンド行ユーティリティを使用して、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。

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。このためには、エントリの作成やデフォルトスキーマの変更に先立って、『iPlanet Directory Server 導入ガイド』をお読みください。

この章は、次の節で構成されています。

- Directory Console 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
- LDIF 更新文
- 参照整合性の管理

Directory Console からのエントリの管理

Directory Server Console の「Directory (ディレクトリ)」タブとプロパティエディタを使用して、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。

Directory Server Console の起動およびユーザインタフェースの使用方法については、26 ページの「iPlanet Directory Server Console の使用」を参照してください。

複数のエントリを同時に追加する場合には、51 ページの「コマンド行からのエントリの管理」で説明されているコマンド行ユーティリティを使用できます。

ここでは、次の項目について説明します。

- ルートエントリの作成
- ディレクトリエントリの作成
- ディレクトリエントリの変更

- ディレクトリエントリの削除

ここでは、オブジェクトクラスと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。オブジェクトクラスと属性の概要については、『iPlanet Directory Server 導入ガイド』を参照してください。iPlanet サーバ製品が提供するすべてのスキーマの定義と使い方については、『iPlanet Directory Server スキーマリファレンス』を参照してください。

注	適切なアクセス制御規則が設定されていない場合、ディレクトリは変更できません。ディレクトリのアクセス制御規則の作成方法については、第 6 章「アクセス制御の管理」を参照してください。

ルートエントリの作成

新しいデータベースを作成するたびに、データベースに格納される接尾辞をそのデータベースに関連付けます。ただし、その接尾辞を表すディレクトリエントリは自動的には作成されません。

データベースのルートエントリを作成するには、次の手順を実行します。

1. Directory Server Console で、「Configuration (構成)」タブを選択します。
2. 81 ページの「データベースの作成と管理」の説明に従って、新しいデータベースを作成します。
3. 「Directory (ディレクトリ)」タブで Directory Server を表す最上位オブジェクトをマウスの右ボタンでクリックし、「New Root Object (新規ルートオブジェクト)」を選択します。

   「New Root Object (新規ルートオブジェクト)」の 2 番目のメニューには、対応するエントリがない接尾辞が一覧表示されます。
4. 作成するエントリに対応する接尾辞を選択します。

   「New Object (新規オブジェクト)」ウィンドウが表示されます。
5. 「New Object (新規オブジェクト)」ウィンドウで、新しいエントリに対応するオブジェクトクラスを選択します。

   選択するオブジェクトクラスには、接尾辞を指定するときに使用した属性が含まれている必要があります。たとえば、接尾辞 `ou=people,dc=siroe,dc=com` に対応するエントリを作成する場合は、`organizationalUnit` オブジェクトクラス (または ou 属性を使用できる別のオブジェクトクラス) を選択できます。

6. 「New Object (新規オブジェクト)」ウィンドウで「OK」をクリックします。

   新しいエントリ用のプロパティエディタが表示されます。「OK」をクリックして現在の値をそのまま使用するか、あるいは 45 ページの「ディレクトリエントリの変更」の説明に従ってエントリを変更できます。

ディレクトリエントリの作成

Directory Server Console には、ディレクトリエントリの作成に使用できる事前に定義されたテンプレートがいくつか用意されています。テンプレートを使用して、次のタイプのエントリを作成できます。

- ユーザ
- グループ
- 組織単位
- ロール
- サービスクラス

表 2-1 は、各テンプレートで使用されるオブジェクトクラスのタイプを示しています。

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	inetOrgPerson
グループ	groupOfUniqueNames
組織単位	organizationalUnit
ロール	nsRoleDefinition
サービスクラス	cosSuperDefinition

これらのテンプレートには、すべての必須の属性と、共通して使用される任意の属性の一部を表すフィールドが含まれています。これらのテンプレートのいずれかを使用してエントリを作成する方法については、43 ページの「事前に定義されたテンプレート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、44 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。

事前に定義されたテンプレートを使用したエントリの作成

1. Directory Server Console で「Directory (ディレクトリ)」タブを選択します。

2. 左側の区画で、新しいエントリをその下に追加したいエントリをマウスの右ボタンでクリックします。「User (ユーザ)」、「Group (グループ)」、「Organizational Unit (組織単位)」、「Role (ロール)」、「Class of Service (サービスクラス)」、または「Other (その他)」から適切なエントリのタイプを選択します。

   対応する「Create (作成)」ウィンドウが表示されます。

3. すべての必須の属性値 (アスタリスクで示される) を指定し、必要な場合は、さらに任意の属性値を指定します。

   「Create (作成)」ウィンドウには、任意の属性がすべて表示されているわけではありません。

4. すべての属性のリストを表示するには、「Advanced (詳細)」ボタンをクリックします。

   プロパティエディタが表示されます。プロパティエディタの使い方については、45 ページの「ディレクトリエントリの変更」を参照してください。

5. 「OK」をクリックして、「Create (作成)」ウィンドウを閉じます。

   右側の区画に新しいエントリが表示されます。

その他のタイプのエントリの作成

1. Directory Server Console で「Directory (ディレクトリ)」タブを選択します。
2. 左側の区画で、新しいエントリの追加先のエントリをマウスの右ボタンでクリックし、「Other (その他)」を選択します。

   「New Object (新規オブジェクト)」ウィンドウが表示されます。

3. オブジェクトクラスのリストで、新しいエントリを定義するオブジェクトクラスを選択します。
4. 「OK」をクリックします。

   選択したオブジェクトクラスに関連付けられたエントリのタイプに対して、事前に定義されたテンプレートを使用できる場合は、対応する「Create (作成)」ウィンドウが表示されます (43 ページの「事前に定義されたテンプレートを使用したエントリの作成」を参照)

   それ以外の場合は、プロパティエディタが表示されます。プロパティエディタには、必須の属性のリストが表示されます。

5. 一覧表示されるすべての属性に対して、属性値を指定します。

   空白のフィールドや、一般的なプレースホルダ値 (仮の値)(**New** など) が含まれているフィールドもあります。これらのフィールドには、使用するエントリにとって有効な値を入れる必要があります。

   一部のオブジェクトクラスは、複数の命名属性を持つことができます。新しいエントリに名前を付けるために使用する命名属性を選択する必要があります。

   リストされない任意の属性値を指定する方法については、45 ページの「ディレクトリエントリの変更」を参照してください。

6. 「OK」をクリックして新しいエントリを保存し、プロパティエディタウィンドウを閉じます。

   右側の区画に新しいエントリが表示されます。

ディレクトリエントリの変更

Directory Server Console でディレクトリエントリを変更するには、プロパティエディタを起動する必要があります。プロパティエディタには、エントリに属するオブジェクトクラスと属性のリストが表示されます。

プロパティエディタでは、次の操作を実行できます。

- エントリにオブジェクトクラスを追加する
- エントリからオブジェクトクラスを削除する
- エントリに属性を追加する
- エントリに属性値を追加する
- エントリから属性値を削除する
- エントリに属性のサブタイプを追加する

ここでは、プロパティエディタを起動する方法と、プロパティエディタを使用してエントリの属性と属性値を変更する方法について説明します。

プロパティエディタの表示

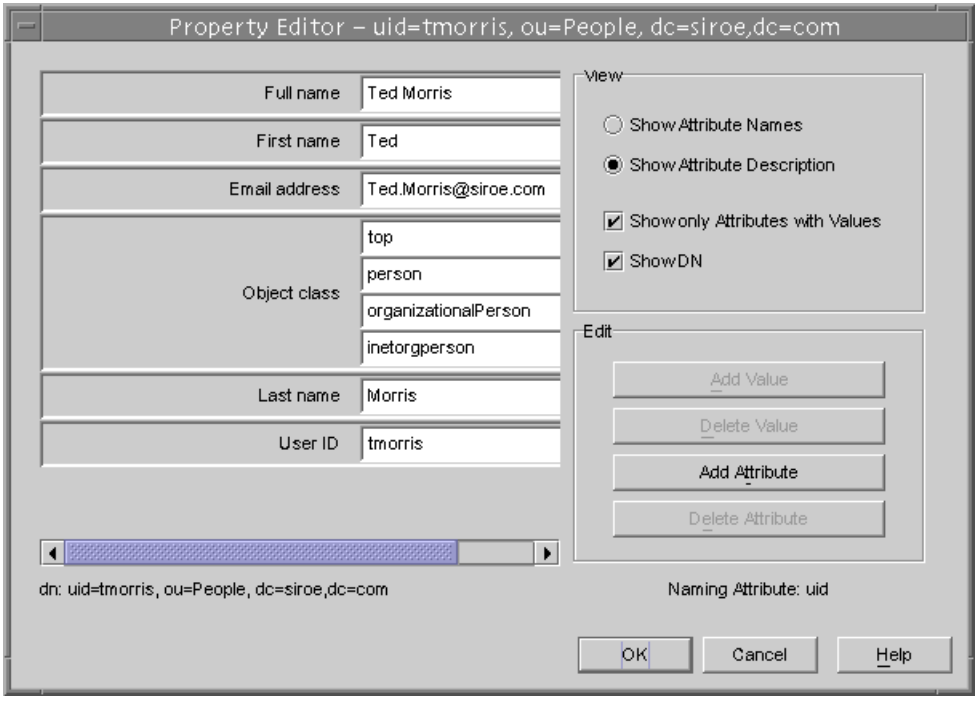
プロパティエディタは、次の方法で起動できます。

- 「Directory (ディレクトリ)」タブで、左側または右側の区画にある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「Properties (属性)」を選択する
- 「Directory (ディレクトリ)」タブで、左側または右側の区画にあるエントリをダブルクリックする

- ユーザ、グループ、組織単位、ロール、およびサービスクラスを作成するための各テンプレートで、「Advanced (詳細)」ボタンをクリックする (43 ページの「事前に定義されたテンプレートを使用したエントリの作成」を参照)
- 「New Object (新規オブジェクト)」ウィンドウで、「OK」をクリックする (44 ページの「その他のタイプのエントリの作成」を参照)

図 2-1 は、inetorgperson を説明するエントリの例を示すプロパティエディタです。

図 2-1 Directory Server Console - プロパティエディタ



エントリへのオブジェクトクラスの追加

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. Directory Server Console の「Directory (ディレクトリ)」タブで、変更する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「Property (プロパティ)」を選択します。

   あるいは、エントリをダブルクリックしても、プロパティエディタが表示されます。

2. オブジェクトクラスのフィールドを選択し、「Add Value (値の追加)」をクリックします。

   「Add Object Class (オブジェクトクラスの追加)」ウィンドウが表示されます。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。

3. 追加するオブジェクトクラスを選択し、「OK」をクリックします。

   選択したオブジェクトクラスが、プロパティエディタ内のオブジェクトクラスのリストに表示されます。「Add Object Class (オブジェクトクラスの追加)」ウィンドウでの指定を取り消すには、「Cancel (取消し)」をクリックします。

4. エントリの編集が完了したら、プロパティエディタで「OK」をクリックします。

   プロパティエディタが閉じます。

オブジェクトクラスの削除

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. Directory Server Console の「Directory (ディレクトリ)」タブで、変更する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「Property (プロパティ)」を選択します。

   あるいは、エントリをダブルクリックしても、プロパティエディタが表示されます。

2. 削除対象のオブジェクトクラスが表示されたテキストボックス内をクリックし、「Delete Value (値の削除)」をクリックします。

3. エントリの編集が完了したら、プロパティエディタで「OK」をクリックします。

   プロパティエディタが閉じます。

エントリへの属性の追加

エントリに属性を追加するには、必須のオブジェクトクラスか許可された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、46 ページの「エントリへのオブジェクトクラスの追加」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. Directory Server Console の「Directory (ディレクトリ)」タブで、変更する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「Property (プロパティ)」を選択します。

   あるいは、エントリをダブルクリックしても、プロパティエディタが表示されます。

2. 「Add Attribute (属性の追加)」をクリックします。

   「Add Attribute (属性の追加)」ダイアログボックスが表示されます。

3. リストから追加する属性を選択し、「OK」をクリックします。

   「Add Attribute (属性の追加)」ウィンドウが閉じ、選択した属性がプロパティエディタ内のリストに表示されます。

4. 属性名の右側にあるテキストボックスに新しい属性値を入力します。

5. エントリの編集が完了したら、プロパティエディタで「OK」をクリックします。

   プロパティエディタが閉じます。

属性値の追加

エントリに複数値属性が含まれている場合は、その属性に対して複数の値を指定できます。

複数値属性に属性値を追加するには、次の手順を実行します。

1. Directory Server Console の「Directory (ディレクトリ)」タブで、変更する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「Property (プロパティ)」を選択します。

   あるいは、エントリをダブルクリックしても、プロパティエディタが表示されます。

2. 値の追加先属性を選択し、「Add Attribute (属性の追加)」をクリックします。

   右側の列に新しい空白のテキストフィールドが表示されます。

3. 新しい属性値の名前を入力します。

4. エントリの編集が完了したら、プロパティエディタで「OK」をクリックします。

   プロパティエディタが閉じます。

属性値の削除

エントリから属性値を削除するには、次の手順を実行します。

1. Directory Server Console の「Directory (ディレクトリ)」タブで、変更する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「Property (プロパティ)」を選択します。

   あるいは、エントリをダブルクリックしても、プロパティエディタが表示されます。

2. 削除対象の属性値が表示されたテキストボックス内をクリックし、「Delete Value (値の削除)」をクリックします。

   属性全体とその値をすべてエントリから削除する場合は、「Edit (編集)」メニューの「Delete Attribute (属性の削除)」を選択します。

3. エントリの編集が完了したら、プロパティエディタで「OK」をクリックします。

   プロパティエディタが閉じます。

属性のサブタイプの追加

エントリに含まれる属性には、言語、バイナリ、および読みという 3 つのサブタイプを追加できます。

言語サブタイプ

ユーザ名をデフォルト言語以外の文字で表記する方が正確な場合があります。たとえば、Noriko という名前を、可能な場合は日本語の文字で表示してほしいと希望しているとします。この場合、`givenname` 属性の言語サブタイプとして「Japanese」を選択すると、ほかのユーザが彼女の名前を日本語で検索できるようになります。

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-`*subtype*

attribute はエントリに追加する属性です。*subtype* は言語を表す 2 文字の略語です。サポートされている言語サブタイプのリストについては、515 ページの表 D-2 を参照してください。たとえば、次のようにします。

```
givenname;lang-ja
```

エントリにある各属性インスタンスには、言語サブタイプを 1 つだけ割り当てることができます。複数の言語サブタイプを割り当てるには、エントリに別の属性インスタンスを追加してから、新しい言語サブタイプを割り当てます。たとえば、次のように指定すると無効になってしまいます。

```
cn;lang-ja;lang-en-GB:Smith
```

複数の言語サブタイプを割り当てるには、次のように指定します。

```
cn; lang-ja: ja_value
cn; lang-en-GB: en-GB_value
```

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形式であることを示します。`usercertificate;binary` はその例です。

ただし、binary サブタイプを含まない属性 (たとえば jpegphoto など) にもバイナリデータを格納することができます。つまり、binary サブタイプは、クライアントに対して複数の異なった属性タイプが存在することを示しています。

発音サブタイプ

属性に発音サブタイプを割り当てることによって、その属性値が発音表記であることを示します。このサブタイプは、attribute;phonetic のように属性名に追加されます。

このサブタイプは、複数の表記を持ち、その一方が発音表記である言語の言語サブタイプと組み合わせて、広く使用されます。

cn または givenname などのユーザ名を含む属性で使用する場合があります。たとえば、givenname;lang-ja;phonetic は、属性値がエントリの日本語名の読みであることを示します。

プロパティエディタを使用したサブタイプの追加手順

1. Directory Server Console の「Directory (ディレクトリ)」タブで、変更する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「Property (プロパティ)」を選択します。

   あるいは、エントリをダブルクリックしても、プロパティエディタが表示されます。

2. 「Add Attribute (属性の追加)」をクリックします。

   「Add Attribute (属性の追加)」ダイアログボックスが表示されます。

3. リストから追加する属性を選択します。

4. 属性に言語サブタイプを割り当てるには、「Language (言語)」ドロップダウンリストから言語サブタイプを選択します。

5. さらに、「Subtype (サブタイプ)」ドロップダウンリストから、バイナリまたは読みのどちらかのサブタイプを割り当てることができます。

6. 「OK」をクリックします。

   「Add Attribute (属性の追加)」ウィンドウが閉じます。

7. エントリの情報の定義が完了したら、プロパティエディタで「OK」をクリックします。

ディレクトリエントリの削除

Directory Server Console 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 Console で「Directory (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または右側の区画で削除対象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「Delete (削除)」を選択します。

   複数のエントリを選択する場合は、Ctrl キーまたは Shift キーを押したままエントリをクリックし、「Edit (編集)」メニューの「Delete (削除)」を選択します。

   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。

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を使用して、ディレクトリの内容を操作できます。コマンド行ユーティリティは、ディレクトリを一括して管理するスクリプトや、Directory Server をテストするスクリプトを記述する場合に便利です。たとえば、アクセス制御情報を変更したあとで、期待どおりの情報が返されることを確認する場合などがあります。

コマンド行ユーティリティを使用すると、コマンド行からの直接の情報入力や、LDIF 形式の入力ファイルを経由した情報の入力ができます。

ここでは、次の項目について説明します。

- コマンド行からの入力
- コマンド行からのルートエントリの作成
- LDIF を使用したエントリの追加
- ldapmodify を使用したエントリの追加と修正
- ldapdelete を使用したエントリの削除
- 特殊文字の使い方

注 適切なアクセス制御規則が設定されていない場合、ディレクトリは変更できません。ディレクトリのアクセス制御規則の作成方法については、第 6 章「アクセス制御の管理」を参照してください。

コマンド行からの入力

`ldapmodify` ユーティリティと `ldapdelete` ユーティリティに直接コマンド行から入力する場合は、LDIF 文を使用する必要があります。LDIF 文については、57 ページの「LDIF 更新文」を参照してください。

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、ユーザが入力した文を読み取ります。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。続いて、入力した内容に従って、処理が開始されます。

EOF エスケープシーケンスは、Control+D (^D) です。たとえば、ldapmodify に、複数の LDIF 更新文を入力すると仮定します。この場合、UNIX システムでは次のように指定します。

```
prompt> ldapmodify -D bindDN -w password -h hostname
>dn: cn=Barry Nixon, ou=people, dc=siroe,dc=com
> changetype: modify
> delete: telephonenumber
> -
> add: manager
> manager: cn=Harry Cruise, ou=people, dc=siroe,dc=com
> ^D
prompt>
```

コマンド行または LDIF を使用してエントリを追加する場合、サブツリーを表すエントリを作成してから、その分岐の下に新しいエントリを作成する必要があります。たとえば、People サブツリー内にエントリを配置する場合は、このサブツリーを表すエントリを作成してから、サブツリー内にエントリを作成します。

たとえば、次のようにします。

```
dn:dc=siroe,dc=com
dn:ou=People, dc=siroe,dc=com
...
People subtree entries.
...
dn: ou=Group, dc=siroe,dc=com
...
Group subtree entries.
...
```

コマンド行からのルートエントリの作成

ldapmodify コマンド行ユーティリティを使用して、データベース内に新しいルートエントリを作成できます。たとえば、次のように新しいルートエントリを追加します。

```
prompt% ldapmodify -a -D "dn=directory manager" -w secret
```

ldapmodify ユーティリティはサーバにバインドして、サーバにエントリを追加させる準備を行います。

新しいルートオブジェクトは次のように作成します。

dn: *Suffix_Name*
objectclass: *newobjectclass*

DN は、ルートの DN、またはデータベースに含まれるサブ接尾辞に対応します。*newobjectclass* の値は、データベースに追加するオブジェクトクラスのタイプによって決まります。追加するルートオブジェクトによっては、必須の属性の追加が必要となる場合もあります。

注 この方法は、接尾辞ごとに 1 つのデータベースがある場合にだけ有効です。複数のデータベースに格納される接尾辞を作成する場合は、`ldif2db` ユーティリティで `-n` オプションを使用して、新しいエントリを格納するデータベースを指定する必要があります。詳細は、140 ページの「コマンド行からのインポート」を参照してください。

LDIF を使用したエントリの追加

LDIF ファイルを使用すると、複数のエントリの追加や、データベース全体のインポートができます。LDIF ファイルと Directory Server Console 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. LDIF ファイル内にエントリを定義します。

   LDIF の詳細については、付録 A「LDIF (LDAP Data Interchange Format)」を参照してください。

2. Directory Server Console から LDIF ファイルをインポートします。

   詳細は、137 ページの「Console を使用したインポートの実行」を参照してください。LDIF ファイルをインポートしたあとで、「Import (インポート)」ダイアログボックスの「Append to database (データベースに追加)」を選択して、現在ディレクトリ内に存在していないエントリだけがインポートされるようにします。

`ldapmodify` コマンドに `-f` オプションを指定して実行すると、LDIF ファイルに記述されたエントリを追加できます。

ldapmodify を使用したエントリの追加と修正

既存の Directory Server データベースに対するエントリの追加と変更には、`ldapmodify` コマンドを使用します。`ldapmodify` コマンドは、ユーザが指定した識別名とパスワードを使用して指定したサーバへの接続を確立し、指定したファイル内に含まれる LDIF 更新文に基づいてエントリを変更します。`ldapmodify` は LDIF 更新文を使用するので、`ldapdelete` で実行できる処理はすべて `ldapmodify` でも実行できます。

このユーティリティの使用時にスキーマの検査がオンになっている場合は、エントリを変更すると、サーバによってエントリ全体のスキーマが検査されます。

- サーバが認識できない属性やオブジェクトクラスがエントリ内に見つかった場合は、エラーを含むエントリに到達した時点で変更操作が失敗する。ただし、エラーが発生する前に処理されたエントリは、すべて正しく追加または変更されている。-c オプションを指定して ldapmodify を実行した場合は、エラーが発生しても処理は停止しない。エラーを含むエントリのあとに記述されている正しいエントリは、すべて正常に追加または変更される。
- 必須の属性が存在しない場合にも、変更操作は失敗する。これは、問題のあるオブジェクトクラスや属性が変更対象ではない場合も同じ。このような状況は、スキーマの検査をオフにして Directory Server を実行し、認識できないオブジェクトクラスや属性を追加したあとに、スキーマの検査をオンにした場合に発生する可能性がある。

詳細は、342 ページの「スキーマ検査のオン / オフの切り替え」を参照してください。

ldapmodify を使用して、dc=siroe,dc=com などのデータベースの接尾辞を作成するには、ディレクトリマネージャとしてディレクトリにバインドする必要があります。

ldapmodify を使用したエントリの追加

次に、ldapmodify ユーティリティを使用してディレクトリにエントリを追加する方法の一般的な例を示します。この例では、次のように仮定しています。

- ファイル **new.ldif** に、エントリの作成情報を指定する。
- エントリを変更する権限を持つデータベース管理者を作成している。この管理者の識別名は、**cn=Directory Manager, dc=siroe,dc=com** である
- データベース管理者のパスワードは **King-Pin** である
- サーバは、**cyclops** という名前のマシンで稼動している
- サーバは、ポート番号 **845** を使用する

この例では、**new.ldif** ファイル内の LDIF 文には変更タイプを指定しません。LDIF 文は、473 ページの「LDIF ファイル形式」で定義した形式に従っています。

エントリを追加するには、次のコマンドを入力する必要があります。

```
ldapmodify -a -D "cn=Directory Manager,dc=siroe,dc=com" -w King-Pin
-h cyclops -p 845 -f new.ldif
```

ldapmodify を使用したエントリの変更

次に、ldapmodify ユーティリティを使用して、ディレクトリ内に存在するエントリを変更する方法の一般的な例を示します。この例では、次のように仮定しています。

- ファイル **modify_statements** に、エントリの変更情報を指定する

- エントリを変更する権限を持つデータベース管理者を作成している。この管理者の識別名は、**cn=Directory Manager, dc=siroe,dc=com** である。
- データベース管理者のパスワードは **King-Pin** である
- サーバは、**cyclops** という名前のマシンで稼動している
- サーバは、ポート番号 **845** を使用する

エントリを変更するには、適切な LDIF 更新文を含む `modify_statements` ファイルを作成して、次のコマンドを入力する必要があります。

```
ldapmodify -D "cn=Directory Manager,dc=siroe,dc=com" -w King-Pin -h
cyclops -p 845 -f modify_statements
```

ldapdelete を使用したエントリの削除

`ldapdelete` コマンド行ユーティリティを使用して、ディレクトリからエントリを削除します。このユーティリティは、ユーザが指定した識別名とパスワードを使用して、指定サーバへの接続を確立し、エントリを削除します。

削除できるエントリは、分岐の末端にあるエントリだけです。ディレクトリツリー内で分岐点になっているエントリは、ldapdelete では削除できません。

たとえば、次の 3 つのエントリがあるとします。

```
ou=People,dc=siroe,dc=com
cn=Paula Simon,ou=People,dc=siroe,dc=com
cn=Jerry O'Connor,ou=People,dc=siroe,dc=com
```

この中で削除できるのは後ろの 2 つのエントリだけです。People サブツリーを識別するエントリは、その下にエントリがない場合に限り削除できます。`ou=People,dc=siroe,dc=com` を削除する必要がある場合は、その前に Paula Simon と Jerry O'Connor のエントリおよびそのサブツリー内にあるほかのすべてのエントリを削除する必要があります。

次に、`ldapdelete` ユーティリティの一般的な使い方の例を示します。この例では、次のように仮定しています。

- `cn=Robert Jenkins,ou=People,dc=siroe,dc=com` および `cn=Lisa Jangles, ou=People,dc=siroe,dc=com` という 2 つの識別名によって識別されるエントリを削除する
- エントリを変更する権限を持つデータベース管理者を作成している。この管理者の識別名は、**cn=Directory Manager, dc=siroe,dc=com** である
- データベース管理者のパスワードは **King-Pin** である
- サーバは、**cyclops** という名前のマシンで稼動している

- サーバは、ポート番号 **845** を使用する

ユーザ Robert Jenkins と Lisa Jangles のエントリを削除するには、次のコマンドを入力します。

```
ldapdelete -D "cn=Directory Manager,dc=siroe,dc=com" -w King-Pin -h
cyclops -p 845 "cn=Robert Jenkins,ou=People,dc=siroe,dc=com"
"cn=Lisa Jangles,ou=People,dc=siroe,dc=com"
```

次の表に、この例で使用されている `ldapdelete` パラメタを示します。

パラメタ名	内容
`-D`	サーバに対する認証に使用する識別名を指定する。ここで指定する値は、Directory Server によって識別され、エントリを変更する権限を持つ DN でなければならない
`-w`	`-D` パラメタで指定された識別名に関連付けられているパスワードを指定する
`-h`	サーバが稼動しているホストの名前を指定する
`-p`	サーバが使用するポート番号を指定する

`ldapdelete` パラメタについては、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照してください。

特殊文字の使い方

Directory Server コマンド行クライアントツールを使用するときは、空白文字 ( )、アスタリスク (*)、バックスラッシュ (\) など、コマンド行インタプリタで特別な意味を持つ文字を含む値の指定が必要となることがあります。このような場合は、その値を引用符 ("") で囲みます。たとえば、次のようにします。

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

使用するコマンド行ユーティリティに応じて、一重引用符と二重引用符を使い分ける必要があります。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、バックスラッシュ (\) で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。

```
-D "cn=Patricia Fuentes,ou=people,o=siroe.com Bolivia\,S.A."
```

siroe.com Bolivia, S.A. ツリーから Patricia Fuentes を削除するには、次のコマンドを入力します。

```
ldapdelete -D "cn=Directory Manager,dc=siroe,dc=com" -w King-Pin -h
cyclops -p 845 "cn=Patricia Fuentes,ou=People,o=siroe.com
Bolivia\,S.A."
```

LDIF 更新文

ldapmodify によるディレクトリの変更方法を定義するには、LDIF 更新文を使用します。LDIF 更新文は、通常、次の操作を実行する一連のステートメントです。

- 変更するエントリの識別名を指定する
- add、delete、modify、modrdn など、特定のエントリの変更方法を定義する変更タイプを指定する
- 一連の属性とその変更後の値を指定する

ldapmodify に -a パラメタを指定した場合を除き、変更タイプを指定する必要があります。-a パラメタを指定すると、追加操作 (changetype: add) であると仮定されます。ただし、そのほかの変更タイプはすべて -a パラメタよりも優先されます。

修正操作 (changetype: modify) を指定した場合は、エントリの変更方法を指定する変更操作が必要です。

changetype: modrdn を指定する場合は、RDN (相対識別名) の修正方法を指定する変更操作が必要です。識別名の RDN は、DN 内の最も左端にある値です。たとえば、識別名 uid=ssarette,dc=siroe,dc=com の RDN は uid=ssarette です。

LDIF 更新文の一般的な形式は次のとおりです。

dn: *distinguished_name*
changetype_identifier
change_operation_identifier
list_of_attributes

-

change_operation_identifier
list_of_attributes
-

複数の変更操作を続けて指定する場合は、ダッシュ (-) を使用して各変更操作の終わりを示す必要があります。たとえば次のステートメントは、電話番号と管理者の属性をエントリに追加します。

```
dn: cn=Lisa Jangles,ou=People,dc=siroe,dc=com
changetype: modify
add: telephonenumber
telephonenumber: (408) 555-2468
-
add: manager
manager: cn=Harry Cruise,ou=People,dc=siroe,dc=com
```

また、行継続演算子として 1 つのスペースを使用します。したがって、次の 2 つの文は同じものになります。

```
dn: cn=Lisa Jangles,ou=People,dc=siroe,dc=com
```

```
dn: cn=Lisa Jangles,
 ou=People,
 dc=siroe,dc=com
```

次の節では、変更タイプについて詳しく説明します。

LDIF を使用したエントリの追加

ディレクトリにエントリを追加するには、`changetype: add` を使用します。エントリを追加する場合は、分岐点を表すエントリを作成してから、その分岐の下に新しいエントリを作成してください。つまり、People サブツリーと Groups サブツリー内にエントリを配置する場合は、これらのサブツリーの分岐点を作成してから、サブツリー内にエントリを作成します。

次の LDIF 更新文を使用して、People サブツリーと Groups サブツリーを作成し、次にそれらのサブツリー内にエントリを作成します。

```
dn:dc=siroe,dc=com
changetype: add
objectclass: top
objectclass:organization
o: siroe.com
```

```
dn:ou=People, dc=siroe,dc=com
changetype: add
objectclass: top
objectclass:organizationalUnit
ou:People
ou: Marketing
```

```
dn: cn=Pete Minsky,ou=People,dc=siroe,dc=com
changetype: add
objectclass: top
objectclass:person
objectclass: organizationalPerson
```

```
objectclass:inetOrgPerson
cn: Pete Minsky
givenName: Pete
sn: Minsky
ou:People
ou: Marketing
uid: pminsky

dn:cn=Sue Jacobs,ou=People,dc=siroe,dc=com
changetype: add
objectclass: top
objectclass:person
objectclass: organizationalPerson
objectclass:inetOrgPerson
cn: Sue Jacobs
givenName: Sue
sn: Jacobs
ou:People
ou: Marketing
uid: sjacobs

dn: ou=Groups,dc=siroe,dc=com
changetype: add
objectclass: top
objectclass:organizationalUnit
ou: Groups

dn: cn=Administrators,ou=Groups,dc=siroe,dc=com
changetype: add
objectclass: top
objectclass: groupOfNames
member:cn=Sue Jacobs,ou=People,dc=siroe,dc=com
member: cn=Pete Minsky,ou=People,dc=siroe,dc=com
cn: Administrators

dn: ou=siroe.com Bolivia\, S.A.,dc=siroe,dc=com
changetype: add
objectclass: top
objectclass:organizationalUnit
ou: siroe.com Bolivia\, S.A.

dn: cn=Carla Flores,ou=siroe.com Bolivia\, S.A.,dc=siroe,dc=com
changetype: add
objectclass: top
objectclass:person
objectclass: organizationalPerson
objectclass:inetOrgPerson
cn: Carla Flores
```

```
givenName: Carla
sn: Flores
ou: siroe.com Bolivia\, S.A.
uid: cflores
```

LDIF を使用したエントリ名の変更

エントリの RDN (相対識別名) を変更するには、changetype:modrdn を使用します。エントリの RDN は、識別名中の最も左端にある値です。たとえば、次のような識別名があるとします。

```
cn=Barry Nixon,ou=People,dc=siroe,dc=com
```

この場合、次の部分が RDN になります。

```
cn=Barry Nixon
```

また、次のような識別名があるとします。

```
ou=People,dc=siroe,dc=com
```

この場合、次の部分が RDN になります。

```
ou=People
```

したがって、この名前変更操作では、エントリの識別名の左端の値を変更できます。

たとえば、次のようなエントリがあるとします。

```
cn=Sue Jacobs,ou=People,dc=siroe,dc=com
```

このエントリを、次のように変更できます。

```
cn=Susan Jacobs,ou=People,dc=siroe,dc=com
```

ただし、次のように変更することはできません。

```
cn=Sue Jacobs,ou=old employees,dc=siroe,dc=com
```

次の例では、Sue Jacobs を Susan Jacobs に変更できます。

```
dn: cn=Sue Jacobs,ou=Marketing,dc=siroe,dc=com
changetype: modrdn
newrdn: cn=Susan Jacobs
deleteoldrdn: 0
```

この例では、deleteoldrdn が 0 なので、それまでの RDN が新しいエントリ内の値として残ります。このため、結果として生成されるエントリは、Sue Jacobs と Susan Jacobs の両方に設定された共通名 (cn) 属性と、元のエントリに含まれるその他すべての属性を持ちます。ただし、次のコードを使用した場合は事情が異なります。

```
dn: cn=Sue Jacobs,ou=Marketing,dc=siroe,dc=com
changetype: modrdn
newrdn: cn=Susan Jacobs
deleteoldrdn: 1
```

cn=Sue Jacobs はサーバから削除され、cn=Susan Jacobs だけがエントリ内に残ります。

エントリ名の変更に関する注意点

modrdn 変更タイプを使用してエントリ名を変更しても、エントリを異なるサブツリーに移動することはできません。エントリを異なる分岐に移動するには、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、分岐点になっているエントリは削除できないのと同じ理由で、子を持っているエントリの名前は変更できません。子を持つエントリの名前を変更すると、子のエントリが親のないエントリになります。これは LDAP プロトコルでは認められていません。たとえば、次の 3 つのエントリがあるとします。

```
ou=People,dc=siroe,dc=com
cn=Paula Simon,ou=People,dc=siroe,dc=com
cn=Jerry O'Connor,ou=People,dc=siroe,dc=com
```

この中で名前を変更できるのは後ろの 2 つのエントリだけです。People サブツリーを識別するエントリは、その下にエントリがない場合に限り、名前を変更できます。

LDIF を使用したエントリの変更

エントリの属性または属性値、あるいはその両方に対して追加、置換、削除を行うには、changetype:modify を使用します。changetype:modify を指定する場合は、エントリの修正方法を示す変更操作も指定する必要があります。次のような変更操作を指定できます。

- add: 属性

  指定した属性または属性値を追加します。その属性のタイプがエントリに含まれていない場合は、属性とそれに対応する値が作成されます。その属性のタイプがすでにエントリに含まれている場合は、指定した属性値が既存の値に追加されます。すでに特定の属性値がエントリに対して指定されている場合は、操作が失敗し、エラーが返されます。

- replace: 属性

  指定した値を使用して属性値全体を置き換えます。対象の属性が存在しない場合は、その属性が作成されます。置換する値を指定しない場合は、その属性が削除されます。

- `delete:` 属性

  指定した属性が削除されます。属性が複数の値を持っている場合は、エントリに含まれる属性の値がすべて削除されます。複数の属性値のうち 1 つだけを削除する場合は、delete 変更操作に続く行で、対象の属性とその属性に関連付けられた値を指定します。

この節では、次の事項について説明します。

- LDIF を使用した既存のエントリへの属性の追加
- LDIF を使用した属性値の変更
- LDIF を使用した 1 つの属性のすべての値の削除
- LDIF を使用した特定の属性値の削除

LDIF を使用した既存のエントリへの属性の追加

エントリに属性および属性値を追加するには、追加操作で `changetype:modify` を使用します。

たとえば、次の LDIF 更新文は、エントリに電話番号を追加します。

```
dn:cn=Barney Fife,ou=People,dc=siroe,dc=com
changetype: modify
add: telephonenumber
telephonenumber: 555-1212
```

次の例は、エントリに 2 つの電話番号を追加します。

```
dn:cn=Barney Fife,ou=People,dc=siroe,dc=com
changetype: modify
add: telephonenumber
telephonenumber: 555-1212
telephonenumber: 555-6789
```

次の例は、2 つの `telephonenumber` 属性と 1 つの `manager` 属性をエントリに追加します。

```
dn:cn=Barney Fife,ou=People,dc=siroe,dc=com
changetype: modify
add: telephonenumber
telephonenumber: 555-1212
telephonenumber: 555-6789
-
add: manager
manager: cn=Sally Nixon,ou=People,dc=siroe,dc=com
```

LDIF を使用した属性値の変更

エントリ内の 1 つの属性の値すべてを変更するには、replace 操作で `changetype:modify` を使用します。

たとえば、次の LDIF 更新文は、Barney の管理者を Sally Nixon から Wally Hensford に変更します。

```
dn:cn=Barney Fife,ou=People,dc=siroe,dc=com
changetype: modify
replace: manager
manager: cn=Wally Hensford, ou=People, dc=siroe,dc=com
```

エントリが対象の属性のインスタンスを複数含んでいる場合に、属性値の 1 つを変更するには、変更する属性値を削除してから、置換用の値を追加する必要があります。たとえば、次のようなエントリがあるとします。

```
cn=Barney Fife,ou=People,dc=siroe,dc=com
objectClass:inetOrgPerson
cn: Barney Fife
sn: Fife
telephonenumber: 555-1212
telephonenumber: 555-5678
```

電話番号を 555-1212 から 555-4321 に変更するには、次の LDIF 更新文を使用します。

```
dn:cn=Barney Fife,ou=People,dc=siroe,dc=com
changetype: modify
delete: telephonenumber
telephonenumber: 555-1212
-
add: telephonenumber
telephonenumber: 555-4321
```

Barney のエントリは次のようになります。

```
cn=Barney Fife,ou=People,dc=siroe,dc=com
objectClass:inetOrgPerson
cn: Barney Fife
sn: Fife
telephonenumber: 555-5678
telephonenumber: 555-4321
```

LDIF を使用した 1 つの属性のすべての値の削除

エントリから属性を削除するには、delete 操作で `changetype:modify` を使用します。エントリが対象の属性のインスタンスを複数含んでいる場合は、削除対象の属性のインスタンスを指定する必要があります。

たとえば、次の LDIF 更新文は、telephonenumber 属性のすべてのインスタンスをエントリから削除します。この属性がエントリ内で何度使用されているかは考慮されません。

```
dn:cn=Barney Fife,ou=People,dc=siroe,dc=com
changetype: modify
delete: telephonenumber
```

telephonenumber 属性の特定のインスタンスだけを削除する場合は、単純にその特定の属性値を削除します。次の節では、属性値を削除する方法について説明します。

LDIF を使用した特定の属性値の削除

エントリから属性値を削除するには、delete 操作で changetype:modify を使用します。

たとえば、次のようなエントリがあるとします。

```
cn=Barney Fife,ou=People,dc=siroe,dc=com
objectClass:inetOrgPerson
cn: Barney Fife
sn: Fife
telephonenumber: 555-1212
telephonenumber: 555-5678
```

電話番号 555-1212 をエントリから削除するには、次の LDIF 更新文を使用します。

```
dn:cn=Barney Fife,ou=People,dc=siroe,dc=com
changetype: modify
delete: telephonenumber
telephonenumber: 555-1212
```

この結果、Barney のエントリは次のようになります。

```
cn=Barney Fife,ou=People,dc=siroe,dc=com
objectClass:inetOrgPerson
cn: Barney Fife
sn: Fife
telephonenumber: 555-5678
```

LDIF を使用したエントリの削除

ディレクトリからエントリを削除するには、changetype:delete を使用します。削除できるエントリは最下位の部分のエントリだけです。したがって、エントリを削除するときは、ディレクトリツリー内で対象のエントリの下にほかのエントリがないことを確認します。つまり、組織単位に属するすべてのエントリを先に削除しないと、組織単位エントリは削除できません。

たとえば、次の 3 つのエントリがあるとします。

```
ou=People,dc=siroe,dc=com
cn=Paula Simon,ou=People,dc=siroe,dc=com
cn=Jerry O'Connor,ou=People,dc=siroe,dc=com
```

この中で削除できるのは後ろの 2 つのエントリだけです。People サブツリーを識別するエントリは、その下にエントリがない場合に限り削除できます。

次の LDIF 更新文を使用して、個人のエントリを削除できます。

```
dn: cn=Pete Minsky,ou=People,dc=siroe,dc=com
changetype: delete

dn:cn=Sue Jacobs,ou=People,dc=siroe,dc=com
changetype: delete
```

警告 接尾辞 o=NetscapeRoot は削除しないでください。iPlanetAdministration Server は、この接尾辞を使用してインストールした iPlanet Server に関する情報を格納します。この接尾辞を削除すると、Directory Server を含むすべての iPlanet サーバの再インストールが必要になります。

国際化ディレクトリのエントリの変更

属性タイプの言語タブでは、英語以外の言語の値であることを指定します。ldapmodify コマンド行ユーティリティを使用して、言語タグに関連付けられた属性を変更する場合は、値と言語タグを正確に一致させる必要があります。これらが一致しないと、修正操作は失敗します。

たとえば、lang-fr の言語タグを持つ属性値を修正する場合は、次の例に示すように、修正操作に lang-fr を含める必要があります。

```
dn: bjensen,dc=siroe,dc=com
changetype: modify
replace: homePostalAddress;lang-fr
homePostalAddress;lang-fr: 34\, rue de Seine
```

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データベースメカニズムです。Directory Server では、参照整合性を使用して、ディレクトリ内の 1 つのエントリに対する更新を、そのエントリを参照するほかのすべてのエントリに正しく反映させることができます。

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。参照整合性が有効になっていないと、管理者が手動でユーザを削除するまで、ユーザはグループのメンバーとして残ってしまいます。Directory Server とその他のユーザとグループの管理をディレクトリに頼っている iPlanet 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、その操作が参照整合性ログファイルに記録されます。

`/var/ds5/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。

デフォルトでは、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、`member`、`uniquemember`、`owner`、および `seeAlso` 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、ユーザが自由に設定できます。次のように設定できます。

- 参照整合性による更新をレプリケーション更新履歴ログに記録する
- 更新間隔を変更する
- 参照整合性を適用する属性を選択する
- 参照整合性を無効にする

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。

- 専用のコンシューマサーバ (読み取り専用レプリカだけを含むサーバ) 上では、参照整合性検査プラグインを有効にすることができない
- 読み取り専用レプリカと読み書き可能レプリカの組み合わせを含むサーバ上では、参照整合性検査プラグインを有効にすることができない
- 読み書き可能レプリカだけを含むマスターサーバ上では、参照整合性検査プラグインを有効にすることができる
- マルチマスターレプリケーションのコンテキストでは、1 つのマスター上だけで参照整合性検査プラグインを有効にすることができる

サプライヤサーバの構成

前述の条件を満たすレプリケーション環境では、参照整合性検査プラグインを有効にすることができます。

1. 参照整合性検査プラグインを有効にします。

   この手順については、67 ページの「参照整合性の有効化 / 無効化の切り替え」を参照してください。

2. すべての整合性更新を更新履歴ログに記録するようにプラグインを構成します。

   この手順については、68 ページの「更新履歴ログへの更新の記録」を参照してください。

3. すべてのコンシューマサーバ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。

注	サプライヤサーバは参照整合性検査プラグインによって実施された変更を、すべてコンシューマサーバに送信します。このため、コンシューマサーバ上で参照整合性検査プラグインを実行する必要はありません。

参照整合性の有効化 / 無効化の切り替え

参照整合性は、Directory Server Console またはコマンド行から有効または無効にすることができます。

Directory Server Console の使用

1. Directory Server Console で、「Configuration (構成)」タブを選択します。

   Directory Server Console の起動方法については、26 ページの「iPlanet Directory Server Console の使用」を参照してください。

2. ナビゲーションツリー内の Plugins フォルダを展開し、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区画に表示されます。

3. プラグインを有効にする場合は「Enable plugin (プラグインを有効にする)」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。

4. 「Save (保存)」をクリックして、変更内容を保存します。

5. 変更を有効にするには、「Tasks (タスク)」タブで、「Restart the Directory Server (Directory Server を再起動する)」を選択します。

更新履歴ログへの更新の記録

`slapd-`*`serverID`*`/logs` ディレクトリの `referint` ファイルではなく、レプリケーションの更新履歴ログに更新情報を記録することもできます。レプリケーションの処理で、参照整合性の更新をコンシューマサーバにレプリケートするには、更新履歴ログに更新情報を記録する必要があります。

この変更は、Directory Server Console から行うことができます。

Directory Server Console の使用

1. Directory Server Console で、「Configuration (構成)」タブを選択します。

2. ナビゲーションツリー内の Plugins フォルダを展開し、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区画に表示されます。

3. 引数のリスト内で、ファイル名 `referint` を更新履歴ログディレクトリへの絶対パスに置き換えます。

4. 「Save (保存)」をクリックして、変更内容を保存します。

5. 変更を有効にするには、「Tasks (タスク)」タブで、「Restart the Directory Server (Directory Server を再起動する)」を選択します。

更新間隔の変更

デフォルトでは、delete 操作または modrdn 操作の直後に、サーバによって参照整合性更新が実行されます。この操作がシステムに与える影響を軽減するには、更新間隔を長くします。更新間隔の最大値は設定されていません。一般的には、次の更新間隔が使用されます。

- ただちに更新
- 90 秒 (90 秒ごとに更新)
- 3600 秒 (1 時間ごとに更新)
- 10.800 秒 (3 時間ごとに更新)
- 28,800 秒 (8 時間ごとに更新)
- 86,400 秒 (1 日に 1 回更新)
- 604,800 秒 (1 週間に 1 回更新)

更新間隔は、Directory Server Console から変更できます。

Directory Server Console の使用

1. Directory Server Console で、「Configuration (構成)」タブを選択します。
2. ナビゲーションツリー内の Plugins フォルダを展開し、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区画に表示されます。
3. 引数のリスト内で、最初のテキストボックスの値を適切な更新間隔に置き換えます。
4. 「Save (保存)」をクリックして、変更内容を保存します。
5. 変更を有効にするには、「Tasks (タスク)」タブで、「Restart the Directory Server (Directory Server を再起動する)」を選択します。

属性リストの変更

デフォルトでは、参照整合性は、member、uniquemember、owner、および seeAlso の各属性を更新するように設定されています。Directory Server Console を使用して、更新対象の属性を追加または削除できます。

Directory Server Console の使用

1. Directory Server Console で、「Configuration (構成)」タブを選択します。

2. ナビゲーションツリー内の Plugins フォルダを展開し、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区画に表示されます。

3. 「Arguments (引数)」セクションで、「Add (追加)」ボタンと「Delete (削除)」ボタンを使用して、リスト内の属性を変更します。

4. 「Save (保存)」をクリックして、変更内容を保存します。

5. 変更を有効にするには、「Tasks (タスク)」タブで、「Restart the Directory Server (Directory Server を再起動する)」を選択します。

注 Directory Server から最高の性能を引き出すためには、更新対象の属性に適切なインデックスを付ける必要があります。インデックス付けについては、第 10 章「インデックスの管理」を参照してください。

第 3 章

ディレクトリデータベースの構成

ディレクトリは複数のデータベースから構成されます。また、ディレクトリツリーは、データベース全体に分散させることができます。この章では、接尾辞の作成方法、ディレクトリツリーの分岐点、および各接尾辞 (suffix) に関連付けられているデータベースの作成方法について説明します。また、リモートサーバにあるデータベースを参照するデータベースリンクの作成方法や、レフェラルを使用して、クライアントにディレクトリデータの外部ソースをポイントさせる方法についても説明します。

この章は、次の節で構成されます。

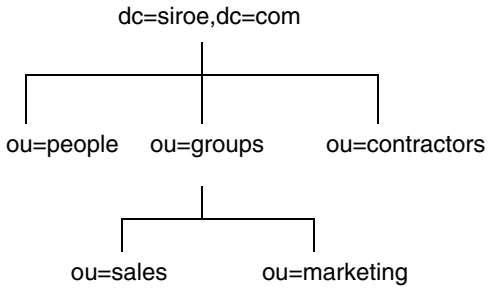
- 接尾辞の作成と管理
- データベースの作成と管理
- データベースリンクの作成と管理
- レフェラルの使い方

ディレクトリデータの分散の概念については、『iPlanet Directory Server 導入ガイド』を参照してください。

接尾辞の作成と管理

ディレクトリツリーの各部分をさまざまなデータベースに格納し、これらのデータベースを複数のサーバに分散させることができます。ディレクトリツリーには、ノードと呼ばれる分岐点があります。これらのノードには、必ずしもデータベースを関連付ける必要はありません。ノードは、Directory Server Console の「ディレクトリ」タブを使用して作成します。このタブでは、ディレクトリツリーに表示されるエントリを自由に編集できます。

接尾辞は、特定のデータベースに関連付けられているディレクトリツリーのノードです。この特殊なノードは、Directory Server Console の「データベース」タブを使用して作成します。次に、単純なディレクトリツリーの例を示します。



ou=people 接尾辞と、それ以下のすべてのエントリおよびノードは、1 つのデータベースに格納されます。ou=groups 接尾辞や ou=contractors 接尾辞はそれぞれ別のデータベースに格納されます。

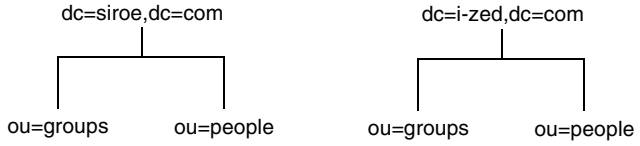
ここでは、Directory Server に接尾辞を作成し、これらの接尾辞をデータベースと関連付ける方法について、次の項目ごとに説明します。

- 72 ページの「接尾辞の作成」
- 78 ページの「接尾辞の管理」

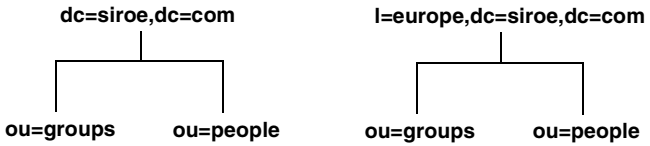
接尾辞の作成

ルート接尾辞とサブ接尾辞の両方を作成して、ディレクトリツリーの内容を編成できます。ルート接尾辞 (root suffix) は、サブ接尾辞 (sub suffix) の親です。Directory Server 用に設計された大きなツリーの一部になることができます。サブ接尾辞は、ルート接尾辞の下にある分岐です。ルート接尾辞とサブ接尾辞のデータはデータベースに格納されます。

ディレクトリに複数のルート接尾辞が含まれることもあります。たとえば、ある ISP が、siroe.com や i-zed.com など、複数の Web サイトにホスティングサービスを提供しているとします。ISP は、1 つは dc=siroe,dc=com 命名コンテクストに対応し、もう 1 つは dc=i-zed,dc=com 命名コンテクストに対応する、2 つのルート接尾辞を作成します。ディレクトリツリーは次のように表されます。

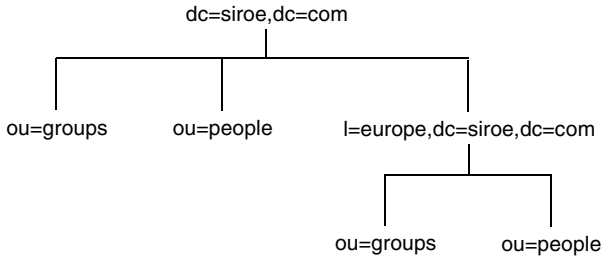


また、検索操作からディレクトリツリーの特定の部分を除外するように、ルート接尾辞を作成することもできます。たとえば、siroe.com Corporation が、企業全体のディレクトリ検索からヨーロッパオフィスを除外することを希望しているとします。このためには、2 つのルート接尾辞を作成します。1 つは siroe.com Corporation 全体のディレクトリツリーである dc=siroe,dc=com に対応するルート接尾辞で、もう 1 つはディレクトリツリーのヨーロッパ分岐 l=europe,dc=siroe,dc=com に対応するルート接尾辞です。クライアントアプリケーションからは、このディレクトリツリーは次のように見えます。



siroe.com Corporation ディレクトリの dc=siroe,dc=com 分岐に対してクライアントアプリケーションが検索を実行しても、l=europe,dc=siroe,dc=com 分岐にあるエントリは返されません。これは、ルート接尾辞が異なっているからです。

siroe.com Corporation が、ディレクトリツリーのヨーロッパ分岐のエントリを全体検索に含めると決めた場合は、この分岐を全体分岐のサブ接尾辞にする必要があります。このためには、siroe.com Corporation のルート接尾辞 dc=siroe,dc=com を作成してから、この下にヨーロッパディレクトリエントリのサブ接尾辞 l=europe,dc=siroe,dc=com を作成します。クライアントアプリケーションからは、このディレクトリツリーは次のように見えます。



ここでは、Directory Server Console またはコマンド行を使用して、ディレクトリにルート接尾辞とサブ接尾辞を作成する方法について説明します。次の手順について説明します。

- 「Console を使用した新しいルート接尾辞の作成」(74 ページ)
- 「Console を使用した新しいサブ接尾辞の作成」(74 ページ)
- 「コマンド行からのルート接尾辞およびサブ接尾辞の作成」(75 ページ)

Console を使用した新しいルート接尾辞の作成

接尾辞を作成し、これにデータベースを関連付ける手順は次のとおりです。

1. Directory Server Console で、「構成」タブを選択します。
2. 左側のナビゲーション区画で Data 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「新規ルート接尾辞」を選択します。

   「新規ルート接尾辞の作成」ダイアログボックスが表示されます。
3. 「新規接頭辞」フィールドに、一意の接尾辞名を入力します。

   接尾辞の名前は、dc (domain-component) 命名規則に従って付ける必要があります。たとえば、`dc=siroe,dc=com` という新しい接尾辞名を入力します。
4. 新しいルート接尾辞の作成時に現在のディレクトリにデータベースも作成する場合は、「関連するデータベースの自動作成」チェックボックスがデフォルトで選択されています。

   新しいルート接尾辞のデータベースを別のディレクトリに作成する場合または後で作成する場合は、このチェックボックスの選択を解除します。新しいルート接尾辞は、データベースが作成されるまで無効になっています。
5. 手順 4 で「関連するデータベースの自動作成」チェックボックスを選択した場合は、「データベース名」フィールドに新しいデータベースの一意な名前を入力します。

   この名前に使用できる文字は、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。たとえば、新しいデータベースに `siroe_2` という名前を付けることができます。
6. 「OK」をクリックして、新しいルート接尾辞を作成します。

   作成したルート接尾辞は、左側のナビゲーション区画にある Data 分岐の下に自動的に表示されます。

Console を使用した新しいサブ接尾辞の作成

既存のルート接尾辞またはサブ接尾辞の下にサブ接尾辞を作成する手順は次のとおりです。

1. Directory Server Console で、「構成」タブを選択します。
2. 左側のナビゲーション区画にある Data で、新しいサブ接尾辞の追加先となる接尾辞を選択します。この接尾辞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「新規サブ接尾辞」を選択します。

   「新規サブ接尾辞の作成」ダイアログボックスが表示されます。

3. 「新規接頭辞」フィールドに、一意の接尾辞を入力します。

   そのルート接尾辞の命名規則に従って接尾辞を指定する必要があります。ルート接尾辞は自動的に名前に追加されます。たとえば、dc=siroe,dc=com ルート接尾辞の下にサブ接尾辞 ou=groups を作成すると、このサブ接尾辞に ou=groups,dc=siroe,dc=com という名前が自動的に付けられます。

4. 新しいルート接尾辞の作成時に現在のディレクトリにデータベースも作成する場合は、「関連するデータベースの自動作成」チェックボックスがデフォルトで選択されています。

   新しいサブ接尾辞のデータベースを別のディレクトリに作成する場合または後で作成する場合は、このチェックボックスの選択を解除します。新しい接尾辞は、データベースが作成されるまで無効になっています。

5. 手順 4 で「関連するデータベースの自動作成」チェックボックスを選択した場合は、「データベース名」フィールドに新しいデータベースの一意な名前を入力します。

   この名前に使用できる文字は、ASCII (7 ビット) 英数字、ハイフン (-)、およびアンダースコア (_) のみです。たとえば、新しいデータベースに siroe_sub2 という名前を付けることができます。

6. 「OK」をクリックして、新しいサブ接尾辞を作成します。

   新しい接尾辞は、左側のナビゲーション区画にある Data ツリーのルート接尾辞の下に自動的に表示されます。

コマンド行からのルート接尾辞およびサブ接尾辞の作成

ldapmodify コマンド行ユーティリティを使用して、ディレクトリ構成ファイルに新しい接尾辞を追加します。この項目で説明している例では、接尾辞の構成情報は cn=mapping tree,cn=config エントリに格納されます。

注 dse.ldif ファイルの cn=config エントリの下には、エントリを作成しないようにしてください。単純で平面的な dse.ldif 構成ファイルの cn=config エントリは、通常のエントリと同じように拡張性が高いデータベースには格納されません。その結果、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、性能が低下します。

性能上の理由から、単純なユーザエントリを cn=config の下に格納することはお勧めできませんが、ディレクトリマネージャまたはレプリケーションマネージャ (サプライヤバインド DN) エントリなどの特別なユーザエントリを cn=config の下に格納すると、構成情報を一元化できるため便利です。

たとえば、ldapmodify ユーティリティを使用して、構成ファイルに新しいルート接尾辞を追加するとします。次のように入力して、ldapmodify を実行します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
```

ldapmodify ユーティリティはサーバにバインドし、構成ファイルにエントリを追加する準備を行います。

次のように入力して、siroe.com Corporation のルート接尾辞エントリを作成します。

```
dn: cn="dc=siroe,dc=com",cn=mapping tree,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsMappingTree
nsslapd-state: backend
nsslapd-backend: UserData
cn:dc=siroe,dc=com
```

このルート接尾辞の下に groups のサブ接尾辞を作成するには、ldapmodify コマンドを使用して、次のエントリを追加します。

```
dn: cn="ou=groups,dc=siroe,dc=com",cn=mapping tree,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsMappingTree
nsslapd-state: backend
nsslapd-backend: GroupData
nsslapd-parent-suffix: "dc=siroe,dc=com"
cn: ou=groups,dc=siroe,dc=com
```

注

Directory Server Console を使用して接尾辞を管理する場合には、空白文字の使い方を、コマンド行によってルート接尾辞やサブ接尾辞に名前を付けたときと同一にする必要があります。

たとえば、ルート接尾辞に ou=groups dc=siroe,dc=com という名前を付けた場合 (groups のあとに 2 つの空白が入っている) は、このルートの下に作成するサブ接尾辞にはすべて、ou=groups のあとに空白を 2 つ指定する必要があります。

次の表に、接尾辞エントリを構成するために使用される属性を示します。

表 3-1　接尾辞の属性

属性名	値
dn	接尾辞の DN を定義する。DN は二重引用符 (") で囲む。値は、cn="dc=domain,dc=com",cn=mapping tree, cn=config という形式になる。 この属性は必須
cn	エントリの相対 DN (RDN) を定義する この属性は必須
objectclass	エントリがルート接尾辞エントリであるか、サブ接尾辞エントリであるかを示す。この属性は常に nsMappingTree という値をとる この属性は必須
nsslapd-state	接尾辞がどのように操作を処理するかを決定する。この属性に指定できる値は次のとおり • backend: すべての操作の処理に、バックエンド (データベース) が使用される • disabled: 操作の処理に使用できるデータベースはない。クライアントアプリケーションの要求に対して、サーバから「No such search object」というエラーが返される • referral: この接尾辞への要求に対して、レフェラルが返される • referral on update: 更新要求以外のすべての操作に対して、データベースが使用される。更新要求ではレフェラルを受け取る デフォルト値は disabled
nsslapd-referral	接尾辞によって返される レプリカ (replica) の LDAP URL を定義する。この属性には、複数の値を指定できるが、1 つの値につき指定できるレフェラルは 1 つ。nsslapd-state 属性の値が referral または referral on update である場合に、この属性が必要

表 3-1　接尾辞の属性 (続き)

属性名	値
nsslapd-backend	要求の処理に使用されるデータベース、または データベースリンク (database link) の名前を指定する。この属性には複数の値を指定できるが、1 つの値で 1 つのデータベースまたはデータベースリンクを指定する。データベースリンクについては、88 ページの「データベースリンクの作成と管理」を参照 nsslapd-state 属性が backend または referral on update に設定されている場合、この属性は必須
nsslapd-distribution-plugin	カスタム分散関数とともに使用される共有ライブラリを指定する。nsslapd-backend 属性で複数のデータベースを指定した場合は、この属性は必須 カスタム分散関数については、81 ページの「データベースの作成と管理」を参照
nsslapd-distribution-funct	カスタム分散関数の名前を指定する。nsslapd-backend 属性で複数のデータベースを指定した場合は、この属性は必須 カスタム分散関数については、81 ページの「データベースの作成と管理」を参照
nsslapd-parent-suffix	サブ接尾辞の親エントリの DN を表す。デフォルトでは、この属性は存在しない。つまり、この接尾辞がルート接尾辞であるとみなされる たとえば、ルート接尾辞 dc=siroe,dc=com の下にサブ接尾辞 o=sales,dc=siroe,dc=com を作成する場合、サブ接尾辞の nsslapd-parent-suffix 属性に次の値を追加する nsslapd-parent-suffix: "dc=siroe,dc=com"

接尾辞の管理

ここでは、次の手順について説明します。

- 「接尾辞でのレフェラルの使い方」(79 ページ)
- 「更新操作中だけのレフェラルの有効化」(79 ページ)
- 「接尾辞の無効化」(80 ページ)
- 「接尾辞の削除」(80 ページ)

接尾辞でのレフェラルの使い方

レ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をポイントするように設定することができます。たとえば、接尾辞と関連付けられたデータベースが保守のためにオフラインになった場合でも、接尾辞にレフェラルを追加しておくことにより、クライアントが別のサーバをポイントするように設定することができます。

レフェラルの概要については、『iPlanet Directory Server 導入ガイド』を参照してください。

接尾辞でレフェラルを設定するには、次の手順を実行します。

1. Directory Server Console で、「構成」タブを選択します。
2. 左側の区画にある Data で、レフェラルの追加先となる接尾辞をクリックします。
3. 「接尾辞の設定」タブをクリックします。「レフェラルを使用する」ラジオボタンを選択します。
4. 「レフェラル」タブをクリックします。「新規レフェラルの入力」フィールドに LDAP URL を入力します。あるいは、「構築」をクリックすると、LDAP URL の作成がガイドされます。

   LDAP URL の構造については、付録 C「LDAP URLs」を参照してください。
5. 「追加」をクリックすると、レプリカ (replica) がリストに追加されます。

   複数のレフェラルを入力できます。クライアントアプリケーションからの要求に対応して、ディレクトリがレフェラルのリスト全体を返します。
6. 「保存」をクリックします。

更新操作中だけのレフェラルの有効化

クライアントアプリケーションから読み取り専用データベースへの更新要求と書き込み要求に対して、リダイレクトされるようにディレクトリを構成することができます。

たとえば、ディレクトリデータのローカルコピーがあり、その所有者がユーザ自身ではない場合に、更新操作のレフェラルを有効にしたとします。ここでこのデータを、検索には使用でき更新には使用できないようにするためには、更新要求中に限定して、レフェラルを有効にします。この設定により、クライアントアプリケーションからエントリの更新が要求されたときに限って、このデータを所有するサーバに対してクライアントが照会され、このサーバで修正要求が処理されるようになります。

更新操作中だけにレフェラルを有効にするには、次の手順を実行します。

1. Directory Server Console で、「構成」タブを選択します。
2. 左側の区画にある Data で、レフェラルの追加先となる接尾辞をクリックします。
3. 「接尾辞の設定」タブをクリックします。「更新時にレフェラルを使用する」ラジオボタンを選択します。

4. 「レフェラル」タブをクリックします。「新規レフェラルの入力」フィールドに LDAP URL を入力します。あるいは、「構築」をクリックすると、LDAP URL の作成がガイドされます。

   LDAP URL の構造については、付録 C「LDAP URLs」を参照してください。

5. 「追加」をクリックすると、レプリカ (replica) がリストに追加されます。

   複数のレフェラルを入力できます。クライアントアプリケーションからの要求に対応して、ディレクトリがレフェラルのリスト全体を返します。

6. 「保存」をクリックします。

接尾辞の無効化

しばしば、保守のためのデータベースの停止が必要になり、しかしそのデータベースに格納されているデータがレプリケートされていない場合があります。このような場合は、レフェラルを返すのではなく、このデータベースを受け持つ接尾辞を無効にすることができます。

接尾辞を無効にすると、クライアントアプリケーションが検索、追加、修正などの LDAP 処理を実行しても、この接尾辞に関連するデータベースのコンテンツは、クライアントアプリケーションからは見えなくなります。

接尾辞を無効にするには、次の手順を実行します。

1. Directory Server Console で、「構成」タブを選択します。
2. 左側のナビゲーション区画にある Data で、無効にする接尾辞を選択します。
3. 「接尾辞の設定」タブをクリックします。「この接尾辞を有効にする」チェックボックスの選択を解除します。

   保存する必要のある変更があることを示す赤い点が、「接尾辞の設定」タブに表示されます。

4. 「保存」をクリックします。

   指定した接尾辞が無効になります。

接尾辞の削除

接尾辞を削除する手順は次のとおりです。

警告	接尾辞を削除すると、この接尾辞に関連付けられているデータベースエントリやレプリケーション情報がすべて削除されます。

1. Directory Server Console で、「構成」タブを選択します。
2. 左側のナビゲーション区画にある Data で、削除する接尾辞を選択します。

3. 「オブジェクト」メニューの「削除」を選択します。

   また、この接尾辞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」を選択することもできます。

4. 選択した接尾辞と一緒に、その下にあるすべてのサブ接尾辞を削除する場合は、「この接尾辞とそのすべてのサブ接尾辞を削除する」を選択します。

   選択した接尾辞だけを削除し、そのサブ接尾辞は残しておく場合は、「この接尾辞だけを削除する」を選択します。

5. 「OK」をクリックして、接尾辞を削除します。

   Console 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。

データベースの作成と管理

接尾辞を作成してディレクトリデータを整理したあとで、ディレクトリデータを格納するためのデータベースを作成します。データベースは、ディレクトリデータを格納するために使用されます。

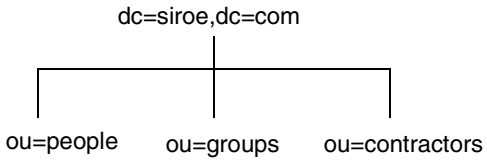
ここでは、ディレクトリデータを格納するためのデータベースの作成、データベースの削除、および一時的にデータベースを読み取り専用にする方法について説明します。

データベースの作成

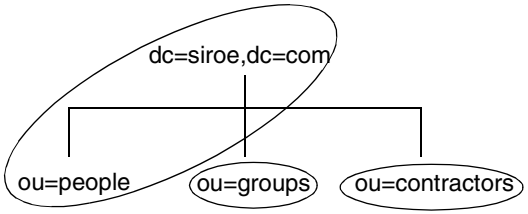
iPlanet Directory Server 5.1 では、複数のデータベースにわたって、ディレクトリツリーを分散させることができます。複数のデータベースにデータを分散させるには、次の 2 つの方法があります。

- 接尾辞 1 つにつき 1 つのデータベース

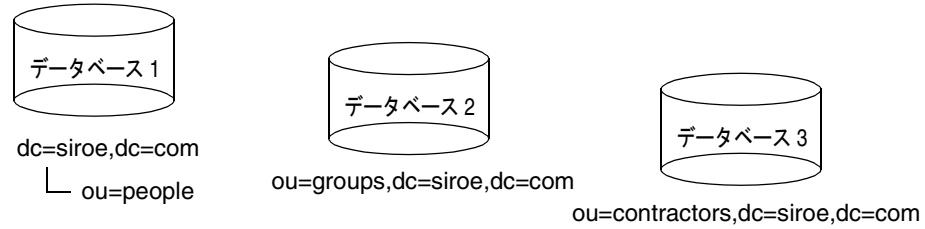
  各接尾辞のデータは、別々のデータベースに格納されます。たとえば、次のようなディレクトリツリーがあるとします。



  ここで、それぞれの接尾辞に含まれるデータを格納するためのデータベースを、次のように 3 つ追加します。



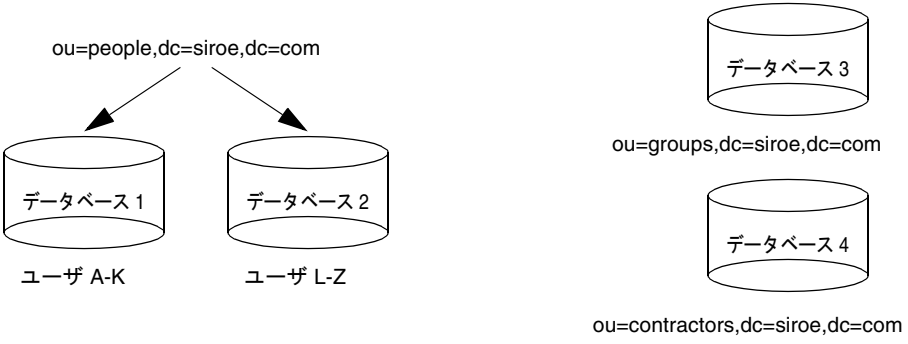
このようなツリーの分岐は、次の 3 つのデータベースに対応します。



データベース 1 には `ou=people` のデータと `dc=siroe,dc=com` のデータが含まれるので、クライアントは `dc=siroe,dc=com` に基づいて検索を実行できます。データベース 2 には `ou=groups` のデータ、データベース 3 には `ou=contractors` のデータが含まれています。

- 接尾辞 1 つに対して複数のデータベース

  たとえば、ディレクトリツリーの `ou=people` 分岐にあるエントリの数が多すぎるので、これらのエントリを 2 つのデータベースに分けて格納するとします。この場合、`ou=people` に含まれるデータは、2 つのデータベースに分散されます。これを図に示すと次のようになります。



データベース 1 には名前が A から K で始まる人、データベース 2 には L から Z で始まる人のデータが格納されます。データベース 3 には ou=groups のデータ、データベース 4 には ou=contractors のデータが含まれます。

カスタム分散プラグインを使用して、1 つの接尾辞から複数のデータベースにデータを分散させる必要があります。使用している Directory Server 用に分散論理を作成する方法については、iPlanet プロフェッショナルサービスにお問い合わせください。プロフェッショナルサービスについては、http://www.iplanet.com/services/ を参照してください。

Console を使用した既存の接尾辞に対する新しいデータベースの作成

すでに作成されている接尾辞にデータベースを追加する手順は次のとおりです。

1. Directory Server Console で、「構成」タブを選択します。
2. 左側の区画にある Data を展開し、新しいデータベースの追加先となる接尾辞をクリックします。
3. この接尾辞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「新規データベース」を選択します。

   「新規データベースの作成」ダイアログボックスが表示されます。
4. 「新規データベースの作成」ダイアログボックスで、データベースの一意の名前を入力します。

   この値には、コンマ、タブ、等号 (=)、アスタリスク (*)、バックスラッシュ (\)、スラッシュ (/)、プラス記号 (+)、一重引用符 (')、二重引用符 (")、および疑問符 (?) は使用できません。たとえば、新しいデータベースに siroe2 という名前を付けることができます。
5. 「データベースの作成位置」フィールドに、新しいデータベースの格納先ディレクトリへのパスを入力します。「参照」をクリックすると、ローカルマシン上のディレクトリを検索できます。

   デフォルトでは、新しいデータベースは、次のディレクトリに格納されます。

   /var/ds5/slapd-*serverID*/db
6. 「OK」をクリックします。確認のダイアログボックスで「はい」をクリックして、新しいデータベースを作成します。

注 「ディレクトリ」タブに新しい接尾辞を表示するには、まず、この接尾辞に関連付けられたルートエントリを作成する必要があります。43 ページの「ディレクトリエントリの作成」を参照してください。

コマンド行を使用した 1 つの接尾辞に対する新しいデータベースの作成

ldapmodify コマンド行ユーティリティを使用して、ディレクトリ構成ファイルに新しいデータベースを追加します。データベース構成情報は、cn=ldbm database,cn=plugins,cn=config エントリに格納されます。

たとえば、サーバ siroe1 に新しいデータベースを追加するとします。次のように ldapmodify を実行して、構成ファイルに新しいエントリを追加します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
```

ldapmodify ユーティリティはサーバにバインドし、構成ファイルにエントリを追加する準備を行います。

次のように入力して、新しいデータベースにエントリを作成します。

```
dn: cn=UserData,cn=ldbm database,cn=plugins,cn=config
objectclass: extensibleObject
objectclass: nsBackendInstance
nsslapd-suffix:ou=people,dc=siroe,dc=com
```

追加されたエントリは、ルート接尾辞またはサブ接尾辞 ou=people,dc=siroe,dc=com のデータを含むデータベース UserData に対応します。

コマンド行からルート接尾辞やサブ接尾辞を作成する方法については、75 ページの「コマンド行からのルート接尾辞およびサブ接尾辞の作成」を参照してください。DN 属性で指定されたデータベース名は、接尾辞エントリの nsslapd-backend 属性の値に対応する必要があります。

1 つの接尾辞に対する複数のデータベースの追加

複数のデータベース全体に、1 つの接尾辞を分散させることができます。ただし、接尾辞を分散させるには、カスタム分散関数を作成して、ディレクトリを展開する必要があります。カスタム分散関数の作成については、iPlanet プロフェッショナルサービスにお問い合わせください。プロフェッショナルサービスについては、http://www.iplanet.com/services/ を参照してください。

注 一度分散させたエントリは、再び分散させることはできません。この場合、次のような制限があります。

- 一度エントリを分散させたら、その分散関数を変更することはできない
- 別のデータベースにエントリが分散されてしまう場合は、LDAP `modrDN` 操作を使用してエントリ名を変更することはできない
- 分散済みのローカルデータベースをレプリケートすることはできない
- 別のデータベースにエントリが分散されてしまう場合は、`ldapmodify` 操作を使用してエントリを変更することはできない

これらの制約を守らない場合、iPlanet Directory Server で正しくエントリを見つけたり、返したりすることができなくなります。

iPlanet プロフェッショナルサービスがカスタム分散論理プラグインの作成をお手伝いします。あとは、このプラグインをディレクトリに追加するだけです。次に、ディレクトリ内の接尾辞に分散論理を追加する手順について説明します。

接尾辞へのカスタム分散関数の追加

分散論理は、接尾辞で宣言される関数です。この関数は、この接尾辞の上から開始されるサブツリー検索操作など、この接尾辞に影響する操作すべてに関して呼び出されます。Console とコマンド行のどちらを使用しても、接尾辞に分散関数を挿入することができます。

カスタム分散論理の作成については、iPlanet プロフェッショナルサービスにお問い合わせください。

Console を使用したカスタム分散の追加

1. Directory Server Console で、「構成」タブを選択します。
2. 左側にあるナビゲーション区画の Data ツリーを展開します。分散関数の適用先となる接尾辞を選択します。
3. 右側のウィンドウで「データベース」タブを選択します。
4. 「追加」をクリックして、追加データベースと接尾辞を関連付けます。

   「データベースリスト」ダイアログボックスが表示されます。データベースをリストから選択し、「OK」をクリックします。
5. 「分散ライブラリ」フィールドに分散ライブラリへのパスを入力します。あるいは、「参照」をクリックして、ローカルマシン上の分散ライブラリを選択します。
6. 「関数名」フィールドに分散関数の名前を入力します。

7. 「保存」をクリックして、変更内容を保存します。

コマンド行からのカスタム分散の追加

`ldapmodify` コマンド行ユーティリティを使用して、接尾辞エントリ自体に次の属性を追加します。

```
nsslapd-backend: Database1
nsslapd-backend: Database2
nsslapd-backend: Database3
nsslapd-distribution-plugin: /full/name/of/a/shared/library
nsslapd-distribution-funct: distribution-function-name
```

`nsslapd-backend` 属性は、この接尾辞に関連付けられているすべてのデータベースを表します。また、`nsslapd-distribution-plugin` 属性は、プラグインで使用されるライブラリ名を表します。`nsslapd-distribution-funct` 属性は、分散関数自体の名前を表します。

`ldapmodify` コマンド行ユーティリティの使い方については、53 ページの「ldapmodify を使用したエントリの追加と修正」を参照してください。

ディレクトリデータベースの管理

ここでは、ディレクトリデータベースの管理に関連する作業について、次の項目ごとに説明します。

データベースを読み取り専用モードへ設定

データベースが読み取り専用モードになっている場合、エントリの作成、変更、削除ができなくなります。たとえば、コンシューマを手作業で初期化する場合には、データベースを読み取り専用モードにしておく必要があります。

Directory Server で複数のデータベースを管理している場合は、サーバ全体を読み取り専用モードにすると、すべてのデータベースを一度に読み取り専用モードにできます。詳細は、37 ページの「Directory Server 全体を読み取り専用モードへ設定」を参照してください。

ここでは、次の手順について説明します。

Console を使用したデータベースの読み取り専用モード設定

Server Console からデータベースを読み取り専用モードに設定するには、次の手順を実行します。

1. Directory Server Console で、「構成」タブを選択します。
2. 左側の区画にある Data ツリーを展開します。読み取り専用モードにするデータベースが含まれる接尾辞を展開します。
3. 読み取り専用モードにするデータベースを選択します。
4. 右側の区画で「データベースの設定」タブを選択します。
5. 「データベースは読み取り専用です」チェックボックスを選択します。
6. 「保存」をクリックします。

コマンド行からのデータベースの読み取り専用設定

データベースを手作業で読み取り専用モードにする場合は、読み取り専用属性 `nsslapd-readonly` を on に変更する必要があります。この操作は、`ldapmodify` コマンド行ユーティリティを使用して行います。特定のデータベースの `nsslapd-readonly` 属性は `cn=`*`database_name`*`,cn=ldbm database,cn=plugins,cn=config` エントリにあります。ここで、*`database_name`* はデータベース名を表します。

注	デフォルトでは、インストール時に作成されるデータベース名は `userRoot` です。

データベースの削除

ここでは、Directory Server Console を使用してディレクトリデータベースを削除する手順について説明します。データベースを削除したときに削除されるのは、このデータベースの構成情報とエントリだけです。物理的なデータベース自体は削除されません。

1. Directory Server Console で、「構成」タブを選択します。
2. 左側のナビゲーション区画で削除するデータベースを選択します。
3. 「オブジェクト」メニューの「削除」を選択します。

   また、このデータベース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」を選択することもできます。

   「データベースを削除しています」確認ダイアログボックスが表示されます。

4. 「はい」をクリックして、データベースの削除を確認します。

   削除中に、Directory Server がどのような処理を実行しているかが、ダイアログボックスに表示されます。

   一度削除されたデータベースは、右側の区画には表示されなくなります。

データベースリンクの作成と管理

連鎖は、サーバがクライアントアプリケーションに代わって別のサーバと通信し、結果の組み合わせを返す方法です。この方法は、データベースリンク (database link) を介して実装されます。データベースリンクは、リモートに格納されているデータをポイントします。クライアントアプリケーションにより、データベースリンクのデータが要求されると、このデータベースリンクがリモートデータベースからデータを取得し、クライアントに返します。

ここでは、データベースの作成方法と設定方法について説明します。連鎖に関する全体的な説明については、『iPlanet Directory Server 導入ガイド』の「ディレクトリトポロジの設計」を参照してください。

Directory Server Console やコマンド行を使用して、データベースリンクを作成および構成することができます。以降では、データベースリンクの作成および管理の手順について説明します。

- 「連鎖ポリシーの構成」(89 ページ)
- 「新しいデータベースリンクの作成」(94 ページ)
- 「SSL を使用した連鎖」(105 ページ)
- 「データベースリンクの管理」(105 ページ)
- 「データベースリンクとアクセス制御の評価」(107 ページ)
- 「拡張機能 : データベースリンクの性能の調整」(108 ページ)
- 「拡張機能 : カスケード型連鎖の構成」(112 ページ)

データベースリンクのアクティビティの監視については、410 ページの「データベースリンクアクティビティの監視」を参照してください。

連鎖ポリシーの構成

ここでは、Directory Server が、クライアントアプリケーションからの要求を受け取り、データベースリンクを含む Directory Server に連鎖する方法の構成について説明します。この 連鎖 (chaining) ポリシーは、Directory Server に作成されたすべてのデータベースリンクに適用されます。

ここでは、次の項目について説明します。

- 「コンポーネント操作の連鎖」(89 ページ)
- 「LDAP 制御の連鎖」(93 ページ)

コンポーネント操作の連鎖

コンポーネントとは、内部処理を行うサーバ内の機能単位です。たとえば、プラグインはフロントエンドで機能するので、コンポーネントとみなされます。ただし、実際には、ACI プラグインのように、プラグインが複数のコンポーネントから構成されている場合があります。

コンポーネントの中には、ローカルデータだけにアクセスするために、内部 LDAP 要求をサーバに送信するものもあります。このようなコンポーネントでは、コンポーネントが正常に操作を完了できるように、連鎖ポリシーを制御する必要があります。たとえば、証明書を確認する関数について考えてみます。証明書を検査するために、この関数からの LDAP 要求を連鎖する場合は、リモートサーバを信頼していることを暗示しています。リモートサーバを信頼していない場合は、セキュリティに問題があります。

デフォルトでは、すべての内部処理が連鎖されるわけではありません。ただし、Console やコマンド行を使用して連鎖するコンポーネントを指定することによって、このデフォルトに優先することができます。デフォルトでは、コンポーネントの連鎖は禁止されています。

また、指定したプラグインがリモートサーバで動作するように、リモートサーバで ACI を作成する必要もあります。この ACI は、データベースリンクに割り当てられた接尾辞 (suffix) で作成します。

次の表に、コンポーネント名、コンポーネントを内部操作に連鎖させたときに発生することのある不具合、およびリモートサーバに作成した ACI で必要な権限を示します。

表 3-2　連鎖できるコンポーネント

コンポーネント名	説明	権限
ACI プラグイン	このプラグインでは、アクセス制御機能が実装される。ローカル ACI 属性とリモート ACI 属性の混在は危険なので、ACI 属性を取得する操作と更新する操作は連鎖できない。ただし、ユーザエントリを検出する要求は連鎖可能 nsActiveChainingComponents 属性に次の値を指定する `nsActiveChainingComponents: cn=ACI Plugin,cn=plugins,cn=config`	読み取り、検索、比較
4.0 プラグイン	このコンポーネント名は、Directory Server 4.0 プラグインすべてを表す。4.0 プラグインの連鎖ポリシーはすべて同じ。 nsActiveChainingComponents 属性に次の値を指定する `nsActiveChainingComponents: cn=old plugin,cn=plugins,cn=config`	連鎖が許可された 4.0 プラグインによって異なる
資源制限コンポーネント	このコンポーネントは、ユーザのバインドした DN に応じて、サーバ制限を設定する。資源制限コンポーネントの連鎖が許可されている場合は、リモートユーザに資源制限を適用することができる。このコンポーネントの操作を連鎖させるには、次のように指定する `nsActiveChainingComponents: cn=resource limits,cn=components,cn=config`	読み取り、検索、比較
証明書ベースの認証確認コンポーネント	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。リモートサーバにあるデータベースからユーザ証明書を取得する。このコンポーネントを連鎖すると、証明書に基づく認証をデータベースリンクで行うことができる。このコンポーネントの操作を連鎖させるには、次のように指定する `nsActiveChainingComponents: cn=certificate-based authentication,cn=components,cn=config`	読み取り、検索、比較

表 3-2　連鎖できるコンポーネント (続き)

コンポーネント名	説明	権限
参照整合性検査プラグイン	このプラグインは、DN を含む属性が変更されると、その属性へのポインタを含むすべてのエントリでその変更が反映されることを保証する。たとえば、グループのメンバーであるエントリを削除すると、そのエントリは自動的にグループから削除される。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静的グループ定義に対してリモートであるときに、静的グループの管理が簡単になる。 このコンポーネントの操作を連鎖させるには、次のように指定する `nsActiveChainingComponents: cn=referential integrity postoperation,cn=plugins,cn=config`	読み取り、書き込み、検索、比較
uid 一意性検査プラグイン	このプラグインでは、指定された uid 属性の値がすべて一意であるか、つまり重複する値がないかどうかが確認される。このプラグインを連鎖させると、データベースリンク経由で変更された属性でも、uid 属性値が一意であるかどうかが確認される。このコンポーネントの操作を連鎖させるには、次のように指定する `nsActiveChainingComponents:cn=uid uniqueness,cn=plugins,cn=config`	読み取り、検索、比較

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

ACI および連鎖の制限については、190 ページの「ACI の制限事項」を参照してください。

連鎖させるコンポーネントを変更したら、変更内容を有効にするためにサーバを再起動する必要があります。

次の節では、Console やコマンド行を使用して連鎖するコンポーネントを指定する方法について説明します。

Console を使用したコンポーネント操作の連鎖

1. Directory Server Console で、「構成」タブを選択します。
2. 左側の区画にある Data を展開し、「データベースリンクの設定」をクリックします。
3. 右側のウィンドウで「設定」タブを選択します。「連鎖可能なコンポーネント」リストにコンポーネントを追加するには、「追加」をクリックします。

   「追加するコンポーネント」ダイアログボックスが表示されます。コンポーネントをリストから選択し、「OK」をクリックします。
4. コンポーネントをリストから削除するには、該当するコンポーネントを選択し、「削除」をクリックします。
5. コンポーネントリストを修正すると、タブに赤い点が表示され、フィールド名がグレーに変わります。「保存」をクリックして、変更内容を保存します。

   変更を有効にするには、サーバを再起動します。

コンポーネントを連鎖させたら、操作の連鎖先となるリモートサーバで接尾辞に ACI を作成する必要があります。たとえば、参照整合性プラグインのために、次の ACI を作成します。

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

コマンド行からのコンポーネント操作の連鎖

構成ファイルの `cn=config,cn=chaining database,cn=plugins,cn=config` エントリにある `nsActiveChainingComponents` 属性を使用して、連鎖に含めるコンポーネントを指定することができます。

たとえば、参照整合性プラグインを使用して操作を連鎖させる場合は、データベースリンク設定ファイルに次の行を追加します。

```
nsActiveChainingComponents:cn=referential integrity postoperation,
 cn=components,cn=config
```

連鎖できるコンポーネントのリストについては、90 ページの表 3-2 を参照してください。

`nsActiveChainingComponents` 属性を修正したら、変更内容を有効にするためにサーバを再起動する必要があります。

コンポーネントを連鎖させたら、操作の連鎖先となるリモートサーバで接尾辞に ACI を作成する必要があります。たとえば、参照整合性コンポーネントのために、次の ACI を作成します。

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

LDAP 制御の連鎖

LDAP 制御による操作要求は、連鎖させないように指定することができます。デフォルトでは、次の制御による要求は、データベースリンクによってリモートサーバに転送されます。

- 管理 DSA : この制御は、レフェラルに従わないで、スマートレフェラルをエントリとして返します。スマートレフェラル自体を変更または削除できます。
- ループの検出 : この制御は、あるサーバが別のサーバと連鎖する回数を記録します。この回数が設定した値に達すると、ループが検出され、クライアントアプリケーションに通知されます。

  この制御の使い方については、120 ページの「ループの検出」を参照してください。
- サーバ側ソート : この制御は、属性値に従ってエントリをソートします。
- VLV (仮想リスト表示) : この制御は、検索結果としてすべてのエントリ情報を一度に返すのではなく、部分的な結果リストを提供します。

注 サーバ側ソート制御と VLV 制御は、検索範囲が 1 つのデータベースである場合にのみ、連鎖を介し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データベースに対して行われた場合は、データベースリンクでは、VLV 制御はサポートされません。

次の節では、Console やコマンド行を使用して、データベースリンクが転送する制御を変更する方法について説明します。

Console を使用した LDAP 制御の連鎖

1. Directory Server Console で、「構成」タブを選択します。
2. 左側の区画にある Data フォルダを展開し、「データベースリンクの設定」をクリックします。
3. 右側のウィンドウで「設定」タブを選択します。リストに LDAP 制御を追加するには、「追加」をクリックします。

   「追加するコントロール OID の選択」ダイアログボックスが表示されます。リストに追加する制御の OID を選択し、「OK」をクリックします。
4. リストから制御を削除するには、該当する制御を「リモートサーバに転送された LDAP 制御」リストから選択し、「削除」をクリックします。

5. コンポーネントリストを修正すると、タブに赤い点が表示され、コンポーネントのフィールド名がグレーに変わります。「保存」をクリックして、変更内容を保存します。

コマンド行からの LDAP 制御の連鎖

データベースリンクによって転送される制御を変更するには、`cn=config,cn=chaining database, cn=plugins,cn=config` エントリの `nsTransmittedControls` 属性を変更します。たとえば、VLV 制御を転送するには、構成ファイルのデータベースリンクエントリに、次の行を追加します。

```
nsTransmittedControls: 2.16.840.1.113730.3.4.9
```

さらに、Directory Server のクライアントによって、独自の制御が作成されているときに、これらの処理をリモートサーバに連鎖させる必要がある場合は、カスタム制御の OID を `nsTransmittedControls` 属性に追加する必要があります。

次の表に、連鎖可能な LDAP 制御と、その OID を示します。

表 3-3 LDAP 制御と OID

制御名	OID
VLV (仮想リスト表示)	2.16.840.1.113730.3.4.9
サーバ側ソート	1.2.840.113556.1.4.473
管理 DSA	2.16.840.1.113730.3.4.2
ループ検出	1.3.6.1.4.1.1466.29539.12

LDAP 制御については、http://docs.iplanet.com/docs/manuals/directory.html にある LDAP C-SDK に関するドキュメントを参照してください。

新しいデータベースリンクの作成

データベースリンクの基本構成には、次の情報が必要です。

接尾辞情報 : ディレクトリツリーに、通常のデータベースではなく、データベースリンクによって管理される接尾辞を作成します。この接尾辞は、データが含まれるリモートサーバの接尾辞に対応します。

バインド資格 : データベースリンクがリモートサーバにバインドするときに、データベースリンクはユーザであるかのように動作します。リモートサーバへのバインド操作のために、データベースリンクで使用される DN と資格を指定する必要があります。

LDAP URL : データベースリンクの接続先リモートサーバの LDAP URL を指定します。

フェイルオーバサーバのリスト : 不具合が発生したときに、データベースリンクの接続先となる代替サーバのリストを指定できます。この構成項目は省略可能です。

注 このオプションは、SSL を使用してセキュリティ保護されたリンクでは使用できません。

次の節では、Directory Server Console およびコマンド行から新しいデータベースリンクを作成する方法について説明します。

Console を使用した新しいデータベースリンクの作成

Directory Server Console を使用して新しいデータベースリンクを作成するには、次の手順を実行します。

1. Directory Server Console で、「構成」タブを選択します。
2. 左側のナビゲーション区画で Data 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「新規ルート接尾辞」または「新規サブ接尾辞」を選択します。

   「新規接尾辞の作成」ダイアログボックスが表示されます。
3. 「新規接尾辞」フィールドに、連鎖先となるリモートサーバの接尾辞名を入力します。

   接尾辞の名前は、dc 命名規則に従って付ける必要があります。たとえば、`dc=siroe,dc=com` という新しい接尾辞名を入力します。
4. 「関連するデータベースの自動作成」チェックボックスの選択を解除します。

   データベースに関連付けられている接尾辞にはデータベースリンクを追加できないので、このチェックボックスの選択を解除します。この接尾辞は、データベースリンクだけで使用されます。
5. 「OK」をクリックして、新しい接尾辞を作成します。

   作成した接尾辞は、左側のナビゲーション区画にある Data 分岐の下に自動的に表示されます。
6. 左側の区画で、作成した接尾辞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「新規データベースリンク」を選択します。

   「新規データベースリンクの作成」ダイアログボックスが表示されます。

7. 「データベースリンク名」フィールドに新しいデータベースリンク名を入力します。

   データベースリンクに名前を付ける場合に使用できる文字は、ASCII (7 ビット) 文字だけです。この値には、コンマ、タブ、等号 (=)、アスタリスク (*)、バックスラッシュ (\)、スラッシュ (/)、プラス記号 (+)、一重引用符 (')、二重引用符 (")、および疑問符 (?) は使用できません。たとえば、新しいデータベースリンクに `siroelink1` という名前を付けることができます。

8. 「バインド DN」フィールドに、データベースリンクがリモートサーバにバインドするときに使用する DN を入力します。

   たとえば、`cn=dblink` と入力します。

9. 「パスワード」フィールドに、データベースリンクがリモートサーバにバインドするときに使用するパスワードを入力します。

10. データベースリンクで、リモートサーバとの通信に SSL を使用する場合は、「サーバ間のセキュリティ保護された LDAP 接続を使用する」チェックボックスを選択します。

11. 「リモートサーバ」フィールドにリモートサーバ名を入力します。バインドに使用するサーバのポート番号を、「リモートサーバのポート」フィールドに入力します。デフォルトは `389` です。

12. フェイルオーバサーバの名前を「サーバをフェイルオーバする」フィールドに入力し、ポート番号を「ポート」フィールドに指定します。デフォルトは `389` です。「追加」をクリックすると、フェイルオーバサーバがリストに追加されます。

    この項目には、複数のフェイルオーバサーバを指定することができます。プライマリリモートサーバで不具合が発生した場合は、データベースリンクは「フェイルオーバサーバ」リストの最初のサーバに接続します。このサーバにも接続できない場合、リスト上のサーバに対して、リストの上から順番に接続していきます。

13. 「OK」をクリックして、新しいデータベースリンクを作成します。データベースリンクの作成が終了したことを示すダイアログボックスが表示されたら、「OK」をクリックしてこのダイアログボックスを閉じます。

    左側のナビゲーション区画にある接尾辞の下に、新しいデータベースリンクが表示されます。

ヒント Console には、データベースリンクが正常にバインドできるように、リモートサーバで必要とされる情報のチェックリストが用意されています。このチェックリストを表示するには、新しいデータベースリンクをクリックし、「認証」タブをクリックします。「リモートサーバのチェックリスト」ボックスに、チェックリストが表示されます。

コマンド行からのデータベースリンクの作成

ldapmodify コマンド行ユーティリティを使用して、コマンド行から新しいデータベースリンクを作成します。

新しいインスタンスは、cn=chaining database,cn=plugins, cn=config エントリにある必要があります。

デフォルトの構成属性は、cn=default config, cn=chaining database,cn=plugins,cn=config エントリに含まれています。これらの構成属性は、すべてのデータベースリンクに対して、作成時に適用されます。デフォルト構成に対する変更は、新しく作成されるデータベースリンクだけに反映されます。既存のデータベースリンクにあるデフォルトの構成属性を変更することはできません。

データベースリンクにはそれぞれ、独自の構成情報が含まれています。この情報は、cn=*database_link_name*,cn=chaining database,cn=plugins,cn=config のようなデータベースリンクエントリ自体に格納されています。構成属性については、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照してください。

ここでは、コマンド行からデータベースリンクを構成する次の手順について説明します。

- 「接尾辞情報の指定」(97 ページ)
- 「バインド資格の指定」(98 ページ)
- 「LDAP URL の指定」(100 ページ)
- 「フェイルオーバサーバのリストの指定」(100 ページ)
- 「カスケード型連鎖構成属性のまとめ」(120 ページ)
- 「データベースリンク構成例」(102 ページ)

接尾辞情報の指定

データベースリンクで管理される接尾辞を定義するには、nsslapd-suffix 属性を使用します。たとえば、社内のリモートサイトにある people の情報をポイントするデータベースリンクが必要な場合は、次の接尾辞情報を入力します。

```
nsslapd-suffix: l=Zanzibar,ou=people,dc=siroe,dc=com
```

接尾辞情報は、cn=*database_link_name*,cn=chaining database,cn=plugins,cn=config エントリに格納されます。

注 nsslapd-suffix 属性の作成後にこの属性に加えた変更を有効にするには、データベースリンクを含むサーバの再起動が必要です。

バインド資格の指定

リモートサーバに連鎖されるクライアントアプリケーションからの要求については、クライアントアプリケーションに対して特別なバインド資格を指定する必要があります。この指定によって、操作を連鎖させるために必要なプロキシ認証権限がリモートサーバに与えられます。バインド資格を指定しない場合は、データベースリンクは匿名でリモートサーバにバインドします。

バインド資格の供与は、次のステップで行われます。

1. リモートサーバで、次の手順を実行します。

   a. データベースリンクの管理ユーザを作成します。

      エントリの追加については、41 ページの「ディレクトリエントリの作成」を参照してください。

   b. データベースリンクによって連鎖されたサブツリーで、手順 1-a で作成した管理ユーザにプロキシ権限を指定します。

      ACI の構成については、187 ページの「アクセス制御の管理」を参照してください。

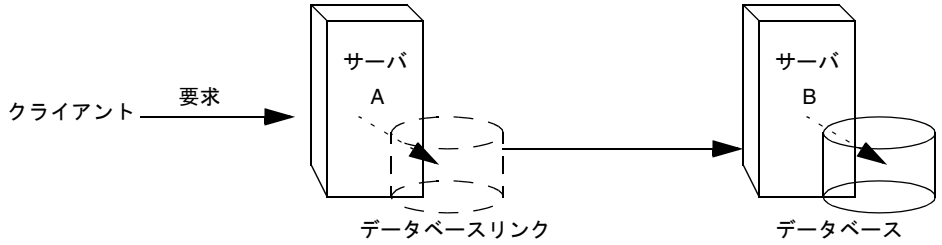
2. データベースリンクが含まれるサーバで、次の手順を実行します。

   a. `ldapmodify` を使用して、`cn=`*database_link_name*`,cn=chaining database,cn=plugins,cn=config` エントリの `nsMultiplexorBindDN` 属性に、データベースリンクのユーザ DN を指定します。

警告 `nsMultiplexorBindDN` に、ディレクトリマネージャの DN を指定することはできません。

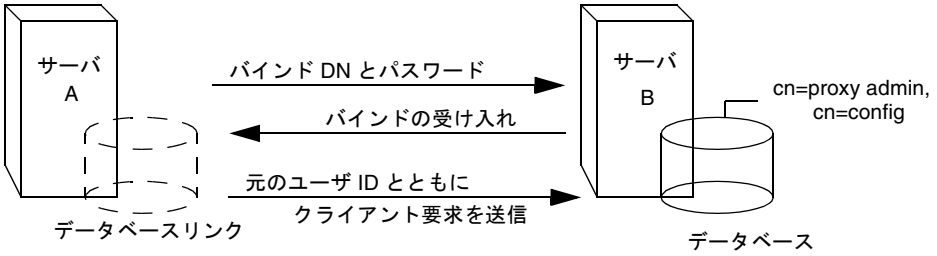
   b. `ldapmodify` を使用して、`cn=`*database_link_name*`,cn=chaining database,cn=plugins,cn=config` エントリの `nsMultiplexorCredentials` 属性に、データベースリンクのユーザパスワードを指定します。

たとえば、クライアントアプリケーションがサーバ A に要求を送信するとします。サーバ A には、サーバ B にあるデータベースに対して要求を連鎖するデータベースリンクが含まれています。



サーバ A のデータベースリンクが、nsMultiplexorBindDN 属性で定義された特別なユーザと、nsMultiplexorCredentials 属性で定義されたユーザパスワードを使用して、サーバ B にバインドします。この例でサーバ A が使用するバインド資格は次のとおりです。

```
nsMultiplexorBindDN: cn=proxy admin,cn=config
nsMultiplexorCredentials: secret
```



サーバ B には nsMultiplexorBindDN に対応するユーザエントリが必要であり、このユーザに対してプロキシ認証権限を設定する必要があります。プロキシ認証権限を設定するには、ほかの ACI と同様、「プロキシ」ACI を設定する必要があります。

警告 連鎖を有効にするときは、ディレクトリの制限領域へのアクセス権を与えないように、アクセス制御を注意深く確認してください。たとえば、分岐にデフォルトのプロキシ ACI を作成すると、データベースリンクを介して接続するユーザは、この分岐の下にあるすべてのエントリを表示できてしまいます。ただし、ユーザがサブツリーをすべて表示することが望ましくない場合もあります。セキュリティホールの発生を回避するには、ACI を追加で作成して、サブツリーへのアクセスを制限する必要があります。

ACI については、187 ページの「アクセス制御の管理」を参照してください。プロキシ認証制御については、http://developer.iplanet.com/docs/manuals/directory.html にある C-SDK に関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からデータベースリンクを使用して、エントリの作成や変更をした場合、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の識別名は反映されません。これらの属性には、リモートデータサーバでのプロキシ認証権限を持つ管理者名が反映されます。

LDAP URL の指定

データベースリンクを含むサーバで、このデータベースリンクが LDAP URL を使用して接続するリモートサーバを識別する必要があります。標準の LDAP URL 形式とは異なり、リモートサーバの URL では、接尾辞は指定されません。この形式は次のようになります。

`ldap://`*`servername:portnumber`*`/`

リモートサーバの URL を指定するには、設定ファイルの `cn=`*`database_link_name`*`,cn=chaining database,cn=plugins,cn=config` エントリの `nsFarmServerURL` 属性を使用します。たとえば、`nsFarmServerURL` は次のようになります。

```
nsFarmServerURL: ldap://siroe.com:389/
```

URL の末尾には必ずスラッシュ (/) を付けてください。

SSL を介して LDAP を使用することによって、データベースリンクをリモートサーバに接続させる場合は、リモートサーバの LDAP URL は次の形式になります。

`ldaps://`*`servername:portnumber`*`/`

連鎖と SSL については、105 ページの「SSL を使用した連鎖」を参照してください。

フェイルオーバサーバのリストの指定

障害の発生に備えて、サーバに対して LDAP URL を追加で指定することができます。このためには、`nsFarmServerURL` 属性に代替サーバを追加します。代替サーバ間はスペースで区切ります。たとえば、次のように入力します。

```
nsFarmServerURL: ldap://siroe.com us.siroe.com:389
 africa.siroe.com:1000/
```

この LDAP URL の例では、データベースリンクは、まず操作をサービスする `siroe.com` 標準ポートにあるサーバに接続します。このサーバが応答しない場合は、ポート `389` にあるサーバ `us.siroe.com` に接続します。これにも失敗した場合は、ポート `1000` にあるサーバ `africa.siroe.com` に接続します。

データベースリンク構成属性のまとめ

次の表に、データベースリンクの構成で使用可能な属性を示します。これらの属性の一部については、これまでの節で説明しました。

アスタリスク (*) が付いている属性は、グローバル属性およびインスタンス属性の両方にすることができます。インスタンス属性はすべて、`cn=`*database_link_name*`,cn=chaining database,cn=plugins,cn=config` エントリで定義されています。

2 つのグローバル構成属性は、`cn=config,cn=chaining database,cn=plugins,cn=config` エントリにあります。グローバル属性は動的です。つまり、グローバル属性を変更すると、ディレクトリにあるデータベースリンクのすべてのインスタンスが、自動的に変更されます。

特定のデータベースリンクについて定義された値は、グローバル属性値よりも優先します。

表 3-4 データベースリンク構成の属性

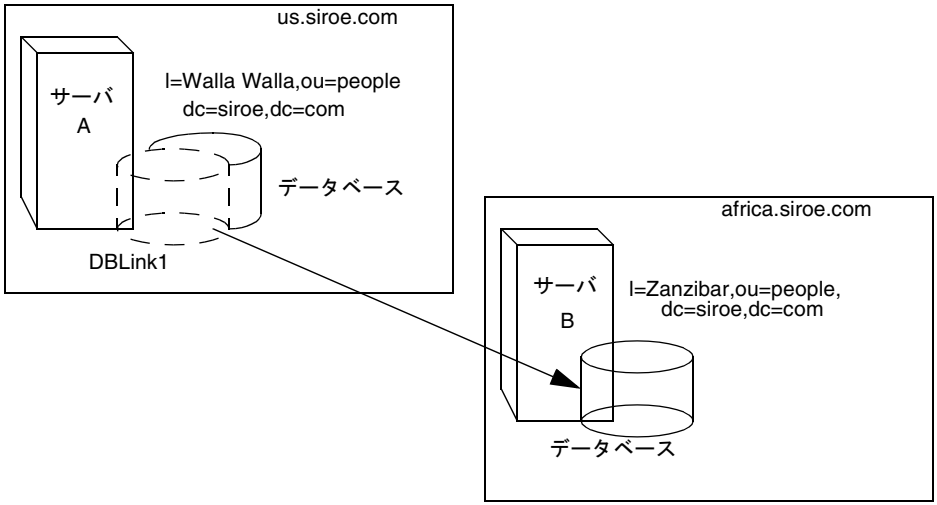
属性	値
`nsTransmittedControls*`	リモートデータサーバに対してデータベースリンクによって転送された LDAP 制御の OID
`nsslapd-suffix`	データベースリンクによって管理される接尾辞。エントリの作成後にこの属性に加えた変更を有効にするには、データベースリンクを含むサーバの再起動が必要
`nsslapd-timelimit`	データベースリンクを検索するときの制限時間のデフォルト値。単位は秒。デフォルトは 3600 秒です。
`nsslapd-sizelimit`	データベースリンクのサイズ制限のデフォルト値。デフォルトは 2000 です。
`nsFarmServerURL`	データを含むリモートサーバ (ファームサーバ) の LDAP URL。この属性には、フェイルオーバ用のオプションサーバを、スペースで区切って指定することができます。カスケード型連鎖を使用している場合は、この URL で別のデータベースリンクをポイントできる。
`nsMultiplexorBindDN`	リモートサーバとの通信に使用される管理エントリの DN。属性名にある *multiplexor* (マルチプレクサ) という単語は、データベースリンクを含み、リモートサーバと通信するサーバを意味する。 このバインド DN にディレクトリマネージャを指定することはできない。この属性が指定されていない場合は、データベースリンクは匿名でバインドする。

表 3-4　データベースリンク構成の属性 (続き)

属性	値
nsMultiplexorCredentials	管理ユーザ用パスワードを、プレーンテキストで指定します。パスワードが指定されていない場合は、ユーザは匿名でバインドできる。パスワードは構成ファイルで暗号化される
nsCheckLocalACI	拡張機能のために予約されています。リモートデータサーバと同様に、データベースリンクでも ACI が評価されるかどうかを制御します。値として、on または off を指定できる。 この属性に対する変更を有効にするには、サーバを再起動する必要がある。デフォルト値は off
nsProxiedAuthorization	拡張機能のために予約されています。プロキシ認証を無効にすることができる。値が off の場合は、プロキシ認証が無効であることを示す。デフォルト値は on
nsActiveChainingComponents*	連鎖を使用するコンポーネントのリスト。コンポーネントとは、サーバ内の機能単位を指す。データベースリンクインスタンスにおけるこの属性値は、グローバル構成属性よりも優先する。特定のデータベースインスタンスで連鎖を無効にするには、値 none を使用する。 デフォルトのポリシーでは、連鎖は禁止。詳細は、89 ページの「コンポーネント操作の連鎖」を参照
nsReferralOnScopedSearch	範囲検索でレフェラルが返されるかどうかを制御する。範囲検索に対してレフェラルを返す方が効率的なため、この属性はディレクトリを最適化するために使用される。値として、on または off を指定できる。デフォルト値は off
nsHopLimit	あるデータベースリンクから別のデータベースリンクに要求を転送できる最大回数。デフォルトは 10

データベースリンク構成例

たとえば、us.siroe.com ドメイン内のサーバにあるデータベースに、サブツリー l=Walla Walla,ou=people,dc=siroe,dc=com があるとします。このとき l=Zanzibar,ou=people,dc=siroe,dc=com に対する操作要求を、africa.siroe.com ドメインにある別のサーバに連鎖させるとします。この操作を図に示すと、次のようになります。



はじめに ldapmodify コマンド行ユーティリティを使用してデータベースリンクをサーバ A に追加します。

```
ldapmodify -a -h us.siroe.com -p port \
          -D "cn=Directory Manager" -w password
```

次に、データベースリンクの構成情報を指定します。

```
dn: cn=DBLink1,cn=chaining database,cn=plugins,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsBackendInstance
nsslapd-suffix: l=Zanzibar,ou=people,dc=siroe,dc=com
nsfarmserverurl: ldap://africa.siroe.com:389/
nsmultiplexorbinddn: cn=proxy admin,cn=config
nsmultiplexorcredentials: secret
cn: DBLink1

dn: cn="l=Zanzibar,ou=people,dc=siroe,dc=com",cn=mapping
 tree,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsMappingTree
nsslapd-state: backend
nsslapd-backend:DBLink1
nsslapd-parent-suffix: "ou=people,dc=siroe,dc=com"
cn: l=Zanzibar,ou=people,dc=siroe,dc=com
```

最初のセクションでは、nsslapd-suffix 属性に、サーバ A からの連鎖先であるサーバ B にある接尾辞が含まれています。nsFarmServerURL 属性には、サーバ B の LDAP URL が含まれています。

2 番目のセクションでは新しい接尾辞が作成されるので、サーバが新しいデータベースリンクに対する要求を送信できるようになります。cn 属性には、データベースリンクの nssalpd-suffix 属性で指定されているのと同じ接尾辞が含まれています。nsslapd-backend 属性には、データベースリンク名が含まれています。nsslapd-parent-suffix 属性は、この新しい接尾辞の親 ou=people,dc=siroe,dc=com を表します。

次のように入力して、サーバ B に管理ユーザを作成します。

```
dn: cn=proxy admin,cn=config
objectclass:person
objectclass: organizationalPerson
objectclass:inetOrgPerson
cn: proxy admin
sn: proxy admin
userPassword: secret
description: Entry for use by database links
```

警告 リモートサーバのプロキシ管理ユーザには、ディレクトリマネージャのユーザを使用しないでください。ディレクトリマネージャのユーザを使用すると、セキュリティホールが発生します。

次のプロキシ認証 ACI を、サーバ B の l=Zanzibar, ou=people,dc=siroe,dc=com エントリに追加します。

```
aci: (targetattr = "*")(version 3.0; acl "Proxied authorization for
 database links"; allow (proxy) userdn = "ldap:///cn=proxy
 admin,cn=config";)
```

この ACI によって、プロキシ管理ユーザに対して、l=Zanzibar,ou=people,dc=siroe,dc=com サブツリー内のみにあるリモートサーバに含まれるデータへの読み取り専用アクセス権が与えられます。

注 ユーザがデータベースリンクにバインドすると、ユーザの ID がリモートサーバに送られます。ここでのアクセス制御は、常にリモートサーバで評価されます。ユーザが正常にリモートサーバのデータを修正したり、リモートサーバにデータを書き込んだりできるようにするには、リモートサーバに正確なアクセス制御を設定する必要があります。

連鎖操作のコンテキストでアクセス制御がどのように評価されるかについては、107 ページの「データベースリンクとアクセス制御の評価」を参照してください。

SSL を使用した連鎖

SSL を使用して、リモートサーバと通信するようにデータベースリンクを構成することができます。SSL を使用した連鎖は、次のステップで行います。

- リモートサーバで、SSL を有効にする

  SSL の有効化については、376 ページの「SSL の有効化 : 手順の概要」を参照してください。

- リモートサーバの LDAP URL を SSL 形式で指定する

  `nsFarmServerURL` 属性に LDAP URL を指定します。この属性については、100 ページの「LDAP URL の指定」を参照してください。

  たとえば、次のように LDAP URL を指定します。

  ```
  nsFarmServerURL: ldaps://africa.siroe.com:636/
  ```

- データベースリンクが含まれるサーバで SSL を有効にする

  SSL の有効化については、376 ページの「SSL の有効化 : 手順の概要」を参照してください。

データベースリンクとリモートサーバに対して SSL を使用して通信するように設定しても、操作を要求しているクライアントアプリケーションは、必ずしも 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、通常のポートを使用してバインドすることもできます。

データベースリンクの管理

ここでは、既存のデータベースリンクの更新方法と削除方法について説明します。次の手順ごとに説明します。

- 「リモートサーバ認証情報の更新」(106 ページ)
- 「データベースリンクの削除」(106 ページ)

リモートサーバ認証情報の更新

リモートサーバに接続するためにデータベースリンクで使用されるバインド DN とパスワードを更新するには、次の手順を実行します。

1. Directory Server Console で、「構成」タブを選択します。
2. 左側の区画で Data を展開し、該当する接尾辞の更新対象のデータベースリンクを特定して選択します。
3. 右側のナビゲーション区画で、「認証」タブをクリックします。
4. リモートサーバ情報を更新するには、「リモートサーバの URL」フィールドに新しい LDAP URL を入力します。

   標準の LDAP URL 形式とは異なり、リモートサーバの URL では、接尾辞は指定されません。この形式は次のようになります。

   `ldap://servername:portnumber/`
5. 「データベースリンクのバインド DN」フィールドに新しい DN を入力して、リモートサーバにバインドするのにデータベースリンクで使用されるバインド DN を更新します。
6. 「データベースリンクのパスワード」フィールドに新しいパスワードを入力して、リモートサーバにバインドするのにデータベースリンクで使用されるパスワードを更新します。確認のため、「データベースリンクパスワードの確認」フィールドに同じパスワードを入力します。

   「リモートサーバのチェックリスト」ボックスに、データベースリンクが正常にバインドするためにリモートサーバで必要とされる管理ユーザエントリ、接尾辞、および ACI が一覧表示されます。
7. 「保存」をクリックして、変更内容を保存します。

データベースリンクの削除

データベースリンクを削除するには、次の手順を実行します。

1. Directory Server Console で、「構成」タブを選択します。
2. 左側のナビゲーション区画で、削除するデータベースリンクを選択します。
3. 「オブジェクト」メニューの「削除」を選択します。

   また、このデータベースリンク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」を選択することもできます。

   「データベースリンクの削除」の確認ダイアログボックスが表示されます。

4. 「はい」をクリックして、データベースリンクの削除を確認します。

   削除中に、Directory Server がどのような処理を実行しているかが、ダイアログボックスに表示されます。

   一度削除されたデータベースリンクは、右側の区画には表示されなくなります。

データベースリンクとアクセス制御の評価

ユーザがデータベースリンクを含むサーバにバインドすると、データベースリンクによって、ユーザの ID がリモートサーバに送られます。ここでのアクセス制御は、常にリモートサーバで評価されます。リモートサーバで評価される LDAP 操作では、プロキシ認証制御を介して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。ユーザが、リモートサーバ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、リモートサーバで操作が成功します。つまり、リモートサーバには、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、ユーザエントリへのアクセス権が必要です。データベースリンクを介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、ユーザエントリがユーザのデータと同じデータベー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の一部が動作しないことがある。

  リモートサーバからは、クライアントアプリケーションはデータベースリンクと同じ IP アドレスにあり、同じ DNS ドメインに存在するものとみなされます。

データベースリンク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場所にある必要がある。これらのグループが動的である場合は、グループのすべてのユーザが、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。静的グループの場合は、リモートユーザが参照される場合がある。

- ACI は、使用する ロール (role) 定義、およびこれらのロールを持つ予定のユーザと同じ場所に存在する必要がある。

- ユーザのエントリ値を参照する ACI (たとえば、`userattr` サブジェクト規則) は、ユーザ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で評価されますが、データベースリンクを含むサーバとリモートサーバ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、ユーザエントリのコンテンツが使用できるとは限らない (データベースリンクが含まれるサーバでアクセス制御が評価され、エントリがリモートサーバにある場合など)。

  性能上の理由から、クライアントがリモート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。

- クライアントアプリケーションによって修正されているエントリへのアクセス権がデータベースリンクにあるとは限らない。

  修正操作を実行するときに、リモートサーバに格納されているすべてのエントリへのアクセス権が、データベースリンクにあるとは限りません。削除操作を実行する場合は、データベースリンクが認識しているのはエントリの DN だけです。アクセス制御で特定の属性が指定されている場合は、データベースリンクを介して削除操作を実行すると失敗します。

注 デフォルトでは、データベースリンクを含むサーバ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=`*`database_link_name`*`,cn=chaining database,cn=plugins,cn=config` エントリの `nsCheckLocalACI` 属性を使用します。ただし、データベースリンクを含むサーバ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。

拡張機能： データベースリンクの性能の調整

ここでは、接続とスレッド管理によって、データベースリンクの性能を調節する方法について、次の項目ごとに説明します。

- 「リモートサーバへの接続の管理」(108 ページ)
- 「標準処理時のエラーの検出」(111 ページ)
- 「スレッド操作の管理」(112 ページ)

リモートサーバへの接続の管理

各データベースリンクは、リモートサーバへの接続のプールを維持します。使用しているディレクトリで資源が最適化されるように、接続を構成することができます。

Directory Server Console やコマンド行を使用して、接続属性を変更することができます。

Console を使用したリモートサーバへの接続の管理

1. Directory Server Console で、「構成」タブを選択します。
2. 左側の区画にある Data フォルダを展開し、変更の対象となるデータベースリンクを特定します。このデータベースリンクをクリックして、右側のナビゲーション区画で「制限と制御」タブをクリックします。
3. 「接続管理」セクションで、次のフィールドを変更します。

   最大 TCP 接続数 : データベースリンクがリモートサーバとの間で確立する TCP 接続の最大数。デフォルトは 3 です。

   バインドタイムアウト : データベースリンクのバインド処理の試行がタイムアウトになるまでの時間 (秒)。デフォルトは 15 秒です。

   接続ごとの最大バインド数 :TCP 接続ごとの未処理のバインド処理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。

   中断までのタイムアウト時間 (秒) : サーバがタイムアウトされた接続を放棄するかどうかを確認するまでの秒数。デフォルトは 2 秒です。

   最大 LDAP 接続数 : データベースリンクがリモートサーバとの間で確立する LDAP 接続の最大数。デフォルトは 10 です。

   最大バインド再試行数 : データベースリンクがリモートサーバとのバインドを試行する回数。「0」を指定すると、データベースリンクによって 1 回だけバインドが試行される。デフォルトは 3 です。

   接続ごとの最大操作数 :LDAP 接続ごとの未処理操作の最大数。デフォルト値は 1 接続あたり 10 です。

   接続継続時間 (秒) : データベースリンクとリモートサーバの間の、継続した接続時間の制限。データベースリンクとリモートサーバの間の接続を無制限に開いたままにしておくことも、あるいは特定の時間が経過したら接続を閉じることもできます。

   接続したままにすると、処理は速くなりますが、資源が多く使用されます。特に、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る。

   「0」は、時間制限を設けないことを示す。デフォルトは 0 です。
4. 「保存」をクリックして、変更内容を保存します。

コマンド行からのリモートサーバへの接続の管理

`ldapmodify` を使用して、データベースリンクエントリに接続属性を追加します。

デフォルトの接続管理属性は、エントリ `cn=default instance config, cn=chaining database,cn=plugins,cn=config` に格納されます。

特定のデータベースリンクで使用される接続管理属性は、エントリ cn=*database_link_name*,cn=chaining database,cn=plugins,cn=config に格納されます。ここで、*database_link_name* はデータベースリンク名です。このエントリで指定された接続管理属性は、cn=default instance config エントリで指定された属性よりも優先します。

次の表に、接続管理に関連する属性を示します。

表 3-5 データベースリンク接続管理属性

属性名	内容
nsOperationConnectionsLimit	データベースリンクがリモートサーバとの間で確立する LDAP 接続の最大数。デフォルト値は、データベースリンクインスタンスごとに 10
nsBindConnectionsLimit	データベースリンクがリモートサーバとの間で確立する TCP 接続の最大数。デフォルトは 3
nsConcurrentOperationsLimit	LDAP 接続ごとの未処理操作の最大数。デフォルト値は 1 接続あたり 10
nsConcurrentBindLimit	TCP 接続ごとの未処理のバインド処理の最大数を指定する。デフォルトは 10
nsBindRetryLimit	データベースリンクがリモートサーバとのバインドを試行する回数。「0」を指定すると、データベースリンクによって 1 回だけバインドが試行される。デフォルトは 3
nsConnectionLife	接続継続時間 (秒)。データベースリンクとリモートサーバの間を無制限に接続しておくことも、あるいは特定の時間が経過したら接続を閉じることもできる 接続したままにすると、処理は速くなるが、資源が多く使用される。特に、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る 「0」は、時間制限を設けないことを示す。デフォルトは 0。この値が 0 で nsFarmServerURL 属性にフェイルオーバサーバのリストが指定されている場合は、フェイルオーバにより代替サーバへ切り替わったあとで、「メイン」サーバへの接続が行われなくなる デフォルトは 0 秒
nsBindTimeout	バインド処理の試行がタイムアウトになるまでの時間 (秒) を指定する。デフォルトは 15 秒
nsAbandonedSearchCheckInterval	サーバが異常終了した操作を確認するまでの秒数。デフォルトは 2 秒

データベースリンク構成属性のリストについては、101 ページの「データベースリンク構成の属性」を参照してください。

標準処理時のエラーの検出

データベースリンクとリモートサーバの間で、標準の連鎖処理が行われている間にエラーを検出できると、サーバの性能の保護が容易になります。データベースリンクには 2 つの属性があり、それらを組み合わせて、リモートサーバが応答しなくなったかどうかが判断されます。

1 つめの属性は nsMaxResponseDelay で、LDAP 操作が完了するまでの最高持続時間を設定します。操作時間がこの属性に指定された時間数を超えると、データベースリンクのサーバでは、リモートサーバがオンラインの状態ではないと認識します。

nsMaxResponseDelay で設定された時間が経過すると、データベースリンクは、リモートサーバに対して ping を実行します。この ping の間、データベースリンクは別の LDAP 要求 (このリモートサーバに存在しないオブジェクトの検索要求) を発行します。ping の持続期間を設定するには、nsMaxTestResponseDelay を使用します。

nsMaxResponseDelay 期間が終了する前に、リモートサーバが応答しなくなった場合は、エラーが返され、接続が切断されたことを示すフラグが立ちます。このとき、サーバの性能が低下しないように、データベースリンクとリモートサーバの間のすべての接続は、30 秒間ブロックされます。30 秒が経過すると、データベースリンクからリモートサーバに対する操作要求は、通常どおりに続けられます。

どちらの属性も cn=config,cn=chaining database,cn=plugins,cn=config エントリに格納されます。次の表に、これらの属性を詳しく説明します。

表 3-6 データベースリンク処理エラー検出パラメタ

属性名	内容
nsMaxResponseDelay	データベースリンクからの LDAP 操作要求に対して、リモートサーバからの応答を待ちつづける期間の最大値 (秒)。この期間が経過すると、エラーの可能性があるとみなされる。デフォルトの遅延期間は 60 秒。 ここで設定された遅延期間が経過すると、データベースリンクは、リモートサーバへの接続をテストする
nsMaxTestResponseDelay	データベースリンクによって発行されるテストの持続期間。このテストでは、リモートサーバが応答するかどうかが確認される。この期間が経過していなくても、リモートサーバから応答が帰ってこなくなった場合は、データベースリンクはリモートサーバが停止しているとみなし、それ以降の操作ではこの接続を使用しない この期間は秒単位で指定する。デフォルトのテスト応答遅延期間は 15 秒。

スレッド操作の管理

一般に、Directory Server では、各操作の処理のために、限られた数のスレッドを使い最高の効率で働きます。スレッドの数を制限していることによって、一般的な操作を、高速に処理することができるため、キューの中で空きスレッドを待つ時間が長くなりすぎるのを防止できます。

ただし、データベースリンクはリモートサーバが処理できるよう、操作を転送します。データベースリンクはリモートサーバに接続し、操作を転送して、結果を待ってから、この結果をクライアントアプリケーションに戻します。この操作全体では、ローカル操作と比較してかなり長く時間がかかることがあります。

リモートサーバからの結果を待っている間、データベースリンクは追加の操作を処理することができます。デフォルトでは、サーバが使用できるスレッド数は 20 です。ただし、データベースリンクを使用する場合は、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、性能を向上させることができます。リモートサーバからの応答を待っている間、ローカル CPU をアイドル状態で待機させるのではなく、ほかの操作を処理させることができます。

操作の処理で使用されるスレッド数を変更するには、`cn=config` エントリの `nsslapd-threadnumber` グローバル構成属性を変更します。デフォルトのスレッド数は 20 です。たとえば、性能向上のために、スレッド数を 50 に増やすことができます。スレッド数を変更したら、サーバを再起動して、変更を反映させます。

拡張機能： カスケード型連鎖の構成

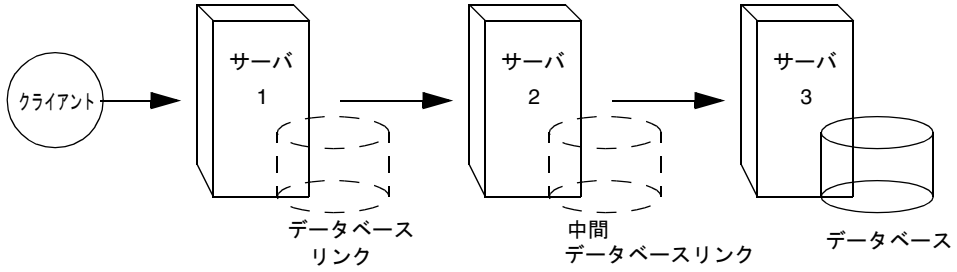
あるデータベースリンクが別のデータベースリンクをポイントするよう構成して、カスケード型連鎖の操作を作成できます。ディレクトリツリーの全データにアクセスするために、複数のホップが必要な場合、カスケード型連鎖が発生します。

ここでは、次の項目について説明します。

カスケード型連鎖の概要

カスケード型連鎖は、クライアントアプリケーションからの要求を処理するために、ディレクトリで複数のホップが必要な場合に発生します。

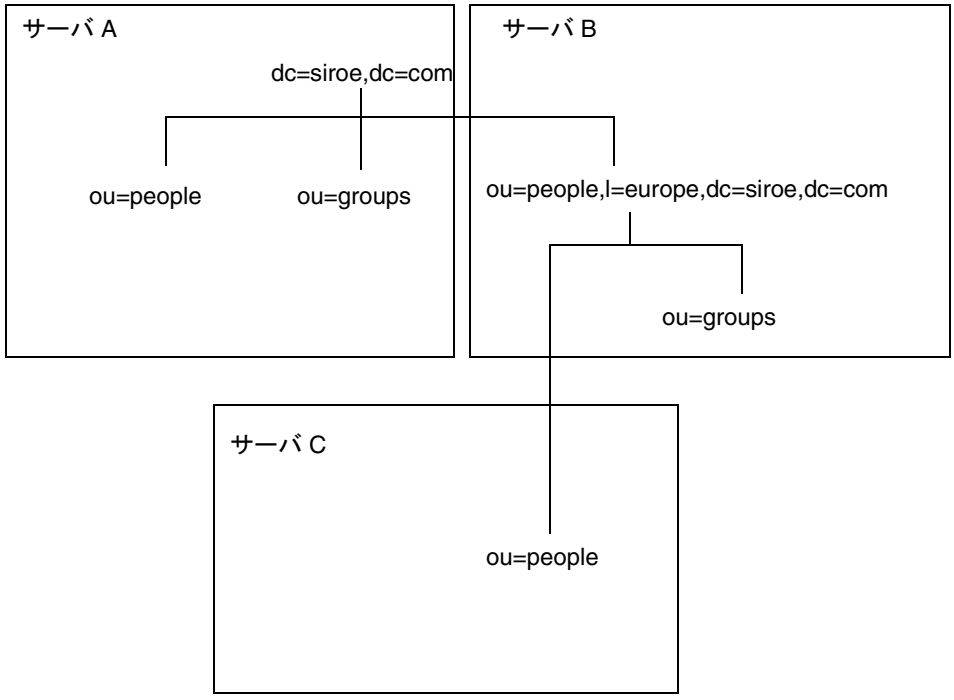
たとえば、次の例について考えてみます。



クライアントアプリケーションはサーバ 1 に修正要求を送信します。サーバ 1 には、サーバ 2 に操作を転送するためのデータベースリンクが含まれています。さらに、サーバ 2 には別のデータベースリンクがあります。サーバ 2 のデータベースリンクは操作をサーバ 3 に転送します。サーバ 3 にあるデータベースには、クライアントによる修正の対象となるデータが格納されています。このクライアントが修正するデータにアクセスするには、2 回のホップが必要です。

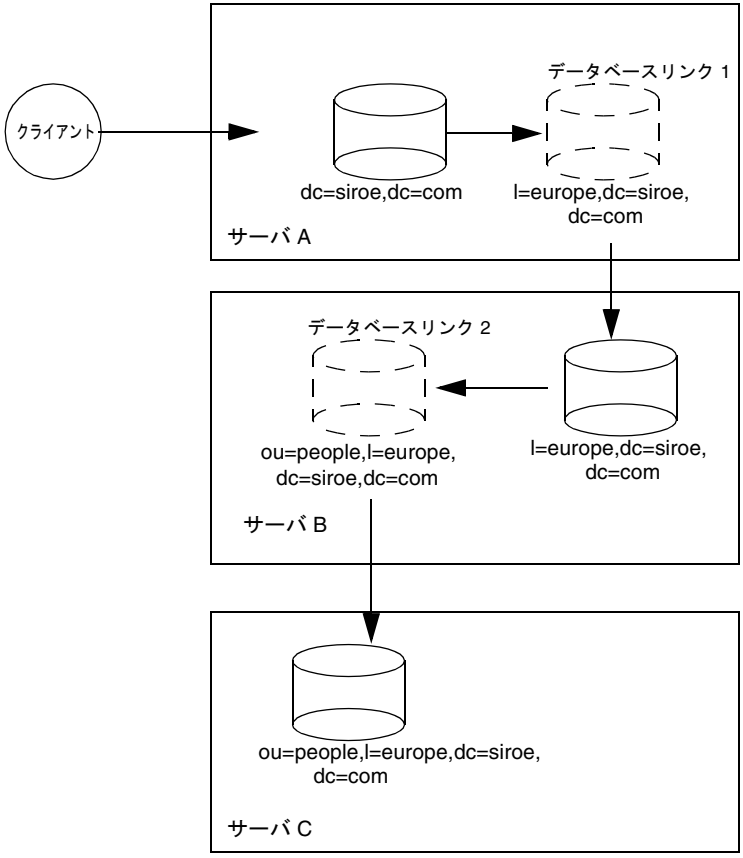
標準の操作要求時に、クライアントはサーバにバインドし、このクライアントに適用されている ACI がすべて評価されます。カスケード型連鎖を使用すると、クライアントのバインド要求はサーバ 1 で評価されますが、クライアントに適用されている ACI は、要求が目的のサーバ (上記の例ではサーバ 2) に連鎖されたときに初めて評価されます。

次のような例を想定します。サーバ A では、ディレクトリツリーは次のように分割されています。



ルート接尾辞 dc=siroe,dc=com、サブ接尾辞 ou=people と ou=groups はサーバ A に格納されています。l=europe,dc=siroe,dc=com および ou=groups 接尾辞は、サーバ B に格納され、l=europe,dc=siroe,dc=com 接尾辞の ou=people 分岐は、サーバ C に格納されています。

ここで、サーバ A、B、C に設定されたカスケード型連鎖を使用すると、ou=people,l=europe,dc=siroe,dc=com エントリをターゲットとするクライアント要求は、ディレクトリによって次のようにルーティングされます。



まず、クライアントがサーバ A にバインドし、データベースリンク 1 を使用してサーバ B に連鎖します。次に、サーバ B はデータベースリンク 2 を使用して `ou=people,l=europe,dc=siroe,dc=com` 分岐のデータにアクセスするために、サーバ C 上のターゲットデータベースに連鎖します。ディレクトリがクライアント要求を実行するには、少なくとも 2 回のホップが必要なので、この場合もカスケード型連鎖とみなされます。

Console を使用したカスケード型連鎖のデフォルト構成

Directory Server で、すべてのデータベースリンクを対象とするカスケード型連鎖のデフォルト値を作成するには、次の手順を実行します。

1. Directory Server Console で、「構成」タブを選択します。

2. 左側の区画にある Data フォルダを展開し、「データベースリンクの設定」をクリックします。「デフォルト作成パラメタ」タブをクリックします。
3. カスケード型連鎖に関連する中間データベースリンクにあるローカル ACI の評価を有効にするには、「ローカル ACI をチェックする」チェックボックスを選択します。このチェックボックスを選択する場合は、中間データベースリンクが含まれるサーバのデータベースに適切なローカル ACI を追加する必要があります。

   これは拡張機能です。詳細は、119 ページの「ローカル ACI の評価の有効化」を参照してください。
4. このデータベースリンクが別のデータベースリンクをポイントできる回数の最大値を、「最大ホップ」フィールドに入力します。

   デフォルトの最大値は 10 ホップです。実行したホップの数が 10 を超えると、サーバによってループが検出され、クライアントアプリケーションにエラーが返されます。
5. 「保存」をクリックして、変更内容を保存します。

注 データベースリンクのデフォルト設定に対する変更は、過去にさかのぼって適用されることはありません。デフォルト設定への変更を保存したあとに作成されたデータベースリンクだけで、変更が反映されます。

Console を使用したカスケード型連鎖の構成

特定のデータベースリンクに対するカスケード型連鎖を設定するには、次の手順を実行します。

1. Directory Server Console で、「構成」タブを選択します。
2. 左側の区画にある Data フォルダを展開し、カスケード型連鎖に含めるデータベースリンクを特定します。このデータベースリンクをクリックして、右側のナビゲーション区画で「制限と制御」タブをクリックします。
3. カスケード型連鎖に関連する中間データベースリンクにあるローカル ACI の評価を有効にするには、「ローカル ACI をチェックする」チェックボックスを選択します。このチェックボックスを選択する場合には、データベースリンクに適切なローカル ACI を追加しておく必要があります。

   これは拡張機能です。詳細は、119 ページの「ローカル ACI の評価の有効化」を参照してください。
4. このデータベースリンクが別のデータベースリンクをポイントできる回数の最大値を、「最大ホップ」フィールドに入力します。

   デフォルトの最大値は 10 ホップです。実行したホップの数が 10 を超えると、サーバによってループが検出され、クライアントアプリケーションにエラーが返されます。

5. 「保存」をクリックして、変更内容を保存します。

コマンド行からのカスケード型連鎖の構成

コマンド行を使用したデータベース連鎖の構成は、次のステップで行います。

- データベースリンクのうち、中間データベースリンクを含むサーバの URL へのものを、1 つポイントする
- プロキシ認証制御を送信するように、中間データベースリンク (前述の例では、サーバ 2) を構成する
- すべての中間データベースリンクに、プロキシ管理ユーザ ACI を作成する。このためには、中間データベースリンクが含まれるサーバごとに、データベースを作成する必要がある
- すべての中間データベースリンクで、ローカル ACI の評価を有効にする
- すべての中間データベースリンクと、最終連鎖先となるデータベースに、クライアント ACI を作成する

ここでは、次の項目について説明します。

- 「ほかのデータベースリンクへのポイント」(117 ページ)
- 「プロキシ認証制御の送信」(118 ページ)
- 「プロキシ管理ユーザ ACI の作成」(118 ページ)
- 「ローカル ACI の評価の有効化」(119 ページ)
- 「クライアント ACI の作成」(119 ページ)
- 「ループの検出」(120 ページ)

ほかのデータベースリンクへのポイント

カスケード型連鎖を作成するには、あるデータベースリンクの `nsFarmServerURL` 属性に、別のデータベースリンクを含むサーバの URL を含める必要があります。たとえば、サーバ `siroe1.com` にあるデータベースリンクに、`africa.siroe.com` というサーバにあるデータベースリンクへのポイントを設定するとします。サーバ 1 にあるデータベースリンクの `cn=`*`database_link_name`*`,cn=chaining database, cn=plugins,cn=config` エントリの内容は、次のようになります。

```
nsFarmServerURL: ldap://africa.siroe.com:389
```

プロキシ認証制御の送信

デフォルトでは、データベースリンクはプロキシ認証制御を送信しません。ただし、データベースリンクが別のデータベースリンクに接続するときには、この制御を使用して、最終連鎖先サーバで必要となる情報を送信することができます。中間データベースリンクもまた、この制御を送信する必要があります。プロキシ認証制御が送信されるようにデータベースリンクを構成するには、中間データベースリンクの `cn=config,cn=chaining database,cn=plugins,cn=config` エントリに次の行を追加します。

```
nsTransmittedControls: 2.16.840.1.113730.3.4.12
```

OID 値はプロキシ認証制御を表します。LDAP 制御の連鎖については、93 ページの「LDAP 制御の連鎖」を参照してください。

プロキシ管理ユーザ ACI の作成

中間データベースリンクを含むサーバでは、次のサーバにリンクが転送される前に、最初のデータベースリンクの権限を確認するための ACI を作成する必要があります。これは、サーバ 2 でサーバ 1 の資格が確認されない場合、誰でも匿名でバインドすれば、プロキシ認証制御を通過することができるので、必要以上の管理権限が付与されてしまう可能性があるためです。

このようなセキュリティホールの発生を防ぐには、中間データベースリンクを含むサーバ上に ACI を作成しておく必要があります。ACI を作成するには、次の手順を実行します。

1. 中間データベースリンクを含むサーバにデータベースが存在しない場合は、これを作成します。このデータベースには、管理ユーザエントリと ACI が含まれます。データベースの作成については、81 ページの「データベースの作成」を参照してください。
2. データベースの管理ユーザに対応するエントリを作成します。
3. 適切な接尾辞をターゲットにする管理ユーザ用の ACI を作成します。これによって、管理者はこのデータベースリンクの接尾辞だけにアクセスできるようになります。管理ユーザエントリに、次の ACI を追加します。

   ```
   aci: (targetattr = "*")(version 3.0; acl "Proxied authorization
    for database links"; allow (proxy) userdn = "ldap:///cn=proxy
    admin,cn=config";)
   ```

   この ACI は、単純な連鎖を構成するときにリモートサーバで作成する ACI に似ています。

警告 連鎖を有効にするときは、ディレクトリの制限領域へのアクセス権を与えないように、アクセス制御を注意深く確認してください。たとえば、分岐にデフォルトのプロキシ ACI を作成すると、データベースリンクを介して接続するユーザは、この分岐の下にあるすべてのエントリを表示できてしまいます。ただし、ユーザがサブツリーをすべて表示することが望ましくない場合もあります。セキュリティホールの発生を回避するには、ACI を追加で作成して、サブツリーへのアクセスを制限する必要があります。

ローカル ACI の評価の有効化

プロキシ管理 ACI を常に確認するには、連鎖に関連するすべての中間データベースリンクで、ローカル ACI の評価を有効にする必要があります。このためには、すべての中間データベースリンクの `cn=`*`database_link_name`*`,cn=chaining database,cn=plugins,cn=config` エントリに、次の属性を追加します。

```
nsCheckLocalACI: on
```

`cn=default instance config,cn=chaining database,cn=plugins,cn=config` エントリでこの属性を on にすると、すべてのデータベースリンクインスタンスでその `cn=`*`database_link_name`*`,cn=chaining database,cn=plugins,cn=config` エントリの `nsCheckLocalACI` 属性が on に設定されます。

クライアント ACI の作成

ローカル ACI の評価を有効にしたので、中間データベースリンクと最終連鎖先データベースの両方に、適切なクライアントアプリケーション ACI を作成する必要があります。

中間データベースリンクに ACI を作成するには、まず最終連鎖先接尾辞のルート接尾辞を表す接尾辞を含むデータベースを作成します。

たとえば、リモートサーバの `c=africa,ou=people,dc=siroe,dc=com` 接尾辞に対するクライアント要求を連鎖させる場合は、関連するすべての中間データベースリンクに `dc=siroe,dc=com` 接尾辞に関連するデータベースを含める必要があります。

次に、この上位接尾辞エントリにクライアント ACI を追加する必要があります。たとえば、次のように追加します。

```
aci: (targetattr = "*")(version 3.0; acl "Client authentication for
 database link users"; allow (all) userdn = "ldap:///uid=*
 ,cn=config";)
```

この ACI によって、サーバ 1 の `cn=config` エントリに `uid` を持つクライアントアプリケーションが、サーバ 3 の `ou=people,dc=siroe,dc=com` 接尾辞の下にあるデータに対して、任意のタイプの操作を実行できるようになります。

ループの検出

Directory Server に含まれている LDAP 制御でループを防ぐことができます。最初に連鎖を試みたときに、この制御に使用可能な最大ホップ (連鎖接続) 数がサーバによって設定されます。これ以降、別のサーバへ連鎖されるたびに、この数が 1 つずつ減らされます。カウントが 0 になると、サーバはループが検出されたと判断し、クライアントアプリケーションに通知します。

使用可能なホップの数は、`nsHopLimit` 属性を使用して指定されます。指定されていない場合は、デフォルト値の 10 になります。

この制御を使用するには、`cn=config,cn=chaining database,cn=plugins,cn=config` エントリの `nsTransmittedControl` 属性に次の OID を追加します。

```
nsTransmittedControl: 1.3.6.1.4.1.1466.29539.12
```

各データベースリンクの構成ファイルにこの制御が存在していなければ、ループ検出は実装されません。

カスケード型連鎖構成属性のまとめ

次の表に、カスケード型連鎖における中間データベースリンク構成に使用される属性を示します。

表 3-7 カスケード型連鎖構成属性

属性	内容
nsFarmServerURL	カスケード型連鎖で、次のデータベースリンクを含むサーバの URL

nsTransmittedControls	カスケード型連鎖に関連するデータベースリンクに次の OID を入力する
	`nsTransmittedControls: 2.16.840.1.113730.3.4.12`
	`nsTransmittedControls: 1.3.6.1.4.1.1466.29539.12`
	最初の OID は、プロキシ認証制御に対応する。2 番目の OID は、ループ検出制御に対応する

aci	この属性には、次の ACI を指定する必要がある `aci: (targetattr = "*")(version 3.0; acl "Proxied authorization for database links"; allow (proxy) userdn = "ldap:///cn=proxy admin,cn=config";)`

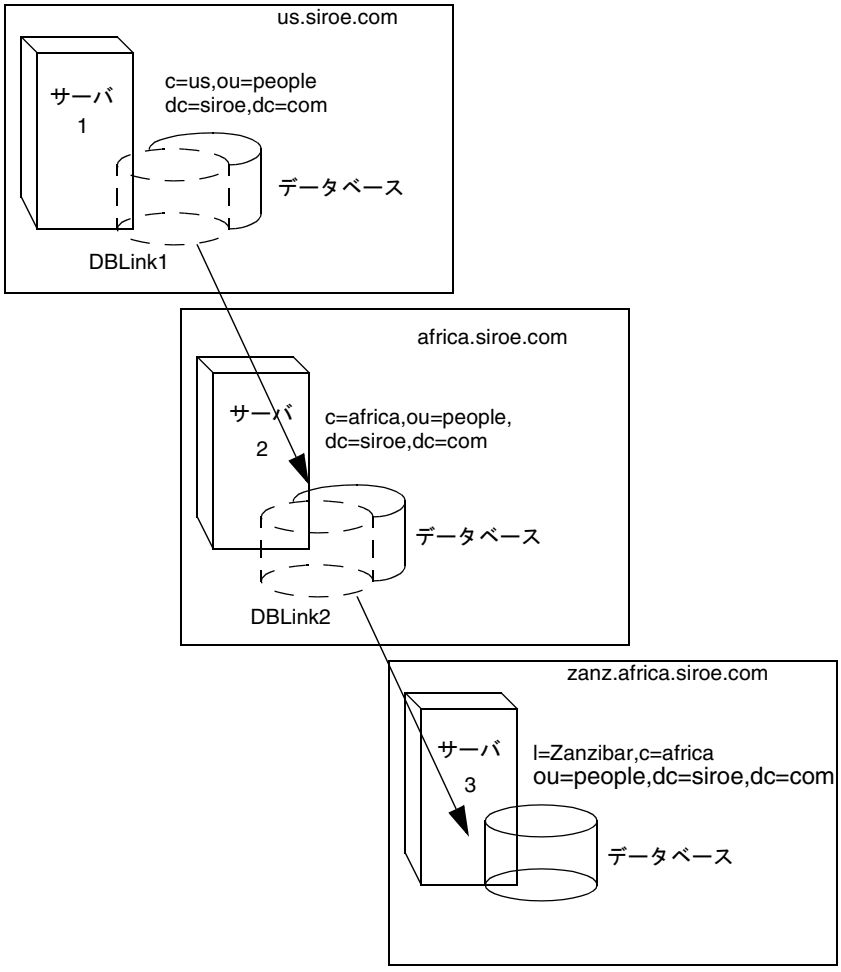
表 3-7　カスケード型連鎖構成属性 (続き)

属性	内容
nsCheckLocalACI	連鎖に関連するすべてのデータベースリンクで、ローカル ACI の評価を有効にするには、次のようにローカル ACI の評価を on にする `nsCheckLocalACI: on`

カスケード型連鎖構成の例

次の図のように、3 つのサーバが関連するカスケード型連鎖を作成するには、3 つのサーバすべてに連鎖コンポーネントを構成する必要があります。ここでは、3 つのサーバが関連するカスケード型連鎖の作成手順について、次の項目にわけて説明します。

- 「サーバ 1 の構成」(122 ページ)
- 「サーバ 2 の構成」(124 ページ)
- 「サーバ 3 の構成」(126 ページ)



サーバ 1 の構成

まず、`ldapmodify` コマンド行ユーティリティを使用して、データベースリンクをサーバ 1 に追加します。次のユーティリティを含むディレクトリに移動します。

Solaris 9 プラットフォーム　`% cd /usr/iplanet/ds5/shared/bin`
その他のプラットフォーム　`% cd` *installDir*`/shared/bin`

次のコマンドを入力して、ユーティリティを実行します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
```

次に、サーバ 1 で、データベースリンク DBLink1 の構成情報を指定します。

```
dn: cn=DBLink1,cn=chaining database,cn=plugins,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsBackendInstance
nsslapd-suffix: l=Zanzibar,c=africa,ou=people,dc=siroe,dc=com
nsfarmserverurl: ldap://africa.siroe.com:389/
nsmultiplexorbinddn: cn=server1 proxy admin,cn=config
nsmultiplexorcredentials: secret
cn:DBLink1
nsCheckLocalACI:off

cn="l=Zanzibar,c=africa,ou=people,dc=siroe,dc=com",cn=mapping
tree,cn: config
objectclass: nsMappingTree
nsslapd-state: backend
nsslapd-backend: DBLink1
nsslapd-suffix: l=Zanzibar,c=africa,ou=people,dc=siroe,dc=com
cn: l=Zanzibar,c=africa,ou=people,dc=siroe,dc=com
```

最初のセクションでは DBLink1 に関連するエントリが作成されます。2 番目のセクションでは新しい接尾辞が作成されるので、サーバでは、データベースリンクに対する要求を正しいサーバに送信できるようになります。ローカル ACI をチェックするために `nsCheckLocalACI` 属性を構成する必要はありません。これはサーバ 2 の DBLink2 データベースリンクでのみ必要です。

ループ検出を実装する必要があるので、サーバ 1 の `cn=config,cn=chaining database,cn=plugins,cn=config` エントリに格納された `nsTransmittedControl` 属性にループ検出制御の OID を指定する必要があります。OID は次のように指定します。

```
dn:cn=config,cn=chaining database,cn=plugins, cn=config
changeType: modify
add: nsTransmittedControl
nsTransmittedControl: 1.3.6.1.4.1.1466.29539.12
```

`nsTransmittedControl` 属性は、通常デフォルトで、ループ検出制御 OID `1.3.6.1.4.1.1466.29539.12` 値によって構成されるため、この値が存在するかどうか前もって確認することをお勧めします。この値が存在しない場合には、この構成手順を実行する必要はありません。

サーバ 2 の構成

次に、サーバ 2 にプロキシ管理ユーザを作成します。この管理ユーザは、サーバ 1 によるサーバ 2 へのバインドと認証のために使用されます。サーバ 1 に一意のプロキシ管理ユーザ名を選択すると便利です。このプロキシ管理ユーザは、サーバ *1* にサーバ 2 へのバインドを許可することができます。次のように入力して、プロキシ管理ユーザを作成します。

```
dn: cn=server1 proxy admin,cn=config
objectclass:person
objectclass: organizationalPerson
objectclass:inetOrgPerson
cn: server1 proxy admin
sn: server1 proxy admin
userPassword: secret
description: Entry for use by database links
```

警告 リモートサーバのプロキシ管理ユーザには、ディレクトリマネージャのユーザを使用しないでください。ディレクトリマネージャのユーザを使用すると、セキュリティホールが発生します。

次に、サーバ 2 でデータベースリンク DBLink2 を構成します。`ldapmodify` を使用して、DBLink2 の構成情報を次のように指定します。

```
dn: cn=DBLink2,cn=chaining database,cn=plugins,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsBackendInstance
nsslapd-suffix: l=Zanzibar,c=africa,ou=people,dc=siroe,dc=com
nsfarmserverurl: ldap://zanz.africa.siroe.com:389/
nsmultiplexorbinddn: cn=server2 proxy admin,cn=config
nsmultiplexorcredentials: secret
cn:DBLink2
nsCheckLocalACI:on

dn:cn="l=Zanzibar,c=africa,ou=people,dc=siroe,dc=com",cn=mapping
tree,cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsMappingTree
nsslapd-state: backend
nsslapd-backend:DBLink2
nsslapd-parent-suffix:"c=africa,ou=people,dc=siroe,dc=com"
cn: l=Zanzibar,c=africa,ou=people,dc=siroe,dc=com
```

データベースリンク DBLink2 は、カスケード型連鎖構成の中間データベースリンクであるため、サーバがクライアントおよびプロキシ管理ユーザがデータベースリンクにアクセスを許可するかどうかをチェックできるように、nsCheckLocalACI を on に設定する必要があります。

サーバ 2 のデータベースリンクは、プロキシ認証制御およびループ検出制御を送信できるように構成されている必要があります。プロキシ認証制御およびループ検出制御を実装するには、両方の対応する OID を指定する必要があります。

cn=config,cn=chaining database, cn=plugins,cn=config エントリに次の情報を追加します。

```
dn:cn=config,cn=chaining database,cn=plugins, cn=config
changeType: modify
add: nsTransmittedControl
nsTransmittedControl: 2.16.840.1.113730.3.4.12
nsTransmittedControl: 1.3.6.1.4.1.1466.29539.12
```

ここで、nsTransmittedControl: 2.16.840.1.113730.3.4.12 はプロキシ認証制御の OID で、nsTransmittedControl: 1.3.6.1.4.1.1466.29539.12 はループ検出制御の OID です。

ループ検出制御が構成済みかどうかを確認し、それに応じて上のコマンドを適用してください。

次の手順では、ACI を設定します。次の操作ができるように、サーバ 2 で l=Zanzibar,c=africa,ou=people,dc=siroe,dc=com 接尾辞の上位に 1 つの接尾辞が存在することを確認する必要があります。

- データベースリンク接尾辞の追加
- ローカルプロキシ認証 ACI を追加。これを使用すると、サーバ 1 は、サーバ 2 で作成されるプロキシ認証管理ユーザを使って接続できる
- ローカルクライアント ACI を追加。これによりクライアント操作がサーバ 2 で正常に実行され、サーバ 3 に転送できる。DBLink2 データベースリンクの ACI のチェックをオンにしているため、このローカル ACI が必要

両方の ACI が c=africa,ou=people,dc=siroe,dc=com 接尾辞を含むデータベースに置かれます。

注 これらの ACI の作成では、エントリを保持するため、c=africa,ou=people,dc=siroe,dc=com 接尾辞に対応するデータベースがすでに存在していることを前提にしています。このデータベースは、各データベースリンクの nsslapd-suffix 属性に指定された接尾辞よりも上にある接尾辞に関連付ける必要があります。つまり、最終連鎖先サーバの接尾辞は、中間サーバで指定された接尾辞のサブ接尾辞と同じである必要があります。

ローカルプロキシ認証 ACI を c=africa,ou=people,dc=siroe,dc=coml エントリに追加します。

```
aci:(targetattr="*")(target="l=Zanzibar,c=africa,ou=people,
dc=siroe,dc=com")(version 3.0; acl "Proxied authorization for
 database links"; allow (proxy) userdn = "ldap:///cn=server1 proxy
 admin,cn=config";)
```

次にローカルクライアント ACI を追加し、ACI 検査がオンになった状態のサーバ 2 でクライアント操作が正常に実行されるようにします。この ACI は、l=Zanzibar,c=africa,ou=people,dc=siroe,dc=com 分岐へのアクセス用に連鎖先のサーバで作成する ACI と同じです。c=us,ou=people,dc=siroe,dc=com 内のすべてのユーザにサーバツリーの l=Zanzibar,c=africa,ou=people,dc=siroe,dc=com 内のエントリへの更新アクセスを許可することもできます。上の文を指している次の ACI は、サーバ 2 の c=africa,ou=people,dc=siroe,dc=com 接尾辞にそれを許可するのに (実現するのに) 作成する必要がある ACI です。

```
aci:(targetattr="*")(target="l=Zanzibar,c=africa,ou=people,
dc=siroe,dc=com")(version 3.0; acl "Client authorization for
 database links"; allow (all) userdn =
"ldap:///uid=*,c=us,ou=people,dc=siroe,dc=com";)
```

この ACI によって、サーバ 1 の c=us,ou=people,dc=siroe,dc=com に uid を持つクライアントは、サーバ 3 の l=Zanzibar,c=africa,ou=people,dc=siroe,dc=com 接尾辞ツリーに対して、任意のタイプの操作を実行できるようになります。サーバ 2 に接尾辞の異なるユーザが存在する場合、サーバ 3 にさらに別の権限が必要になるため、サーバ 2 に別のクライアント ACI を追加する必要があります。

サーバ 3 の構成

カスケード型連鎖の使用例の構成手順の最後は、サーバ 3 の構成です。最初に、サーバ 2 がプロキシ認証で使用する管理ユーザをサーバ 3 に作成します。

```
dn: cn=server2 proxy admin,cn=config
objectclass:person
objectclass: organizationalPerson
objectclass:inetOrgPerson
cn: server2 proxy admin
sn: server2 proxy admin
userPassword: secret
description: Entry for use by database links
```

次にサーバ 2 と同様に、サーバ 3 に対して、同じローカルプロキシ認証 ACI を追加します。次のプロキシ認証 ACI を l=Zanzibar,ou=people,dc=siroe,dc=com エントリに追加します。

```
aci: (targetattr = "*")(version 3.0; acl "Proxied authorization for
 database links"; allow (proxy) userdn = "ldap:///cn=server2 proxy
 admin,cn=config";)
```

この ACI によって、サーバ 2 のプロキシ管理ユーザに対して、`l=Zanzibar,ou=people,dc=siroe,dc=com` サブツリー内にあるリモートサーバであるサーバ 3 に含まれるデータへの読み取り専用アクセス権が与えられます。

次に、`l=Zanzibar,ou=people,dc=siroe,dc=com` サブツリー、つまりもとのクライアントアプリケーションに対応する場所に、ローカルクライアント ACI を作成する必要があります。次のように、サーバ 2 のクライアント用に作成した ACI と同じ ACI を使用します。

```
aci: (targetattr =
"*")(target="l=Zanzibar,c=africa,ou=people,dc=siroe,dc=com")(versio
n 3.0; acl "Client authentication for  database link users"; allow
(all) userdn = "ldap:///uid=*,c=us,ou=people,dc=siroe,dc=com";)
```

これらの手順すべてを完了すると、カスケード型の連鎖構成が設定されます。このカスケード型構成では、サーバ 1 にバインドし、サーバ 3 の `l=Zanzibar,c=africa,ou=people,dc=siroe,dc=com` 分岐情報を変更できます。セキュリティ要件に応じて、さらに詳細なアクセス制御を提供するかどうかを決定してください。

レフェラルの使い方

レフェラルを使用して、指定された情報を取得するためにどのサーバに接続すべきかを、クライアントアプリケーションに通知することができます。このリダイレクトは、ローカルサーバには存在しないディレクトリエントリをクライアントアプリケーションが要求した場合、または保守のためにデータベースがオフラインになっている場合に発生します。ここでは、レフェラルに関する次の項目について説明します。

- デフォルトレフェラルの設定
- スマートレフェラルの作成
- 接尾辞レフェラルの作成

ディレクトリにおけるレフェラルの使用の概念については、『iPlanet Directory Server 導入ガイド』を参照してください。

デフォルトレフェラルの設定

デフォルトレフェラルは、ディレクトリで管理されている接尾辞のいずれにも含まれない 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。ここでは、Console とコマンド行ユーティリティを使用して、ディレクトリにデフォルトレフェラルを設定する手順を示します。

Console を使用したデフォルトレフェラルの設定

ディレクトリにデフォルトレフェラルを設定するには、次の手順を実行します。

1. Directory Server Console で、「構成」タブを選択します。
2. 左側の区画のナビゲーションツリーで、トップのエントリを選択します。
3. 右側の区画で「設定」タブを選択します。
4. 「転送先」テキストボックスに LDAP URL を入力し、「OK」をクリックします。

   たとえば、次のようにします。

   ```
   ldap://directory.siroe.com:389/dc=siroe,dc=com
   ```

   複数のレフェラル URL を入力するには、次のようにスペースで区切って、それぞれを引用符で囲みます。

   ```
   "ldap://d1.siroe.com:389/dc=siroe,dc=com" "ldap://d2.siroe.com/"
   ```

LDAP URL については、付録 C「LDAP URLs」を参照してください。

コマンド行からのデフォルトレフェラルの設定

`ldapmodify` コマンド行ユーティリティを使用して、ディレクトリ構成ファイルの `cn=config` エントリにデフォルトレフェラルを追加します。

たとえば、Directory Server である siroe.com から、サーバ Zanzibar.com に新しいデフォルトレフェラルを追加するには、`cn=config` エントリに行を追加します。次のように、`ldapmodify` ユーティリティを実行します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
```

`ldapmodify` ユーティリティはサーバにバインドし、構成ファイルのエントリを変更する準備を行います。

次に、Zanzibar.com サーバにデフォルトレフェラルを追加します。

```
dn:cn=config
changetype: modify
replace:nsslapd-referral
nsslapd-referral: ldap://zanzibar.com/
```

ディレクトリの cn=config エントリにデフォルトレフェラルを追加すると、クライアントアプリケーションからの要求に対して、このディレクトリからデフォルトレフェラルが返されるようになります。サーバを再起動する必要はありません。

スマートレフェラルの作成

スマートレ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。スマートレ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、特定のサーバや特定のサーバにある特定のエントリを参照させることができます。

たとえば、クライアントアプリケーションがディレクトリエントリ uid=bjensen,ou=people,dc=siroe,dc=com. を要求したとします。サーバ directory.europe.siroe.com にあるエントリ cn=Babs Jensen,o=people,l=europe,dc=siroe,dc=com をポイントするスマートレフェラルをクライアントに返します。

ディレクトリでのスマートレフェラルの使い方は、RFC 2251 セクション 4.1.11 で指定されている規格に準拠しています。詳細については、RFC テキスト http://www.ietf.org/rfc/rfc2251.txt を参照してください。

次に、Console とコマンド行ユーティリティの両方を使用して、スマートレフェラルを作成する手順について説明します。

Directory Server Console を使用したスマートレフェラルの作成

1. Directory Server Console で「ディレクトリ」タブを選択します。
2. ツリーからレフェラルの追加先エントリを選択します。
3. エントリをマウスの右ボタンでクリックし、ドロップダウンメニューから「レフェラルの設定」を選択します。

   「レフェラルの編集」ダイアログボックスが表示されます。レフェラルをこのエントリではじめて作成する場合、「レフェラルリスト」は空白になっています。
4. 「レフェラルを有効にする」チェックボックスを選択します。
5. 「新規レフェラルの入力」フィールドに LDAP URL を入力します。または「構築」をクリックすると、正しい URL の作成を支援するダイアログボックスが表示されます。

   URL の要素には、レフェラルエントリを保持する Directory Server のホスト名および LDAP ポート番号、さらにレフェラルエントリの DN (ターゲット DN) が含まれています。この DN は、接尾辞、サブツリー、または最下位のエントリにすることができます

6. 「レフェラルの編集」ダイアログボックスで「追加」をクリックし、レフェラルのリストに新しい LDAP URL を追加します。
7. 同じ「レフェラルの編集」ダイアログボックスの「認証」をクリックしてダイアログボックスを表示します。このダイアログボックスで、リモートサーバに対するレフェラルに続くため、現在のサーバがバインド操作で使用する資格を指定します。
8. レフェラル DN に対するアクセスを許可されたユーザの DN およびパスワードを入力します。「OK」をクリックしてこのダイアログボックスを閉じます。
9. 「レフェラルの編集」ダイアログボックスの「OK」をクリックしてウィンドウを閉じます。

   ナビゲーションツリーで、レフェラルを作成した元のエントリの代わりにレフェラルサブツリーまたはレフェラルエントリが表示されていることを確認します。元のエントリが表示されている場合は、その隣に警告アイコンが表示されます。このアイコンは、手順 7 を実行していないか、または指定したバインド DN およびパスワードにレフェラル DN にアクセスする権限がない場合に表示されます。

コマンド行を使用したスマートレフェラルの作成

`ldapmodify` コマンド行ユーティリティを使用して、コマンド行からスマートレフェラルを作成します。

スマートレフェラルを作成するには、関連するディレクトリエントリを作成し、`Referral` オブジェクトクラスを追加します。このオブジェクトクラスでは、単一の属性 `ref` を使用できます。`ref` 属性には、LDAP URL が含まれているとみなされます。

たとえば、既存のエントリ `uid=bjensen` に対するスマートレフェラルを返すには、次の行を追加します。

```
dn:uid=bjensen,ou=people,dc=siroe,dc=com
objectclass:referral
ref: ldap://directory.europe.siroe.com/cn=babs%20jensen,ou=people,
 l=europe,dc=siroe,dc=com
```

注 サーバでは、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、レフェラルとして使用する予定のある LDAP URL では、スペースの代わりに `%20` を使用する必要があります。

`directory.europe.siroe.com` へのレフェラルを持つエントリ `uid=ssarette,ou=people,dc=siroe,dc=com` を追加するには、インポート前に、LDIF ファイルに次の行を追加します。

```
dn: uid=ssarette, ou=people, dc=siroe,dc=com
objectclass: top
objectclass:person
objectclass: organizationalperson
objectclass:inetOrgPerson
objectclass:referral
cn: somi sarette
sn: sarette
uid: ssarette
ref: ldap://directory.europe.siroe.com/cn=somi%20sarette,ou=people,
 l=europe,dc=siroe,dc=com
```

DN パスにすでにレフェラルがある場合は、`ldapmodify` で `-M` オプションを使用します。`-M` オプションについては、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照してください。

スマートレフェラルについては、『iPlanet Directory Server 導入ガイド』を参照してください。`ldapmodify` ユーティリティについては、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照してください。

接尾辞レフェラルの作成

次に、接尾辞 (suffix) でレフェラルを作成する手順について説明します。つまり、データベースまたはデータベースリンクではなく、レフェラルを使用して、接尾辞が処理を行います。レフェラルについては、『iPlanet Directory Server 導入ガイド』を参照してください。

警告 レフェラルが返されるように接尾辞が構成されている場合は、接尾辞と関連付けられているデータベースに含まれる ACI は無視されます。

Console を使用した接尾辞フェラルの作成

Console を使用して接尾辞レフェラルを作成するには、次の手順を実行します。

1. Directory Server Console で、「構成」タブを選択します。
2. 左側の区画にある Data で、レフェラルの追加先となる接尾辞をクリックします。
3. 「接尾辞の設定」タブで、次のラジオボタンの 1 つを選択します。

   レフェラルを使用する : この接尾辞がクライアントアプリケーションから何らかの要求を受け取ったときに、レフェラルが返されます。

更新したレフェラルを使用する：この接尾辞がクライアントアプリケーションから更新要求を受け取ったときに、レフェラルが返されます。このオプションは、クライアントアプリケーションからの更新要求および書き込み要求を読み取り専用データベースにリダイレクトするために使用されます。

4. 「レフェラル」タブをクリックします。「新規レフェラルの入力」フィールドに LDAP URL を入力します。あるいは、「構築」をクリックすると、LDAP URL の作成がガイドされます。

   LDAP URL の構造については、付録 C「LDAP URLs」を参照してください。

5. レフェラルをリストに追加するには、「追加」をクリックします。

   複数のレフェラルを入力できます。クライアントアプリケーションからの要求に対応して、ディレクトリがレフェラルのリスト全体を返します。

6. 「保存」をクリックします。

コマンド行からの接尾辞レフェラルの作成

`ldapmodify` コマンド行ユーティリティを使用して、ディレクトリ構成ファイルのエントリに接尾辞レフェラルを追加します。接尾辞レフェラル情報が、`cn=mapping tree,cn=config` 分岐の下にあるルート接尾辞またはサブ接尾辞に追加されます。

たとえば、`ou=people,dc=siroe,dc=com` ルート接尾辞に新しい接尾辞レフェラルを追加するには、`ldapmodify` を実行します。次のように入力して、`ldapmodify` を実行します。

```
ldapmodify -a -h host -p port -D "cn=Directory Manager" -w password
```

`ldapmodify` ユーティリティはサーバにバインドし、構成ファイルに情報を追加する準備をします。

次に、接尾辞レフェラルを `ou=people,dc=siroe,dc=com` ルート接尾辞に追加します。

```
dn: cn="ou=people,dc=siroe,dc=com",cn=mapping tree,cn=config
objectclass: extensibleObject
objectclasss: nsmappingtree
nsslapd-state:referral
nsslapd-referral: ldap://zanzibar.com/
```

`nsslapd-state` 属性が `referral` に設定されます。これは、この接尾辞への要求に対してレフェラルが返されることを表します。`nsslapd-referral` 属性には、接尾辞によって返されたレフェラル (前述の例では Zanzibar.com サーバへのレフェラル) の LDAP URL が含まれます。

また、nsslapd-state 属性に referral on update を設定することもできます。つまり、更新要求以外のすべての操作に対して、このデータベースが使用されます。クライアントアプリケーションが referral on update に設定された接尾辞に更新を要求したときに、クライアントはレフェラルを受け取ります。

接尾辞構成属性については、77 ページの「接尾辞の属性」を参照してください。

第 4 章

ディレクトリデータベースへのデータの実装

データベースには、Directory Server によって管理されるディレクトリのデータが含まれます。この章では、次のディレクトリデータベースの実装手順について説明します。

- 読み取り専用モードの有効化と無効化
- 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と復元

読み取り専用モードの有効化と無効化

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、対象となるデータベースに対して読み取り専用モードを有効にすると、その時点でのデータベースの状態の正確なイメージを確保できます。

Directory Server Console とコマンド行ユーティリティでは、エクスポート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、システムが多重マスター構成になっている場合、この問題は発生しません。

読み取り専用モードの有効化

1. Directory Server Console で「構成」タブを選択し、ナビゲーションツリーの Data フォルダを展開します。
2. 読み取り専用モードにするデータベースを選択し、右側の区画にある「データベースの設定」タブをクリックします。
3. 「データベースは読み取り専用です」チェックボックスを選択します。

4. 「保存」をクリックします。

   変更内容はすぐに有効になります。

インポートまたは復元の操作を実行する前に、操作対象のデータベースが読み取り専用モードになっていないことを確認してください。読み取り専用になっている場合は、次の手順に従ってデータベースを更新できるようにします。

読み取り専用モードの無効化

1. Directory Server Console で「構成」タブを選択し、Data ツリーを展開します。
2. 更新可能にするデータベースを選択し、右側の区画にある「データベースの設定」タブをクリックします。
3. 「データベースは読み取り専用です」チェックボックスの選択を解除します。
4. 「保存」をクリックします。

   変更内容はすぐに有効になります。

データのインポート

iPlanet Directory Server では、3 つの方法でデータをインポートできます。

- Directory Server Console からインポートする

  Directory Server Console を使用して、データベースリンクを含め、すべてのデータベースにデータを追加します。

- データベースを初期化する

  Directory Server Console を使用して、1 つのデータベースにデータをインポートします。この方法では、データベース内のすべてのデータが上書きされます。

- コマンド行を使用してデータをインポートする

  コマンド行ユーティリティを使用してデータをインポートします。

注	インポートする LDIF (LDAP Data Interchange Format) 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。

次の表に、データベースのインポートと初期化の違いを示します。

表 4-1　データのインポートとデータベースの初期化の比較

比較ドメイン	データのインポート	データベースの初期化
データベースの上書き	×	○
LDAP 処理	追加、変更、削除	追加のみ
性能	低速	高速
パーティション特性	すべてのパーティションが対象	ローカルパーティションのみ
サーバの障害への対応	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)	不可分 (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	Console と同じマシン上	Console またはサーバと同じマシン上
構成情報のインポート (cn=config)	○	×

次の節では、データのインポートについて説明します。

- 「Console を使用したインポートの実行」(137 ページ)
- 「Console を使用したデータベースの初期化」(139 ページ)
- 「コマンド行からのインポート」(140 ページ)

警告　インポートしたすべての LDIF ファイルには、ルート接尾辞が含まれている必要があります。

Console を使用したインポートの実行

Directory Server Console からインポート操作を実行する場合は、エントリの変更と削除を行うのと同様にデータの追加を行うために `ldapmodify` 処理が実行されます。この処理は、Directory Server によって管理されるすべてのデータベース、および Directory Server がデータベースリンクを保持しているリモートデータベースが対象になります。

インポートを実行するには、ディレクトリマネージャ (Directory Manager) としてログインする必要があります。

Directory Server Console からデータをインポートするには、次の手順を実行します。

1. Directory Server Console で、「タスク」タブを選択します。画面の一番下までスクロールし、「データベースのインポート」を選択します。

   また、「構成」タブの「Console」メニューから「インポート」を選択してインポートすることもできます。

   「データベースのインポート」ダイアログボックスが表示されます。

2. 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」をクリックしてインポートするファイルを選択します。

   そのディレクトリがあるマシンとは別のリモートマシン上で Console を実行している場合、フィールド名は「LDIF ファイル (Console を実行するマシン上)」と表示されます。これによって、参照しているディレクトリがカレントディレクトリではないことがわかります。ここで参照するファイルシステムは、コンソールを実行しているマシン上にあります。

3. このボックスで、次のオプション (複数可) を選択します。

   追加のみ : LDIF ファイルでは、デフォルトの追加命令に加えて、変更命令と削除命令を含むことがあります。サーバが追加以外の処理を無視するように設定する場合は、「追加のみ」チェックボックスを選択します。

   エラー時に続行 : エラーが発生してもサーバがインポートを続けるように設定する場合は、「エラー時に続行」チェックボックスを選択します。たとえば、新しいエントリとすでにデータベース上に存在するデータの両方を含む 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。既存エントリが拒否エントリ用ファイルに記録され、すべての新しいエントリが追加されます。

4. 「拒否エントリ用ファイル」フィールドには、サーバ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」ボタンをクリックして拒否データを保存するファイルを選択します。

   たとえば、サーバはデータベース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。Console は、サーバ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。

   このフィールドを空白のままにすると、拒否されたエントリは記録されません。

5. 「OK」をクリックします。

   インポートが実行され、さらにインデックスが作成されます。

Console を使用したデータベースの初期化

データベース上に存在するデータは上書きできます。次の節では、Console を使用したデータベースの初期化について説明します。

データベースを初期化するには、ディレクトリマネージャ (Directory Manager) としてログインする必要があります。これは、ルートエントリを含む LDIF ファイルをインポートするには、ディレクトリマネージャ (root DN) としてディレクトリにバインドする必要があるためです。ルートエントリへのアクセス権が認められるのは、ディレクトリマネージャだけです (たとえば、`dc=siroe,dc=com` などがルートエントリです)。

警告 LDIF ファイルからデータベー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` 接尾辞を上書きしないように注意してください。この接尾辞を上書きしてしまうと、重要な情報が削除されてしまい、iPlanet サーバの再インストールが必要になります。

Directory Server Console を使用してデータベースを初期化するには、次の手順を実行します。

1. Directory Server Console で、「構成」タブを選択します。
2. 左側にあるナビゲーション区画の Data ツリーを展開します。初期化するデータベースの接尾辞で示されたディレクトリを展開してから、データベースをクリックします。
3. データベースをマウスの右ボタンでクリックし、「データベースの初期化」を選択します。

   「データベースの初期化」は、「オブジェクト」メニューから選択することもできます。
4. 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」をクリックしてマシン上のファイルを選択します。
5. 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 Console 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のリモートマシンから Console 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。

   サーバマシンから : LDIF ファイルがリモートサーバ上にあることを示します。デフォルトでは、Console は次のディレクトリ内でファイルを検索します。

   `/var/ds5/slapd-`*`serverID`*`/ldif`
6. 「OK」をクリックします。

コマンド行からのインポート

コマンド行を使用してデータをインポートするには、3 つの方法があります。

- `/usr/sbin/directoryserver ldif2db` を使用する。

  この方法でインポートすると、データベースの内容は上書きされます。また、サーバをあらかじめ停止させておく必要があります。

- `/usr/sbin/directoryserver ldif2db-task` を使用する。

  この方法でインポートすると、データベースの内容は上書きされます。サーバを停止させておく必要はありません。

- `/usr/sbin/directoryserver ldif2ldap` を使用する。

  この方法では、LDAP によって LDIF ファイルが追加されます。この方法を使用すると、すべてのデータベースにデータを追加できます。

ldif2db コマンドを使用したインポート

`/usr/sbin/directoryserver ldif2db` コマンドは、指定したデータベースのデータを上書きします。このコマンドでは、インポートを開始する前に、サーバの停止が要求されます。

デフォルトでは、まず既存の `o=NetscapeRoot` 構成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` 構成情報と結合されます。

警告 このコマンドは、データベース内のデータを上書きします。

サーバを停止して 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、サーバを停止させます。

   ```
   # /usr/sbin/directoryserver stop
   ```

2. `ldif2db` コマンドを使用します。

   ```
   # /usr/sbin/directoryserver ldif2db
   ```

次の例では、コマンドを使用して LDIF ファイルを 1 つのデータベースにインポートします。

警告 `-n` オプションで、LDIF ファイルに含まれる接尾辞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。

```
#!/bin/sh
/usr/sbin/directoryserver ldif2db -n Database1 \
 -i /usr/iplanet/servers/slapd-siroe/ldif/demo.ldif \
 -i /usr/iplanet/servers/slapd-siroe/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	説明
-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 引数を使用できる。複数のファイルをインポートする場合、サーバ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

ldif2db-task コマンドを使用したインポート

上記と同様に /usr/sbin/directoryserver ldif2db-task は、指定したデータベースのデータを上書きします。このコマンドを使用してインポートを実行する場合は、サーバを動作させておく必要があります。

警告　このコマンドは、データベース内のデータを上書きします。

次の例では、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、ディレクトリマネージャとして認証する必要があります。

```
#!/bin/sh
/usr/sbin/directoryserver ldif2db-task \
  -D "cn=Directory Manager" -w password -n Database1 \
  -i /usr/iplanet/servers/slapd-siroe/ldif/demo.ldif
```

表 4-3　例で使用した ldif2db.pl オプションの説明

オプション	説明
-D	ディレクトリマネージャの DN を指定する
-w	ディレクトリマネージャ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 引数を使用できる。複数のファイルをインポートする場合、サーバ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

ldif2ldap コマンドを使用したインポート

`usr/sbin/directoryserver ldif2ldap` を使うと、LDAP を通して LDIF ファイルが追加されます。このコマンドを使用すると、すべてのディレクトリデータベースに対して同時にデータをインポートできます。このコマンドを使用してインポートを実行するには、サーバを動作させておく必要があります。

次の例では、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、コマンド行でディレクトリマネージャに資格を付与する必要があります。最後のパラメタは、インポートする 1 つ以上の LDIF ファイル名です。

```
#!/bin/sh
/usr/sbin/directoryserver ldif2ldap "cn=Directory Manager" password \
  /usr/iplanet/servers/slapd-siroe/ldif/demo.ldif
```

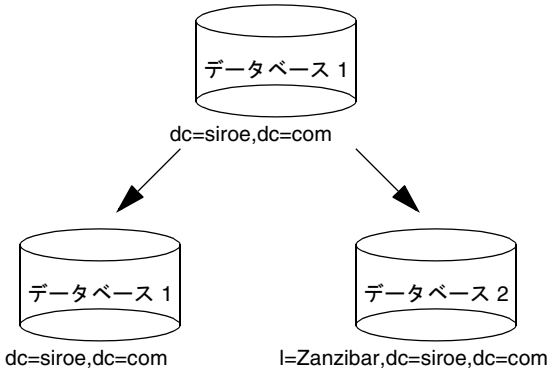
データのエクスポート

LDIF (LDAP Data Interchange Format) を使用すると、データベースのエントリをデータベースからエクスポートできます。LDIF は、RFC 2849「The LDAP Data Interchange Format (LDIF) - Technical Specification」に記載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。

- データベース上のデータのバックアップ
- ほかの Directory Server へのデータコピー
- ほか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データベースの再実装

たとえば、ディレクトリが 1 つのデータベース内に置かれている場合に、その内容を次のように 2 つのデータベースに分割するとします。



新しいデータベースにデータを実装するには、データベース 1 の内容をエクスポートして、それを新しいデータベース 1 および 2 にインポートする必要があります。

データをエクスポートするには、iPlanet Directory Server の Console か、コマンド行ユーティリティを使用します。次の節では、各方法の詳細について説明します。

- 「Console を使用した LDIF へのディレクトリデータのエクスポート」(143 ページ)
- 「Console を使用した LDIF への単一のデータベースのエクスポート」(144 ページ)
- 「コマンド行からの LDIF へのエクスポート」(145 ページ)

エクスポート処理を実行しても、構成情報 (`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、サーバを停止しないでください。

Console を使用した LDIF へのディレクトリデータのエクスポート

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 ファイルがサーバ上にある場合は、サーバと同じマシン上にあるデータベースの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、データベースおよびデータベースリンクのすべてをエクスポートできます。

サーバの動作中に、Directory Server Console 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 Console で、「タスク」タブを選択します。画面の一番下までスクロールし、「データベースのエクスポート」を選択します。

   また、「構成」タブの「Console」メニューから「エクスポート」を選択してすべてのデータベースをエクスポートすることもできます。

   「データベースのエクスポート」ダイアログボックスが表示されます。

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。

   リモートサーバ上で Console 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保存されています。

   `/var/ds5/slapd-`*`serverID`*`/ldif`

3. サーバ以外のリモートマシン上で Console 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。Console の実行マシン上の LDIF ファイルにエクスポート先を指定する場合は、「ローカルマシンへ」を選択します。サーバのマシン上に置かれている LDIF ファイルにエクスポート先に指定する場合は、「サーバマシンへ」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、「データベース全体」ラジオボタンを選択します。

   データベースに含まれる接尾辞の 1 つのサブツリーだけをエクスポートする場合は、「サブツリー」ラジオボタンを選択して、「サブツリー」テキストボックスに接尾辞の名前を入力します。これによって、複数のデータベースに含まれるサブツリーをエクスポートできます。

   「参照」をクリックして接尾辞またはサブツリーを選択することもできます。

5. 「OK」をクリックすると、ファイルがエクスポートされます。

Console を使用した LDIF への単一のデータベースのエクスポート

サーバの動作中に、Directory Server Console から LDIF に単一のデータベー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 Console で、「構成」タブを選択します。
2. 左側にあるナビゲーション区画の Data ツリーを展開します。エクスポートするデータベースによって維持される接尾辞で示されたディレクトリを展開します。接尾辞で示されたディレクトリの下から、エクスポートするデータベースを選択します。

3. データベースをマウスの右ボタンでクリックし、「データベースのエクスポート」を選択します。

   「データベースのエクスポート」は、「オブジェクト」メニューから選択することもできます。

   「パーティションのエクスポート」ダイアログボックスが表示されます。

4. 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」をクリックしてファイルを選択します。

   「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保存されています。

   `/var/ds5/slapd-`*serverID*`/ldif`

5. 「OK」をクリックすると、ファイルがエクスポートされます。

コマンド行からの LDIF へのエクスポート

`/usr/sbin/directoryserver db2ldif` を使用すると、データベースを LDIF にエクスポートできます。このコマンドは、サーバが動作中または停止中であるときに、データベー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。

```
# /usr/sbin/directoryserver db2ldif
```

次の例では、2 つの接尾辞のデータベースが 1 つの LDIF ファイルにエクスポートされます。

```
# /usr/sbin/db2ldif -n database1 -a output.ldif \
        -s "dc=siroe,dc=com" -s "o=NetscapeRoot"
```

次の表に、これらの例で使用されている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	説明
`-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
`-a`	サーバ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 `/var/ds5/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接尾辞を指定する。複数の `-s` 引数を使用すると、複数の接尾辞を指定することができる

データのバックアップと復元

Directory Server Console やコマンド行ユーティリティを使用して、データベースのバックアップと復元を行うことができます。

次の節では、データのバックアップと復元の手順について説明します。

警告　バックアップや復元の処理中には、サーバを停止しないでください。

すべてのデータベースのバックアップ

次に、Directory Server Console を使って、あるいはコマンド行から、ディレクトリ内のすべてのデータベースをバックアップするための手順を示します。

注　データベースリンクによって連鎖させたリモートサーバ上にデータベースがある場合は、この方法でデータをバックアップすることはできません。

Server Console を使用したすべてのデータベースのバックアップ

Directory Server Console を使用してデータベースをバックアップする場合は、すべてのデータベースの内容と、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、サーバが動作中でも実行できます。

Server Console を使用してデータベースをバックアップするには、次の手順を実行します。

1. Directory Server Console で、「タスク」タブを選択します。
2. 「Directory Server のバックアップ」をクリックします。

   「ディレクトリのバックアップ」ダイアログボックスが表示されます。

3. 「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。

   /var/ds5/slapd-*serverID*/bak/*YYYY_MM_DD_hh_mm_ss*

   ここでの *serverID* とは、ディレクトリサーバの名前で、バックアップが作成された日時を入れるため *backupDir* 名が生成されます。

4. 「OK」をクリックすると、バックアップが作成されます。

コマンド行からのすべてのデータベースのバックアップ

`/usr/sbin/directoryserver db2bak` コマンドを使用すると、コマンド行からデータベースをバックアップできます。このコマンドは、サーバ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、この方法では構成情報をバックアップできません。構成情報のバックアップについては、148 ページの「dse.ldif 構成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。

```
# /usr/sbin/directoryserver db2bak backupDir
```

backupDir パラメタに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` から生成されます。

次の例では、指定したディレクトリに、すべてのデータベースがバックアップされます。

```
# /usr/sbin/directoryserver db2bak /var/ds5/slapd-sv/bak/checkpoint
```

単一のデータベースのバックアップ

次の条件を満たしている場合は、ここで説明する方法を使用できます。

- ディレクトリサーバが停止中である
- 作成したバックアップを使って同じサーバにデータベースを復元する

注 このバックアップ方法は、リモートサーバ上のデータベース (データベースリンクによって連鎖されたデータベース) のデータのバックアップには使用できません。またそのバックアップデータを使ってコンシューマまたはハブのレプリカを初期化することはできません。

単一のデータベースをバックアップするには、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、サーバを停止させます。

   ```
   # /usr/sbin/directoryserver stop
   ```

2. バックアップするデータベースが置かれたディレクトリに移動します。

   ```
   # cd /var/ds5/slapd-serverID/db
   ```

3. このディレクトリ内のすべてのファイルを、作成したバックアップ用のディレクトリにコピーします。`slapd-serverID/bak/` の下にディレクトリを作成しないでください。これは Directory Server Console が、このディレクトリ内のバックアップがグローバルなものであると認識してしまうためです。

dse.ldif 構成ファイルのバックアップ

Directory Server は、自動的に `dse.ldif` 構成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。

```
/var/ds5/slapd-serverID/config
```

`dse.ldif` ファイルの内容を変更する場合は、サーバが `dse.ldif` ファイルに変更を書き込む前に、`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。構成を保存する必要がある場合には、いずれかのファイルのコピーを作成してください。

すべてのデータベースの復元

次に、Directory Server Console を使って、あるいはコマンド行から、ディレクトリ内のすべてのデータベースを復元するための手順を示します。

注	データベースを復元するときは、サーバが動作している必要があります。ただし、復元中にデータベースの処理を行うことはできません。

Console を使用したすべてのデータベースの復元

データベースが壊れた場合、Directory Server Console 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。このプロセスでは、まずサーバを停止してから、データベースおよび関連するインデックスファイルをバックアップファイルからデータベースのディレクトリにコピーします。

警告 データベースを復元すると、既存のデータベースファイルが上書きされます。

以前に作成したバックアップからデータベースを復元するには、次の手順を実行します。

1. Directory Server Console で、「タスク」タブを選択します。
2. 「Directory Server の復元」をクリックします。

   「ディレクトリの復元」ダイアログボックスが表示されます。
3. 「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たすべてのバックアップが表示されます。

   `/var/ds5/slapd-`*`serverID`*`/bak`
4. 「OK」をクリックすると、データベースが復元されます。

コマンド行からのデータベースの復元

次に示すコマンドを使用すると、コマンド行からデータベースを復元できます。

- `/usr/sbin/directoryserver bak2db` コマンドを使用する。このコマンドを使用する場合は、サーバを停止させる必要がある
- `/usr/sbin/directoryserver bak2db-task` コマンドを使用する。このコマンドは、サーバが動作中でも実行できる

bak2db コマンドの使用

サーバ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、サーバを停止させます。

   ```
   # /usr/sbin/directoryserver stop
   ```
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。

   ```
   # /usr/sbin/directoryserver bak2db backupDir
   ```

警告 データベースを復元すると、既存のデータベースファイルが上書きされます。

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。

```
# /usr/sbin/directoryserver bak2db /var/ds5/slapd-sv/bak/2001_07_01_11_34_00
```

bak2db-task コマンドの使用

サーバの実行時にコマンド行を使ってディレクトリを復元するには、次のコマンドを使用します。

```
# /usr/sbin/directoryserver bak2db-task
```

警告 データベースを復元すると、既存のデータベースファイルが上書きされます。

次の例では、LDIF ファイルをインポートします。

```
#!/bin/sh
/usr/sbin/directoryserver bak2db-task -D "cn=Directory Manager" \
   -w password -a /usr/iplanet/servers/slapd-siroe/bak/checkpoint
```

表 4-5 例で使用した bak2db-task オプションの説明

オプション	説明
-D	ディレクトリマネージャの DN を指定する
-w	ディレクトリマネージャのパスワードを指定する
-a	バックアップディレクトリの絶対パスを定義する

単一のデータベースの復元

次の条件を満たしている場合は、ここで説明する方法を使用できます。

- Directory Server が停止中である
- 以前に作成した同じサーバ上の同じデータベースのバックアップからデータベースを復元する

単一のデータベースを復元するには、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、サーバを停止させます。

   ```
   # /usr/sbin/directoryserver stop
   ```

2. 復元するバックアップが置かれたディレクトリに移動します。

3. バックアップ内容で上書きするデータベースが置かれたディレクトリに、すべてのファイルをコピーします。データベースのディレクトリの位置は、次のとおりです。

   ```
   /var/ds5/slapd-serverID/db
   ```

   たとえば、次のように入力します。

   ```
   cp backupDir/* /var/ds5/slap-siroe/db/databaseDir
   ```

レプリケートされたエントリを含むデータベースの復元

ここでは、サプライヤサーバおよびコンシューマサーバ上のデータベースを復元させる方法、および復元後にサプライヤとコンシューマを同期させる方法について説明します。

サプライヤレプリカの復元

ほかのサーバ (サプライヤレプリカ) にデータを提供しているデータベースを復元する場合は、復元されたデータベースから更新を受け取るすべてのコンシューマレプリカ (コンシューマサーバ、ハブサーバ、マルチマスターレプリケーション環境ではほかのサプライヤサーバ) を初期化し直す必要があります。

復元されたデータベースの更新履歴ログは、復元処理中に消去されます。再初期化が必要であることを示すメッセージが、サプライヤサーバのログファイルに記録されます。

コンシューマの初期化については、第 8 章「レプリケーションの管理」を参照してください。

コンシューマレプリカの復元

サプライヤサーバ (supplier server) から受け取ったデータを含むデータベースを復元する場合、次のいずれの状況が想定されます。

- サプライヤサーバでは、更新履歴ログのエントリの期限が切れていない

  ただし、更新履歴ログの最大維持期間属性で設定された値よりも短い期間内にバックアップが作成された場合に限られます。この属性は `nsslapd-changelogmaxage` という名前で、`cn=changelog5,cn=config` エントリ内に置かれます。このオプションについては、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照してください。

  ローカル コンシューマ (consumer) を復元して通常の操作を続けることができます。ただし、コンシューマレプリカを復元中は、コンシューマサーバを停止する必要があります。コンシューマレプリカの復元中にレプリケーションが行われると、レプリケーションプロセスで多くのエラーが発生します。

- ローカルバックアップ後のサプライヤサーバでは、更新履歴ログのエントリの期限が過ぎている

  コンシューマの初期化をやり直す必要があります。コンシューマの再初期化については、315 ページの「コンシューマの初期化」を参照してください。

レプリケーション管理については、第 8 章「レプリケーションの管理」を参照してください。

dse.ldif 構成ファイルの復元

`dse.ldif` 構成ファイルを復元するには、サーバを停止してから、150 ページの「単一のデータベースの復元」の手順に従って、自分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を作成します。データのコピーが完了したら、サーバを再起動します。

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。

`/var/ds5/slapd-`*`serverID`*`/config`

`dse.ldif.startOK` ファイルには、サーバ起動時に `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。

第 5 章

高度なエントリの管理

ユーザがしばしば必要とするグループを作成したり、共通の属性値を共有したりするなどディレクトリ内のデータの階層構造を超えて、エントリを管理することがあります。iPlanet 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。

この章では、次のグループ化メカニズムとグループ化の手順について説明します。

- 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

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 (topology) を決定しておく必要があります。詳細は、『iPlanet Directory Server 導入ガイド』を参照してください。

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。このメカニズムは、Directory Server の以前のバージョンでも提供されており、主に以前のバージョンのサーバとの互換性を維持するために使用されます。同等のロール定義の作成手順については、156 ページの「ロールの割り当て」を参照してください。

次の節では、静的グループと動的グループの管理方法について説明します。グループの概念については、『iPlanet Directory Server 導入ガイド』を参照してください。グループの管理については、『Managing Servers with Directory Console』を参照してください。

グループ定義は特別なエントリで、静的なリストにメンバーの名前を指定するか、または動的なエントリセットを定義するフィルタを指定します。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、ルート接尾辞の下の `ou=Groups` に格納されます。

静的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リストされます。

動的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は、`memberURL` 属性に指定されたフィルタによって定義されます。動的グループのメンバーは、評価のたびにフィルタにマッチするエントリです。

エントリエディタは、両方のタイプのグループエントリを管理します。このダイアログボックスを使用すると、グループに名前を付けたあと、メンバーのリストまたはフィルタを作成または変更できます。この節では、グループの作成と変更に関する次の手順について説明します。

- 「新しい静的グループの追加」(154 ページ)
- 「新しい動的グループの追加」(155 ページ)
- 「グループ定義の変更」(155 ページ)
- 「グループ定義の削除」(156 ページ)

新しい静的グループの追加

1. Directory Server Console で、「ディレクトリ」タブを選択します。
2. 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マウスの右ボタンでクリックします。「新規」の「グループ」を選択します。

   あるいは、「オブジェクト」メニューで「新規」の「グループ」を選択します。
3. 左側の区画で、「一般」をクリックします。「グループ名」フィールドに新しいグループの名前を入力します。

   グループ名は省略できません。
4. 「説明」フィールドに新しいグループの説明を入力します。

5. 左側の区画で、「メンバー」をクリックします。右側の区画で、「静的グループ」タブを選択します。「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「ユーザとグループの検索」ダイアログボックスが表示されます。

6. 「検索」ドロップダウンリストで、検索対象のエントリの種類 (ユーザ、グループ、またはその両方) を選択し、「検索」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、「OK」をクリックします。

注 静的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静的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、89 ページの「連鎖ポリシーの構成」を参照してください。

---

7. 左側の区画で「言語」をクリックし、グループが使用する言語に特有の情報を追加します。

8. 「OK」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、そのグループを作成した位置の子の 1 つとして表示されます。

新しい動的グループの追加

1. 154 ページの「新しい静的グループの追加」の手順 1 ～ 4 を実行します。

2. 左側の区画で、「メンバー」をクリックします。右側の区画で、「動的グループ」タブを選択します。「追加」をクリックして、データベースを照会するための LDAP URL を作成します。

   標準の「LDAP URL の構築とテスト」ダイアログボックスが表示されます。

3. テキストフィールドに LDAP URL を入力するか、または「構築」を選択し、ガイドに従って、グループに適用するフィルタを含む LDAP URL を作成します。URL の構築が完了したら「OK」をクリックします。

4. 左側の区画で「言語」をクリックし、グループが使用する言語に特有の情報を追加します。

5. 「OK」をクリックすると、新しいグループが作成されます。

   新しいグループがディレクトリツリーに表示されます。

グループ定義の変更

1. Directory Server Console で、「ディレクトリ」タブを選択します。

2. ディレクトリツリーで、変更するグループを表すエントリをダブルクリックするか、または「オブジェクト」メニューの「開く」を選択します。

   グループ定義エントリの「エントリの編集」ダイアログボックスが表示されます。

3. 「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。「OK」をクリックします。

   変更を確認するには、「表示」メニューの「再読み込み」を選択します。

グループ定義の削除

いずれかのタイプのグループを削除するには、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、アプリケーションからより効率的で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 (entry) のロールを読み取るだけで済みます。

この節では、次の事項について説明します。

- 「ロールについて」(156 ページ)
- 「ロールの制限事項」(157 ページ)
- 「Console を使用したロールの管理」(158 ページ)
- 「コマンド行からのロールの管理」(163 ページ)
- 「ロールの安全な使い方」(165 ページ)

ロールについて

各ロールはメンバー、またはそのロールを所有するエントリを持ちます。グループと同じようにロールのメンバーを明示的または動的に指定できます。ロールメカニズムは、エントリが所属するすべてのロール定義の DN を含む、`nsRole` 属性を自動的に生成します。

ロールのメンバーの指定方法は、使用するロールのタイプによって異なります。iPlanet Directory Server では、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 : 明示的にメンバーエントリにロールを割り当てる

- フィルタを適用したロール : 指定した LDAP フィルタにマッチするエントリを割り当てる。これにより、各エントリに含まれている属性に応じてロールが異なる
- 入れ子状のロール : 別のロールを含むロールを作成できる

管理されているロールを使用すると、管理者は、対象となるエントリに nsRoleDN 属性を追加することにより、特定の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、静的グループと同じです。

フィルタを適用したロールは、動的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。ただし、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。サーバが、フィルタ文字列にマッチする、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、そのエントリには常にロールを識別する nsRole 属性が含まれています。

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。ロールを使って処理を実行すると、グループを使う場合よりもサーバ側でより多くの資源が消費されます。これは、クライアントアプリケーションのためにサーバがその処理を実行するためです。ただし、ロールのメンバーの検査方法は一貫しており、サーバ側で透過的に実行されます。

注

1. ロールメカニズムで使用されるのは、nsRole 属性だけで、この属性はすべての変更操作から保護されています。ただし、読み取りは可能です。読み取ることができないようにアクセス制御を定義することもできます。

2. 検索フィルタでは、nsRole 属性を使用できません。アプリケーションが nsRole 属性を読み取るようにするには、まず別のフィルタを使用して検索を実行し、次に検索処理が返したエントリの nsRole 属性の値を読み取ります。

ディレクトリでのロールの使用方法については、『iPlanet Directory Server 導入ガイド』を参照してください。

ロールの制限事項

ディレクトリサービスをサポートするロールを作成する場合は、次の制限事項を考慮する必要があります。

ロールと連鎖：連鎖機能を使用してディレクトリツリーを複数のサーバに分散している場合は、ロールを定義するエントリをそれらのロールを所有するエントリと同じサーバに配置する必要があります。連鎖を介して、サーバ A が別のサーバ B からエントリを受け取る場合は、それらのエントリにはサーバ B で定義されたロールが含まれますが、サーバ A で定義されたロールは割り当てられません。

フィルタを適用したロールでは、CoS によって生成された属性を使用できない：フィルタを適用したロールでは、CoS 仮想属性の値に基づいたフィルタ文字列を使用できません (167 ページの「CoS について」を参照)。ただし、CoS 定義の指示子属性は、ロール定義によって生成された `nsRole` 属性を参照できます (182 ページの「ロールに基づく属性の作成」を参照)。

Console を使用したロールの管理

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。

ロールを作成するときに、ユーザが本人をロールへ追加したり削除したりする権限を付与するかどうかを決めておく必要があります。ロールとアクセス制御については、165 ページの「ロールの安全な使い方」を参照してください。

管理されているロールの作成

管理されているロールを使用して、メンバーを明示的に列挙するリストを作成できます。管理されているロールは、`nsRoleDN` 属性をそのエントリに追加することによってエントリに追加されます。

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 Console で「ディレクトリ」タブを選択します。
2. ディレクトリツリーから新しいロールの親エントリを選択します。

3. 「オブジェクト」メニューで「新規」の「ロール」を選択します。あるいは、エントリをマウスの右ボタンでクリックして、「新規」の「ロール」を選択することもできます。

   「新規ロールの作成」ダイアログボックスが表示されます。

4. 左側の区画で、「一般」をクリックします。「ロール名」フィールドに新しいロールの名前を入力します。

   ロール名は省略できません。

5. 「説明」フィールドに新しいロールの説明を入力します。

6. 左側の区画で、「メンバー」をクリックします。

7. 右側の区画で、「管理されているロール」を選択します。「追加」をクリックして、メンバーリストに新しいエントリを追加します。

   標準の「ユーザとグループの検索」ダイアログボックスが表示されます。

8. 「検索」ドロップダウンリストから「ユーザ」を選択し、「検索」をクリックします。表示された検索結果からいずれかのエントリを選択し、「OK」をクリックします。

9. ロールへのエントリの追加が完了したら、「OK」をクリックします。

   ディレクトリに新しいロールと管理されているロールのアイコンが表示されます。

フィルタを適用したロールの作成

各エントリに含まれる特定の属性に基づいて、フィルタを適用したロールにエントリを割り当てます。この操作を行うには、LDAP フィルタを指定する必要があります。フィルタにマッチするエントリは、そのロールを所有すると言われます。

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. 158 ページの「管理されているロールの作成」の手順 1 ～ 5 を実行します。

2. 左側の区画で、「メンバー」をクリックします。

3. 右側の区画で、「フィルタが適用されているロール」を選択します。

4. テキストフィールドに LDAP フィルタを入力するか、または「構築」をクリックし、ガイドに従って LDAP フィルタを作成します。

5. 「構築」をクリックすると、標準の LDAP URL 構築ダイアログボックスが表示されます。「LDAP サーバホスト」、「ポート」、「ベース DN」、および「検索」の各フィールドは無視します (フィルタを適用したロール定義の検索範囲を指定することができないため)。

a. 「適用先」ドロップダウンリストから、フィルタを適用するエントリのタイプを選択します。

   ユーザ、グループ、またはその両方から選択できます。

b. 「属性」ドロップダウンリストから属性を選択します。この次の 2 つのフィールドを使用して、修飾子をドロップダウンリストから選択して、検索を詳しく定義し (含む、含まない、同一、同一でないなど)、テキストボックスに属性値を入力します。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. 「OK」をクリックして、フィルタを保存します。

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、フィルタにマッチするエントリが表示されます。

7. 「OK」をクリックします。

   ディレクトリに新しいロールとフィルタを適用したロールのアイコンが表示されます。

入れ子状のロールの作成

入れ子状のロールを使用して、別のロールを含むロールを作成できます。入れ子状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。入れ子状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。入れ子状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。

入れ子状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. 158 ページの「管理されているロールの作成」の手順 1 ～ 5 を実行します。
2. 左側の区画で、「メンバー」をクリックします。
3. 右側の区画で、「入れ子状態になっているロール」を選択します。
4. 「追加」をクリックして、ロールをリストに追加します。入れ子状のロールのメンバーは、ほかの既存のロールのメンバーです。

   「ロールセレクタ」ダイアログボックスが表示されます。

5. 「使用可能なロール」のリストからロールを選択し、「OK」をクリックします。
6. 「OK」をクリックします。

   ディレクトリに新しいロールと入れ子状のロールのアイコンが表示されます。

エントリのロールの表示と編集

1. Directory Server Console で、「ディレクトリ」タブを選択します。

2. ディレクトリツリーを参照し、ロールを表示または編集するエントリを選択します。「オブジェクト」メニューの「ロールの設定」を選択します。

   「ロール」ダイアログボックスが表示されます。

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。

4. 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。「OK」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、「編集」をクリックします。「エントリの編集」ダイアログボックスが表示されます。一般情報やメンバーを変更し、「OK」をクリックします。

5. 「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状のロールが表示されます。

6. 「編集」をクリックすると、エントリに関連付けられた、フィルタを適用したロールや入れ子状のロールを変更できます。「OK」をクリックして、変更を保存します。

7. ロールの変更が完了したら、「OK」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 Console 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。

   「エントリの編集」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、左側の区画で「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状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、左側の区画で「メンバー」をクリックします。

5. 「OK」をクリックして、変更を保存します。

ロールの無効化

特定のロールを無効にすることによって、そのロールに所属するメンバーを一時的に無効にすることができます。ロールを無効にすると、そのロールに所属するエントリは無効になりますが、ロール自体は無効になりません。ロールメンバーのエントリがディレクトリユーザを表す場合は、ロールによってエントリが無効になっている間、エントリはディレクトリにアクセスできません。

ロールのメンバーを一時的に無効にするには、次の手順を実行します。

1. Directory Server Console 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。
3. ロールを選択します。「オブジェクト」メニューの「無効」を選択します。

   あるいは、ロールをマウスの右ボタンでクリックして、メニューから「無効」を選択することもできます。

   ロールが無効になります。

   無効になっているエントリを表示するには、「表示」メニューの「アクティブでない状態」を選択します。ロールメンバーのアイコンの赤い横棒は、そのロールが無効になっていることを示します。

ロールの再有効化

1. Directory Server Console 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. ロールを選択します。「オブジェクト」メニューの「有効」を選択します。

   あるいは、ロールをマウスの右ボタンでクリックして、メニューから「有効」を選択することもできます。

   ロールが再び有効になります。

   無効になっているエントリを表示するには、「表示」メニューの「アクティブでない状態」を選択します。ロールが正常に表示され、そのロールが有効になっていることを示します。

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 Console 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. ロールをマウスの右ボタンでクリックし、「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。「OK」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、参照整合性プラグインを有効にし、nsRoleDN 属性を設定します。詳細は、66 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マッチするエントリとなる
- 入れ子状のロールのメンバーは、入れ子状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。

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)
入れ子状のロール	nsComplexRoleDefinition nsNestedRoleDefinition	nsRoleDN Description (省略可能)

注 場合によっては、ACI を使用して、nsRoleDN 属性の値を保護する必要があります。これは、この属性が書き込み可能であるためです。セキュリティとロールについては、165 ページの「ロールの安全な使い方」を参照してください。

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 -D "cn=Directory Manager" -w secret -h host -p 389
dn: cn=Marketing,ou=people,dc=siroe,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,ou=people,dc=siroe,dc=com
changetype: modify
add:nsRoleDN
nsRoleDN: cn=Marketing,ou=people,dc=siroe,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、次のロール定義の DN で判別されます。
cn=Marketing,ou=people,dc=siroe,dc=com

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、次の ldapmodify コマンドを実行します。

```
ldapmodify -a -D "cn=Directory Manager" -w secret -h host -p 389
dn: cn=SalesManagerFilter,ou=people,dc=siroe,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass:nsRoleDefinition
objectclass:nsComplexRoleDefinition
objectclass:nsFilteredRoleDefinition
cn: SalesManagerFilter
nsRoleFilter: o=sales managers
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。nsRoleFilter 属性は、sales managers という値を持つ o (組織) 属性がある同一サブツリー内のすべてのエントリがロールのメンバーになることを示します。

入れ子状のロール定義の例

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 ldapmodify コマンドを使用します。

```
ldapmodify -a -D "cn=Directory Manager" -w secret -h host -p 389
dn: cn=MarketingSales,ou=people,dc=siroe,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass:nsRoleDefinition
objectclass:nsComplexRoleDefinition
objectclass:nsNestedRoleDefinition
cn: MarketingSales
nsRoleDN: cn=SalesManagerFilter,ou=people,dc=siroe,dc=com
nsRoleDN: cn=Marketing,ou=people,dc=siroe,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 Bob と Pat は、どちらもこの新しい入れ子状のロールのメンバーになります。

ロールの安全な使い方

セキュリティの状況によっては、ロールの使用が適していない場合があります。新しいロールを作成するときは、エントリへのロールの割り当てやエントリからのロールの削除がどの程度簡単にできるかを考慮します。ロールへのユーザの追加やロールから削除をユーザ自身が簡単に実行できることが望ましい場合もあります。たとえば、Mountain Biking という名前の同好会のロールがある場合は、興味のあるユーザが自身を簡単に追加または削除できるようにする必要があります。

ただし、セキュリティの状況によっては、このようなオープンなロールが適していない場合があります。たとえば、アカウントの無効化に関するロールがあるとします。デフォルトでは、アカウントの無効化に関するロールには、その接尾辞に対して定義された ACI が含まれています。アカウントの無効化については、267 ページの「ユーザとロールの無効化」を参照してください。サーバ管理者は、ロールを作成するときに、ロールへのユーザの追加やロールからの削除をユーザ自身が実行できるようにするかどうかを決めます。

たとえば、ユーザ A が、管理されているロール MR を持っているとします。さらに、MR ロールが、コマンド行からアカウントの無効化を使用してロックされたとします。つまり、ユーザ A の nsAccountLock 属性は「true」として計算されるので、ユーザ A はサーバにバインドできません。ただし、ユーザがバインド済みで、MR ロールに関して現在ロックされているという通知を受けたとします。ユーザの行為を禁止する ACI がない場合は、ユーザは、自分のエントリから nsRoleDN 属性を削除し、自分でロックを解除できます。

ユーザが nsRoleDN 属性を削除できないようにするには、使用中のロールのタイプに応じて、次の ACI を使用します。

管理されているロール：管理されているロールのメンバーになっているエントリの場合は、次の ACI を使用し、該当する nsRoleDN を削除することによってユーザが自分でロック解除できないようにします。

```
aci : (targetattr="nsRoleDN")
    (targattrfilters="
 add=nsRoleDN:(!(nsRoleDN=cn=AdministratorRole,dc=siroe,dc=com)),
 del=nsRoleDN:(!(nsRoleDN=cn=nsManagedDisabledRole,dc=siroe,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：フィルタの一部になっている属性を保護することで、ユーザが属性を変更してフィルタを適用したロールを放棄できないようにします。フィルタを適用したロールで使用する属性をユーザが追加、削除、および変更できないようにする必要があります。フィルタ属性の値が計算される場合は、フィルタ属性値を変更する可能性のあるすべての属性を同様に保護する必要があります。

入れ子状のロール：入れ子状のロールは、フィルタを適用したロールと管理されているロールで構成されます。したがって、入れ子状のロールを構成する各ロールについて、前述のすべての注意点を考慮する必要があります。

サービスクラス (CoS) の定義

サービスクラス (CoS) メカニズムを使用すると、エントリに格納されない仮想属性を作成できます。属性値は、エントリがクライアントアプリケーションに送信される時に、CoS メカニズムによって生成されます。CoS を使用すると、エントリの管理が簡素化され、格納領域の必要量が減少します。

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、Console またはコマンド行を使用して構成できます。次の節では、CoS について詳しく説明し、Console 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。

- 「CoS について」(167 ページ)
- 「CoS の制限事項」(171 ページ)
- 「Console を使用した CoS の管理」(172 ページ)
- 「コマンド行からの CoS の管理」(175 ページ)
- 「ロールに基づく属性の作成」(182 ページ)
- 「CoS のセキュリティ保護」(184 ページ)

CoS について

CoS は、仮想属性とその値を CoS の適用範囲内のあらゆるエントリであるターゲットエントリすべてに定義します。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様々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よって、そのエントリで使用されるテンプレートが決まる。ターゲットエントリのこの属性には、DN が含まれている。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。CoS 値を含むテンプレートは、ターゲットの指示子属性の RDN (relative domain name) 値とテンプレートのベース DN (base DN) を組み合わせることにより決まる

注

サーバでは、CoS 仮想属性を参照するフィルタを含む LDAP 検索要求はサポートされません。LDAP 検索フィルタでは、エントリに格納されている実際の属性だけがサポートされます。この属性には、CoS 属性や `nsRole` 属性は含まれません。CoS 定義を使用して生成する属性を決定するときは、十分に注意してください。

仮想属性の値に基づいてエントリを検索するには、ディレクトリクライアントでエントリのスーパーセット (分岐全体など) を取得し、それらを並べ替えて希望するエントリを選択する必要があります。

次の節では、CoS 定義のエントリとテンプレートエントリについてさらに詳しく説明し、CoS のタイプごとに例を示します。

CoS 定義のエントリとテンプレートエントリ

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。CoS 定義のエントリは、CoS のタイプを指定する、次のオブジェクトクラスのいず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 定義エントリ (CoS definition entry) 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 (CoS template entry) には、CoS メカニズムによって生成された 1 つ以上の値があります。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。

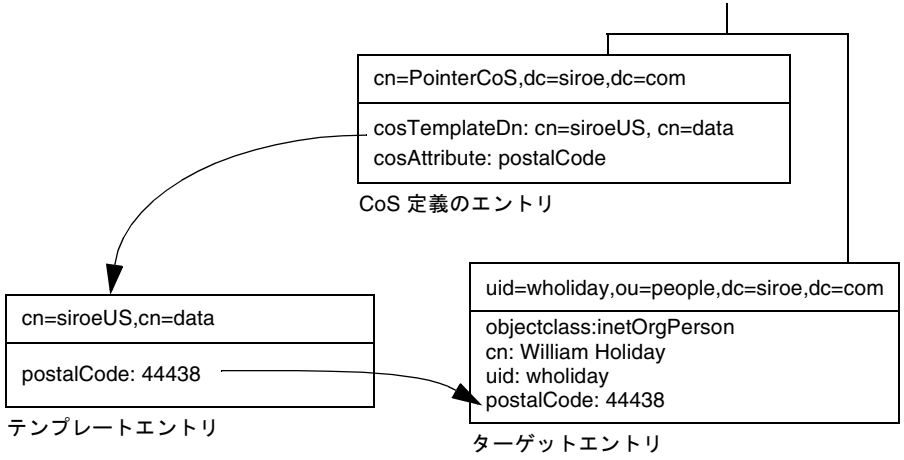
管理を容易にするため、可能なかぎり、定義エントリ、テンプレート、およびテンプレートエントリを同じ場所に置いてください。また、それらが提供する機能を説明するような名前を付けてください。たとえば、定義エントリ DN に `"cn=classicCosGenerateEmployeeType,ou=People,dc=siroe,dc=com"` などの名前を付けると、`"cn=ClassicCos1,ou=People,dc=siroe,dc=com"` よりもわかりやすくなります。

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、175 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。

ポインタ CoS の例

この例では、`dc=siroe,dc=com` の下に格納されるすべてのエントリに共通の郵便番号を定義する CoS を示します。この例の 3 つのエントリを次の図に示します。

図 5-1 ポインタ CoS 定義とテンプレートの例

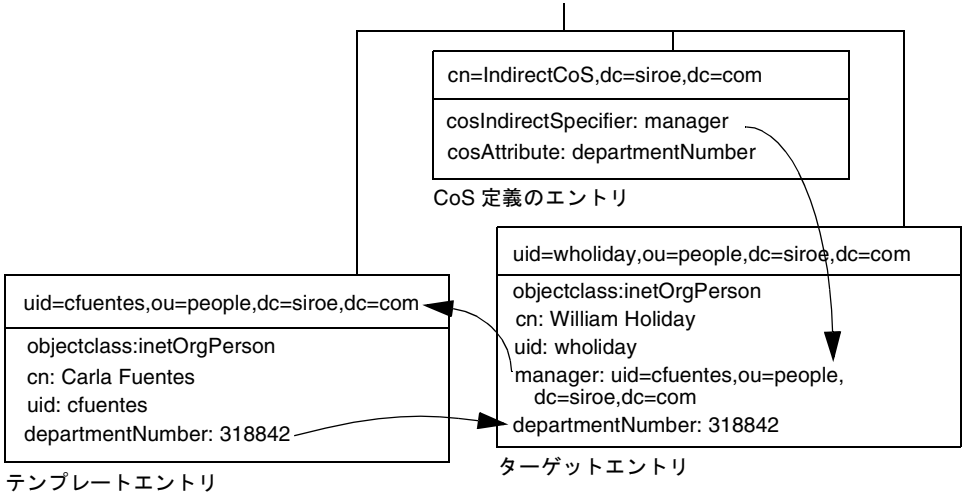


テンプレートエントリ (template entry) は、CoS 定義エントリ (CoS definition entry) 内でテンプレートエントリの DN、`cn=siroeUS,cn=data` によって識別されます。エントリ `dc=siroe,dc=com` で `postalCode` 属性が照会されるたびに、Directory Server は、テンプレートエントリ `cn=siroeUS,cn=data` 内の使用可能な値を返します。したがって、郵便コードは、エントリ `uid=wholiday,ou=people,dc=siroe,dc=com` と一緒に表示されますが、このエントリには格納されません。このメカニズムでは、CoS によっていくつかの共有属性が生成されるため、数千または数百万ものエントリのために記憶容量を大幅に節約できます。

間接 CoS の例

この間接 CoS (indirect CoS) の例では、ターゲットエントリ (target entry) の `manager` 属性を使用してテンプレートエントリ (template entry) を識別します。CoS メカニズムでは、この方法を使って、すべての従業員に対してマネージャと同じ部署番号を生成することにより、常に最新の状態を維持できます。この例の 3 つのエントリを次の図に示します。

図 5-2　間接 CoS 定義とテンプレートの例



間接 CoS 定義のエントリは、指示子属性の名前を指定します。この例では、`manager` 属性です。William Holiday のエントリは、この CoS のターゲット エントリの 1 つであり、その `manager` 属性には、`cn=Carla Fuentes,ou=people,dc=siroe,dc=com` の DN が含まれます。したがって、Carla Fuentes のエントリは、`departmentNumber` 属性値 `318842` を提供するテンプレートです。

間接指示子を使用することにより、間接 CoS はディレクトリ内のエントリをテンプレートとして使用できます。セキュリティおよび性能上の理由から、このタイプの CoS は注意深く使用してください。多くの場合、クラシック CoS 使用してターゲットエントリの位置を制限するか、柔軟性の低いポインタ CoS メカニズムを使用することにより、同じ結果を得ることができます。

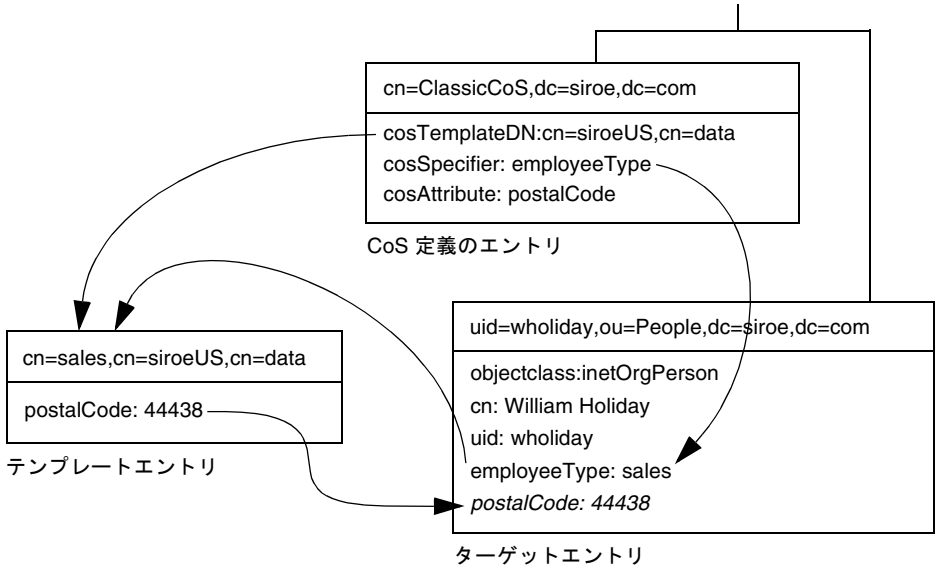
クラシック CoS の例

クラシック CoS メカニズムでは、定義エントリで指定されたベース DN とターゲットエントリの指示子からテンプレートの DN が決まります。指示子属性の値は、テンプレート DN の `cn` 値として使用されます。したがって、クラシック CoS のテンプレート DN は、次のような構造になります。

cn=*specifierValue*, *baseDN*

次の図の例は、郵便番号の値を生成するクラシック CoS (classic CoS) 定義を示しています。

図 5-3　クラシック CoS 定義とテンプレートの例



この例では、CoS 定義エントリの `cosSpecifier` 属性が、`employeeType` 属性を指定します。この属性とテンプレート DN を組み合わせると、`cn=sales,cn=siroeUS,cn=data` としてテンプレートエントリ (template entry) を識別できます。このテンプレートエントリは、`postalCode` 属性の値をターゲットエントリに与えます。

CoS の制限事項

CoS 機能は、複雑なメカニズムであり、性能およびセキュリティ上の理由から次の制限事項が適用されます。

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できません。

- `userPassword`：CoS で生成されたパスワード値は、Directory Server 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、サーバによるロールの生成に使用されない

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レフェラルからは取得できません。

CoS 仮想値と実際の値を組み合わせることはできない : CoS 属性の値では、エントリの実際の値とテンプレートの仮想値を組み合わせることはできません。CoS により実際の属性値が上書きされると、実際の値はすべてテンプレートの値に置き換えられます (178 ページの「実際の属性値の上書き」を参照)。ただし、178 ページの「複数の値を持つ CoS 属性」で説明しているように、CoS メカニズムでは、複数の CoS 定義の仮想値を組み合わせることができます。

フィルタを適用したロールでは、CoS によって生成された属性を使用できない : フィルタを適用したロールでは、CoS 仮想属性の値に基づくフィルタ文字列を使用できません。ただし、CoS 定義の指示子属性は、ロール定義によって生成された `nsRole` 属性を参照できます (182 ページの「ロールに基づく属性の作成」を参照)。

ACI (Access Control Instruction) : 格納されている通常の属性へのアクセスと同様に、CoS によって生成された属性へのアクセスが制御されます。ただし、CoS によって生成された属性値に依存するアクセス制御規則は、171 ページの「CoS の制限事項」で説明されている条件に従います。

CoS キャッシュ応答時間 : CoS キャッシュは、性能を向上させるためにすべての CoS データをメモリに保持する Directory Server の内部構造です。このキャッシュは、仮想属性の算出時に使用される CoS データの取得用に最適化されており、CoS 定義エントリおよびテンプレートエントリの更新中でも使用できます。したがって、定義エントリおよびテンプレートエントリを追加または変更すると、変更内容が反映されるまでわずかに時間がかかる場合があります。この遅延時間は、CoS 定義の数と複雑さ、および現在のサーバの負荷によって異なりますが、通常、数秒しかかかりません。

Console を使用した CoS の管理

ここでは、Directory Server Console 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。この章は、次の節で構成されています。

- 「新しい CoS の作成」 (173 ページ)

- 「既存の CoS の編集」(175 ページ)
- 「CoS の削除」(175 ページ)

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 Console で、「ディレクトリ」タブを選択します。
2. ディレクトリツリーから、テンプレートエントリを格納する親エントリを選択します。
3. 「オブジェクト」メニューをクリックするか、またはエントリをマウスの右ボタンでクリックし、「新規」の「その他」を選択します。次に「新規オブジェクト」ダイアログボックスのリストから「costemplate」を選択します。

   「属性エディタ」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。
4. 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. 命名属性を新しい cn 属性に変更する

      ほかの属性を追加して、それを命名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を削除する
   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
5. 「属性エディタ」ダイアログボックスの「OK」をクリックしてテンプレートエントリを作成します。
6. 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」の「DN のコピー」を選択します。

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. ディレクトリツリーから、新しいサービスクラスを有効にする親エントリを選択します。

2. 「オブジェクト」メニューをクリックするか、またはエントリをマウスの右ボタンでクリックし、「新規」の「サービスクラス」を選択します。

   「サービスの新規クラスの作成」ダイアログボックスが表示されます。

3. 左側の区画で、「一般」を選択します。右側の区画で、「クラス名」フィールドに新しいサービスクラスの名前を入力します。CoS 定義のエントリの cn 命名属性に名前が表示されます。「説明」フィールドにクラスの説明を入力します。

4. 左側の区画で、「属性」をクリックします。右側の区画に、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。

   使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。

5. 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、次の上書き動作のいずれかを選択します。

   - **ターゲットエントリ属性を上書きしない**: 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上書きする**: 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上書きし、operational な状態にする**: 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より上書きし、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。

6. 左側の区画で、「テンプレート」をクリックします。右の区画で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 を貼り付けます。
   - **ターゲットエントリの属性値の 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値の 1 つを使用する**: 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 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。

7. 「OK」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 Console で、「ディレクトリ」タブを選択します。
2. ディレクトリツリーから、CoS 定義を含む親エントリを選択します。CoS エントリがこの親エントリの子として表示されます。
3. CoS をダブルクリックします。

   「エントリの編集」ダイアログボックスが表示されます。
4. CoS の名前と説明を変更するには、左側の区画で「一般」をクリックします。
5. 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、左側の区画で「属性」をクリックします。
6. 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 の名前を再定義するには、左側の区画の「テンプレート」をクリックします。このダイアログボックスを使うと、CoS 定義のタイプを再定義できます。
7. 「OK」をクリックして、変更を保存します。

CoS の削除

1. Directory Server Console で、「ディレクトリ」タブを選択します。
2. ディレクトリツリーから、CoS 定義を含む親エントリを選択します。CoS エントリがこの親エントリの子として表示されます。
3. CoS をマウスの右ボタンでクリックし、「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. CoS 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。「OK」をクリックします。

コマンド行からの CoS の管理

構成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、標準的な LDAP ツールを使用して、CoS の構成と管理を行うことができます。この節では、次の事項について説明します。

- 「コマンド行からの CoS 定義のエントリの作成」(176 ページ)
- 「コマンド行からの CoS テンプレートエントリの作成」(179 ページ)
- 「ポインタ CoS の例」(180 ページ)
- 「間接 CoS の例」(180 ページ)

- 「クラシック CoS の例」(181 ページ)

コマンド行からの CoS 定義のエントリの作成

すべての CoS 定義のオブジェクトクラスは、LDAPsubentry オブジェクトクラスと 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 CoS 定義エントリ (CoS definition entry) に関連付けられたオブジェクトクラスと属性を示します。

表 5-1　CoS 定義のエントリ

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_string* cosAttribute: *list_of_attributes qualifier*
間接 CoS	objectclass: top objectclass: LDAPsubentry objectclass:cosSuperDefinition objectclass:cosIndirectDefinition cosIndirectSpecifier: *attribute_name* cosAttribute: *list_of_attributes qualifier*
クラシック CoS	objectclass: top bbjectclass: LDAPsubentry objectclass:cosSuperDefinition objectclass:cosClassicDefinition cosTemplateDn: *DN_string* cosSpecifier: *attribute_name* cosAttribute: *list_of_attributes qualifier*

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、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照)。

表 5-2　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:` *attribute_name 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。特別な状況下では、修飾子により CoS 属性値の算出方法を指定する
`cosIndirectSpecifier:` *attribute_name*	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 (template entry) 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
`cosSpecifier:` *attribute_name*	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 (template entry) 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_string*	ポインタ CoS 定義用にテンプレートエントリ (template entry) 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 (template entry) 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のみ、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、178 ページの「複数の値を持つ CoS 属性」を参照

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=siroe,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass:cosSuperDefinition
objectclass:cosPointerDefinition
cosTemplateDn:cn=siroeUS,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=siroeUS,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、その属性が通常の属性としても存在する CoS 適用範囲内のエントリでは、その属性を手動で更新できなくなります。

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、次に説明するように、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もつ指示子があるが、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。cosPriority 属性は、次で説明する CoS テンプレートエントリに定義されます。

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 を使用する場合、テンプレートエントリは LDAPsubentry オブジェクトクラスから継承され、cosTemplate オブジェクトクラスのインスタンスでもあります。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、構成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 メカニズムは、ディレクトリ内の任意の既存テンプレートエントリを参照します。テンプレートエントリをあらかじめ識別したり、LDAPsubentry オブジェクトクラスを指定する必要はありません。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のみアクセスします。

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=siroeUS,cn=data,dc=siroe,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
postalCode: 44438
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

dc=siroe,dc=com ツリーのすべてのエントリで共通の郵便番号を共有させるポインタ CoS (pointer CoS) を作成するには、次の ldapmodify コマンドを実行します。

```
ldapmodify -a -D "cn=directory manager" -w secret -h host -p 389
```

```
dn: cn=pointerCoS,dc=siroe,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass:cosSuperDefinition
objectclass:cosPointerDefinition
cosTemplateDn: cn=siroeUS,cn=data,dc=siroe,dc=com
cosAttribute: postalCode
```

```
dn: cn=siroeUS,cn=data,dc=siroe,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
postalCode: 44438
```

ここで作成した CoS テンプレートエントリ cn=siroeUS,dn=cata,dc=siroe,dc=com は、dc=siroe,dc=com 接尾辞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。

間接 CoS の例

ここで説明する間接 CoS (indirect CoS) は、ターゲットエントリ (target entry) の team 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。新しい間接 CoS 定義のエントリを dc=siroe,dc=com 接尾辞に追加するには、次の ldapmodify コマンドを実行します。

```
ldapmodify -a -D "cn=directory manager" -w secret -h host -p 389
```

```
dn: cn=indirectCoS,dc=siroe,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass:cosSuperDefinition
objectclass:cosIndirectDefinition
cosIndirectSpecifier: manager
cosAttribute: departmentNumber
```

さらに、マネージャ Carla Fuentes 用のテンプレートエントリを作成します。

```
dn:cn=Carla Fuentes,cn=data,dc=siroe,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
departmentNumber: 318842
```

最後に、マネージャ Sue Jacobs 用の 2 番目のテンプレートエントリを作成します。

```
dn:cn=Sue Jacobs,cn=data,dc=siroe,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
departmentNumber: 71776
```

定義エントリは、dc=siroe,dc=com の下にあるターゲットエントリを調べて、manager 属性を含むエントリを探します。これは、定義エントリの cosIndirectSpecifier 属性内にこの属性が指定されているためです。ターゲットエントリの manager 属性は、cn=Carla Fuentes,cn=data,dc=siroe,dc=com と cn=Sue Jacobs,cn=data,dc=siroe,dc=com という 2 つのテンプレートのどちらかをポイントすることができます。部門番号は、マネージャによって異なります。

クラシック CoS の例

次の例では、テンプレートの DN と cosSpecifier 属性内で指定された属性の組み合わせを使用して、自動的に郵便番号を生成するクラシック CoS (classic CoS) です。クラシック CoS 定義のエントリを作成するには、次の ldapmodify コマンドを実行します。

```
ldapmodify -a -D "cn=directory manager" -w secret -h host -p 389
```

```
dn: cn=classicCoS,dc=siroe,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass:cosSuperDefinition
```

```
objectclass:cosClassicDefinition
cosTemplateDn: cn=siroeUS,cn=data,dc=siroe,dc=com
cosSpecifier: employeeType
cosAttribute: postalCode override
```

最後に、セールス部門とマーケティング部門用のテンプレートエントリを作成します。

```
dn: cn=sales,cn=siroeUS,cn=data,dc=siroe,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
postalCode: 44438

dn: cn=marketing,cn=siroeUS,cn=data,dc=siroe,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
postalCode: 99111
```

ここで作成したクラシック CoS 定義のエントリは、`dc=siroe,dc=com` 接尾辞の下にあるすべてのエントリに適用されます。使用されるテンプレートには、エントリ内で検出された `employeeType` 属性と cosTemplate の DN の組み合わせに応じて、2 つのテンプレートのどちらかが指定されます。セールス部門のテンプレートは、セールス部門の社員に固有の郵便コードを提供します。マーケティング部門のテンプレートは、マーケティング部門の社員に固有の郵便コードを提供します。

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 (role-based attributes) を使用して、サーバのロック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 (template entry) 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 (CoS template entry) 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=siroe,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass:nsRoleDefinition
objectclass:nsComplexRoleDefinition
objectclass:nsFilteredRoleDefinition
cn: ManagerRole
nsRoleFilter: o=managers
Description: filtered role for managers
```

次のようにクラシック CoS 定義エントリ (CoS definition entry) を指定します。

```
dn: cn=managerCOS,dc=siroe,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass:cosSuperDefinition
objectlass:cosClassicDefinition
cosTemplateDn: cn=managerCOS,dc=siroe,dc=com
cosSpecifier: nsRole
cosAttribute: mailboxquota override
```

`cosTemplateDn` 属性が提供する値と `cosSpecifier` 属性内に指定された属性 (例では、ターゲットエントリの `nsRole` 属性) を組み合わせて、CoS テンプレートエントリ (CoS template entry) が識別されます。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` 属性値に上書きするように指定します。

対応する CoS テンプレートエントリは、次のように定義されます。

```
dn:cn="cn=ManagerRole,ou=people,dc=siroe,dc=com",cn=managerCOS,
  dc=siroe,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectlass: cosTemplate
mailboxquota: 1000000
```

テンプレートエントリは、`mailboxquota` 属性値として、1000000 を提供します。

注 ロール (role) エントリおよび CoS 定義のエントリは、適用範囲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。

CoS のセキュリティ保護

読み取り用のアクセス制御は、エントリの実際の属性と仮想属性の両方に適用されます。サービスクラスメカニズムによって生成された仮想属性は、通常の属性として読み取ることができるので、読み取り保護も同様の方法で指定します。

ただし、CoS 値をセキュリティで保護するには、定義エントリ、テンプレートエントリ、ターゲットエントリなど、使用するすべての情報のソースを保護する必要があります。これは更新処理でも同様です。情報のソースから生成された値を保護するために、各情報のソースに対する書き込みを制御する必要があります。

次の節では、各 CoS エントリのデータに読み取りおよび書き込み保護を設定する際の一般的な原則について説明します。個別の ACI (Access Control Instruction) の定義手順については、第 6 章「アクセス制御の管理」を参照してください。

CoS 定義のエントリの保護

CoS 定義のエントリには、生成された属性の値は含まれませんが、値を検索するための情報を提供します。CoS 定義のエントリを読み取ると、値を含むテンプレートエントリの検索方法がわかり、このエントリを書き込むと、仮想属性の生成方法を変更できます。

したがって、CoS 定義のエントリに読み取りと書き込みの両方のアクセス制御を定義する必要があります。

CoS テンプレートエントリの保護

CoS テンプレートエントリには、生成された CoS 属性の値が含まれます。したがって、少なくともテンプレートの CoS 属性の読み取りと更新を保護する必要があります。

ポインタ CoS の場合は、名前の変更が禁止されているテンプレートエントリが 1 つ存在します。通常、テンプレートエントリ全体を保護するのがもっとも簡単な方法です。

クラシック CoS では、すべてのテンプレートエントリは、定義エントリで指定された共通の親を持ちます。この親エントリにテンプレートを格納するだけで、親に対するアクセス制御によってテンプレートが保護されます。親の下のほかのエントリにアクセスする場合は、テンプレートエントリを個別に保護する必要があります。

間接 CoS の場合は、アクセスする必要があるユーザエントリを含む、ディレクトリ内の任意のエントリにテンプレートを指定できます。必要に応じて、ディレクトリ全体の CoS 属性に対するアクセスを制御するか、またはテンプレートとして使用される各エントリの CoS 属性のセキュリティを保護する必要があります。

CoS のターゲットエントリの保護

仮想 CoS 属性が生成される、CoS 定義の適用範囲内のすべてのエントリも値の算出に役立ちます。

CoS 属性がターゲットエントリにすでに存在する場合は、デフォルトでは、CoS メカニズムはこの値を上書きしません。この動作を変更する場合は、ターゲットエントリを上書きするように CoS を定義する (178 ページの「実際の属性値の上書き」を参照) か、すべてのターゲットエントリで CoS 属性を保護します。

間接 CoS とクラシック CoS は、ターゲットエントリの指示子属性に依存します。この属性は、使用するテンプレートエントリの DN または RDN を指定します。この属性を保護する場合は、CoS の適用範囲全体でグローバルに保護するか、または各ターゲットエントリで必要に応じて個別に保護する必要があります。

その他の従属関係の保護

最後に、生成されたその他の CoS 属性およびロールに関して仮想 CoS 属性を定義することができます。仮想 CoS 属性を確実に保護するために、これらの従属関係を理解し保護する必要があります。

たとえば、ターゲットエントリの CoS 指示子属性には `nsRole` を指定できます。したがってロール定義も保護する必要があります。詳細は、165 ページの「ロールの安全な使い方」を参照してください。

通常、仮想属性値の算出に関係する属性またはエントリには、読み取りおよび書き込みアクセス制御を設定します。このため、複雑な従属関係は、十分に計画してから設定するか、簡素化する必要があります。その他の仮想属性との従属関係を最小限に抑えると、ディレクトリの性能を向上させ、管理作業を削減することができます。

第 6 章

アクセス制御の管理

iPlanet Directory Server には、ディレクトリへのアクセスを制御する機能があります。この章では、アクセス制御のメカニズム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。

- アクセス制御の原則
- デフォルト ACI
- 手動による ACI の作成
- バインド規則
- Console を使用した ACI の作成
- アクセス制御の使用例
- エントリの ACI 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

アクセス制御メカニズムの機能と柔軟性を活用するには、ディレクトリ導入の計画段階において、全体的なセキュリティポリシーの重要部分として、アクセス制御戦略を決定する必要があります。アクセス制御戦略のヒントについては、『iPlanet Directory Server 導入ガイド』を参照してください。

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。サーバが要求を受け取ると、バインド操作でユーザが提供する認証情報、およびサーバ内で定義された 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。ユーザに与えられるアクセス権のレベルは、そのユーザ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 (構成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、特定のグループまたはロールに属するすべてのユーザ、またはそのディレクトリのすべてのユーザに対して設定できます。また、IP アドレスや DNS 名などの特定位置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` 属性は操作属性です。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関わらず、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、Directory Server 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することができます。

ACI 文は 3 つの主要部分から構成されます。

- ターゲット
- アクセス権
- バインド規則

ACI のアクセス権およびバインド規則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。指定されたアクセス権が与えられるか拒否されるかは、これに付随する規則が true であると判定されるかどうかによって決まります。

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ディレクトリ接尾辞の間にあるすべてのエントリの ACI と、そのエントリ自身の 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、汎用的な規則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。より限定的な規則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規則を置きます。

注 ルート DSE エントリに置かれた 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を評価する場合は、サーバによって、そのエントリ上と、Directory Server に格納された最上位レベルエントリ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、この順番でサーバにより ACI が処理されます。ACI の評価は、特定の Directory Server のすべてのデータベースが対象ですが、複数の Directory Server にわたる評価は行われません。

複数の ACI 間の優先規則は、アクセスを許可する ACI よりもアクセスを拒否する ACI の方が優先するということだけです。アクセスを許可する複数の ACI 間では、和集合の演算が適用され、サーバは、ターゲットエントリに近い ACI から先に処理されたとしても、各 ACI 間の処理に優先規則はありません。

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、ユーザに特定のアクセス権を与えても、どのユーザもディレクトリに書き込めなくなります。特定ユーザ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上に置く必要がある。そのグループが動的である場合は、そのメンバーすべても同じサーバ上にエントリを持つ必要がある。グループが静的である場合は、リモートサーバ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 を持つサーバ上にバインドユーザ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、107 ページの「データベースリンクとアクセス制御の評価」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 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。このキーワードについての詳細は、211 ページの「userattr キーワードの使用」を参照。CoS についての詳細は、第 5 章「高度なエントリの管理」を参照。
- アクセス制御規則の評価は、常にローカルサーバ上で行われる。したがって、ACI キーワードで使用される LDAP URL で、サーバのホスト名やポート番号を指定する必要はない。指定しても、LDAP URL は無視される。LDAP URL については、付録 C「LDAP URLs」を参照。
- プロキシ権限を与える場合と、ユーザに 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ません。

デフォルト ACI

Directory Server をインストールすると、userRoot データベースに格納されたディレクトリ情報に次のデフォルト ACI が適用されます。

- ユーザは、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 属性と nsroledn 属性は変更できません。
- ユーザは、ディレクトリに匿名でアクセスして、検索、比較、および読み込み操作を行うことができる
- 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

ディレクトリに新しいデータベースを作成すると、最初のエントリには必ず前述のデフォルト ACI が置かれます。

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
- ユーザは NetscapeRoot サブツリーに匿名でアクセスして、検索、比較、および読み込み操作を行うことができる
- グループの展開
- 認証されたすべてのユーザには、管理サーバを識別する構成属性に対する検索、比較、および書き込みの権限が与えられる

次の節では、ユーザ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。

手動による 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` ユーティリティ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。次の節では、LDIF 文の作成方法について詳しく説明します。

ヒント LDIF ACI 文は、非常に複雑になることがあります。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、Console よりも LDIF を使用する方が時間を節約できるので、LDIF をお勧めします。

ただし、LDIF ACI 文に慣れるために、Directory Server Console を使用して、ACI を設定後、アクセス制御エディタの「Edit Manually (手動での編集)」ボタンをクリックするということもできます。正しい LDIF 構文が表示されます。また、オペレーティングシステムが対応している場合は、アクセス制御エディタから LDIF をコピーし、作成中の LDIF ファイルに貼り付けることもできます。

ACI の構文

`aci` 属性の構文は次のとおりです。

`aci(`*target*`)(version 3.0;acl "`*name*`";`*permission bind_rules*`;)`

各要素の意味は次のとおりです。

- *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。ターゲット部分は省略可
- `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することができる。ACI 名は省略不可。
- *permission* は、許可または拒否する権限 (読み込み権限や検索権限など) を指定する。
- *bind_rules* は、ユーザがアクセス権を与えられるために必要な資格およびバインドパラメタを指定する。また、バインド規則により、特定のユーザまたはユーザグループへのアクセスを拒否できる

各ターゲットには、複数のアクセス権とバインド規則のペアを設定できます。これによって、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。

target (*permission bind_rule*) (*permission bind_rule*) . . .

1 つの ACI 文中にいくつかの ACR がある場合は、その構文は次のような形式になります。

```
aci(target)(version 3.0;acl "name"; permission bind_rule;
permission bind_rule; ... permission bind_rule;)
```

ACI の例

LDIF ACI の例を次に示します。

```
aci (target="ldap:///uid=bjensen,dc=siroe,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。ターゲットを指定しないと、ACI は `aci` 属性を含むエントリおよびその下位のエントリに適用されます。

ターゲットとして、次のものを指定できます。

- ディレクトリエントリ、またはサブツリー内のすべてのエントリ (194 ページの「ディレクトリエントリのターゲット指定」を参照)
- エントリの属性 (196 ページの「属性のターゲット指定」を参照)
- 指定された LDAP フィルタにマッチするエントリまたは属性のセット (197 ページの「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」を参照)
- 指定された LDAP フィルタにマッチする属性値、または値のペア (198 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)

ターゲットの一般的な構文は次のとおりです。

(*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
- 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、非等号 (!=) 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* はターゲットを示す

expression を囲む引用符 ("") は、省略できません。*expression* の形式は、ユーザが指定する *keyword* によって異なります。

次の表に、各キーワードとそれに対応する式を示します。

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
ターゲット	ldap:///*distinguished_name*	可
`targetattr`	属性	可
`targetfilter`	*LDAP_filter*	可
`targattrfilters`	*LDAP_operation:LDAP_filter*	可

いずれの場合も、エントリに ACI を置くと、そのエントリが最下位のエントリでない限り、下位のエントリすべてにもその ACI が適用されます。たとえば、`ou=accounting,dc=siroe,dc=com` というエントリをターゲットにした場合は、設定するアクセス権は、siroe.com ツリーの accounting 分岐にあるすべてのエントリに適用されます。ただし、accounting ツリーの下にない `uid=sarette,ou=people,dc=siroe,dc=com` エントリには適用されません。

ディレクトリエントリのターゲット指定

ディレクトリエントリ (およびその下位のエントリ) をターゲットとして指定するには、`target` キーワードを使用する必要があります。

`target` キーワードには、次の形式の値を適用することができます。

```
target="ldap:///distinguished_name"
```

これは、アクセス制御規則が適用されるエントリの識別名を示します。たとえば、次のようにします。

```
(target = "ldap:///uid=bjensen,dc=siroe,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 つのバックスラッシュ (\) 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=lfuentes,o=siroe.com
Bolivia\, S.A.")
```

target キーワードで識別名をターゲットとして指定する場合は、ワイルドカードを使用することもできます。ワイルドカードは、任意の文字、文字列、または部分文字列がそのワイルドカードにマッチすることを示します。パターンマッチングの基本は、ワイルドカードで指定された任意の文字列です。

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=siroe,dc=com")

  siroe.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。

- (target="ldap:///uid=*Anderson,dc=siroe,dc=com")

  siroe.com ノードの下にあるエントリで、uid の最後に Anderson が付くすべてのエントリを示します。

- (target="ldap:///*Anderson,dc=siroe,dc=com")

  siroe.com ノードの下にあるエントリで、RDN の最後に Anderson が付くすべてのエントリを示します。

DN の一部にワイルドカードを使用することができます。たとえば、uid=andy*,dc=siroe,dc=com は、これにマッチする uid 属性を持つ siroe.com ツリー全体のディレクトリエントリをターゲットとして指定できます。これは、dc=siroe,dc=com ノードの直下にあるエントリに限りません。このターゲットは、次の両方の式にマッチします。

```
uid=andy,ou=eng,dc=siroe,dc=com
uid=andy,ou=marketing,dc=siroe,dc=com .
```

uid=*,ou=*,dc=siroe,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 の両方の属性を含む、siroe.com ツリーのすべてのエントリとマッチします。したがって、次の式にマッチします。

```
uid=fchen,ou=Engineering,dc=siroe,dc=com
uid=claire,ou=Engineering,ou=people,dc=siroe,dc=com
```

ただし、次の式にはマッチしません。

```
uid=bjensen,dc=siroe,dc=com
ou=Engineering,dc=siroe,dc=com
```

注 識別名の接尾辞部分には、ワイルドカードを使用できません。つまり、ディレクトリの接尾辞が c=US と c=GB である場合に、両方の接尾辞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=siroe,c=*")

また、uid=bjensen,dc=*.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することができます。あるいは、パスワードなど、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。

また、ターゲットが特定の属性と等しい、または等しくないという指定ができます。ここで指定する属性は、スキーマで定義されている必要はありません。スキーマの検査を行わないことにより、作成する ACL が現在のディレクトリの内容に適用されない場合でも、はじめてディレクトリサービスを設定するとき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` 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")
```

ここで、*attribute* は、ターゲットとして指定する属性名です。

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")
```

`targetattr` キーワードで指定された属性は、ACI のターゲットになっているエントリと、その下位にあるすべてのエントリに適用されます。つまり、`uid=bjensen,ou=Marketing,dc=siroe,dc=com` というエントリ上でパスワード属性をターゲットとして指定する場合、ACI の影響を受けるのは、`bjensen` エントリ上のパスワード属性だけです。これは、`bjensen` が最下位のエントリであるためです。

ただし、ツリーの分岐点 `ou=Marketing,dc=siroe,dc=com` をターゲットに指定すると、パスワード属性を含むことができる分岐点より下位のすべてのエントリが ACI によって影響を受けることになります。

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。つまり、

```
aci : (targetattr = "uid")(access_control_rules ;)
```

という ACI を ou=Marketing,dc=siroe,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:///ou=Marketing, dc=siroe,dc=com")
(targetattr="uid") (access_control_rules ;)
```

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、一定の基準にマッチするエントリのグループ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 を使用する必要があります。

targetfilter キーワードの構文は次のとおりです。

```
(targetfilter = "LDAP_filter")
```

ここで、*LDAP_filter* は、標準的な LDAP 検索フィルタです。フィルタの構文についての詳細は、493 ページの「LDAP 検索フィルタ」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 = "(|(employment=contractor)(fulltime<=99))")
```

注 ACI では、国際化値のマッチング規則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=siroe,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=siroe,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、値基準 ACI と呼ばれます。

たとえば、組織内のすべてのユーザに、ユーザ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、これらのユーザが、自身に対して「トップレベルの管理者」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。

値基準 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。

エントリを修正するときに、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 Server Console を使用して値基準の 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

単一のディレクトリエントリをターゲットとして指定することは、アクセス制御メカニズムの設計ポリシーに反するので、容易ではありません。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入力にマッチするバインド規則を作成する。詳細は、211 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。
- `targetattr` および `targetfilter` キーワードを使用する

`targetattr` キーワードを使用すると、ターゲットとして指定するエントリだけに含まれ、その下位のエントリには含まれない属性を指定できます。たとえば、`ou=people,dc=siroe,dc=com` をターゲットとして指定するときに、そのノードの下位に組織単位 (ou) が定義されていない場合は、次の指定を含む ACI を定義できます。

```
targetattr=ou
```

より確実な方法としては、`argetfilter` キーワードを使用して、そのエントリだけに含まれる属性値を明示的に指定する方法があります。たとえば、Directory Server 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";)
```

この ACI は、o=NetscapeRoot 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することができます。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。どのようなときにアクセスを許可または拒否するかのガイドラインについては、『iPlanet Directory Server 導入ガイド』を参照してください。

注	Server Console 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。

Read (読み取り): ユーザ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。

Write (書き込み): ユーザが属性を追加、変更、または削除することによって、エントリを修正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。

Add (追加): ユーザ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。

Delete (削除): ユーザ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。

Search (検索): ユーザ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、Search (検索) および Read (読み込み) 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。

Compare (比較): ユーザ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。

Selfwrite (本人による書き込み): グループに対する個人の DN の追加や削除を、ユーザ自身によって実行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。Selfwrite (本人による書き込み) は、プロキシ認証で使用できます。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（バインドユーザの DN とは異なる）。

Proxy (プロキシ認証): 指定された DN が、ほか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、246 ページの「プロキシ認証を使用した 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、『iPlanet Directory Server 導入ガイド』を参照してください。

All (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がエントリを作成しても、そのユーザ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、ユーザに許可する LDAP 操作のタイプに応じて、そのユーザ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

エントリの属性を修正する場合

- 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 を修正する場合

- 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 -L -h host -s suffix -b "uid=bjensen,dc=siroe,dc=com" \
             objectclass=* mail
```

bkolics というユーザ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 を修正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、バインド規則が 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=siroe,dc=com") (version 3.0; acl
"example";
 allow (read, search, compare) bind_rule ;)
```

バインド規則

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、対応するバインド規則が存在します。

バインド規則は単純なものです。たとえば、バインド規則で、ディレクトリにアクセスするユーザ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規則を設定することもできます。たとえば、バインド規則で、ユーザ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。

バインド規則により、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、バインド規則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、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、複雑なバインド規則を定義できます。詳細は、220 ページの「論理型バインド規則の使用」を参照してください。

サーバでは、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 (資源の制約によって、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。

バインド規則の構文

アクセスが許可されるか拒否されるかは、ACI のバインド規則が true であると判定されるかどうかによって決まります。バインド規則には、次の 2 つのパターンのどちらかが使用されます。

keyword = "*expression*";

keyword != "*expression*";

等号 (=) は、バインド規則を true とするには *keyword* と *expression* がマッチしなければならないことを示し、非等号 (!=) は、バインド規則を true とするには *keyword* と *expression* がマッチ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 バインド規則キーワード

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)
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規則の構文を詳しく説明します。

ユーザアクセスの定義 : userdn キーワード

ユーザアクセスは 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、または anyone、all、self、parent の 1 つです。

`userdn = "ldap:///anyone"`: 匿名によるアクセスを定義

`userdn = "ldap:///all"`: 汎用アクセスを定義

`userdn = "ldap:///self"`: 自己アクセスを定義

`userdn = "ldap:///parent"`: 親エントリへのアクセスを定義

`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バックスラッシュ (\) 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。

匿名アクセスは、アクセス制御エディタを使用して Server Console から定義します。221 ページの「Console を使用した ACI の作成」を参照してください。

汎用アクセス (all キーワード)

バインド規則を使用すると、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、つまり認証されたすべてのユーザに対してアクセス権を許可することができます。これは、匿名アクセスを防ぐ一方、一般的なアクセスを許可します。

匿名アクセスは、アクセス制御エディタを使用して Server Console から定義します。詳細は、221 ページの「Console を使用した ACI の作成」を参照してください。

自己アクセス (self キーワード)

ユーザ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 とマッチするかどうかで、エントリへのアクセス権が許可または拒否されます。

自己アクセスは、アクセス制御エディタを使用して Server Console から設定します。詳細は、221 ページの「Console を使用した ACI の作成」を参照してください。

親アクセス (parent キーワード)

ユーザのバインド 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。

親アクセス制御は、Server Console では設定できません。

LDAP URL

フィルタ付きの URL を使用すると、次のように ACI 内で動的にユーザを指定できます。

```
userdn = "ldap:///<suffix>??sub?(filter)"
```

たとえば、siroe.com ツリーの accounting および engineering の分岐に含まれる、すべてのユーザのターゲット資源に対するアクセスを、次の URL に基づいて動的に許可または拒否できます。

```
userdn = "ldap:///dc=siroe,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に適用されます。

LDAP URL についての詳細は、付録 C「LDAP URLs」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、ユーザのセットを指定することもできます。たとえば、`uid=u*,dc=siroe,dc=com` というユーザ 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 を持つユーザのアクセスだけを許可または拒否できます。

ユーザアクセスは、アクセス制御エディタを使用して Server Console から設定します。詳細は、221 ページの「Console 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=siroe,dc=com";
```

ユーザが、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、バインド規則は 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=siroe,dc=com
uid=tjaz,ou=Accounting,dc=siroe,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=siroe,dc=com
```

LDAP URL の論理 OR を含む userdn キーワード

```
userdn="ldap:///uid=bj,c=siroe.com ||
ldap:///uid=kc,dc=siroe,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、バインド規則は 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=siroe,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、バインド規則は true と判定されます。このバインド規則は、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、バインド規則は true と判定されます。つまり、ユーザが `uid=ssarette, dc=siroe,dc=com` としてバインドし、`uid=ssarette,dc=siroe,dc=com` エントリで操作を試行すれば、バインド規則は true と判定されます。

たとえば、siroe.com ツリー内のすべてのユーザに対して、`userPassword` 属性への書き込みアクセス権を与える場合は、`dc=siroe,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、バインド規則は 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=siroe,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";
```

すべてのユーザに対して、バインド規則は 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、siroe.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=siroe,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、バインド規則は true と判定されます。

たとえば、すべてのユーザ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=siroe,dc=com` ノードに作成します。

```
aci : (version 3.0; acl "parent access";
 allow (write) userdn="ldap:///parent";)
```

ユーザが engineering または sales サブツリーに属していれば、バインド規則は 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、ターゲット資源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、バインド規則は true と判定されます。

注	DN にコンマが含まれる場合、1 つのバックスラッシュ (\) を使用してコンマをエスケープする必要があります。

指定グループは、アクセス制御エディタを使用して Server Console から定義します。詳細は、221 ページの「Console 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=siroe,dc=com";
```

バインド DN が Administrators グループに属していれば、バインド規則は 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=siroe,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=siroe,dc=com";)
```

LDAP URL の論理 OR を含む groupdn キーワード

```
groupdn = "ldap:///cn=Administrators,dc=siroe,dc=com" ||
"ldap:///cn=Mail Administrators,dc=siroe,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、バインド規則は 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、ターゲット資源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、バインド規則は true と判定されます。

注	DN にコンマが含まれる場合、1 つのバックスラッシュ (\) 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

バインド規則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マッチする必要があ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 がユーザエントリの `manager` 属性の DN にマッチする必要があるように指定できます。この場合は、ユーザ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マッチ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に対し、同じ値の `favoriteDrink` 属性を持つほかのユーザであればすべてのエントリの読み込みを許可する 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、どの属性値がマッチする必要があるかを指定するのに使用することができます。

次のタイプを指定できます。

- ユーザ 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 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` の一つです。
- *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規則を含む 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 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 属性値とマッチすれば、バインド規則は true と判定されます。これによって、ユーザのマネージャが社員の属性を修正できるように設定できます。このメカニズムは、ターゲットエントリの manager 属性が、フル 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するフルアクセス権が与えられています。

```
aci : (target="ldap:///dc=siroe,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、バインド規則は 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することができます。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。

ポイントするグループを動的なグループにすることも、グループの DN をデータベース内の任意の接尾辞の下に置くこともできます。ただし、サーバでこのタイプの ACI を評価するには、多くの資源を必要とします。

ターゲットエントリと同じ接尾辞の下にある静的グループを使用する場合は、次の式を使用します。

```
userattr = "ldap:///dc=siroe,dc=com?owner#GROUPDN"
```

この例では、グループエントリは dc=siroe,dc=com という接尾辞の下にあります。サーバ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 = "siroeEmployeeReportsTo#ROLEDN"
```

バインド DN がターゲットエントリの siroeEmployeeReportsTo 属性で指定されたロールに属していれば、バインド規則は 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。

注 この例は、スキーマに siroeEmployeeReportsTo 属性が追加されていて、すべての社員のエントリにこの属性が含まれていることを前提としています。また、この属性値がロールエントリの DN であることも前提です。

スキーマデザインについては、『iPlanet Directory Server 導入ガイド』を参照してください。スキーマへの属性の追加については、336 ページの「属性の作成」を参照してください。

ロールの DN を、データベース内の任意の接尾辞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、サーバがこのタイプの ACI を評価するためには、多くの資源を必要とします。

静的ロール定義を使用するときに、ロールエントリがターゲットエントリと同じ接尾辞の下にある場合は、次の式を使用します。

```
userattr = "ldap:///dc=siroe,dc=com?employeeReportsTo#ROLEDN"
```

この例では、グループエントリは dc=siroe,dc=com という接尾辞の下にあります。サーバ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。

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マッチすれば、バインド規則は 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、バインド規則は 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].attrName#bindType"
```

ユーザ 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 = "parent[inheritance_level].attrName#attrValue"
```

ここで：

- *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* は、USERDN,GROUPDN,LDAPURL の一つです。

次に例を示します。

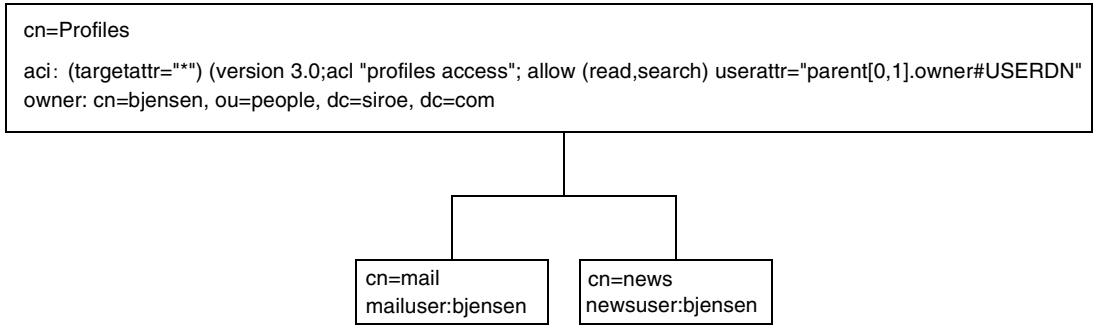
```
userattr = "parent[0,1].manager#USERDN"
```

バインド DN とターゲットエントリの manager 属性がマッチすれば、このバインド規則は true と判定されます。バインド規則が 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 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 が作成するエントリのアクセス権を確認しますが、親エントリのアクセス権は確認されません。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=siroe,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=siroe,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) ことが可能になってしまいます。

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=siroe,dc=com
objectclass: top
...
cn: Trojan Horse
manager: cn=Joe,ou=eng,dc=siroe,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=siroe,dc=com 内にあってバインド DN にマッチする 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=siroe,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 属性がマッチするユーザ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

バインド規則を使用して、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る

IP アドレスに基づくバインド規則を設定するための LDIF 構文は、次のとおりです。

```
ip = "IP_address" or ip != "IP_address"
```

IP アドレスは必ずドット表記で示します。ワイルドカード文字 (*) を使用して、複数のマシンを指定することもできます。たとえば、次のように指定できます。

```
ip = "12.123.1.*";
```

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、バインド規則は 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。

たとえば、12.3.45.* などのワイルドカード IP アドレス を使用して、特定のサブネットワークを指定したり、123.45.6.*+255.255.255.115 などを使用して、サブネットワークマスクを指定したりできます。

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 Server Console から定義します。詳細は、221 ページの「Console 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

バインド規則を使用して、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る

DNS ホスト名に基づくバインド規則設定のための 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が 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、アクセス元のマシンを特定するには、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.siroe.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 = "*.siroe.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、バインド規則は 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、216 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

バインド規則を使用して、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規則を設定できます。アクセス権の評価に使用される時刻は Directory Server 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規則を設定するための 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 (=)、非等号 (!=)、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、サーバ上の時刻が使用されます。

曜日に基づくバインド規則を設定するための 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 文字で示される曜日の略号が使用されます。(sun, mon, tue, wed, thu, fri, sat)

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、バインド規則は `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、バインド規則は `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、バインド規則は `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、バインド規則は `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、バインド規則は `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、バインド規則は `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、バインド規則は `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、バインド規則を設定できます。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す。匿名アクセスを表します。
- Simple : クライアントはユーザ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) 接続を経由して、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「SSL の管理」を参照してください。
- SASL : クライアントは、SASL (Simple Authentication and Security Layer) 接続を経由して、ディレクトリにバインドする必要があります。iPlanet 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規則は、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規則を設定するための 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";
```

バインド規則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、バインド規則は 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、バインド規則は 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、バインド規則は 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、バインド規則は true と判定されます。このほかにも EXTERNAL という SASL メカニズムがサポートされています。

論理型バインド規則の使用

AND、OR、NOT のブール式を使用して細かいアクセス規則を設定すると、複雑なバインド規則を作成できます。ブール型バインド規則は、Server Console では作成できません。LDIF 文を作成する必要があります。

ブール型バインド規則の LDIF 構文は、次のとおりです。

bind_rule [*boolean*] [*bind_rule*] [*boolean*] [*bind_rule*] . . . ;)

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 siroe.com ドメイン内から実行されていれば、次のバインド規則は true と判定されます。

```
(groupdn = "ldap:///cn=administrators,dc=siroe,dc=com" or
groupdn = "ldap:///cn=mail administrators,dc=siroe,dc=com" and
dns = "*.siroe.com";)
```

最後のセミコロン (;) は省略できません。この区切り文字は、最後のバインド規則の後に付ける必要があります。

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規則があるとします。

（*bind_rule_A*） OR （*bind_rule_B*）

（*bind_rule_B*） OR （*bind_rule_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規則 B の前にバインド規則 A が評価され、下の例ではバインド規則 A の前にバインド規則 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。

（*bind_rule_A*） AND NOT （*bind_rule_B*）

ここでは、「左から右へ」の原則は適用されず、バインド規則 A よりも先にバインド規則 B が評価されます。

Console を使用した ACI の作成

Directory Server Console を使用すると、ディレクトリに対するアクセス制御命令を表示、作成、編集、および削除できます。この節では、次の作業の一般的な手順について説明します。

- アクセス制御エディタの表示
- 現在の ACI の表示
- 新しい ACI の作成
- ACI の編集
- ACI の削除

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 Console のステップバイステップな使い方については、227 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 (203 ページの「アクセス権の構文」を参照)

- 値基準 ACI の作成 (198 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 (207 ページの「親アクセス (parent キーワード)」を参照)
- ブール型バインド規則を含む ACI の作成 (220 ページの「論理型バインド規則の使用」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「Edit Manually (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修正した内容をいつでも LDIF で確認できます。

アクセス制御エディタの表示

1. Directory Server Console を起動します。特権ユーザのバインド DN とパスワードを使用してログインします。特権ユーザとは、ディレクトリに対して構成された ACI への書き込みアクセス権を持つディレクトリマネージャなどです。

   手順については、26 ページの「iPlanet Directory Server Console の使用」を参照してください。

2. Directory Server Console で「Directory (ディレクトリ)」タブを選択します。
3. ナビゲーションツリーで、アクセス制御を設定するエントリをマウスの右ボタンでクリックし、「Set Access Permissions (アクセス権の設定)」を選択します。あるいは、エントリを選択して、「オブジェクト」メニューから「Set Access Permissions (アクセス権の設定)」を選択します。

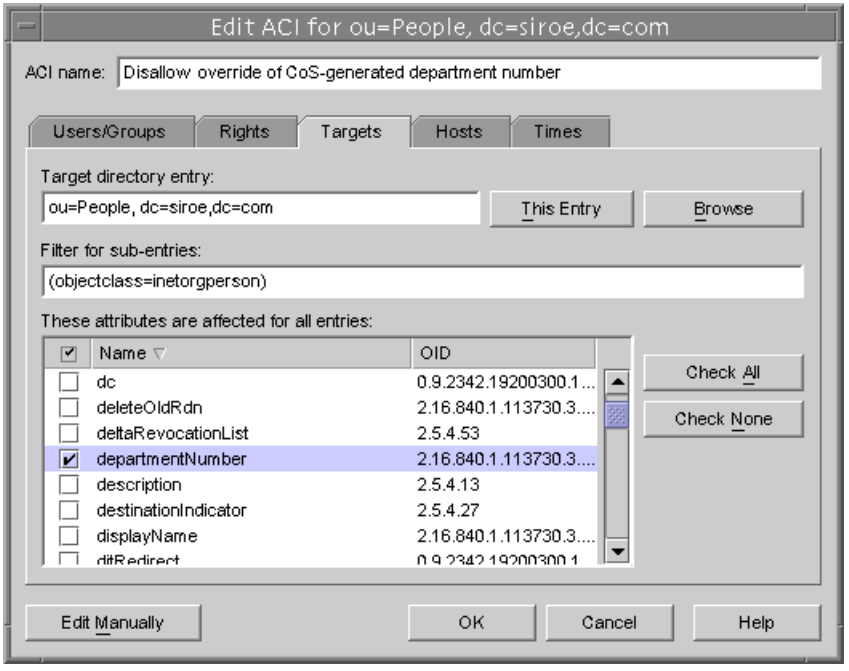
   次の図に「アクセス制御の管理」ダイアログボックスを示します。このダイアログボックスには、選択したエントリで定義されたすべての ACI についての説明が一覧表示され、ACI を修正、削除、および新しく作成することができます。

   「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。

図 6-2 「Access Control Management (アクセス制御の管理)」ダイアログボックス

4. 「New (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、アクセス制御エディタが表示されます。

図 6-3　「Access Control Editor (アクセス制御エディタ)」ダイアログボックス



ダイアログボックス最上部の「ACI name (ACI 名)」には、「Access Control Management (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 の説明が表示されます。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。

「Access Control Editor (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、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タを指定できます。「Access Control (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。

現在の ACI の表示

ディレクトリ内の特定のサブツリーに適用される ACI を表示するには、次の手順を実行します。

1. 「Directory (ディレクトリ)」タブで、サブツリーの一番上のエントリをマウスの右ボタンでクリックし、「Set Access Permissions (アクセス権の設定)」を選択します。

   「Access Control Manager (アクセス制御マネージャ)」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. 選択したエントリに適用されるすべての ACI を表示する場合は、「Show Inherited ACIs (継承された ACI の表示)」チェックボックスを選択します。

新しい ACI の作成

新しい ACI を作成するには、次の手順を実行します。

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、222 ページの「アクセス制御エディタの表示」を参照してください。

   表示画面が 224 ページの図 6-3 と異なる場合は、「Edit Visually (ビジュアル編集)」ボタンをクリックします。

2. 「ACI name (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 **`unnamed ACI`** という名前が付けられます。

3. 「Users/Groups (ユーザ / グループ)」タブで「All Users (すべてのユーザ)」を強調表示してアクセス権を与えるユーザを選択するか、「Add (追加)」ボタンをクリックして追加するユーザのディレクトリを検索します。

   「Add Users and Groups (ユーザ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、「Search (検索)」フィールドに検索文字列を入力してから、「Search (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、「Add (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. 「OK」をクリックして、「Add Users and Groups (ユーザ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「Users/Groups (ユーザ / グループ」タブに表示されます。

4. アクセス制御エディタで「Rights (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. 「Targets (ターゲット)」タブをクリックし、「This Entry (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 の値は修正できますが、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、「Filter for Sub-entries (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. 「Hosts (ホスト)」タブをクリックしてから「Add (追加)」ボタンをクリックして、「Add Host Filter (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. 「Times (時間)」タブをクリックして、アクセス権が与えられる時刻のテーブルを表示します。

   デフォルトでは、いつでもアクセス権が与えられます。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、アクセス時間を修正できます。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、「OK」をクリックします。

   ACI エディタが閉じ、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「Edit Manually (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。この文は修正できますが、加えた修正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. 「Directory (ディレクトリ)」タブで、サブツリーの一番上のエントリをマウスの右ボタンでクリックし、「Set Access Permissions (アクセス権の設定)」を選択します。

   「Access Control Manager (アクセス制御マネージャ)」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. 「Access Control Manager (アクセス制御マネージャ)」ウィンドウで、編集する ACI を選択し、「Edit (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、必要な修正を加えます。

4. ACI の修正が完了したら、「OK」をクリックします。

   ACI エディタが閉じ、ACI マネージャのウィンドウに修正された 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. 「Directory (ディレクトリ)」タブで、サブツリーの一番上のエントリをマウスの右ボタンでクリックし、「Set Access Permissions (アクセス権の設定)」を選択します。

   「Access Control Manager (アクセス制御マネージャ)」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. 「Access Control Manager (アクセス制御マネージャ)」ウィンドウで、削除する ACI を選択します。

3. 「Remove (削除)」をクリックします。

   削除した ACI は、アクセス制御マネージャに表示されなくなります。

アクセス制御の使用例

この節に示す例では、架空の ISP である siroe.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、Console 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。

siroe.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。siroe.com 社の Web ホスティングサービスには、クライアント企業のディレクトリのホスト業務も含まれます。同社は、Company333 および Company999 という 2 つの中規模企業のディレクトリのホスティングと管理の一部を担当しています。また、多数の個人加入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、siroe.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。

- siroe.com 社の社員に、siroe.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 (228 ページの「匿名アクセスの許可」を参照)
- siroe.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 (231 ページの「ユーザエントリへの書き込みアクセス権の許可」を参照)
- siroe.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 (234 ページの「重要なロールに対するアクセスの制限」を参照)
- siroe.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 (235 ページの「接尾辞に対するグループフルアクセスの許可」を参照)
- siroe.com 社のすべての社員に対し、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 (237 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)
- siroe.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 (244 ページの「ユーザ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、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 (239 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 (231 ページの「ユーザ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 (241 ページの「アクセスの拒否」を参照)
- 世界のユーザに対し、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となることがあり、毎日 1 回更新される 228 ページの「匿名アクセスの許可」および 244 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 つの接尾辞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もあります。これは siroe.com 社内のケースであり、「ACI "Anonymous siroe.com"」にその例が示されています。

siroe.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"」に例が示されています。

ACI "Anonymous siroe.com"

siroe.com 社の社員に siroe.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
 Siroe"; allow (read, search, compare) userdn= "ldap:///anyone" and
 dns="*.siroe.com";)
```

この例では、`aci` を `dc=siroe`、`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、Console を使用して次の手順を実行します。

1. 「Directory (ディレクトリ)」タブの左側のナビゲーションツリーで siroe.com ノードをマウスの右ボタンでクリックし、「Set Access Permissions (アクセス権の設定)」を選択してアクセス制御マネージャを表示します。
2. 「New (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. 「Users/Groups (ユーザ / グループ)」タブの ACI 名フィールドに、「Anonymous siroe.com」と入力します。アクセス権が与えられたユーザのリストに、「All Users (すべてのユーザ)」と表示されていることを確認します。
4. 「Rights (権限)」タブで、読み取り、比較、および検索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. 「Targets (ターゲット)」タブで「This Entry (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接尾辞 `dc=siroe,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。「Name (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. 「Host (ホスト)」タブの「Add (追加)」をクリックし、「DNS host filter (DNS ホストフィルタ)」フィールドに「`*.siroe.com`」と入力します。「OK」をクリックしてダイアログボックスを閉じます。
7. 「Access Control Editor (アクセス制御エディタ)」ウィンドウの「OK」ボタンをクリックします。

   「Access Control Manager (アクセス制御マネージャ)」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=siroe, 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、244 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、Console を使用して次の手順を実行します。

1. 「Directory (ディレクトリ)」タブの左側のナビゲーションツリーで siroe.com ノードの下にある Subscribers エントリをマウスの右ボタンでクリックし、「Set Access Permissions (アクセス権の設定)」を選択してアクセス制御マネージャを表示します。
2. 「New (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. 「Users/Groups (ユーザおよびグループ)」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のリストに、「All Users (すべてのユーザ)」と表示されていることを確認します。
4. 「Rights (権限)」タブで、読み取りと検索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. 「Targets (ターゲット)」タブで「This Entry (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接尾辞 `dc=subscribers, dc=siroe,dc=com` が表示されます。
   a. 「filter for subentries (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`, 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。「Check None (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「Name (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. 「OK」をクリックします。

   「Access Control Manager (アクセス制御マネージャ)」ウィンドウの ACI リストに、新しい ACI が追加されます。

ユーザエントリへの書き込みアクセス権の許可

多くの場合、内部ユーザが個人で修正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。siroe.com 社のディレクトリ管理者は、ユーザが修正できる対象を、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 siroe.com"」に例が示されています。

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、siroe.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」に例が示されています。

ACI "Write siroe.com"

注	このアクセス権を設定することによって、ユーザは属性値の削除アクセス権も与えられます。

siroe.com 社の社員が、個人のパスワード、自宅の電話番号、自宅住所を修正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 siroe.com"; allow (write) userdn=
"ldap:///self" and dns="*.siroe.com";)
```

この例では、ACI を `ou=siroe-people,dc=siroe, dc=com` エントリに追加することを仮定しています。

このアクセス権を設定するには、Console を使用して次の手順を実行します。

1. 「Directory (ディレクトリ)」タブの左側のナビゲーションツリーで siroe.com ノードをマウスの右ボタンでクリックし、「Access Control Manager (アクセス権の設定)」を選択してアクセス制御マネージャを表示します。
2. 「New (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. 「Users/Groups (ユーザ / グループ)」タブの ACI 名フィールドに、「Write siroe.com」と入力します。アクセス権が与えられたユーザのリストで、次の手順を実行します。
   a. 「All Users (すべてのユーザ)」を選択して削除し、「Add (追加)」をクリックします。

      「Add Users and Groups (ユーザ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. 「Search area (検索領域)」を「Special Rights (特殊権限)」に設定し、「Search results (検索結果)」リストで「Self (自己)」を選択します。

c. 「Add (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のリストに「Self (自己)」が追加されます。

d. 「OK」をクリックして、「Add Users and Groups (ユーザ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. 「Rights (権限)」タブで、書き込み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. 「Targets (ターゲット)」タブで「This Entry (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接尾辞 `dc=siroe,dc=com` が表示されます。属性テーブルで、`homePhone`、`homePostalAddress`、および `userPassword` attributes 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。「Check None (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「Name (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. 「Hosts (ホスト)」タブの「Add (追加)」ボタンをクリックして、「Add Host Filter (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 host filter (DNS ホストフィルタ)」フィールドに「`*.siroe.com`」と入力します。「OK」をクリックしてダイアログボックスを閉じます。

7. 「Access Control Editor (アクセス制御エディタ)」ウィンドウの「OK」ボタンをクリックします。

   「Access Control Manager (アクセス制御マネージャ)」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、ユーザは属性値の削除アクセス権も与えられます。

siroe.com 社の契約者が個人のパスワードと自宅の電話番号を修正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=siroe, dc=com` エントリに追加することを仮定しています。

住所は siroe.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、Console を使用して次の手順を実行します。

1. 「Directory (ディレクトリ)」タブの左側のナビゲーションツリーで siroe.com ノードの下にある Subscribers エントリをマウスの右ボタンでクリックし、「Set Access Permissions (アクセス権の設定)」を選択してアクセス制御マネージャを表示します。
2. 「New (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. 「Users/Groups (ユーザ / 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のリストで、次の手順を実行します。
   a. 「All Users (すべてのユーザ)」を選択して削除し、「Add (追加)」をクリックします。

      「Add Users and Groups (ユーザ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. 「Search area (検索領域)」を「Special Rights (特殊権限)」に設定し、「Search results (検索結果)」リストで「Self (自己)」を選択します。
   c. 「Add (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のリストに「Self (自己)」が追加されます。
   d. 「OK」をクリックして、「Add Users and Groups (ユーザ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. 「Rights (権限)」タブで、書き込み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. 「Targets (ターゲット)」タブで「This Entry (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接尾辞 `dc=subscribers, dc=siroe,dc=com` が表示されます。
   a. 「filter for subentries (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。「Check None (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「Name (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. ユーザが SSL を使用して認証するように設定する場合は、「Edit Manually (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. 「OK」をクリックします。

   「Access Control Manager (アクセス制御マネージャ)」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することができます。

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` ロールを作ることになるかもしれません。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、156 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、siroe.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」に例を示します。

ACI : Roles

siroe.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=siroe, dc=com")") (version 3.0; acl "Roles"; allow (write)
userdn= "ldap:///self" and dns="*.siroe.com";)
```

この例では、ACI を `ou=siroe-people,dc=siroe, 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、Console を使用して次の手順を実行します。

1. 「Directory (ディレクトリ)」タブの左側のナビゲーションツリーで siroe.com ノードをマウスの右ボタンでクリックし、「Set Access Permissions (アクセス権の設定)」を選択してアクセス制御マネージャを表示します。
2. 「New (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. 「Users/Groups (ユーザ / 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のリストで、次の手順を実行します。

a. 「All Users (すべてのユーザ)」を選択して削除し、「Add (追加)」をクリックします。

「Add Users and Groups (ユーザ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. 「Add Users and Groups (ユーザおよびグループの追加)」ダイアログボックスの「Search area (検索領域)」を「Special Rights (特殊権限)」に設定し、「Search results (検索結果)」リストで「Self (自己)」を選択します。

c. 「Add (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のリストに「Self (自己)」が追加されます。

d. 「OK」をクリックして、「Add Users and Groups (ユーザ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. 「Rights (権限)」タブで、書き込み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. 「Hosts (ホスト)」タブの「Add (追加)」ボタンをクリックして、「Add Host Filter (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 host filter (DNS ホストフィルタ)」フィールドに「`*.siroe.com`」と入力します。「OK」をクリックしてダイアログボックスを閉じます。

6. ロール用に値を基準にしたフィルタを作成するには、「Edit Manually (手動での編集)」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=siroe,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=siroe,dc=com")") (target =
"ldap:///dc=siroe,dc=com") (version 3.0; acl "Roles"; allow
(write) (userdn = "ldap:///self") and (dns="*.siroe.com");)
```

7. 「OK」をクリックします。

「Access Control Manager (アクセス制御マネージャ)」ウィンドウの ACI リストに、新しい ACI が追加されます。

接尾辞に対するグループフルアクセスの許可

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するフル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を設定せずに済みます。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をそのメンバーに与えることができます。

siroe.com 社の Human Resources のグループには、ou=siroe-people 分岐へのフルアクセス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ータベースを更新できます。これについては、「ACI : HR」に例を示します。

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=siroe-people,dc=siroe,dc=com";)
```

この例では、ACI を ou=siroe-people,dc=siroe, 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、Console を使用して次の手順を実行します。

1. 「Directory (ディレクトリ)」タブの左側のナビゲーションツリーで siroe.com ノードの下にある siroe.com-people エントリをマウスの右ボタンでクリックし、「Set Access Permissions (アクセス権の設定)」を選択してアクセス制御マネージャを表示します。
2. 「New (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. 「Users/Groups (ユーザ / 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のリストで、次の手順を実行します。
   a. 「All Users (すべてのユーザ)」を選択して削除し、「Add (追加)」をクリックします。

      「Add Users and Groups (ユーザ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. 「Search area (検索領域)」を「Users and Groups (ユーザおよびグループ)」に設定し、「Search (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. 「Add (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のリストに HR のグループが追加されます。
   d. 「OK」をクリックして、「Add Users and Groups (ユーザ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. 「Rights (権限)」タブで、「Check All (すべてチェック)」をクリックします。

   プロキシ権限以外のすべてのチェックボックスが選択されます。

5. 「OK」をクリックします。

   「Access Control Manager (アクセス制御マネージャ)」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は、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、siroe.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。siroe.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」に例を示します。siroe.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、244 ページの「ユーザ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修正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」に例を示します。

ACI : Create Group

siroe.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=siroe,dc=com)
(targetattr="*")
(targattrfilters="add=objectClass:(objectClass=groupOfNames)")
(version 3.0; acl "Create Group"; allow (read,search,add)
(userdn= "ldap:///uid=*,ou=siroe-people,dc=siroe,dc=com") and
dns="*.siroe.com";)
```

注	この ACI は、書き込みアクセス権を与えません。つまり、エントリは作成できても、修正はできないことを示します。

この例では、ACI を `ou=social committee, dc=siroe,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、Console を使用して次の手順を実行します。

1. 「Directory (ディレクトリ)」タブの左側のナビゲーションツリーで siroe.com ノードの下にある Social Committee エントリをマウスの右ボタンでクリックし、「Set Access Permissions (アクセス権の設定)」を選択してアクセス制御マネージャを表示します。
2. 「New (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. 「Users/Groups (ユーザ / 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のリストで、次の手順を実行します。

   a. 「All Users (すべてのユーザ)」を選択して削除し、「Add (追加)」をクリックします。

      「Add Users and Groups (ユーザ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. 「Search area (検索領域)」を「Special Rights (特殊権限)」に設定し、「Search results (検索結果)」リストで「All Authenticated Users (すべての認証ユーザ)」を選択します。

   c. 「Add (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のリストに「All Authenticated Users (すべての認証ユーザ)」が追加されます。

   d. 「OK」をクリックして、「Add Users and Groups (ユーザ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. 「Rights (権限)」タブで、読み取り、検索、および追加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. 「Targets (ターゲット)」タブで「This Entry (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接尾辞 `ou=social committee, dc=siroe,dc=com` が表示されます。

6. 「Hosts (ホスト)」タブの「Add (追加)」ボタンをクリックして、「Add Host Filter (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 host filter (DNS ホストフィルタ)」フィールドに「`*.siroe.com`」と入力します。「OK」をクリックしてダイアログボックスを閉じます。

7. 値を基準にした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、「Edit Manually (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=siroe,dc=com) (version
   3.0; acl "Create Group"; allow (read,search,add) (userdn=
   "ldap:///all") and (dns="*.siroe.com"); )
   ```

8. 「OK」をクリックします。

   「Access Control Manager (アクセス制御マネージャ)」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

siroe.com 社の社員が ou=Social Comittee branch 分岐の下のグループエントリを編集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=siroe,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=siroe,dc=com エントリに追加しています。

Console 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、siroe.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。siroe.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、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。

- 接続が SSL によって認証されていること
- 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自社の分岐へのフル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=siroe,dc=com")
(targetattr = "*") (version 3.0; acl "Company333"; allow (all)
(roledn=
"ldap:///cn=DirectoryAdmin,ou=Company333,ou=corporate-clients,
dc=siroe,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=siroe,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、Console を使用して次の手順を実行します。

1. 「Directory (ディレクトリ)」タブの左側のナビゲーションツリーで siroe.com ノードの下にある Company333 エントリをマウスの右ボタンでクリックし、「Set Access Permissions (アクセス権の設定)」を選択してアクセス制御マネージャを表示します。
2. 「New (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. 「Users/Groups (ユーザ / 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のリストで、次の手順を実行します。
   a. 「All Users (すべてのユーザ)」を選択して削除し、「Add (追加)」をクリックします。

      「Add Users and Groups (ユーザ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. 「Search area (検索領域)」を「Users and Groups (ユーザおよびグループ)」に設定し、「Search (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. 「Add (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のリストに管理者ロールが追加されます。

   d. 「OK」をクリックして、「Add Users and Groups (ユーザ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. 「Rights (権限)」タブで、「Check All (すべてチェック)」をクリックします。
5. 「Targets (ターゲット)」タブで「This Entry (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接尾辞 `ou=Company333,ou=corporate-clients,dc=siroe,dc=com` が表示されます。

6. 「Hosts (ホスト)」タブの「Add (追加)」ボタンをクリックして、「Add Host Filter (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。「OK」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 siroe.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. 「Times (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、「Edit Manually (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=siroe,dc=com")
(version 3.0; acl "Company333"; allow (all) (roledn=
"ldap:///cn=DirectoryAdmin,ou=Company333,ou=corporate-clients,
dc=siroe,dc=com") and (dayofweek="Mon,Tues,Wed,Thu") and
(timeofday >= "0800" and timeofday <= "1800") and
(ip="255.255.123.234") and (authmethod="ssl"); )
```

9. 「OK」をクリックします。

   「Access Control Manager (アクセス制御マネージャ)」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、siroe.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」に説明があります。

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=siroe,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、Console を使用して次の手順を実行します。

1. 「Directory (ディレクトリ)」タブの左側のナビゲーションツリーで siroe.com ノードの下にある Subscribers エントリをマウスの右ボタンでクリックし、「Set Access Permissions (アクセス権の設定)」を選択してアクセス制御マネージャを表示します。
2. 「New (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. 「Users/Groups (ユーザ / 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のリストで、次の手順を実行します。
   a. 「All Users (すべてのユーザ)」を選択して削除し、「Add (追加)」をクリックします。

      「Add Users and Groups (ユーザ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. 「Add Users and Groups (ユーザおよびグループの追加)」ダイアログボックスの「Search area (検索領域)」を「Special Rights (特殊権限)」に設定し、「Search results (検索結果)」リストで「Self (自己)」を選択します。
   c. 「Add (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のリストに「Self (自己)」が追加されます。
   d. 「OK」をクリックして、「Add Users and Groups (ユーザ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. 「Rights (権限)」タブで、検索と読み取り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. 「Targets (ターゲット)」タブで「This Entry (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接尾辞 ou=subscribers,dc=siroe,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。「Check None (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「Name (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. 「OK」をクリックします。

   「Access Control Manager (アクセス制御マネージャ)」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修正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=siroe,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、Console を使用して次の手順を実行します。

1. 「Directory (ディレクトリ)」タブの左側のナビゲーションツリーで siroe.com ノードの下にある Subscribers エントリをマウスの右ボタンでクリックし、「Set Access Permissions (アクセス権の設定)」を選択してアクセス制御マネージャを表示します。
2. 「New (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. 「Users/Groups (ユーザ / 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のリストで、次の手順を実行します。
   a. 「All Users (すべてのユーザ)」を選択して削除し、「Add (追加)」をクリックします。

      「Add Users and Groups (ユーザ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. 「Add Users and Groups (ユーザおよびグループの追加)」ダイアログボックスの「Search area (検索領域)」を「Special Rights (特殊権限)」に設定し、「Search results (検索結果)」リストで「Self (自己)」を選択します。
   c. 「Add (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のリストに「Self (自己)」が追加されます。
   d. 「OK」をクリックして、「Add Users and Groups (ユーザ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. 「Rights (権限)」タブで、書き込み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. 「Edit Manually (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`** に修正します。

6. 「Targets (ターゲット)」タブで「This Entry (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接尾辞 `ou=subscribers, dc=siroe,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。「Check None (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「Name (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. 「OK」をクリックします。

   「Access Control Manager (アクセス制御マネージャ)」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、アクセス制御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。

次に、`bjensen` というユーザに対して、部署番号、自宅の電話番号、自宅住所、JPEG 写真、および経理部門の全メンバーのマネージャ属性に対する書き込みアクセス権を与える手順を示します。

これらのアクセス権を設定する前に、accounting 分岐点 (`ou=accounting,dc=siroe,dc=com`) を作成する必要があります。組織単位の分岐点は、Directory Server Console の「Directory (ディレクトリ)」タブを使用して作成できます。

ユーザ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。

siroe.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

siroe.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=siroe-people,dc=siroe,dc=com") ;)
```

この例では、ACI を `ou=social committee, dc=siroe,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、Console を使用して次の手順を実行します。

1. 「Directory (ディレクトリ)」タブの左側のナビゲーションツリーで siroe.com ノードの下にある siroe-people エントリをマウスの右ボタンでクリックし、「Set Access Permissions (アクセス権の設定)」を選択してアクセス制御マネージャを表示します。
2. 「New (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. 「Users/Groups (ユーザ / 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のリストで、次の手順を実行します。
   a. 「All Users (すべてのユーザ)」を選択して削除し、「Add (追加)」をクリックします。

      「Add Users and Groups (ユーザ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. 「Add Users and Groups (ユーザおよびグループの追加)」ダイアログボックスの「Search area (検索領域)」を「Special Rights (特殊権限)」に設定し、「Search results (検索結果)」リストで「All Authenticated Users (すべての認証ユーザ)」を選択します。
   c. 「Add (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のリストに「All Authenticated Users (すべての認証ユーザ)」が追加されます。
   d. 「OK」をクリックして、「Add Users and Groups (ユーザ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. 「Rights (権限)」タブで、本人による書き込み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. 「Targets (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=siroe,dc=com」という接尾辞を入力します。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。「Check None (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「Name (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. 「OK」をクリックします。

   「Access Control Manager (アクセス制御マネージャ)」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規則部分で、1 つのバックスラッシュ (\) 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=siroe.com Bolivia\, S.A.,dc=com
objectClass: top
objectClass:organization
aci: (target="ldap:///dc=siroe.com Bolivia\,
 S.A.,dc=com")(targetattr="*") (version 3.0; acl "aci 2"; allow (all)
 groupdn = "ldap:///cn=Directory Administrators,dc=siroe.com
 Bolivia\, S.A.,dc=com";)
```

プロキシ認証を使用した ACI の例

プロキシ認証 (proxy authorization) 方式は、特殊な形式の認証です。自分のユーザ ID を使用してディレクトリにバインドするユーザには、プロキシ認証を使いほかのユーザ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=siroe,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=siroe,dc=com」
- ディレクトリ内に、ou=Accounting,dc=siroe,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=siroe,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=siroe,dc=com")
 (targetattr="*") (version 3.0; acl "allowAll-AcctAdmin"; allow (all)
 userdn="uid=AcctAdministrator,ou=Administrators,dc=siroe,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=siroe,dc=com")
 (targetattr="*") (version 3.0; acl "allowproxy-accountingsoftware";
 allow (proxy)
 userdn="uid=MoneyWizAcctSoftware,ou=Applications,dc=siroe,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 のアクセス権を要求する ldapsearch や ldapmodify などの LDAP コマンドを送信することができます。

注 ディレクトリマネージャの DN をプロキシ DN として使用することはできません。ディレクトリマネージャにプロキシ権限を与えることはできません。また、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。

エントリの ACI の表示

次に示す ldapsearch コマンドを実行することによって、ディレクトリ内の 1 つの接尾辞の下にあるすべての ACI を表示できます。

```
ldapsearch -h host -p port -b baseDN -D rootDN -w rootPassword (aci=*) aci
```

ldapsearch ユーティリティの使い方については、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照してください。

Console のアクセス制御マネージャで、特定のエントリに適用されるすべての ACI を表示できます。

1. Directory Console の「Directory (ディレクトリ)」タブで、ナビゲーションツリーのエントリをマウスの右ボタンでクリックし、「Set Access Permissions (アクセス権の設定)」を選択します。

   アクセス制御マネージャが表示されます。アクセス制御マネージャには、選択したエントリに属する ACI のリストが表示されます。

2. 「継承された ACI の表示」チェックボックスを選択すると、選択されたエントリの上にあるエントリに対して作成され、同様に適用されるすべての ACI が表示されます。

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 の数を最適化することができます。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規則部分、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、LDAP 操作のターゲットとなる資源に対して ACI マクロのマッチングが行われます。マッチした場合、マクロは対象となる資源の DN の値に置き換えられます。続けて、Directory Server は通常どおりに ACI を評価します。

マクロ ACI の例

マクロ ACI の利点と最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。

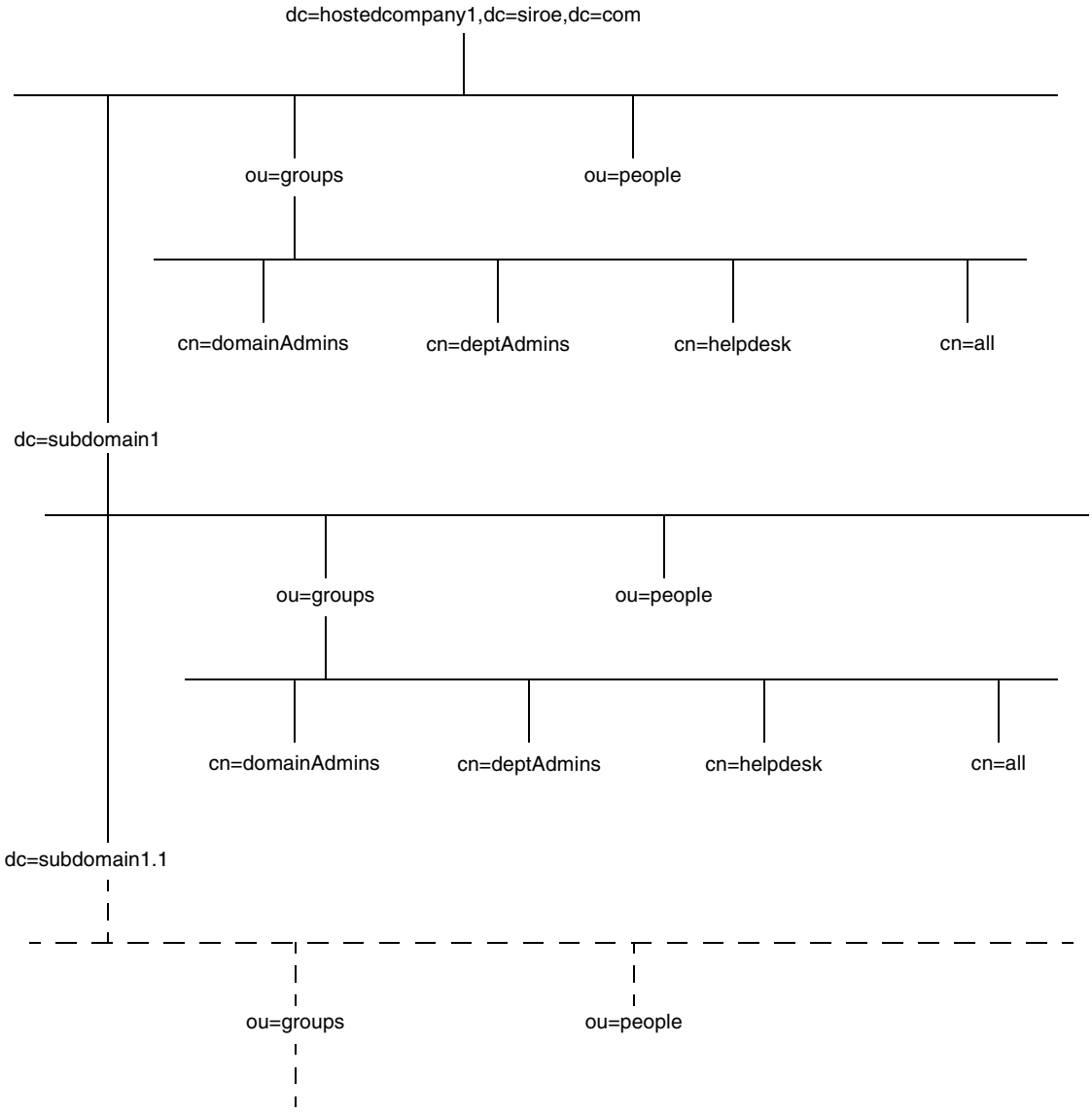
この例では、同じツリー構造のサブドメインが同じパターンで繰り返されています (ou=groups, ou=people)。siroe.com ディレクトリツリーには、接尾辞 dc=hostedCompany2, dc=siroe,dc=com および dc=hostedCompany3,dc=siroe,dc=com が格納されているので、このパターンはツリー内でも繰り返されています。

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=siroe,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=siroe,
dc=com";)
```

この ACI は、dc=hostedCompany1,dc=siroe,dc=com ツリー内のすべてのエントリの DomainAdmins グループに対して、読み取りおよび書き込み権限を与えます。

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=siroe,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=siroe,dc=com";)
```

次の ACI は、dc=subdomain1,dc=hostedCompany1, dc=siroe,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=siroe,dc=com";)
```

次の ACI は、dc=hostedCompany2,dc=siroe,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=siroe,dc=com";)
```

次の ACI は、dc=subdomain1,dc=hostedCompany2, dc=siroe,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=siroe,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=siroe,dc=com ノードに置かれた 1 つの ACI に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,($dn),dc=siroe,dc=com")
 (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,[$dn],dc=siroe,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

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)

- [$dn]
- ($attr.*attrName*)、*attrName* はターゲットエントリの属性

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、ACI のターゲットに対して、まとめてサブジェクトと呼びます。マクロ ACI は、ACI のターゲット部分またはサブジェクト部分で使用できます。

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。

- targetfilter、userdn、roledn、groupdn、userattr で ($dn) を使用する場合は、必ず ($dn) を含むターゲットを定義してください。
- targetfilter、userdn、roledn、groupdn、userattr で [$dn] を使用する場合は、必ず ($dn) を含むターゲットを定義してください。

つまり、マクロを使用するときは、ターゲット定義に必ず ($dn) マクロを含める必要があります。

($dn) マクロと ($attr.attrName) マクロは組み合わせることができます。

($dn) に対するマクロマッチング

($dn) マクロは、LDAP 要求のターゲットである資源のマッチング部分に置き換えられます。たとえば、cn=all, ou=groups,dc=subdomain1,dc=hostedCompany1,dc=siroe,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=siroe,dc=com")
```

この場合、($dn) マクロは「dc=subdomain1, dc=hostedCompany1」とマッチします。

ACI のサブジェクトも ($dn) を使用すると、サブジェクトの展開には、そのターゲットに一致するサブストリングが使用されます。たとえば、次のようにします。

```
aci: (targetattr="*") (target="ldap:///ou=*,($dn),dc=siroe,dc=com")
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,($dn),dc=siroe,dc=com";)
```

この場合、($dn) にマッチするターゲット内の文字列が dc=subdomain1, dc=hostedCompany1 であれば、サブジェクト内でも同じ文字列が使用されます。上の ACI は、次のように展開されます。

```
aci: (targetattr="*")
 (target="ldap:///ou=Groups,dc=subdomain1,dc=hostedCompany1,
 dc=siroe,dc=com") (version 3.0; acl "Domain access"; allow
 (read,search) groupdn="ldap:///cn=DomainAdmins,ou=Groups,
 dc=subdomain1,dc=hostedCompany1,dc=siroe,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。

[$dn] に対するマクロマッチング

[$dn] のマッチングメカニズムは ($dn) のものと少し異なります。ターゲット資源の DN は数回にわたって確認されますが、マッチ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=siroe,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,($dn),dc=siroe,dc=com") (version 3.0;
 acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,[$dn],dc=siroe,dc=com";)
```

この ACI は次の手順で展開されます。

1. ターゲットの ($dn) が dc=subdomain1,dc=hostedCompany1 にマッチします。
2. サブジェクトの [$dn] を dc=subdomain1,dc=hostedCompany1 に置き換えます。

   結果は groupdn="ldap:///cn=DomainAdmins,ou=Groups, dc=subdomain1,dc=hostedCompany1,dc=siroe, 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場合は、マッチングプロセスは中止され、ACI が評価されます。マッチしない場合は、プロセスが続行されます。
3. サブジェクトの [$dn] を dc=hostedCompany1 に置き換えます。

   結果は groupdn="ldap:///cn=DomainAdmins,ou=Groups, dc=hostedCompany1,dc=siroe,dc=com" になります。この場合、バインド DN がそのグループのメンバーでなければ、ACI は評価されません。メンバーであれば、ACI が評価されます。

[$dn] マクロの利点は、ディレクトリツリー内のすべてのサブドメインにドメインレベルの管理者へのアクセスを、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=*, ($dn),dc=siroe,dc=com")
 (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,[$dn],dc=siroe,dc=com";)
```

この ACI は、cn=DomainAdmins,ou=Groups, dc=hostedCompany1,dc=siroe,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=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)"
```

ここで、サーバ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=Heather Blue, ou=People, dc=HostedCompany1, dc=siroe, dc=com
cn: Heather Blue
sn: Blue
ou: Engineering, dc=HostedCompany1, dc=siroe, dc=com
...
```

ACI の roledn 部分を評価するために、サーバはターゲットエントリ内に格納された ou 属性を探し、この属性値を使用してマクロを展開します。したがって、この例における roledn は次のように展開されます。

```
roledn = "ldap:///cn=DomainAdmins,ou=Engineering,dc=HostedCompany1,
dc=siroe,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。

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。

次のような例を想定します。

```
dn: cn=Heather Blue,ou=People,dc=HostedCompany1,dc=siroe,dc=com
cn: Heather Blue
sn: Blue
ou: Engineering, dc=HostedCompany1, dc=siroe, dc=com
ou: People, dc=HostedCompany1,dc=siroe, dc=com
...
```

この場合、Directory Server は、ACI 評価時に、次のように展開された式に対して論理和を実行します。

```
roledn = "ldap:///cn=DomainAdmins,ou=Engineering,dc=HostedCompany1,
dc=siroe,dc=com"

roledn = "ldap:///cn=DomainAdmins,ou=People,dc=HostedCompany1,
dc=siroe,dc=com"
```

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データベースの一部に 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がマスターサーバにレ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。

Console 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. Console 上で「Directory (ディレクトリ)」タブをクリックし、config ノードをマウスの右ボタンでクリックして、「Property (プロパティ)」を選択します。

   この操作を行うと、`cn=config` エントリの属性エディタが表示されます。

2. 属性値の組み合わせをスクロールして、`nsslapd-errorlog-level` 属性を探します。

3. `nsslapd-errorlog-level` 値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 に修正します。エラーログレベルについては『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照してください。

4. 「OK」をクリックして属性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、iPlanet Directory Server 5.1 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以下に示します。

- `userdnattr`
- `groupdnattr`

このため、旧バージョンのサプライヤサーバと Directory Server 5.1 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。

第 7 章

ユーザアカウントの管理

ユーザが Directory Server に接続すると、まずユーザ情報が認証されます。認証が完了すると、認証 (authentication) 中に確立された識別情報に基づいて、アクセス権限 (access rights) と資源制限がユーザに割り当てられます。

この章では、ユーザ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構成、ディレクトリに対するユーザのグループのアクセス拒否、およびバインド DN に応じたユーザのシステムリソースの使用制限などを行います。

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管理
- ユーザとロールの無効化
- バインド DN に基づく資源制限の設定

パスワードポリシーの管理

パスワードポリシー (password policy) を使用して、次の項目を義務付けることによって、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

ディレクトリに対するパスワードポリシーを一度確立すると、アカウントのロックアウトポリシーを設定して、ユーザのパスワードを潜在的な脅威から保護することができます。たとえば、ハッカーがユーザのパスワードを繰り返し入力して推測し、ディレクトリに侵入するのを、アカウントロックアウトによって防ぐことができます。

ここでは、パスワードポリシーとアカウントロックアウトポリシーの構成について、次の項目ごとに説明します。

- 258 ページの「パスワードポリシーの構成」

- 263 ページの「ユーザパスワードの設定」
- 264 ページの「アカウントのロックアウトポリシーの構成」
- 266 ページの「レプリケーション環境でのパスワードポリシーの管理」

パスワードポリシーの構成

構成したパスワードポリシーは、ディレクトリマネージャ (Directory Manager) を除く、ディレクトリ内のすべてのユーザに適用されます。パスワードポリシーは次の情報で構成されます。

パスワードの追加と変更に関する情報 : パスワード情報には、パスワードの構文と詳細な履歴が含まれます。

バインド情報 : バインド情報には、失敗したバインドの追跡情報とパスワードの有効期限の属性が含まれます。

ここでは、次のようなパスワードポリシーを構成するための手順について説明します。

- 258 ページの「Console を使用したパスワードポリシーの構成」
- 259 ページの「コマンド行を使用したパスワードポリシーの構成」

パスワードポリシーを構成したあと、アカウントロックアウトポリシーを構成することをお勧めします。アカウントロックアウトポリシーの構成については、264 ページの「アカウントのロックアウトポリシーの構成」を参照してください。

Console を使用したパスワードポリシーの構成

Directory Server にパスワードポリシーを構成するには、次の手順を実行します。

1. Directory Server Console で「構成」タブを選択し、次に Data ノードを選択します。
2. 右側の区画で「パスワード」タブを選択します。

   このタブには、Directory Server のパスワードポリシーが表示されます。
3. 「リセット後、ユーザにパスワード変更を要求」チェックボックスを選択すると、初回ログオン時にユーザがパスワードを変更しなければならないように指定できます。

   このチェックボックスを選択した場合、ディレクトリマネージャだけが、手順 9 で説明するフィールドを使用してユーザのパスワードをリセットできるようになります。一般の管理者は、パスワードの更新をユーザに義務付けることはできません。
4. 各ユーザが、自分のパスワードを変更できるようにするには、「ユーザによるパスワード変更可」チェックボックスを選択します。

5. ユーザが一定期間パスワードを変更できないようにするには、「再変更を許すまでの日数」テキストボックスに日数を入力します。
6. 各ユーザが使用したパスワードの履歴リストをサーバ上で管理するには、「パスワードの履歴を保持」チェックボックスを選択します。「パスワードの保存回数」テキストボックスで、各ユーザに対して記録しておくパスワードの回数を指定します。
7. ユーザのパスワードを無期限にする場合は、「無期限のパスワード」ラジオボタンを選択します。
8. ユーザにパスワードを定期的に変更させる場合は、「パスワードの有効期限まで」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。
9. 「パスワードの有効期限まで」ラジオボタンを選択した場合、パスワードが期限切れになる何日前にユーザに警告を送信するかを指定する必要があります。「パスワード警告の送信日 (期限切れまでの日数)」テキストボックスに、パスワードの期限切れの何日前に警告を送信するかを示す日数を入力します。
10. ユーザ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側で確認する場合は、「パスワード構文を検査する」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、受け入れ可能なパスワードの最低長を指定します。
11. 「パスワードの暗号化」プルダウンメニューで、パスワードの保存時にサーバで使用する暗号化方式を指定します。

    暗号化方式については、260 ページの表 7-1 の「パスワードポリシーの属性」を参照してください。

    「パスワードの暗号化」メニューには、表に記載されているもの以外の暗号化方式が表示される場合があります。これは、ディレクトリ内で検出された暗号化方式に基づいて、ディレクトリが動的にメニューを作成するためです。
12. 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。

コマンド行を使用したパスワードポリシーの構成

ここでは、サーバのパスワードポリシーを作成するために設定する属性について説明します。`cn=config` エントリ内にあるこれらの属性を変更するには、`ldapmodify` コマンドを使用します。

次の表に、パスワードポリシーを構成するために使用する属性を示します。

表 7-1　パスワードポリシーの属性

属性名	定義
passwordMustChange	オンの場合は、ユーザが最初にディレクトリにログインしたとき、またはディレクトリマネージャがパスワードをリセットしたあとに、パスワードの変更をユーザに要求する。オンの場合は、ユーザが定義したパスワードが無効になっていても、ユーザはパスワードの変更を要求される この属性をオフに設定すると、パスワードは、明白な規約に従うことなく、ディレクトリマネージャによって割り当られたものとなり、ユーザにとってパスワードがわかりにくくなる デフォルトでは、この属性はオフになる
passwordChange	オンの場合は、ユーザが各自のパスワードを変更できる。パスワードを各ユーザに設定させると、ユーザが憶えやすいパスワードを選択する可能性がある ただし、効果的なユーザパスワードを設定するには、多くの管理作業が必要になる。さらに、ユーザにとって意味を持たないパスワードを提供すると、ユーザが見つかりやすい場所にパスワードを書き留めてしまう可能性が生じる デフォルトでは、この属性はオンになっている
passwordExp	オンの場合は、passwordMaxAge 属性で指定された期間が経過すると、ユーザのパスワードが期限切れになる。パスワードの有効期間を設定すると、ディレクトリデータの保護に役立つ。これは、同じパスワードを使用する期間が長くなると、他人に知られる可能性が高くなるためである デフォルトでは、この属性はオフになる
passwordMaxAge	ユーザパスワードが期限切れになるまでの期間を秒数で指定する。この属性を使用する場合は、passwordExp 属性を設定してパスワードの有効期間を有効にする必要がある。 一般的なポリシーでは、パスワードを 30 から 90 日で期限切れにする。デフォルトでは、パスワードの有効期間は 8640000 秒 (100 日間) に設定される

表 7-1 パスワードポリシーの属性 (続き)

属性名	定義
passwordWarning	パスワードの期限切れが近づいたユーザに対して、期限切れの何秒前に警告を送信するかを指定する LDAP クライアント (LDAP client) アプリケーションによっては、警告の送信時に、ユーザに対してパスワードの変更を要求する場合がある。iPlanet Directory Express と Directory Server Gateway の両方でこの機能が実装されている デフォルトでは、パスワードの期限が切れる 86400 秒 (1 日) 前に警告を送信する。ただし、警告メッセージが受信されるまで、パスワードを期限切れにはしない。したがって、ユーザは、passwordMaxAge の指定より長い間 Directory Server にバインドしていない場合でも、パスワードを変更するのに合わせて、ユーザは警告メッセージを受け取る
passwordCheckSyntax	オンの場合は、パスワードの保存前にサーバによってパスワードの構文が検査される パスワードの構文検査によって、パスワードの文字列が、パスワードの最低長の要件以上の長さを持ち、また「安易な」単語を含んでいないことが検査される。安易な単語とは、ユーザのエントリの uid、cn、sn、givenName、ou、mail の属性のいずれかに格納されている値を意味する デフォルトでは、この属性はオフになる
passwordMinLength	パスワードの最小文字数を指定する。短いパスワードほど不正な手段で解読されやすい 最小文字数は、2 ～ 512 文字の範囲で指定できる。一般的に、不正な手段で解読することが難しく、ユーザが記録しておかなくても覚えられる長さは、6 ～ 8 文字である デフォルトでは、この属性は 6 に設定されている

表 7-1　パスワードポリシーの属性 (続き)

属性名	定義
passwordMinAge	ユーザが設定したパスワードに対して、それを変更できない期間を秒数で指定する。この属性を passwordInHistory 属性と合わせて使うことにより、ユーザが古いパスワードを再使用しないように設定できる たとえば、passwordMinAge 属性を 2 日に設定すると、1 つのセッションの間にパスワードを繰り返し変更して古いパスワードをいったんなくし、そのあとで古いパスワードを再使用するという行為を防止できる。 この値は、0 ～ 2147472000 秒 (24,855 日) の間で指定できる。0 を指定すると、ユーザがただちにパスワードを変更できる デフォルトでは、この属性の値は 0 になっている
passwordHistory	パスワードの履歴を、ディレクトリに保存するかどうかを指定する。オンに設定した場合は、passwordInHistory 属性で指定した数のパスワードが、ディレクトリによって履歴内に保存される。ユーザが保存されているパスワードのいずれかの再使用を試みても、拒否される この属性をオフに切り替えた場合でも、一度履歴に保存されたパスワードはすべて残っている。この属性をオンに戻すと、この属性を無効にする前に履歴に記録されたパスワードであっても、再使用はできなくなる デフォルトでは、この属性はオフになる。つまり、ユーザは古いパスワードを再使用できる
passwordInHistory	ディレクトリによって履歴に保存されるパスワードの数を指定する。2 ～ 24 個のパスワードを履歴に保存できる。この機能は、passwordHistory 属性をオンに設定しないと有効にはならない デフォルトでは、この属性は 6 に設定されている

表 7-1　パスワードポリシーの属性 (続き)

属性名	定義
passwordStorageScheme	Directory Server のパスワードを保存するために使用する暗号化のタイプを指定する。Directory Server では、次の暗号化のタイプがサポートされている • SSHA (Salted Secure Hash Algorithm): この方式がもっとも安全であり、推奨されている。デフォルトの暗号化方式 • SHA (Secure Hash Algorithm): 一方向のハッシュアルゴリズムであり、Directory Server 4.x でのデフォルトの暗号化スキーマである • crypt : UNIX システムの一般的な暗号化アルゴリズムで、UNIX パスワードとの互換性を保つために提供されている • clear: この暗号化タイプは、パスワードがプレーンテキストで表示されることを示します。 crypt、SHA、または SSHA 形式を使用して保存されたパスワードは、SASL Digest MD5 を使用したセキュアなログインには使用できないので注意する必要がある 独自にカスタマイズした保存スキーマを使用する場合は、iPlanet プロフェッショナルサービスまでご連絡ください。

ユーザパスワードの設定

userpassword 属性を含み、無効にされていないエントリだけが、ディレクトリへのバインドに使用できます。ユーザパスワードはディレクトリに保存されるので、通常使用する任意の LDAP 操作を使用してディレクトリを更新し、ユーザパスワードを設定またはリセットすることができます。

ディレクトリエントリの作成および変更については、第 2 章「ディレクトリエントリの作成」を参照してください。ユーザアカウントを無効にする方法については、267 ページの「ユーザとロールの無効化」を参照してください。

さらに、Administration Server の「ユーザおよびグループ」領域または Directory Server Console を使用して、ユーザパスワードの設定とリセットができます。「ユーザおよびグループ」領域の使い方については、Administration Server に付属するオンラインヘルプを参照してください。Gateway を使用してディレクトリエントリを作成または変更する方法については、Gateway に付属するオンラインヘルプを参照してください。

アカウントのロックアウトポリシーの構成

ロックアウトポリシーをパスワードポリシーと組み合わせて使用すると、セキュリティが向上します。アカウントのロックアウト機能を使用すると、ハッカーがユーザパスワードを繰り返し推測して、ディレクトリに侵入しようとするのを防止できます。アカウントロックアウトのカウンタは Directory Server 固有です。この機能は、ディレクトリサービスからグローバルにロックアウトするようには設定されていません。つまり、レプリケーション環境でもアカウントロックアウトのカウンタはレプリケートされません。詳細については、266 ページの「レプリケーション環境でのパスワードポリシーの管理」を参照してください。

パスワードポリシーを設定し、ユーザが一定の回数バインドに失敗したら、そのユーザをディレクトリからロックアウトすることができます。

以降では、アカウントロックアウトポリシーの構成について説明します。

- 264 ページの「Console を使用したアカウントロックアウトポリシーの設定」
- 265 ページの「コマンド行を使用したアカウントロックアウトポリシーの構成」

Console を使用したアカウントロックアウトポリシーの設定

Directory Server にアカウントロックアウトポリシーを設定するには、次の手順を実行します。

1. Directory Server Console で「構成」タブを選択し、次に Data ノードを選択します。
2. 右側の区画で「アカウントのロックアウト」タブを選択します。
3. アカウントロックアウトを有効にするには、「アカウントのロックアウトを有効にする」チェックボックスを選択します。
4. 「アカウントのロックアウトまでのログイン失敗回数」テキストボックスに、バインド失敗の最大許容回数を入力します。ここで指定した制限値を超えたユーザはロックアウトされます。
5. 「失敗カウンタのリセットまでの時間 (分)」テキストボックスに、サーバのバインド失敗カウンタが 0 にリセットされるまでのサーバの待ち時間を分単位で入力します。
6. ユーザをディレクトリからロックアウトさせる間隔を設定します。

   管理者がユーザのパスワードをリセットするまで、ユーザをロックアウトさせるように設定する場合は、「無期限にロックアウトする」ラジオボタンを選択します。

   「ロックアウトの時間」ラジオボタンを選択して、テキストボックスに分単位で時間を入力することによって、ロックアウト期間を設定できます。
7. アカウントロックアウトポリシーの変更が終了したら、「保存」をクリックします。

コマンド行を使用したアカウントロックアウトポリシーの構成

この節では、サーバに保存されたパスワードを保護するアカウントロックアウトポリシーの作成に必要な属性について説明します。cn=config エントリ内にあるこれらの属性を変更するには、ldapmodify コマンドを使用します。

次の表に、アカウントロックアウトポリシーを構成するために使用できる属性を示します。

表 7-2 アカウントロックアウトポリシーの属性

属性名	定義
passwordLockout	特定回数のバインド試行の失敗後に、ユーザをディレクトリからロックアウトするかどうかを指定する。ユーザによるバインド試行の失敗の許容回数を設定するには、passwordMaxFailure 属性を使用する この回数に達するとユーザはロックアウトされる。定した期間ユーザをロックアウトすることも、管理者がパスワードをリセットするまでロックアウトすることもできる デフォルトでは、この属性はオフに設定されている。つまり、ユーザはディレクトリからロックアウトされない
passwordMaxFailure	ユーザによるバインド失敗の許容回数を指定する。この回数に達すると、ユーザはディレクトリからロックアウトされる この属性は、passwordLockout 属性がオンに設定されている場合にだけ有効になる デフォルトでは、バインド失敗は 3 回に設定される
passwordLockoutDuration	ユーザがディレクトリからロックアウトされる期間を秒数で指定する。passwordUnlock 属性を使用して、管理者がユーザのパスワードをリセットするまでユーザがロックアウトされるように指定することもできる デフォルトでは、ユーザは 3600 秒間ロックアウトされる

表 7-2　アカウントロックアウトポリシーの属性 (続き)

属性名	定義
passwordResetFailureCount	パスワード失敗カウンタが 0 にリセットされるまでの間隔を秒単位で指定する ユーザアカウントから無効なパスワードが送信されるたびに、パスワード失敗カウンタの値が増分される。passwordLockout 属性がオンに設定されている場合は、カウンタの数値が passwordMaxFailure 属性で指定した失敗の回数に達すると、ユーザはディレクトリからロックアウトされる。アカウントは、passwordLockoutDuration 属性で指定された期間にわたってロックアウトされる。その期間が経過すると、カウンタはゼロ (0) にリセットされる カウンタの目的はハッカーがシステムにアクセスしようとしているかどうかを判断することなので、カウンタはハッカーを検出するのに十分な期間だけ有効になっている必要がある。ただし、カウンタを長期間にわたって無限に増加させると、正規のユーザが不注意でロックアウトされてしまう場合がある デフォルトでは、パスワード失敗カウントのリセット属性は、600 秒に設定される

レプリケーション環境でのパスワードポリシーの管理

レプリケーション環境では、パスワードポリシーとアカウントロックアウトポリシーが次のように適用されます。

- パスワードポリシーが データマスター (data master) に適用される
- アカウントロックアウトポリシーは、レプリケーション (replication) の対象となるすべてのサーバに適用される

ディレクトリ内にあるパスワードポリシー情報の一部はレプリケートされます。次の属性がレプリケートされます。

- passwordMinAge および passwordMaxAge
- passwordExp
- passwordWarning

ただし、設定情報はローカルだけで保持され、レプリケートされません。この情報には、パスワードの構文とパスワードの変更履歴が含まれます。アカウントロックアウトのカウンタもレプリケートされません。

レプリケートされた環境でパスワードポリシーを設定するときは、次の点について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ことを知らせるサーバからの警告は、すべてのレプリカによって発行される。この情報はローカルの各サーバ上に保持される。したがって、ユーザが複数のレプリカに順番にバインドした場合、ユーザは同じ警告を数回受信する。また、ユーザがパスワードを変更した場合は、この情報が複製にフィルタされるまで時間がかかることがある。また、ユーザがパスワードを変更し直後に再バインドした場合は、この情報がレプリカに登録されるまでバインドが失敗することがある
- マスターやレプリカを含むすべてのサーバでバインドの動作を一致させたい場合は、各サーバで同じパスワードポリシーの設定情報を作成する必要がある
- 多重マスター環境では、アカウントロックアウトのカウンタが予測できない動作をする場合がある
- レプリケーションのために作成したエントリ (サーバの識別情報など) には、無期限のパスワードを設定する必要がある。これらの特別なユーザに確実に無期限のパスワードを使用させるには、`passwordExpirationTime` 属性をエントリに追加し、この属性に `20380119031407Z` (有効範囲の上限の値) を指定する

ユーザとロールの無効化

1 つのユーザ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、ユーザはディレクトリにバインドできないため、このユーザの認証操作は失敗します。

ユーザとロールを無効にするには、操作属性 `nsAccountLock` を使用します。エントリに `true` の値を持つ `nsAccountLock` 属性が含まれている場合、サーバはバインドを拒否します。

ユーザとロールの無効化にも、同じ手法を使用します。ただし、ロールを無効にする場合は、ロール (role) のメンバーを無効にしているだけで、ロールエントリ自体は無効にしません。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。

以降では、次の手順について説明します。

- 「Console を使用したユーザとロールの無効化」(268 ページ)
- 「コマンド行を使用したユーザとロールの無効化」(268 ページ)
- 「Console を使用したユーザとロールの有効化」(269 ページ)
- 「コマンド行を使用したユーザとロールの有効化」(270 ページ)

警告 データベース上のルートエントリ (ルートまたはサブ接尾辞に対応するエントリ) は、無効にすることができません。

ルートまたはサブ接尾辞に対応するエントリの作成については、第 2 章「ディレクトリエントリの作成」を参照してください。ルートおよびサブ接尾辞の作成については、第 3 章「ディレクトリデータベースの構成」を参照してください。

Console を使用したユーザとロールの無効化

次に、Console を使用してユーザまたはロールを無効にする手順について説明します。

1. Directory Server Console で、「ディレクトリ」タブを選択します。
2. 左側のナビゲーション区画にあるツリーから、無効にするユーザまたはロールをダブルクリックします。

   「エントリの編集」ダイアログボックスが表示されます。

   「オブジェクト」メニューの「無効」を選択することもできます。
3. 左側の区画で、「アカウント」をクリックします。右側の区画に、無効になっているロールまた有効はユーザが表示されます。ユーザまたはロールを有効にするには、「有効」をクリックします。
4. 「OK」をクリックすると、ダイアログボックスが閉じ、変更が保存されます。

   オブジェクトを無効にすると、「表示」メニューの「アクティブでない状態」を選択することによって、オブジェクトの状態を表示できます。オブジェクトの状態は、Console の右側の区画に、赤い斜線が入ったオブジェクトのアイコンとして表示されます。

コマンド行を使用したユーザとロールの無効化

コマンド行からユーザアカウントを無効にするには、`/usr/sbin/directoryserver account-inactivate` コマンドを使用します。

```
# /usr/sbin/directoryserver account-inactivate
```

次のコードは、コマンドを使用して、Joe Frasier のユーザアカウントを無効にする例を示しています。

```
/usr/sbin/directoryserver account-inactivate -h server.siroe.com \
                -p 389 -D "cn=Directory Manager" -w password \
                 -I "uid=jfrasier,ou=people,dc=siroe,dc=com"
```

この例で使用されているオプションを、次に示します。

表 7-3　例で使用した account-inactivate オプションの説明

オプション	内容
-h	Directory Server のホストマシン名を指定する
-p	Directory Server によって使用されるポートを指定する
-D	ディレクトリマネージャの DN を指定する
-w	ディレクトリマネージャのパスワードを指定する
-I	無効にするユーザアカウントまたはロールの DN を指定する

Console を使用したユーザとロールの有効化

次に、Console を使用してユーザまたはロールを有効にする手順について説明します。

1. Directory Server Console で、「ディレクトリ」タブを選択します。
2. 左側のナビゲーション区画にあるツリーから、有効にするユーザまたはロールをダブルクリックします。

   「エントリの編集」ダイアログボックスが表示されます。

   「オブジェクト」メニューの「有効」を選択することもできます。
3. 左側の区画で、「アカウント」をクリックします。右側の区画に、有効になっているロールまたはユーザが表示されます。ユーザまたはロールを有効にするには、「有効」をクリックします。
4. 対象のユーザまたはロールが別の無効なロールのメンバーである場合は、Console に無効なロールを表示するためのオプションが表示されます。「無効なロールの表示」をクリックすると、ユーザまたはロールが属するロールのリストが表示されます。
5. 処理が終了したら、「OK」をクリックします。

   オブジェクトを再び有効にすると、「表示」メニューの「アクティブでない状態」を選択することによって、オブジェクトの状態を表示できます。Console の右側の区画に、ロールまたはユーザのアイコンが通常の外観で表示されます。無効であることを示すアイコン内の赤い斜線は消えています。

コマンド行を使用したユーザとロールの有効化

コマンド行からユーザアカウントを無効にするには、/usr/sbin/directoryserver account-activate コマンドを使用します。

```
# /usr/sbin/directoryserver account-activate
```

次のコードは、account-activate コマンドを使用して、Joe Frasier のユーザアカウントを有効にする例を示しています。

```
/usr/sbin/directoryserver account-activate -h server.siroe.com \
                -p 389 -D "cn=Directory Manager" -w password \
                -I "uid=jfrasier,ou=people,dc=siroe,dc=com"
```

この例で使用されているオプションを、次に示します。

表 7-4 例で使用した account-activate オプションの説明

オプション	内容
-h	Directory Server のホストマシン名を指定する
-p	Directory Server により使用されるポートを指定する
-D	ディレクトリマネージャの DN を指定する
-w	ディレクトリマネージャのパスワードを指定する
-I	有効にするユーザアカウントまたはロールの DN を指定する

バインド DN に基づく資源制限の設定

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、特別な操作属性値を使用して、検索操作に関するサーバの制限を制御することができます。検索操作に関しては、次の制限を設定できます。

検索制限 : 検索操作で検査できるエントリの数を指定します。

サイズ制限 : サーバ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。

時間制限 : サーバ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。

アイドルタイムアウト : サーバ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。

注 デフォルトでは、ディレクトリマネージャは無制限に資源を利用できます。

クライアントアプリケーションに対して設定した資源制限は、グローバルなサーバ告¨で設定したデフォルトの資源制限より優先されます。

ここでは、次の手順について説明します。

- 「Console を使用した資源制限の設定」(271 ページ)
- 「コマンド行を使用した資源制限の設定」(271 ページ)

Console を使用した資源制限の設定

ここでは、Console を使用してユーザまたはロールの資源制限を設定する手順について説明します。

1. Directory Server Console で、「ディレクトリ」タブを選択します。
2. 左側のナビゲーション区画にあるツリーから、資源制限を設定する対象のユーザまたはロールをダブルクリックします。

   「エントリの編集」ダイアログボックスが表示されます。
3. 左側の区画で、「アカウント」をクリックします。右側の区画の「資源制限」セクションに、設定できる 4 つの制限が表示されます。

   「-1」を指定すると、数の制限がなくなります。
4. 処理が終了したら、「OK」をクリックします。

コマンド行を使用した資源制限の設定

コマンド行を使用して、各エントリに対して次の操作属性を設定できます。`ldapmodify` を使用して、エントリに次の属性を追加します。

属性	内容
`nsLookThroughLimit`	検索操作で検査できるエントリの数を指定する。エントリの数を指定する。この属性値を -1 に設定すると無制限になる
`nsSizeLimit`	サーバ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する。この属性値を -1 に設定すると無制限になる
`nsTimeLimit`	サーバ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する。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する。値の単位は秒数。この属性値を -1 に設定すると無制限になる

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 myserver -p 389 -D "cn=directory manager" -w secretpwd
dn:uid=bjensen,ou=people,dc=siroe,dc=com
changetype: modify
add:nsSizeLimit
nsSizeLimit: 500
```

この ldapmodify 文は、Babs 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

レプリケーションは、使用しているディレクトリサービスを単一サーバの構成から拡張するための重要なメカニズムです。この章では、単一マスターレプリケーション、マルチマスターレプリケーション、およびカスケード型レプリケーションを設定するために、サプライヤサーバおよびコンシューマサーバ上で実行される処理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。

- レプリケーションの概要
- レプリケーションのモデル
- 複雑なレプリケーション構成手順のまとめ
- レプリケーションのための作業の詳細
- 単一マスターレプリケーションの構成
- マルチマスターレプリケーションの構成
- カスケード型レプリケーションの構成
- 更新履歴ログの削除
- コンシューマの初期化
- レプリカの同期の維持
- SSL を介したレプリケーション
- 旧バージョンからのレプリケーション
- レトロ履歴ログのプラグインの使用
- 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

ディレクトリでのレプリケーションの使い方に関する、全体的な情報については、『iPlanet Directory Server 導入ガイド』を参照してください。

レプリケーションの概要

レプリケーションとは、ある Directory Server から別のサーバへディレクトリデータを自動的にコピーするメカニズムです。すべての更新処理 (エントリの追加、変更、削除など) は、レプリケーションを使用して別の Directory Server に自動的にミラーされます。ここでは、次のレプリケーションの概念について説明します。

レプリカ

レプリケーションに関与するデータベースのことを、レプリカと定義します。レプリカには、いくつかの種類があります。

- マスターレプリカ : ディレクトリデータのマスターコピーを含む、読み書き可能データベース。マスターレプリカは、ディレクトリクライアントからの更新要求を処理できる
- コンシューマレプリカ : マスターレプリカに保持されている情報のコピーを含む、読み取り専用データベース。コンシューマレプリカは、ディレクトリクライアントからの検索要求を処理できるが、更新要求はマスターレプリカを照会する
- ハブレプリカ : コンシューマレプリカと同じ読み取り専用データベース。コンシューマレプリカとの相違は、このデータベースに格納された情報が、レプリケートされた情報のコンシューマおよびサプライヤ (ハブ) として動作する Directory Server によって使用される点である

Directory Server は、複数のデータベースを管理するように構成できます。各データベースには、レプリケーションにおいて異なる役割を持たせることができます。たとえば、マスターレプリカに `dc=engineering,dc=siroe,dc=com` 接尾辞、およびコンシューマレプリカに `dc=sales,dc=siroe,dc=com` 接尾辞を格納する 1 つの Directory Server を持つことができます。

サプライヤとコンシューマ

ほかのサーバにそれがレプリケートするマスターレプリカを管理するサーバをサプライヤ (supplier) サーバ またはマスターサーバと呼びます。別のサーバによって更新されるコンシューマレプリカを管理するサーバは、コンシューマサーバと呼ばれます。

ここでは便宜のため、サプライヤまたはコンシューマとしてのサーバの役割について説明します。サーバは、サプライヤとコンシューマの両方に指定できるため、常に正しいとは限りません。これは、次の場合に当てはまります。

- Directory Server がマスターレプリカとコンシューマレプリカの組み合わせを管理する場合
- Directory Server が ハブサプライヤ (hub supplier) として機能する場合。つまり、マスターサーバから更新を受け取り、変更内容をコンシューマサーバにレプリケートする場合。詳細については、284 ページの「カスケード型レプリケーション」を参照してください。
- マルチマスターレプリケーションで、マスターレプリカが 2 つの異なる Directory Server に保持される場合。この場合、各 Directory Server が、もう一方の Directory Server のサプライヤおよびコンシューマとして機能する。詳細については、281 ページの「マルチマスターレプリケーション」を参照してください。

iPlanet Directory Server 5.1 では、レプリケーションは常にサプライヤサーバから開始されます。コンシューマサーバから開始されることはありません。この処理は、サプライヤ主導レプリケーション (supplier-initiated replication) と呼ばれます。この処理により、1 つ以上のコンシューマサーバへデータをプッシュするようにサプライヤサーバを構成できます。

iPlanet Directory Server の旧バージョンでは、コンシューマ主導レプリケーション (consumer-initiated replication) を許可しており、コンシューマサーバがサプライヤサーバからデータをプルするように設定できました。iPlanet Directory Server 5.1 では、コンシューマがサプライヤに更新の送信を促すことができる手順に変更されました。この機能については、318 ページの「レプリカの同期の維持」を参照してください。

更新履歴ログ

すべてのサプライヤサーバは、更新履歴ログ (change log) を保持しています。更新履歴ログとは、マスターレプリカに対して行われた変更を記述しておく記録のことです。サプライヤサーバは、コンシューマサーバに格納されているレプリカに対して、またはマルチマスターレプリケーションの場合はほかのサプライヤに対して、これらの変更をリプレイします。

エントリが変更されると、実行された LDAP 操作を記述する変更レコードが更新履歴ログに記録されます。

更新履歴ログは、通常の LDBM データベースと同じ方法で構成できます。

iPlanet Directory Server 5.0 では、更新履歴ログの形式が変更されました。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、レトロログの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。詳細については、324 ページの「レトロ履歴ログのプラグインの使用」を参照してください。

レプリケーションの単位

iPlanet Directory Server 5.1 では、レプリケーションの最小単位はデータベースです。つまり、データベース全体をレプリケートすることはできますが、データベース内のサブツリーだけをレプリケートすることはできません。そのため、ディレクトリツリーを作成するときは、レプリケーション計画を考慮に入れる必要があります。ディレクトリツリーの設定方法については、『iPlanet Directory Server 導入ガイド』を参照してください。

レプリケーションメカニズムでは、接尾辞とデータベースが 1 対 1 で対応している必要があります。つまり、カスタム分散論理を使用している 2 つ以上のデータベースにまたがって分散されている接尾辞 (またはネームスペース) はレプリケーションできません。このトピックについては、81 ページの「データベースの作成と管理」を参照してください。

レプリケーションの識別情報

2 つのサーバ間でレプリケーションが行われる場合、レプリケーション更新データを送信するためにバインドするときに、コンシューマサーバがサプライヤを認証します。この認証処理を実行するには、サプライヤがコンシューマにバインドする際に使用するエントリがコンシューマサーバに格納されていなければなりません。このエントリをレプリケーションマネージャエントリまたはサプライヤバインド DN と呼びます。

レプリケーションマネージャエントリあるいはそのロールを遂行するために作成したエントリは、次のような条件を満たしている必要があります。

- コンシューマレプリカ (またはハブレプリカ) を管理するすべてのサーバに少なくとも 1 つのエントリが必要
- セキュリティ上の理由から、このエントリをレプリケートされたデータの一部にしないこと

注 このエントリは、コンシューマサーバに定義されたすべてのアクセス制御規則を迂回する、特別なユーザプロファイルとなります。

2 つのサーバ間でレプリケーションを構成する場合は、両方のサーバにレプリケーションマネージャ (サプライヤバインド DN) を識別する必要があります。

- コンシューマサーバまたはハブサプライヤ上に、コンシューマレプリカまたはハブレプリカを構成する場合、レプリケーション更新を実行する権限をもつエントリに対応する 1 以上のサプライヤバインド DN を指定する必要がある
- サプライヤサーバ上にレプリケーションアグリーメントを構成する場合、レプリケーションアグリーメントでレプリケーションマネージャの DN を指定しておく必要がある

注 Directory Server Console では、このレプリケーションマネージャのエントリをサプライヤバインド DN と呼びますが、実際にはサプライヤサーバにそのようなエントリが存在しないため、この呼び方は誤解を招くおそれがあります。これをサプライヤバインド DN と呼ぶのは、コンシューマにレプリケーション更新を提供するためのバインド時にサプライヤを認証できるように、コンシューマに置く必要があるエントリだからです。

レプリケーションマネージャエントリの作成については、289 ページの「サプライヤバインド DN エントリの作成」を参照してください。

レプリケーションアグリーメント

Directory Server では、レプリケーションアグリーメントを使用してレプリケーション構成を定義します。1 つのレプリケーションアグリーメント (replication agreement) には、1 つのサプライヤと 1 つのコンシューマだけの間のレプリケーションについて記述されています。契約はサプライヤサーバ上に設定され、次のものを特定します。

- レプリケートされるデータベース
- データがレプリケートされるコンシューマサーバ
- レプリケーションを実行できる時間帯
- サプライヤサーバがバインドに使用する必要のある DN と資格 (レプリケーションマネージャエントリ、またはサプライヤバインド DN と呼ばれる)
- 接続のためのセキュリティ確保 (SSL、クライアント認証)

Directory Server の旧バージョンとの互換性

iPlanet Directory Server 5.0 および 5.1 のレプリケーションメカニズムは、Directory Server の旧バージョンで使用されていたメカニズムとは異なります。互換性は次のプラグインによって保持されます。

- 旧バージョンのレプリケーションプラグイン
- レトロログプラグイン

古いバージョンのレプリケーションプラグインを使用すると、Directory Server 5.1 は、コンシューマロールで Directory Server 4.x のように動作します。このプラグインを使用した旧バージョンのレプリケーションの実装方法については、322 ページの「旧バージョンからのレプリケーション」を参照してください。

レトロログのプラグインを使用すると、Directory Server 5.1 サプライヤに 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、iPlanet 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。レトロログのプラグインについては、324 ページの「レトロ履歴ログのプラグインの使用」を参照してください。

レプリケーションのモデル

ここでは、もっともよく使用される次のレプリケーションモデルについて説明します。

- 「単一マスターレプリケーション」(279 ページ)
- 「マルチマスターレプリケーション」(281 ページ)
- 「カスケード型レプリケーション」(284 ページ)

これらの基本的なモデルの組み合わせによって、最適なレプリケーション環境を構築することができます。

注 どのレプリケーションモデルを選択した場合でも、スキーマのレプリケーションを考慮するようにしてください。詳細は、『iPlanet Directory Server 導入ガイド』を参照してください。

単一マスターレプリケーション

もっとも単純なレプリケーションモデルでは、ディレクトリデータのマスターコピーは、サプライヤサーバと呼ばれる 1 つのサーバ上の単一のマスターレプリカに保持されます。このサーバによって、コンシューマサーバに格納されたコンシューマレプリカに更新が供給されます。

サプライヤサーバでは、マスターレプリカに対するすべての更新を記録した更新履歴ログを管理します。サプライヤサーバには、レプリケーションアグリーメントも格納されます。

コンシューマサーバには、サプライヤバインド DN に対応するエントリが格納されるので、レプリケーション更新を送信するためにバインドするときにサプライヤを認証できます。

図 8-1　単一マスターレプリケーション

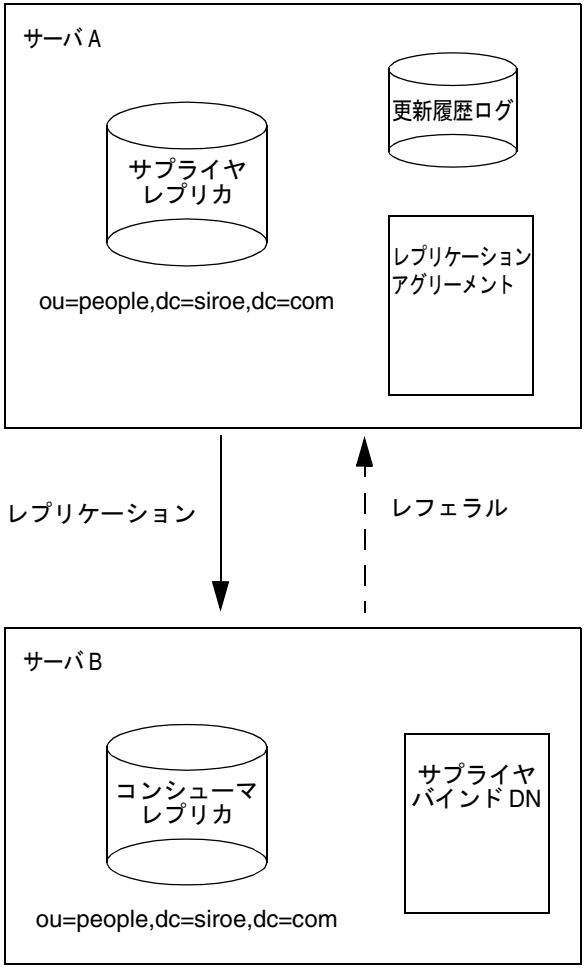


図 8-1 に示した例では、ou=people,dc=siroe,dc=com 接尾辞が、クライアントから多数の検索および更新要求を受け取ります。したがって、負荷を分散するために、サーバ A 上にマスターのあるこの接尾辞は、サーバ B 上にあるコンシューマレプリカにレプリケートされます。

サーバ B は、クライアントからの検索要求を処理できますが、ディレクトリエントリの変更要求は処理できません。サーバ B は、クライアントにサーバ A に対するレフェラルを返すことによって、クライアントから受け取った変更要求を処理します。

注 レプリケーションでは、コンシューマサーバにサプライヤサーバのレフェラル情報が格納されますが、変更要求はクライアントからサプライヤに転送されません。クライアントは、コンシューマから返送されるレフェラルに従う必要があります。

単一マスターレプリケーション環境の構成方法については、296 ページの「単一マスターレプリケーションの構成」を参照してください。

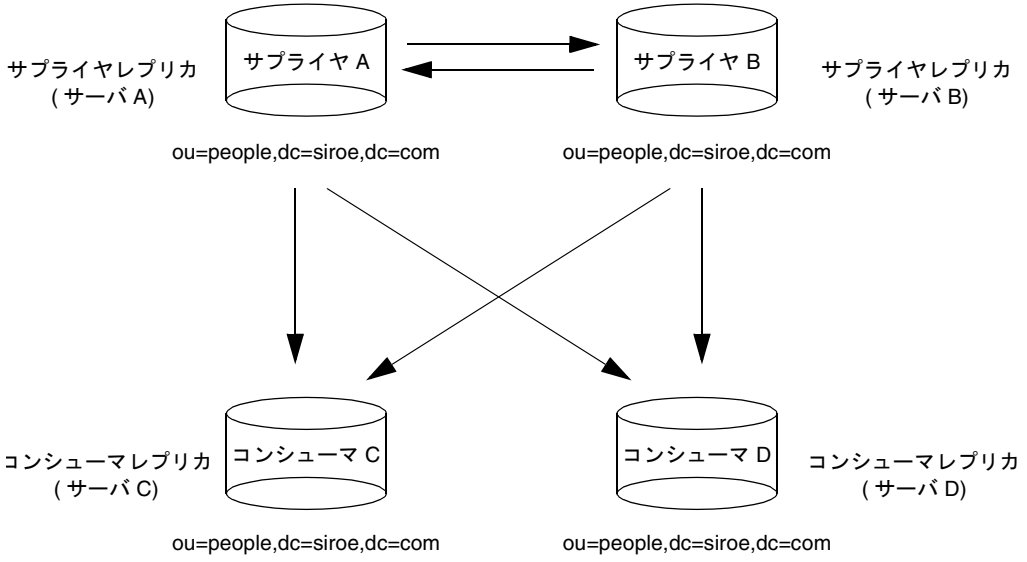
マルチマスターレプリケーション

iPlanet Directory Server 5.1 では、同じ情報が 2 つのサーバ上で、それぞれマスターとして保持されるというより複雑なレプリケーションモデルもサポートします。この情報は、各サーバのマスターレプリカに保持されています。このため、各サーバがレプリカに対する更新履歴ログを、それぞれ維持しています。

このタイプの構成も、任意の数のコンシューマサーバをコピー先として設定できます。2 つのサプライヤから更新を受け取ります。コンシューマでは、両方のサプライヤを定義したレフェラルも保持しています。このようなモデルを、マルチマスター構成と呼びます。

図 8-2 は、マルチマスターレプリケーションのモデルを示したものです。マルチマスターレプリケーションの設定に必要なレプリケーションアグリーメント、更新履歴ログ、およびサプライヤバインド DN については、図 8-3 を参照してください。

図 8-2　マルチマスターレプリケーション



マルチマスター構成には、次の利点があります。

- 1 つのサプライヤにアクセスできなくなった場合でも、自動的に書き込み処理のフェイルオーバが実行される
- 地域分散型環境のローカルサプライヤで更新処理を実行できる

注　レプリケーションの中でも特にマルチマスターレプリケーションは、地域分散型環境で使用される WAN のような接続速度の遅い接続方式ではなく、より高速な接続方式で使用する方が効果的です。

図 8-3　マルチマスターレプリケーションの詳細

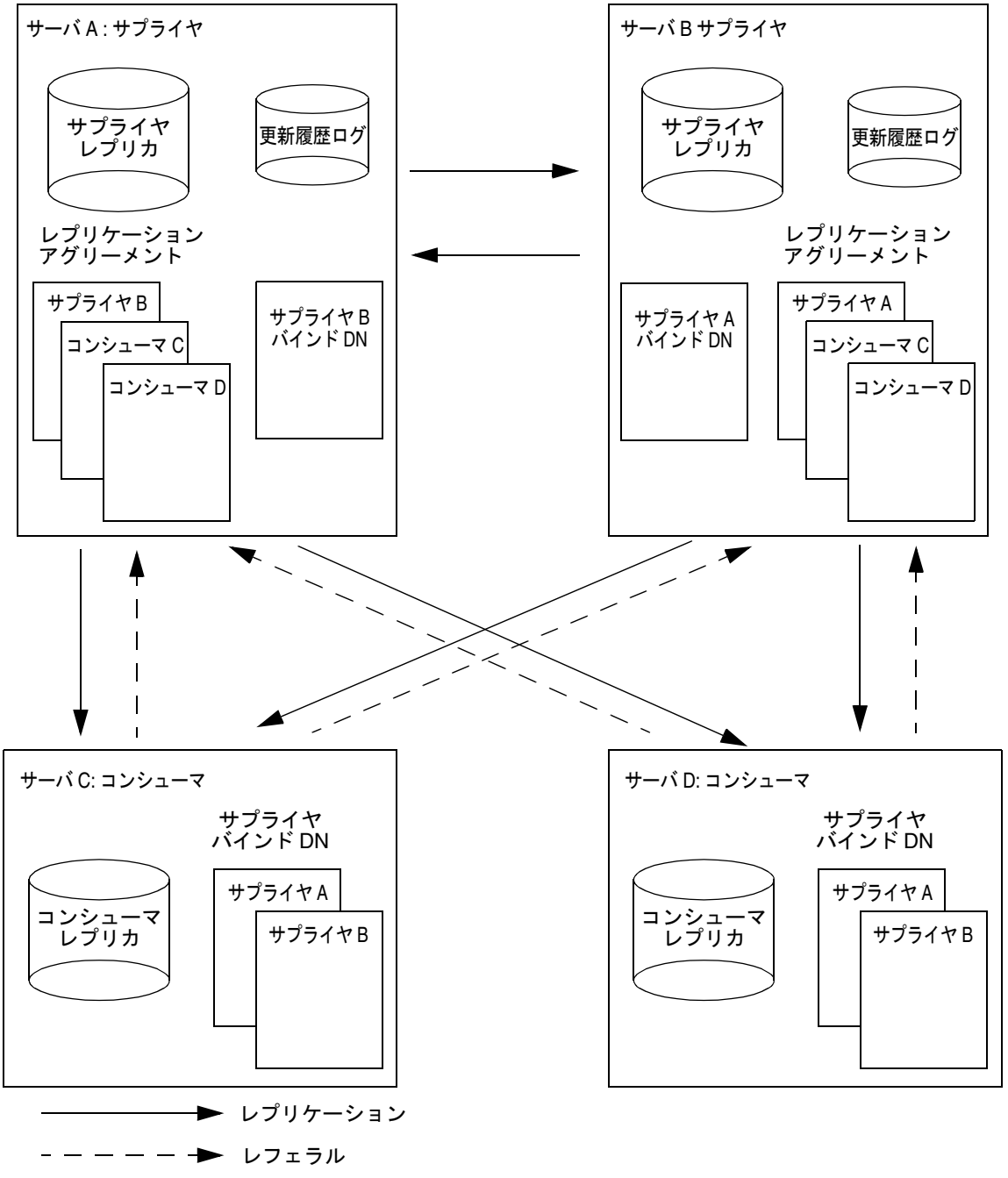


図 8-3 の例では、変更処理で ou=people,dc=siroe,dc=com 接尾辞を常に使用できるようにするために、この接尾辞を 2 つのサプライヤサーバにマスターとして保持します。各サプライヤサーバは、自身の更新履歴ログを管理します。マスターの 1 つがクライアントからの変更要求を処理した場合は、その処理が更新履歴ログに記録され、レプリケーション更新がほかのサプライヤサーバおよびコンシューマに送信されます。

このため、サプライヤサーバでは、コンシューマサーバとのレプリケーションアグリーメントだけでなく互いのレプリケーションアグリーメントを保持する必要があります。各サプライヤサーバには、ほかのマスターがレプリケーション更新を提供するためにバインドできるようにバインド DN も保存されます。

この例では、各コンシューマサーバに、サプライヤバインド DN に対応する 2 つのエントリが格納されるので、レプリケーション更新を送信するためにバインドするときにサプライヤを認証できます。それぞれのコンシューマに、サプライヤバインド DN のエントリを 1 つだけ格納することも可能です。この場合、両方のサプライヤは、同じサプライヤバインド DN を使用してバインドします。

マルチマスターレプリケーションでは、コンシューマがクライアントから変更要求を受け取った場合、両方のサプライヤに対するレフェラルがクライアントに返されます。

注 コンシューマサーバには、サプライヤサーバについてのレフェラル情報が格納されます。コンシューマサーバは、クライアントからの変更要求をサプライヤに転送しません。クライアントは、コンシューマによって返されたレフェラルに従う必要があります。

2 つのサプライヤサーバおよび 2 つのコンシューマサーバからなるマルチマスターレプリケーションの設定方法については、300 ページの「マルチマスターレプリケーションの構成」を参照してください。

カスケード型レプリケーション

カスケード型レプリケーションモデルでは、しばしばハブサプライヤと呼ばれる 1 つのサーバが特定のレプリカに対してコンシューマとサプライヤの両方の役割を受け持ちます。このサーバは、データのマスターコピーを保持するサプライヤサーバから更新を受け取り、次にコンシューマにこの更新を供給します。カスケード型レプリケーションは、次の場合に便利です。

- 過大なトラフィック負荷の均衡を図る必要がある場合。たとえば、サプライヤサーバですべての更新トラフィックを処理しなければならないため、そのことで、コンシューマに対するすべてのレプリケーショントラフィックもサポートするために、サプライヤサーバに過度の負荷がかかってしまうとき。多数のコンシューマに対するレプリケーション更新を実行可能なハブサーバに、レプリケーショントラフィックを任せることにより負荷を軽減できる

- 地域分散型環境でローカルハブサプライヤを使用することにより、接続費用を削減したい場合。
- ディレクトリサービスの性能を向上させたい場合。読み取り操作を実行するすべてのクライアントアプリケーションをコンシューマで実行させ、更新処理を実行するアプリケーションをサプライヤに実行させると、ハブサーバからすべてのインデックス (システムインデックスを除く) を削除できる。これにより、サプライヤとハブサーバ間のレプリケーションの速度が飛躍的に向上する

図 8-4 は、カスケード型レプリケーションの例を示したものです。この例では、もっとも単純なカスケード型レプリケーションモデルを示しています。複数のハブサプライヤおよび多数のコンシューマを含む、さらに複雑なモデルを作成することも可能です。

図 8-4　カスケード型レプリケーション

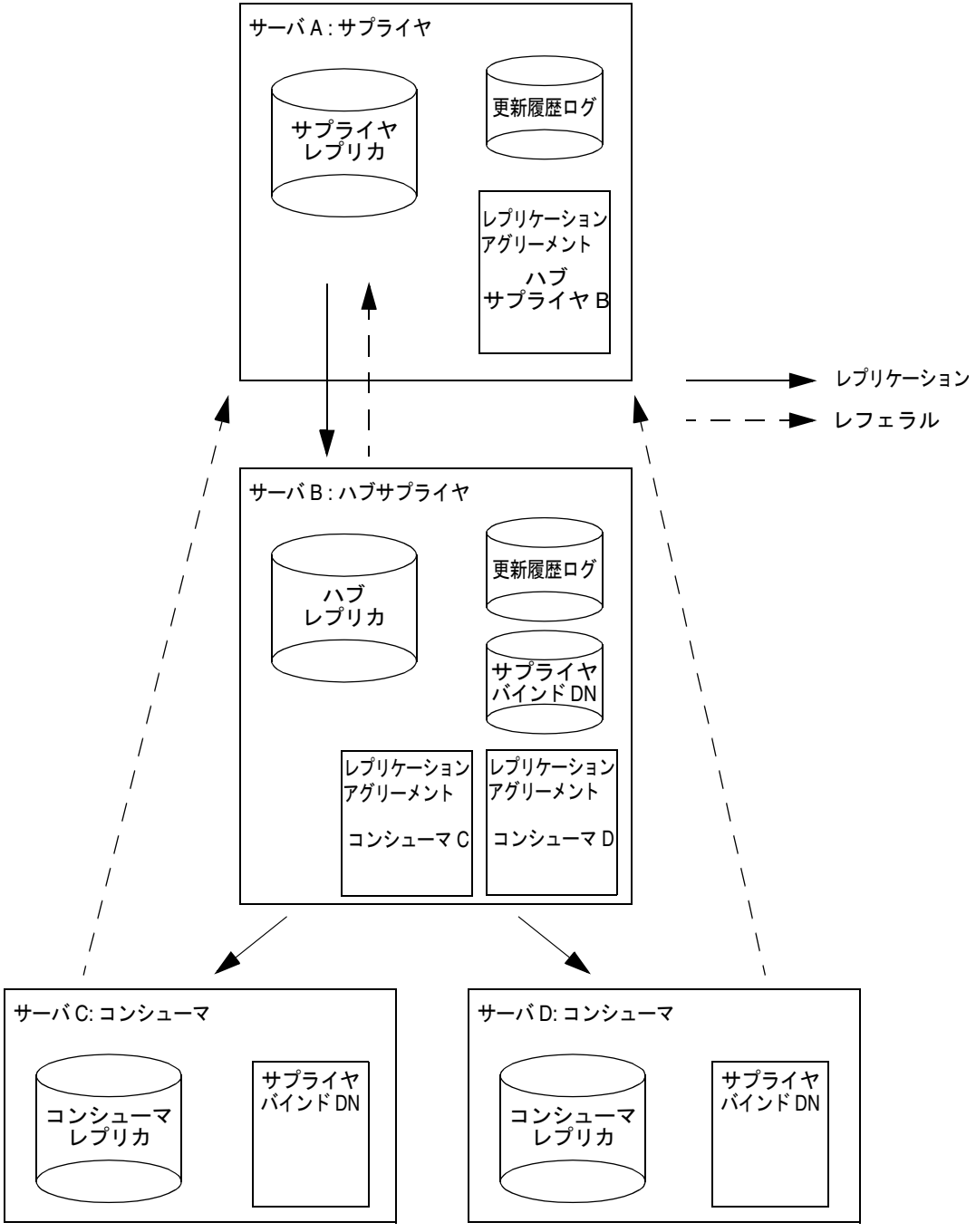


図 8-4 に示した例では、ハブサプライヤを使ってレプリケーション更新処理をサプライヤサーバとハブサプライヤに分散することにより、レプリケーション更新の負荷の均衡を図っています。

サプライヤサーバとハブサプライヤサーバは、いずれも更新履歴ログを管理します。ただし、クライアントからの変更要求を直接処理できるのは、サプライヤサーバだけです。

コンシューマサーバおよびハブサプライヤは、クライアントからの検索要求を処理できますが、変更要求の場合には、サプライヤサーバへのレフェラルをクライアントに返します。図 8-4 は、コンシューマサーバ C および D にサプライヤサーバ A へのレフェラルが設定されていることを示しています。これらは、コンシューマレプリカの構成中にサプライヤサーバを指定したときに作成された自動レフェラルです。

注 コンシューマサーバおよびハブサプライヤには、サプライヤサーバのレフェラル情報が格納されます。これらのサーバは、クライアントからの変更要求をサプライヤに転送しません。クライアントは、コンシューマによって返されたレフェラルに従う必要があります。

カスケード型レプリケーションの構成方法については、306 ページの「カスケード型レプリケーションの構成」を参照してください。

注 マルチマスターレプリケーションとカスケード型レプリケーションを組み合わせて使用することもできます。たとえば、283 ページの図 8-3 のマルチマスターモデルでは、サーバ C とサーバ D を、ハブサプライヤとして構成することにより、任意の数のコンシューマサーバにそれらのサーバからレプリケーションを実行できます。

複雑なレプリケーション構成手順のまとめ

レプリケーションを構成するサーバが多数存在し、レプリケーションが比較的複雑な構成をとる場合、設定作業の効率を高めるためには、次の手順で作業を進めてください。

1. すべてのコンシューマサーバ上で次の操作を行います。
   - レプリカデータベースの作成
   - 少なくとも 1 つのレプリケーションマネージャまたはサプライヤバインド DN エントリの作成
   - コンシューマレプリカに対する設定の指定
2. すべてのハブサプライヤ上で次の操作を行います。

- レプリカデータベースの作成
- レプリケーションマネージャまたはサプライヤ DN エントリの作成
- レプリケーションに対するサプライヤ設定の指定 (更新履歴ログの構成を含む)
- ハブレプリカに対する設定の指定

3. すべてのサプライヤ上で次の操作を行います。
   - レプリカデータベースの作成
   - レプリケーションに対するサプライヤ設定の指定 (更新履歴ログの構成を含む)
   - サプライヤレプリカに対する設定の指定
4. 次のすべてのサプライヤ上にレプリケーションアグリーメントを構成します。
   - マルチマスターセットのサプライヤ間
   - サプライヤとコンシューマの間
   - サプライヤとハブサプライヤの間

   コンシューマサーバおよびハブサプライヤ上のレプリカの初期化は、この時点でも実行できます (任意)。マルチマスターレプリケーションの場合は、1 つのサプライヤレプリカを別のレプリカから初期化します。サプライヤレプリカ間の相互初期化を実行しないでください。
5. すべてのハブサプライヤ上で、ハブサプライヤと対応するコンシューマの間のレプリケーションアグリーメントを構成します。

   コンシューマサーバ上のレプリカの初期化は、この時点でも実行できます (任意)。

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、構成しておくことです。このことによって、レプリケーションアグリーメントの作成が完了すると、コンシューマレプリカをただちに初期化することができるようになります。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。

レプリケーションのための作業の詳細

ここでは、レプリケーションを構成するために必要な処理について説明します。

サプライヤバインド DN エントリの作成

レプリケーションの設定において重要なことは、サプライヤがレプリケーションの更新を行うためにコンシューマサーバにバインドするときに使用するエントリの作成です。このエントリは、レプリケーションマネージャまたはサプライヤバインド DN エントリと呼ばれます。

サプライヤバインド DN エントリが満たす必要がある条件および特徴については、276 ページの「レプリケーションの識別情報」を参照してください。

サプライヤバインド DN エントリを作成するには、次の手順を実行します。

1. レプリケーションアグリーメントでコンシューマとして動作する各サーバに、サプライヤがバインドするために使用する特別なエントリを作成します。

   このエントリをレプリケートされたデータベースの一部にしないでください。ここでは例として、`cn=Replication Manager,cn=config` を使用します。エントリは、必ずレプリケーションアグリーメントで定めた認証方法に必要となる属性を含めて作成してください。

2. 属性と値がペアになった `userPassword` を指定します。

3. パスワードの有効期限ポリシーを有効にしているか、将来有効にする場合は、パスワードの期限切れによってレプリケーションが失敗しないように、このポリシーを忘れずに無効にしてください。`userPassword` 属性でパスワードの有効期限ポリシーを無効にするには、`passwordExpirationTime` 属性に 20380119031407Z という値を追加します。こうするとパスワードの有効期限が切れなくなります。

コンシューマレプリカを構成する場合は、このエントリの DN を使用してサプライヤバインド DN を定義する必要があります。

注　サプライヤバインド DN は、アクセス制御の影響を受けない特権ユーザです。

サプライヤの構成

サプライヤレプリカまたはハブレプリカを保持するサーバでは、サプライヤ設定 (すなわち更新履歴ログパラメタ) を指定する必要があります。

サプライヤを構成するには、次の手順を実行します。

1. Directory Server Console で、「構成」タブをクリックします。

   Directory Server Console の起動については、26 ページの「iPlanet Directory Server Console の使用」を参照してください。

2. 左側のナビゲーションツリーで、Replication ノードを選択します。
3. 右側のナビゲーションウィンドウで、「サプライヤ設定」タブをクリックします。
4. 「更新履歴ログを有効にする」チェックボックスを選択します。

   これにより、ウィンドウの下の区画にある無効にされていたフィールドが有効になります。

5. 「デフォルトの使用」ボタンをクリックするか、または「参照」をクリックしてファイルセレクタを表示し、更新履歴ログを指定します。
6. 更新履歴ログのパラメタ (数および保存期間) を設定します。

   別の値を指定する場合は、無期限のチェックボックスの選択を解除する必要があります。

7. 「保存」をクリックして、Directory Server のサプライヤ設定を保存します。

サプライヤレプリカの構成

各サプライヤレプリカについて、適切なレプリケーション設定を指定する必要があります。

サプライヤレプリカを設定するには、次の手順を実行します。

1. Directory Server Console で、「構成」タブをクリックします。

   Directory Server Console の起動については、26 ページの「iPlanet Directory Server Console の使用」を参照してください。

2. 左側のナビゲーションツリーで、Replication フォルダを展開し、レプリケーション対象のデータベースを選択します。

   「レプリカの設定」タブが右側のナビゲーションウィンドウに表示されます。

3. 「レプリカを有効にする」チェックボックスを選択します。
4. 「レプリカロール」セクションの「単一マスター」または「マルチマスター」ラジオボタンのいずれかを選択します。
5. 「共通設定」セクションで、レプリカ ID (1 ～ 65534 の整数) を指定します。

   各サプライヤレプリカのレプリカ ID は、一意である必要があります。このサーバおよびほかのサーバ上のほかのサプライヤレプリカで使用される ID とは異なる ID を指定するようにしてください。

6. 「共通設定」セクションの「パージ遅延」フィールドにパージ遅延を指定します。

   このオプションは、レプリケートされたエントリに状態情報を格納する期間を示します。パージ前の遅延期間は、レプリケーションのシャットダウンまたはエラーと復元に必要な長さで、同時にエントリに過度に多くのデータを保持させない長さにする必要があります。デフォルト値は 1 週間です。

7. 「保存」をクリックして、データベースに対するレプリケーション設定を保存します。

コンシューマレプリカの構成

各コンシューマレプリカについて、適切なレプリケーション設定を指定する必要があります。

1. Directory Server Console で、「構成」タブをクリックします。

   Directory Server Console の起動については、26 ページの「iPlanet Directory Server Console の使用」を参照してください。

2. 左側のナビゲーションツリーで、Replication フォルダを展開し、レプリカデータベースを選択します。

   「レプリカの設定」タブが右側のナビゲーションウィンドウに表示されます。

3. 「レプリカを有効にする」チェックボックスを選択します。

4. 「レプリカロール」セクションの「専用コンシューマ」ラジオボタンを選択します。

5. 「共通設定」セクションの「パージ遅延」フィールドにパージ遅延を指定します。

   このオプションは、レプリケートされたエントリに状態情報を格納する期間を示します。パージ前の遅延期間は、レプリケーションのシャットダウンまたはエラーと復元に必要な長さで、同時にエントリに過度に多くのデータを保持させない長さにする必要があります。デフォルト値は 1 週間です。

   コンシューマに対してはレプリカ ID を指定する必要がないため、「レプリカ ID」フィールドは無効のままになります (このフィールドは、すべてのコンシューマレプリカで自動的に「65535」に設定されます)。

6. 「設定の更新」セクションで、サプライヤがレプリカにバインドするために使用するサプライヤバインド DN (レプリケーションマネージャ DN) を指定します。

   はじめてレプリカを構成する場合は、「現在のサプライヤ DN」のリストには何も表示されません。1 つのレプリカに対して複数のサプライヤバインド DN を指定できますが、1 つのレプリケーションアグリーメントには 1 つのサプライヤ DN しか指定できません。

   新しいサプライヤバインド DN を指定するには、次の手順を実行します。

a. 対応するフィールドに新しいサプライヤバインド DN を入力します。

   ここで入力する DN は、コンシューマサーバで作成したエントリに対応していなければなりません (例 : `cn=Replication Manager,cn=config`)

b. 「追加」をクリックします。

   サプライヤバインド DN が「現在のサプライヤ DN」のリストに表示されます。

c. リストに含めたいすべてのサプライヤバインド DN で同じ操作を繰り返します。

7. 更新の参照先として、サーバの LDAP URL を指定します。

   はじめてレプリカを構成する場合は、「現在のレフェラルのレプリカデータのマスターを保持しているサーバの URL は表示されません (このレフェラルは、コンシューマサーバによって自動的に作成されます)。

   自動レフェラルでは、クライアントが通常の接続を介してバインドするので、`ldap://`*servername*`:`*port* の形式であると想定しています。SSL を使用してクライアントをサプライヤにバインドする場合は、このフィールドを `ldaps://`*servername*`:`*port* の形式でレフェラルを指定するのに使用できます。ここで `ldaps` の `s` は、セキュリティで保護された接続を意味しています。

   レフェラルの LDAP URL を指定すると、Directory Server では、最初にその URL で修正要求が照会されます。URL を指定しない場合は、現在のレプリカのサプライヤに修正要求が照会されます。

   レフェラルの新しい URL を指定するには、次の手順を実行します。

   a. 対応するフィールドに新しい LDAP URL を入力します。または「構築」をクリックすると、LDAP URL の作成を支援するダイアログボックスが表示されます。

   b. 「追加」をクリックします。

      「現在のレフェラルの URL」リストのすぐ上に、入力した LDAP URL が表示されます。

   c. 同じ操作を繰り返して、リストにレフェラルを追加します。

8. 「保存」をクリックして、レプリケーションの設定を保存します。

ハブレプリカの構成

カスケード型レプリケーション環境で、ハブサプライヤを次のように構成します。

1. Directory Server Console で、「構成」タブをクリックします。

   Directory Server Console の起動については、26 ページの「iPlanet Directory Server Console の使用」を参照してください。

2. 左側のナビゲーションツリーで、Replication フォルダを展開し、レプリケーション対象のデータベースを選択します。

   「レプリカの設定」タブが右側のナビゲーションウィンドウに表示されます。

3. 「レプリカを有効にする」チェックボックスを選択します。

4. 「レプリカロール」セクションの「ハブ」ラジオボタンを選択します。

5. 「共通設定」セクションの「パージ遅延」フィールドにパージ遅延を指定します。

   このオプションは、レプリケートされたエントリに状態情報を格納する期間を示します。パージ前の遅延期間は、レプリケーションのシャットダウンまたはエラーと復元に必要な長さで、同時にエントリに過度に多くのデータを保持させない長さにする必要があります。デフォルト値は 1 週間です。

   ハブサプライヤにレプリカ ID を指定する必要がないため、「レプリカ ID」フィールドは無効のままになります (このフィールドは、コンシューマレプリカと同じように自動的に「65535」に設定されます)。

6. 「設定の更新」セクションで、サプライヤがハブレプリカへのバインドに使用するサプライヤバインド DN (レプリケーションマネージャ DN) を指定します。

   はじめてレプリカを構成する場合は、「現在のサプライヤ DN」のリストには何も表示されません。1 つのレプリカに複数のサプライヤバインド DN を指定できますが、1 つのレプリケーションアグリーメントには 1 つのサプライヤ DN しか指定できません。

   新しいサプライヤバインド DN を指定するには、次の手順を実行します。

   a. 対応するフィールドに新しいサプライヤバインド DN を入力します。

      ここで入力する DN は、コンシューマサーバで作成したエントリに対応していなければなりません (例 : `cn=Replication Manager,cn=config`)

   b. 「追加」をクリックします。

      サプライヤバインド DN が「現在のサプライヤ DN」リストのすぐ上に表示されます。

   c. 同じ操作を繰り返して、サプライヤバインド DN をリストに追加します。

7. 更新の参照先として、サーバの LDAP URL を指定します。

   はじめてレプリカを構成する場合は、「現在のレフェラルの URL」のリストには何も表示されません。デフォルトでは、このリストには、レプリカデータのマスターを保持しているサーバの URL は表示されません (このレフェラルは、ハブサーバによって自動的に作成されます)。

   自動レフェラルでは、クライアントが通常の接続を介してバインドするので、`ldap://`*servername*`:`*port* の形式であることを想定しています。SSL を使用してクライアントをサプライヤにバインドする場合は、このフィールドに `ldaps://`*servername*`:`*port* の形式でレフェラルを指定する必要があります。ここで `ldaps` の `s` は、セキュリティで保護された接続を意味しています。

   レフェラルの LDAP URL を指定すると、Directory Server では、最初にその URL で修正要求が照会されます。URL を指定しない場合は、現在のレプリカのサプライヤで修正要求が照会されます。

   レフェラルの新しい URL を指定するには、次の手順を実行します。

   a. 対応するフィールドに新しい LDAP URL を入力します。または「構築」をクリックすると、LDAP URL の作成を支援するダイアログボックスが表示されます。

   b. 「追加」をクリックします。

      「現在のレフェラルの URL」リストのすぐ上に、入力した LDAP URL が表示されます。

   c. 同じ操作を繰り返して、リストにレフェラルを追加します。

8. 「保存」をクリックして、データベースに対するレプリケーション設定を保存します。

レプリケーションアグリーメントの作成

ここでは、レプリケーションアグリーメントの作成方法を説明します。レプリケーションアグリーメントは、サプライヤサーバ上で各コンシューマサーバやハブのサプライヤに供給されるサプライヤレプリカに対して作成しておく必要があります。

レプリケーションアグリーメントを作成する前に、次の手順を実行しておく必要があります。

- 289 ページの「サプライヤの構成」の説明に従って、サプライヤをサーバに設定する
- 290 ページの「サプライヤレプリカの構成」の説明に従って、サプライヤに対するレプリケーション設定を構成する

- 292 ページの「ハブレプリカの構成」および 291 ページの「コンシューマレプリカの構成」の説明に従って、ハブサプライヤ (存在する場合) およびコンシューマに対するレプリケーションを設定する

レプリケーションアグリーメントを作成するには、次の手順を実行します。

1. Directory Server Console で、「構成」タブをクリックします。
2. ナビゲーションツリーで、レプリケートするデータベースをマウスの右ボタンでクリックし、「新規レプリケーションアグリーメント」を選択します。

   または、データベースを選択して、「オブジェクト」メニューから「新規レプリケーションアグリーメント」を選択することもできます。この操作を行うと、レプリケーションアグリーメントウィザードが開始されます。
3. 「次」をクリックしてステップを進め、レプリケーションウィザードの全ステップを完了します。

   各フィールドへの入力方法については、オンラインヘルプを参照してください。

   ウィザードが完了すると、レプリケーションアグリーメントを表すアイコンがデータベースアイコンの下に表示されます。このレプリケーションアグリーメントアイコンは、レプリケーションアグリーメントが設定されたことを示します。

注

SSL を経由したレプリケーションアグリーメントでは、コンシューマサーバのホスト名は完全指定によるドメイン名 (例えば、server.remote.siroe.com など) として指定する必要があります。エイリアス、IP アドレス、またはドメイン名のローカルの部分だけを指定しないでください。SSL を経由したレプリケーションが許可されなくなります。さらに、「man-in-the-middle」による攻撃からレプリケーションを保護できなくなります。

デフォルトでは、サプライヤはコンシューマサーバの証明書のあるパスを有効にします。サプライヤの信頼する CA (認証局) のルートストアは、SSL を経由したレプリケーションまたはクライアント認証に使用されている CA からの証明書のみである必要があります。SSL を経由したレプリケーションで、「man-in-the-middle」による攻撃から保護するため、コンシューマサーバの証明書に CN 属性にサブジェクト識別名があるか、拡張子が完全指定によるドメイン名と一致することが分っている場合、nsSslServerAuth 構成属性は cncheck 値を持つ必要があります。

単一マスターレプリケーションの構成

ここでは、単一マスターレプリケーションの構成方法の手順を説明します。280 ページの図 8-1 に示すように、サプライヤレプリカを保持するサプライヤサーバ A とコンシューマレプリカを保持するコンシューマサーバ B の間に単一マスターレプリケーションを設定するには、次の手順を実行します。

1. コンシューマサーバ (サプライヤバインド DN とオプションで変更要求のレフェラル) およびコンシューマレプリカを構成します

   この手順については、296 ページの「コンシューマサーバおよびレプリカの構成」を参照してください。

2. サプライヤサーバ (更新履歴ログとレプリカ ID) およびサプライヤレプリカを構成します。

   この手順については、298 ページの「サプライヤサーバおよびレプリカの構成」を参照してください。

3. コンシューマサーバ上のレプリカを初期化します。

   この手順については、300 ページの「単一マスターレプリケーションにおけるレプリカの初期化」を参照してください。

コンシューマサーバおよびレプリカの構成

1. レプリカの対象となるデータベースがない場合は、これを作成します。

   手順については、72 ページの「接尾辞の作成」を参照してください。

2. コンシューマサーバ上に、サプライヤバインド DN に対応するエントリがない場合は、これを作成します。これは、サプライヤがバインドするために使用する特別なエントリです。

   a. Directory Server Console で、「属性名」タブをクリックし、エントリを作成します。ここでは例として、`cn=Replication Manager,cn=config` を使用します。

   b. 属性と値がペアになった `userPassword` を指定します。

   c. パスワードの有効期限ポリシーを有効にしているか、将来有効にする場合は、パスワードの期限切れによりレプリケーションが失敗しないように、このポリシーを忘れずに無効にしてください。`userPassword` 属性でパスワードの有効期限ポリシーを無効にするには、`passwordExpirationTime` 属性に 20380119031407Z という値を追加します。こうするとパスワードの有効期限が切れなくなります。

注 サプライヤバインド DN は、アクセス制御の影響を受けない特権ユーザです。このエントリをレプリケートされたデータベースの一部にしないでください。

3. コンシューマレプリカに必要なレプリケーション設定を指定します。
   a. Directory Server Console で、「構成」タブをクリックします。
   b. ナビゲーションツリーで、Replication フォルダを展開し、レプリカデータベースを選択します。

      「レプリカの設定」タブがウィンドウの右側に表示されます。
   c. 「レプリカを有効にする」チェックボックスを選択します。
   d. 「レプリカロール」セクションの「専用コンシューマ」ラジオボタンを選択します。
   e. 「共通設定」セクションの「パージ遅延」フィールドにパージ遅延を指定します。

      このオプションは、レプリケートされたエントリに状態情報を格納する期間を示します。パージ前の遅延期間は、レプリケーションのシャットダウンまたはエラーと復元に必要な長さで、同時にエントリに過度に多くのデータを保持させない長さにする必要があります。デフォルト値は 1 週間です。

      コンシューマにレプリカ ID を指定する必要がないため、「レプリカ ID」フィールドは無効のままになります (このフィールドは、すべてのコンシューマレプリカで自動的に「65535」に設定されます)。
   f. 「設定の更新」セクションで、サプライヤがレプリカにバインドするために使用するバインド DN (レプリケーションマネージャ DN) を指定します。

      はじめてレプリカを構成する場合は、「現在のサプライヤ DN」のリストには何も表示されません。1 つのレプリカに複数のサプライヤバインド DN を指定できますが、1 つのレプリケーションアグリーメントには 1 つのサプライヤ DN しか指定できません。

      新しいサプライヤバインド DN を指定するには、次の手順を実行します。

      - 対応するフィールドに新しいサプライヤバインド DN を入力します。ここで入力する DN は、手順 2 で作成したエントリに対応していなければなりません (例 : `cn=Replication Manager,cn=config`)
      - 「追加」をクリックします。サプライヤバインド DN が「現在のサプライヤ DN」リストのすぐ上に表示されます。
      - 同じ操作を繰り返して、サプライヤバインド DN をリストに追加します。

g. 更新の参照先として、サーバの LDAP URL を指定します (任意)。

はじめてレプリカを構成する場合は、「現在のレフェラルの URL」のリストには何も表示されません。デフォルトでは、このリストには、レプリカデータのマスターを保持しているサーバの URL は表示されません (このレフェラルは、サーバによって自動的に作成されます)。

自動レフェラルでは、クライアントが通常の接続を介してバインドするので、`ldap://`*servername*`:`*port* の形式であることを想定しています。SSL を使用してクライアントをサプライヤにバインドする場合は、このフィールドに `ldaps://`*servername*`:`*port* の形式でレフェラルを指定する必要があります。ここで `ldaps` の `s` は、セキュリティで保護された接続を意味しています。

レフェラルに LDAP URL を指定すると、Directory Server では、最初にその URL で更新要求が照会されます。指定しない場合は、現在のレプリカのサプライヤで更新が照会されます。

レフェラルの新しい URL を指定するには、次の手順を実行します。

- 対応するフィールドに新しい LDAP URL を入力します。または「構築」をクリックすると、LDAP URL の作成を支援するダイアログボックスが表示されます。
- 「追加」をクリックします。「現在のレフェラルの URL」リストのすぐ上に、入力した LDAP URL が表示されます。
- 同じ操作を繰り返して、リストにレフェラルを追加します。

4. 「保存」をクリックして、レプリケーションの設定を保存します。

サプライヤサーバおよびレプリカの構成

1. サーバに対するサプライヤ設定を指定します。
   a. Directory Server Console で、「構成」タブをクリックします。
   b. ナビゲーションツリーで、Replication ノードを選択します。
   c. ウィンドウの右側にある「サプライヤ設定」タブの「更新履歴ログを有効にする」チェックボックスを選択します。

   これにより、ウィンドウの下の区画の無効にされていたフィールドが有効になります。
   d. 「デフォルトの使用」ボタンをクリックするか、または「参照」ボタンをクリックしてファイルセレクタを表示し、更新履歴ログを指定します。
   e. 更新履歴ログのパラメタ (数および保存期間) を設定します。

   別の値を指定する場合は、無期限のチェックボックスの選択を解除する必要があります。
   f. 「保存」をクリックして、サプライヤ設定を保存します。

2. サプライヤレプリカに必要なレプリケーション設定を指定します。

   a. 「構成」タブのナビゲーションツリーで、Replication ノードを展開し、レプリケーション対象のデータベースを選択します。

      「レプリカの設定」タブがウィンドウの右側に表示されます。

   b. 「レプリカを有効にする」チェックボックスを選択します。

   c. 「レプリカロール」セクションの「単一マスター」ラジオボタンを選択します。

   d. 「共通設定」セクションで、レプリカ ID (1 ～ 65534 の整数) を指定します。

      各サプライヤレプリカのレプリカ ID は、一意である必要があります。このサーバおよびほかのサーバ上のほかのサプライヤレプリカで使用される ID とは異なる ID を指定するようにしてください。

   e. 「共通設定」セクションの「パージ遅延」フィールドにパージ遅延を指定します。

      このオプションは、レプリケートされたエントリに状態情報を格納する期間を示します。パージ前の遅延期間は、レプリケーションのシャットダウンまたはエラーと復元に必要な長さで、同時にエントリに過度に多くのデータを保持させない長さにする必要があります。デフォルト値は 1 週間です。

   f. 「保存」をクリックして、データベースに対するレプリケーション設定を保存します。

3. このサプライヤとコンシューマ間のレプリケーションアグリーメントを作成します。

   a. 「構成」タブのナビゲーションツリーで、レプリケートするデータベースをマウスの右ボタンでクリックし、「新規レプリケーションアグリーメント」を選択します。

      または、データベースを選択して、「オブジェクト」メニューから「新規レプリケーションアグリーメント」を選択することもできます。この操作を行うと、レプリケーションアグリーメントウィザードが開始されます。

   b. 「次」をクリックしてステップを進め、レプリケーションウィザードの全ステップを完了します。

      各フィールドへの入力方法については、オンラインヘルプを参照してください。

   c. ウィザードが完了すると、レプリケーションアグリーメントが設定されます。

単一マスターレプリケーションにおけるレプリカの初期化

コンシューマレプリカは、レプリケーションアグリーメントウィザードから初期化することができ、その後ならいつでも可能です。レプリカの初期化については、315 ページの「コンシューマの初期化」を参照してください。

マルチマスターレプリケーションの構成

ここでは、マルチマスターレプリケーションの構成方法について説明します。マルチマスター構成では、2 つのサプライヤが更新を受け取り、互いに同期をとりすべてのコンシューマを更新することができます。コンシューマは、両方のマスターに更新要求を照会できます。ここでは、マルチマスターレプリケーションの設定手順について説明します。

282 ページの図 8-2 のように、それぞれがサプライヤレプリカを保持するサーバ A とサーバ B という 2 つのサプライヤ、およびそれぞれがコンシューマレプリカを保持するサーバ C とサーバ D という 2 つのコンシューマといった、マルチマスターレプリケーションを設定するには、次の処理を実行します。

1. コンシューマサーバ (サプライヤバインド DN およびオプションで変更要求のレフェラル) およびコンシューマレプリカを構成します

   この手順については、300 ページの「コンシューマサーバおよびレプリカの構成」を参照してください。

2. サプライヤサーバ (更新履歴ログとレプリカ ID) およびサプライヤレプリカを構成します。

   この手順については、303 ページの「サプライヤサーバおよびレプリカの構成」を参照してください。

3. コンシューマサーバ上のコンシューマレプリカを初期化します。

   この手順については、306 ページの「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」を参照してください。

コンシューマサーバおよびレプリカの構成

各コンシューマサーバに対して、次の手順を実行します。

1. レプリカの対象となるデータベースがない場合は、これを作成します。

   手順については、72 ページの「接尾辞の作成」を参照してください。

2. サプライヤバインド DN に対応するエントリがない場合は、これを作成します。これは、サプライヤがバインドするために使用する特別なエントリです。

   a. Directory Server Console で、「属性名」タブをクリックし、エントリを作成します。ここでは例として、`cn=Replication Manager,cn=config` を使用します。

   b. 属性と値がペアになった `userPassword` を指定します。

   c. パスワードの有効期限ポリシーを有効にしているか、将来有効にする場合は、パスワードの期限切れによってレプリケーションが失敗しないように、このポリシーを忘れずに無効にしてください。`userPassword` 属性でパスワードの有効期限ポリシーを無効にするには、`passwordExpirationTime` 属性に 20380119031407Z という値を追加します。こうするとパスワードの有効期限が切れなくなります。

注 サプライヤバインド DN は、アクセス制御の影響を受けない特権ユーザです。このエントリをレプリケートされたデータベースの一部にしないでください。

---

3. コンシューマレプリカに必要なレプリケーション設定を指定します。

   a. Directory Server Console で、「構成」タブをクリックします。

   b. ナビゲーションツリーで、Replication フォルダを展開し、レプリカデータベースを選択します。

      「レプリカの設定」タブがウィンドウの右側に表示されます。

   c. 「レプリカを有効にする」チェックボックスを選択します。

   d. 「レプリカロール」セクションの「専用コンシューマ」ラジオボタンを選択します。

   e. 「共通設定」セクションの「パージ遅延」フィールドにパージ遅延を指定します。

      このオプションは、レプリケートされたエントリに状態情報を格納する期間を示します。パージ前の遅延期間は、レプリケーションのシャットダウンまたはエラーと復元に必要な長さで、同時にエントリに過度に多くのデータを保持させない長さにする必要があります。デフォルト値は 1 週間です。

      コンシューマにレプリカ ID を指定する必要がないため、「レプリカ ID」フィールドは無効のままになります (このフィールドは、すべてのコンシューマレプリカで自動的に「65535」に設定されます)。

f. 「設定の更新」セクションで、サプライヤがレプリカにバインドするために使用するバインド DN (レプリケーションマネージャ DN) を指定します。

はじめてレプリカを構成する場合は、「現在のサプライヤ DN」のリストには何も表示されません。1 つのレプリカに複数のサプライヤバインド DN を指定できますが、1 つのレプリケーションアグリーメントには 1 つのサプライヤ DN しか指定できません。

新しいサプライヤバインド DN を指定するには、次の手順を実行します。

- 対応するフィールドに新しいサプライヤバインド DN を入力します。ここで入力する DN は、手順 2 で作成したエントリに対応していなければなりません (例 : `cn=Replication Manager,cn=config`)
- 「追加」をクリックします。サプライヤバインド DN が「現在のサプライヤ DN」リストのすぐ上に表示されます。
- 同じ操作を繰り返して、サプライヤバインド DN をリストに追加します。

g. 更新の参照先として、サーバの LDAP URL を指定します (任意)。

はじめてレプリカを構成する場合は、「現在のレフェラルの URL」のリストには何も表示されません。デフォルトでは、このリストには、レプリカデータのマスターを保持しているサーバの URL は表示されません (このレフェラルは、コンシューマサーバによって自動的に作成されます)。

自動レフェラルでは、クライアントが通常の接続を介してバインドするので、`ldap://`*servername*`:`*port* の形式であると想定しています。SSL を使用してクライアントをサプライヤにバインドする場合は、このフィールドに `ldaps://`*servername*`:`*port* の形式でレフェラルを指定する必要があります。ここで `ldaps` の `s` は、セキュリティで保護された接続を意味しています。

レフェラルに LDAP URL を指定すると、Directory Server では、最初にその URL で更新要求が照会されます。指定しない場合は、現在のレプリカのサプライヤで更新が照会されます。

レフェラルの新しい URL を指定するには、次の手順を実行します。

- 対応するフィールドに新しい LDAP URL を入力します。または「構築」をクリックすると、LDAP URL の作成を支援するダイアログボックスが表示されます。
- 「追加」をクリックします。「現在のレフェラルの URL」リストのすぐ上に、入力した LDAP URL が表示されます。
- 同じ操作を繰り返して、リストにレフェラルを追加します。

4. 「保存」をクリックして、レプリケーションの設定を保存します。

レプリケーション構成におけるすべてのコンシューマサーバに対して上記の手順を繰り返します。

サプライヤサーバおよびレプリカの構成

各サプライヤサーバに対して、次の手順を実行します。

1. サーバ A およびサーバ B 上で、サーバに対するサプライヤ設定を指定します。
   a. Directory Server Console で、「構成」タブをクリックします。
   b. ナビゲーションツリーで、Replication ノードを選択します。
   c. ウィンドウの右側にある「サプライヤ設定」タブをクリックします。
   d. 「更新履歴ログを有効にする」チェックボックスを選択します。

      これにより、ウィンドウの下の区画の無効にされていたフィールドが有効になります。
   e. 「デフォルトの使用」ボタンをクリックするか、または「参照」ボタンをクリックしてファイルセレクタを表示し、更新履歴ログを指定します。
   f. 更新履歴ログのパラメタ (数および保存期間) を設定します。

      別の値を指定する場合は、無期限のチェックボックスの選択を解除する必要があります。
   g. 「保存」をクリックして、サプライヤ設定を保存します。
2. サプライヤバインド DN に対応するエントリがない場合は、これを作成します。マルチマスターレプリケーションでは、レプリケーションがほかのサプライヤサーバに対してコンシューマとサプライヤの両方の役割を果たすため、サプライヤサーバ (と同様にコンシューマ) にこのサプライヤバインド DN を作成する必要があります。
   a. Directory Server Console で、「属性名」タブをクリックし、エントリを作成します。ここでは例として、`cn=Replication Manager,cn=config` を使用します。
   b. 属性と値がペアになった `userPassword` を指定します。
   c. パスワードの有効期限ポリシーを有効にしているか、将来有効にする場合は、パスワードの期限切れによってレプリケーションが失敗しないように、このポリシーを忘れずに無効にしてください。`userPassword` 属性でパスワードの有効期限ポリシーを無効にするには、`passwordExpirationTime` 属性に 20380119031407Z という値を追加します。こうするとパスワードの有効期限が切れなくなります。

注 サプライヤバインド DN は、アクセス制御の影響を受けない特権ユーザです。このエントリをレプリケートされたデータベースの一部にしないでください。

3. サーバ A およびサーバ B 上で、マルチマスターサプライヤレプリカに必要なレプリケーション設定を指定します。

   a. 「構成」タブのナビゲーションツリーで、Replication ノードを展開し、レプリケーション対象のデータベースを選択します。

      「レプリカの設定」タブがウィンドウの右側に表示されます。

   b. 「レプリカを有効にする」チェックボックスを選択します。

   c. 「レプリカロール」セクションの「複数マスター」ラジオボタンを選択します。

   d. 「共通設定」セクションで、レプリカ ID (1 ～ 65534 の整数) を指定します。

      各サプライヤレプリカのレプリカ ID は、一意である必要があります。このサーバおよびほかのサーバ上の他のサプライヤレプリカで使用される ID とは異なる ID を指定するようにしてください。

   e. 「共通設定」セクションの「パージ遅延」フィールドにパージ遅延を指定します。

      このオプションは、レプリケートされたエントリに状態情報を格納する期間を示します。パージ前の遅延期間は、レプリケーションのシャットダウンまたはエラーと復元に必要な長さで、同時にエントリに過度に多くのデータを保持させない長さにする必要があります。デフォルト値は 1 週間です。

   f. 「設定の更新」セクションで、サプライヤがレプリカにバインドするために使用するバインド DN (レプリケーションマネージャ DN) を指定します。

      はじめてレプリカを構成する場合は、「現在のサプライヤ DN」の一覧には何も表示されません。1 つのレプリカに複数のサプライヤバインド DN を指定できますが、1 つのレプリケーションアグリーメントには 1 つのサプライヤ DN しか指定できません。

      新しいサプライヤバインド DN を指定するには、次の手順を実行します。

   - 対応するフィールドに新しいサプライヤバインド DN を入力します。ここで入力する DN は、手順 2 で作成したエントリに対応していなければなりません (例 : `cn=Replication Manager,cn=config`)
   - 「追加」をクリックします。サプライヤバインド DN が「現在のサプライヤ DN」リストのすぐ上に表示されます。
   - 同じ操作を繰り返して、サプライヤバインド DN をリストに追加します。

注 サプライヤサーバでは、レフェラルの LDAP URL を指定する必要はありません。

g. 「保存」をクリックして、データベースに対するレプリケーション設定を保存します。

4. サーバ A 上で、次のレプリケーションアグリーメントを設定します。

   - サプライヤサーバ B との契約。サーバ B がレプリカのコンシューマとして構成される
   - サーバ C およびサーバ D の各コンシューマのための契約

   a. 「構成」タブのナビゲーションツリーで、レプリケートするデータベースをマウスの右ボタンでクリックし、「新規レプリケーションアグリーメント」を選択します。

      または、データベースを選択して、「オブジェクト」メニューから「新規レプリケーションアグリーメント」を選択することもできます。この操作を行うと、レプリケーションアグリーメントウィザードが開始されます。

   b. 「次」をクリックしてステップを進め、レプリケーションウィザードの全ステップを完了します。

      各フィールドへの入力方法については、オンラインヘルプを参照してください。

      サーバ B 上のコンシューマレプリカおよびサプライヤレプリカは、レプリケーションアグリーメントウィザードから初期化することができ、その後ならいつでも可能です。コンシューマレプリカの初期化の順序および手順については、306 ページの「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」および 315 ページの「コンシューマの初期化」を参照してください。

   ウィザードが完了すると、レプリケーションアグリーメントが設定されます。

5. サーバ B 上で、次のレプリケーションアグリーメントを設定します。

   - サプライヤサーバ A との契約。サーバ A がレプリカのコンシューマとして宣言される。手順 4 でサーバ A からサーバ B を初期化している場合は、この処理中にサーバ B からサーバ A を初期化してはならない
   - サーバ C およびサーバ D の各コンシューマのための契約

注 上記の手順を完了すると、サーバ A およびサーバ B には相互レプリケーションアグリーメントが作成されます。つまり相互の更新処理が可能になります。

サプライヤレプリカを保持するサーバ、必要なレプリケーションアグリーメント、およびコンシューマレプリカを保持するサーバを構成したら、レプリケーションを初期化することが準備できます。この作業は、サプライヤサーバ上にレプリケーションアグリーメントを作成するとき、またはそれ以降に実行できます。

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。このマスターを使用して、マルチマスターレプリケーションセットのほかのマスター上のサプライヤレプリカを初期化します。
2. コンシューマサーバ上のコンシューマレプリカを、2 つのマスターのうちのどれか 1 つから初期化します。

レプリカの初期化については、315 ページの「コンシューマの初期化」を参照してください。

カスケード型レプリケーションの構成

ここでは、カスケード型レプリケーションの設定について説明します。カスケード型レプリケーションモデルでは、サプライヤサーバは中間サーバ (ハブサーバと呼ばれる) を更新します。次に中間サーバは 1 つまたは複数のコンシューマサーバを更新します。ここでは、カスケード型レプリケーションの設定手順について説明します。

286 ページの図 8-4 に示す構成のように、サーバ A 上のサプライヤ、ハブサーバ B、およびコンシューマサーバ C 間にカスケード型レプリケーションを設定するには、次の手順を実行します。

1. コンシューマサーバ (サプライヤバインド DN とオプションで変更要求のレフェラル) およびコンシューマレプリカを構成します

   この手順については、307 ページの「コンシューマサーバおよびレプリカの構成」を参照してください。

2. ハブサプライヤ (更新履歴ログ、サプライヤバインド DN、およびオプションで変更要求のレフェラル) およびハブレプリカを構成します

   この手順については、309 ページの「ハブサプライヤおよびレプリカの構成」を参照してください。

3. サプライヤサーバ (更新履歴ログとレプリカ ID) およびサプライヤレプリカを構成します。

   この手順については、311 ページの「サプライヤサーバおよびレプリカの構成」を参照してください。

4. サプライヤサーバおよびハブサプライヤ上にレプリケーションアグリーメントを構成します。

   この手順については、312 ページの「レプリケーションアグリーメントの構成」を参照してください。

5. ハブサプライヤおよびコンシューマサーバ上のレプリカを初期化します。

   この手順については、314 ページの「カスケード型レプリケーションでのレプリカの初期化」を参照してください。

コンシューマサーバおよびレプリカの構成

1. コンシューマサーバ上に、レプリカの対象となるデータベースがない場合は、これを作成します。

   手順については、72 ページの「接尾辞の作成」を参照してください。

2. コンシューマサーバ上に、サプライヤバインド DN に対応するエントリがない場合は、これを作成します。これは、サプライヤがバインドするために使用する特別なエントリです。

   a. Directory Server Console で、「属性名」タブをクリックし、エントリを作成します。ここでは例として、`cn=Replication Manager,cn=config` を使用します。

   b. 属性と値がペアになった `userPassword` を指定します。

   c. パスワードの有効期限ポリシーを有効にしているか、将来有効にする場合は、パスワードの期限切れによってレプリケーションが失敗しないように、このポリシーを忘れずに無効にしてください。`userPassword` 属性でパスワードの有効期限ポリシーを無効にするには、`passwordExpirationTime` 属性に 20380119031407Z という値を追加します。こうするとパスワードの有効期限が切れなくなります。

注 サプライヤバインド DN は、アクセス制御の影響を受けない特権ユーザです。このエントリをレプリケートされたデータベースの一部にしないでください。

---

3. コンシューマサーバ上に、コンシューマレプリカに対するレプリケーション設定を指定します。

   a. Directory Server Console で、「構成」タブをクリックします。

b. ナビゲーションツリーで、Replication フォルダを展開し、レプリカデータベースを選択します。

「レプリカの設定」タブがウィンドウの右側に表示されます。

c. 「レプリカを有効にする」チェックボックスを選択します。

d. 「レプリカロール」セクションの「専用コンシューマ」ラジオボタンを選択します。

e. 「共通設定」セクションの「パージ遅延」フィールドにパージ遅延を指定します。

このオプションは、レプリケートされたエントリに状態情報を格納する期間を示します。パージ前の遅延期間は、レプリケーションのシャットダウンまたはエラーと復元に必要な長さで、同時にエントリに過度に多くのデータを保持させない長さにする必要があります。デフォルト値は 1 週間です。

コンシューマにレプリカ ID を指定する必要がないため、「レプリカ ID」フィールドは無効のままになります (このフィールドは、すべてのコンシューマレプリカで自動的に「65535」に設定されます)。

f. 「設定の更新」セクションで、サプライヤがレプリカにバインドするために使用するバインド DN (レプリケーションマネージャ DN) を指定します。

はじめてレプリカを構成する場合は、「現在のサプライヤ DN」のリストには何も表示されません。1 つのレプリカに複数のサプライヤバインド DN を指定できますが、1 つのレプリケーションアグリーメントには 1 つのサプライヤ DN しか指定できません。

新しいサプライヤバインド DN を指定するには、次の手順を実行します。

- 対応するフィールドに新しいサプライヤバインド DN を入力します。ここで入力する DN は、手順 2 で作成したエントリに対応していなければなりません (例 : `cn=Replication Manager,cn=config`)
- 「追加」をクリックします。サプライヤバインド DN が「現在のサプライヤ DN」リストのすぐ上に表示されます。
- 同じ操作を繰り返して、サプライヤバインド DN をリストに追加します。

g. 更新の参照先として、サーバの LDAP URL を指定します (省略可能)。

はじめてレプリカを構成する場合は、「現在のレフェラルの URL」のリストには何も表示されません。デフォルトでは、このリストには、レプリカデータのマスターを保持しているサーバの URL は表示されません (このレフェラルは、コンシューマサーバによって自動的に作成されます)。

自動レフェラルでは、クライアントが通常の接続を介してバインドするので、`ldap://`*servername*`:`*port* の形式であることを想定しています。SSL を使用してクライアントをサプライヤにバインドする場合は、このフィールドに `ldaps://`*servername*`:`*port* の形式でレフェラルを指定する必要があります。ここで `ldaps` の `s` は、セキュリティで保護された接続を意味しています。

レフェラルに LDAP URL を指定すると、Directory Server では、最初にその URL で更新要求が照会されます。指定しない場合は、現在のレプリカのサプライヤで更新が照会されます。

レフェラルの新しい URL を指定するには、次の手順を実行します。

- 対応するフィールドに新しい LDAP URL を入力します。または「構築」をクリックすると、LDAP URL の作成を支援するダイアログボックスが表示されます。
- 「追加」をクリックします。「現在のレフェラルの URL」リストのすぐ上に、入力した LDAP URL が表示されます。
- 同じ操作を繰り返して、リストにレフェラルを追加します。

4. 「保存」をクリックして、レプリケーションの設定を保存します。

ハブサプライヤおよびレプリカの構成

マスターからレプリケーションの更新を受け取り、それらをコンシューマに伝達するハブサプライヤ上で、次の手順を実行します。

1. レプリカの対象となるデータベースがない場合は、これを作成します。

   手順については、72 ページの「接尾辞の作成」を参照してください。

2. サプライヤバインド DN に対応するエントリがない場合は、これを作成します。これは、サプライヤがバインドするために使用する特別なエントリです。

   a. Directory Server Console で、「属性名」タブをクリックし、エントリを作成します。ここでは例として、`cn=Replication Manager,cn=config` を使用します。

   b. 属性と値がペアになった `userPassword` を指定します。

c. パスワードの有効期限ポリシーを有効にしているか、将来有効にする場合は、パスワードの期限切れによってレプリケーションが失敗しないように、このポリシーを忘れずに無効にしてください。`userPassword` 属性でパスワードの有効期限ポリシーを無効にするには、`passwordExpirationTime` 属性に 20380119031407Z という値を追加します。こうするとパスワードの有効期限が切れなくなります。

注 サプライヤバインド DN は、アクセス制御の影響を受けない特権ユーザです。このエントリをレプリケートされたデータベースの一部にしないでください。

---

3. ハブレプリカに対するレプリケーション設定を指定します。

   a. Directory Server Console で、「構成」タブをクリックします。

   b. ナビゲーションツリーで、Replication フォルダを展開し、レプリカデータベースを選択します。

      「レプリカの設定」タブがウィンドウの右側に表示されます。

   c. 「レプリカを有効にする」チェックボックスを選択します。

   d. 「レプリカロール」セクションの「ハブ」ラジオボタンを選択します。

   e. 「共通設定」セクションの「パージ遅延」フィールドにパージ遅延を指定します。

      このオプションは、レプリケートされたエントリに状態情報を格納する期間を示します。パージ前の遅延期間は、レプリケーションのシャットダウンまたはエラーと復元に必要な長さで、同時にエントリに過度に多くのデータを保持させない長さにする必要があります。デフォルト値は 1 週間です。

      ハブサプライヤにレプリカ ID を指定する必要はないため、「レプリカ ID」フィールドは無効のままになります (このフィールドは、コンシューマレプリカと同じように自動的に「65535」に設定されます)。

   f. 「設定の更新」セクションで、サプライヤがレプリカにバインドするために使用するバインド DN (レプリケーションマネージャ DN) を指定します。

      はじめてレプリカを構成する場合は、「現在のサプライヤ DN」のリストには何も表示されません。1 つのレプリカに複数のサプライヤバインド DN を指定できますが、1 つのレプリケーションアグリーメントには 1 つのサプライヤ DN しか指定できません。

      新しいサプライヤバインド DN を指定するには、次の手順を実行します。

      ❍ 対応するフィールドに新しいサプライヤバインド DN を入力します。ここで入力する DN は、手順 2 で作成したエントリに対応していなければなりません (例: `cn=Replication Manager,cn=config`)

- 「追加」をクリックします。サプライヤバインド DN が「現在のサプライヤ DN」リストのすぐ上に表示されます。
- 同じ操作を繰り返して、サプライヤバインド DN をリストに追加します。

g. 更新の参照先として、サーバの LDAP URL を指定します (任意)。

はじめてレプリカを構成する場合は、「現在のレフェラルの URL」のリストには何も表示されません。デフォルトでは、このリストには、レプリカデータのマスターを保持しているサーバの URL は表示されません (このレフェラルは、ハブサーバによって自動的に作成されます)。

自動レフェラルでは、クライアントが通常の接続を介してバインドするので、`ldap://`*servername*`:`*port* の形式であることを想定しています。SSL を使用してクライアントをサプライヤにバインドする場合は、このフィールドに `ldaps://`*servername*`:`*port* の形式でレフェラルを指定する必要があります。ここで `ldaps` の `s` は、セキュリティで保護された接続を意味しています。

レフェラルの LDAP URL を指定すると、Directory Server は、最初にその URL で変更要求を照会します。URL を指定しない場合は、現在のレプリカのサプライヤで変更要求が照会されます。

レフェラルの新しい URL を指定するには、次の手順を実行します。

- 対応するフィールドに新しい LDAP URL を入力します。または「構築」をクリックすると、LDAP URL の作成を支援するダイアログボックスが表示されます。
- 「追加」をクリックします。「現在のレフェラルの URL」リストのすぐ上に、入力した LDAP URL が表示されます。
- 同じ操作を繰り返して、リストにレフェラルを追加します。

4. 「保存」をクリックして、レプリカの設定を保存します。

サプライヤサーバおよびレプリカの構成

データベースの元のコピーを保持するサプライヤサーバ上で、次の手順を実行します。

1. サーバに対するサプライヤ設定を指定します。
   a. Directory Server Console で、「構成」タブをクリックします。
   b. ナビゲーションツリーで、Replication ノードを選択します。
   c. ウィンドウの右側にある「サプライヤ設定」タブをクリックします。
   d. 「更新履歴ログを有効にする」チェックボックスを選択します。

   これにより、ウィンドウの下の区画の無効にされていたフィールドが有効になります。

e. 「デフォルトの使用」ボタンをクリックするか、または「参照」ボタンをクリックしてファイルセレクタを表示し、更新履歴ログを指定します。

f. 更新履歴ログのパラメタ (数および保存期間) を設定します。

別の値を指定する場合は、無期限のチェックボックスの選択を解除する必要があります。

g. 「保存」をクリックして、サプライヤ設定を保存します。

2. 必要なレプリケーション設定を指定します。

a. 「構成」タブのナビゲーションツリーで、Replication ノードを展開し、レプリケーション対象のデータベースを選択します。

「レプリカの設定」タブがウィンドウの右側に表示されます。

b. 「レプリカを有効にする」チェックボックスを選択します。

c. 「レプリカロール」セクションの「単一マスター」ラジオボタンを選択します。

d. 「共通設定」セクションで、レプリカ ID (1 ～ 65534 の整数) を指定します。

各サプライヤレプリカのレプリカ ID は、一意である必要があります。このサーバおよびほかのサーバ上の他のサプライヤレプリカで使用される ID とは異なる ID を指定するようにしてください。

e. 「共通設定」セクションの「パージ遅延」フィールドにパージ遅延を指定します。

このオプションは、レプリケートされたエントリに状態情報を格納する期間を示します。パージ前の遅延期間は、レプリケーションのシャットダウンまたはエラーと復元に必要な長さで、同時にエントリに過度に多くのデータを保持させない長さにする必要があります。デフォルト値は 1 週間です。

f. 「保存」をクリックして、データベースに対するレプリケーション設定を保存します。

レプリケーションアグリーメントの構成

カスケード型のレプリケーション環境を構成する場合は、まず次の順序で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。

- 最初にサプライヤサーバでサプライヤとハブサプライヤ間のレプリケーションを定義する
- 次にハブサプライヤでハブサプライヤとコンシューマ間のレプリケーションを定義する

この順序で操作を実行することにより、レプリケーションアグリーメントの作成時に、ハブサプライヤ上のレプリカおよびコンシューマ上のレプリカを初期化することもできます。

1. サプライヤサーバ上に、このサーバとハブサプライヤの間のレプリケーションアグリーメントを設定します。

   a. 「構成」タブのナビゲーションツリーで、レプリケートするデータベースをマウスの右ボタンでクリックし、「新規レプリケーションアグリーメント」を選択します。

      または、データベースを選択して、「オブジェクト」メニューから「新規レプリケーションアグリーメント」を選択することもできます。この操作を行うと、レプリケーションアグリーメントウィザードが開始されます。

   b. 「次」をクリックしてステップを進め、レプリケーションウィザードの全ステップを完了します。

      各フィールドへの入力方法については、オンラインヘルプを参照してください。

      この時点以降、ハブサプライヤ上のレプリカを初期化できます。レプリカを後で初期化する場合の手順については、315 ページの「コンシューマの初期化」を参照してください。

2. ハブサプライヤとコンシューマの間のレプリケーションアグリーメントを、ハブのサプライヤ上に設定します。

   手順 1 と同じ手順を実行します。レプリケーションウィザードからコンシューマサーバ上のレプリカを初期化できます。コンシューマサーバを後で初期化する場合の手順については、315 ページの「コンシューマの初期化」を参照してください。

注 SSL を経由したレプリケーションアグリーメントでは、コンシューマサーバのホスト名は完全指定によるドメイン名 (例えば、server.remote.siroe.com など) として指定する必要があります。エイリアス、IP アドレス、またはドメイン名のローカルの部分だけを指定しないでください。SSL を経由したレプリケーションが許可されなくなります。さらに、「man-in-the-middle」による攻撃からレプリケーションを保護できなくなります。

デフォルトでは、サプライヤはコンシューマサーバの証明書のあるパスを有効にします。サプライヤの信頼する CA (認証局) のルートストアは、SSL を経由したレプリケーションまたはクライアント認証に使用されている CA からの証明書のみである必要があります。SSL を経由したレプリケーションで、「man-in-the-middle」による攻撃から保護するため、コンシューマサーバの証明書に CN 属性にサブジェクト識別名があるか、拡張子が完全指定によるドメイン名と一致することが分っている場合、nsSslServerAuth 構成属性は cncheck 値を持つ必要があります。

カスケード型レプリケーションでのレプリカの初期化

レプリケーションアグリーメントの構成中にレプリカを初期化しない場合は、いつでも 315 ページの「コンシューマの初期化」で説明した手順に従って、この処理を実行できます。ただし、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. サプライヤサーバから、ハブサプライヤ上のレプリカを初期化します。
2. ハブサプライヤから、コンシューマ上のレプリカを初期化します。

更新履歴ログの削除

更新履歴ログとは、指定されたレプリカに対するすべての変更内容を記録したもので、これを使用してサプライヤはコンシューマサーバ (またはマルチマスターレプリケーションの場合はマスター) 上のレプリカにその変更をリプレイします。サプライヤサーバがオフラインになった場合、更新履歴ログには、すべての変更内容を正確に記録することができず、レプリケーションには使用すべきではないので、更新履歴ログを削除可能なことは重要です。更新履歴ログを削除すると、コンシューマを初期化したり、レプリケーションを新しく始めたりすることができます。更新履歴ログを削除するには、それを削除するか、ほかの場所に移動します。

ここでは、次の手順を説明します。

- 314 ページの「更新履歴ログの削除」
- 315 ページの「更新履歴ログの移動」

更新履歴ログの削除

Directory Server Console を使用すると、更新履歴ログを削除することができます。サプライヤサーバから更新履歴ログを削除するには、次の手順を実行します。

1. Directory Server Console で、「構成」タブを選択します。
2. 左側のナビゲーションツリーにある Replication フォルダを選択してから、右側の区画にある「サプライヤサーバ設定」タブを選択します。
3. 「更新履歴ログを有効にする」チェックボックスの選択を解除します。

   これにより、更新履歴ログが削除されます。
4. 「保存」をクリックします。
5. Directory Server を再起動します。

6. コンシューマを初期化し直します。

   詳細は、315 ページの「コンシューマの初期化」を参照してください。

注	更新履歴ログを削除したあとで、コンシューマサーバを初期化し直す必要があります。

更新履歴ログの移動

サーバが稼働しており更新履歴ログへの記録を続けている時に更新履歴ログを削除するには、更新履歴ログを単に新しい場所に移動します。更新履歴ログを移動すると、指定したディレクトリに新しい更新履歴ログが作成され、古い更新履歴ログは削除されます。更新履歴ログの位置の変更は、ログの再初期化と同じことなので、コンシューマも初期化し直す必要があります。

たとえば、更新履歴ログをデフォルト位置の `/var/ds5/slapd-`*serverID*`/changelogdb` から `/var/ds5/slapd-`*serverID*`/newchangelog` に移動することができます。この操作は、Directory Server Console から実行する必要があります。ファイルシステムの `rename` または `mv` コマンドを使用しないでください。

コンシューマの初期化

一度レプリケーションアグリーメントを作成すると、コンシューマを初期化、つまりデータを物理的にサプライヤサーバからコンシューマサーバにコピーする必要があります。ここでは、コンシューマの初期化について詳細に解説したあと、コンシューマを初期化する 2 つの方法で操作を説明します。この節は、次の項目で構成されています。

- 315 ページの「コンシューマの初期化のタイミング」
- 316 ページの「Console を使用したオンラインでのコンシューマの初期化」
- 317 ページの「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」

コンシューマの初期化のタイミング

コンシューマの初期化には、サプライヤサーバのデータをコンシューマサーバにコピーする処理が含まれます。サブツリーがコンシューマ上に物理的に配置されると、サプライヤサーバはコンシューマサーバに対して更新処理を再実行できるようになります。

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。ただし、何らかの理由によりサプライヤサーバのデータがバックアップから復元された場合、対応するコンシューマは、すべて初期化し直す必要があります。

Console を使用してオンラインでコンシューマを初期化するか、コマンド行を使用して手動で初期化できます。Console を使用したオンラインコンシューマの初期化は、少数のコンシューマを初期化するのに効果的な方法です。ただし、この方法は、レプリカを 1 つずつ初期化していくため、多数のレプリカを処理する場合には適していません。オンラインによるコンシューマの初期化は、サプライヤサーバでのレプリケーションアグリーメントの設定中、コンシューマの初期化に使用する方法です。

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は、1 つの LDIF ファイルから多数のコンシューマを初期化する場合により効果的な方法です。

Console を使用したオンラインでのコンシューマの初期化

Console を使用したオンラインコンシューマの初期化は、コンシューマの初期化と再初期化にもっとも簡単な方法です。ただし、低速の接続を経由したレプリケーションの場合は、この方法では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、317 ページの「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」を参照してください。

注 コンシューマサーバがオンラインコンシューマの作成方法を使用して初期化されている場合、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サプライヤサーバを参照します。

オンラインでのコンシューマ初期化の実行

コンシューマをオンラインで初期化または再初期化するには、次の手順を実行します。

1. Directory Server Console でサプライヤサーバを選択し、次に「構成」タブを選択します。
2. Replication フォルダを展開し、レプリケートされたデータベースを展開します。レプリケーションアグリーメントをマウスの右ボタンでクリックし、ドロップダウンメニューから「コンシューマの初期化」を選択します。

   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警告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。

ただちにオンラインコンシューマの初期化が開始されます。レプリケーションアグリーメントでオンラインコンシューマの初期化の状態を確認できます。オンラインコンシューマの初期化の処理中は、レプリカを初期化中であるというコンシューマ初期化の状態が示されます。

このウィンドウを更新するには、ナビゲーションツリーのレプリケーションアグリーメントアイコンをマウスの右ボタンでクリックし、「再読み込み」を選択します。オンラインコンシューマの初期化が終了すると、状態がこれを反映するものに変わります。

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。

コマンド行を使用した手動によるコンシューマの初期化

コマンド行を使用して手動でコンシューマを初期化するのは、多数のエントリのレプリケーションがあるサイトで、コンシューマの初期化が最も早い方法です。ただし、手動によるコンシューマの作成は、オンラインコンシューマの作成と比べてプロセスが複雑です。性能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。

この節は、次の項目で構成されています。

手動によるコンシューマ初期化の概要

手動でサーバを初期化または再初期化するには、次の手順を実行します。

1. サプライヤサーバのレプリカを LDIF ファイルにエクスポートします。

   318 ページの「LDIF ファイルへのレプリカのエクスポート」を参照してください。

2. サプライヤ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サーバにインポートします。

   手順については、318 ページの「コンシューマサーバへの LDIF ファイルのインポート」を参照してください。

注 カスケード型のレプリケーション環境では、サプライヤサーバからエクスポートされた LDIF ファイルを使用して、ハブサーバとハブサーバのコンシューマの両方を初期化できます。

LDIF ファイルへのレプリカのエクスポート

レプリカを LDIF ファイルに変換するには、次の 3 つの手順のうち 1 つを実行します。

1. レプリケーションウィザードの「コンシューマの初期化」ダイアログボックスで「コンシューマの初期化ファイルの作成」を選択して、レプリケーションアグリーメントを作成します。
2. Directory Server Console で、任意のときに Replication フォルダのレプリケーションアグリーメント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「レプリカのエクスポート」を選択します。
3. コマンド行で、145 ページの「コマンド行からの LDIF へのエクスポート」に記載されている export コマンドを使用します。

コンシューマサーバへの LDIF ファイルのインポート

サプライヤ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サーバにインポートするには、Directory Server Console のインポート機能を使用するか、`directoryserver ldif2db` または `directoryserver ldif2db-task` コマンドを使用します。これらのインポート方法については、140 ページの「コマンド行からのインポート」を参照してください。

`ldif2db-task` を使用する場合、コンシューマサーバ上に構成されているサプライヤバインド DN を使用してバインドする必要があります。

注 `ldif2db-task` を使用する場合は、LDIF ファイルのインポート操作をする前にサーバをシャットダウンする必要がありません。

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には、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確実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では、マルチマスターセットのもう 1 つのサーバ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、ハブのサプライヤや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、サプライヤサーバ側により更新をする必要があります。

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 ここに記載されている手順は、レプリケーションが設定が完了し、さらにコンシューマを初期化した直後にだけ使用できます。

レプリケーションの再試行アルゴリズム

サプライヤサーバがコンシューマへのレプリケーションに失敗した場合は、時間間隔を増加させながら定期的にレプリケーションを再試行します。再試行パターンは、10、20、40、80 秒と間隔が 5 分になるまで繰り返されます。その後、5 分間隔で無限に再試行が繰り返されます。

サプライヤサーバとコンシューマサーバの間で、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 分を超えたサーバを直ちに最新の状態に戻すのに、これは不十分であることに注意してください。

サーバがオンラインに戻った直後にディレクトリ情報を確実に同期させるには、ディレクトリ情報の参照コピーを保持するサプライヤサーバで Directory Server Console を使用するか、カスタマイズ可能なスクリプトを使用します。

Console を使用した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構成のサプライヤ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情報を保持しているサプライヤサーバ上で次の手順を実行します。

1. Directory Server Console で「構成」タブをクリックし、Replication フォルダとデータベースノードを展開して、更新が必要なレプリカに対応するレプリケーションアグリーメントを選択します。
2. レプリケーションアグリーメントをマウスの右ボタンでクリックし、ドロップダウンリストから「今すぐ更新」を選択します。

   これにより、更新が必要な情報を保持しているサーバに対してレプリケーションが開始されます。

SSL を介したレプリケーション

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 を構成できます。

SSL を経由してレプリケーションを使用するには、まず次の手順を実行する必要があります。

- サプライヤサーバとコンシューマサーバの両者を、SSL を使用するように構成する
- サプライヤサーバの証明書をサプライヤ DN として認識するようにコンシューマサーバを構成するただしこの手順は、単純な認証ではなく SSL クライアント認証を使用する場合にだけ実行すること

手順については、第 11 章「SSL の管理」を参照してください。

注 SSL を経由したレプリケーションは、次の場合には失敗します。

- サプライヤの証明書が自己署名である場合
- サプライヤの証明書が SSL サーバ専用のものである場合 (SSL ハンドシェークの実行中にクライアントとして動作できない)

サーバが SSL を使用するように構成されている場合、次の方法を使用して、必ず SSL 接続を経由したレプリケーションを実行するようにします。

- 2 つの Directory Server 間のレプリケーションアグリーメントを設定するときに、レプリケーションウィザードを使用する
- 最初のレプリケーションアグリーメントを構成したあとで、Directory Server Console を使用する

レプリケーションウィザードを使用した SSL によるレプリケーションの構成

1. サプライヤサーバの Directory Server Console で、「構成」タブをクリックし、Replication フォルダを展開して、レプリケーション対象のデータベースを選択します。
2. データベースをマウスの右ボタンでクリックし、ドロップダウンメニューから「新規レプリケーションアグリーメント」を選択します。

   レプリケーションアグリーメントウィザードが表示されます。

3. レプリケーションアグリーメントウィザードの各ステップを実行していくと、「複製元と複製先」ウィンドウが表示されます。
4. 「接続」セクションで、「暗号化 SSL 接続の使用」をオンにします。
5. 「SSL クライアント認証」または「簡易認証」を選択します。

   「SSL クライアント認証」を選択すると、サプライヤサーバとコンシューマサーバは、証明書を使用してお互いを認証し合います。

   「簡易認証」を選択すると、サプライヤサーバとコンシューマサーバは、バインド DN とパスワードを使用してお互いを認証し合います。この場合、バインド DN とパスワードをテキストフィールドに入力する必要があります。このオプションを選択すると、セキュリティが確保されたチャネルを経由して単純な認証が実行されますが、証明書は発行されません。
6. 「次」をクリックして、レプリケーションの設定に進みます。

注 SSL を経由したレプリケーションアグリーメントでは、コンシューマサーバのホスト名は完全指定によるドメイン名 (例えば、`server.remote.siroe.com` など) として指定する必要があります。エイリアス、IP アドレス、またはドメイン名のローカルの部分だけを指定しないでください。SSL を経由したレプリケーションが許可されなくなります。さらに、「man-in-the-middle」による攻撃からレプリケーションを保護できなくなります。

デフォルトでは、サプライヤはコンシューマサーバの証明書のあるパスを有効にします。サプライヤの信頼する CA (認証局) のルートストアは、SSL を経由したレプリケーションまたはクライアント認証に使用されている CA からの証明書のみである必要があります。SSL を経由したレプリケーションで、「man-in-the-middle」による攻撃から保護するため、コンシューマサーバの証明書に CN 属性にサブジェクト識別名があるか、拡張子が完全指定によるドメイン名と一致することが分っている場合、`nsSslServerAuth` 構成属性は `cncheck` 値を持つ必要があります。

Console を使用した SSL によるレプリケーションの設定

1. サプライヤサーバ上で Directory Server Console を起動し、「構成」タブをクリックして Replication フォルダを展開し、修正するレプリケーションアグリーメントを選択して SSL によるレプリケーションを有効にします。
2. 右側のナビゲーションウィンドウで「接続」タブをクリックします。

   レプリケーション接続設定が表示されます。
3. 「接続」セクションで、「暗号化 SSL 接続の使用」をオンにします。

4. 「SSL クライアント認証」または「簡易認証」を選択します。

   「SSL クライアント認証」を選択すると、サプライヤサーバとコンシューマサーバは、証明書を使用してお互いを認証し合います。

   「簡易認証」を選択すると、サプライヤサーバとコンシューマサーバは、バインド DN とパスワードを使用してお互いを認証し合います。この場合、バインド DN とパスワードをテキストフィールドに入力する必要があります。このオプションを選択すると、セキュリティが確保されたチャネルを経由して単純な認証が実行されますが、証明書は発行されません。

5. 「保存」をクリックします。

旧バージョンからのレプリケーション

ここでは、iPlanet Directory Server の旧バージョンからのレプリケーションを最適化する方法について説明します。iPlanet Directory Server 5.1、Directory Server の旧バージョンとのレプリケーションモデルに関係させるには、次の条件を満たす必要があります。

- Directory Server 5.1 がコンシューマとしてレプリケーションアグリーメントに定義されている
- 旧バージョンのサプライヤが Directory Server 4.0 または 4.1x である

この場合、次のような制限があります。

- 旧バージョンの Directory Server と 5.1 Directory Server は、同じレプリカを更新することはできない。しかし、5.1 Directory Server は旧バージョンの Directory Server から提供されたものと 5.1 Directory Server から提供されたものの、異なるレプリカを保持することになる
- Directory Server 5.1 は、ほかのレプリカのサプライヤ例になることはできない

旧バージョンの Directory Server のコンシューマとして Directory Server 5.1 を使用することのできる利点は、レプリケートされた環境の移行が簡単になることです。

旧バージョンの Directory Server のコンシューマとしての Directory Server 5.1 の構成

Directory Server 5.1 を Directory Server の旧バージョンに対応するコンシューマとして使用する場合は、次のように構成する必要があります。

1. Directory Server Console で、「構成」タブをクリックします。

2. 「構成」タブで、Replication ノードを選択し、右側の区画で「古いバージョンのコンシューマ設定」タブをクリックします。
3. 「古いバージョンのコンシューマを有効にする」チェックボックスを選択します。

   これにより、「認証」ボックスのフィールドが有効になります。
4. 旧バージョンのサプライヤサーバがバインドするために使用するサプライヤ DN を指定します。

   サプライヤのパスワードを指定します (任意)。ここでパスワードは、8 文字以上にする必要があります。
5. 「保存」をクリックします。

   ここでは、旧バージョンのサプライヤから更新を受け取るそれぞれのレプリカについて、コンシューマ設定を構成する必要があります。
6. ナビゲーションツリーで Replication ノードを展開し、旧バージョンのサプライヤから更新を受け取るレプリカを選択します。
7. ウィンドウの右側にある「レプリカの設定」タブの「共通設定」セクションで、「レプリカを有効にする」および「4.x のレプリカによる更新」チェックボックスを選択します。

   これらのオプションは、レプリケーションの実行に必要な唯一の設定です。手順 4 で指定したサプライヤ DN が使用されるため、サプライヤ DN の指定は必要ありません。
8. 「保存」をクリックします。

   旧バージョンのサプライヤから更新を受け取るコンシューマレプリカに対して、それぞれ 手順 7 から 手順 8 までを繰り返します。

   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 5.1 にレプリケートする旧バージョンのサプライヤを構成する必要があります。4.x Directory Server 上にレプリケーションアグリーメントを構成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。

注 Directory Server Console では、データベースのサプライヤレプリカとしての構成と旧バージョンのコンシューマの有効化を防げません。これによって、Directory Server 5.1 への移行後の構成にし、移行の期間だけに限って旧バージョンのコンシューマ設定を有効にすることができるため、移行作業がより簡単になります。

レトロ履歴ログのプラグインの使用

レトロログのプラグインを使用すると、Directory Server 4.x に実装されたログと互換性のある更新履歴ログを維持するように iPlanet Directory Server 5.1 を設定できます。レトロログの維持は、iPlanet Directory Server 5.1 と iPlanet Meta Directory を共存させる場合に重要です。また、Directory Server 4.x スタイルの更新履歴ログに依存しているディレクトリクライアントについても、レトロログを維持する必要がある場合があります。

レトロログのプラグインを使用するには、iPlanet Directory Server 5.1 を単一マスターレプリケーションモデルのサプライヤサーバとして構成する必要があります。

レトロログを維持するように iPlanet Directory Server 5.1 を設定すると、更新履歴ログは `cn=changelog` という接尾辞の付いた別のデータベースに格納されます。

レトロログは、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、表 8-1 にリストされている属性を含めることができます。

表 8-1 レトロ履歴ログエントリの属性

属性	定義
`changeNumber`	1 つの値からなるこの属性は常に存在し、各更新を一意に識別する整数を含む。この番号は、更新の発生した順序に関連し、番号が大きいほど、更新の順序は後ろであることを表す
`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` 属性にはエントリが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間を指定する
`changeType`	LDAP 処理のタイプが指定される。この属性は、**`add`**、**`delete`**、**`modify`**、または **`modrdn`** の値のどれか 1 つである
`changes`	add または modify 処理の場合、エントリに対する更新が LDIF 形式で含まれる
`newRDN`	`modrdn` 処理の場合、エントリの新しい RDN が指定される
`deleteOldRdn`	`modrdn` 処理の場合、古い RDN が削除されたかどうかが指定される
`newSuperior`	`modrdn` 処理の場合、エントリの `newSuperior` 属性が指定される

この節は、レトロログに関する次の項目で構成されています。

レトロ履歴ログのプラグインの有効化

レトロログプラグインの構成情報は、`dse.ldif` の `cn=Retro Changelog Plugin,cn=plugins,cn=config` エントリに保持されています。

Directory Server Console からレトロログのプラグインを有効にする手順は、Directory Server のその他のプラグインの場合と同じです。詳細は、444 ページの「Server Console を使用したプラグインの有効化と無効化」を参照してください。

コマンド行からレトロログのプラグインを有効にするには、次の手順を実行します。

1. 次の LDIF 更新文を含む LDIF ファイルを作成します。

```
dn: cn=Retro Changelog Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginenabled
nsslapd-pluginenabled:on
```

2. `ldapmodify` コマンドを使用して、LDIF ファイルをディレクトリにインポートします。
3. サーバを再起動します。

   サーバの再起動については、35 ページの「iPlanet Directory Server の起動と停止」を参照してください。

レトロログは、ディレクトリツリーの `cn=changelog` という特別な接尾辞の下に作成されます。

レトロ履歴ログの削除

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` 構成属性を設定する必要があります。

`nsslapd-changelogmaxage` 属性は、次の形式の単一値属性です。

`nsslapd-changelogmaxage:` *Integer timeUnit*

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。

注	*Integer* 変数と *timeUnit* 変数の間には空白を挿入しません。上の構文では、属性値が 1 つではなく 2 つの変数部からなることを表すために、空白が挿入されています。

`nsslapd-changelogmaxage` 値の例 :

```
nsslapd-changelogmaxage: 2d
```

レトロ履歴ログの検索と変更

このような情報へのアクセスを許可されるのは、認証アプリケーションおよびユーザに制限する必要があります。更新履歴ログは検索処理をサポートしており、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、レトロ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。レトロ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。

レトロ履歴ログとアクセス制御ポリシー

レトロ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。

- レトロ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 (`userdn=anyone` のユーザ。`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
- ディレクトリマネージャ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に付与しないでください。認証済みのアプリケーションおよびユーザにのみ、この情報に対するアクセス権を付与すべきです。

レトロ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` 属性を変更します。

レプリケーション状態の監視

Directory Server Console を使用して、レプリケーションの状態を監視することができます。

レプリケーション状態の概要を表示するには、次の手順を実行します。

1. Directory Server Console で、「状態」タブをクリックし、左側のナビゲーションツリーにある「レプリケーション状態」を選択します。

   サーバに構成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、右側の区画に表示されます。

2. 「再読み込み」をクリックして、タブのコンテンツを更新します。

表示される状態情報については、表 8-2 に記載されています。

表 8-2 Directory Server Console - 「状態」タブ

テーブルの見出し	内容
Agreement	レプリケーションアグリーメントを設定するときに指定する名前を含む
Replica suffix	レプリケートされた接尾辞を含む
Supplier	契約内のサプライヤサーバを表示する
Consumer	契約内のコンシューマサーバを表示する
Number of changes	サーバの起動時からレプリカに対して送られた更新の数を示す
Last replica update began	最終のレプリケーション更新開始日時を示す
Last replica update ended	最終のレプリケーション更新終了日時を示す
Last update message	最終のレプリケーション更新の状態を示す
Consumer initialization	コンシューマ初期化の現在の状態 (実行中かどうか) を示す
Last consumer initialization update message	コンシューマを最後に初期化したときの状態を示す
Last consumer initialization began	コンシューマレプリカの初期化開始日時を示す
Last consumer initialization ended	コンシューマレプリカの初期化終了日時を示す

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から同時に変更することができます。同一エントリへの変更が実施されたあと、2 つのサーバ間でレプリケーションが発生した場合、更新の競合を解決する必要があります。ほとんどの場合、各サーバ上での更新に関連したタイムスタンプを基に競合は自動的に解決され、最終の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、競合マーカー属性 `nsds5ReplConflict` が含まれます。`nsdsReplConflict` 属性は操作属性 (operational attribute) です。これによって、この属性を含むエントリを簡単に検索できます。

たとえば、次の `ldapsearch` コマンドを使用できます。

```
% ldapsearch -D adminDN -w passwd \
-b "dc=siroe,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。

ここでは、次の競合を解決する手順について説明します。

- 「命名の競合の解決」(328 ページ)
- 「親のないエントリの競合の解決」(330 ページ)
- 「潜在的な相互運用性の問題の解決」(331 ページ)

命名の競合の解決

レプリケーション中に同一の DN を持つ 2 つのエントリが別々のサーバ上に作成されていた場合、DN にそのエントリの一意の識別子を含めることによって、競合を解決する自動処理により、最後に作成されたエントリの名前が変更されます。各ディレクトリエントリは、操作属性 `nsuniqueid` によって指定された一意の識別子を持ちます。命名の競合が発生すると、この一意の ID が、一意でない DN に追加されます。

たとえば、サーバ A では t1 という時刻にエントリ `uid=adamss,ou=people,dc=siroe,dc=com` が作成され、サーバ B では t1 より遅い t 2 という時刻に同一のエントリが作成された場合、レプリケーションが実行されると、サーバ A とサーバ B のエントリは、次のようになります。

- `uid=adamss,ou=people,dc=siroe,dc=com` (t1 の時刻に作成)
- `nsuniqueid=66446001-1dd211b2+uid=adamss,dc=siroe,dc=com` (t2 の時刻に作成)

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。名前変更の手順は、命名属性が 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。

複数の値からなる命名属性を持つエントリの名前変更

複数の値からなる命名属性を持つエントリに対して名前を変更するには、次の手順を実行します。

1. 命名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。

```
prompt% ldapmodify -D adminDN -w passwd

>dn: nsuniqueid=66446001-1dd211b2+uid=adamss,dc=siroe,dc=com
>changetype: modrdn
>newrdn: uid=NewValue
>deleteoldrdn: 0
```

2. 命名属性の古い 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。

```
prompt% ldapmodify -D adminDN -w passwd

dn: uid=NewValue,dc=siroe,dc=com
changetype: modify
delete:uid
uid: adamss
-
delete: nsds5ReplConflict
-
```

注 一意の識別子属性 nsuniqueid は削除できないため、この処理は 2 段階で実行されます。

ldapmodify コマンドについては、51 ページの「コマンド行からのエントリの管理」および『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照してください。

1 つの値からなる命名属性を持つエントリの名前変更

1 つの値からなる命名属性を持つエントリに対して名前を変更するには、次の手順を実行します。

1. 別の命名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。

```
prompt% ldapmodify -D adminDN -w passwd
```

```
>dn: nsuniqueid=66446001-1dd211b2+dc=pubs,dc=siroe,dc=com
>changetype: modrdn
>newrdn: cn=TempValue
>deleteoldrdn: 0
```

2. 命名属性の古い 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。

```
prompt% ldapmodify -D adminDN -w passwd

>dn: cn=TempValue,dc=siroe,dc=com
>changetype: modify
>delete:dc
>dc: pubs
>-
>delete: nsds5ReplConflict
>-
```

注 一意の識別子属性 nsuniqueid は削除できないため、この処理は 2 段階で実行されます。

3. 目的の属性と値のペアを含む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。

```
prompt% ldapmodify -D adminDN -w passwd

>dn: cn=TempValue,dc=siroe,dc=com
>changetype: modrdn
>newrdn: dc=NewValue
>deleteoldrdn: 1
```

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 cn=*TempValue* 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。

ldapmodify コマンドについては、51 ページの「コマンド行からのエントリの管理」を参照してください。

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、エントリの追加のあとにレプリケーションが実行され、コンシューマサーバ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、マッチ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、一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 passwd
```

```
> dn:dc=siroe,dc=com
>changetype: modify
>delete:aci
>aci: (target ="ldap:///dc=siroe,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=siroe,dc=com")(targetattr!="userPassword")
(targetfilter="(!(nsds5ReplConflict=*))")(version 3.0;acl "Anonymous
read-search access";allow (read, search, compare)
(userdn="ldap:///anyone");)
```

```
> -
```

新しい ACI 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

iPlanet 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 (schema) 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を作成など、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。

- スキーマ拡張の概要
- スキーマ検査のオン / オフの切り替え
- オブジェクトクラスの管理
- 属性の管理

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の詳細と、スキーマ拡張のガイドラインについては、『iPlanet Directory Server 導入ガイド』を参照してください。標準の属性およびオブジェクトクラスについては、『iPlanet Directory Server スキーマリファレンス』を参照してください。

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. 新しい属性を作成します。詳細は、336 ページの「属性の作成」を参照してください。
2. 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、340 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。

属性の管理

Directory Server Console では、スキーマ内の全属性を表示したり、そのスキーマへの属性拡張を作成、編集、および削除したりできます。次の節では、属性の管理方法を説明します。

- 「属性の表示」(334 ページ)
- 「属性の作成」(336 ページ)
- 「属性の編集」(337 ページ)
- 「属性の削除」(338 ページ)

オブジェクトクラスの管理については、338 ページの「オブジェクトクラスの管理」を参照してください。

属性の表示

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 Console で、「構成」タブを選択します。
2. 左側のナビゲーションツリーで、Schema フォルダを選択し、右側の区画で「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 (読み取り専用) およびユーザ定義属性を一覧表示する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。

   次の表に、属性テーブルのフィールドを示します。

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 は通常、ピリオドで区切られた 10 進数からなる文字列。オブジェクトクラスや属性などのオブジェクトを一意に識別する。OID を指定しないと、Directory Server は自動的に *attribute_name*-oid を使用する。たとえば、OID を指定しないで属性 birthdate を作成した場合、Directory Server は OID として自動的に birthdate-oid を使用する OID に関する詳細情報や、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/iana/ にアクセスすること
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、335 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 属性構文の定義

構文および OID	定義
Binary (以前は bin)	属性値がバイナリ形式であることを示す
Boolean	この属性の値が True または False のどちらか一方であることを示す
Country String	属性値が印刷可能な 2 文字に制限されることを示す (fr など)
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)	属性値が大文字と小文字を区別しないことを示す
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。必ず GMT を使用すること
IA5String (以前は ces)	属性値が大文字と小文字を区別することを示す
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　属性構文の定義 (続き)

構文および OID	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* 内のバックスラッシュとドル記号は、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。多くのサーバで、postal address は最大 30 文字の 6 行に制限されている。たとえば、次のように指定する `1234 Main St.$Anytown, TX 12345$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	この属性値が、http://、https://、ftp、LDAP などの文字列で始まる URL 形式であることを示す。URI は IA5String と同じである。RFC 2396 を参照

属性の作成

Directory Server Console を使用して、新しい属性を作成できます。新しい属性は、スキーマに追加したあとで、新規に作成したオブジェクトクラスに含める必要があります。詳細は、340 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。

新しい属性を作成するには、次の手順を実行します。

1. 「属性」タブを表示します。

   ここまでの手順は、334 ページの「属性の表示」で説明されています。

2. 「作成」をクリックします。

   「属性の作成」ダイアログボックスが表示されます。

3. 「属性名」テキストボックスに属性の一意な名前を入力します。

4. 「属性 OID (省略可能)」テキストボックスに属性のオブジェクト識別子を入力します。

   OID については、334 ページの表 9-1 を参照してください。

5. 「構文」ドロップダウンメニューから、属性に保持させるデータを記述するための構文を選択します。

   使用可能な構文は、334 ページの表 9-1 に記載されています。

6. 属性に複数の値を設定できるようにする場合は、「複数値」チェックボックスを選択します。

   Directory Server では、1 エントリに対して、複数値属性のインスタンスを複数指定できます。

7. 「OK」をクリックします。

属性の編集

編集できる属性は、ユーザが作成した属性だけです。標準の属性は編集できません。

属性を編集するには、次の手順を実行します。

1. 「属性」タブを表示します。

   ここまでの手順は、334 ページの「属性の表示」で説明されています。

2. 「ユーザ定義属性」テーブルで、編集する属性を選択し、「編集」をクリックします。

   「属性の編集」ダイアログボックスが表示されます。

3. 属性の名前を変更するには、「属性名」テキストボックスに新しい名前を入力します。

4. 属性のオブジェクト識別子を変更するには、「属性 OID (省略可能)」テキストボックスに新しい識別子を入力します。

   OID については、334 ページの表 9-1 を参照してください。

5. 属性に保持させるデータを記述する構文を変更するには、「構文」ドロップダウンメニューから新しい構文を選択します。

6. 使用可能な構文は、334 ページの表 9-1 に記載されています。

7. 属性に複数の値を設定できるようにする場合は、「複数値」チェックボックスを選択します。

   Directory Server では、1 エントリにつき複数値属性のインスタンスを複数指定できます。

8. 属性の編集を終えたら、「OK」をクリックします。

属性の削除

削除できる属性はユーザが作成した属性だけです。標準の属性は削除できません。

属性を削除するには、次の手順を実行します。

1. 「属性」タブを表示します。

   ここまでの手順は、334 ページの「属性の表示」で説明されています。

2. 「ユーザ定義属性」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、削除のボタンをクリックします。

   この結果、ただちに属性が削除されます。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の管理

Directory Server Console を使用して、スキーマのオブジェクトクラスを管理できます。Console では、スキーマの全オブジェクトクラスの表示のほか、スキーマへのオブジェクトクラス拡張の作成、編集、および削除ができます。次の節では、オブジェクトクラスの管理方法を説明しています。

- 「オブジェクトクラスの表示」(338 ページ)
- 「オブジェクトクラスの作成」(340 ページ)
- 「オブジェクトクラスの編集」(341 ページ)
- 「オブジェクトクラスの削除」(342 ページ)

属性の管理については、334 ページの「属性の管理」を参照してください。

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あるすべてのオブジェクトクラスに対して、その内容を表示するには、次の手順を実行します。

1. 「Directory Server Console」で、「構成」タブを選択します。
2. ナビゲーションツリーで、Schema フォルダを選択し、右側の区画で「オブジェクトクラス」タブを選択します。
3. 「オブジェクトクラス」リストで、内容を表示するオブジェクトクラスを選択します。

   タブのほかのフィールドには、選択した (標準またはユーザ定義の) オブジェクトクラスに関する情報が表示されます。

次の表に、「オブジェクトクラス」タブのフィールドを示します。

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
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。たとえば、`inetOrgPerson` オブジェクトクラスの親オブジェクトは、`organizationalPerson` オブジェクトである。これは、`inetOrgPerson` オブジェクトクラスを持つエントリは、`organizationalPerson` オブジェクトクラスから必須の属性および許可された属性を自動的に継承することを示す 一般に、ユーザ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る。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` になる。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` になる
OID	オブジェクトクラスのオブジェクト識別子 OID は通常、ピリオドで区切られた 10 進数からなる文字列。オブジェクトクラスや属性などのオブジェクトを一意に識別する。OID を指定しないと、Directory Server は自動的に *ObjectClass_name*`-oid` を使用する。たとえば、OID を指定しないで `division` というオブジェクトクラスを作成した場合、Directory Server は自動的に `division-oid` という OID を使用する OID に関する詳細情報や、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/iana/` にアクセスすること
オブジェクトクラス	このリストには、Directory Server スキーマ内にあるすべての (標準およびユーザ定義の) オブジェクトクラスが含まれている
必須の属性	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
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

オブジェクトクラスの作成

オブジェクトクラスを作成するには、まず一意となる名前を指定し、次にその新しいオブジェクトクラスの親オブジェクトを選択してから、必須の属性および省略可能な属性を追加します。

オブジェクトクラスを作成するには、次の手順を実行します。

1. 「オブジェクトクラス」タブを表示します。

   ここまでの手順は、338 ページの「オブジェクトクラスの表示」で説明されています。

2. 「オブジェクトクラス」タブで「作成」をクリックします。

   「オブジェクトクラスの作成」ダイアログボックスが表示されます。

3. 「名前」テキストボックスにオブジェクトクラスの一意となる名前を入力します。

4. 「OID」テキストボックスに、新しいオブジェクトクラスのオブジェクト識別子を入力します (省略可能)。

   OID については、339 ページの表 9-3 を参照してください。

5. 「親」ドロップダウンメニューから、オブジェクトクラスの親オブジェクトを選択します。

   既存のすべてのオブジェクトクラスから選択できます。親オブジェクトクラスについては、339 ページの表 9-3 を参照してください。

6. 新しいオブジェクトクラスを使用するエントリ内に必ず存在する必要がある属性を追加するには、「使用可能な属性」リストで属性を強調表示し、「必須の属性」ボックスの左にある「追加」ボタンをクリックします。

   標準の属性を使用することも、新しい属性を作成することもできます。詳細は、334 ページの「属性の管理」を参照してください。

7. 新しいオブジェクトクラスを使用するエントリ内に存在することができる属性を追加するには、「使用可能な属性」リストで属性を強調表示し、「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。

8. 以前に追加した属性を削除するには、「必須の属性」リストまたは「許可された属性」リスト内の属性を強調表示し、「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許可された属性および必須の属性は、どちらも削除できません。

9. 指定したオブジェクトクラス定義に問題がない場合は、「OK」をクリックしてダイアログボックスを閉じます。

オブジェクトクラスの編集

Directory Server Console を使用して、作成済みのオブジェクトクラスを編集できます。ただし、標準のオブジェクトクラスは編集できません。

オブジェクトクラスを編集するには、次の手順を実行します。

1. 「オブジェクトクラス」タブを表示します。

   ここまでの手順は、338 ページの「オブジェクトクラスの表示」で説明されています。

2. 「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。

   「オブジェクトクラスの編集」ダイアログボックスが表示されます。

3. オブジェクトクラスの名前を変更するには、「名前」テキストボックスに新しい名前を入力します。

4. オブジェクトクラスに対するオブジェクト識別子を変更するには、「OID」テキストボックスに新しい OID を入力します (省略可能)。

   OID については、339 ページの表 9-3 を参照してください。

5. オブジェクトクラスの親オブジェクトを変更するには、「親」プルダウンメニューから新しい親オブジェクトを選択します。

6. 新しいオブジェクトクラスを使用するエントリ内に存在する必要がある属性を追加するには、「使用可能な属性」リストで属性を強調表示し、「必須の属性」ボックスの左にある「追加」ボタンをクリックします。

   標準の属性を使用することも、新しい属性を作成することもできます。詳細は、334 ページの「属性の管理」を参照してください。

7. 新しいオブジェクトクラスを使用するエントリ内に存在することができる属性を追加するには、「使用可能な属性」リストで属性を強調表示し、「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。

8. 以前に追加した属性を削除するには、「必須の属性」リストまたは「許可された属性」リスト内の属性を強調表示し、「削除」ボタンをクリックします。

   継承による許可された属性および必須の属性は、どちらも削除できません。

9. 指定したオブジェクトクラス定義に問題がない場合は、「OK」をクリックしてダイアログボックスを閉じます。

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が作成したオブジェクトクラスだけです。標準のオブジェクトクラスは削除できません。

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. 「オブジェクトクラス」タブを表示します。

   ここまでの手順は、338 ページの「オブジェクトクラスの表示」で説明されています。

2. 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、削除のボタンをクリックします。

   この結果、ただちにオブジェクトクラスが削除されます。この処理を元に戻すことはできません。

スキーマ検査のオン / オフの切り替え

スキーマ検査をオンにすると、Directory Server によって次のことが検査されます。

- 使用中のオブジェクトクラスおよび属性がディレクトリスキーマに定義されているか
- オブジェクトクラスに必須の属性がエントリに含まれているか
- オブジェクトクラスに許可された属性だけがエントリに含まれているか

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、常にスキーマの検査をオンにしておくべきです。LDAP のインポート処理を高速化するのが、スキーマ検査をオフにしてもよい唯一の場合です。しかし、その場合スキーマに適合しないエントリがインポートされるリスクがあります。したがってこのようなエントリは検索することができません。

スキーマ検査のオン / オフを切り替えるには、次の手順を実行します。

1. 「Directory Server Console」で、「構成」タブを選択します。
2. ナビゲーションツリーの一番上にあるサーバアイコンを強調表示し、右側の区画にある「設定」タブを選択します。
3. スキーマ検査をオンにする場合は「スキーマチェックを有効にする」チェックボックスを選択し、オフにする場合は選択を解除します。
4. 「保存」をクリックします。

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。詳細は、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照してください。

第 10 章

インデックスの管理

『iPlanet Directory Server 導入ガイド』には、インデックス作成の概念、そのコストと利点、および iPlanet Directory Server に実装されているさまざまなインデックスのタイプについて記載されています。この章では、インデックスメカニズムの導入にあたり、まず検索アルゴリズムについて説明し、続けてインデックスの作成、削除、管理方法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。

- インデックスについて
- インデックスの作成
- インデックスの削除
- インデックスの管理
- 属性名のクイックリファレンス

インデックスについて

ここでは、Directory Server におけるインデックス作成の概要について、次の項目ごとに説明します。

インデックスのタイプについて

インデックスは、ディレクトリデータベース内のファイルに格納されます。ファイル名は、ファイルに含まれているインデックスのタイプではなく、インデックス付き属性に基づいて付けられています。特定の属性に複数のインデックスが付けられている場合、インデックスファイルにさまざまなタイプのインデックスが含まれることがあります。たとえば、共通名属性に付いているインデックスはすべて `cn.db3` ファイルに保持されます。

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`)

  実在インデックス (presence index) には、特定の属性を含むエントリのリストが含まれます。たとえば、アクセス制御情報を含むすべてのエントリを調べる場合などは、このインデックスを使用すると便利です。実在インデックスを含む `aci.db3` ファイルを生成すると、`ACI=*` を効率的に検索して、サーバ用のアクセス制御リストを生成することができます。

  実在インデックスは、ベースオブジェクトの検索には使用できません。

- 等価インデックス (`eq`)

  等価インデックス (equality index) では、特定の属性値を含むエントリを効率的に検索することができます。たとえば、cn 属性の等価インデックスを使用すると、`cn=Babs Jensen` をはるかに効率的に検索することができます。

- 近似インデックス (`approx`)

  近似インデックス (approximate index) では、類似または似た音の用語を探すのに有効な近似検索を行うことができます。たとえば、あるエントリに属性値 cn=Robert E Lee が含まれる可能性があるとします。この場合、近似検索を使用して、cn~=Robert Lee、cn~=Robert、または cn~=Lee を検索し、この値を返すことができます。同様に、l~=San Fransisco (スペルミスに注目) を検索すると、l=San Francisco を含むエントリが返されます。

- 部分文字列インデックス (`sub`)

  部分文字列インデックス (substring index) は、維持管理にコストがかかるインデックスですが、エントリ内の部分文字列の検索に効果的です。

  たとえば、次のような形式の検索を実行します。

  ```
  cn=*derson
  ```

  この場合、次のような文字列を含む共通名にマッチします。

  ```
  Bill Anderson
  Jill Anderson
  Steve Sanderson
  ```

  同様に、次のエントリを検索します。

```
telephonenumber= *555*
```

この場合、ディレクトリのエントリから、555 を含む電話番号がすべて返されます。

注 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。

- 国際化インデックス

  国際化インデックス (international index) を使用すると、国際化ディレクトリにある情報の検索を高速化することができます。国際化インデックスの作成手順は、通常のインデックス作成手順と似ていますが、属性がインデックス化されるために属性とロケール (locale) (OID) を関連付けて、マッチング規則 (matching rule) を適用します。

  サポートされるロケールと関連付けられた OID のリストについては、付録 D「国際化」を参照してください。追加のマッチング規則 (matching rule) が使用できるように Directory Server を構成する場合には、iPlanet プロフェッショナルサービスにお問い合わせください。

- ブラウズ (仮想リスト表示) インデックス

  ブラウズインデックス (browsing index) (別名、仮想リスト表示インデックス (virtual list view index)) を使用すると、Directory Server Console でエントリの表示を高速化することができます。`ou=people` 分岐などのように、ディレクトリの分岐に数百ものエントリが含まれている場合には、特にこのインデックスが有効です。表示性能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することができます。この操作は、Directory Server Console または `/usr/sbin/directoryserver vlvindex` コマンドを使用して行います。

デフォルトインデックス、システムインデックス、および標準インデックスについて

Directory Server をインストールすると、データベースインスタンスごとに、デフォルトインデックスとシステムインデックス (system index) のセットが 1 組作成されます。これらのインデックスを維持するために、ディレクトリは標準インデックス (standard index) を使用します。

デフォルトインデックスの概要

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 (default index) を修正することができますが、インデックスを削除する前に、企業内のサーバプラグインやその他のサーバが、このインデックスに依存していないことを確認する必要があります。

次の表に、ディレクトリとともにインストールされるデフォルトインデックスのリストを示します。

表 10-1 デフォルトインデックス

属性	等価	属性	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ディレクトリ検索の性能を向上させる
givenName	X	X	X	もっとも一般的なタイプのユーザディレクトリ検索の性能を向上させる
mail	X	X	X	もっとも一般的なタイプのユーザディレクトリ検索の性能を向上させる
mailHost	X			iPlanet Messaging Server で使用される
member	X			iPlanet サーバの性能を向上させる。このインデックスは、参照整合性検査プラグインでも使用される。詳細については、66 ページの「参照整合性の管理」を参照
owner	X			iPlanet サーバの性能を向上させる。このインデックスは、参照整合性検査プラグインでも使用される。詳細については、『iPlanet Directory Server 管理者ガイド』を参照
seeAlso	X			iPlanet サーバの性能を向上させる。このインデックスは、参照整合性検査プラグインでも使用される。詳細については、66 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ディレクトリ検索の性能を向上させる
telephoneNumber	X	X	X	もっとも一般的なタイプのユーザディレクトリ検索の性能を向上させる
uid	X			iPlanet サーバの性能を向上させる

表 10-1　デフォルトインデックス (続き)

属性	等価	属性	部分文字列	目的
uniquemember	X			iPlanet サーバの性能を向上させる。このインデックスは、参照整合性検査プラグインでも使用される。詳細については、66 ページの「参照整合性の管理」を参照

システムインデックスの概要

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。ディレクトリを適切に機能させるためには、システムインデックスが必要です。次の表に、ディレクトリとともにインストールされるシステムインデックスのリストを示します。

表 10-2　システムインデックス

属性	等価	属性	目的
aci		X	Directory Server が、データベース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
dnComp	X		ディレクトリのサブツリー検索を高速化するために使用される
objectClass	X		ディレクトリのサブツリー検索を高速化するために使用される
entryDN	X		DN 検索に基づくエントリの取得を高速化する
parentID	X		1 レベル検索におけるディレクトリの性能を強化する
numSubordinates		X	「ディレクトリ」タブの表示性能を強化するために、Directory Server Console で使用される
nsUniqueID	X		特定のエントリの検索に使用される

標準インデックスの概要

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。これらのファイルを自分で生成する必要はありません。

- id2entry.db3 : 実際のディレクトリデータベースのエントリが含まれる。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

- 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
- 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
- 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

検索アルゴリズムの概要

インデックスは、検索を高速化するために使用されます。検索アルゴリズムを理解していると、ディレクトリでどのようにインデックスが使用されるかを理解する上で役立ちます。各インデックスには、cn、共通名、属性などの属性リストと、それぞれの値に対応するエントリへのポインタが含まれます。Directory Server では、次のように検索要求が処理されます。

1. Netscape Communicator や Directory Server Console などの LDAP クライアントアプリケーションから、ディレクトリに検索要求が送信されます。
2. ディレクトリは、受信された要求を調べ、指定されたベース DN が、そのデータベースやデータベースリンクに含まれる接尾辞とマッチするかどうかを確認します。
   - マッチする場合、ディレクトリは要求を処理する
   - マッチしない場合、ディレクトリはクライアントに対し、接尾辞がマッチしないことを表すエラーを返す。cn=config の nsslapd-referral 属性でレフェラルが指定されている場合、ディレクトリはそのリクエストの続行をクライアントが試みることができる LDAP URL も返す
3. 各データベース属性に対する検索要求が単一のインデックスによって満たされた場合は、サーバはこのインデックスを読み込み、マッチする可能性のあるエントリのリストを生成します。属性に対するインデックスがない場合、ディレクトリはデータベースにあるエントリをすべて含めて候補エントリのリストを生成するため、検索速度は大幅に低下します (サーバが使用するインデックスキー (index key) 用に、All ID のトークンが設定されている場合は、ディレクトリでも同じ処理が行われます。All ID については、367 ページの「インデックスの管理」を参照してください)。

   検索要求に複数の属性が含まれている場合、ディレクトリは複数のインデックスに問い合わせ、候補となるエントリの結果リストをバインドします。
4. 属性に対するインデックスが存在する場合、ディレクトリは、一連のエントリ ID 番号の形式で、インデックスファイルからマッチする候補を取得します。

5. ディレクトリは、返されたエントリ ID 番号を使用して、`id2entry.db3` ファイルから対応するエントリを読み込みます。次に、Directory Server が候補エントリを 1 つずつ調べ、検索条件とマッチするものがあるかどうかを確認します。マッチするエントリが見つかるたびに、ディレクトリはそのエントリを返します。

   候補エントリがすべて調べられるか、次の属性の 1 つで設定されている制限に達するまで、ディレクトリは検索を続行します。

   - `nsSizeLimit`：検索操作から返されるエントリ数の最大値を指定する。この制限値に達すると、ディレクトリはそれまでに見つかった、検索要求とマッチするすべてのエントリとともに、制限サイズを超えたことを示すエラーを返す
   - `nsTimeLimit`：検索要求に対して割り当てる最大時間を、秒単位で指定する。この制限値に達すると、ディレクトリはそれまでに見つかった、検索要求とマッチするすべてのエントリとともに、制限時間を超えたことを示すエラーを返す
   - `nsLookthroughLimit`：検索要求に応答して候補エントリを調べるときにチェックするエントリ数の最大値を指定する

これらの属性については、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照してください。

また、ディレクトリでは、メタフォン音声アルゴリズムのバリエーションを使用して、近似インデックスの検索を行います。各値は一連の単語として扱われ、各単語について音声コードが生成されます。

注 iPlanet Directory Server 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用してください。

近似検索に入力された値も同様に、一連の音声コードに変換されます。次の条件の両方が満たされる場合は、エントリは照会にマッチするとみなされます。

- すべての照会文字列が、エントリ文字列で生成されたコードとマッチする
- すべての照会文字列が、エントリ文字列コードと同じ順序で並べられている

たとえば、次の表は、音声コードが `ALS B SRT` であるエントリ名 `Alice B. Sarette` と複数の照会をマッチさせる方法を示しています。

表 10-3 音を使用した近似検索

照会文字列	音声コード	マッチ / コメント
Alice Sarette	ALS SRT	マッチ。コードは正しい順序で指定されている
Alice Sarrette	ALS SRT	マッチ。Sarette のスペルは間違っているが、コードの指定順序は正しい

表 10-3　音を使用した近似検索 (続き)

照会文字列	音声コード	マッチ / コメント
Surette	SRT	マッチ。Sarette のスペルは間違っているが、生成されたコードが元の名前にある
Bertha Sarette	BR0 SRT	マッチしない。比較される名前に BR0 はない
Sarette, Alice	SRT ALS	マッチしない。コードの指定順序が正しくない

インデックス付けの利点とコストの比較

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
- 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。暗号化されたデータを含むパスワードや、写真を格納するための属性などのように、値が大きくなる場合は、等価インデックスも避ける必要がある
- 検索であまり使用されない属性のインデックスを保持しても、負荷が高くなるだけで、グローバル検索の性能は改善されない
- 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、検索の性能が大幅に低下する場合がある
- 保持するインデックスの数が多くなるほど、必要となるディスク容量も増える

次の例は、どのような場合にインデックスの時間がかかるかを示しています。次のような手順で特定の属性を作成していると想定します。

1. Directory Server は、追加または修正操作を受け取る
2. Directory Server はインデックスが付いている属性を調べて、この属性値に対するインデックスが維持されているかどうかを特定する
3. 作成された属性値にインデックスが付いている場合、Directory Server は新しいインデックスエントリを生成する
4. サーバによるインデックスの生成が完了すると、クライアントの要求に応じて、実際の属性値が作成される

たとえば、Directory Server から次のようなエントリの追加が要求されたとします。

```
dn: cn=Bill Pumice, ou=People, o=siroe.com
objectclass: top
objectClass:person
objectClass: orgperson
objectClass: inetorgperson
cn: Bill Pumice
cn: Bill
sn: Pumice
ou: Manufacturing
ou: people
telephonenumber: 408 555 8834
description: Manufacturing lead for the Z238 line.
```

さらに、Directory Server で次のインデックスを維持しているとします。

- 共通名属性と姓属性に対する、等価インデックス、近似インデックス、および部分文字列インデックス
- 電話番号属性に対する、等価インデックスと部分文字列インデックス
- 説明属性に対する、部分文字列インデックス

この場合、このエントリをディレクトリに追加するためには、Directory Server で次の処理を実行する必要があります。

1. 「Bill」と「Bill Pumice」用に共通名の等価インデックスエントリを作成します。
2. 「Bill」と「Bill Pumice」用に共通名の近似インデックスエントリを作成します。
3. 「Bill」と「Bill Pumice」用に共通名の部分文字列インデックスエントリを作成します。
4. 「Pumice」用に、姓の等価インデックスエントリを作成します。
5. 「Pumice」用に、適切な姓の近似インデックスエントリを作成します。
6. 「Pumice」用に、適切な姓の部分文字列インデックスエントリを作成します。
7. 「408 555 8834」用に、電話番号の等価インデックスエントリを作成します。
8. 「408 555 8834」用に、適切な電話番号の部分文字列インデックスエントリを作成します。
9. 「Manufacturing lead for the Z238 line of widgets」用に適切な説明の部分文字列インデックスエントリを作成します。この文字列用には、大量の部分文字列エントリが生成されます。

この例のように、インデックスの作成にはコストがかかります。

インデックスの作成

ここでは、Directory Server Console とコマンド行を使用して、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の作成方法を説明します。またブラウズインデックスを作成する別の手順についても説明します。

注 iPlanet Directory Server 5.1 は、単一データベース環境とマルチデータベース環境のどちらでも動作します。別のデータベースでは、新しいインデックスは自動的に作成されないので、すべてのデータベースインスタンスに新しいインデックスを必ず作成する必要があります。

ただし、デフォルトインデックスは後続のデータベースインスタンスに自動的に作成および維持されますが、既存のデータベースインスタンスには追加されません。言い換えれば、後続のデータベースでは、一番最後に作成されたデフォルトインデックスが使用されます。これは、2 番目のデータベースインスタンスにデフォルトインデックスを追加した場合、このインデックスは最初のデータベースインスタンスに維持されるのではなく、後続のインスタンスに維持されることを意味します。

ここでは、次の手順について説明します。

- Server Console を使用したインデックスの作成
- コマンド行からのインデックスの作成
- Server Console を使用したブラウズインデックスの作成
- コマンド行からのブラウズインデックスの作成

Server Console を使用したインデックスの作成

Directory Server Console を使用して、特定の属性用に実在インデックス、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することができます。

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 Console で、「構成」タブを選択します。
2. 「データ」ノードを展開し、インデックスを作成するデータベースの接尾辞を展開して、データベースを選択します。
3. 右側の区画で「インデックス」タブを選択します。

注 「データベースの設定」ノードはクリックしないでください。このノードをクリックすると、データベースごとにインデックスを構成するためのウィンドウではなく、「デフォルトインデックスの設定」ウィンドウが表示されてしまいます。

4. 「追加インデックス」テーブルにインデックスを作成する属性がリストされている場合は、手順 6 に進みます。それ以外の場合は、「属性の追加」をクリックします。

   ダイアログボックスが開き、サーバスキーマにある使用可能な属性のリストが表示されます。

5. インデックスを作成する属性を選択し、「OK」をクリックします。

   選択した属性が「追加インデックス」テーブルに追加されます。

6. 各属性について、保持するインデックスのタイプに対応するチェックボックスを選択します。

7. 英語以外の言語のインデックスを作成する場合は、「マッチング規則」フィールドで使用する照合順序 (collation order) の OID を入力します。

   複数の OID をコンマ (スペースではなく) で区切って指定することにより、属性に複数の言語を使用したインデックスを付けることができます。言語とその OID のリスト、および照合順序に関する詳しい説明については、付録 D「国際化」を参照してください。

8. 「保存」をクリックします。

   「インデックス」ダイアログボックスが表示されます。このダイアログボックスには、インデックスの作成状態と作成日時が表示されます。作成したインデックスの状態を表示するには、「状態ロゴ」ボックスをクリックします。インデックスの作成が完了したら、「閉じる」をクリックして、「インデックス」ダイアログボックスを閉じます。

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すぐに有効になります。サーバ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作成

コマンド行を使用して、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することができます。

コマンド行を使用したインデックスの作成は、次の 2 つのステップで行われます。

- `ldapmodify` コマンド行ユーティリティを使用して、新しいインデックスエントリを追加するか、または既存のインデックスエントリを編集する

- /usr/sbin/directoryserver db2index-task コマンドを実行して、サーバに保持される新しいインデックスのセットを生成する

注	システムインデックスは、Directory Server にハードコードされているので、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。

次の節では、インデックスの作成に必要な手順について説明します。

インデックスエントリの追加

ldapmodify を使用して、使用しているディレクトリに新しいインデックス属性を追加します。デフォルトインデックスの 1 つとなる新しいインデックスを作成する場合は、新しいインデックス属性を cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config エントリに追加します。

特定のデータベース用に新しいインデックスを作成するには、インデックスを cn=index,cn=*instanceName*,cn=ldbm database,cn=plugins,cn=config エントリに追加します。ここで、cn=*instanceName* はデータベース名を示します。

注	dse.ldif ファイルの cn=config エントリの下には、エントリを作成しないようにしてください。単純で平面的な dse.ldif 構成ファイルの cn=config エントリは、通常のエントリと同じように拡張性が高いデータベースには格納されません。その結果、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、性能が低下します。 性能上の理由から、単純なユーザエントリを cn=config の下に格納することはお勧めできませんが、ディレクトリマネージャまたはレプリケーションマネージャ (サプライヤバインド DN) エントリなどの特別なユーザエントリを cn=config, の下に格納すると、構成情報を一元化できるため便利です。

エントリを追加するために必要な LDIF 更新文については、57 ページの「LDIF 更新文」を参照してください。

たとえば、Siroe1 データベースに、sn (姓) 属性の実在インデックス、等価インデックス、および部分文字列インデックスを作成するとします。

次のように ldapmodify コマンド行ユーティリティを実行します。

```
ldapmodify -a -h server.siroe.com -p 389 \
           -D "cn=Directory Manager" -w password
```

ldapmodify ユーティリティはサーバにバインドし、構成ファイルにエントリを追加する準備を行います。ldapmodify コマンド行ユーティリティについては、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照してください。

次に、新規インデックスに次のエントリを追加します。

```
dn: cn=sn,cn=index,cn=Siroe1,cn=ldbm database,cn=plugins,cn=config
objectClass:top
objectClass:nsIndex
cn:sn
nsSystemIndex:false
nsIndexType:pres
nsIndexType:eq
nsIndexType:sub
nsMatchingRule:2.16.840.1.113730.3.3.2.3.1
```

cn 属性には、インデックスを作成する属性名 (この例では、sn 属性) が含まれます。エントリは、nsIndex オブジェクトクラスのメンバーです。nsSystemIndex 属性は false で、Directory Server 操作にインデックスがなくてもかまわないことを示します。複数値 nsIndexType 属性には、実在 (pres) インデックス、等価 (eq) インデックス、および部分文字列 (sub) インデックスが指定されています。キーワードは、別々の行に入力する必要があります。nsMatchingRule 属性は、OID がブルガリア語の照会順序で指定されていることを示します。

インデックスエントリを指定し、nsIndexType 属性には何も指定しなかった場合、国際化インデックスを除くすべてのインデックスが、指定された属性で保持されます。たとえば、上記の例の代わりに、新しい sn インデックスに対して、次のエントリを指定したとします。

```
dn: cn=sn,cn=index,cn=instance,cn=ldbm database,cn=plugins,cn=config
objectClass:top
objectClass:nsIndex
cn:sn
nsSystemIndex:false
nsIndexType:
```

この新しいエントリによって、sn (姓) 属性用にすべてのインデックスが作成されます。

この属性用にインデックスを保持しないことを指定するには、nsIndexType 属性でキーワード none を使用します。たとえば、Siroe1 データベースで今作成した sn インデックスを一時的に無効にするとします。次のように nsIndexType を none に変更します。

```
dn: cn=sn,cn=index,cn=Siroe1,cn=ldbm database,cn=plugins,cn=config
objectClass:top
objectClass:nsIndex
cn:sn
nsSystemIndex:false
nsIndexType:none
```

照会順序と OID のリストについては、付録 D「国際化」を参照してください。

インデックスの構成属性については、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』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、プライマリ名を常に使用する必要があります。属性のプライマリ名は、スキーマでその属性にリストされた最初の名前です。たとえば、userid 属性では uid がプライマリ名になります。属性のプライマリ名とエイリアス名のリストについては、373 ページの表 10-6 を参照してください。

db2index-task コマンドの実行

インデックスが付いたエントリを作成するか、既存のインデックスエントリにインデックスのタイプを追加したら、/usr/sbin/directoryserver db2index-task コマンドを実行して、Directory Server で維持される新しいインデックスセットを生成します。コマンドを実行すると、使用するディレクトリに追加した新規データおよびディレクトリ内の既存データのすべてで、新しいインデックスセットが有効になります。

このコマンドは、次のとおりです。

```
# /usr/sbin/directoryserver db2index-task
```

次の例では、インデックスを作成します。

```
#!/bin/sh
/usr/sbin/directoryserver db2index-task \
  -D "cn=Directory Manager" -w password -n Database1 -t sn
```

表 10-4 例で使用した db2index-task オプションの説明

オプション	内容
-D	ディレクトリマネージャの DN を指定する
-w	ディレクトリマネージャのパスワードを指定する
-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	インデックスを作成するデータベース上の、属性の名前を指定する

Server Console を使用したブラウズインデックスの作成

Directory Server Console を作成して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 Console で「ディレクトリ」タブを選択します。
2. 左側のナビゲーションツリーで、インデックスの作成対象のエントリ (`People` など) を選択し、「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。

   ナビゲーションツリーで、インデックスの作成対象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作成」を選択します。
3. 「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。作成したインデックスの状態を表示するには、「状態ログ」ボックスをクリックします。
4. 「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。

   ディレクトリに追加された新しいデータで、新しいインデックスがすぐに有効になります。サーバを再起動する必要はありません。

コマンド行からのブラウズインデックスの作成

コマンド行を使用し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 つのステップで行われます。

- `ldapmodify` を使用して、新しいブラウズインデックスエントリを追加するか、または既存のブラウズインデックスエントリを編集する
- `/usr/sbin/directoryserver vlvindex` コマンドを実行して、サーバ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する

次の節では、ブラウズインデックスの作成に必要な手順について説明します。

ブラウズインデックスエントリの追加

作成するブラウズインデックスエントリのタイプは、高速化を望む `ldapsearch` の属性ソートのタイプによって異なります。次の項目を考慮することが重要です。

- 検索の範囲 (ベース、1 レベル、サブ)。
- 検索のベース (検索の開始点として使用されるエントリ)。

- ソート対象の属性
- 検索フィルタ。検索用フィルタの指定については、付録 B「ディレクトリエントリの検索」を参照
- 検索のベースの形成するエントリが属している ldbm データベース

注 ブラウズインデックスを作成できるのは、ldbm データベースだけです。

たとえば、ブラウズインデックスを作成して、Siroe1 データベース内のエントリ "dc=siroe,dc=com" で ldapsearch を高速化するとします。ここで、検索ベースには "dc=siroe,dc=com"、検索フィルタには (|(objectclass=*)(objectclass=ldapsubentry))、範囲には one、返される属性のソート順には cn、givenname、o、ou、および sn を指定します。

次のように ldapmodify コマンド行ユーティリティを実行します。

```
ldapmodify -a -h server -p 389 -D "cn=directory manager" -w password
```

ldapmodify ユーティリティはサーバにバインドし、構成ファイルにエントリを追加する準備を行います。

次に、ブラウズインデックスを定義するブラウズインデックスエントリを 2 つ追加します。

最初のエントリでは、ブラウズインデックスのベース、範囲、フィルタを指定します。

```
dn: cn="dc=siroe,dc=com",cn=Siroe1,cn=ldbm
database,cn=plugins,cn=config
objectClass:top
objectClass:vlvSearch
cn:"dc=siroe,dc=com"
vlvbase:"dc=siroe,dc=com"
vlvscope:one
vlvfilter:(|(objectclass=*)(objectclass=ldapsubentry))
```

cn にはブラウズインデックスを作成するエントリを示す、ブラウズインデックスの識別子が含まれます。この例では、"dc=siroe,dc=com" エントリです。ブラウズインデックス識別子には、エントリに dn を使用することをお勧めします。これは Directory Server Console で採用されたアプローチで、これによって同じブラウズインデックスが複数作成されることを回避することができます。エントリは、vlvSearch オブジェクトクラスのメンバーです。vlvbase 属性値は、ブラウズインデックスの作成先エントリを示します。この例では、"dc=siroe,dc=com" エントリ、つまりブラウズインデックス識別子になります。vlvscope 属性は one で、これは高速にする検

索のベースが one (1 レベル) であることを示します。one という検索ベースは、エントリ自体ではなく、cn 属性で指定されたエントリすぐ下の子だけが検索されることを意味します。vlvfilter は検索で使用されるフィルタを示します。この例では、(|(objectclass=*)(objectclass=ldapsubentry)) です。

2 番目 のエントリは、返される属性のソート順を示します。

```
dn:cn=sort_cn_givenname_o_ou_sn,cn="dc=siroe,dc=com",cn=Siroe1,
cn=ldbm database,cn=plugins,cn=config
objectClass:top
objectClass:vlvIndex
cn:cn=sort_cn_givenname_o_ou_sn
vlvsort:cn givenname o ou sn
```

cn には、ブラウズインデックスのソート識別子が含まれます。作成したブラウズインデックスの検索ソート順を明確に示すソート識別子を使用することをお勧めします。この例では、明示的なソート識別子 cn=sort_cn_givenname_o_ou_sn が使用されています。エントリは、vlvIndex オブジェクトクラスのメンバーです。vlvsort 属性値は好みの属性のソート順を示します。上記の例では、cn、givenname、o、ou、sn の順にソートされます。

注	この最初のブラウズインデックスエントリは、cn=*instanceName*,cn=ldbm database,cn=plugins,cn=config ディレクトリのノードに追加する必要があります。また、2 番目のエントリは最初のエントリの子になるようにする必要があります。

vlvindex コマンドの実行

2 つのブラウズインデックスエントリを作成するか、既存のブラウズインデックスエントリに追加の属性タイプを追加したあと、/usr/sbin/directoryserver vlvindex コマンドを実行して、Directory Server で維持される新しいブラウズインデックスセットを生成します。スクリプトを実行すると、使用するディレクトリに追加された新規データおよびディレクトリ内の既存データのすべてで、新しいブラウズインデックスセットが有効になります。

```
# /usr/sbin/directoryserver vlvindex
```

次の例では、vlvindex コマンドを使用してブラウズインデックスが生成されます。

```
# /usr/sbin/directoryserver vlvindex -n Database1 -T \
"dc=siroe,dc=com"
```

表 10-5　例で使用した 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	ブラウズインデックスの作成時に使用するブラウズインデックス識別子を指定します

インデックスの削除

ここでは、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、国際化インデックス、およびブラウズインデックスを削除する方法を説明します。

注　iPlanet Directory Server 5.1 は、シングルデータベース環境、マルチデータベース環境のどちらでも動作するので、すべてのデータベースインスタンスから不要なインデックスをすべて削除する必要があります。

削除するデフォルトインデックスは、既存のデータベースインスタンスにある以前のインデックスセットから削除されません。

ブラウズインデックスの削除手順は異なるので、別の節で説明します。ここでは、次の手順について説明します。

- Server Console を使用したインデックスの削除
- コマンド行からのインデックスの削除
- Server Console を使用したブラウズインデックスの削除
- コマンド行からのブラウズインデックスの削除

警告　システムインデックスを削除すると、Directory Server の性能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=instance,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。

システムインデックスとデフォルトインデックスについては、『iPlanet Directory Server 導入ガイド』を参照してください。

Server Console を使用したインデックスの削除

Directory Server Console を使用して、作成したインデックス、Messaging Server や Calendar Server などほかの iPlanet サーバが使用しているインデックス、およびデフォルトインデックスを削除できます。システムインデックスは削除できません。

Directory Server Console を使用して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 Console で、「構成」タブを選択します。
2. 「データ」ノードを展開し、インデックスが含まれるデータベースに関連付けられた接尾辞を展開します。削除するインデックスを含むデータベースを選択します。
3. 削除するインデックスを含む属性を特定します。インデックスの下にあるチェックボックスの選択を解除します。

   特定の属性に保持されているすべてのインデックスを削除する場合は、「属姓名」の下にある属性セルを選択し、「属性の削除」をクリックします。
4. 「保存」をクリックします。

   「インデックスの削除」警告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、インデックスを削除します。
5. 「ブラウズインデックスの削除」ダイアログボックスが開き、インデックス削除の状態が表示されます。削除したインデックスの状態を表示するには、「状態ロゴ」ボタンをクリックします。インデックスの削除が完了したら、「閉じる」をクリックして、「ブラウズインデックスの削除」ダイアログボックスを閉じます。

コマンド行からのインデックスの削除

次のように `ldapdelete` コマンド行ユーティリティを使用してインデックスを削除できます。

- `ldapdelete` コマンド行ユーティリティを使用して、インデックスエントリ全体を削除するか、または既存のインデックスエントリから不要なインデックスのタイプを削除する
- `/usr/sbin/directoryserver db2index-task` コマンドを使用して残りのインデックスがサーバによって維持されるようにする

次の節では、インデックスの削除に必要な手順について説明します。

インデックスエントリの削除

インデックスエントリ全体、または既存のエントリから不要なインデックスのタイプを削除するには、ldapdelete コマンド行ユーティリティを使用します。

特定のデータベースのインデックスを削除するには、cn=index,cn=*instanceName*,cn=ldbm database,cn=plugins,cn=config エントリからインデックスエントリを削除します。ここで、cn=*instanceName* はデータベース名を示します。

デフォルトインデックスを削除するには、このインデックスを cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config エントリから削除します。

たとえば、Siroe1 データベースで、sn (姓) 属性の実在インデックス、等価インデックス、および部分文字列インデックスを削除するとします。

この場合、次のエントリを削除します。

```
dn: cn=sn,cn=index,cn=Siroe1,cn=ldbm database,cn=plugins,cn=config
objectClass:top
objectClass:nsIndex
cn:sn
nsSystemIndex:false
nsIndexType:pres
nsIndexType:eq
nsIndexType:sub
nsMatchingRule:2.16.840.1.113730.3.3.2.3.1
```

次に、ldapdelete を実行します。

```
ldapdelete -h server.siroe.com -p 389 \
  -D "cn=Directory Manager" -w password \
  "cn=sn,cn=index,cn=Siroe1,dn=ldbm database,cn=plugins,dn=config"
```

このエントリを削除すると、sn 属性の実在インデックス、等価インデックス、部分文字列インデックスは、Siroe1 データベースでは維持されなくなります。

残りのインデックスの再生成

インデックスエントリを削除またはインデックスエントリからいくつかのインデックスタイプを削除した場合、Directory Server によって維持されるように残りのインデックスセットを再生成する必要があります。

インデックスを再生成するには、358 ページの「db2index-task コマンドの実行」で説明されている手順に従います。コマンドを実行すると、使用するディレクトリに追加した新規データおよびディレクトリ内の既存データのすべてで、新しいインデックスセットが有効になります。

Server Console を使用したブラウズインデックスの削除

Directory Server Console を使用して、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 Console で、「データベース」タブを選択します。
2. ナビゲーションツリーで、インデックスの削除対象のエントリ (`People` など) を選択し、「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。また、ナビゲーションツリーで削除する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「ブラウズインデックスの削除」を選択します。
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、インデックスを削除します。
4. 「ブラウズインデックスの削除」ダイアログボックスが開き、インデックス削除の状態が表示されます。

コマンド行からのブラウズインデックスの削除

コマンド行を使用したブラウズインデックス、または仮想リスト表示 (VLV) の削除は、次の 2 つのステップで行われます。

- `ldapdelete` コマンド行ユーティリティを使用して、ブラウズインデックスエントリを削除するか、または既存のブラウズインデックスエントリを編集する
- `/usr/sbin/directoryserver vlvindex` を実行して残りのインデックスを再生成する

次の節では、ブラウズインデックスの削除に必要な手順について説明します。

ブラウズインデックスエントリの削除

ブラウズインデックスエントリを削除したり、既存のブラウズインデックスエントリを編集するには、`ldapdelete` コマンド行ユーティリティを使用します。

特定のデータベースのブラウズインデックスを削除するには、`cn=index,cn=`*`instanceName`*`,cn=ldbm database,cn=plugins,cn=config` エントリからブラウズインデックスエントリを削除します。ここで、`cn=`*`instanceName`* はデータベース名を示します。

たとえば、Siroe1 データベース内のエントリ dc=siroe,dc=com で ldapsearch の操作を高速化するブラウズインデックスを削除するとします。ここで、検索ベースには dc=siroe,dc=com、検索フィルタには (|(objectclass=*)(objectclass=ldapsubentry))、範囲には one、返される属性のソート順には cn、givenname、o、ou、および sn となっています。

このブラウズインデックスを削除するには、次の 2 つの対応するブラウズインデックスエントリを削除する必要があります。

```
dn: cn="dc=siroe,dc=com",cn=Siroe1,cn=ldbm
database,cn=plugins,cn=config
objectClass:top
objectClass:vlvSearch
cn:"dc=siroe,dc=com"
vlvbase:"dc=siroe,dc=com
vlvscope:one
vlvfilter:(|(objectclass=*)(objectclass=ldapsubentry))
```

および

```
dn:cn=sort_cn_givenname_o_ou_sn,cn="dc=siroe,dc=com",cn=Siroe1,
cn=ldbm database,cn=plugins,cn=config
objectClass:top
objectClass:vlvIndex
cn:cn=sort_cn_givenname_o_ou_sn
vlvsort:cn givenname o ou sn
```

次に、ldapdelete を実行します。

```
ldapdelete -h siroe.server.com -p 389 -D "cn=Directory Manager" -w password \
  "cn="dc=siroe,dc=com",cn=Siroe1,cn=ldbm database,cn=plugins,cn=config" \
  "cn=sort_cn_givenname_o_ou_sn,cn="dc=siroe,dc=com",cn=Siroe1, \
cn=ldbm database,cn=plugins,cn=config"
```

検索ベースが dc=siroe,dc=com、検索フィルタが (|(objectclass=*)(objectclass=ldapsubentry))、範囲が one、返される属性のソート順が cn、givenname、o、ou、sn であるこれら 2 つのブラウズインデックスエントリを削除すると、Siroe1 データベースに格納されているエントリ dc=siroe,dc=com の ldapsearch 操作を高速化するためのブラウズインデックスは Siroe1 データベースでは維持されなくなります。

残りのインデックスの再生成

ブラウズインデックスエントリを削除するか、既存のインデックス参照エントリから不要なインデックスのタイプを削除したあと、Directory Server によって維持されるように残りのインデックスセットを再生成する必要があります。

ブラウズインデックスを再生成するには、361 ページの「vlvindex コマンドの実行」で説明されている手順に従います。スクリプトを実行すると、使用するディレクトリに追加した新規データおよびディレクトリ内の既存データのすべてで、新しいインデックスセットが有効になります。

インデックスの管理

ディレクトリが使用するインデックスは、インデックスキーのテーブルとマッチングのエントリ ID リスト (entry ID list) で構成されます。エントリ ID リストは、クライアントアプリケーションの検索要求とマッチする可能性がある候補エントリのリストを構築するために、ディレクトリが使用します (345 ページの「インデックスについて」を参照)。

各エントリ ID リストに、`nsslapd-allidsthreshold` 属性で指定されたサイズ制限が適用されます。このサイズ制限はサーバによって管理されるすべてのインデックスキー全体に適用され、論理的にはすべての ID のしきい値 (All IDs Threshold) と呼ばれます。1 つの ID リストのサイズがこの制限値に達すると、その ID リストがすべての ID のトークンと置き換えられます。

すべての ID のトークンによって、ディレクトリエントリがすべてインデックスキーとマッチしているとサーバが認識します。実際には、すべての ID のトークン (All IDs token) によって、サーバは指定された検索のタイプで利用可能なインデックスが存在しないかのように動作します。ディレクトリは、サーバが検索要求の別の側面によって、要求を処理する前に候補リストを絞り込んでいると認識します。

次の節では、すべての ID メカニズムの長所と短所について説明します。また、すべての ID のしきい値の調整についてのアドバイスも示します。

すべての ID メカニズムの長所

すべての ID メカニズムは、検索結果がディレクトリエントリのほとんどまたは全体になる場合 (`cn=*` の検索など) に、検索性能を向上させる重要なメカニズムです。Directory Server から、すべてのエントリ ID が返された場合には、次のような仮定をします。

- 無限に増加するエントリ ID リストを保持する必要がなくなり、Directory Server のディスク使用量を最小限にすることができる
- すべての結果を含んだディレクトリエントリを返す検索要求に対応するために、不必要に大きいエントリ ID リストをメモリにロードする必要がなくなり、大量のディスク読み込み処理が減り、検索性能が向上する

- 不必要に大きいエントリ ID リストをメモリに保持する必要がなくなり、大量の RAM が必要とされなくなる

すべての ID メカニズムの短所

ディレクトリのサイズと比べて、すべての ID のしきい値の設定が低すぎたり (ほとんどの場合は、これが問題となる)、高すぎたりすると、性能上の問題が発生します。

すべての ID のしきい値が低すぎる場合

すべての ID のしきい値の設定が低すぎると、必要以上に大量のインデックスキーにすべての ID のトークンが含まれるようになります。この結果、必要以上に多くのディレクトリですべてのエントリが調べられることになります。この検索によって性能に重大な影響が及びます。

たとえば、共通名 (cn) 属性で等価インデックスを管理しているとします。cn インデックスに格納されているインデックスキーの 1 つは cn=James です。対応するエントリ ID リストには、James に設定されている属性を含むすべてのエントリ ID 番号が含まれます。

ディレクトリにあるエントリの一部にしか cn=James は含まれないため、cn 属性の等価インデックスの維持は簡単です。検索要求に対応する場合、エントリ ID の一部だけを調べれば済むため、cn=James フィルタを使用する検索の性能は向上します。

ただし、ディレクトリは時間とともに大きくなり続ける可能性があります。このため、James が追加されていく可能性がありますが、ディレクトリエントリ全体に比べれば、その比率は同じく相対的に小さなものです。最終的に、cn=James エントリ ID リストは非常に大きくなる可能性がありますが、検索性能のためにはこのリストが必要です。すべての ID のしきい値に達するほどたくさんの cn=James エントリが追加されディレクトリが大きくなった場合、cn=James エントリ ID リストは、すべての ID トークンで置き換えられます。cn=James を検索するたびに、Directory Server は、検索要求に応じてディレクトリ内のすべてのエントリを調べます。

データベースが大きくなると、すべてのインデックスキーの大部分がすべての ID のしきい値に設定され、検索性能は大幅に低下します。

すべての ID のしきい値が高すぎる場合

すべての ID のしきい値の設定値が高すぎても、性能上の問題が発生します。すべての ID のしきい値が極端に高いと、検索要求に応答するために維持され、検索時にメモリに読み込まれるリストが、より大きくなります。極端に高く設定されたすべての ID のしきい値によって、すべての ID メカニズムの長所はすべて失われます (詳細は、367 ページの「すべての ID メカニズムの長所」を参照)。

単一のエンタープライズディレクトリにおけるすべての ID のしきい値の調整に関するアドバイス

サーバで、デフォルトのすべての ID のしきい値を変更するときには、注意が必要です。このしきい値を適切でない値に変更すると、サーバの性能は向上せずに、かえって低下する結果になります。この調整に関するアドバイスは、80,000 エントリまでの単一のエンタープライズディレクトリを主に対象にしています。

ディレクトリサイズが一定の場合は、すべての ID のしきい値を、ディレクトリに格納されている総エントリ数の約 5 % に設定します。つまり、ディレクトリに 50,000 のエントリある場合は、すべての ID のしきい値を 2,500 に設定します。

将来、ディレクトリに大量のエントリを追加する予定がある場合は、すべての ID のしきい値を慎重に設定する必要があります。次の点を考慮してください。

- すべての ID のしきい値を変更すると、データベースの再構築が必要となる。この操作には、コストがかかる可能性がある。特に、数百万ものエントリを含むディレクトリでは、かなりのコストがかかる
- すべての ID のしきい値には、ディレクトリサイズの 5 % に設定することが推奨されているが、しきい値を現在のデータベースサイズの 0.5 % から 50 % までの間に設定しても、大きな性能上の問題は発生しない。しかし、できる限り、5 % 前後にすることを推奨する

現在のディレクトリの要件と将来の拡張計画のバランスを考えて計画することで、すべての ID のしきい値をあとから変更 (データベースの再構築が必要) しないで済むようにする必要があります。

たとえば、現在、ディレクトリに 50,000 エントリあるとします。しかし、数年後には、ディレクトリのサイズが 1,000,000 エントリまで大きくなることが予測されます。すべての ID のしきい値を 50,000 の 5 %、つまり 2,500 に設定した場合、ディレクトリが 1,000,000 エントリまで大きくなると、性能上の問題が発生します。1,000,000 エントリのデータベースにおけるしきい値の最小値は、1,000,000 の 0.5%、つまり 5,000 エントリなので、2,500 エントリでは少なすぎます。

将来、ディレクトリが大きくなることが予想される場合は、次のどちらかの方法を選択します。

- すべての ID のしきい値を現在の最適値 (2,500) に設定しておき、このしきい値の設定では正常な動作が保証されないほどエントリ数が増加したときに、データベースを再構築する。データベースを再構築するには、再構築している間ディレクトリを停止するか、少なくともディレクトリを読み取り専用モードにする。また、該当する Directory Server によってエントリがレプリケートされるコンシューマサーバも、初期化し直さなければならない

- 現在の要件と比べると高すぎるが、将来の要件を考えたときにも問題なく動作する値を見つける。たとえば、現在のディレクトリに 50,000 エントリに対して、すべての ID のしきい値を 20,000 に設定する。これは 50,000 の 40 % (現在のディレクトリの要件の範囲内) であり、1,000,000 の 2 % (将来のディレクトリの要件の範囲内) でもある

どちらの戦略を選ぶかは、ディレクトリを導入する際の要件によって決めます。データベース (および関連するコンシューマサーバ) の再構築にかかるコストと、すべての ID のしきい値を理想的な 5 % の設定から変更することによる性能への影響を比較検討してください。

注	コンシューマサーバはさまざまな検索に対して応答するように調整されるため、コンシューマサーバのすべての ID のしきい値は、異なる値を設定するほうが適切な場合もあります。

また、ディレクトリが大きくなる速度や、ディレクトリのサイズを大きくするのにかかる時間も考慮してください。ディレクトリが大きくなるのに何年もかかる場合は、データベースの再構築を計画します。数か月のうちに、ディレクトリが急速に大きくなる場合は、データベースを再構築する回数を最小限にできるように、すべての ID のしきい値を設定する方法を考えます。

サービスプロバイダおよびエクストラネットにおけるすべての ID のしきい値の調整に関するアドバイス

ホストサービスプロバイダ、エクストラネットのディレクトリ、およびエントリ数が 80,000 を超えるディレクトリの調整に関しては、iPlanet プロフェッショナルサービスにお問い合わせください。

すべての ID のしきい値のデフォルト値

デフォルトでは、Directory Server のすべての ID のしきい値は 4000 に設定されています。この値は、80,000 エントリまでのデータベースに適しています。データベースが 80,000 エントリを超える可能性がある場合は、データベースを実装する前に、すべての ID のしきい値を増やすことをお勧めします。

すべての ID のしきい値が適切でない場合の徴候

すべての ID のしきい値の設定が適切でないと、検索性能が低下します。しかし、その他の理由でも、検索性能が低下することがあります。たとえば、次のようにします。

- インデックスを維持していないエントリに対して検索が頻繁に行われた
- 使用しているデータベースのキャッシュサイズとエントリキャッシュサイズの設定が正しくない。詳細は、419 ページの「Directory Server の性能の調整」を参照

これらの可能性を慎重に検討してから、すべての ID のしきい値を変更してください。

すべての ID のしきい値が低すぎることが原因でサーバに問題があると思われる場合は、アクセスログを確認します (第 12 章「サーバとデータベースアクティビティの監視」を参照)。すべてのエントリ ID を返す検索には、`notes=U` フラグが付きます。`notes=U` フラグは、次の検索に対して返されます。

- インデックスが維持されてないエントリに対する検索
- そのインデックスキーですべての ID のしきい値に達したために、ID リストが維持されていないエントリに対する検索

検索結果が、インデックスを作成しておくべき検索に属するかどうかを判断するには、RESULT 行の `conn` と `op` の値を、アクセスログファイルにある以前の SRCH 行と一致させる必要があります。SRCH 行には、検索要求で使用された検索フィルタが表示されます。指定された検索フィルタにインデックスが付いている場合は、`notes=U` フラグは、そのインデックスキーで、すべての ID のしきい値に達したことを示します。たとえば、次のようなアクセスログがあるとします。

```
[24/July/1998:15:12:20 -0800] conn=2 op=1 SRCH base="o=siroe.com"
scope=0 filter="(cn=James)"

[24/July/1998:15:12:20 -0800] conn=2 op=1 RESULT err=0 tag=101
nentries=10000 notes=U
```

`notes=U` フラグは、`cn` 属性のインデックスに対してすべての ID のしきい値に達したことを示します。

すべての ID のしきい値の変更

使用しているサーバのすべての ID のしきい値を変更するには、次の手順を実行します。

1. Directory Server を停止します。

2. コマンド行を使用して、すべてのディレクトリデータベースを、LDIF にエクスポートします。

   詳細は、第 4 章「ディレクトリデータベースへのデータの実装」を参照してください。

3. `ldapmodify` ユーティリティを使用して、`nsslapd-allidsthreshold` エントリを編集するか、次のファイルを編集します。

   `/var/ds5/slapd-`*serverID*`/config/dse.ldif`

4. `nsslapd-allidsthreshold` 属性を探して、必要な値に変更します。

5. `ldif2db` を使用して、すべてのデータベースを初期化します。

   第 4 章「ディレクトリデータベースへのデータの実装」を参照。

6. Directory Server を再起動します。

すべての ID のしきい値を増やしたあと、データベースのキャッシュサイズを確認してください。

すべての ID のしきい値を増やすと、エントリ ID リストも大きくなるので、より大きなメモリが必要になります。必要となるメモリの増加量は、維持しているインデックスの数とタイプによって異なりますが、`nsslapd-allidsthreshold` 属性値で増やした量よりも大きくなることはありません。つまり、`nsslapd-allidsthreshold` 属性値を 2 倍にした場合でも、データベースのキャッシュサイズを現在のサイズの 2 倍以上に増やす必要はありません。

すべての ID のしきい値を増やしたのと同じ割合でデータベースのキャッシュサイズを増やすのは、特殊な方法です。物理メモリが使用可能な場合は、`nsslapd-allidsthreshold` 値の増分の 25 % だけ、データベースのキャッシュサイズを増やしてください。たとえば、すべての ID のしきい値を 2 倍にしたときは、データベースのキャッシュサイズを 50 % 増やします。必要に応じて、サーバの性能に満足するまで、キャッシュサイズを徐々に増やしてください。

属性 `nsslapd-dbcachesize` を使用して、データベースのキャッシュサイズを設定します。詳細は、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』の `nsslapd-dbcachesize` 属性を参照してください。

属性名のクイックリファレンス

次の表に、プライマリ名 (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、必ずプライマリ名を使用してください。

表 10-6 属性のプライマリ名とエイリアス

属性のプライマリ名	属性のエイリアス名
dn	distinguishedName
cn	commonName
sn	surName
c	countryName
l	localityName
st	stateOrProvinceName
street	streetAddress
o	organization
ou	organizationalUnitName
facsimileTelephoneNumber	fax
uid	userId
mail	rfc822mailbox
mobile	mobileTelephoneNumber
pager	pagerTelephoneNumber
co	friendlyCountryName
labeledUri	labeledUri
ttl	timeToLive
dc	domainComponent
authorCn	documentAuthorCommonName
authorSn	documentAuthorSurname
drink	favoriteDrink

第 11 章

SSL の管理

iPlanet Directory Server には、ネットワーク上でセキュリティ保護された通信を行うため、LDAPS 通信プロトコルが用意されています。LDAPS は標準の LDAP プロトコルのうち、SSL (Secure Sokets Layer) の最上位レベルで実行されるものです。

この章では、Directory Server での SSL の使い方について、次の項目ごとに説明します。

- Directory Server への SSL の導入
- サーバ証明書の入手とインストール
- SSL の有効化
- セキュリティの設定
- 証明書に基づく認証の使用
- LDAP クライアントで SSL を使用するための構成

Directory Server への SSL の導入

SSL を使用すると、LDAP クライアントと Directory Server の通信、レプリケーションアグリーメントによってバインドされた複数の Directory Server 間の通信、およびデータベースリンクとリモートデータベースとの通信の安全性を確保することができます。SSL で使用できる認証の種類には、簡易認証 (バインド DN およびパスワード) と証明書に基づく認証があります。

簡易認証による SSL を使用すると、機密性とデータの整合性が保証されます。さらに、簡易認証で使用されるバインド DN やパスワードの代わりに、Directory Server での認証に証明書を使用すると、次のような効果が得られます。

- 効率の向上

使用しているアプリケーションで、証明書データベースへのパスワード入力が 1 回だけ必要な場合、以降のバインド処理や認証処理などで証明書が必要になっても、バインド DN やパスワードの情報が以降の処理に継承されます。

- セキュリティの向上

  証明書に基づく認証は、証明書を使用しないバインド処理と比較して、より安全確実にセキュリティが保護されています。これは、証明書に基づく認証が公開鍵暗号方式を使用するためです。この結果、バインドに必要な情報は、ネットワーク上で傍受できなくなります。

Directory Server では、SSL による通信と SSL を使用しない通信を同時に実行できます。これは、Directory Server で SSL による通信と SSL を使用しない通信のどちらかだけを選択する必要はなく、2 つを同時に使用できることを意味します。

注	Directory Server を UNIX プラットフォーム上で実行している場合は、SSL を有効にすると、StartTLS 拡張処理も有効になります。StartTLS 拡張処理によって、通常の LDAP 接続のセキュリティが提供されます。

SSL の有効化：手順の概要

LDAPS を使用するには、次の手順を実行します。

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 (CA) の証明書を信頼するように Directory Server を構成します。

   詳細は、377 ページの「サーバ証明書の入手とインストール」を参照してください。

2. ディレクトリで、SSL を有効にします。

   詳細は、382 ページの「SSL の有効化」を参照してください。

3. SSL 対応の Directory Server に接続するように Administration Server を設定します。

   詳細は、『Managing Servers with iPlanet Console』を参照してください。

4. SSL を使用して認証を行うすべてのクライアント用に、Directory Server の各ユーザが個人の証明書を入手してインストールするように設定します (省略可能)。

   詳細は、386 ページの「LDAP クライアントで SSL を使用するための構成」を参照してください。

SSL、インターネットセキュリティ、および証明書については、『Managing Servers with iPlanet Console』を参照してください。

サーバ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、および証明機関 (CA) の証明書を信頼して Directory Server を構成するそれぞれのプロセスについて説明します。

このプロセスは、各 Directory Server で SSL を有効にする前に必要となる最初のステップです。次に説明する作業をすでに完了している場合は、382 ページの「SSL の有効化」を参照してください。

証明書を入手およびインストールするには、次の手順を実行します。

- ステップ 1 : 証明書要求の作成
- ステップ 2 : 証明書要求の送信
- ステップ 3 : 証明書のインストール
- ステップ 4 : CA の信頼
- ステップ 5 : 新しい証明書のインストールの確認

証明書要求ウィザードを使用して、証明書要求を作成し (ステップ 1)、CA にその要求を送ります (ステップ 2)。次に、証明書インストールウィザードを使用して、証明書をインストールし (ステップ 3)、CA の証明書を信頼するように設定します (ステップ 4)。

これらのウィザードは、証明書データベースの作成および鍵のペアのインストール処理を自動的に行います。

ステップ 1 : 証明書要求の作成

証明書要求を作成して CA に送るには、次の手順を実行します。

1. Directory Server Console で、「タスク」タブを選択し、「証明書の管理」をクリックします。

   「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。

   証明書要求ウィザードが表示されます。

3. 「次」をクリックします。
4. 空白のテキストフィールドに要求者の情報を入力し、「次」をクリックします。

   次の情報を絶対パスで入力します。

サーバ名 :DNS 検索で使用される、Directory Server の完全修飾でホスト名を入力します。
例 : `dir.siroe.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) : 部署名を入力します。

所在地 (省略可能) : 会社の所在地 (市町村名) を入力します。

都道府県名 : 会社の所在地 (都道府県名) を完全名で入力します。(省略形は不可)

国 : 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 の完全な国コードリストは、『iPlanet Directory Server スキーマリファレンス』に記載されています。

5. 非公開鍵を保護するために使用されるパスワードを入力し、「次」をクリックします。

   パスワードを入力するまで、「次」フィールドはグレー表示されています。「次」をクリックすると、「要求の送信」ダイアログボックスが表示されます。

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、証明書要求ウィザードを終了します。

証明書要求を作成したら、CA に要求を送ることができます。

ステップ 2 : 証明書要求の送信

証明書情報を CA に送るには、次の手順を実行します。

1. 電子メールプログラムを使用して、新しい電子メールメッセージを作成します。

2. クリップボードまたは保存されたファイルからメッセージの本文に証明書要求情報をコピーします。

   メッセージの内容は、次の例のようになります。

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBrjCCARcCAQAwbjELMAkGA1UEBhMCVXMxEzARBgNVBAgTCkNBTElGT1JOSUExLD
AqBgVBAoTI25ldHNjYXBlIGNvbW11bmljYXRpb25zIGNvcnBvcmF0aW9uMRwwGgYDV
QQDExNtZWxsb24ubmV0c2NhcGUuY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQK
BgQCwAbskGh6SKYOgHy+UCSLnm3ok3X3u83Us7ug0EfgSLR0f+K41eNqqWRftGR83e
mqPLDOf0ZLTLjVGJaH4Jn4l1gG+JDf/n/zMyahxtV7+mT8GOFFigFfuxJaxMjr2j7I
vELlxQ4IfZgWwqCm4qQecv3G+N9YdbjveMVXW0v4XwIDAQABoAAwDQYJKoZIhvcNAQ
EEBQADgYEAZyZAm8UmP9PQYwNy4Pmypk79t2nvzKbwKVb97G+MT/gw1pLRsI1uBoKi
nMfLgKp1Q38K5Py2VGW1E47K7/rhm3yVQrIiwV+Z8Lcc=
-----END NEW CERTIFICATE REQUEST-----
```

3. このメッセージを電子メールで CA に送ります。

証明書要求を電子メールで送ったら、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。証明書のインストール時にこのデータが必要となります。

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再インストールすることができます。

証明書を受け取ると、サーバの証明書データベースに証明書をインストールすることができます。

ステップ 3 : 証明書のインストール

サーバの証明書をインストールするには、次の手順を実行します。

1. Directory Server Console で、「タスク」タブを選択し、「証明書の管理」をクリックします。

   「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. 証明書の場所として次のオプションから 1 つ選択し、「次」をクリックします。

   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内 : CA からの電子メール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、このフィールドに貼り付けます。たとえば、次のようにします。

```
-----BEGIN CERTIFICATE-----
MIICMjCCAZugAwIBAgICCEEwDQYJKoZIhvcNAQEFBQAwfDELMAkGA1UEBhMCVVMx
IzAhBgNVBAoTGlBhbG9va2FWaWxsZSBXaWRnZXRzLCBJbmMuMR0wGwYDVQQLExRX
aWRnZXQgTWFrZXJzICdSJyBVczEpMCcGA1UEAxMgVGVzdCBUZXN0IFRlc3QgVGVz
dCBUZXN0IFRlc3QgQ0EwHhcNOTgwMzEyMDIzMzU3WhcNOTgwMzI2MDIzMzU3WjBP
MQswCQYDVQQGEwJVUzEoMCYGA1UEChMfTmV0c2NhcGUgRGlyZWN0b3J5IFB1Ymxp
Y2F0aW9uczEWMBQGA1UEAxMNZHVgh49dq2itLmNvbTBaMA0GCSqGSIb3DQEBAQUA
A0kAMEYCQQCksMR/aLGdfp4m0OiGcgijG5KgOsyRNvwGYW7kfW+8mmijDtZRjYNj
jcgpF3VnlsbxbclX9LVjjNLC57u37XZdAgEDozYwNDARBglghkgBhvhCAQEEBAMC
APAwHwYDVR0jBBgwFoAU67URjwCaGqZuUpSpdLxlzweJKiMwDQYJKoZIhvcNAQEF
```

```
BQADgYEAJ+BVem3vBOP/BveNdLGfjlb9hucgmaMcQa98A/db8qimKT/ue9UGOJqL
bwbMKBBopsD56p2yV3PLJIsBgrcuSoBCuFFnxBnqSiTS/7YiYgCWqWaUAExJFmD6
6hBLseqkSWulk+hXHN7L/NrViO+7zNtKcaZLlFPf7d7j2MgX4Bo=
-----END CERTIFICATE-----
```

4. 表示された証明書情報が正しいことを確認し、「次」をクリックします。
5. 証明書の名前を指定し、「次」をクリックします。
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、377 ページの「ステップ 1 : 証明書要求の作成」で入力したものと同じです。

これで、サーバの証明書のインストールは終了しました。次に、サーバの証明書の入手元の CA を信頼してサーバを構成する必要があります。

ステップ 4 : CA の信頼

CA を信頼して Directory Server を構成することには、CA の証明書の入手および証明書データベースへのサーバのインストールが含まれます。このプロセスは、使用する CA によって異なります。商用 CA の中には、証明書を自動的にダウンロードできる Web サイトを提供するものもあります。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。

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によって、CA を信頼して Directory Server を構成できます。

1. Directory Server Console で、「タスク」タブを選択し、「証明書の管理」をクリックします。

   「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 発行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、ヘッダを含む証明書をコピーし、与えられたテキストフィールドに貼り付けます。「次」をクリックします。
4. 表示された証明書情報が正しいことを確認し、「次」をクリックします。
5. 証明書の名前を指定し、「次」をクリックします。

6. CA の信頼目的を次の中から選択します (両方を選択することも可能)。

   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) : サーバはクライアントの証明書が信頼された CA によって発行されたものであることを確認します。

   他のサーバへの接続を受け入れる (サーバ認証) : サーバは、(たとえば、レプリケーション更新のために) 接続している Directory Server に、信頼された CA によって発行された証明書があることを確認します。

7. 「完了」をクリックして、ウィザードを終了します。

サーバの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。ただし、証明書が正しくインストールされていることをまず確認する必要があります。

ステップ 5 : 新しい証明書のインストールの確認

1. Directory Server Console で、「タスク」タブを選択し、「証明書の管理」をクリックします。

   「証明書の管理」ウィンドウが表示されます。

2. 「サーバ証明書」タブを選択します。

   該当するサーバについてインストール済み証明書のリストが表示されます。

3. リストをスクロールします。インストールした証明書が表示されていることを確認します。

   これで、サーバで SSL を有効にする準備が整いました。

SSL の有効化

通常、サーバは 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トランザクションを処理する前に、SSL を必ず有効にしてください。

SSL を有効にする前に、377 ページの「サーバ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。

SSL による通信を有効にするには、次の手順を実行します。

1. SSL 通信にサーバで使用したいセキュリティ保護されたポートを設定します。詳細は、36 ページの「Directory Server のポート番号の変更」を参照してください。

   指定する暗号化されたポート番号は、通常の LDAP 通信に使用するポート番号とは異なる必要があります。デフォルトでは、標準のポート番号は 389 で、セキュリティ保護されたポート番号は 636 です。

2. Directory Server Console で「構成」タブを選択し、左側の区画のナビゲーションツリーで一番上にあるエントリを選択します。
3. 右側の区画で「暗号化」タブを選択します。

   このタブには、サーバの現在の暗号化設定が表示されます。

4. 「このサーバの 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。
5. 「この暗号ファミリを使用する」チェックボックスを選択します。
6. 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。
7. 「暗号の設定」をクリックします。

   「暗号の設定」ダイアログボックスが表示されます。

8. 使用する暗号の横にあるチェックボックスを選択し、「OK」をクリックして、「暗号の設定」ダイアログボックスを閉じます。

   特定の暗号については、383 ページの「セキュリティの設定」を参照してください。

9. 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、クライアントの証明書が無視されます。これは、バインドが失敗するという意味ではありません。

   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「証明書に基づく認証の使用」を参照してください。

   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、クライアントからの認証が要求されます。

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を構成する必要があります。

10. Directory Server との通信に SSL を使用するよう iPlanet Console を設定する場合は、「iPlanet Console で SSL を使用する」を選択します。
11. 「保存」をクリックします。
12. Directory Server を再起動します。

    詳細は、39 ページの「SSL が有効になった状態でのサーバの起動」を参照してください。

セキュリティの設定

SSL 通信に使用する暗号のタイプを選択できます。暗号とは、暗号化に使用するアルゴリズムのことです。一部の暗号は、ほかの方式と比べて、安全でしかも強力です。一般に、暗号化に使用するビット数の多い暗号ほど、鍵の復号化は難しくなります。各種アルゴリズムとその特性については、『Managing Servers with iPlanet Console』を参照してください。

クライアントがサーバとの SSL 接続を開始すると、クライアントは情報の暗号化にどの暗号を使用するかをサーバに通知します。双方向の暗号化プロセスでは、サーバとクライアントで同じ暗号を使用する必要があります。使用可能な暗号は多数ありますが、サーバに接続されているクライアント側のアプリケーションで使用する暗号をサーバで使用できるようにする必要があります。

iPlanet Directory Server が提供する SSL 3.0 暗号は、次のとおりです。

- 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
- 40 ビットの暗号化と MD5 メッセージ認証を使用する RC2 暗号

- 暗号化なし。MD5 メッセージ認証のみ
- 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
- 56 ビットの暗号化と SHA メッセージ認証を使用した FIPS DES。この暗号は、暗号化モジュールの実装用 FIPS 140-1 米国政府規格に準拠する
- 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。この暗号は、暗号化モジュールの実装用 FIPS 140-1 米国政府規格に準拠する

サーバで使用する暗号を選択するには、次の手順を実行します。

1. SSL がサーバで有効になっていることを確認します。

   詳細は、382 ページの「SSL の有効化」を参照してください。

2. Directory Server Console で「構成」タブを選択し、左側の区画のナビゲーションツリーで一番上にあるエントリを選択します。
3. 右側の区画で「暗号化」タブを選択します。

   このタブには、サーバの現在の暗号化設定が表示されます。

4. 「暗号の設定」をクリックします。

   「暗号の設定」ダイアログボックスが表示されます。

5. 「暗号の設定」ダイアログボックスで、サーバで使用する暗号をリストから選択し、「OK」をクリックします。

   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。

6. 「暗号化」タブで「保存」をクリックします。

警告 `none,MD5` は選択しないでください。これは、クライアントで使用可能な暗号がほかにない場合に、サーバによってこのオプションが使用されるためです。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。

---

iPlanet Console 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
- 暗号化なし。MD5 メッセージ認証のみ
- 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

証明書に基づく認証の使用

Directory Server では、コマンド行ツール (LDAP クライアント) やレプリケーションの通信に、証明書に基づく認証を使用できます。証明書に基づく認証は、次のクライアントとサーバの間で使用できます。

- Directory Server に接続している LDAP クライアント
- 別の Directory Server (レプリケーション (replication) または連鎖 (chaining)) に接続している Directory Server

証明書に基づく認証の設定

証明書に基づく認証を設定するには、次の手順を実行します。

1. クライアントとサーバ用、または複製元と複製先の 2 つのサーバ用に証明書データベースを作成します。

   Directory Server に証明書をインストールすると、証明書データベースが自動的に作成されます。クライアント用の証明書データベースの作成については、386 ページの「LDAP クライアントで SSL を使用するための構成」を参照してください。

2. 証明書を入手し、クライアントとサーバの両方、または複製元と複製先の 2 つのサーバにインストールします。

3. クライアントからアクセスされるサーバまたは複製元と複製先の 2 つのサーバで、SSL を有効にします。

   SSL の有効化については、382 ページの「SSL の有効化」を参照してください。

注 iPlanet Console から SSL を経由して Directory Server に接続する場合は、「クライアント認証を要求する」を選択すると、通信が無効になります。これは、iPlanet Console が SSL をサポートしているにもかかわらず、クライアント認証に使用する証明書を持たないためです。

4. 証明書の識別名をディレクトリが認識できる識別名に割り当てます。

   これにより、証明書を使用してバインドするときに、クライアントに対するアクセス制御を設定できます。このマッピングプロセスについては、『Managing Servers with iPlanet Console』を参照してください。

クライアント認証の許可と要求

SSL を使用して Directory Server に接続するよう iPlanet Console を構成済みで、さらに Directory Server がクライアント認証を要求する場合は、iPlanet Console を使用して iPlanet サーバ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、iPlanet Console を使用できるように、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あとからディレクトリ構成を変更することもできます。次の手順を実行します。

1. Directory Server を停止します。

   コマンド行を使用したサーバの停止および起動については、35 ページの「コマンド行からのサーバの起動と停止」を参照してください。

2. `cn=encryption,cn=config` エントリの `nsSSLClientAuth` 属性値を `required` から `allowed` に修正します。

   コマンド行を使用したエントリの修正については、第 2 章「ディレクトリエントリの作成」を参照してください。

3. Directory Server を起動します。

   これで、iPlanet Console を起動できるようになりました。

LDAP クライアントで SSL を使用するための構成

LDAP クライアントアプリケーションを使用して接続するときに、Directory Server のユーザ全員に SSL または証明書に基づく認証を使用してほしい場合は、そのユーザ全員に次の手順を必ず実行してもらう必要があります。

- 証明書データベースの作成
- サーバ証明書を発行する CA の信頼

LDAP クライアントでサーバの証明書を認識するには、これらの操作で十分です。ただし、LDAP クライアントで独自の証明書を使用して Directory Server の認証を行う場合は、Directory Server のユーザ全員に個人の証明書を入手してインストールしてもらう必要があります。

注 クライアントアプリケーションによっては、信頼された証明書がサーバにあるかどうかを検証しません。

次に、Netscape Communicator 4.7 を使用してこれらの処理を実行する方法を説明します。

1. 証明書を作成するには、Netscape Communicator 4.7 を起動するだけです。

   証明書データベースが存在していない場合は、作成されます。

2. Communicator を使用して CA に接続します。

   内部に導入された iPlanet Certificate Server を使用している場合は、次の URL にアクセスします。

   ```
   https://hostname:444
   ```

   CA には、CA の証明書をダウンロードできるリンクを提供しているところもあります。

3. CA を信頼します。

   この処理は、使用する CA によって異なります。iPlanet Certificate Server に接続している場合などは、Communicator から CA を信頼するかどうかを確認するプロンプトが自動的に表示されます。

これらの手順で十分に、クライアントアプリケーションは Directory Server との接続を受け入れるようになります。これは、Directory Server の証明書が信頼された CA 発行のものであることをクライアントが識別するためです。

ただし、Directory Server でクライアントの証明書を使用してクライアントを認証する場合は、以降の追加手順を実行する必要があります。

4. クライアントシステム上で、CA からクライアント証明書を入手します。
5. クライアントシステム上に、クライアント証明書をインストールします。

   証明書の入手方法 (電子メールまたは Web ページから) にかかわらず、証明書をインストールするためのリンクが提供されます。そ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。

   受け取った証明書情報は、必ずファイルに記録するようにしてください。特に証明書のサブジェクト DN を憶えておいてください。これは、サブジェクト DN をディレクトリのエントリに割り当てるよう、サーバを構成する必要があるためです。クライアント証明書の内容は、次の例のようになります

```
-----BEGIN CERTIFICATE-----
MIICMjCCAZugAwIBAgICCEEwDQYJKoZIhvcNAQEFBQAwfDELMAkGA1UEBhMCVVMx
IzAhBgNVBAoTGlBhbG9va2FWaWxsZSBXaWRnZXRzLCBJbmMuMR0wGwYDVQQLExRX
aWRnZXQgTWFrZXJzICdSJyBVczEpMCcGA1UEAxMgVGVzdCBUZXN0IFRlc3QgVGVz
dCBUZXN0IFRlc3QgQ0EwHhcNOTgwMzEyMDIzMzU3WhcNOTgwMzI2MDIzMzU3WjBP
MQswCQYDVQQGEwJVUzEoMCYGA1UEChMfTmV0c2NhcGUgRGlyZWN0b3J5IFB1Ymxp
Y2F0aW9uczEWMBQGA1UEAxMNZHVgh49dq2itLmNvbTBaMA0GCSqGSIb3DQEBAQUA
A0kAMEYCQQCksMR/aLGdfp4m0OiGcgijG5KgOsyRNvwGYW7kfW+8mmijDtZRjYNj
jcgpF3VnlsbxbclX9LVjjNLC57u37XZdAgEDozYwNDARBglghkgBhvhCAQEEBAMC
APAwHwYDVR0jBBgwFoAU67URjwCaGqZuUpSpdLxlzweJKiMwDQYJKoZIhvcNAQEF
BQADgYEAJ+BVem3vBOP/BveNdLGfjlb9hucgmaMcQa98A/db8qimKT/ue9UGOJqL
bwbMKBBopsD56p2yV3PLJIsBgrcuSoBCuFFnxBnqSiTS/7YiYgCWqWaUAExJFmD6
6hBLseqkSWulk+hXHN7L/NrViO+7zNtKcaZLlFPf7d7j2MgX4Bo=
-----END CERTIFICATE-----
```

6. クライアント証明書は、certutil ユーティリティを使用してバイナリ形式に変換する必要があります。証明書を変換するには、次の手順を実行します。

   a. http://www.iplanet.com から certutil ユーティリティをダウンロードします。

      iPlanet のホームページで、**certutil** を検索します。最新の PKCS パッケージをダウンロードします。このパッケージには、certutil ユーティリティが含まれています。

   b. certutil を次の構文で実行します。

      certutil -L -d *cert7.db_path* -n *user_cert_name* -r > *user_cert.bin*

      ここで *cert7.db_path* は証明書データベースの場所を表し、*user_cert_name* は証明書のインストール時に指定した証明書の名前を表します。また、*user_cert.bin* はバイナリ形式の証明書データを含む出力ファイルの名前で、必ず指定する必要があります。

7. サーバで certmap.conf ファイルを編集して、入手した証明書のサブジェクト DN を適切なディレクトリエントリに割り当てます。

   この手順については、『Managing Servers with iPlanet Console』を参照してください。certmap.conf ファイルで、verifyCert パラメタに **on** が設定されていることを確認します。

注 このパラメタに **on** が設定されていない場合、Directory Server は certmap.conf ファイルの情報に一致するディレクトリ内のエントリを検索するだけです。検索処理が成功すると Directory Server は、userCertificate 属性の値を実際に検査することなくアクセス権を与えます。

---

8. Directory Server で、クライアント証明書を所有するユーザのディレクトリエントリを修正し、userCertificate 属性を追加する必要があります。

   a. 「ディレクトリ」タブを選択し、ユーザエントリに移動します。

   b. ユーザエントリをダブルクリックし、「属性エディタ」を使用して binary サブタイプを持つ userCertificate 属性を追加します。

      この属性を追加すると、編集可能なフィールドの代わりに、「値の設定」ボタンが表示されます。

   c. 「値の設定」をクリックします。

      ファイルセレクタが表示されます。セレクタを使用して、手順 6 で作成したバイナリファイルを選択します。

   Directory Server Console を使用したエントリの編集については、45 ページの「ディレクトリエントリの変更」を参照してください。

第 12 章

サーバとデータベースアクティビティの監視

この章では、データベースログとサーバログの監視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。

- ログファイルの表示と構成
- 手動によるログファイルのローテーション
- サーバアクティビティの監視
- データベースアクティビティの監視
- データベースリンクアクティビティの監視

SNMP を使用してサーバ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。

ログファイルの表示と構成

iPlanet Directory Server では、3 種類のログを使用して、ディレクトリの管理と性能調整を行うことができます。ログの種類は以下のとおりです。

- アクセスログ
- エラーログ
- 監査ログ

次に、すべてのタイプのログ構成に共通する項目を示します。

- ログファイルの作成ポリシーの定義
- ログファイルの削除ポリシーの定義

次の節では、ログファイルの作成ポリシーと削除ポリシーの定義方法と、それぞれのタイプのログの表示方法および構成方法を説明します。

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 Console 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。次のパラメタを構成します。

- 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。1 に設定すると、ログのローテーションは行われず、ログが無限に増大してしまうため、この値は 1 には設定できない
- 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、または月を単位とすることを指定して決定する。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 Console 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によって評価されます。

次のパラメタを構成します。

- 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この値に達すると、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、このパラメタは無視される

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。

ここでは、次の手順について説明します。

- 「アクセスログの表示」(393 ページ)
- 「アクセスログの構成」(393 ページ)

アクセスログの表示

アクセスログを表示するには、次の手順を実行します。

1. Directory Server Console で「状態」タブを選択し、ナビゲーションツリーの Logs フォルダを展開して、「Access Log」アイコンを選択します。

   アクセスログの最後の 25 エントリが、テーブルに一覧表示されます。

2. 表示を更新するには、「再読み込み」をクリックします。「連続」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. アーカイブ済みのアクセス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、「再読み込み」をクリックします。
5. 指定した文字列を含むメッセージを表示することもできます。このためには、「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、「再読み込み」をクリックします。

アクセスログの構成

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、アクセスログを無効にする前に、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構成するには、次の手順を実行します。

1. Directory Server Console で、「構成」タブを選択します。次に、ナビゲーションツリーの Logs フォルダを展開し、Access Log アイコンを選択します。

   アクセスログの構成属性が右側の区画に表示されます。

2. アクセスログを有効にするには、「ログを有効にする」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。

   `/var/ds5/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。

   これらのパラメタについては、392 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。

   これらのパラメタについては、392 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。

ここでは、次の手順について説明します。

- 「エラーログの表示」(394 ページ)
- 「エラーログの構成」(395 ページ)

エラーログの表示

エラーログを表示するには、次の手順を実行します。

1. Directory Server Console で「状態」タブを選択し、ナビゲーションツリーの Logs フォルダを展開して、Error Log アイコンを選択します。

   エラーログの最後の 25 エントリが、テーブルに一覧表示されます。

2. 表示を更新するには、「再読み込み」をクリックします。「連続」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、「再読み込み」をクリックします。
5. 指定した文字列を含むメッセージを表示することもできます。このためには、「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、「再読み込み」をクリックします。

エラーログの構成

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。

エラーログを構成するには、次の手順を実行します。

1. Directory Server Console で、「構成」タブを選択します。次に、ナビゲーションツリーの Logs フォルダを展開し、Error Log アイコンを選択します。

   エラーログの構成属性が右側の区画に表示されます。
2. 右側の区画で「エラーログ」タブを選択します。
3. エラーログを有効にするには、「ログを有効にする」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。
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。

   `/var/ds5/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。

   これらのパラメタについては、392 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。

   これらのパラメタについては、392 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. ログレベルを設定する場合は、Ctrl キーを押しながら「ログレベル」リストボックス内の目的のオプションをクリックします。

   ログレベルオプションについては、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』の「ログのレベル」を参照してください。

   これらの値をデフォルトから変更すると、エラーログがすぐに大きくなってしまうことがあります。したがって、iPlanet カスタマサポートに指示された場合を除き、ログレベルを変更しないことをお勧めします。

8. 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、サーバの構成だけでなく、各データベースに対する変更に関する詳細情報が記録されます。

ここでは、次の手順について説明します。

- 「監査ログの表示」(396 ページ)
- 「監査ログの構成」(397 ページ)

監査ログの表示

監査ログを表示するには、そのディレクトリの監査ログの機能を有効にする必要があります。詳細は、397 ページの「監査ログの構成」を参照してください。

監査ログを表示するには、次の手順を実行します。

1. Directory Server Console で、「状態」タブを選択します。次に、ナビゲーションツリーの Logs フォルダを展開し、Audit Log アイコンを選択します。

   監査ログの最後の 25 エントリが、テーブルに一覧表示されます。

2. 表示を更新するには、「再読み込み」をクリックします。「連続」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、「再読み込み」をクリックします。

5. 指定した文字列を含むメッセージを表示することもできます。このためには、「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、「再読み込み」をクリックします。

監査ログの構成

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 Console を使用して行います。

監査ログを構成するには、次の手順を実行します。

1. Directory Server Console で、「構成」タブを選択します。次に、ナビゲーションツリーの Logs フォルダを展開し、Audit Log アイコンを選択します。

   監査ログの構成属性が右側の区画に表示されます。

2. 監査ログを有効にするには、「ログを有効にする」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。

   `/var/ds5/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。

   これらのパラメタについては、392 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。

   これらのパラメタについては、392 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。

手動によるログファイルのローテーション

Directory Server では、3 種類のログすべてにおいて、ログファイルの自動ローテーションが可能です。ただし、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。

`/var/ds5/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. サーバを停止します。手順については、35 ページの「iPlanet 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後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. サーバを再起動します。手順については、35 ページの「iPlanet Directory Server の起動と停止」を参照してください。

サーバアクティビティの監視

Directory Server Console 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。また、すべてのデータベースのキャッシュを監視することもできます。ここでは、次の項目について説明します。

- 398 ページの「Directory Server Console を使用したサーバの監視」
- 402 ページの「コマンド行からのサーバの監視」

Directory Server Console を使用したサーバの監視

この節では、Directory Server Console を使用してサーバを監視する方法と、性能モニターによって確認できる情報について説明します。

サーバ性能モニター情報の表示

Directory Server Console を使用してサーバアクティビティを監視するには、次の手順を実行します。

1. Directory Server Console で、「状態」タブを選択します。ナビゲーションツリーで、「性能カウンタ」を選択します。

   右側の区画の「状態」タブに、現在のサーバアクティビティに関する情報が表示されます。サーバが実行されていない場合は、このタブには性能監視情報は表示されません。
2. 「再読み込み」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」チェックボックスを選択します。

サーバ性能モニター情報の概要

続く節で解説する次の項目に関する監視情報が提供されます。

- 一般情報 (サーバ)
- 資源の概要
- 現在の資源使用状況
- 接続状態

- グローバルデータベースのキャッシュ情報

一般情報 (サーバ)

次のような一般情報が表示されます。

- サーバのバージョン

  現在のサーバのバージョンを特定します。

- 構成 DN

  `ldapsearch` コマンド行ユーティリティを使用して、結果を取得するために検索ベースとして使用する識別名を示します。このフィールドには `cn=monitor` と表示されます。

- データのバージョン

  サーバのデータに関する識別情報が表示されます。通常ここに表示される情報は、サーバによってレプリカがコンシューマサーバに提供されるときに意味を持ちます。データのバージョン情報は、次のように表示されます。

  - サーバのホスト名
  - サーバのポート番号
  - データベース生成番号。廃止される予定。LDIF ファイル内にマシンのデータエントリを設定しないでディレクトリデータベースを作成した場合にだけ作成される識別子で、ほかと重複しない固有の値を持つ。
  - 現在の更新履歴ログ番号。ディレクトリに加えられた最後の変更に対応する番号。この番号は 1 から始まり、データベースに変更を加えるたびに 1 ずつ加算される。

- サーバの起動日時

  サーバが起動した日付と時刻が表示されます。

- サーバの現在の日時

  サーバ上の現在の日付と時刻が表示されます。

資源の概要

Console によって表示される資源の概要テーブルには、次のような資源関連情報が提供されます。

表 12-1 サーバ性能監視 : 資源の概要テーブル

資源	起動時からの使用状況	1 分あたりの平均
接続回数	サーバの起動時から現在までの合計接続回数	サーバの起動時から現在までの 1 分あたりの平均接続回数

表 12-1 サーバ性能監視 : 資源の概要テーブル (続き)

資源	起動時からの使用状況	1 分あたりの平均
開始した処理	サーバの起動時から現在までに開始された処理の総数。処理には、検索、追加、変更などのサーバ処理を求めるすべてのクライアント要求が含まれる。1 回の接続で複数の処理が実行されることもある	サーバの起動時から現在までの 1 分あたりの平均処理回数
完了した処理	サーバの起動時から現在までにサーバによって完了された処理の回数	サーバの起動時から現在までの 1 分あたりの平均処理回数
クライアントに送信されたエントリ数	サーバの起動時から現在までのクライアントへの合計送信エントリ数。エントリは、検索要求の結果としてクライアントに送られる	サーバの起動時から現在までの、クライアントへの 1 分あたりの平均エントリ数
クライアントに送信されたバイト数	サーバの起動時から現在までのクライアントへの合計送信バイト数	サーバの起動時から現在までの、クライアントへの 1 分あたりの平均バイト数

現在の資源使用状況

Directory Server Console の資源の概要テーブルには、次のような資源関連情報が表示されます。

表 12-2 サーバ性能監視 : 現在の資源使用状況テーブル

資源	現在の使用状況
有効なスレッド数	要求処理で使われる、現在有効なスレッドの数が表示される。レプリケーションや連鎖などのサーバの内部タスクによって、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	開いている接続の合計数が表示される。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
使用可能な接続数	同時に接続することが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識別子の数で示される
クライアントへの書き込み待機中のスレッド数	クライアントへの書き込み待機状態にあるスレッドの合計数。サーバからクライアントへのデータ送信時に、送信を一時的に停止しなければならない場合は、スレッドによる書き込みもすぐには行われない。一時停止の理由としては、ネットワークの速度が遅い、クライアントの処理速度が遅い、クライアントに送る情報量が非常に大きい、などが考えられる

表 12-2 サーバ性能監視 : 現在の資源使用状況テーブル (続き)

資源	現在の使用状況
クライアントから読み取り待機中のスレッド数	クライアントからの読み取り待機状態にあるスレッドの合計数。サーバがクライアントからの要求受信を開始した後に、その要求の送信が何らかの理由で中断された場合は、スレッドの読み取りもすぐには行われない。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
スレッド多重度	スレッド多重度を示す
使用中のデータベース	サーバが使用している合計データベース数

接続状態

Directory Server Console の接続状態テーブルには、現在開かれている個々の接続が使用中の資源の量に関して、次の情報が表示されます。

表 12-3 サーバ性能監視 : 接続状況テーブル

テーブルの見出し	説明
接続が行われた時刻	最初の接続時のサーバ上の時刻を示す
開始された処理	この接続が開始された処理の数を示す
完了した処理	この接続で完了した処理の数を示す
バインド DN	クライアントがサーバへのバインド処理に使用した識別名を示す。クライアントがサーバに対して認証していない場合は、このフィールドには「not bound」と表示される
読み取り / 書き込み	クライアントによるサーバへの読み取りまたは書き込みアクセスが、現在ブロックされているかどうかを示す。次の 2 つの値が可 • ブロックなし : サーバがアイドル状態にあることを示す。クライアントへのアクティブなデータ送信、またはクライアントからのアクティブなデータ読み取りが可 • ブロック : サーバがクライアントへのデータ送信、またはクライアントからのデータ読み取りを試みているが、どちらも処理できないことを示す。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

グローバルデータベースのキャッシュ情報

Directory Server Console のグローバルデータベースのキャッシュ情報テーブルには、次の情報が表示されます。

表 12-4　サーバ性能監視 : グローバルデータベースのキャッシュテーブル

テーブルの見出し	説明
ヒット数	サーバがディスクにアクセスせずに、キャッシュからデータを取得することによって要求を処理できる回数を示す
試行数	サーバの起動時からディレクトリ上で実行された要求の合計数を示す
ヒット率	キャッシュヒットを試みて成功した率を示す。この数値が 100% に近いほど、性能が高くなる
読み取りページ	ディスクからキャッシュに読み取られたページ数を示す
書き込みページ	キャッシュからふたたびディスクに書き込まれたページ数を示す
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。ページがキャッシュから破棄されると、サーバの性能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、性能は高くなる
破棄された読み取り / 書き込みページ	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 / 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 ページがキャッシュから破棄されると、サーバの性能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、性能は高くなる

コマンド行からのサーバの監視

次の特性を使用して検索操作を実行することによって、あらゆる 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。

- 検索属性 : `objectClass=*`
- 検索ベース : `cn=monitor`
- 検索対象 : `base`

たとえば、次のようにします。

```
ldapsearch -h directory.siroe.com -s base -b "cn=monitor"
"(objectclass=*)"
```

Directory Server の検索については、489 ページの「ldapsearch の使用」を参照してください。

サーバの監視属性は、`cn=monitor,cn=config` エントリ内にあります。

ldapsearch を使用してサーバアクティビティを監視すると、次の情報を見ることができます。

- version: ディレクトリの現在のバージョン番号が表示される
- threads: 要求処理で使われる、現在有効なスレッドの数が表示される。レプリケーションや連鎖などのサーバの内部タスクによって、追加のスレッドが生成されることもある
- connection:*fd*:*opentime*:*opsinitiated*:*opscompleted*:*binddn*:[rw]: 開いている各接続について、次のような概要情報を与える (ディレクトリマネージャとしてディレクトリにバインドする場合にだけ使用可能)
  - fd: この接続に使用されているファイルディスクリプタ。
  - opentime: この接続が開かれた時刻。
  - opsinitiated: この接続によって開始された処理の数。
  - opscompleted: 完了した処理の数。
  - binddn: ディレクトリへの接続のためこの接続に使用された識別名。
  - rw: 読み取りまたは書き込みに対して接続がブロックされているかを表すフィールド。

  デフォルトでは、この情報が使用できるのは、ディレクトリマネージャとしてディレクトリにバインドした場合だけ。ただし、この情報に関連する ACI を変更すれば、ほかのユーザにもこの情報へのアクセスを許可できる
- currentconnections: 現在ディレクトリが使用中の接続の数を示す
- totalconnections: ディレクトリの起動時から現在までに処理された接続数を示す
- dtablesize: そのディレクトリで使用できるファイルディスクリプタの数を表す。接続ごとに 1 つのファイルディスクリプタが必要。1 つはすべての開いたインデックス用、1 つはログファイルの管理用、1 つは ns-slapd 自身用。本来この値は、そのディレクトリであといくつの同時接続が可能であるかを示す。ファイルディスクリプタについては、オペレーティングシステムのマニュアルを参照
- readwaiters: クライアントからの読み取り待機状態にあるスレッドの数を示す
- opsinitiated: サーバの起動時から現在までに開始された操作の数を示す
- opscompleted: サーバの起動時から現在までに完了した操作の数を示す
- entriessent: サーバの起動時から現在までのクライアントへの送信エントリ数を示す
- bytessent: サーバの起動時から現在までのクライアントへの送信バイト数を示す

- currentime : このサーバのスナップショットの出力時刻を示す。この時刻にはグリニッジ標準時 (GMT) が使用され、UTC 形式で表示される
- starttime : サーバが起動した時刻を示す。この時刻にはグリニッジ標準時 (GMT) が使用され、UTC 形式で表示される
- nbackends : サーバのサービス対象となるバックエンド (データベース) の数を示す
- concurrency : 現在のスレッド多重度を表示する
- backendmonitordn : 各ディレクトリデータベースの DN を示す

データベースアクティビティの監視

Directory Server Console またはコマンド行から、データベースの現在のアクティビティを監視できます。ここでは、次の項目について説明します。

- 404 ページの「Server Console を使用したデータベースアクティビティの監視」
- 408 ページの「コマンド行からのデータベースの監視」

Server Console を使用したデータベースアクティビティの監視

この節では、Directory Server Console を使用してデータベース性能モニターを表示する方法と、モニターに表示される情報の内容について説明します。

データベース性能モニターの表示

データベースアクティビティを監視するには、次の手順を実行します。

1. Directory Server Console で、「状態」タブを選択します。ナビゲーションツリーの「Performance Counters」フォルダを展開し、監視するデータベースを選択します。

   タブに、現在のデータベースアクティビティに関する情報が表示されます。サーバが実行されていない場合は、このタブには性能監視情報は表示されません。

2. 「再読み込み」をクリックすると、現在表示されている情報が更新されます。表示される情報を継続して更新するには、「連続」チェックボックスを選択して、「再読み込み」をクリックします。

データベース性能モニター情報の概要

次の節で説明するように、データベース監視情報が表示されます。

- 一般情報 (データベース)
- 概要情報テーブル
- データベースのキャッシュ情報テーブル
- データベースファイル固有テーブル

一般情報 (データベース)

次のような一般情報が表示されます。

- データベース

  監視対象のデータベースのタイプを示します。

- 構成 DN

  `ldapsearch` コマンド行ユーティリティを使用して、結果を取得するために検索ベースとして使用する識別名を示します。

概要情報テーブル

概要情報テーブルには、次の情報が表示されます。

表 12-5 データベース性能監視 : 概要情報

性能の基準	内容
読み込み専用の状態	現在データベースが読み取り専用モードであるかどうかを示す。`readonly` 属性が `on` に設定されていれば、データベースは読み取り専用モードになっている
エントリキャッシュの検索ヒット数	エントリキャッシュを使用した検索が成功した回数。つまり、サーバがディスクにアクセスせずに、キャッシュからデータを取得することによって、検索要求を処理できた回数
エントリキャッシュの検索試行数	ディレクトリが最後に起動してから現在までの、エントリキャッシュ検索の回数。つまり、サーバの起動時からサーバに対して実行された検索操作の合計数

表 12-5　データベース性能監視 : 概要情報 (続き)

性能の基準	内容
エントリキャッシュの検索ヒット率	エントリキャッシュの検索に対する、エントリキャッシュ試行が成功した割合。この値は、ディレクトリが最後に起動してから現在までの、検索回数とヒット回数の合計に基づく。この値が 100% に近いほど、性能が高くなる。エントリキャッシュにないエントリを検索するには、ディレクトリがディスクにアクセスして検索を実行する必要がある。したがって、この率がゼロに近くなれば、それだけディスクアクセスの回数が増え、ディレクトリの検索性能は低下する このヒット率を上げるために、「最大キャッシュエントリ数」属性の値を大きくして、エントリキャッシュ内に維持できるエントリ数を増やすことができる。Server Console を使用してこの値を変更する方法については、420 ページの「データベースの性能の調整」を参照
現在のエントリキャッシュサイズ (バイト単位)	現在エントリキャッシュ内にある、ディレクトリエントリの合計サイズを示す
最大エントリキャッシュサイズ (バイト単位)	ディレクトリが維持するエントリキャッシュのサイズを示す。この値は、「最大キャッシュサイズ」属性によって管理される。Server Console を使用してこの値を変更する方法については、420 ページの「データベースの性能の調整」を参照
現在のエントリキャッシュサイズ (エントリ単位)	現在エントリキャッシュ内にある、ディレクトリエントリの合計数を示す
最大エントリキャッシュサイズ (エントリ単位)	現在エントリキャッシュ内で維持できる、ディレクトリエントリの最大数を示す。この値は、「最大キャッシュエントリ数」属性によって管理される。Server Console を使用してこの値を変更する方法については、420 ページの「データベースの性能の調整」を参照

データベースのキャッシュ情報テーブル

データベースのキャッシュ情報テーブルには、次のキャッシュ情報が表示されます。

表 12-6　データベース性能監視 : データベースキャッシュ情報

性能の基準	内容
ヒット数	要求されたページをデータベースキャッシュから正常に提供した回数を示す。1 ページは、サイズ 2 K のバッファ
試行数	データベースキャッシュにあるページが要求された回数を示す

表 12-6　データベース性能監視 : データベースキャッシュ情報 (続き)

性能の基準	内容
ヒット率	データベースキャッシュの「ヒット数」と「試行数」の比率を示す。この値が 100% に近いほど、性能が高くなる。データベースキャッシュにないデータベースのデータを検索するには、ディレクトリがディスクにアクセスして、適切なページを取得する必要がある。したがって、この率がゼロに近くなれば、それだけディスクアクセスの回数が増え、ディレクトリの性能は低下する このヒット率を上げるために、「最大キャッシュサイズ」属性の値を大きくして、データベースキャッシュ内に維持できるデータ量を増やすことができる。Server Console を使用してこの値を変更する方法については、420 ページの「データベースの性能の調整」を参照
読み取りページ	ディスクからデータベースキャッシュに読み取られたページ数を示す
書き込みページ	キャッシュからふたたびディスクに書き込まれたページ数を示す。読み書き可能ページに変更が加えられると、データベー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。キャッシュがいっぱいになり、ディレクトリ操作のために現在キャッシュにないデータベースページが必要になると、古いページがデータベースキャッシュから削除される
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
破棄された読み取り / 書き込みページ	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 / 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

データベースファイル固有テーブル

ディレクトリは、データベースを構成するインデックスファイルのテーブルを表示します。テーブルには、次の情報が表示されます。

表 12-7　データベース性能監視 : データベースファイル固有テーブル

性能の基準	内容
キャッシュのヒット数	この特定ファイルについて、検索結果としてのキャッシュヒットの回数。つまり、このファイルからのデータを必要とする検索をクライアントが実行し、これに対してディレクトリがキャッシュから必要なデータを取得した回数
キャッシュのミス数	この特定ファイルについて、検索結果としてキャッシュヒットとならなかった回数。つまり、このファイルからのデータを必要とする検索が実行され、これに対してキャッシュ内に必要なデータが見つからなかった回数
読み取りページ	このファイルからキャッシュに移動されたページ数を示す

表 12-7　データベース性能監視 : データベースファイル固有テーブル (続き)

性能の基準	内容
書き込みページ	このファイル用にキャッシュからふたたびディスクに書き込まれたページ数を示す

コマンド行からのデータベースの監視

次の特性を使用して検索操作を実行することによって、すべての LDAP クライアントから、ディレクトリデータベースのアクティビティを監視できます。

- 検索属性 : `objectClass=*`
- 検索ベース : `cn=monitor,cn=`*database_instance*`,cn=ldbm database, cn=plugins, cn=config`。ここで、database は監視するデータベースの名前
- 検索対象 : `base`

たとえば、次のようにします。

```
ldapsearch -h directory.siroe.com -s base -b
"cn=monitor,cn=Siroe,cn=ldbm database,cn=plugins, cn=config"
"objectclass=*"
```

この例では、ldapsearch 操作は Siroe データベースを対象に実行されます。ディレクトリの検索については、489 ページの「ldapsearch の使用」を参照してください。

サーバアクティビティを監視するときに、次の情報を見ることができます。

- `database` : 現在監視しているデータベースのタイプを示す
- `readonly` : データベースが読み取り専用モードになっているかどうかを示す。`0` は、読み取り専用モードではないことを示し、`1` は読み取り専用モードであることを示す
- `entrycachehits` : 405 ページの表 12-5 の「エントリキャッシュの検索ヒット数」で説明した内容と同じ情報を与える
- `entrycachetries` : 405 ページの表 12-5 の「エントリキャッシュの検索試行数」で説明した内容と同じ情報を与える
- `entrycachehitratio` : 表 12-5 の 406 ページの「エントリキャッシュの検索ヒット率」で説明した内容と同じ情報を与える
- `currententrycachesize` : 表 12-5 の 406 ページの「現在のエントリキャッシュサイズ (エントリ単位)」で説明した内容と同じ情報を与える
- `maxentrycachesize` : 表 12-5 の 406 ページの「最大エントリキャッシュサイズ (エントリ単位)」で説明した内容と同じ情報を与える

- dbchehits : 406 ページの表 12-6 の「ヒット数」で説明した内容と同じ情報を与える
- dbcachetries : 406 ページの表 12-6 の「試行数」で説明した内容と同じ情報を与える
- dbcachehitratio : 406 ページの表 12-6 の「ヒット率」で説明した内容と同じ情報を与える
- dbcachepagein : 406 ページの表 12-6 の「読み取りページ」で説明した内容と同じ情報を与える
- dbcachepageout : 406 ページの表 12-6 の「書き込みページ」で説明した内容と同じ情報を与える
- dbcacheroevict : 406 ページの表 12-6 の「破棄された読み取り専用ページ」で説明した内容と同じ情報を与える
- dbcacherwevict : 406 ページの表 12-6 の「破棄された読み取り / 書き込みページ」で説明した内容と同じ情報を与える

次に、データベースを構成する各ファイルについて、次の情報が表示されます。

- dbfilename-*number* : この属性はファイルの名前を示します。*number* は、連続した整数で表されるファイルの識別子 (0 から始まる) 。そのファイルに関連付けられるすべての統計情報にも、同じ整数の識別子が付けられる
- dbfilecachehit-*number* : 407 ページの表 12-7 の「キャッシュのヒット数」で説明した内容と同じ情報を与える
- dbfilecachemiss-*number* : 407 ページの表 12-7 の「キャッシュのミス数」で説明した内容と同じ情報を与える
- dbfilepagein-*number* : 407 ページの表 12-7 の「読み取りページ」で説明した内容と同じ情報を与える
- dbfilepageout-*number* : 407 ページの表 12-7 の「書き込みページ」で説明した内容と同じ情報を与える

データベースリンクアクティビティの監視

監視用属性を使用すると、コマンド行からデータベースリンクのアクティビティを監視できます。ldapsearch コマンド行ユーティリティを使用して、必要な属性値を返します。監視用の属性は、cn=monitor,cn=*database_link_name*,cn=chaining database,cn=plugins,cn=config に格納されます。

たとえば、ldapsearch コマンド行ユーティリティを使用すると、DBLink1 という特定のデータベースリンクが受け取った追加操作の数を調べることができます。以下のように入力して ldapsearch を実行します。

```
ldapsearch -h server.siroe.com -p 389 \
-D "cn=Directory Manager" -w password -s sub -b \
"cn=monitor,cn=DBLink1,cn=chaining database,cn=plugins,cn=config" \
"(objectclass=*)" nsAddCount
```

注 前述のコマンドは、1 行で入力してください。ここでは、ページサイズの都合上、複数の行に分けて示しています。

次のデータベースリンク監視属性を検索できます。

表 12-8 データベースリンク監視属性

属性名	説明
nsAddCount	受け取った追加操作の数
nsDeleteCount	受け取った削除操作の数
nsModifyCount	受け取った変更操作の数
nsRenameCount	受け取った名前変更操作の数
nsSearchBaseCount	受け取ったベースレベル検索の数
nsSearchOneLevelCount	受け取った 1 レベル検索の数
nsSearchSubtreeCount	受け取ったサブツリー検索の数
nsAbandonCount	受け取った中止操作の数
nsBindCount	受け取ったバインド要求の数
nsUnbindCount	受け取ったバインド解除要求の数
nsCompareCount	受け取った比較操作の数
nsOperationConnectionCount	通常操作に対して開かれた接続の数
nsBindConnectionCount	バインド操作に対して開かれた接続の数

第 13 章

SNMP を使用した Directory Server の監視

第 12 章「サーバとデータベースアクティビティの監視」に記載されている、サーバおよびデータベースアクティビティの監視ログの設定方法は、iPlanet Directory Server に固有のものです。Directory Server の監視には、簡易ネットワーク管理プロトコル (Simple Network Management Protocol) (SNMP) を使用することもできます。SNMP は、リアルタイムで各種デバイスを監視するためのネットワークアクティビティ監視用管理プロトコルです。

SNMP はグローバルなネットワーク制御と監視に最適な標準メカニズムです。SNMP により、ネットワーク管理者は、Directory Server を含むすべてのネットワークアクティビティを一括して監視することができます。

この章では、次の項目について説明します。

- SNMP について
- Directory Server MIB の概要
- SNMP の設定
- iPlanet Directory Server のための SNMP の構成

SNMP について

SNMP は、ネットワークアクティビティに関するデータを交換するために使用されるプロトコルです。SNMP を使用すると、管理対象デバイスと、ユーザがネットワークをリモート管理する Network Management Station (NMS) の間で、データが移動します。管理対象デバイスとは、ホスト、ルータ、Directory Server などの SNMP が走るあらゆるデバイスです。NMS とは通常、1 つ以上のネットワーク管理アプリケーションがインストールされた強力なワークステーションを指します。ネットワーク管理アプリケーションは、管理対象デバイスに関する情報 (どのデバイスがアクティブまたは非アクティブか、どのエラーメッセージをいくつ受け取ったか、など) をグラフィカルに表示します。

情報は、サブエージェントとマスターエージェント (master agent) の 2 種類のエージェントを使用して、NMS と管理対象デバイスの間で転送されます。サブエージェントは、管理対象デバイスに関する情報を集め、その情報をマスターエージェントに渡します。iPlanet Directory Server にはサブエージェントがあります。マスターエージェントは、各種サブエージェントと NMS の間で情報を交換します。マスターエージェントは、通信先のサブエージェントと同じホストマシン上で動作します。

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。UNIX 環境では、マスターエージェントは iPlanet Administration Server と一緒にインストールされます。

照会可能な SNMP 属性 (変数とも呼ばれる) の値は、管理対象デバイス上に保持され、必要に応じて NMS に報告されます。各変数は 管理対象オブジェクト (managed object) と呼ばれます。エージェントはこのオブジェクトにアクセス可能で、これを NMS に送ることができます。管理対象オブジェクトはすべて、ツリーのような階層を持つデータベースである管理情報ベース (management information base) (MIB) に定義されています。この階層の最上位には、ネットワークに関する一般的な情報が含まれています。下位の各階層には、個々のネットワーク領域に関するより詳細な情報が含まれています。

SNMP の概要

SNMP は、Protocol Data Unit (PDU) の形式でネットワーク情報を交換します。PDU には、管理対象デバイス上に格納された変数に関する情報が含まれています。これらの変数は管理対象オブジェクトとも呼ばれ、必要に応じて NMS に報告される値と名前を保持しています。NMS と管理対象デバイスとの間の通信は、次の 2 つのどちらかの方法で行われます。

- NMS 主導の通信
- 管理対象デバイス主導の通信

NMS 主導の通信

NMS によって開始される通信は、NMS と管理対象デバイス間の通信のうち、一般的なタイプのものです。このタイプの通信では、NMS は管理対象デバイスから情報を要求するか、または管理対象デバイス上に格納された変数の値を変更します。

NMS によって開始される SNMP セッションは、次のように実行されます。

1. NMS が、どの管理対象デバイスおよびオブジェクトに監視が必要であるかを判断する。

2. NMS が、マスターエージェントを介して、PDU を管理対象デバイスのサブエージェントに送る。この PDU は、管理対象デバイスから情報を要求するか、または管理対象デバイス上に格納された変数の値を変更するようサブエージェントに指示する。
3. 管理対象デバイスのサブエージェントが、マスターエージェントから PDU を受け取る。
4. NMS から送られた PDU が変数に関する情報を要求するものである場合、サブエージェントはマスターエージェントにその情報を送り、マスターエージェントは別の PDU として NMS に情報を送り返す。次に、NMS が、この情報を文字またはグラフィックで表示する。

   NMS から送られた PDU が、変数に値を設定するようサブエージェントに要求するものである場合、サブエージェントは変数値を要求のとおりに設定する。

管理対象デバイス主導の通信

このタイプの通信は、管理対象デバイスが発生したイベントを NMS に通知する必要があるときに発生します。管理対象デバイス主導の通信では、停止または起動の通知が、管理対象デバイスから NMS へ送られます。管理対象デバイスによって開始される通信は、トラップとも呼ばれます。Directory Server は、Directory Server が起動または停止するたびに、NMS にトラップを送ります。

管理対象デバイスによって開始される SNMP セッションは、次のように確立されます。

1. 管理対象デバイス上でイベントが発生する。
2. サブエージェントが、マスターエージェントにイベントを通知する。
3. マスターエージェントが、NMS にイベントを通知するための PDU を送る。
4. NMS が、この情報を文字またはグラフィックで表示する。

Directory Server MIB の概要

iPlanet の各サーバには、独自の MIB があります。Directory Server の MIB は、次のファイル内に定義されます。

```
/usr/iplanet/ds5/plugins/snmp/netscape-ldap.mib
```

この MIB には、Directory Server のネットワーク管理に関する変数の定義が含まれています。これらの変数は、管理対象オブジェクトと呼ばれます。Directory Server MIB および Sun Net Manager などのネットワーク管理ソフトウェアを使用すると、ネットワーク上のほかのすべての管理対象デバイスと同じようにディレクトリを監視できます。

Directory Server 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 })
```

Directory Server MIB を使用すると、Directory Server に関する管理情報を確認したり、サーバをリアルタイムで監視したりできます。Directory Server MIB は、次の 3 つの管理対象オブジェクトテーブルに分類されます。

- 処理テーブル
- エントリテーブル
- 相互作用テーブル

Directory Server MIB は使う前に、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。

```
/usr/iplanet/ds5/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。次の節では、各テーブルについて詳しく説明します。

処理テーブルについて

処理テーブルは、Directory Server のアクセス、処理、およびエラーに関する統計情報を提供します。次のテーブルには、`netscape-ldap.mib` ファイルの処理テーブルに格納されている管理対象オブジェクトが記載されています。

表 13-1 処理テーブルの管理対象オブジェクトとその説明

管理対象オブジェクト	内容
`dsAnonymousBinds`	サーバ起動以降に、Directory Server に対して行われた匿名バインドの数

表 13-1　処理テーブルの管理対象オブジェクトとその説明 (続き)

管理対象オブジェクト	内容
dsUnauthBinds	サーバ起動以降に、Directory Server に対して認証されなかったバインドの数
dsSimpleAuthBinds	サーバ起動以降に、ディレクトリに対して簡易認証方法 (パスワード保護など) を使用して確立されたバインドの数
dsStrongAuthBinds	サーバ起動以降に、ディレクトリに対して強力な認証方法 (SSL または Kerberos などの SASL 認証メカニズム) を使用して確立された、Directory Server に対するバインドの数
dsBindSecurityErrors	サーバ起動以降に、認証や資格が無効であるという理由で Directory Server によって拒否されたバインド要求の数
dsInOps	サーバ起動以降に、別の Directory Server からこの Directory Server に転送された処理の数
dsReadOps	アプリケーション起動以降に、Directory Server が行った読み取り処理の数。読み取り処理は、LDAP の検索処理によって間接的に実装されるので、このオブジェクトの値は常に 0 (ゼロ)
dsCompareOps	サーバ起動以降に、Directory Server が行なった比較処理の数
dsAddEntryOps	サーバ起動以降に、Directory Server が行なった追加処理の数
dsRemoveEntryOps	サーバ起動以降に、Directory Server が行なった削除処理の数
dsModifyEntryOps	サーバ起動以降に、Directory Server が行なった変更処理の数
dsModifyRDNOps	サーバ起動以降に、Directory Server が行なった RDN 変更処理の数
dsListOps	サーバ起動以降に、このディレクトリが行なったリスト処理の数。一覧表示処理は、LDAP の検索処理によって間接的に実装されるので、このオブジェクトの値は常に 0 (ゼロ)
dsSearchOps	サーバ起動以降に、Directory Server が行なった検索処理の合計数
dsOneLevelSearchOps	サーバ起動以降に、Directory Server が行なった 1 階層を対象にした検索処理の数

表 13-1　処理テーブルの管理対象オブジェクトとその説明 (続き)

管理対象オブジェクト	内容
dsWholeSubtreeSearchOps	サーバ起動以降に、Directory Server が行なったサブツリー全体を対象とした検索処理の数
dsReferrals	サーバ起動以降に、クライアントの要求に対して Directory Server が返したレフェラルの数
dsSecurityErrors	Directory Server に転送された処理のうち、セキュリティ要件を満たさなかった処理の数
dsErrors	エラー (セキュリティエラーおよびレフェラルエラー以外) の発生により実行されなかった要求の数。エラーには名前エラー、更新エラー、属性エラー、およびサービスエラーが含まれる。部分的に実行された要求は、エラーとしてカウントされない

エントリテーブル

エントリテーブルには、ディレクトリエントリの内容に関する情報を提供します。表 13-2 に、netscape-ldap.mib ファイルのエントリテーブルに格納されている管理対象オブジェクトを示します。

表 13-2　エントリテーブルの管理対象オブジェクトとその説明

管理対象オブジェクト	内容
dsMasterEntries	Directory Server にマスターエントリを保持している場合のディレクトリエントリの数。現在は更新処理が実行されていないので、このオブジェクトの値は常に 0 (ゼロ)
dsCopyEntries	Directory Server にスレーブコピーを保持している場合のディレクトリエントリの数。現在は更新処理が実行されていないので、このオブジェクトの値は常に 0 (ゼロ)
dsCacheEntries	Directory Server 内にキャッシュされたエントリの数
dsCacheHits	アプリケーション起動以降に、ローカルに保持されたキャッシュから実行された処理の数
dsSlaveHits	ローカルに保持されたレプリケーション (シャドウエントリ) から実行された処理の数。このオブジェクトの値は常に 0 (ゼロ)

SNMP の設定

Directory Server に対する SNMP サポートを設定するには、次の手順を実行します。

1. Administration Server Console を使用して、マスターエージェントを構成し、起動します。

注	デフォルトのポート設定 (SNMP では 161、SMUX では 199) を使用している場合は、root ユーザになる必要があります。マスターエージェントを構成し直して、ポート設定に 1000 より大きい値を指定した場合は、root ユーザになる必要はありません。

   マスターエージェントの設定については、『Managing Servers with iPlanet Console』を参照してください。

2. Directory Server のサブエージェントを有効にします。
3. Directory Server のサブエージェントを起動します。

   詳細は、417 ページの「SNMP サブエージェントの起動と停止」を参照してください。

SNMP サブエージェントの起動と停止

SNMP サブエージェント (SNMP subagent) 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 Console で「構成」タブを選択し、左側の区画のナビゲーションツリーで一番上にあるエントリを選択します。
2. 右側の区画で「SNMP」タブを選択します。
3. サブエージェント (subagent) を起動するには「起動」、停止するには「停止」、再起動するには「再起動」をクリックします。

   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。

注	別のサーバインスタンスを追加して、そのインスタンスを SNMP ネットワークの一部にする場合は、サブエージェントを再起動する必要があります。

iPlanet Directory Server のための SNMP の構成

Directory Server Console から SNMP 設定を構成するには、次の手順を実行します。

1. Directory Server が動作していることを確認します。
2. Directory Server Console で「構成」タブを選択し、左側の区画のナビゲーションツリーで一番上にあるエントリを選択します。
3. 右側の区画で「SNMP」タブを選択します。
4. Directory Server の統計収集を有効にするには、「統計収集を有効化」チェックボックスを選択します。チェックボックスの選択を解除すると、統計収集は無効になります。
5. マスターエージェントがあるホストの名前、およびマスターエージェントとの通信に使用されるポート番号を、「マスターホスト」テキストボックスと「マスターポート」テキストボックスにそれぞれ入力します。

注 ホスト名とポート番号は必須入力項目です。

デフォルトの設定は、それぞれ `localhost` と `199` です。

6. 「説明」テキストボックスに、各 Directory Server インスタンスに固有の説明を入力します。
7. 「組織」テキストボックスに、Directory Server の所属先である企業名または組織名を入力します。
8. 「位置」テキストボックスに、Directory Server がある企業または組織内の Directory Server が存在する場所を入力します。
9. 「連絡先」テキストボックスに、Directory Server を管理する責任者の電子メールアドレスを入力します。
10. 「保存」をクリックします。
11. サブエージェントを再起動します。

    詳細は、417 ページの「SNMP サブエージェントの起動と停止」を参照してください。

第 14 章

Directory Server の性能の調整

この章では、性能を最適化するための iPlanet Directory Server とともに提供されるツールについて説明します。また、ディレクトリの性能を向上させるためのヒントも示します。

この章は、次の節で構成されています。

- サーバの性能の調整
- データベースの性能の調整
- その他の調整のヒント

サーバの性能の調整

クライアントの検索要求を処理するために使用される資源の量を制限することによって、サーバの性能を管理することができます。次の項目を定義できます。

- 検索操作に応答してクライアントに返される最大エントリ数 (サイズ制限属性)
- 検索要求の実行に費やされる最大実時間 (秒) (時間制限属性)
- 接続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 (秒) (アイドルタイムアウト属性)
- Directory Server で使用可能なファイルディスクリプタの最大数 (ファイルディスクリプタの最大数属性)

性能を最適化するように Directory Server を構成するには、次の手順を実行します。

1. Directory Server Console で「構成」タブを選択し、左側の区画のナビゲーションツリーで一番上にあるエントリを選択します。

   右側の区画に表示されるタブによって、サーバ全体に影響する構成の属性を制御できます。

2. 右側の区画にある「性能」タブを選択します。

   現在のサーバ性能の設定が表示されます。

3. 「サイズ制限」テキストボックスに新しい値を入力して、検索操作への応答としてクライアントに返される最大エントリ数を設定します。

   制限を設定しない場合は、テキストボックスに -1 を入力します。

4. 「時間制限」テキストボックスに、サーバが検索要求を実行するために使用できる最大実時間 (秒) を入力します。

   制限を設定しない場合は、テキストボックスに -1 を入力します。

5. 「アイドルタイムアウト」テキストボックスに、アイドル状態の接続が切断されるまでの時間 (秒) を入力します。

   制限を設定しない場合は、テキストボックスにゼロ (0) を入力します。

6. 「ファイルディスクリプタの最大数」テキストボックスに、Directory Server で使用可能なファイルディスクリプタの最大数を入力します。

これらのパラメタがサーバの検索性能に及ぼす影響については、345 ページの「インデックスについて」を参照してください。

データベースの性能の調整

この節は、データベースの性能の調整方法について次の各項目で説明します。

検索性能の最適化

データベースの設定を調整することによって、サーバの検索性能を向上させることができます。性能に影響を与えるデータベース属性では、主にサーバで使用可能なメモリ量を定義します。

検索操作におけるキャッシュヒット率を向上させるには、Directory Server がデータベースキャッシュに維持できるデータの量を増やします。このためには、キャッシュに格納されるエントリ数とキャッシュサイズを増やします。これらの属性に対して設定できる最大値は、マシン上の実際のメモリ容量によって決まります。マシンのメモリ容量は、概算で次の値よりも大きくする必要があります。

(キャッシュ内の最大エントリ数 + 最大キャッシュサイズ) x 平均エントリサイズ

これらの 2 つの属性を変更する場合は、注意が必要です。これらの属性によってどの程度のサーバ性能の向上が望めるかは、データベースのサイズ、マシン上の使用可能な物理的メモリ容量、およびディレクトリ検索がランダムかどうか (つまり、ディレクトリクライアントの検索が、ランダムに広く分散したディレクトリデータを対象としているかどうか) によって異なります。

データベースがメモリ容量に対して大きすぎる場合や、検索がランダムである場合は、これらの属性値を大きくしても、ディレクトリ性能を向上させることはできません。逆に、これらの属性値を変更することによって、全体的な性能が低下することもあります。

次に示す属性は調整可能です。

- 他のすべてのデータベースインスタンスを管理するデータベースの属性。Directory Server Console で参照できるデータベースは、ディレクトリデータを含むデータベースと `NetscapeRoot` データベースのみ。ただし、サーバでは、これらを管理するために他のデータベースが使用される。このデータベースでは、次に示す属性を変更して性能を向上させることができる
  - すべてのデータベースで使用できるメモリの量 (最大キャッシュサイズ属性)
  - 検索要求に応答して確認される最大エントリ数 (検索制限属性)
- ディレクトリデータを格納するために使用する各データベースの属性。`NetscapeRoot` データベースのサーバ構成データを含む。これらのデータベースでは、次に示す属性を変更して性能を向上させることができる
  - サーバのメモリに保持するエントリの最大数 (最大キャッシュエントリ数属性)
  - キャッシュエントリで使用できるメモリの量 (キャッシュに使用可能なメモリ属性)

他のすべてのデータベースインスタンスに適用されるデフォルトのデータベース属性を構成するには、次の手順を実行します。

1. Directory Server Console で「構成」タブを選択し、ナビゲーションツリーの「データアイコン」を展開して、Database Settings ノードを強調表示します。

   右側の区画に「データベース」タブが表示されます。

2. 右側の区画で「LDBM プラグイン設定」タブを選択します。

   このタブには、このサーバ上に格納されているすべてのデータベースのデータベース属性が含まれています。

3. 「最大キャッシュサイズ」フィールドに、すべてのデータベースで使用できるメモリの量を入力します。
4. 検索制限フィールドに、検索要求に応答して確認される最大エントリ数を入力します。

   制限を設定しない場合は、テキストボックスに **-1** を入力します。ディレクトリマネージャとしてディレクトリにバインドする場合は、デフォルトで検索制限は無制限になり、ここでの設定に優先します。

ディレクトリデータを格納する各データベースの属性を構成するには、次の手順を実行します。

1. Directory Server Console で「構成」タブを選択し、ナビゲーションツリーの「データアイコン」を展開します。調整するデータベースの接尾辞を展開し、データベースを強調表示します。

   右側の区画に表示されるタブによって、このデータベースのパラメタ設定を制御できます。
2. 右側の区画で「データベースの設定」タブを選択します。
3. 「最大キャッシュエントリ数」フィールドに、サーバのメモリに保持するエントリ数を入力します。
4. 「キャッシュに使用可能なメモリ」フィールドに、キャッシュエントリで使用できるメモリの量を入力します。

   LDIF で非常に大きなデータベースを作成する場合は、マシンの使用可能メモリに応じて、この属性をできるだけ大きな値に設定します。このパラメタの値を大きくするほど、データベースの作成速度は高まります。

   データベースの作成が終了したら、サーバを実務の環境で稼動させる前に、このパラメタをいくらか小さな値に戻してください。

トランザクションログの調整

すべての Directory Server には、管理するすべてのデータベースに対する操作を記録するトランザクションログがあります。書き込みなどのディレクトリデータベース処理が実行されると、その処理はトランザクションログに記録されます。最高の性能を実現するために、ディレクトリはすぐには処理を開始しません。処理の内容は、処理が完了するまで Directory Server 上の一時的なメモリキャッシュに格納されます。

停電などの問題によってサーバが異常終了すると、キャッシュに格納された問題発生前のディレクトリ変更に関する情報は失われます。ただし、サーバを再起動すると、ディレクトリが自動的にエラー状態を検出し、データベーストランザクションログを使用してデータベースを復元します。

データベーストランザクションログへの記録とデータベースの復元は自動的に行われるのでユーザによる操作は必要はありませんが、一部のデータベーストランザクションログ属性を調整して、性能を最適化することができます。

警告 トランザクションログ属性は、システムの変更と診断だけを目的としています。これらの設定は、必ず iPlanet プロフェッショナルサービスまたは iPlanet サポートの指示に従って行なってください。

これらの属性や他の構成属性の設定に矛盾があると、ディレクトリの動作が不安定になることがあります。

データベーストランザクションログの保存場所の変更

デフォルトでは、データベーストランザクションログファイルは、データベースファイルとともに次のディレクトリに格納されます。

`/var/ds5/slapd-`*`serverID`*`/db`

異常終了したディレクトリデータベースの復元を補助することがトランザクションログの目的なので、データベーストランザクションログとディレクトリデータベースは別のディスクに格納することをお勧めします。また、データベーストランザクションログを別の物理ディスク上に置くと、ディレクトリの性能が向上します。

データベーストランザクションログファイルの保存場所を変更するには、次の手順を実行します。

1. Directory Server を停止します。

   手順については、35 ページの「Console からのサーバの起動と停止」を参照してください。

2. `ldapmodify` コマンド行ユーティリティを使用して、`cn=config,cn=ldbm database,cn=plugins,cn=config` エントリに `nsslapd-db-logdirectory` 属性を追加します。属性のログディレクトリは、絶対パスで指定します。

   `nsslapd-db-logdirectory` 属性の構文については、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照してください。
   `ldapmodify` の使い方については、53 ページの「ldapmodify を使用したエントリの追加と修正」を参照してください。

3. Directory Server を再起動します。

データベースのチェックポイント間隔の変更

Directory Server によって、トランザクションログに記録された処理が一定の間隔でディスクに書き込まれ、データベーストランザクションログにチェックポイントエントリが記録されます。チェックポイントエントリは、すでにディレクトリに書き込まれた変更を示すことによって、トランザクションログのどこから復元を開始するかを示し、復元処理にかかる時間を短縮します。

デフォルトでは、Directory Server は、データベーストランザクションログに 60 秒ごとにチェックポイントエントリを送るように設定されています。チェックポイント間隔を大きくすると、ディレクトリの書き込み処理の性能が向上します。ただし、チェックポイント間隔を大きくすると異常終了後のディレクトリデータベースの復元にかかる時間が長くなり、データベーストランザクションログファイルも大きくなるので、結果として必要なディスクの空き容量も大きくなります。したがって、データベースの最適化について十分な知識を持ち、変更による効果を正確に予測できる場合を除き、できるだけこの属性は変更しないでください。

サーバの実行中にチェックポイント間隔を変更するには、次の手順を実行します。

1. `ldapmodify` コマンド行ユーティリティを使用して、`cn=config,cn=ldbmdatabase,cn=plugins,cn=config` エントリに `nsslapd-db-checkpoint-interval` 属性を追加します。

   `nsslapd-db-checkpoint-interval` 属性の構文については、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照してください。`ldapmodify` の使い方については、53 ページの「ldapmodify を使用したエントリの追加と修正」を参照してください。

永続トランザクションの無効化

永続トランザクションログでは、一時的なデータベーストランザクションログも実際にディスクに書き込まれます。

永続トランザクションログが無効になっていると、すべてのディレクトリデータベース処理がデータベーストランザクションログファイルに書き込まれますが、すぐにディスクに物理的に書き込まれるわけではありません。システムのクラッシュ時に、ディレクトリの変更が論理的なデータベーストランザクションログファイルに書き込まれただけで、物理ディスクに書き込まれていない場合は、その変更を復元することはできません。永続トランザクションが無効になっていると、復元されたデータベースの内容は以前と同じものになりますが、システムのクラッシュ直前に完了した LDAP 書き込み処理の結果は反映されません。

デフォルトでは、永続トランザクションログは有効になっています。永続トランザクションログを無効にするには、次の手順を実行します。

1. Directory Server を停止します。

   手順については、35 ページの「コマンド行からのサーバの起動と停止」を参照してください。

2. ldapmodify コマンド行を使用して、cn=config,cn=ldbm database,cn=plugins,cn=config エントリに nsslapd-db-durable-transactions 属性を追加し、この属性の値を off にします。

   nsslapd-db-durable-transactions 属性の構文については、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照してください。ldapmodify の使い方については、53 ページの「ldapmodify を使用したエントリの追加と修正」を参照してください。

3. Directory Server を再起動します。

トランザクションバッチの指定

トランザクションに完全な永続性が要求されない場合に更新の性能を向上させるには、nsslapd-db-transaction-batch-val 属性を使用してトランザクションログにコミットされる前にバッチ (一括) 処理されるトランザクション数を指定します。この属性に 0 以上の値を設定すると、待ち行列内のトランザクション数が属性値に達するまでサーバによるトランザクションのコミットが遅延されます。トランザクションのバッチ処理を有効にするには、nsslapd-db-durable-transaction 属性を on に設定する必要があります。

サーバの実行時にトランザクションのバッチ処理を指定または変更するには、ldapmodify コマンド行ユーティリティを使用して、cn=config,cn=ldbm database,cn=plugins,cn=config エントリに nsslapd-db-transaction-batch-val 属性を追加します。

nsslapd-db-transaction-batch-val 属性の構文および値については、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照してください。ldapmodify の使い方については、53 ページの「ldapmodify を使用したエントリの追加と修正」を参照してください。

その他の調整のヒント

ここでは、考慮すべき性能についてのヒントおよび概念について説明します。

cn=config の下へのエントリの作成

cn=config エントリは、通常のエントリと同じように拡張性が高いデータベースではなく、単純で平面的な dse.ldif 構成ファイルに格納されます。その結果、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、性能が低下します。

性能上の理由から、単純なユーザエントリを cn=config の下に格納することはお勧めできませんが、ディレクトリマネージャまたはレプリケーションマネージャ (サプライヤバインド DN) エントリなどの特別なユーザエントリを cn=config の下に格納すると、構成情報を一元化できるため便利です。

第 15 章

Directory Server プラグインの管理

Directory Server プラグインは、サーバの機能を拡張します。iPlanet Directory Server は、ディレクトリの管理に役立つプラグインをいくつか付属しています。この章では、利用可能な各種プラグインの概要と、それらのプラグインを有効または無効にする方法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。

- サーバプラグイン機能のリファレンス
- Server Console を使用したプラグインの有効化と無効化

サーバプラグイン機能のリファレンス

次の表に、iPlanet Directory Server 5.1 で提供されているプラグインについて、構成可能なオプション、構成可能な引数、デフォルト設定、従属関係、一般的な性能関連情報、および詳細情報の参照先を示します。これらの表を利用して、プラグインによる性能とコストを比較し、導入する環境に最適な構成を選択してください。より詳細な情報が入手可能な場合は、「詳細情報」にその参照先を示します。

7 ビット検査プラグイン

プラグイン名	7-bit check (NS7bitAtt)
構成エントリの DN	`cn=7-bit check,cn=plugins,cn=config`
内容	特定の属性が 7 ビットクリーンであるかどうかを検査する
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	属性のリスト (`uid mail userpassword`) のリスト、コンマ (,)、検査を実行する接尾辞 (複数可) の順に指定する
従属変数	なし

プラグイン名	7-bit check (NS7bitAtt)
性能関連情報	なし
詳細情報	ASCII 以外の文字 (日本語など) を Directory Server で使用する場合は、このプラグインをオフにする

ACL プラグイン

プラグイン名	ACL Plugin
構成エントリの DN	`cn=ACL Plugin,cn=plugins,cn=config`
内容	ACL のアクセス検査プラグイン
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	なし
性能関連情報	なし
詳細情報	第 6 章「アクセス制御の管理」

ACL 前処理用プラグイン

プラグイン名	ACL preoperation
構成エントリの DN	`cn=ACL preoperation,cn=plugins,cn=config`
内容	ACL のアクセス検査プラグイン
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	database
性能関連情報	なし
詳細情報	第 6 章「アクセス制御の管理」

バイナリ構文プラグイン

プラグイン名	Binary Syntax
構成エントリの DN	cn=Binary Syntax,cn=plugins,cn=config
内容	バイナリデータの処理用構文
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	

論理構文プラグイン

プラグイン名	Boolean Syntax
構成エントリの DN	cn=Boolean Syntax,cn=plugins,cn=config
内容	ブール値の処理用構文
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	

大文字と小文字に差異がある文字列構文プラグイン

プラグイン名	Case Exact String Syntax
構成エントリの DN	cn=Case Exact String Syntax,cn=plugins, cn=config

プラグイン名	Case Exact String Syntax
内容	大文字と小文字に差異がある文字列の処理用構文
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	

大文字と小文字に差異がない文字列構文プラグイン

プラグイン名	Case Ignore String Syntax
構成エントリの DN	`cn=Case Ignore String Syntax,cn=plugins, cn=config`
内容	大文字と小文字に差異がない文字列の処理用構文
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	

連鎖データベースプラグイン

プラグイン名	Chaining Database
構成エントリの DN	`cn=Chaining database,cn=plugins,cn=config`
内容	DN の処理用構文
設定可能なオプション	on | off
デフォルト設定	on

プラグイン名	Chaining Database
設定可能な引数	なし
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	第 3 章「ディレクトリデータベースの構成」

サービスクラスプラグイン

プラグイン名	Class of Service
構成エントリの DN	`cn=Class of Service,cn=plugins,cn=config`
内容	エントリ間での属性の共有を可能にする
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	第 5 章「高度なエントリの管理」

国名文字列構文プラグイン

プラグイン名	国名文字列構文プラグイン
構成エントリの DN	`cn=Country String Syntax,cn=plugins,cn=config`
内容	国名の処理用構文
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	なし

プラグイン名	国名文字列構文プラグイン
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	

識別名構文プラグイン

プラグイン名	Distinguished Name Syntax
構成エントリの DN	`cn=Distinguished Name Syntax,cn=plugins, cn=config`
内容	DN の処理用構文
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	

汎用時間構文プラグイン

プラグイン名	Generalized Time Syntax
構成エントリの DN	`cn=Generalized Time Syntax,cn=plugins, cn=config`
内容	日付、時刻、およびタイムゾーンの処理用構文
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある

プラグイン名	Generalized Time Syntax
詳細情報	汎用時間文字列は、次の文字で構成される 4 桁の年、2 桁の月 (たとえば、1 月は 01)、2 桁の日、2 桁の時、2 桁の分、2 桁の秒、オプションの秒の小数部、およびタイムゾーン指定。グリニッジ標準時を意味する、Z タイムゾーン指定の使用を強く推奨

整数構文プラグイン

プラグイン名	Integer Syntax
構成エントリの DN	`cn=Integer Syntax,cn=plugins,cn=config`
内容	整数の処理用構文
設定可能なオプション	on | off
デフォルト設定	`on`
設定可能な引数	なし
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	

国際化プラグイン

プラグイン名	Internationalization Plugin
構成エントリの DN	`cn=Internationalization Plugin,cn=plugins, cn=config`
内容	DN の処理用構文
設定可能なオプション	on | off
デフォルト設定	`on`
設定可能な引数	国際化プラグインは、引数を 1 つとる。この引数は修正してはならない `/var/ds5/slapd-`*`serverID`*`/config/slapd-collations.conf` このディレクトリには、国際化プラグインが使用する照合順序とロケールが格納される

プラグイン名	Internationalization Plugin
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	付録 D「国際化」を参照

ldbm データベースプラグイン

プラグイン名	ldbm database Plug-in
構成エントリの DN	`cn=ldbm database plug-in,cn=plugins,cn=config`
内容	ローカルデータベースを実装する
設定可能なオプション	なし
デフォルト設定	on
設定可能な引数	なし
従属変数	なし
性能関連情報	データベースプラグインの属性の詳細については、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照
詳細情報	第 3 章「ディレクトリデータベースの構成」

古いバージョンのレプリケーションプラグイン

プラグイン名	Legacy Replication plug-in
構成エントリの DN	`cn=Legacy Replication plug-in,cn=plugins, cn=config`
内容	iPlanet Directory Server 5.1 が、バージョン 4.1 のサプライヤのコンシューマとして動作できるようにする
設定可能なオプション	on	off
デフォルト設定	on
設定可能な引数	なし。サーバがバージョン 4.1 サーバのコンシューマではない (また、今後もそうする予定がない) 場合は、このプラグインを無効にできる
従属変数	database
性能関連情報	なし

プラグイン名	Legacy Replication plug-in
詳細情報	第 8 章「レプリケーションの管理」

マルチマスターレプリケーションプラグイン

プラグイン名	Multimaster Replication Plugin
構成エントリの DN	`cn=Multimaster Replication plugin,cn=plugins,cn=config`
内容	2 つのバージョン 5.0 の Directory Server 間でのレプリケーションを有効にする
設定可能なオプション	on	off
デフォルト設定	`on`
設定可能な引数	なし
従属変数	database
性能関連情報	なし
詳細情報	サーバが 1 つしかなく、そのサーバでレプリケーションを行なわない場合は、このプラグインをオフにできる。第 8 章「レプリケーションの管理」も参照

8 進文字列構文プラグイン

プラグイン名	Octet String Syntax
構成エントリの DN	`cn=Octet String Syntax,cn=plugins,cn=config`
内容	8 進文字列の処理用構文
設定可能なオプション	on	off
デフォルト設定	`on`
設定可能な引数	なし
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	

CLEAR パスワード保存スキーマプラグイン

プラグイン名	CLEAR
構成エントリの DN	`cn=CLEAR,cn=Password Storage Schemes,cn=plugins,cn=config`
内容	パスワードの暗号化に使用される、CLEAR パスワード保存スキーマ
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	第 7 章「ユーザアカウントの管理」

CRYPT パスワード保存スキーマプラグイン

プラグイン名	CRYPT
構成エントリの DN	`cn=CRYPT,cn=Password Storage Schemes,cn=plugins,cn=config`
内容	パスワードの暗号化に使用される、CRYPT パスワード保存スキーマ
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	第 7 章「ユーザアカウントの管理」

NS-MTA-MD5 パスワードの保存スキーマプラグイン

プラグイン名	NS-MTA-MD5
構成エントリの DN	`cn=NS-MTA-MD5,cn=Password Storage Schemes,cn=plugins,cn=config`
内容	パスワードの暗号化に使用される NS-MTA-MD5 パスワード保存スキーマ
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。iPlanet ではこのプラグインを常に実行しておくことを推奨している
詳細情報	NS-MTA-MD5 パスワード保存スキーマを使用したパスワードの暗号化は、選択できなくなった。この保存スキーマは存在はするが、Directory Server の以前のバージョンと iPlanet Directory Server 5.1 の下位互換性のためにのみ用意されている。第 7 章「ユーザアカウントの管理」を参照

SHA パスワード保存スキーマプラグイン

プラグイン名	SHA
構成エントリの DN	`cn=SHA,cn=Password Storage Schemes, cn=plugins,cn=config`
内容	パスワードの暗号化に使用される SHA パスワード保存スキーマ
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	なし
性能関連情報	SHA パスワード保存スキーマを使用して暗号化されたパスワードがディレクトリに含まれていない場合は、このプラグインをオフにできる。SSHA の方がより安全性が高いので、SHA より SSHA を選択することを推奨する
詳細情報	第 7 章「ユーザアカウントの管理」

SSHA パスワード保存スキーマプラグイン

プラグイン名	SSHA
構成エントリの DN	cn=SSHA,cn=Password Storage Schemes, cn=plugins,cn=config
内容	パスワードの暗号化に使用される SSHA パスワード保存スキーマ
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	第 7 章「ユーザアカウントの管理」

住所文字列構文プラグイン

プラグイン名	Postal Address Syntax
構成エントリの DN	cn=Postal Address Syntax,cn=plugins,cn=config
内容	住所の処理用構文
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	

PTA (パススルー認証) プラグイン

プラグイン名	Pass-Through Authentication Plugin
構成エントリの DN	cn=Pass Through Authentication,cn=plugins, cn=config

プラグイン名	Pass-Through Authentication Plugin
内容	パススルー認証 (バインド要求を認証するために、1 つのディレクトリから別のディレクトリへの参照を可能にするメカニズム) を有効にする。ユーザディレクトリと構成ディレクトリを同一のサーバで使用する場合は、Directory Server Console 内にこのプラグインがリストされない
設定可能なオプション	on	off
デフォルト設定	`off`
設定可能な引数	`ldap://iplanet.com:389/o=iplanet`
従属変数	なし
性能関連情報	第 16 章「パススルー認証プラグインの使用」
詳細情報	第 16 章「パススルー認証プラグインの使用」

レフェラル整合性の後処理用プラグイン

プラグイン名	Referential Integrity Postoperation
構成エントリの DN	`cn=Referential Integrity Postoperation,cn=plugins, cn=config`
内容	サーバによる参照整合性の確認を有効にする
設定可能なオプション	すべての構成、および on	off
デフォルト設定	off

プラグイン名	Referential Integrity Postoperation
設定可能な引数	有効に設定されている場合、後処理用の参照整合性プラグインは、削除処理または名前変更処理のあと、member、uniquemember、owner、および seeAlso 属性に対する整合性更新をただちに実行する。このプラグインを再構成することにより、ほかのすべての属性に対しても整合性検査を実行できる 設定可能な引数は次のとおり 1. 参照整合性の検査 -1 = 参照整合性を検査しない 0 = 参照整合性検査をただちに実行する 正の整数を指定すると、参照整合性の要求がキューに入れられ、後で処理される。指定した整数に対応する時間間隔で、要求を処理するスレッドが呼び出される 2. 変更履歴の格納に使用される、/var/ds5/slapd-*serverID*/logs/referint などのログファイル 3. 参照整合性検査をしたいすべての追加する属性の名前
従属変数	database
性能関連情報	マルチマスターレプリケーション環境では、際限ない競合状態が続くのを回避するため、参照整合性プラグインをオンにするのは 1 つのマスターだけにすること。連鎖バインドされたサーバでこのプラグインをオンにするときは、整合性の要件だけでなく、性能、資源、および時間に関する要件も必ず分析すること
詳細情報	66 ページの「参照整合性の管理」を参照

レトロ履歴ログプラグイン

プラグイン名	Retro Changelog Plugin
構成エントリの DN	cn=Retro Changelog Plugin,cn=plugins,cn=config
内容	バージョン 4.x の Directory Server とのアプリケーション互換性を維持するために LDAP クライアントが使用する。Directory Server で発生するすべての変更のログを管理する。逆更新履歴ログは、Directory Server の更新履歴ログと同じ機能を提供する
設定可能なオプション	on | off
デフォルト設定	off

プラグイン名	Retro Changelog Plugin
設定可能な引数	レトロログのプラグインの 2 つの構成可能な属性については、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』を参照
従属変数	なし
性能関連情報	Directory Server の性能を低下させる可能性がある
詳細情報	第 8 章「レプリケーションの管理」

ロールプラグイン

プラグイン名	Roles Plugin
構成エントリの DN	`cn=Roles Plugin,cn=plugins,cn=config`
内容	Directory Server でロールを使用できるようにする
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	第 5 章「高度なエントリの管理」

電話番号構文プラグイン

プラグイン名	Telephone Syntax
構成エントリの DN	`cn=Telephone Syntax,cn=plugins,cn=config`
内容	電話番号の処理用構文
設定可能なオプション	on | off
デフォルト設定	on
設定可能な引数	なし
従属変数	なし

プラグイン名	Telephone Syntax
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	

uid 一意性検査プラグイン

プラグイン名	UID Uniqueness plug-in
構成エントリの DN	`cn=UID Uniqueness,cn=plugins,cn=config`
内容	エントリ上で変更が発生するたびに、指定した属性の値が一意であるかどうかを検査する
設定可能なオプション	on | off
デフォルト設定	`off`

設定可能な引数	指定したすべてのサブツリーで uid 属性の一意性を検査する場合は、次の引数を入力する
	`uid`
	`"DN"`
	`"DN"...`
	ただし、リスト上のすべてのサブツリー内で uid 属性の一意性を検査する場合は、次の引数を入力するときは注意すること
	`attribute="uid"`
	`MarkerObjectclass = "ObjectClassName"`
	また、オプションとして次の引数も入力できる
	`requiredObjectClass = "ObjectClassName"`
	これらの引数は、`MarkerObjectClass` 属性で定義されている `ObjectClass` を含む親エントリから開始して、`requiredObjectClass` エントリの追加または更新を行うときに、uid 属性の一意性を検査する場合に入力する

従属変数	なし

プラグイン名	UID Uniqueness plug-in
性能関連情報	このプラグインを使用すると、Directory Server の性能を低下させることがある マルチマスターレプリケーション環境では、uid 一意性検査プラグインはまったく機能しないので、無効にする必要がある uid 一意性検査プラグインが有効で、サブツリー内にレフェラルが作成されているサーバに新しいエントリを追加する場合は、uid 一意性検査プラグインは機能しない。一意性検査プラグインが機能しない原因は、レフェラルが作成されているときに、エントリが存在しないことを意味する `noSuchObject` 以外のエラーをプラグインが検出し、新しいエントリを追加できないようにプラグインが操作エラーを返すためである。このような操作エラーによって操作がブロックされないようにするには、レフェラルを作成したサーバ上のプラグインを無効にする。ただし、それでも uid 一意性検査を実行する必要がある場合は、最後に参照されるサーバ上でだけプラグインを有効にし、プラグインによってレフェラルメカニズムがブロックされないようにする
詳細情報	第 17 章「属性一意性検査プラグインの使い方」

URI プラグイン

プラグイン名	URI Syntax
構成エントリの DN	`cn=URI Syntax,cn=plugins,cn=config`
内容	URL (Unique Resource Locator) を含む URI (Unique Resource Identifier) の処理用構文
設定可能なオプション	on | off
デフォルト設定	`on`
設定可能な引数	なし
従属変数	なし
性能関連情報	このプラグインの構成は修正しないこと。また、このプラグインは、常に実行しておく必要がある
詳細情報	

Server Console を使用したプラグインの有効化と無効化

Directory Server Console を使用し、LDAP を介してプラグイン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 Console で、「構成」タブを選択します。
2. ナビゲーションツリー内の Plug-ins フォルダをダブルクリックします。
3. プラグインのリストからプラグインを選択します。
4. プラグインを無効にするには、「有効」チェックボックスの選択を解除します。プラグインを有効にするには、このチェックボックスを選択します。
5. 「保存」をクリックします。
6. Directory Server を再起動します。

第 16 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 Directory Server が別の Directory Server 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、ローカルデータベー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、Directory Server で受け入れることができるようになります。

iPlanet Directory Server 5.1 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構成ディレクトリを管理できるようになります。

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 5.1 での PTA の使用
- PTA プラグインの構文
- PTA プラグインの構成
- PTA プラグインの構文例

Directory Server 5.1 での PTA の使用

構成ディレクトリとユーザディレクトリを Directory Server の別のインスタンスにインストールした場合は、構成の管理者 (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` ユーザのエントリが構成ディレクトリ内の `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、ユーザディレクトリが、資格情報を構成ディレクトリに転送できるようになります。続けて、構成ディレクトリで、資格が検証されます。検証が完了すると、ユーザディレクトリは、`admin` ユーザによるバインドを許可します。

この例のユーザディレクトリは、PTA directory server として機能します。つまり、バインド要求をほかの Directory Server にパススルーするサーバです。構成ディレクトリは、認証ディレクトリとして機能します。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です。

この章では、パススルー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 ディレクトリ上に存在しないサブツリーです。ユーザのバインド DN にこのサブツリーが含まれている場合は、ユーザの資格情報が認証ディレクトリに渡されます。

注	ユーザディレクトリと構成ディレクトリを同じサーバ上に置いた場合は、Directory Server Console 内に PTA プラグインは表示されません。

次に、パススルー認証のしくみについて説明します。

1. Configuration Directory Server (認証ディレクトリ) をマシン A にインストールします。
   - サーバ名 : **configdir.siroe.com**
   - 接尾辞 : **o=NetscapeRoot**
2. ユーザ Directory Server (PTA ディレクトリ) をマシン B にインストールします。
   - サーバ名 : **userdir.siroe.com**
   - 接尾辞 : **dc=siroe,dc=com**
3. マシン B にユーザディレクトリをインストールするときに、LDAP URL を指定するように要求されます。この URL がマシン A の構成ディレクトリを示します。
4. インストールプログラムによってユーザディレクトリ上の `dse.ldif` ファイルにエントリが追加されます。これで PTA プラグインが有効になります。

   このエントリは、指定した LDAP URL を含んでいます。たとえば、次のようにします。

```
dn: cn=Pass Through Authentication,cn=plugins,
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: Pass Through Authentication
nsslapd-pluginPath: /usr/iplanet/ds5/lib/passthru-plugin.so
nsslapd-pluginInitfunc: passthruauth_init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled:on
nsslapd-pluginarg0: ldap://config.siroe.com/ou=NetscapeRoot
nsslapd-plugin-depends-on-type:database
```

```
nsslapd-pluginId: passthruauth
nsslapd-pluginVersion: 5.0
nsslapd-pluginVendor:Sun | Netscape Alliance
nsslapd-pluginDescription: pass through authentication plugin
```

これで、DN に o=NetscapeRoot が含まれるエントリに対するバインド要求がすべて構成ディレクトリ configdir.siroe.com に送信されるようにユーザディレクトリが構成されます。

5. インストールの完了後に、admin ユーザでユーザディレクトリに接続して、ユーザの追加を試みます。
6. 設定プログラムによって、admin ユーザのエントリが uid=admin, ou=TopologyManagement,o=NetscapeRoot としてディレクトリに追加されます。これによって、ユーザディレクトリは、PTA プラグインの構成で定義されたとおりに、バインド要求を構成ディレクトリにパススルーします。
7. 構成ディレクトリは、ユーザの資格情報を認証し、その情報をユーザディレクトリに返します。
8. ユーザディレクトリは、admin ユーザのバインドを許可します。

PTA プラグインの構文

PTA プラグインの構成情報は、PTA ディレクトリ (バインド要求を認証ディレクトリにパススルーするように構成されたユーザディレクトリ) 上の dse.ldif ファイル内にある cn=Pass Through Authentication,cn=plugins,cn=config エントリ内に指定します。このとき、この節で説明する構文を使用します。

```
dn: cn=Pass Through Authentication,cn=plugins, cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: Pass Through Authentication
nsslapd-pluginPath: /usr/iplanet/ds5/lib/passthru-plugin.extension
nsslapd-pluginInitfunc: passthruauth_init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled: state
nsslapd-pluginarg0: ldap|ldaps://authDS/subtree [maxconns,maxops,timeout,ldver,connlifetime]
```

PTA プラグイン構文の変数コンポーネントについては、表 16-1 を参照してください。

注

- LDAP URL (ldap|ldaps://*authDS*/*subtree*) と省略可能なパラメタ (*maxconns*、*maxops*、*timeout*、*ldver*、*connlifetime*) の間は、1 つの空白で区切る必要があります。
- 省略可能なパラメタを明示的に定義する場合は、必要な指定が 1 つのデフォルト値だけであっても、すべての値を定義する必要があります。
- `nsslapd-pluginarg` 属性の接尾辞をそれぞれの回で 1 ずつ増やすことによって、複数の認証ディレクトリやサブツリーを指定できます。詳細は、455 ページの「複数の Authenticating Directory Server の指定」を参照してください。

次の表に、省略可能なパラメタを示します。表内の順序は、構文で使用するときの順序と同じです。

表 16-1 PTA プラグインのパラメタ

変数	定義
state	プラグインが有効か、または無効かを指定する。指定できる値は **`on`** または **`off`**。詳細は、450 ページの「プラグインのオンとオフの切り替え」を参照
extension	プラグインのファイル拡張子。拡張子は常に `.so` になる
ldap|ldaps	Directory Server 間の通信に SSL が使用されるかどうかを定義する。詳細は、451 ページの「セキュリティ保護された接続を使用するためのサーバの構成」を参照
authDS	認証ディレクトリのホスト名。コロンの頃哉にポート番号を指定することによって、Directory Server のポート番号を指定できる。たとえば、次のようにする `ldap://dirserver.siroe.com:390/` ポート番号を指定しない場合は、PTA サーバによって、次のポート番号を使用して接続が試行される • URL 内で `ldap://` が指定されている場合は、ポート 389 • URL 内で `ldaps://` が指定されている場合は、ポート 636 詳細は、452 ページの「Authenticating Directory Server の指定」を参照
subtree	パススルーサブツリー (pass-through subtree) PTA Directory Server は、このサブツリー内に含まれる DN を持つクライアントすべてからのバインド要求をすべて認証ディレクトリにパススルーする 詳細は、453 ページの「パススルーサブツリーの指定」を参照

表 16-1　PTA プラグインのパラメタ (続き)

変数	定義
maxconns	省略可能。PTA ディレクトリが認証ディレクトリに対して同時に開くことができる接続の最大数。デフォルトは 3 詳細は、453 ページの「省略可能なパラメタの構成」を参照
maxops	省略可能。単一の接続中に、PTA ディレクトリが認証ディレクトリに同時に送信できる処理 (通常はバインド要求) の最大数。デフォルトは 5 詳細は、453 ページの「省略可能なパラメタの構成」を参照
timeout	省略可能。PTA ディレクトリが Authenticating Directory Server からの応答を待機できる制限時間 (秒)。このタイムアウトを超えると、サーバはクライアントにエラーを返す デフォルトは 300 秒 (5 分)。制限時間を設定しない場合は 0 を指定する 詳細は、453 ページの「省略可能なパラメタの構成」を参照
ldver	省略可能。認証ディレクトリに接続するために使用される LDAP プロトコルのバージョン。iPlanet Directory Server は、LDAP バージョン 2 および 3 をサポートする 詳細は、453 ページの「省略可能なパラメタの構成」を参照
connlifetime	省略可能。接続使用の制限時間 (秒)。この制限時間が経過したあとに、クライアントからバインド要求が開始された場合、サーバはいったん接続を切断してから、認証ディレクトリへの新しい接続を開く。バインド要求が開始され、ディレクトリによって接続制限時間を超過していると判断されない限り、サーバは接続を切断しない このオプションを指定しない場合、または一覧表示されているホストが 1 つだけの場合は、接続の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 詳細は、453 ページの「省略可能なパラメタの構成」を参照

PTA プラグインの構成

PTA プラグインを構成する唯一の方法は、dse.ldif ファイル内のエントリ cn=Pass Through Authentication,cn=plugins,cn=config を変更することです。dse.ldif ファイルを修正するには、次の手順を実行します。

1. ldapmodify コマンドを使用して、cn=Pass Through Authentication,cn=plugins,cn=config を修正します。
2. Directory Server を再起動します。

この節で説明したいずれかのパラメタを構成するには、dse.ldif ファイル内に PTA プラグインのエントリが含まれている必要があります。このエントリがない場合は、447 ページの「PTA プラグインの構文」で説明しているように、適切な構文を使用してエントリを作成する必要があります。

注 ユーザディレクトリと構成ディレクトリを別のディレクトリの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ディレクトリの dse.ldif ファイルに追加されます。ユーザディレクトリと構成ディレクトリを同じインスタンスにインストールした場合は、PTA プラグイン構文が自動的には追加されないので、手動でこの構文を追加する必要があります。

ここでは、プラグインの構成について説明します。

- プラグインのオンとオフの切り替え
- セキュリティ保護された接続を使用するためのサーバの構成
- Authenticating Directory Server の指定
- パススルーサブツリーの指定
- 省略可能なパラメタの構成

プラグインのオンとオフの切り替え

コマンド行から PTA プラグインを有効にするには、次の手順を実行します。

1. 次の LDIF 更新文を含む LDIF ファイルを作成します。

```
dn: cn=Pass Through Authentication,cn=plugins, cn=config
cn: Pass Through Authentication
changetype: modify
replace: nsslapd-pluginenabled
nsslapd-pluginenabled:on
```

2. ldapmodify コマンドを使用して、LDIF ファイルをディレクトリにインポートします。
3. プラグインを有効にする場合は、プラグインの初期化機能が適切に定義されていることを確認する必要もあります。

   エントリ cn=Pass Through Authentication,cn=plugins,cn=config に次の属性 - 属性値のペアが含まれている必要があります。

```
nsslapd-pluginPath: /usr/iplanet/ds5/lib/passthru-plugin.so
nsslapd-pluginInitfunc: passthruauth_init
```

4. サーバを再起動します。

   サーバの再起動については、35 ページの「iPlanet Directory Server の起動と停止」を参照してください。

プラグインを無効にするには、LDIF 更新文を nsslapd-pluginenabled: on 文を削除し、nsslapd-pluginenabled: off 文を追加するように変更します。コマンド行から PTA プラグインを有効または無効にした場合は、サーバを再起動する必要があります。

セキュリティ保護された接続を使用するためのサーバの構成

PTA ディレクトリが SSL を介して認証ディレクトリと通信するように構成することができます。このためには、PTA ディレクトリの LDAP URL 内に LDAPS を指定します。

SSL を使用するように PTA ディレクトリと認証ディレクトリを構成するには、次の手順を実行します。

1. 次の LDIF 更新文を含む LDIF ファイルを作成します。

```
dn: cn=Pass Through Authentication,cn=plugins, cn=config
cn: Pass Through Authentication
changetype: modify
replace: nsslapd-pluginarg0
nsslapd-pluginarg0: ldaps://authDS/subtree [optional_parameters]
```

   この構文中の変数コンポーネントについては、448 ページの「PTA プラグインのパラメタ」を参照してください。

2. ldapmodify コマンドを使用して、LDIF ファイルをディレクトリにインポートします。

3. サーバを再起動します。

   サーバの再起動については、35 ページの「iPlanet Directory Server の起動と停止」を参照してください。

Authenticating Directory Server の指定

認証ディレクトリには、エントリのバインド資格が含まれています。クライアントはこのバインド資格を使用してバインドを試行します。PTA ディレクトリは、認証ディレクトリとして定義されたホストにバインド要求を渡します。認証ディレクトリサーバを指定するには、PTA ディレクトリの LDAP URL 内にある *authDS* を認証ディレクトリのホスト名に置き換えます。

PTA の認証ディレクトリを指定するには、次の手順を実行します。

1. 次の LDIF 更新文を含む LDIF ファイルを作成します。

```
dn: cn=Pass Through Authentication,cn=plugins, cn=config
cn: Pass Through Authentication
changetype: add
add: nsslapd-pluginarg0
nsslapd-pluginarg0: ldap://authDS/subtree [optional_parameters]
```

   オプションで、コロン (:) の頃哉にポート番号を指定できます。ポート番号を指定しない場合は、PTA ディレクトリによって、次のポート番号を使用して接続が試行されます。

   - URL 内で ldap:// が指定されている場合は、ポート 389
   - URL 内で ldaps:// が指定されている場合は、ポート 636

   たとえば、`nsslapd-pluginarg0` 属性値を次のように設定できます。

   "ldap://dirserver.siroe.com:389/*subtree* [*Parameters*]"

   この構文中の変数コンポーネントについては、448 ページの「PTA プラグインのパラメタ」を参照してください。

2. `ldapmodify` コマンドを使用して、LDIF ファイルをディレクトリにインポートします。

3. サーバを再起動します。

   サーバの再起動については、35 ページの「iPlanet Directory Server の起動と停止」を参照してください。

パススルーサブツリーの指定

PTA ディレクトリは、パススルーサブツリー (pass-through subtree) 内に定義された DN を持つクライアントからのバインド要求を認証ディレクトリにパススルーします。サブツリーを指定するには、PTA ディレクトリの LDAP URL 内にある *subtree* パラメタを置き換えます。

パススルーサブツリーは、PTA ディレクトリ内に存在することはできません。PTA ディレクトリに内に存在している場合は、PTA ディレクトリが自身のコンテンツを使用してバインド要求を解決しようとするので、バインドが失敗します。

パススルーサブツリーを指定するには、次の手順を実行します。

1. 次の LDIF 更新文を含む LDIF ファイルを作成します。

```
dn: cn=Pass Through Authentication,cn=plugins, cn=config
cn: Pass Through Authentication
changetype: add
add: nsslapd-pluginarg0
nsslapd-pluginarg0: ldap://authDS/subtree [optional_parameters]
```

   たとえば、`nsslapd-pluginarg0` 属性値を次のように設定できます。

```
"ldap://dirserver.siroe.com/o=NetscapeRoot [Parameters]"
```

   この構文中の変数コンポーネントについては、448 ページの「PTA プラグインのパラメタ」を参照してください。

2. `ldapmodify` コマンドを使用して、LDIF ファイルをディレクトリにインポートします。

3. サーバを再起動します。

   サーバの再起動については、35 ページの「iPlanet Directory Server の起動と停止」を参照してください。

省略可能なパラメタの構成

PTA プラグインに対して、次の省略可能なパラメタを構成できます。

- PTA ディレクトリが認証ディレクトリに対して同時に開くことができる接続の最大数。PTA 構文内の *maxconns* で指定する。デフォルト値は 3
- 単一の接続中に、PTA Directory Server が Authenticating Directory Server に同時に送信できるバインド要求の最大数。PTA 構文内では、このパラメタは *maxops* として指定する。デフォルト値は 5

- PTA Directory Server が Authenticating Directory Server からの応答を待機できる制限時間。PTA 構文内では、このパラメタは *timeout* として指定する。デフォルト値は 300 秒 (5 分)
- PTA Directory Server が Authenticating Directory Server に接続するために使用する LDAP プロトコルのバージョン。PTA 構文内では、このパラメタは *ldver* として指定する。デフォルトは LDAPv3
- 接続使用の制限時間 (秒)。この制限時間が経過した後に、クライアントからバインド要求が開始された場合、サーバはいったん接続を切断してから、Authenticating Directory Server への新しい接続を開く。バインド要求が開始され、サーバ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 *authDS* パラメタ内でリストされた認証ディレクトリサーバが 1 つだけの場合は、制限時間は強制されない。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 構文内では、このパラメタは *connlifetime* として指定する

注 以上のパラメタは省略可能です。ただし、1 つでもパラメタを指定する場合は、その他のパラメタにデフォルト値を使用するのであっても、すべてのパラメタの指定が必要です。

1. 次の LDIF 更新文を含む LDIF ファイルを作成します。

```
dn: cn=Pass Through Authentication,cn=plugins, cn=config
cn: Pass Through Authentication
changetype: add
add: nsslapd-pluginarg0
nsslapd-pluginarg0: ldap://authDS/subtree [maxconns,maxops,timeout,ldver,connlifetime]
```

   subtree パラメタと省略可能なパラメタの間に空白が挿入されていることを確認します。

   たとえば、nsslapd-pluginarg0 属性値を次のように設定できます。

   "ldap://dirserver.siroe.com/o=NetscapeRoot 3,5,300,3,300"

   この例では、省略可能な各パラメタはそれぞれのデフォルト値に設定されています。

2. ldapmodify コマンドを使用して、LDIF ファイルをディレクトリにインポートします。
3. サーバを再起動します。

   サーバの再起動については、35 ページの「iPlanet Directory Server の起動と停止」を参照してください。

PTA プラグインの構文例

この節では、dse.ldif ファイル内の PTA プラグインの次のような構文例を示します。

- 1 つの Authenticating Directory Server と 1 つのサブツリーの指定
- 複数の Authenticating Directory Server の指定
- 1 つの Authenticating Directory Server と複数のサブツリーの指定
- デフォルト以外のパラメタ値の使用
- Authenticating Directory Server ごとに異なる省略可能パラメタとサブツリーの指定

1 つの Authenticating Directory Server と 1 つのサブツリーの指定

この例では、省略可能な変数のデフォルトをすべてそのまま使用して PTA プラグインを構成します。この構成によって、PTA Directory Server は、o=NetscapeRoot サブツリーへのバインド要求があるたびに Authenticating Directory Server に接続します。Authenticating Directory Server のホスト名は、config-dir.siroe.com です。

```
dn: cn=Pass Through Authentication,cn=plugins, cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: Pass Through Authentication
nsslapd-pluginPath: /usr/iplanet/ds5/lib/passthru-plugin.so
nsslapd-pluginInitfunc: passthruauth_init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled:on
nsslapd-pluginarg0: ldap://config-dir.siroe.com/ou=NetscapeRoot
nsslapd-plugin-depends-on-type:database
nsslapd-pluginId: passthruauth
nsslapd-pluginVersion: 5.0
nsslapd-pluginVendor:Sun | Netscape Alliance
nsslapd-pluginDescription: pass through authentication plugin
```

複数の Authenticating Directory Server の指定

PTA Directory Server と Authenticating Directory Server の間の接続が切断された場合や接続を開けない場合に、次に指定されたサーバがあるときは、PTA Directory Server は、そのサーバに要求を送信します。Authenticating Directory Server は、必要な数だけ指定できます。

```
dn: cn=Pass Through Authentication,cn=plugins, cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
```

```
cn: Pass Through Authentication
nsslapd-pluginPath: /usr/iplanet/ds5/lib/passthru-plugin.so
nsslapd-pluginInitfunc: passthruauth_init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled:on
nsslapd-pluginarg0: ldap://config-dir.siroe.com/ou=NetscapeRoot
nsslapd-pluginarg1: ldap://config2-dir.siroe.com/ou=NetscapeRoot
nsslapd-plugin-depends-on-type:database
nsslapd-pluginId: passthruauth
nsslapd-pluginVersion: 5.0
nsslapd-pluginVendor:Sun | Netscape Alliance
nsslapd-pluginDescription: pass through authentication plugin
```

1 つの Authenticating Directory Server と複数のサブツリーの指定

次の例では、パラメタのデフォルト値を使用して複数のサブツリーに対するバインド要求をパススルーするように、PTA Directory Server を構成します。

```
dn: cn=Pass Through Authentication,cn=plugins, cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: Pass Through Authentication
nsslapd-pluginPath: /usr/iplanet/ds5/lib/passthru-plugin.so
nsslapd-pluginInitfunc: passthruauth_init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled:on
nsslapd-pluginarg0: ldap://config-dir.siroe.com/ou=NetscapeRoot
nsslapd-pluginarg1: ldap://config-dir.siroe.com/dc=siroe,dc=com
nsslapd-plugin-depends-on-type:database
nsslapd-pluginId: passthruauth
nsslapd-pluginVersion: 5.0
nsslapd-pluginVendor:Sun | Netscape Alliance
nsslapd-pluginDescription: pass through authentication plugin
```

デフォルト以外のパラメタ値の使用

この例では、接続の最大数を示すパラメタ *maxconns* だけにデフォルト以外の値 (10) を使用します。ほかの各パラメタは、デフォルト値に設定されています。ただし、1 つのパラメタを明示的に指定しているため、すべてのパラメタを構文内で明示的に指定する必要があります。

```
dn: cn=Pass Through Authentication,cn=plugins, cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: Pass Through Authentication
nsslapd-pluginPath: /usr/iplanet/ds5/lib/passthru-plugin.so
nsslapd-pluginInitfunc: passthruauth_init
```

```
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled:on
nsslapd-pluginarg0: ldap://config-dir.siroe.com/ou=NetscapeRoot 10,5,300,3,300
nsslapd-plugin-depends-on-type:database
nsslapd-pluginId: passthruauth
nsslapd-pluginVersion: 5.0
nsslapd-pluginVendor:Sun | Netscape Alliance
nsslapd-pluginDescription: pass through authentication plugin
```

Authenticating Directory Server ごとに異なる省略可能パラメタとサブツリーの指定

Authenticating Directory Server ごとに異なるパススルーサブツリーと省略可能パラメタを指定する場合は、複数の LDAP URL と省略可能パラメタのペアを指定する必要があります。LDAP URL と省略可能パラメタのペアは、次のように 1 つの空白で区切ります。

```
dn: cn=Pass Through Authentication,cn=plugins, cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: Pass Through Authentication
nsslapd-pluginPath: /usr/iplanet/ds5/lib/passthru-plugin.so
nsslapd-pluginInitfunc: passthruauth_init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled:on
nsslapd-pluginarg0: ldap://config-dir.siroe.com/ou=NetscapeRoot 7.7,300,3,300
nsslapd-pluginarg1: ldap://config2-dir.siroe.com/dc=siroe,dc=com 7,7,300,3,300
nsslapd-plugin-depends-on-type:database
nsslapd-pluginId: passthruauth
nsslapd-pluginVersion: 5.0
nsslapd-pluginVendor:Sun | Netscape Alliance
nsslapd-pluginDescription: pass through authentication plugin
```

第 17 章

属性一意性検査プラグインの使い方

属性一意性検査プラグインを使用すると、指定した属性に対して、ディレクトリ内で常に一意の値を持たせることができます。一意の値を持たせる属性ごとに、プラグインのインスタンスを新しく作成する必要があります。

iPlanet Directory Server 5.1 では、uid 属性の一意性を管理するための uid 一意性検査プラグインを使用できます。

この章では、属性一意性検査プラグインおよび uid 一意性検査プラグインについて、次の項目ごとに説明します。

- 属性一意性検査プラグインの概要
- uid 一意性検査プラグインの概要
- 属性一意性検査プラグインの構文
- 属性一意性検査プラグインのインスタンスの作成
- 属性一意性検査プラグインの構成
- 属性一意性検査プラグインの構文例
- レプリケーションと属性一意性検査プラグイン

属性一意性検査プラグインの概要

属性一意性検査プラグインは、前処理用のプラグインです。つまり、LDAP 処理が実行される前に、プラグインによってすべての更新操作が検査されます。プラグインによって、その処理が監視用に構成された属性および接尾辞に適用されるものであるかどうかが判別されます。

属性一意性検査プラグインによって監視されている属性や接尾辞に対して更新操作が実行され、2 つのエントリが同じ属性値を持った場合、サーバは処理を終了し、クライアントに `LDAP_CONSTRAINT_VIOLATION` エラーを返します。

属性一意性検査プラグインの検査対象を次に示します。

- 1 つの属性
- 1 つまたは数個のサブツリー

複数の属性に対して一意性検査を実行する場合は、検査する属性ごとにプラグインのインスタンスを作成する必要があります。

また、属性一意性検査プラグインの動作を、次のように構成することもできます。

- 指定したサブツリー内の各エントリに対して検査する

  たとえば、siroe.com という会社で Company333 および Company999 という 2 つの会社のディレクトリをホスティングしている場合、`uid=jlittle,ou=people,o=Company333,dc=siroe,dc=com` のようなエントリを追加するときは、`o=Company333,dc=siroe,dc=com` サブツリー内だけを対象に一意性を設定する必要があります。このためには、サブツリーの DN を uid 一意性検査プラグインの構成に明示的に記述します。

  この構成オプションについては、467 ページの「接尾辞またはサブツリーの指定」を参照してください。

- 更新済みエントリの DN 内のエントリに属するオブジェクトクラスを指定し、その下にあるすべてのエントリについて一意性検査を実行する

  このオプションは、ホストされる側の環境で便利です。たとえば、`uid=jlittle,ou=people,o=Company333,dc=siroe,dc=com` のようなエントリを追加する場合は、`o=Company333,dc=siroe,dc=com` サブツリーをその構成に明示的にリストせずに *marker* オブジェクトクラスを指定することで、そのサブツリーの下位レベルに対して一意性検査を強制できます。marker オブジェクトクラスに `organization` を指定すると、一意性検査のアルゴリズムによってこのオブジェクトクラス (`o=Company333`) を持つ DN 内のエントリが検索され、その下にあるすべてのエントリに対して一意性検査が実行されます。

  さらに、更新済みエントリに特定のオブジェクトクラスが含まれる場合にだけ、一意性検査を実行するように指定することもできます。たとえば、更新済みエントリに `objectclass=inetorgperson` が含まれている場合にだけ、一意性検査を実行するように指定できます。

  この構成オプションについては、467 ページの「markerObjectClass および requiredObjectClass キーワードの使い方」を参照してください。

レプリケーション環境で属性一意性検査プラグインを使用する場合は、471 ページの「レプリケーションと属性一意性検査プラグイン」を参照してください。

uid 一意性検査プラグインの概要

iPlanet Directory Server5.0 には、属性一意性検査プラグインのインスタンスである uid 一意性検査プラグインを提供します。デフォルトでは、プラグインを実行することにより、そのプラグインはディレクトリを構成した接尾辞内で、uid 属性に指定された値が一意になることを保証します (この接尾辞は、userRoot データベースに対応)。

このプラグインは、別の接尾辞やサブツリーを指定したり、特定のオブジェクトクラスを含むエントリの下位レベルに対してだけ一意性検査を実行したりするように構成を変更できます。uid 一意性検査プラグインの構文および構成は、ほかの属性と同じです。構成の変更については、465 ページの「属性一意性検査プラグインの構成」を参照してください。

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。レプリケーション環境での属性一意性検査プラグインの使用については、471 ページの「レプリケーションと属性一意性検査プラグイン」を参照してください。

属性一意性検査プラグインの構文

属性一意性検査プラグインの構成情報は、cn=plugins,cn=config エントリの下にあるエントリで指定します。nsslapd-pluginarg 属性の構文には、2 つの種類があります。2 種類の構文の違いは、以下の属性一意性検査プラグインの構文中に太字で示されています。

次の例では、接尾辞またはサブツリー以下だけを対象に、一意性検査を実行しています。

```
dn: cn=descriptive_plugin_name,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: descriptive_plugin_namee
nsslapd-pluginPath: /usr/iplanet/ds5/lib/uid-plugin.so
nsslapd-pluginInitfunc: NSUniqueAttr_Init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled: state
nsslapd-pluginarg0: attribute_name
nsslapd-pluginarg1: dn1
[ nsslapd-pluginarg2: dn2 ]
nsslapd-plugin-depends-on-type: database
```

```
nsslapd-pluginId: NSUniqueAttr
nsslapd-pluginVersion: 5.0
nsslapd-pluginVendor:Sun | Netscape Alliance
nsslapd-pluginDescription: Enforce unique attribute values
```

注:

- cn 属性には、任意のプラグイン名を指定できる。プラグインの内容を説明するような名前を使用する。この属性には、一意性を検査する属性の名前は含まれない
- 一意性検査が実行される属性は、1 つしか指定できない
- nsslapd-pluginarg 属性の末尾の数字をそれぞれの回で 1 ずつ増分することによって、一意性検査を実行する接尾辞またはサブツリーの複数の DN を指定できる

属性一意性検査プラグインの構文で使用する変数については、表 17-1 で説明しています。

次の例では、特定のオブジェクトクラスを持つエントリより下のものだけを対象に、一意性検査を実行するように指定しています。

```
dn: cn=descriptive_plugin_name,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: descriptive_plugin_namee
nsslapd-pluginPath: /usr/iplanet/ds5/lib/uid-plugin.so
nsslapd-pluginInitfunc: NSUniqueAttr_Init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled: state
nsslapd-pluginarg0: attribute=attribute_name
nsslapd-pluginarg1: markerObjectClass=objectclass1
[ nsslapd-pluginarg2: requiredObjectClass=objectclass2 ]
nsslapd-plugin-depends-on-type: database
nsslapd-pluginId: NSUniqueAttr
nsslapd-pluginVersion: 5.0
nsslapd-pluginVendor:Sun | Netscape Alliance
nsslapd-pluginDescription: Enforce unique attribute values
```

注:

- cn 属性には、任意のプラグイン名を指定できる。プラグインの内容を説明するような名前を使用する。この属性には、一意性を検査する属性の名前は含まれない
- 一意性検査が実行される属性は、1 つしか指定できない
- nsslapd-pluginarg0 属性が attribute= *attribute_name* で始まる場合は、nsslapd-pluginarg1 属性に markerObjectClass が含まれると想定される

属性一意性検査プラグインの構文で使用する変数については、表 17-1 で説明しています。

表 17-1 属性一意性検査プラグインの変数

変数	定義
descriptive_plugin_name	属性一意性検査プラグインのインスタンスの名前を指定する。一意性を保持する属性名を含める必要はないが、含めることを推奨する。 例：`cn=mail uniqueness,cn=plugins,cn=config`
extension	プラグインのファイル拡張子。拡張子は常に `.so` になる
state	プラグインが有効か無効かを指定する。指定できる値は **`on`** または **`off`**。詳細は、466 ページの「プラグインのオンとオフの切り替え」を参照
attribute_name	一意の値を保持する対象の属性名。属性名は 1 つしか指定できない
dn	属性一意性を保持する接尾辞またはサブツリーの DN。`nsslapd-pluginarg` 属性の末尾の数字を接尾辞またはサブツリーごとに 1 ずつ増分することによって、複数の接尾辞またはサブツリーを指定できる
`attribute=`*attribute_name*	一意の値を保持する対象の属性名。属性名は 1 つしか指定できない
`markerObjectClass=`*objectclass1*	`markerObjectClass` キーワードで指定したオブジェクトクラスを持つ更新エントリの DN に属するエントリの下位レベルに対して、属性一意性検査が実行される 等号 (=) の前後には、空白を挿入しないようにする
`requiredObjectClass=`*objectclass2*	省略可能。DN の代わりに `markerObjectClass` キーワードを使用して一意性検査の範囲を指定する場合は、`requiredObjectClass` キーワードで指定したオブジェクトクラスを持つ更新エントリだけを検査するように指定できる 等号 (=) の前後には、空白を挿入しないようにする

属性一意性検査プラグインのインスタンスの作成

ディレクトリ中の特定の属性が常に一意の値を持つようにするには、検査する属性に対して属性一意性検査プラグインのインスタンスを作成する必要があります。たとえば、ディレクトリ内の mail 属性を含む各エントリが一意の属性値を持つようにする場合は、mail 一意性検査プラグインを作成する必要があります。

属性一意性検査プラグインのインスタンスを作成するには、dse.ldif ファイルを修正して cn=plugins,cn=config エントリの下に新しいプラグインに対するエントリを追加します。新しいエントリは、461 ページの「属性一意性検査プラグインの構文」に記載されている構文の形式に準拠している必要があります。

たとえば、mail 属性に対する属性一意性検査プラグインをインスタンス化するには、次の手順を実行します。

1. dse.ldif ファイルで、uid 一意性検査プラグインを表すエントリ cn=uid uniqueness,cn=plugins,cn=config を検索します。
2. 次のように、uid 一意性検査プラグインエントリの前または後ろに、mail 一意性検査プラグインのエントリを表す行を追加します。

```
dn: cn=mail uniqueness,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: mail uniqueness
nsslapd-pluginPath: /usr/iplanet/ds5/lib/uid-plugin.so
nsslapd-pluginInitfunc: NSUniqueAttr_Init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled:on
nsslapd-pluginarg0:mail
nsslapd-pluginarg1:dc=siroe,dc=com
nsslapd-plugin-depends-on-type: database
nsslapd-pluginId: NSUniqueAttr
nsslapd-pluginVersion: 5.0
nsslapd-pluginVendor:Sun | Netscape Alliance
nsslapd-pluginDescription: Enforce unique attribute values
```

3. Directory Server を再起動します。

この例では、dc=siroe,dc=com エントリ内の mail 属性を持つ各エントリを対象に、一意性検査が実行されます。

属性一意性検査プラグインの構成

この節では、ディレクトリに対して構成されたプラグインを表示するための Directory Server Console の使い方、および属性一意性検査プラグインの構成の修正方法を説明します。

プラグイン構成情報の表示

Directory Server Console を使用して属性一意性検査プラグインの構成エントリを表示するには、次の手順を実行します。

1. Directory Server Console で、「ディレクトリ」タブをクリックします。
2. 左側のナビゲーションツリーで、config フォルダを展開してから、plugins フォルダを展開します。

   プラグインの一覧が右側のナビゲーションウィンドウに表示されます。このウィンドウには、uid 一意性検査プラグインと、464 ページの「属性一意性検査プラグインのインスタンスの作成」の説明に従って作成したその他の属性一意性検査プラグインが表示されます。
3. 右側のナビゲーションウィンドウで、表示するプラグインエントリをダブルクリックします。

   属性エディタが表示されます。このエディタには、プラグインに関する属性および値がすべて一覧表示されます。

Directory Server Console を使用した属性一意性検査プラグインの構成

Directory Server Console を使用してプラグイン構成を更新するには、次の方法を使用します。

- 属性エディタ

  465 ページの「プラグイン構成情報の表示」の説明に従って属性エディタを表示し、属性値フィールドを編集します。
- 「構成」タブ

Directory Server Console の「構成」タブで属性一意性検査プラグインの構成を変更するには、次の手順を実行します。

1. Directory Server Console で「構成」タブを選択し、ナビゲーションツリーの Plugins フォルダを展開して、修正する属性一意性検査プラグインを選択します。

   プラグインの構成パラメタが右側の区画に表示されます。

2. プラグインのオンとオフを切り替えるには、「プラグインを有効にする」チェックボックスを選択または選択解除します。

3. 接尾辞またはサブツリーを追加するには、「追加」をクリックし、空白のテキストフィールドに DN を入力します。

   DN を追加しない場合は、`markerObjectClass` キーワードを使用できます。この構文を使用する場合は、もう一度「追加」をクリックし、461 ページの「属性一意性検査プラグインの構文」の説明に従って requiredObjectClass を指定します。

注	属性名はリストに追加できません。その他の属性の一意性を検査する場合は、検査する属性について属性一意性検査プラグインのインスタンスを新しく作成する必要があります。詳細は、464 ページの「属性一意性検査プラグインのインスタンスの作成」を参照してください。

4. リストから項目を削除するには、削除するテキストフィールドにカーソルを置き、「削除」をクリックします。

5. 「保存」をクリックして、変更内容を保存します。

コマンド行からの属性一意性検査プラグインの設定

ここでは、コマンド行からの属性一意性検査プラグインの構成に関する次の作業について説明します。

- プラグインのオンとオフの切り替え
- 接尾辞またはサブツリーの指定
- markerObjectClass および requiredObjectClass キーワードの使い方

プラグインのオンとオフの切り替え

コマンド行からプラグインを有効にするには、次の LDIF 更新文を含む LDIF ファイルを作成する必要があります。

```
dn: cn=descriptive_plugin_name,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginenabled
nsslapd-pluginenabled: on
```

ldapmodify コマンドを使用して、LDIF ファイルをディレクトリにインポートします。

プラグインを無効にするには、LDIF 更新文の nsslapd-pluginenabled:on 文を、nsslapd-pluginenabled:off 文に置き換えるように変更します。

プラグインを有効または無効に指定した場合は、必ずサーバを再起動してください。サーバの再起動については、35 ページの「iPlanet Directory Server の起動と停止」を参照してください。

接尾辞またはサブツリーの指定

プラグインを定義するエントリ内の nsslapd-pluginarg 属性を使用して、属性の一意性を保持する対象の接尾辞またはサブツリーを指定します。

次の例と同じような更新文を含む LDIF ファイルを作成することによって、1 つ以上のサブツリーを指定できます。

```
dn: cn=mail uniqueness,cn=plugins,cn=config
changetype: add
nsslapd-pluginarg2: dc=iplanet,dc=sun,dc=com
nsslapd-pluginarg3: dc=iplanet,dc=netscape,dc=com
```

この例では、LDIF ファイルは dc=siroe,dc=com、dc=iplanet,dc=sun,dc=com、および dc=iplanet, dc=netscape.com の各サブツリーの下位レベルにある mail 属性の一意性を検査します。

ldapmodify コマンドを使用して、LDIF ファイルをディレクトリにインポートします。

このような構成の変更を行なった場合は、必ずサーバを再起動してください。サーバの再起動については、35 ページの「iPlanet Directory Server の起動と停止」を参照してください。

markerObjectClass および requiredObjectClass キーワードの使い方

属性一意性検査プラグインの構成で、接尾辞またはサブツリーを指定する代わりに、更新エントリの DN に属するエントリの下位レベルに対して、検査を実行するように指定できます。更新エントリとは、markerObjectClass キーワードで指定したオブジェクトクラスを持つものを指します。

組織単位 (ou) オブジェクトクラスを持つ更新済みエントリの DN 内にあるエントリの下位レベルに対して一意性検査を実行するように指定するには、次の例に示すような LDIF ファイルを作成します。

```
dn: cn=mail uniqueness,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: mail uniqueness
nsslapd-pluginPath: /usr/iplanet/ds5/lib/uid-plugin.so
nsslapd-pluginInitfunc: NSUniqueAttr_Init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled: on
nsslapd-pluginarg0: attribute=mail
nsslapd-pluginarg1: markerObjectClass=ou
nsslapd-plugin-depends-on-type:database
nsslapd-pluginId: NSUniqueAttr
nsslapd-pluginVersion: 5.0
nsslapd-pluginVendor: Sun | Netscape Alliance
nsslapd-pluginDescription: Enforce unique attribute values
```

組織単位のエントリの下にあるすべてのエントリを検査する必要がない場合は、更新エントリに特定のオブジェクトクラスが含まれている場合にだけ検査するように指定することによって、範囲を限定できます。

たとえば、mail 属性の一意性を検査する場合に、person または inetorgperson オブジェクトクラスを持つエントリを追加または変更する場合にだけ検査を実行する必要があるとします。

次の例に示すように、requiredObjectClass キーワードを使用して、検査の範囲を限定できます。

```
dn: cn=mail uniqueness,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: mail uniqueness
nsslapd-pluginPath: /usr/iplanet/ds5/lib/uid-plugin.so
nsslapd-pluginInitfunc: NSUniqueAttr_Init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled: on
nsslapd-pluginarg0: attribute=mail
nsslapd-pluginarg1: markerObjectClass=ou
nsslapd-pluginarg2: requiredObjectClass=person
nsslapd-plugin-depends-on-type: database
nsslapd-pluginId: NSUniqueAttr
nsslapd-pluginVersion: 5.0
nsslapd-pluginVendor:Sun | Netscape Alliance
nsslapd-pluginDescription: Enforce unique attribute values
```

nsslapd-pluginarg 属性の末尾の数字を増分することによって、markerObjectClass または requiredObjectClass の各キーワードを繰り返し使用することはできません。

注	nsslapd-pluginarg0 属性には、常に一意性を保持する属性名を指定します。

属性一意性検査プラグインの構文例

ここでは、dse.ldif ファイルで指定する属性一意性検査プラグインの構文例を示します。例に示されている構文はすべて、UNIX マシンで現れるようなプラグイン構文です。

- 1 つの属性および 1 つのサブツリーの指定
- 1 つの属性および複数のサブツリーの指定

1 つの属性および 1 つのサブツリーの指定

次の例では、dc=siroe,dc=com サブツリーの下にある mail 属性の一意性を保持するようにプラグインを構成しています。

```
dn: cn=mail uniqueness,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: mail uniqueness
nsslapd-pluginPath: /usr/iplanet/ds5/lib/uid-plugin.so
nsslapd-pluginInitfunc: NSUniqueAttr_Init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled: on
nsslapd-pluginarg0: mail
nsslapd-pluginarg1: dc=siroe,dc=com
nsslapd-plugin-depends-on-type: database
nsslapd-pluginId: NSUniqueAttr
nsslapd-pluginVersion: 5.0
nsslapd-pluginVendor:Sun | Netscape Alliance
nsslapd-pluginDescription: Enforce unique attribute values
```

1 つの属性および複数のサブツリーの指定

次の例では、l=Chicago,dc=siroe,dc=com および l=Boston,dc=siroe,dc=com の各サブツリーの下にある mail 属性の一意性を保持するようにプラグインを構成しています。

```
dn: cn=mail uniqueness,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: mail uniqueness
nsslapd-pluginPath: /usr/iplanet/ds5/lib/uid-plugin.so
nsslapd-pluginInitfunc: NSUniqueAttr_Init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled: on
nsslapd-pluginarg0: mail
nsslapd-pluginarg1: l=Chicago,dc=siroe,dc=com
nsslapd-pluginarg2: l=Boston,dc=siroe,dc=com
nsslapd-plugin-depends-on-type: database
nsslapd-pluginId: NSUniqueAttr
nsslapd-pluginVersion: 5.0
nsslapd-pluginVendor:Sun | Netscape Alliance
nsslapd-pluginDescription: Enforce unique attribute values
```

注 nsslapd-pluginarg0 属性には、常に一意性を保持する属性名を指定します。その他のすべての nsslapd-pluginarg 属性 (nsslapd-pluginarg1 から nsslapd-pluginarg*x*) には、DN を指定します。

この構成では、mail 属性の同じ値のインスタンスが l=Chicago,dc=siroe,dc=com サブツリーの下と l=Boston,dc=siroe,dc=com サブツリーの下に存在することがプラグインにより許されます。たとえば、次のような場合が考えられます。

```
mail=bjensen,l=Chicago,dc=siroe,dc=com
```

```
mail=bjensen,l=Boston,dc=siroe,dc=com
```

両方のサブツリーの下に同じ値のインスタンスが 1 つだけ存在するようにする場合は、dc=siroe,dc=com サブツリー全体に対して一意性を保持するようプラグインを設定する必要があります。

レプリケーションと属性一意性検査プラグイン

レプリケーションアグリーメントに関係している Directory Server 上で属性一意性検査プラグインを使用する場合は、各サーバでのプラグインの構成方法について慎重に検討する必要があります。

次のケースを想定します。

- 1 つのサプライヤと 1 つから数台のコンシューマによる単純なレプリケーション
- マルチマスターによる複雑なレプリケーション

属性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。

単純な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サプライヤサーバ上で行われるので、属性一意性検査プラグインをサプライヤサーバ上で有効にする必要があります。コンシューマサーバでは、属性一意性検査プラグインを有効にする必要はありません。

属性一意性検査プラグインをコンシューマサーバ上で有効にしても、Directory Server は正常に処理を行いますが、性能が低下する場合があります。

マルチマスターレプリケーションモデル

マルチマスターレプリケーションモデルでは、2 つのマスターが同一のレプリカに対して、サプライヤとコンシューマの両方の役割を果たします。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、1 つのサーバ上で属性一意性検査プラグインを有効にしても、2 つのマスター上のすべての属性値が、常に一意であることを確保するには不十分です。逆に、1 つのサーバだけで属性一意性検査プラグインを有効にすると、各レプリカで保持されるデータの整合性がとれなくなる可能性もあります。

ただし、次の条件をすべて満たしている場合は、属性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命名属性である
- 属性一意性検査プラグインが 2 つのマスター上で有効になっている

これらの条件を満たしている場合は、属性の一意性に関する競合は、レプリケーション時の命名競合として報告されます。ただし、レプリケーション時の命名競合は、手動で解決する必要があります。レプリケーションの競合を解決する方法については、328 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。

LDIF (LDAP Data Interchange Format)

Directory Server では、ディレクトリとディレクトリエントリをテキスト形式で記述する場合に、LDIF (LDAP Data Interchange Format) が使用されます。LDIF は、初期ディレクトリデータベースを構築したり、一度に大量のエントリをディレクトリに追加したりするために使用されます。また、ディレクトリエントリに対する変更を記述するためにも使用されます。このため、Directory Server のコマンド行ユーティリティの大部分では、入出力を LDIF に頼っています。

LDIF ファイルはテキスト形式なので、どのような言語でも LDIF ファイルを作成することができます。ディレクトリデータはすべて、Unicode の UTF-8 エンコードを使用して格納されます。したがって、作成する LDIF ファイルも UTF-8 でエンコードされている必要があります。

この章では、LDIF について、次の項目ごとに説明します。

LDIF を使用したディレクトリエントリの修正については、第 2 章「ディレクトリエントリの作成」を参照してください。

LDIF ファイル形式

LDIF は、1 つ以上のディレクトリエントリから構成され、エントリ間は空白行で区切られています。LDIF エントリはそれぞれ、エントリ ID (省略可能)、識別名 (必須)、1 つ以上のオブジェクトクラス、および複数の属性定義から構成されます。

LDIF 形式は RFC 2849 *The LDAP Data Interchange Format (LDIF)* で定義されています。iPlanet Directory Server は、この標準に準拠しています。

LDIF で表されるディレクトリエントリの基本形式は次のとおりです。

```
dn: distinguished_name
objectClass: object_class
objectClass: object_class
...
attribute_type[;subtype]:attribute_value
attribute_type[;subtype]:attribute_value
...
```

DN と、少なくとも 1 つのオブジェクトクラス定義を指定する必要があります。さらに、エントリに対して定義するオブジェクトクラスで必要とされる属性もすべて指定します。その他の属性やオブジェクトは省略可能です。オブジェクトクラスや属性は任意の順序で指定できます。また、コロンのあとの空白も省略可能です。標準オブジェクトクラスと属性については、『iPlanet Directory Server スキーマリファレンス』を参照してください。

表 A-1 に、前述の定義で示されている LDIF フィールドについてまとめます。

表 A-1 LDIF フィールド

フィールド	定義
[*id*]	省略可能。エントリ ID を表す正の 10 進数値。この ID は、データベース作成ツールによって自動的に生成される。この値を追加したり、編集したりしてはならない
`dn:` *distinguished_name*	エントリの識別名を指定する。識別名に関する詳細は、『iPlanet Directory Server 導入ガイド』を参照
`objectClass:` *object_class*	このエントリで使用されるオブジェクトクラスを指定する。オブジェクトクラスは、エントリで使用可能かつ必要な属性のタイプ (スキーマ) を識別する。標準オブジェクトクラスのリストについては『iPlanet Directory Server スキーマリファレンス』、スキーマのカスタマイズについては第 9 章「ディレクトリスキーマの拡張」を参照
attribute_type	このエントリで使用される属性の種類を指定する。属性はスキーマで定義する必要がある。標準属性のリストについては『iPlanet Directory Server スキーマリファレンス』、スキーマのカスタマイズについては第 9 章「ディレクトリスキーマの拡張」を参照

表 A-1　LDIF フィールド (続き)

フィールド	定義
[*subtype*]	省略可能。言語、バイナリ、発音などのサブタイプを指定する。このタグを使用して、対応する属性値を表現している言語や、属性値がバイナリであるか発音であるかを識別する。属性のサブタイプについては、49 ページの「属性のサブタイプの追加」を参照。サポートされているサブタイプタグのリストについては、515 ページの表 D-2 を参照
attribute_value	属性タイプとともに使用される属性値を指定する

ディレクトリのエントリに対する変更を表す LDIF 構文は、前述の構文とは異なります。LDIF を使用したディレクトリエントリの修正については、第 2 章「ディレクトリエントリの作成」を参照してください。

LDIF での断続行

LDIF を指定するときに、行を改行または継続させたり、折り返したりするには、行の継続する部分を 1 スペース分だけインデントします。たとえば、次の 2 つの文は同じものです。

```
dn: cn=Jake Lupinski,dc=siroe,dc=com
```

```
dn: cn=Jake Lup
 inski, dc=sir
 oe,dc=comcom
```

LDIF 行を改行したり、継続させたりすることは必須ではありません。ただし、これによって、LDIF ファイルが読みやすくなります。

バイナリデータの表記

JPEG 画像のようなバイナリデータを LDIF で表すには、Base 64 エンコードを使用します。

Base 64 符号化の使用

Base64 で符号化されたデータは、`::` 記号を使用して識別します。たとえば、次のようにします。

`jpegPhoto::` *encoded_data*

バイナリデータ以外に、Base 64 エンコードで処理する必要のある値は、次のとおりです。

- セミコロン (;) や空白で始まる値
- 新しい行を含む、ASCII 以外のデータが入った値

`ldif` コマンド行ユーティリティに `-b` パラメタを指定して、バイナリデータを LDIF 形式に変換します。

```
# /usr/sbin/directoryserver ldif -b attributeName
```

ここで、*attributeName* は バイナリデータを与える属性名です。バイナリデータは標準入力から読み込まれ、その結果は標準出力に書き込まれます。したがって、リダイレクト演算子を使用して、入力ファイルと出力ファイルを選択する必要があります。

コマンドは任意の入力を受け取り、断続行処理を行い、適切な属性情報を追加して LDIF 形式に変換します。コマンドを使用して、入力で Base 64 エンコードが必要であるかどうかを確認することもできます。たとえば、次のようにします。

```
/usr/sbin/directoryserver ldif -b jpegPhoto < mark.jpg > out.ldif
```

この例では、JPEG 形式を含むバイナリファイルが取得され、このファイルが `jpegPhoto` という属性のために LDIF 形式に変換されます。出力は `out.ldif` に保存されます。

オプション `-b` は、`ldif` ユーティリティによって、入力全体が 1 つのバイナリ値として解釈されることを表します。`-b` を指定しない場合は、各行が異なる入力値とみなされます。

次に出力ファイルを編集して、バイナリ値を含むディレクトリエントリの作成または修正に必要な LDIF 文を追加できます。たとえば、テキストエディタで `out.ldif` ファイルを開き、ファイルの最初に次の行 (太字で表示) を追加します。

```
dn:cn=Barney Fife,ou=People,dc=siroe,dc=com
changetype: modify
add: jpegPhoto
jpegPhoto:: encoded_data
```

この例では、*encoded_data* は、`ldif` コマンドによって作成された `out.ldif` ファイルの内容を表しています。

LDIF を使用したディレクトリエントリの指定

ディレクトリには、さまざまなタイプのエントリを格納できます。ここでは、ディレクトリで使用される最も一般的な 3 つのタイプのエントリである組織エントリ、組織単位エントリ、および組織メンバーエントリについて、重点的に説明します。

エントリに対して定義されるオブジェクトクラスは、エントリが組織、組織単位、組織メンバー、またはその他のタイプのエントリのいずれを表しているかを示しています。ディレクトリで作成可能なエントリのタイプについては、『iPlanet Directory Server 導入ガイド』を参照してください。デフォルトで、ディレクトリで使用できるオブジェクトクラスすべてのリスト、およびもっとも一般的に使用される属性のリストについては、『iPlanet Directory Server スキーマリファレンス』を参照してください。

組織エントリの指定

ディレクトリが、少なくとも 1 つの組織エントリを持つことはよくあります。一般に、ディレクトリの最初のエントリ、または一番上のエントリは組織エントリです。多くの場合、組織エントリはディレクトリに対して設定されている接尾辞に対応します。たとえば、ディレクトリで接尾辞 `dc=siroe,dc=com` が使用されるように定義されていて、ディレクトリ `dc=siroe,dc=com` に組織エントリを持つ予定であるとします。

組織エントリを定義するには、次のように LDIF を指定します。

```
dn: distinguished_name
objectClass: top
objectClass:organization
o: organization_name
list_of_optional_attributes
...
```

LDIF 形式で表した組織エントリの例は次のとおりです。

```
dn:dc=siroe,dc=com
objectclass: top
objectclass:organization
o: siroe.com Corporation
description: Fictional company for example purposes
telephonenumber: 555-5555
```

次の例にある組織名では、コンマが使用されています。

```
dn: o="siroe.com Chile\\, S.A."
objectclass: top
objectclass:organization
o: "siroe.com Chile\\, S.A."
description: Fictional company for example purposes
telephonenumber: 555-5556
```

表 A-2 に、LDIF 形式の組織エントリの各要素についてまとめます。

表 A-2 組織エントリの LDIF 要素

LDIF 要素	内容
`dn:` *distinguished_name*	エントリの 識別名 (distinguished name) を指定する。DN については、『iPlanet Directory Server 導入ガイド』を参照。DN は必須
`objectClass: top`	必須。`top` オブジェクトクラスを指定する
`objectClass: organization`	`organization` オブジェクトクラスを指定する。この行では、エントリが組織として定義される。このオブジェクトクラスで使用できる属性のリストについては、『iPlanet Directory Server スキーマリファレンス』を参照
`o:` *organization_name*	組織名を表す属性。組織名にコンマが含まれている場合は、\ を 1 つ使用してコンマをエスケープし、組織引数全体を引用符で囲む必要がある。ただし、UNIX シェルで作業している場合は、この \ を 2 つ使用してエスケープする必要がある。たとえば、接尾辞を siroe.com Bolivia, S.A. に設定するには、UNIX マシンでは「`"o: siroe.com Bolivia\\, S.A."` 」と入力する
list_of_attributes	エントリで管理する必要のあるオプション属性のリストを指定する。このオブジェクトクラスで使用できる属性のリストについては、『iPlanet Directory Server スキーマリファレンス』を参照

組織単位エントリの指定

組織単位エントリは、ディレクトリツリーに含まれる主要な分岐点、つまりサブディレクトリを表すのによく使用されます。このエントリは、ユーザを含むサブツリーやグループを含むサブツリーなど、企業内の主要で適度に静的なエンティティに対応します。ただし、エントリに含まれる組織単位の属性は、マーケティングやエンジニアリングのような、企業の主要な組織も表すことがあります。

ディレクトリツリーには通常、複数の組織単位、つまり分岐点が存在します。ディレクトリツリーの設計方法については、『iPlanet Directory Server 導入ガイド』を参照してください。

組織単位エントリを定義するには、次のように LDIF を指定します。

```
dn: distinguished_name
objectClass: top
objectClass:organizationalUnit
ou: organizational_unit_name
list_of_optional_attributes
...
```

LDIF 形式で表した組織単位エントリの例は次のとおりです。

```
dn: ou=people, dc=siroe,dc=com
objectclass: top
objectclass:organizationalUnit
ou: people
description: Fictional organizational unit for example purposes
```

表 A-3 に、LDIF 形式の組織単位エントリの各要素についてまとめます。

表 A-3 組織単位エントリの LDIF 要素

LDIF 要素	内容
`dn:` *distinguished_name*	エントリの 識別名 (distinguished name) を指定する。DN は必須。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バックスラッシュ (\) を付けて区別する必要がある。たとえば、次のようにする `dn: ou=people,o=siroe.com Bolivia\,S.A.`
`objectClass: top`	必須。top オブジェクトクラスを指定する
`objectClass: organizationalUnit`	organizationalUnit オブジェクトクラスを指定する。この行では、エントリが organizationalUnit として定義される。このオブジェクトクラスで使用できる属性のリストについては、『iPlanet Directory Server スキーマリファレンス』を参照
`ou:` *organizational_unit_name*	組織単位の名前を識別する属性
list_of_attributes	エントリで管理する必要のあるオプション属性のリストを指定する。このオブジェクトクラスで使用できる属性のリストについては、『iPlanet Directory Server スキーマリファレンス』を参照

組織ユーザのエントリの指定

ディレクトリにあるエントリの大部分は、組織メンバーを表します。

LDIF では、組織メンバーは次のように定義されます。

```
dn: distinguished_name
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: common_name
sn: surname
list_of_optional_attributes
```

LDIF 形式で表した組織ユーザエントリの例は次のとおりです。

```
dn: uid=bjensen,ou=people,dc=siroe,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Babs Jensen
sn: Jensen
givenname: Babs
uid: bjensen
ou: Marketing
ou: people
description: Fictional person for example purposes
telephonenumber: 555-5557
userpassword: {sha}dkfljlk34r2kljdsfk9
```

表 A-4 に、LDIF 形式の組織ユーザエントリの各要素についてまとめます。

表 A-4 組織ユーザエントリの LDIF 要素

LDIF 要素	内容
`dn: distinguished_name`	エントリの 識別名 (distinguished name) を指定する。DN は必須。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バックスラッシュ (\) を付けて区別する必要がある。 例 : `dn:uid=bjensen,ou=people,o=siroe.com` `Bolivia\,S.A.`
`objectClass: top`	必須。`top` オブジェクトクラスを指定する

表 A-4　組織ユーザエントリの LDIF 要素 (続き)

LDIF 要素	内容
`objectClass: person`	person オブジェクトクラスを指定する。このオブジェクトクラスは、個人や組織メンバーに対する検索操作を行うときに、多くの LDAP クライアントで必要とされるので、必ず指定する必要がある
`objectClass: organizationalPerson`	organizationalPerson オブジェクトクラスを指定する。このオブジェクトクラスは、組織メンバーに対する検索操作を行うときに、LDAP クライアントの一部で必要とされるので、必ず指定する必要がある
`objectClass:` *inetOrgPerson*	inetOrgPerson オブジェクトクラスを指定する。inetOrgPerson オブジェクトクラスには、もっとも広い範囲の属性が含まれるので、組織ユーザエントリの作成に適している。このオブジェクトクラスでは、uid 属性が必要とされるので、このオブジェクトクラスが含まれるエントリには uid 属性値に基づく名前が付けられる。このオブジェクトクラスで使用できる属性のリストについては、『iPlanet Directory Server スキーマリファレンス』を参照
`cn:` *common_name*	ある個人が通常使用しているフルネームである、共通名を指定する。例 : `cn: Bill Anderson`。少なくとも 1 つの共通名が必要である
`sn:` *surname*	個人の姓を指定する。例 : `sn: Anderson`。姓は必須
list_of_attributes	エントリで管理する必要のあるオプション属性のリストを指定する。このオブジェクトクラスで使用できる属性のリストについては、『iPlanet Directory Server スキーマリファレンス』を参照

LDIF を使用したディレクトリの定義

LDIF を使用して、ディレクトリ全体のコンテンツを定義することができます。ディレクトリに追加するエントリが大量にある場合、LDIF の使用によって、ディレクトリの作成作業を効率化できます。

LDIF を使用してディレクトリを作成するには、次の手順を実行します。

1. 追加するエントリを含む ASCII ファイルを LDIF 形式で作成します。

   各エントリを分割できるように、各エントリ間に空白行を挿入してください。1 行だけを使用すべきで、ファイルの先頭行は空行であってはなりません。そうでない場合 `ldapmodify` Utility は終了します。詳細は、477 ページの「LDIF を使用したディレクトリエントリの指定」を参照してください。

2. ファイルは、データベースの先頭エントリ、つまりルートエントリで始めます。

   このルートエントリは、データベースに含まれる接尾辞またはサブ接尾辞を表している必要があります。たとえば、データベースに接尾辞 `dc=siroe`、`dc=com` がある場合は、ディレクトリの最初のエントリは次のようにならなければなりません。

   ```
   dn: dc=siroe,dc=com
   ```

   接尾辞については、『iPlanet Directory Server 構成、コマンド、およびファイルのリファレンス』の 「Suffix」パラメタを参照してください。

3. LDIF ファイルで、分岐点を表すエントリがその分岐の下に作成するエントリの前に置かれていることを確認します。

   たとえば、people サブツリーと group サブツリーにエントリを配置する場合は、これらのサブツリー内にエントリを作成する前に、これらのサブツリーの分岐点を作成します。

4. 次のいずれかの方法を使用して、LDIF ファイルからディレクトリを作成します。

   - Directory Server Console

     インポートするデータベースが小さい (1000 エントリ未満) 場合は、この方法を使用します。137 ページの「Console を使用したインポートの実行」を参照

   - `directoryserver ldif2db` コマンド

     インポートするデータベースが大きい (1000 エントリ以上) 場合は、この方法を使用します。140 ページの「ldif2db コマンドを使用したインポート」を参照

   - `-a` パラメタともなった `ldapmodify` コマンド行ユーティリティ

現在、ディレクトリデータベースを使用していて、このデータベースに新しいサブツリーを追加する場合は、この方法を使用します。LDIF ファイルからディレクトリを作成するその他の方法と異なり、`ldapmodify` を使用してサブツリーを追加する前に Directory Server を実行しておく必要があります。53 ページの「ldapmodify を使用したエントリの追加と修正」を参照してください。

LDIF ファイルの例

次に、組織エントリ 1 つ、組織単位エントリ 2 つ、および組織メンバーエントリ 3 つを含む LDIF ファイルの例を示します。

```
dn: o=siroe.com Corp,dc=siroe,dc=com
objectclass: top
objectclass: organization
o: siroe.com Corp
description: Fictional organization for example purposes

dn: ou=People,o=siroe.com Corp,dc=siroe,dc=com
objectclass: top
objectclass: organizationalUnit
ou: People
description: Fictional organizational unit for example purposes
tel: 555-5559

dn: cn=June Rossi,ou=People,o=siroe.com Corp,dc=siroe,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: June Rossi
sn: Rossi
givenName: June
mail: rossi@siroe.com
userPassword: {sha}KDIE3AL9DK
ou: Accounting
ou: people
telephoneNumber: 2616
roomNumber: 220

dn: cn=Marc Chambers,ou=People,o=siroe.com Corp,dc=siroe,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: Marc Chambers
sn: Chambers
givenName: Marc
```

```
mail: chambers@siroe.com
userPassword: {sha}jdl2alem87dlacz1
telephoneNumber: 2652
ou: Manufacturing
ou:People
roomNumber: 167

dn: cn=Robert Wong,ou=People,o=siroe.com Corp,dc=siroe,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: Robert Wong
cn: Bob Wong
sn: Wong
givenName: Robert
givenName: Bob
mail: bwong@siroe.com
userPassword: {sha}nn2msx761
telephoneNumber: 2881
roomNumber: 211
ou: Manufacturing
ou: people

dn: ou=Groups,o=siroe.com Corp,dc=siroe,dc=com
objectclass: top
objectclass: organizationalUnit
ou: groups
description: Fictional organizational unit for example purposes
```

複数言語での情報の保存

ディレクトリに言語が 1 つだけ含まれている場合は、ディレクトリに新しいエントリを追加するために、特別な操作を実行する必要はありません。ただし、多国籍組織の場合は、情報をさまざまな言語で格納し、異なるロケールのユーザが母国語でディレクトリ情報を表示できるようにしておく必要があります。

ディレクトリ内の情報に複数の言語が含まれている場合、言語タグが属性値と関連付けられます。新しいエントリを追加する場合は、相対識別名 (RDN) で使用されている属性値を、言語コードを指定せずに設定する必要があります。

1 つの属性に複数の言語を格納することもできます。1 つの属性に複数の言語を格納すると、属性のタイプは同じになりますが、値には異なる言語コードが指定されます。

Directory Server でサポートされている言語と、関連付けられている言語タグのリストについては、513 ページの「サポートされているロケールの特定」を参照してください。

注 ディレクトリ内への文字列の格納方法は言語タグには影響しません。オブジェクトクラスと属性文字列は、すべて UTF-8 を使用して格納されます。

たとえば、siroe.com Corporation はアメリカとフランスにオフィスがあるので、従業員が母国語でディレクトリ情報を参照できるようにする必要があるとします。ディレクトリエントリを追加するときに、ディレクトリ管理者は英語とフランス語の両方の属性値の設定を選択します。新入社員 Babs Jensen のためにディレクトリエントリを追加する場合は、管理者は次のような LDIF エントリを作成します。

```
dn: uid=bjensen,ou=people,dc=siroe,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
name: Babs Jensen
cn: Babs Jensen
sn: Jensen
uid: bjensen
streetAddress: 1 University Street
streetAddress;lang-en: 1 University Street
streetAddress;lang-fr: 1\, rue de l'Université
preferredLanguage: fr
```

このディレクトリエントリに対して、優先言語が英語に設定された LDAP クライアントからアクセスした場合、住所は `1 University Street` と表示されます。ただし、LDAP クライアントの優先言語がフランス語に設定されている場合には、住所は `1 rue de l'Université` と表示されます。

ディレクトリエントリの検索

ディレクトリ内のエントリは、すべての LDAP クライアントから検索できます。クライアントは、通常、何らかの形式の検索用インタフェースを備えているので、簡単にディレクトリを検索してエントリ情報を取得することができます。

注 ユーザまたは管理者がディレクトリで設定したアクセス制御によって、検索結果が決まります。一般的なユーザは、通常、ディレクトリをあまり参照することはありませんが、ディレクトリの管理者は、構成を含むすべてのデータへのフルアクセス権限を持っています。

ディレクトリへのアクセス制御の設定については、187 ページの「アクセス制御の管理」を参照してください。

この章では、次の項目について説明します。

- Server Console を使用したエントリの検索
- LDAP 検索フィルタ
- ldapsearch の使用
- 国際化ディレクトリの検索

Server Console を使用したエントリの検索

Directory Server Console の「ディレクトリ」タブを使用してディレクトリツリーの内容を参照し、ディレクトリ内の特定のエントリを検索します。

1. Directory Server が起動していることを確認します。
2. Directory Server Console を起動します。

   起動方法については、26 ページの「iPlanet Directory Server Console の起動」を参照してください。

3. Directory Server Console で、「ディレクトリ」タブを選択します。

   ディレクトリへの認証に使用した DN に応じて、このタブには表示アクセスが許可されたディレクトリの内容が表示されます。ツリーの内容を順に参照するか、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「検索」を選択します。

4. 「検索」ダイアログボックスでは、ディレクトリ内の名前を検索する単純なインタフェースが提供されます。このダイアログボックスでは、ダイアログが起動するときに選択されていたディレクトリのノードから検索が実行されます。広範囲の検索ではディレクトリの最上位から検索が実行され、短時間で行う検索では下位のサブツリーから検索が実行されます。

   「高度な検索」では、特定の属性および値に絞った検索が可能です。独自の LDAP 文字列を使って検索する場合は、検索フィルタも使用できます。

   この機能の使い方については、オンラインヘルプを参照してください。

注　「検索」ダイアログボックスを使用した検索は、レフェラルに従いません。

5. 検索が返したエントリを表示または編集する場合は、「OK」をクリックして「検索」ダイアログボックスを閉じてください。検索結果は別のウィンドウに表示され、表示された名前をダブルクリックすると、フルエントリが表示されます。エントリは「編集」ダイアログボックスに表示され、アクセス制御により許可されている場合はエントリを修正できます。

   許可されていない場合は、「キャンセル」をクリックし、検索結果を表示せずに「検索」ダイアログボックスを閉じます。

ldapsearch の使用

ldapsearch コマンド行ユーティリティを使用すると、ディレクトリエントリの場所を指定し、検索することができます。このユーティリティは、指定された識別名とパスワードを使用して、指定されたサーバへの接続を確立し、指定された検索フィルタに基づいてエントリを検索します。検索範囲には、単一のエントリ、エントリの直下のサブエントリ、ツリー全体、またはサブツリー全体を含めることができます。

ここでは、次の事項について説明します。

- 特殊文字の使い方
- ldapsearch コマンド行の形式
- よく使用される ldapsearch オプション
- ldapsearch の例

特殊文字の使い方

ldapsearch コマンド行ユーティリティの使用時に、コマンド行インタプリタにとって特別な意味を持つ文字 (スペース [ ]、アスタリスク [*]、バックスラッシュ [\] など) を含む値の指定が必要となることがあります。特殊文字を使用する場合は、その値を引用符 ("") で囲みます。たとえば、次のようにします。

```
-D "cn=Barbara Jensen,ou=Product Development,dc=siroe,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。

ldapsearch コマンド行の形式

ldapsearch を使用するときには、次の形式でコマンドを入力します。

ldapsearch [*optional_options*] [*optional_search_filter*] [*optional_list_of_attributes*]

各要素の意味は次のとおりです。

- *optional_options* には一連のコマンド行オプションを指定する。検索フィルタを指定する場合は、必ず検索フィルタの前にオプションを指定する
- *optional_search_filter* には LDAP 検索フィルタを指定する。このフィルタについては、493 ページの「LDAP 検索フィルタ」を参照。-f オプションを使用して検索フィルタを指定する場合は、コマンド行で検索フィルタを指定してはならない

- *optional_list_of_attributes* には属性のリストを指定する。個々の属性は空白で区切る。属性のリストを指定すると、検索結果として返される属性の数が少なくなる。この属性のリストは、必ず検索フィルタの後に指定すること (493 ページの「属性サブセットの表示」の例を参照)。属性のリストを指定しないで検索を実行すると、検索の結果、ディレクトリに設定されたアクセス制御セットで許可されている、すべての属性 (操作属性は除く) に対して値が返される

注	検索操作の結果として操作属性が返されるようにするには、検索コマンドで操作属性を明示的に指定する必要があります。明示的に指定した操作属性のほかに、通常の属性も含めて検索を実行するには、`ldapsearch` コマンドの属性リストにアスタリスク (*) を指定します。

よく使用される ldapsearch オプション

次の表に、よく使用される ldapsearch コマンド行オプションの一覧を示します。スペース [ ] を含む値を指定する場合は、`-b "ou=groups, dc=siroe,dc=com"` のように、値を二重引用符で囲んでください。

`-b`

検索の開始点を指定する。ここで指定する値は、現在データベース内に存在する識別名でなければならない。このオプションは、`LDAP_BASEDN` 環境変数が Base DN に設定されている場合には省略可

このオプションで指定する値は、次のように、二重引用符で囲むこと。たとえば、次のようにする

```
-b "cn=Barbara Jensen, ou=Product Development,
dc=siroe,dc=com"
```

ルート DSE エントリを検索する場合は、空の文字列を指定する。たとえば、次のようにする

```
-b ""
```

`-D`

サーバに対する認証に使用する識別名を指定する。匿名アクセスがサーバによってサポートされている場合、このオプションは省略可能。ここで指定する値は、Directory Server によって認識され、エントリを検索できる権限を持つ DN でなければならない。たとえば、次のようにする

```
-D "uid=bjensen, dc=siroe,dc=com"
```

`-h`

Directory Server がインストールされているマシンのホスト名または IP アドレスを指定する。ホストを指定しなかった場合は、localhost が `ldapsearch` によって使用される (たとえば、`-h mozilla`)

`-l` 検索要求に割り当てる最大時間を、秒単位で指定する。ここでどのような値を指定しても、`ldapsearch` が、サーバの `nsslapd-timelimit` 属性によって許可されている時間より長く待機することはない (たとえば、`-l 300`)。`nsslapd-timelimit` 属性のデフォルト値は 3,600 秒

`-p` Directory Server で使用する TCP ポート番号を指定する (たとえば、`-p 1049`)。デフォルトは、389。`-Z` が使用される場合は、デフォルトは 636

`-s` 検索の範囲を指定する。範囲には次のいずれかを指定できる

- `base` : `-b` オプションで指定したエントリ、または `LDAP_BASEDN` 環境変数で定義されているエントリだけが検索される
- `one` : `-b` オプションで指定したエントリのすぐ下の子エントリだけが検索される。検索されるのは子エントリだけで、`-b` オプションで指定した実際のエントリは検索されない
- `sub` : `-b` オプションで指定したエントリとその子孫のエントリすべてが検索される。つまり、`-b` オプションで指定した開始点からのサブツリー全体に対して、検索が実行される。この値がデフォルト

`-w` `-D` オプションで指定した識別名に関連付けられたパスワードを指定する。このオプションを指定しなかった場合は、匿名アクセスが使用される (たとえば、`-w diner892`)

`-z` 検索要求に対して返されるエントリの最大数を指定する (たとえば、`-z 1000`)

通常、ここでどのような値を指定しても、`ldapsearch` が、サーバの `nsslapd-sizelimit` 属性によって許可されている数を超えるエントリを返すことはない。ただし、このコマンド行引数を指定するときに root DN としてバインドすれば、この制限を上書きすることができる。root DN としてバインドすると、このオプションがデフォルトで 0 に設定される。`nsslapd-sizelimit` 属性のデフォルト値は 2,000 エントリ

ldapsearch の例

次に示す例は、以下の内容を前提にしています。

- ユーザはディレクトリ内のすべてのエントリを検索する
- ディレクトリは、検索および読み取りのための匿名アクセスを許可するように構成されている。この場合、検索を実行するためにバインド情報を指定する必要はない。匿名アクセスについては、205 ページの「ユーザアクセスの定義 : userdn キーワード」を参照

- サーバが **mozilla** という名前のホストに置かれている
- サーバは、ポート番号 **389** を使用する。これはデフォルトのポートなので、検索要求時にポート番号を指定する必要はない
- すべてのデータは、接尾辞 **dc=siroe,dc=com** に格納される

すべてのエントリを返す場合

前述の条件に従って次のコールを指定すると、ディレクトリ内のすべてのエントリが返されます。

```
ldapsearch -h mozilla -b "dc=siroe,dc=com" -s sub "objectclass=*"
```

"objectclass=*" は、ディレクトリ内のすべてのエントリにマッチする検索フィルタです。

コマンド行での検索フィルタの指定

検索フィルタは、直接コマンド行で指定できます。このように指定した場合は、フィルタを必ず二重引用符で囲んでください ("filter")。また、-f オプションは指定しないでください。たとえば、次のようにします。

```
ldapsearch -h mozilla -b "dc=siroe,dc=com" "cn=babs jensen"
```

ルート DSE エントリの検索

ルート DSE は、ローカルの Directory Server がサポートするすべての接尾辞のリストを含む特殊なエントリです。このエントリは、"" という検索ベースを使用して検索できます。また、検索対象として base を、検索フィルタとして "objectclass=*" を指定する必要があります。たとえば、次のようにします。

```
ldapsearch -h mozilla -b "" -s base "objectclass=*"
```

スキーマエントリの検索

iPlanet Directory Server では、すべての Directory Server スキーマが cn=schema という特別なエントリに格納されています。このエントリには、すべてのオブジェクトクラスおよび Directory Server に対して定義された属性に関する情報が含まれています。

このエントリの内容は、次のコマンドで確認できます。

```
ldapsearch -h mozilla -b "cn=schema" -s base "objectclass=*"
```

属性サブセットの表示

ldapsearch コマンドでは、検索結果はすべて LDIF 形式で返されます。デフォルトでは、ldapsearch によって、エントリの識別名と、読み取りが許可されているすべての属性が返されます。ディレクトリアクセス制御は、任意のディレクトリエントリの属性サブセットに対し、読み取り専用でアクセスできるように設定できます。ただし、操作属性は返されません。検索操作の結果として操作属性が返されるようにするには、検索コマンドで操作属性を明示的に指定する必要があります。

検索結果として返される属性のうち、一部の属性だけを表示するとします。このような場合は、コマンド行で検索フィルタの直後に必要な属性を指定することによって、返される属性を特定のものだけに制限できます。たとえば、ディレクトリ内のすべてのエントリの cn および sn 属性が必要な場合は、次のコマンド行コールを使用します。

```
ldapsearch -h mozilla -b "dc=siroe, dc=com" "objectclass=*" sn cn
```

検索フィルタでのコンマを含む DN の指定

検索フィルタ内の DN の値がコンマを含む場合は、そのコンマをバックスラッシュ (\) でエスケープする必要があります。たとえば、siroe.com Bolivia, S.A. サブツリーに属する全員を検索するには、次のようなコマンドを使用します。

```
ldapsearch -h mozilla -s base -b "o=siroe.com Bolivia\, S.A.,dc=siroe,dc=com"
"objectclass=*"
```

LDAP 検索フィルタ

検索フィルタを使って、検索結果として返されるエントリを選択することができます。検索フィルタは、通常 ldapsearch コマンド行ユーティリティと併用されます。ldapsearch を使用する場合は、あらかじめファイル内に複数のフィルタを格納しておくことができます。その場合、ファイル内では、各フィルタを別々の行に指定する必要があります。なお、検索フィルタは、コマンド行に直接指定することもできます。

たとえば、次に示すフィルタは、Babs Jesen という共通名を検索します。

```
cn=babs jensen
```

この検索フィルタは、Babs Jensen という共通名を含むすべてのエントリを返します。共通名の値の検索では、大文字と小文字は区別されません。

共通名属性が言語タグに関連付けられた値を持つ場合は、その値すべてが返されます。したがって、次の属性値は両方ともこのフィルタにマッチします。

```
cn: babs jensen
```

```
cn;lang-fr: babs jensen
```

サポートされている言語タグのリストについては、513 ページの表 D-1 を参照してください。

検索フィルタの構文

検索フィルタの基本的な構文を次に示します。

attribute operator value

たとえば、次のようにします。

```
buildingname>=alpha
```

この例では `buildingname` が属性、`>=` が演算子、**`alpha`** が値です。異なる属性をブール演算子と組み合わせたフィルタも定義できます。

以降の節では、検索フィルタについて詳しく説明します。

- 検索フィルタでの属性の使用
- 検索フィルタでの演算子の使い方
- 複合検索フィルタの使い方
- 検索フィルタの例

検索フィルタでの属性の使用

エントリを検索するときは、そのエントリのタイプに関連付けられた属性を指定できます。たとえば、人に関するエントリを検索する場合は、`cn` 属性を使用して特定の共通名を持つ人を探すことができます。

人に関するエントリに含まれる属性としては、次のようなものが考えられます。

- `cn` (共通名)
- `sn` (姓)
- `telephoneNumber` (電話番号)
- `buildingName` (居住する建物の名前)
- `l` (所属地域)

エントリのタイプに関連付けられた属性のリストについては、『iPlanet Directory Server スキーマリファレンス』を参照してください。

検索フィルタでの演算子の使い方

検索フィルタ内で使用できる演算子のリストについては、表 B-1 を参照してください。

表 B-1 検索フィルタ用演算子

検索タイプ	演算子	内容
等価	=	指定された値と完全にマッチする属性値を含むエントリを返す。例 : `cn=Bob Johnson`

部分文字列	=*string** *string*	指定された部分文字列を属性に含むエントリを返す。次に例を示す
		`cn=Bob*`
		`cn=*Johnson`
		`cn=*John*`
		`cn=B*John`
		(アスタリスク (*) は 0 個または 1 個以上の任意の数字を表す)

大きいまたは等しい	>=	属性が指定された値以上のエントリを返す。次に例を示す `buildingname>=alpha`
小さいまたは等しい	<=	属性が指定された値以下のエントリを返す。次に例を示す `buildingname<=alpha`

実在	=*	指定された属性に値が設定されたエントリを返す。次に例を示す
		`cn=*`
		`telephonenumber=*`
		`manager=*`

表 B-1　検索フィルタ用演算子 (続き)

検索タイプ	演算子	内容

近似	~=	検索フィルタで指定された値とほぼ等価を持ち、指定された属性を含むエントリを返す。次に例を示す
		`cn~=suret`
		`l~=san fransico`
		これらのフィルタからは、次の属性値を持つエントリが返される
		`cn=sarette`
		`l=san francisco`

注　これらの検索フィルタに加え、希望する言語の照合順序で動作する特殊なフィルタも指定できます。国際化文字セットを含むディレクトリの検索方法については、498 ページの「国際化ディレクトリの検索」を参照してください。

複合検索フィルタの使い方

次の例に示すように、ブール演算子を使って、複数のフィルタ要素を組み合わせることができます。

(*Boolean-operator* (*filter*) (*filter*) (*filter*) . . .)

ここで、*Boolean-operator* は、表 B-2 にリストされる論理演算子のうちのいずれか 1 つを表します。

さらに、複数のブール演算子を組み合わせたり入れ子状にして、次のように複雑な式を作成することもできます。

(*Boolean-operator* (*filter*) ((*Boolean-operator* (*filter*) (*filter*)))

検索フィルタで使用できるブール演算子には、次のものがあります。

表 B-2　検索フィルタ用ブール演算子

演算子	記号	内容
AND	&	ステートメントが true になるには、指定したすべてのフィルタ条件が true である必要がある。次に例を示す (&(filter)(filter)(filter)...)
OR	|	ステートメントが true になるには、指定したフィルタのうち、少なくとも 1 つが true である必要がある。次に例を示す (|(filter)(filter)(filter)...)
NOT	!	ステートメントが true になるには、指定したフィルタは false である必要がある。NOT 演算子が作用するフィルタは 1 つだけである。次に例を示す (!(filter))

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

検索フィルタの例

次に示すフィルタは、manager 属性に値が設定されたエントリを検索します。これは、実在検索とも呼ばれます。

```
manager=*
```

次に示すフィルタは、Ray Kultgen という共通名を持つエントリを検索します。これは、等価検索とも呼ばれます。

```
cn=Ray Kultgen
```

次に示すフィルタは、共通名が Ray Kultgen ではないすべてのエントリを返します。

```
(!(cn=Ray Kultgen))
```

次に示すフィルタは、説明属性に X.500 という部分文字列を含むすべてのエントリを返します。

```
description=*X.500*
```

次に示すフィルタは、組織単位が Marketing で、説明フィールドに X.500 という部分文字列を含まないすべてのエントリを返します。

```
(&(ou=Marketing)(!(description=*X.500*)))
```

次に示すフィルタは、組織単位が Marketing で、マネージャ属性が Julie Fulmer または Cindy Zwaska に一致するすべてのエントリを返します。

```
(&(ou=Marketing)(|(manager=cn=Julie
Fulmer,ou=Marketing,dc=siroe,dc=com)(manager=cn=Cindy
Zwaska,ou=Marketing,dc=siroe,dc=com)))
```

次に示すフィルタは、人を表すエントリ以外のすべてのエントリを返します。

```
(!(objectClass=person))
```

次に示すフィルタは、人を表すエントリ以外で、printer3b に似た共通名を持つすべてのエントリを返します。

```
(&(!(objectClass=person))(cn~=printer3b))
```

国際化ディレクトリの検索

検索を実行するときに、照合順序がサポートされた任意の言語に基づいて結果をソートするように指定できます。ディレクトリがサポートする照合順序のリストは、513 ページの「サポートされているロケールの特定」を参照してください。

ここでは、ldapsearch 構文のマッチング規則フィルタ部分について詳しく説明します。一般的な ldapsearch 構文については、493 ページの「LDAP 検索フィルタ」を参照してください。iPlanet Console のユーザおよびグループの部分を使用して国際化ディレクトリを検索する方法については、オンラインヘルプまたは『Managing Servers with iPlanet Console』を参照してください。

ここでは、次の項目について説明します。

- マッチング規則フィルタの構文
- サポートされている検索タイプ
- 国際化検索の例

マッチング規則フィルタの構文

検索中にディレクトリが文字列の比較を行う方法について、マッチング規則には特別なガイドラインが定められています。国際化検索の実行時に使用する照合順序と演算子は、マッチング規則によって指定されます。たとえば、国際化検索のマッチング規則は、スペイン語のマッチング順序で「llama」以降にある属性値を検索するように、サーバに命令します。マッチング規則フィルタの構文は次のとおりです。

attr:*matchingRule*:=*value*

各オプションは、次のように指定します。

- *attr* は、cn や mail など、検索対象となるエントリに属する属性
- *matchingRule* は、照合順序、または照合順序と関係演算子を識別する文字列。この指定はユーザが使用する形式によって異なる。マッチング規則形式については、499 ページの「マッチング規則の形式」を参照
- *value* は検索対象の属性値、または関係演算子と検索対象となる属性値の組み合わせ。フィルタの値部分の構文は、どのマッチング規則形式を使用するかによって異なる

マッチング規則の形式

検索フィルタのマッチング規則部分は、いくつかの異なる方法で記述できます。どれを使用するかは、個人の好みで選択できます。マッチング規則は、次に示す方法で記述できます。

- 検索ベースとするロケール用の照合順序の OID
- 検索ベースとする照合順序に関連付けられた言語タグ
- 照合順序の OID と関係演算子を表す接尾辞
- 照合順序に関連付けられた言語タグと、関係演算子を表す接尾辞

以降の項目では、各オプションの構文について説明します。

- マッチング規則での OID の使用
- マッチング規則での言語タグの使用
- マッチング規則での OID と接尾辞の使用
- マッチング規則での言語タグと接尾辞の使用

マッチング規則での OID の使用

Directory Server によってサポートされる各ロケールには、関連付けられた照合順序 OID があります。ディレクトリサーバによってサポートされるロケールと関連付けられた OID のリストについては、513 ページの表 D-1 を参照してください。

照合順序 OID は、マッチング規則フィルタのマッチング規則部分で、次のように使用できます。

attr:*OID*:=(*relational_operator value*)

関係演算子は式の値部分に含まれ、1 つの空白で値とは区切られます。たとえば、スウェーデン語照合順序で N4709 以降にあるすべての departmentNumber 属性を検索するには、次のようなフィルタを使用します。

```
departmentNumber:2.16.840.1.113730.3.3.2.46.1:=>= N4709
```

マッチング規則での言語タグの使用

Directory Server によってサポートされる各ロケールには、関連付けられた言語タグがあります。Directory Server によってサポートされるロケールと関連付けられた言語タグのリストについては、513 ページの表 D-1 を参照してください。

言語タグは、マッチング規則フィルタのマッチング規則部分で、次のように使用できます。

attr:*language-tag*:=(*relational_operator value*)

関係演算子は式の値部分に含まれ、1 つの空白で値とは区切られます。たとえば、スペイン語照合順序で `estudiante` という値を持つすべての記述属性をディレクトリで検索するには、次のようなフィルタを使用します。

```
cn:es:== estudiante
```

マッチング規則での OID と接尾辞の使用

関係演算子と値の組み合わせを使用する代わりに、フィルタのマッチング規則部分にある OID に、特定の演算子を表す接尾辞を追加することもできます。OID と接尾辞は、次のように組み合わせます。

attr:*OID+suffix*:=*value*

たとえば、ドイツ語照合順序で `softwareproduckte` という値を持つ `businessCategory` 属性を検索するには、次のようなフィルタを使用します。

```
businessCategory:2.16.840.1.113730.3.3.2.7.1.3:=softwareprodukte
```

この例で、「`.3`」は等価を表す接尾辞です。

Directory Server によってサポートされるロケールと関連付けられた OID のリストについては、513 ページの表 D-1 を参照してください。関係演算子と、それに対応する接尾辞のリストについては、501 ページの表 B-3 を参照してください。

マッチング規則での言語タグと接尾辞の使用

関係演算子と値の組み合わせを使用する代わりに、フィルタのマッチング規則部分にある言語タグに、特定の演算子を表す接尾辞を追加することもできます。言語タグと接尾辞は、次のように組み合わせます。

attr:*language-tag+suffix*:=*value*

たとえば、フランス語照合順序で `La Salle` 以降にあるすべての姓を検索するには、次のようなフィルタを使用します。

```
sn:fr.4:=La Salle
```

ディレクトリサーバによってサポートされるロケールと関連付けられた言語タグのリストについては、513 ページの表 D-1 を参照してください。関係演算子と、それに対応する接尾辞のリストについては、501 ページの表 B-3 を参照してください。

マッチング規則フィルタでのワイルドカードの使用

マッチング規則フィルタを使用して部分文字列検索を実行するときに、ワイルドカードとしてアスタリスク (*) を使用できます。アスタリスクは、0 個以上の文字を表します。

たとえば、`l` という文字で始まって `n` で終わる属性値を検索するには、検索フィルタの値部分に `l*n` と入力します。同様に、`u` で始まるすべての属性値を検索するには、検索フィルタの値部分に `u*` と入力します。

アスタリスク (*) の文字そのものを含む値を検索するには、エスケープキャラクタ \5c2a を使用して、そのアスタリスクをワイルドカードと区別する必要があります。たとえば、`businessCategory` 属性値が `Siroe*Net product line` である従業員全員を検索するには、次のような検索フィルタを指定します。

```
Siroe\2a*Net product line
```

サポートされている検索タイプ

Directory Server では、次に示すタイプの国際化検索がサポートされています。

- 等価 (=)
- 部分文字列 (*)
- 大きい (>)
- 大きいまたは等しい (>=)
- 小さい (<)
- 小さいまたは等しい (<=)

近似、発音、属性による検索がサポートされているのは英語だけです。

通常の `ldapsearch` 検索と同様に、国際化検索時は演算子を使用して検索タイプを決定します。ただし、国際化検索を実行するときに、検索文字列の値部分に標準演算子 (=、>=、>、<、<=) を使用するか、フィルタのマッチング規則部分に接尾辞 (ディレクトリ接尾辞とは異なる) と呼ばれる特別なタイプの演算子を使用できます。表 B-3 に検索タイプ、演算子、および等価な接尾辞の概要を示します。

表 B-3 検索タイプ、演算子、および接尾辞

検索タイプ	演算子	接尾辞
小さい	<	.1
小さいまたは等しい	<=	.2
等価	=	.3

表 B-3　検索タイプ、演算子、および接尾辞 (続き)

検索タイプ	演算子	接尾辞
大きいまたは等しい	>=	.4
大きい	>	.5
部分文字列	*	.6

国際化検索の例

以降の節では、ディレクトリデータに対する国際化検索の実行例を紹介します。それぞれの例には、使用可能なすべてのマッチング規則フィルタの形式を示してあるので、さまざまな形式の詳細を確認してから最適なフィルタを選ぶことができます。

検索タイプを「小さい」にした場合の例

「小さい」演算子 (<) または接尾辞 (.1) を使用してロケール固有の検索を実行すると、特定の照合順序内の指定された属性よりも前にあるすべての属性値が検索されます。

たとえば、スペイン語の照合順序で Marquez よりも前にあるすべての姓を検索するには、次のいずれかのマッチング規則フィルタを使用します。

```
sn:2.16.840.1.113730.3.3.2.15.1:=< Marquez
sn:es:=< Marquez
sn:2.16.840.1.113730.3.3.2.15.1.1:=Marquez
sn:es.1:=Marquez
```

検索タイプを「小さいまたは等しい」にした場合の例

「小さいまたは等しい」演算子 (<=) または接尾辞 (.2) を使用してロケール固有の検索を実行すると、特定の照合順序内の指定された属性と同じか、前にあるすべての属性値が検索されます。

たとえば、ハンガリー語の照合順序で部屋番号 CZ422 と同じかそれより前にあるすべての部屋番号を検索するには、次のいずれかのマッチング規則フィルタを使用します。

```
roomNumber:2.16.840.1.113730.3.3.2.23.1:=<= CZ422
roomNumber:hu:=<= CZ422
roomNumber:2.16.840.1.113730.3.3.2.23.1.2:=CZ422
roomNumber:hu.2:=CZ422
```

検索タイプを「等価」にした場合の例

「等価」演算子 (=) または接尾辞 (.3) を使用してロケール固有の検索を実行すると、特定の照合順序内の指定された属性にマッチするすべての属性値が検索されます。

たとえば、ドイツ語の照合順序で softwareprodukte という値を持つ businessCategory 属性を検索するには、次に示す任意のマッチング規則フィルタを使用します。

```
businessCategory:2.16.840.1.113730.3.3.2.7.1:== softwareprodukte
businessCategory:de:== softwareprodukte
businessCategory:2.16.840.1.113730.3.3.2.7.1.3:=softwareprodukte
businessCategory:de.3:=softwareprodukte
```

検索タイプを「大きいまたは等しい」にした場合の例

「大きいまたは等しい」演算子 (>=) または接尾辞 (.4) を使用してロケール固有の検索を実行すると、特定の照合順序内の指定された属性の位置と同じか、その後にあるすべての属性値が検索されます。

たとえば、フランス語の照合順序で Québec と同じ位置かそれ以降にあるすべての地域を検索するには、次のいずれかのマッチング規則フィルタを使用します。

```
locality:2.16.840.1.113730.3.3.2.18.1:=>= Québec
locality:fr:=>= Québec
locality:2.16.840.1.113730.3.3.2.18.1.4:=Québec
locality:fr.4:=Québec
```

検索タイプを「大きい」にした場合の例

「大きい」演算子 (>) または接尾辞 (.5) を使用してロケール固有の検索を実行すると、特定の照合順序内の指定された属性のあとにあるすべての属性値が検索されます。

たとえば、チェコスロバキア語の照合順序で schranka4 よりもあとにあるすべてのメールホストを検索するには、次のいずれかのマッチング規則フィルタを使用します。

```
mailHost:2.16.840.1.113730.3.3.2.5.1:=> schranka4
mailHost:cs:=> schranka4
mailHost:2.16.840.1.113730.3.3.2.5.1.5:=schranka4
mailHost:cs.5:=schranka4
```

検索タイプを部分文字列にした場合の例

部分文字列を使用した国際化検索を実行すると、指定された照合順序のパターンにマッチするすべての値が検索されます。

たとえば、中国語の照合順序で ming で終わるすべてのユーザ ID を検索するには、次のいずれかのマッチング規則フィルタを使用します。

```
uid:2.16.840.1.113730.3.3.2.49.1:=* *ming
uid:zh:=* *ming
uid:2.16.840.1.113730.3.3.2.49.1.6:=* *ming
uid:zh.6:=* *ming
```

付録 C

LDAP URLs

LDAP クエリーを指定するには、URL を使用して Directory Server ホストマシンと DN または検索フィルタを指定する方法があります。iPlanet Directory Server では、LDAP URL として送信されたクエリーを処理し、結果を表す HTML ページを返します。これにより、匿名検索が許可されている場合は、Web ブラウザでディレクトリの検索を実行できます。

また、Directory Server のレフェラルやアクセス制御命令を管理する場合も、LDAP URL を使用してターゲットエントリを指定します。

この付録は、次の節で構成されています。

- LDAP URL のコンポーネント
- 安全でない文字のエスケープ
- LDAP URL の例

LDAP URL のコンポーネント

LDAP URL の構文は次のとおりです。

`ldap[s]://`*hostname*`:`*port*`/`*base_dn*`?`*attributes*`?`*scope*`?`*filter*

`ldap://` プロトコルは、セキュリティで保護されていない接続を経由して LDAP サーバに接続する場合に使用されます。また、`ldaps://` プロトコルは、SSL 接続を経由して LDAP サーバに接続する場合に使用されます。表 C-1 に、LDAP URL のコンポーネントを示します。

表 C-1　LDAP URL のコンポーネント

コンポーネント	内容
hostname	LDAP サーバの名前 (またはピリオドで区切った IP アドレス)。たとえば、次のようにする `ldap.siroe.com` or `192.202.185.90`
port	LDAP サーバのポート番号 (例 : 696) port を指定しないと、標準の LDAP ポート (389) または LDAPS ポート (636) が使用される
base_dn	ディレクトリ内のエントリの識別名 (DN)。この DN は、検索の開始ポイントとなるエントリを識別する base DN を指定しないと、検索はディレクトリツリーのルートから開始される
attributes	検索結果として返される属性。2 つ以上の属性を指定する場合は、コンマ (,) を使用して各属性を区切る (例 : 「`"cn,mail,telephoneNumber"`」) attributes を URL に指定しないと、すべての属性が返される
scope	検索の適用範囲。次のいずれかの値をとる • `base` を指定すると、URL に指定された識別名 (*base_dn*) に関するデータだけを検出する • `one` を指定すると、URL に指定された識別名 (*base_dn*) の 1 つ下のレベルにあるエントリに関するデータを検出する。ベースエントリは、この適用範囲には含まれない • `sub` を指定すると、URL に指定された識別名 (*base_dn*) のすべての下位レベルにあるエントリに関するデータを検出する。ベースエントリは、この適用範囲に含まれる scope を指定しないと、`base` 検索が実行される
filter	検索の指定範囲内のエントリに適用するための検索フィルタ filter を指定しないと、フィルタ (`objectClass=*`) が使用される

attributes、scope、および filter の各コンポーネントは、URL 中の位置によって識別されます。属性をまったく指定しない場合も、フィールドの区切り文字として疑問符 (?) を挿入する必要があります。

たとえば、「"(sn=Jensen)"」にマッチするエントリの属性をすべて返すサブツリー検索が「"dc=siroe,dc=com"」から開始されるように指定するには、次の LDAP URL を使用します。

```
ldap://ldap.siroe.com/dc=siroe,dc=com??sub?(sn=Jensen)
```

連続した 2 つの疑問符 ?? は、attributes が指定されていないことを表します。この場合、URL で特定の属性が識別されないので、検索結果としてすべての属性が返されます。

安全でない文字のエスケープ

URL 中の「安全でない」文字はすべて特別な文字列で表す必要があります。このことを、安全でない文字のエスケープと呼びます。

たとえば、空白文字は「安全でない」文字に含まれるので、URL 中では %20 と表す必要があります。したがって、識別名「"o=siroe.com corporation"」は「"o=siroe.com%20corporation"」と符号化する必要があります。

次の表に、URL 中で使用する場合に「安全でない」とみなされる文字と、代わりに使用するエスケープ文字を示します。

「安全でない」文字	エスケープ文字
空白文字	%20
<	%3c
>	%3e
"	%22
#	%23
%	%25
{	%7b
}	%7d
|	%7c
\	%5c
^	%5e
~	%7e
[	%5b
]	%5d

「安全でない」文字	エスケープ文字
‘	%60

LDAP URL の例

- 次の LDAP URL では、識別名 dc=siroe,dc=com を持つエントリのベース検索を指定する

```
ldap://ldap.siroe.com/dc=siroe,dc=com
```

  - ポート番号が指定されていないので、標準の LDAP ポート番号 (389) が使用される
  - 属性が指定されていないので、検索結果にはすべての属性が含まれる
  - 検索の適用範囲が指定されていないので、検索処理はベースエントリ dc=siroe,dc=com に限定される
  - フィルタが指定されていないので、ディレクトリにはデフォルトのフィルタ (objectclass=*) が使用される

- 次の LDAP URL では、識別名 dc=siroe,dc=com を持つエントリの postalAddress 属性を検出する

```
ldap://ldap.siroe.com/dc=siroe,dc=com?postalAddress
```

  - 検索の適用範囲が指定されていないので、検索処理はベースエントリ dc=siroe,dc=com に限定される
  - フィルタが指定されていないので、ディレクトリにはデフォルトのフィルタ (objectclass=*) が使用される

- 次の LDAP URL では、Barbara Jensen のエントリの cn、mail、および telephoneNumber 属性を検出する

```
ldap://ldap.siroe.com/cn=Barbara%20Jensen,dc=siroe,dc=com?cn,mail, telephoneNumber
```

  - 検索の適用範囲が指定されていないので、検索処理はベースエントリ cn=Barbara Jensen,dc=siroe,dc=com に限定される
  - フィルタが指定されていないので、ディレクトリにはデフォルトのフィルタ (objectclass=*) が使用される

- 次の LDAP URL では、dc=siroe,dc=com のすべての下位レベルを対象に、Jensen という姓を持つエントリの検索を指定する

```
ldap://ldap.siroe.com/dc=siroe,dc=com??sub?(sn=Jensen)
```

  - 属性が指定されていないので、検索結果にはすべての属性が含まれる

- 検索の適用範囲に sub が指定されているので、検索対象にはベースエントリ dc=siroe,dc=com とそのすべての下位レベルにあるエントリが含まれる

- 次の LDAP URL では、dc=siroe,dc=com の 1 つ下のレベルにあるすべてのエントリに対するオブジェクトクラスの検索を指定する

```
ldap://ldap.siroe.com/dc=siroe,dc=com?objectClass?one
```

- 検索の適用範囲に one が指定されているので、検索対象にはベースエントリ dc=siroe,dc=com の 1 つ下のレベルにあるすべてのエントリが含まれる。ただし、ベースエントリそのものは、検索対象には含まれない
- フィルタが指定されていないので、ディレクトリにはデフォルトのフィルタ (objectclass=*) が使用される

注 LDAP URL の構文では、資格やパスワードを指定できません。LDAP URL による検索の処理要求は、LDAP URL をサポートする LDAP クライアントが認証メカニズムを提供しない限り、LDAP URL は未認証 (匿名) となります。

付録 D

国際化

Directory Server では、さまざまな言語を使用して、エントリや関連する属性を格納、管理、および検索できます。多言語化ディレクトリは、会社にとって貴重な資産であり、従業員やビジネスパートナーが理解できる言語で、必要な情報をすばやく提供します。

ディレクトリデータは UTF-8 で格納されているので、ディレクトリはすべての国際的な文字集合をデフォルトでサポートします。さらに Directory Server における検索操作では、言語の設定に基づいて、マッチング規則や照合順序を指定することができます。

注	属性名とオブジェクトクラス名には ASCII 文字を使用する必要があります。

この付録は、次の節で構成されています。

- ロケールについて
- サポートされているロケールの特定
- サポートされている言語サブタイプ

ロケールについて

Directory Server では、ロケールを使用して、複数の言語をサポートしています。ロケールは、ある言語によるデータの変換方法、格納方法、照合順序など、特定の地域、文化、風習を持つユーザに合わせたデータの表示方法について、言語固有の情報を特定します。

また、ロケール情報は、指定された言語を表すために必要なコードページも特定します。コードページとは、キーボードのキーと画面に表示される文字フォントを関連付けるために、オペレーティングシステムによって使用される内部テーブルのことです。

ロケールで定義される項目は、具体的には次のとおりです。

- 照合順序

  ある言語の文字のソート方法に関する、言語および文化に固有の情報を提供します。照合順序は、アルファベットで文字の順序を指定するようなもので、アクセント付きの文字をアクセントのない文字と比較する方法、文字列を比較する際に無視してもかまわない文字の有無などを示します。また、文章の方向 (左から右、右から左、または上下) などの文化に固有の情報も考慮されます。

- 文字タイプ

  文字タイプは、英字を数字やほかの文字と区別します。さらに、大文字と小文字のマッピングも定義されます。たとえば、パイプ文字 (|) を句読点と解釈する言語もあれば、アルファベットと解釈する言語もあります。

- 通貨形式

  特定の地域で使用されている通貨記号、通貨記号が値の前と後ろのどちらに付くのか、および通貨単位の表現形式を指定します。

- 時刻 / 日付の形式

  特定の地域における日付と時刻の、習慣的な表示形式を決定します。時刻形式は、地域が 12 時間法と 24 時間法のどちらを使用するかを示します。日付形式には、`MM/dd/yy` (月、日、年) または `dd/MM/yy` (日、月、年) などの短い日付形式、および指定された言語での月および曜日の名前を含む長い日付形式があります。たとえば、「January 10, 1996」は、チェコ語では「10. leden 1996」、フランス語では「10 janvier 1996」と表示されます。

ロケールは文化、習慣、地域などの違いに加え、言語の機械的な違いも表しているので、ディレクトリデータをユーザが理解できる特定言語に翻訳できることに加えて、データを特定地域のユーザが期待する方法で表現することができます。

ロケール情報は、Directory Server のインストール時に自動的に次のディレクトリにコピーされます。

```
/usr/iplanet/ds5/lib/nls/locale31
```

サポートされているロケールの特定

検索操作など、でロケールの指定が必要なディレクトリ操作を実行するときには、言語タグや照合順序 OID (オブジェクト識別子) を使用することができます。

言語タグは、言語を識別する小文字 2 文字で始まる文字列で、ISO 標準 639 で定義されています。言語の地域的な違いを区別する必要がある場合は、言語タグに国コード (ISO 標準 3166 で定義されている 2 文字の文字列) を入れることもできます。言語コードと国コードはハイフン (-) で区切られます。たとえば、イギリス英語のロケールを特定する言語タグは en-GB です。

OID は、属性やオブジェクトクラスなどのオブジェクトを一意に識別するために使用される 10 進数値です。国際化ディレクトリの検索やインデックス付けに使用される OID は、Directory Server でサポートされている固有の照合順序を示します。たとえば、OID `2.16.840.1.113730.3.3.2.17.1` はフィンランド語の照合順序を示します。

ディレクトリで多言語検索を実行する場合は、使用する照合順序を表すために、言語タグまたは OID が使用されます。ただし、国際化インデックスを設定する場合は、OID を使用する必要があります。詳細については、第 10 章「インデックスの管理」を参照してください。

次の表に、Directory Server でサポートされているロケールと、関連する言語タグおよび OID を示します。

表 D-1 サポートされているロケール

ロケール	言語タグ	照合順序 OID (オブジェクト識別子)
アルバニア語	`sq`	`2.16.840.1.113730.3.3.2.44.1`
アラビア語	`ar`	`2.16.840.1.113730.3.3.2.1.1`
ベラルーシ語	`be`	`2.16.840.1.113730.3.3.2.2.1`
ブルガリア語	`bg`	`2.16.840.1.113730.3.3.2.3.1`
カタロニア語	`ca`	`2.16.840.1.113730.3.3.2.4.1`
中国語 (簡体字)	`zh`	`2.16.840.1.113730.3.3.2.49.1`
中国語 (繁体字)	`zh-TW`	`2.16.840.1.113730.3.3.2.50.1`
クロアチア語	`hr`	`2.16.840.1.113730.3.3.2.22.1`
チェコスロバキア語	`cs`	`2.16.840.1.113730.3.3.2.5.1`
デンマーク語	`da`	`2.16.840.1.113730.3.3.2.6.1`
英語 (米国)	`en` または `en-US`	`2.16.840.1.113730.3.3.2.11.1`

表 D-1　サポートされているロケール (続き)

ロケール	言語タグ	照合順序 OID (オブジェクト識別子)
エストニア語	et	2.16.840.1.113730.3.3.2.16.1
フィンランド語	fi	2.16.840.1.113730.3.3.2.17.1
フランス語	fr または fr-FR	2.16.840.1.113730.3.3.2.18.1
ドイツ語	de	2.16.840.1.113730.3.3.2.7.1
ギリシア語	el	2.16.840.1.113730.3.3.2.10.1
ヘブライ語	iw	2.16.840.1.113730.3.3.2.27.1
ハンガリー語	hu	2.16.840.1.113730.3.3.2.23.1
アイスランド語	is	2.16.840.1.113730.3.3.2.24.1
日本語	ja	2.16.840.1.113730.3.3.2.28.1
韓国語	ko	2.16.840.1.113730.3.3.2.29.1
ラトビア語、レット語	lv	2.16.840.1.113730.3.3.2.31.1
リトアニア語	lt	2.16.840.1.113730.3.3.2.30.1
マケドニア語	mk	2.16.840.1.113730.3.3.2.32.1
ノルウェー語	no	2.16.840.1.113730.3.3.2.35.1
ポーランド語	pl	2.16.840.1.113730.3.3.2.38.1
ルーマニア語	ro	2.16.840.1.113730.3.3.2.39.1
ロシア語	ru	2.16.840.1.113730.3.3.2.40.1
セルビア語 (キリル文字)	sr	2.16.840.1.113730.3.3.2.45.1
セルビア語 (ラテン文字)	sh	2.16.840.1.113730.3.3.2.41.1
スロバキア語	sk	2.16.840.1.113730.3.3.2.42.1
スロベニア語	sl	2.16.840.1.113730.3.3.2.43.1
スペイン語	es または es-ES	2.16.840.1.113730.3.3.2.15.1
スウェーデン語	sv	2.16.840.1.113730.3.3.2.46.1
トルコ語	tr	2.16.840.1.113730.3.3.2.47.1
ウクライナ語	uk	2.16.840.1.113730.3.3.2.48.1

サポートされている言語サブタイプ

クライアントは言語サブタイプを使用して、検索する特定の値を判断することができます。言語サブタイプの使い方については、49 ページの「属性のサブタイプの追加」を参照してください。

次の表に、サポートされている言語サブタイプを示します。

表 D-2 サポートされている言語サブタイプ

言語タグ	言語
af	アフリカーンス語
be	ベラルーシ語
bg	ブルガリア語
ca	カタロニア語
cs	チェコスロバキア語
da	デンマーク語
de	ドイツ語
el	ギリシア語
en	英語
es	スペイン語
eu	バスク語
fi	フィンランド語
fo	フェロー語
fr	フランス語
ga	アイルランド語
gl	ガリシア語
hr	クロアチア語
hu	ハンガリー語
id	インドネシア語
is	アイスランド語
it	イタリア語
ja	日本語
ko	韓国語

表 D-2　サポートされている言語サブタイプ (続き)

言語タグ	言語
nl	オランダ語
no	ノルウェー語
pl	ポーランド語
pt	ポルトガル語
ro	ルーマニア語
ru	ロシア語
sk	スロバキア語
sl	スロベニア語
sq	アルバニア語
sr	セルビア語
sv	スウェーデン語
tr	トルコ語
uk	ウクライナ語
zh	中国語

用語集

ACI　Access Control Instruction の略称。ディレクトリ内のエントリに対するアクセス権を許可または拒否する命令。

ACL　アクセス制御リスト。ディレクトリへのアクセスを制御するメカニズム。

CA　「認証局 (Certificate Authority)」を参照。

ciphertext　この情報を復号化する適切な鍵がないと読むことができない、暗号化された情報。

CIR　「コンシューマ主導レプリケーション (consumer-initiated replication)」を参照。

CoS　アプリケーションに認識されない方法で、エントリ間で属性を共有する方法。

CoS 定義エントリ (CoS definition entry)　使用中の CoS のタイプを特定する。対象とする分岐の下に LDAP サブエントリとして格納される。

CoS テンプレートエントリ (CoS template entry)　共有属性値のリストを含む。

DAP　Directory Access Protocol の略称。クライアントがディレクトリにアクセスするための ISO X.500 標準プロトコル。

Directory Access Protocol　「DAP」を参照。

Directory Server Console　ディレクトリの内容を参照、構成、および管理するためのグラフィカルユーザインタフェースを提供する LDAP クライアントアプリケーション。iPlanet Directory Server 製品のコンポーネント。

DIT　「ディレクトリツリー (directory tree)」を参照。

DM　「ディレクトリマネージャ (Directory Manager)」を参照。

DN　「識別名 (distinguished name)」を参照。

DNS　ドメインネームシステム。標準の IP アドレス (198.93.93.10 など) をホスト名 (`www.iPlanet.com` など) と関連付けるためにネットワーク上のマシンが使用するシステム。マシンは通常、ホスト名の IP アドレスを DNS サーバから取得するか、システム上で維持されているテーブルから検索する。

DNS エイリアス (DNS alias)　DNS サーバが認識しているホスト名で、別のホスト (特に、DNS CNAME レコード) をポイントするための文字列。マシンは常に実際の名前を 1 つ持つが、1 つ以上のエイリアスを持つこともできる。たとえば、`www.[yourdomain].[domain]` などのエイリアスは、現在サーバが存在する `realthing.[yourdomain].[domain]` という名前の実際のマシンをポイントできる。

HTML　ハイパーテキストマークアップ言語。World Wide Web 上のドキュメントで使用されるフォーマット化言語。HTML ファイルはフォーマット化コードを含むプレーンテキストファイルであり、Netscape Navigator などのブラウザにテキストの表示方法、グラフィックの配置方法、および項目の配列方法を指示し、ほかのページへのリンクを表示する。

HTTP　ハイパーテキスト転送プロトコル。HTTP サーバとクライアントの間で情報を交換するメソッド。

HTTP-NG　次世代のハイパーテキスト転送プロトコル。

HTTPD　HTTP デーモンまたはサービスの略称で、HTTP プロトコルを使用して情報を提供するプログラム。一般に、このデーモンまたはサービスは、httpd と呼ばれる。

HTTPS　セキュリティ保護を強化した HTTP。SSL (Secure Sockets Layer) を使用して実装される。

IP アドレス (IP address)　インターネットプロトコルアドレス。ドットで区切られた一組の数字で、インターネット上にあるマシンの実際の位置を指定する。たとえば、198.93.93.10 など。

ISO　国際標準化機構。

LDAP　Lightweight Directory Access Protocol の略称。TCP/IP を介して複数のプラットフォーム間で動作するように設計されたディレクトリサービスプロトコル。

LDAP Data Interchange Format　「LDAP Data Interchange Format」を参照。

LDAP URL　DNS を使用して Directory Server を検出し、LDAP を介して照会を完了する方法を提供する。たとえば、`ldap://ldap.iplanet.com` など。

LDAP クライアント (LDAP client)　LDAP Directory Server からの LDAP エントリを要求および表示するために使用されるソフトウェア。「ブラウザ (browser)」も参照。

LDAPv3　LDAP プロトコルのバージョン 3。Directory Server のスキーマ形式は、このプロトコルに基づく。

LDBM データベース (LDBM database)　高性能なディスクベースのデータベースで、このデータベースに割り当てられたすべてのデータを含む一連の大きなファイルで構成される。Directory Server の主となるデータの保存先。

LDIF　LDAP Data Interchange Format の略称。Directory Server のエントリをテキスト形式で表すために使用される形式。

Lightweight Directory Access Protocol　「LDAP」を参照。

MD5　RSA Data Security, Inc. によるメッセージダイジェストアルゴリズム。データの短いダイジェストの生成に使用できる。このダイジェストは、高い確率で一意となるため、同じメッセージダイジェストを生成するデータの作成は、数学的に見て非常に困難である。

MD5 シグニチャ (MD5 signature)　MD5 アルゴリズムで生成されたメッセージダイジェスト。

MIB　管理情報ベース。SNMP ネットワークと関連付けられたすべてのデータ、またはその一部。MIB は、すべての SNMP 管理対象オブジェクトの定義を含むデータベースとみなすことができる。MIB は、ツリーに似た階層を持つ。最上位にはネットワークに関する最も一般的な情報が含まれており、下位では個別のネットワーク領域に固有の情報を扱う。

MIB ネームスペース (MIB namespace)　管理情報ネームスペース。ディレクトリのデータに名前を設定し、参照する方法。ディレクトリツリーとも呼ばれる。

N + 1 ディレクトリ問題 (n + 1 directory problem)　さまざまなディレクトリで同じ情報の複数のインスタンスを管理する場合の問題。結果的に、ハードウェアにかかる費用と人的費用が増大する。

Network Management Station　「NMS」を参照。

NIS　Network Information Service の略称。UNIX マシンが制御する、プログラムとデータファイルから構成されるシステムで、コンピュータネットワーク全体のマシンやユーザ、ファイルシステム、およびネットワークパラメタを対象に、各マシン固有の情報に関する収集や照合、共有のサービスを提供する。

NMS　Network Management Station の略称。1 つ以上のネットワーク管理アプリケーションがインストールされたパワフルなワークステーション。

ns-slapd　iPlanet LDAP Directory Server のデーモンまたはサービスで、Directory Server のすべてのアクションに関連する。「slapd」も参照。

OID　「オブジェクト識別子 (object identifier)」を参照。

PDU　Protocol Data Unit の略称。SNMP デバイス間のデータ交換の基礎となる符号化されたメッセージ。

Protocol Data Unit　「PDU」を参照。

PTA　パススルー認証。バインド資格を確認するために、1 つの Directory Server がほかの Directory Server と交信するメカニズム。

PTA directory server　パススルー認証 (PTA) において、受信したバインド要求を認証ディレクトリサーバに送信 (パススルー) するサーバ。

PTA LDAP URL　パススルー認証において、認証ディレクトリサーバ (Authenticating Directory Server)、パススルーサブツリー、および省略可能なパラメタを定義する URL。

RAM　ランダムアクセスメモリ。コンピュータ内部にあり、多数の半導体で構成された物理的な記憶装置。RAM 内に格納されている情報は、コンピュータが停止すると消失する。

RDN　相対識別名。完全な識別名を形成するために文字列にエントリの祖先を追加する前の、エントリ自体の名前。

RFC　Request For Comments の略称。インターネットコミュニティに提出される手順あるいは標準文書。技術が標準として受け入れられる前に、ユーザは技術に関してコメントを送ることができる。

root　Unix マシン上で最も高いレベルの特権を持つユーザ。root ユーザは、マシン上のすべてのファイルに対して完全なアクセス特権を持つ。

Secure Sockets Layer　「SSL」を参照。

SIE　サーバインスタンスのエントリ。

SIR　「サプライヤ主導レプリケーション (supplier-initiated replication)」を参照。

slapd　LDAP Directory Server のデーモンまたはサービス。レプリケーション以外のディレクトリのほとんどの機能を受け持つ。「ns-slapd」も参照。

SNMP　簡易ネットワーク管理プロトコル。ネットワーク処理に関するデータを交換することによって、サーバ上で実行しているアプリケーションプロセスを監視および管理するために使用される。

SNMP サブエージェント (SNMP subagent)　管理対象のデバイスに関する情報を収集し、その情報をマスターエージェントに渡すソフトウェア。

SNMP マスターエージェント (SNMP master agent)　さまざまなサブエージェントと NMS の間で情報を交換するソフトウェア。

SSL　Secure Sockets Layer の略称。クライアントとサーバとの間にセキュリティ保護された接続を確立するソフトウェアライブラリ。セキュリティ保護が強化された HTTP である HTTPS の実装に使用される。

TCP/IP　Transmission Control Protocol/Internet Protocol の略称。インターネットや企業内ネットワークにおける主要なネットワークプロトコル。

TLS　Transport Layer Security の略称。SSL の新標準で、公開鍵に基づいたプロトコル。

Transport Layer Security　「TLS」を参照。

uid　Unix システム上で、各ユーザと関連付けられた一意の番号。

URL　Uniform Resource Locator の略称。サーバおよびクライアントが文書の要求に使用するアドレス指定システム。ロケーションとも呼ばれる。URL の形式は、`[protocol]://[machine:port]/[document]`。ポート番号は一部のサーバでのみ必要であり、多くの場合サーバによって割り当てられるので、その場合ユーザは URL でポート番号を指定する必要はない。

X.500 標準 (X.500 standard)　Directory Server の実装で使用される、推奨する情報モデル、オブジェクトクラス、および属性を概説する、一連の ISO/ITU-T 文書。

アカウントの無効化 (account inactivation)　ユーザアカウント、アカウントのグループ、またはドメイン全体を無効にして、すべての認証の試行に対して、自動的に拒否するようにする。

アクセス権 (permission)　アクセス制御のコンテキスト内で、ディレクトリ情報へのアクセスの許可または拒否、および許可または拒否されるアクセスのレベルを規定する。「アクセス権限」も参照。

アクセス権限 (access rights)　アクセス制御に関しては、許可または拒否されているアクセスのレベルを指す。アクセス権限は、ディレクトリで実行できる操作のタイプと関連している。読み取り、書き込み、追加、削除、検索、比較、本人による書き込み、プロキシなど、すべての権利を許可または拒否できる。

アクセス制御命令 (access control instruction)　「ACI」を参照。

アクセス制御リスト (access control list)　「ACL」を参照。

入れ子のロール (nested role)　ほかのロールを含むロールの作成が可能。

インデックスキー (index key)　ディレクトリが使用する各インデックスは、インデックスキーのテーブルとマッチングエントリ ID リストで構成されている。

エントリ (entry)　オブジェクトに関する情報を含む LDIF ファイル内の行のグループ。

エントリ ID リスト (entry ID list) ディレクトリが使用する各インデックスは、インデックスキーのテーブルとマッチングエントリ ID リストで構成されている。エントリ ID リストは、クライアントアプリケーションの検索要求とマッチする可能性があるエントリ候補のリストを構築するために、ディレクトリが使用する。

エントリの配布 (entry distribution) 多数のエントリをサポートできるようスケーリングするために複数のサーバにディレクトリエントリを配布するメソッド。

オブジェクトクラス (object class) どの属性がそのエントリ内に含まれるのかを定義することにより、ディレクトリ内のエントリのタイプを定義する。

オブジェクト識別子 (object identifier) オブジェクト指向システムにおいて、オブジェクトクラスや属性などのスキーマ要素を一意に特定する、通常 10 進数の数字の文字列。オブジェクト識別子は、ANSI、IETF、または同様の組織が割り当てる。

親アクセス (parent access) この権限が与えられた場合、バインド DN がアクセス先のエントリの親であるときに、ユーザディレクトリツリー内で自分の下にあるエントリにアクセスできることを示す。

カスケード型レプリケーション (cascading replication) カスケード型レプリケーションでは、1 つのサーバ (しばしばハブサプライヤと呼ばれる) が特定のレプリカでコンシューマとサプライヤの両方として動作する。このサーバは読み取り専用のレプリカを保持し、更新履歴ログを管理する。また、データのマスターコピーを保持するサプライヤサーバから更新を受け取り、次にコンシューマにこの更新を供給する。

仮想リスト表示インデックス (virtual list view index) ブラウズインデックスとも呼ばれる。Directory Server Console でエントリ内の表示を高速化する。仮想リスト表示インデックスは、表示の性能を向上させるために、ディレクトリツリー内のすべての分岐点で作成可能。

簡易ネットワーク管理プロトコル (Simple Network Management Protocol) 「SNMP」を参照。

間接 CoS (indirect CoS) 間接 CoS は、ターゲットエントリの属性のうちの 1 つの値を使用してテンプレートエントリを特定する。

管理されているロール (managed role) ユーザは、メンバーの明示的な列挙リストを作成できる。

管理情報ベース (management information base) 「MIB」を参照。

管理対象オブジェクト (managed object) SNMP エージェントがアクセス可能で、NMS に対しても送信できる標準値。各管理対象オブジェクトは、ドット表記法で表現される正式名および数字の識別子で識別される。

近似インデックス (approximate index)　類似あるいは音の近い用語を探すのに有効な近似検索を許可する。

クライアント (client)　「LDAP クライアント (LDAP client)」を参照。

クラシック CoS (classic CoS)　DN およびターゲットエントリの属性値の 1 つを使用して、テンプレートエントリを特定する。

クラス定義 (class definition)　特定のオブジェクトのインスタンスを作成するために必要な情報を指定し、ディレクトリ内のほかのオブジェクトに関連してそのオブジェクトがどのように動作するのかを決定する。

コードページ (code page)　オペレーティングシステムが、キーボードのキーを画面に表示するための文字フォントと関連付けるときに使用する、国際化プラグインのコンテキストで、ロケールが使用する内部テーブル。

更新履歴ログ (change log)　レプリカに対する変更を記述した記録。サプライヤサーバは、コンシューマサーバに格納されているレプリカに対して、またはマルチマスターレプリケーションの場合はほかのマスターに対して、これらの変更をリプレイする。

国際化インデックス (international index)　多言語情報を含むディレクトリで、検索にかかる時間を短縮する。

国際標準化機構 (International Standards Organization)　「ISO」を参照。

コンシューマ (consumer)　サプライヤサーバからのレプリケートされたディレクトリツリーまたはサブツリーを含むサーバ。

コンシューマサーバ (consumer server)　レプリケーションのコンテキスト内で、ほかのサーバからコピーしたレプリカを保持するサーバは、そのレプリカのコンシューマと呼ばれる。

コンシューマ主導レプリケーション (consumer-initiated replication)　コンシューマ (consumer) サーバがサプライヤサーバからディレクトリのデータを引き出すレプリケーション構成。

コンシューマレプリカ (consumer replica)　すべての追加および修正処理に関してマスターレプリカを参照するレプリカ。サーバは任意の数のコンシューマレプリカを保持できる。

サーバサービス (server service)　実行されると、クライアントからの要求を待機し、受け入れる Windows NT 上のプロセス。Windows NT 上の SMB サーバがこれに当たる。

サーバセレクタ (Server Selector)　ユーザがブラウザを使用してサーバを選択および設定できるインタフェース。

サーバデーモン (server daemon)　実行されると、クライアントからの要求を待機し、受け入れるプロセス。

サーバルート (server root)　サーバのプログラム、設定、管理、および情報のファイルの保持専用の、サーバマシン上のディレクトリ。

サービスクラス (class of service)　「CoS」を参照。

最下位のエントリ (leaf entry)　その下にほかのエントリが 1 つもないエントリ。最下位のエントリは、ディレクトリツリーで分岐点になることはできない。

サブエージェント (subagent)　「SNMP サブエージェント (SNMP subagent)」を参照。

サブ接尾辞 (sub suffix)　ルート接尾辞の下の分岐。

サプライヤ (supplier)　コンシューマサーバにレプリケートされるディレクトリツリーあるいはサブツリーのマスターコピーを保持するサーバ。

サプライヤサーバ (supplier server)　レプリケーションのコンテキストで、別のサーバにコピーされるレプリカを保持するサーバは、そのレプリカのサプライヤと呼ばれる。

サプライヤ主導レプリケーション (supplier-initiated replication)　サプライヤ (supplier) サーバがコンシューマサーバにディレクトリのデータをレプリケートするレプリケーション構成。

サプライヤレプリカ (supplier replica)　ディレクトリ情報のマスターコピーを含む、更新可能なレプリカ。サーバは任意の数のマスターレプリカを保持できる。

参照整合性 (referential integrity)　関連するエントリ間の関係が、ディレクトリ内で管理されることを保証するメカニズム。

識別名 (distinguished name)　エントリの名前と LDAP ディレクトリ内での位置を文字列で表したもの。

自己アクセス (self access)　この権限が与えられた場合、バインド DN がアクセス先のエントリとマッチしたときに、ユーザが自分の所有するエントリにアクセスできることを示す。

時刻 / 日付の形式 (time / date format)　特定の地域における時刻および日付の習慣的な形式を示す。

システムインデックス (system index)　Directory Server の操作に必須なので削除および修正はできない。

実在インデックス (presence index)　特定のインデックス化された属性を含むエントリの検索を可能にする。

照合順序 (collation order)　ある言語の文字のソート方法について、言語および文化に固有の情報を提供する。この情報には、その文字体系における文字の順序、あるいはアクセント付きの文字とアクセントのない文字とを比較する方法などが含まれる。

証明書 (certificate)　ネットワークユーザの公開鍵を、ディレクトリ内にあるそれらの DN と関連付けるデータの集合。証明書は、ユーザオブジェクトの属性としてディレクトリ内部に格納される。

スーパーユーザ (superuser)　Unix マシン上で最も高いレベルの特権を持つユーザ。root とも呼ばれる。スーパーユーザは、マシン上のすべてのファイルに対して完全なアクセス権を持つ。

スキーマ (schema)　ディレクトリにどのようなタイプの情報をエントリとして格納できるかについての定義。スキーマとマッチしない情報がディレクトリに格納されている場合は、そのディレクトリにアクセスを試みているクライアントが正しい結果を表示できないことがある。

スキーマ検査 (schema checking)　ディレクトリ内で追加または変更されたエントリが、定義したスキーマに確実に従うことを確認する。スキーマ検査はデフォルトでオンになっている。したがって、スキーマに従っていないエントリを格納しようとした場合、エラーメッセージが表示される。

すべての ID のしきい値 (All IDs Threshold)　サーバが管理するすべてのインデックスキーに広域的に適用されるサイズ制限。個々の ID リストのサイズがこの制限値に達すると、サーバによってその ID リストがすべての ID のトークンと置き換えられる。

すべての ID のトークン (All IDs token)　すべてのディレクトリエントリがインデックスキーとマッチするサーバに想定させるメカニズム。実際には、すべての ID のトークンによって、サーバは検索要求で利用可能なインデックスが存在しないかのように動作する。

接尾辞 (suffix)　ディレクトリツリーの頂点にあるエントリの名前で、この下にデータが格納される。同じディレクトリ内に複数の接尾辞が存在できる。各データベースは接尾辞を 1 つだけ持つ。

操作属性 (operational attribute)　操作属性は、ディレクトリが修正およびサブツリーのプロパティを追跡するために内部で使用する情報を含む。明示的に要求しない限り、操作属性は検索の応答として返されることはない。

相対識別名 (relative distinguished name)　「RDN」を参照。

属性 (attribute)　エントリを説明する情報を保持する。属性にはラベルと値がある。また、各属性は、属性値として格納される情報のタイプに応じた標準の構文に従う。

属性リスト (attribute list)　特定のエントリタイプまたはオブジェクトクラスに対応する、必須の属性と省略可能な属性のリスト。

ターゲット (target)　アクセス制御において、ターゲットは特定の ACI が適用されるディレクトリ情報を識別する。

ターゲットエントリ (target entry)　CoS の適用範囲内のエントリ。

対称暗号化 (symmetric encryption)　暗号化と復号化の両方で同じキーを使用する暗号化。対称暗号化アルゴリズムの一例として DES が挙げられる。

単一マスターレプリケーション (single-master replication)　2 つのサーバがそれぞれ、コンシューマサーバに対して同じ読み書き可能レプリカのコピーを保持する最も基本的なレプリケーションモデル。単一マスターレプリケーションモデルでは、サプライヤサーバが更新履歴ログを管理する。

知識参照 (knowledge reference)　さまざまな データベースに格納されているディレクトリ情報へのポインタ。

通貨形式 (monetary format)　特定の地域で使用されている通貨記号や、通貨記号が数値の前と後ろのどちらに付くのか、および通貨単位の表記方法を指定する。

データベースリンク (database link)　連鎖を実装したもの。データベースリンクはデータベースのように動作するが、持続的な記憶領域を持たない。代わりに、リモートに格納されているデータを指し示す。

データマスター (data master)　特定データ部分のマスターソースであるサーバ。

デーモン (daemon)　特定のシステムタスクを担当する、Unix マシン上のバックグラウンドプロセス。デーモンプロセスは、動作の継続に人の介入を必要としない。

定義エントリ (definition entry)　「CoS 定義エントリ (CoS definition entry)」を参照。

ディレクトリサービス (directory service)　組織内の人材および資源に関する記述的な属性ベースの情報を管理するように設計されたデータベースアプリケーション。

ディレクトリツリー (directory tree)　ディレクトリに格納されている情報の論理表現。多くのファイルシステムで使用されているツリーモデルを模倣しており、ツリーのルート点が階層の頂点にある。DIT とも呼ばれる。

ディレクトリマネージャ (Directory Manager)　UNIX の root ユーザに相当する、特権を持ったデータベース管理者。ディレクトリマネージャにはアクセス制御が適用されない。

デフォルトインデックス (default index)　データベースインスタンスごとに作成されるデフォルトインデックスセットの 1 つ。デフォルトインデックスは変更できるが、デフォルトインデックスに依存しているプラグインもあるので、削除する場合は注意が必要。

テンプレートエントリ (template entry)　「CoS テンプレートエントリ (CoS template entry)」を参照。

等価インデックス (equality index)　特定の属性値を含むエントリを効果的に検索できる。

匿名アクセス (anonymous access)　この権限が与えられた場合、ユーザが持つ資格の有無およびバインドの条件とは無関係に、すべてのユーザがディレクトリ情報にアクセスできる。

トポロジ (topology)　ディレクトリツリーが複数の物理的なサーバに渡って、どのように分割されているのか、およびこれらのサーバがどのように相互にリンクをしているのかを示す。

名前の衝突 (name collisions)　同じ識別名を持った複数のエントリ。

認証 (authentication)　(1) クライアントユーザの ID を Directory Server に対して示すプロセス。ユーザがディレクトリへのアクセスを許可されるには、バインド DN および、対応するパスワードまたは証明書のどちらかを提示する必要がある。ディレクトリ管理者がユーザに許可したアクセス権に基づき、Directory Server はユーザに機能の実行やファイルおよびディレクトリへのアクセスを許可する。

(2) ほかのコンピュータがそのサーバであるかのように偽装したり、あるいはセキュリティ保護されていないコンピュータにもかかわらず保護されているように装うことを防ぎ、クライアント (client) がセキュリティ保護されたサーバに接続されていることを保証する。

認証局 (Certificate Authority)　認証証明書を販売および発行する会社または組織。ユーザは、信頼する認証局から認証証明書を購入できる。CA とも呼ばれる。

認証証明書　置き換えや偽造の不可能な、第三者が発行するデジタルファイル。認証証明書は、他方を検証し認証するために、サーバからクライアントへ、あるいはクライアントからサーバへ送信される。

認証ディレクトリサーバ (Authenticating Directory Server)　PTA (パススルー認証) における、要求元クライアントの認証資格を保持する Directory Server を指す。PTA が有効なホストは、クライアントから受信した PTA 要求をホストに送信する。

ネットワーク管理アプリケーション (authentication certificate)　稼動または停止しているデバイス、受信したエラーメッセージやその数など、SNMP 管理対象のデバイスに関する情報をグラフィックで表示する Network Management Station コンポーネント。

バインド DN (bind DN)　操作を実行するときに、Directory Server に対する認証で使用される識別名。

バインド規則 (bind rule)　アクセス制御に関して、ディレクトリ情報にアクセスするために特定のユーザまたはクライアントが満たす必要がある資格および条件を指定する。

バインド識別名 (bind distinguished name)　「バインド DN (bind DN)」を参照。

パススルーサブツリー (pass-through subtree)　パススルー認証では、PTA directory server は、このサブツリーに DN が含まれているすべてのクライアントからバインド要求を認証ディレクトリサーバ (Authenticating Directory Server) に渡す (パススルー)。

パススルー認証 (pass-through authentication)　「PTA」を参照。

パスワードファイル (password file)　Unix ユーザのログイン名、パスワード、およびユーザ ID 番号が格納されている Unix マシン上のファイル。格納場所から、`/etc/passwd` とも呼ばれる。

パスワードポリシー (password policy)　ディレクトリ内でのパスワードの使い方の基準となる規則のセット。

ハブサプライヤ (hub supplier)　レプリケーションのコンテキスト内で、ほかのサーバからコピーされたレプリカを保持するサーバのことで、このレプリカを第三のサーバにレプリケートする。「カスケード型レプリケーション」も参照。

汎用アクセス (general access)　この権限が与えられた場合、認証されたすべてのユーザがディレクトリの情報にアクセスできることを示す。

標準インデックス (standard index)　デフォルトで維持されるインデックス。

ファイル拡張子 (file extension)　通常、ファイルタイプを定義する、ファイル名のピリオドまたはドット (.) より後ろの部分。たとえば、.GIF、.HTML など。`index.html` というファイル名の場合、ファイル拡張子は `html` である。

ファイルタイプ (file type)　特定のファイルの形式。たとえば、グラフィックファイルは GIF 形式で格納される場合が多く、テキストファイルは通常 ASCII テキスト形式で格納される。ファイルタイプは、通常ファイル拡張子で識別される。たとえば、.GIF、.HTML など。

フィルタ (filter)　ディレクトリの照会に適用される制約で、返される情報を制限する。

フィルタを適用したロール (filtered role)　各エントリに含まれる属性に応じて、エントリをロールに割り当てることができるようにする。この操作を行うには、LDAP フィルタを指定する必要がある。フィルタにマッチするエントリは、そのロールを所有すると言われる。

部分文字列インデックス (substring index)　エントリ内の部分文字列の効率的な検索を可能にする。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る。

ブラウザ (browser)　HTML ファイルとして格納されている World Wide Web コンテンツを要求および表示する、Netscape Navigator などのソフトウェア。ブラウザは、ホストサーバとの通信に HTTP プロトコルを使用する。

ブラウズインデックス (browsing index)　仮想表示インデックスとも呼ばれる。Directory Server Console でエントリの表示を高速化する。ディレクトリの性能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作成可能。

プロキシ DN (proxy DN)　プロキシ認証で使用される。プロキシ DN とは、クライアントアプリケーションが操作を実行しようとしているターゲットへのアクセス権を持つエントリの DN。

プロキシ認証 (proxy authorization)　特殊な形式の認証で、ユーザは、自分の ID でバインドするがほかのユーザのアクセス権を使用してディレクトリへのアクセスが許可される。このほかのユーザとは、プロキシユーザであり、その DN はプロキシ DN である。

プロトコル (protocol)　ネットワーク上のデバイスが情報を交換する方法を記述した規則のセット。

分岐エントリ (branch entry)　ディレクトリ内でサブツリーの頂点を表すエントリ。

ベース DN (base DN)　ベース識別名。検索処理はベース DN に対して行われる。ベース DN とは、ディレクトリツリー内でエントリおよびその下にあるすべてのエントリの DN のこと。

ベース識別名 (base distinguished name)　「ベース DN (base DN)」を参照。

ポインタ CoS (pointer CoS)　ポインタ CoS は、テンプレート DN だけを使用してテンプレートエントリを識別する。

ホスト名 (hostname)　machine.domain.dom のような書式のマシン名で、IP アドレスに変換される。たとえば、`www.iPlanet.com` は、com ドメインの iPlanet サブドメインにある www マシンである。

マスターエージェント (master agent)　「SNMP マスターエージェント (SNMP master agent)」を参照。

マッチング規則 (matching rule)　検索処理中にサーバが文字列をどのように比較するかを定めるガイドライン。多言語検索では、サーバが使用する必要がある照合順序および演算子をマッチング規則で規定する。

マッピングツリー (mapping tree)　接尾辞 (サブツリー) の名前をデータベースと関連付けるデータ構造。

マルチプレクサ (multiplexor)　データベースリンクを含むサーバで、リモートサーバと通信する。

マルチマスターレプリケーション (multi-master replication) 2 つのサーバがそれぞれ、同じ読み書き可能レプリカのコピーを保持する高度なレプリケーションモデル。各サーバが、このレプリカの更新履歴ログを維持する。1 つのサーバに対する変更は、自動的にほかのサーバにもレプリケートされる。変更が競合した場合、タイムスタンプを使用してどちらのサーバが最新の変更を保持しているかを決定する。

文字タイプ (character type) 英字を、数字またはほかの文字と識別し、また大文字の小文字へのマッピングを識別する。

ルート接尾辞 (root suffix) 1 つ以上のサブ接尾辞の親。ディレクトリツリーは複数のルート接尾辞を含むことができる。

レプリカ (replica) レプリケーションに関与するデータベース。「コンシューマレプリカ (consumer replica)」および「サプライヤレプリカ (supplier replica)」も参照。

レプリケーション (replication) ディレクトリツリーまたはサブツリーをサプライヤサーバからコンシューマサーバにコピーする処理。

レプリケーションアグリーメント (replication agreement) サプライヤサーバに格納されている設定パラメタのセット。レプリケーション対象のデータベース、データをプッシュする先のコンシューマサーバ、レプリケーションを実行できる時間、コンシューマにバインドするためにサプライヤが使用する DN と資格、および接続をセキュリティ保護する方法を特定する。

レフェラル (referral) (1) サーバが自身では処理できない検索要求あるいは更新要求を LDAP クライアントから受信すると、サーバは通常、その要求を処理できる LDAP サーバへのポインタをクライアントに返信する。

(2) レプリケーションで、コンシューマレプリカが更新要求を受信すると、対応するマスターレプリカを保持するサーバにこの要求を転送する。この転送プロセスをレフェラルと呼ぶ。

連鎖 (chaining) 要求をほかのサーバに中継するための手法。要求の結果は収集、コンパイルされてから、クライアントに返される。

ロール (role) エントリをグループ化するメカニズム。各ロールは、そのロールを所有するエントリであるメンバーを持つ。

ロールに基づく属性 (role-based attributes) 関連付けられた CoS テンプレート内にエントリが特定のロールを所有しているため、エントリに記述される属性。

ロケール (locale) 住む地域や、文化、習慣の異なるユーザが、データを表すために使用するもので、照合順序、文字タイプ、通貨形式、時刻 / 日付の形式を識別する。ロケールには、特定言語のデータの解釈方法、格納方法、または照合方法に関する情報が含まれる。また、特定言語を表現するために使用するコードページを提供する。

索引

A

B

C

D

E

F

M

N

O

P

あ

い

え

お

か

き

く

け

こ

さ

し

す

せ

そ

た

ち

つ

て

と

に

ね

は

ひ

ふ

へ

ほ

ま

め

も

ゆ

よ

る

れ

ろ

わ