Red Hat Directory Server 10

管理ガイド

Directory Server 10.6 の更新

Last Updated: 2021-11-01

Red Hat Directory Server 10 管理ガイド

Directory Server 10.6 の更新

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

概要

本ガイドでは、Directory Server インスタンスおよびデータベースを管理する GUI およびコマンドラインの手順を説明します。 このドキュメントは維持されなくなりました。詳細は、 を参照してください。

目次

非推奨のドキュメント

重要

2020 年 11 月 30 日にて、Red Hat Directory Server 10 のサポート終了は終了しました。詳細は、「「Red Hat Directory Server Life Cycle policy」」を参照してください。Red Hat は、Directory Server 10 を最新バージョンに更新することを推奨します。

本製品のメンテナンスフェーズの終了により、本ドキュメントは更新されなくなりました。参照資料としてのみご使用ください。

第1章 RED HAT DIRECTORY SERVER の基本設定

Red Hat Directory Server にはディレクトリーサービス、複数のサーバーインスタンスを管理する管理サーバー、およびグラフィカルインターフェースを使用してサーバーインスタンスを管理するための Java ベースのコンソールが含まれています。本章では、ディレクトリーサービスを管理する基本的なタスクの概要を説明します。

Directory Server は、ユーザーおよびリソースのエンタープライズ全体のディレクトリーを管理するために設計された、堅牢でスケーラブルなサーバーです。LDAP(Lightweight Directory Access Protocol) と呼ばれるオープンシステムサーバープロトコルに基づいています。サーバーはディレクトリーデータベースを管理し、クライアント要求に応答します。

Directory Server は、連携する複数のコンポーネントで構成されています。

- **Directory Server** は、コア LDAP サーバーデーモンです。これは LDAP v3 標準に準拠しています。このコンポーネントには、データベースのエクスポートやバックアップなどの一般的な操作を行うためのコマンドラインサーバー管理プログラムおよびスクリプトが含まれます。
- **Directory Server コンソールは**、ユーザー、グループ、およびその他の LDAP データの管理を簡素化するユーザーインターフェースです。コンソールは、バックアップ、セキュリティー、レプリケーション、データベース設定、サーバーの監視、統計の表示など、サーバー管理のすべての側面に使用されます。
- **管理サーバーは、Directory Server** インスタンスを管理する管理エージェントです。Directory Server コンソールと通信し、Directory Server インスタンスで操作を実行します。また、簡単な HTML インターフェースおよびオンラインヘルプページも提供します。

コマンドラインユーティリティーを使用して Directory Server を管理できますが、Directory Server コンソールを使用することもできます。

1.1. システム要件

『『Red Hat Directory Server 10 リリースノート』の該当するセクションを参照してください』。

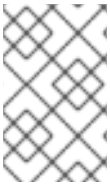
1.2. ファイルの場所

『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス の該当するセクションを参照してください』。

1.3. DIRECTORY SERVER 管理コンソールの起動

管理コンソールは、以下のような管理タスクを実行できるグラフィカルユーザーインターフェースを提供します。

- Directory Server インスタンスの管理
- 管理サーバーの管理
- ユーザーおよびグループの管理



注記

管理コンソールは Java を使用します。サポートされる Java ランタイム環境およびバージョンの詳細は、『『Red Hat Directory Server リリースノート』を参照してください』。

管理コンソールを開くには、次のコマンドを入力します。

```
# redhat-idm-console
```

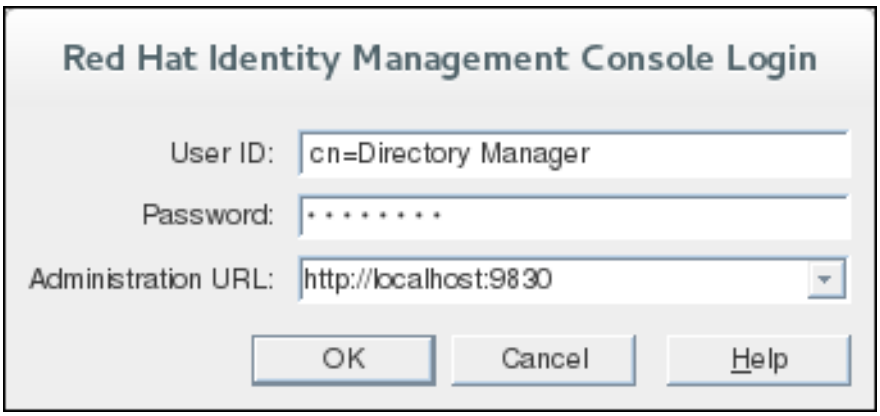
サポートされるコマンドラインオプションは、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス の該当するセクションを参照してください』。

1.3.1. Directory Server コンソールを開く

1. Directory Server 管理コンソールを起動します。

   ```
   # redhat-idm-console
   ```

2. **cn=Directory Manager** ユーザーとしてログインします。



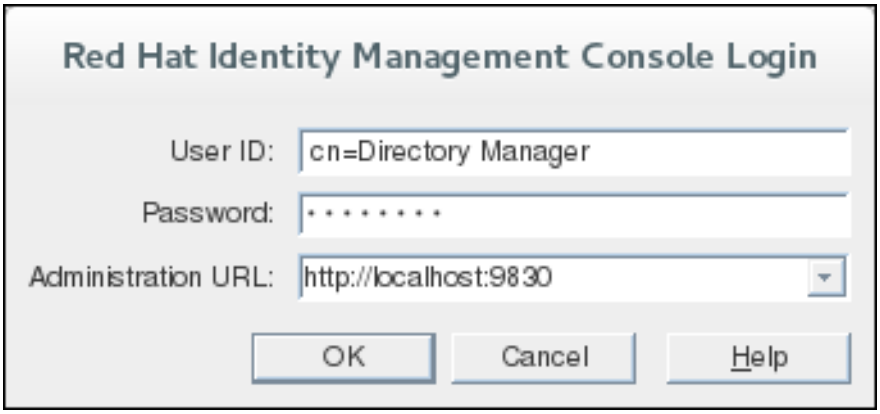
3. **Servers and Applications** タブで *administration_domain_namehost_name* → Server GroupDirectory Server(*instance_name*) に移動し、**Open** をクリックします。

1.3.2. 管理コンソールを開く

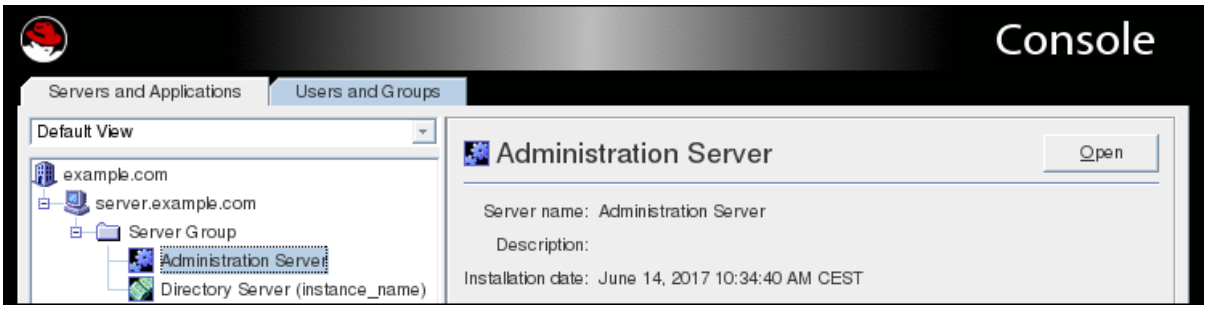
1. Directory Server 管理コンソールを起動します。

   ```
   # redhat-idm-console
   ```

2. **cn=Directory Manager** ユーザーとしてログインします。



3. **Servers and Applications** タブで *administration_domain_namehost_name* → Server Group → Administration Server に移動し、**Open** をクリックします。



1.4. DIRECTORY SERVER インスタンスの起動および停止

1.4.1. コマンドラインを使用した Directory Server インスタンスの起動および停止

systemctl ユーティリティーを使用して、インスタンスを起動、停止、または再起動します。

- インスタンスを起動するには、以下のコマンドを実行します。

```
# systemctl start dirsrv@instance_name
```

- インスタンスを停止するには、以下のコマンドを実行します。

```
# systemctl stop dirsrv@instance_name
```

- インスタンスを再起動するには、以下のコマンドを実行します。

```
# systemctl restart dirsrv@instance_name
```

必要に応じて、システムの起動時に Directory Server インスタンスが自動的に起動するようにすることができます。

- 単一のインスタンスの場合：

```
# systemctl enable dirsrv@instance_name
```

- サーバー上のすべてのインスタンスの場合：

```
# systemctl enable dirsrv.target
```

詳細は、『Red Hat システム管理者』 ガイドのシステムサービスの管理 セクションを参照してください。

1.4.2. コンソールを使用した Directory Server インスタンスの起動および停止

コマンドラインの横にある Directory Server コンソールを使用して、インスタンスの起動、停止、再起動を行うことができます。

重要

SELinux を **Enforcing** モードで実行する場合は、コンソールを使用してインスタンスの起動または停止を行うことはできません。この問題を回避するには、コマンドラインを使用してサービスを管理します。「Directory Server インスタンスの起動および停止」を参照してください。

重要

インスタンスの TLS 暗号化を有効にすると、インスタンスの起動時に Directory Server は TLS 証明書のパスワードを要求します。Directory Server コンソールでは、GUI にこのパスワードプロンプトを表示できません。この問題を回避するには、以下を実行します。

- コマンドラインを使用してサービスを管理します。「コマンドラインを使用した Directory Server インスタンスの起動および停止」 を参照してください。
- パスワードファイルを作成します。「Directory Server のパスワードファイルの作成」を参照してください。

Directory Server インスタンスを起動、停止、または再起動するには、以下を実行します。

1. Directory Server コンソールを起動し、**cn=Directory Manager** ユーザー名を使用してログインします。

   詳細は、「管理コンソールを開く」 を参照してください。

2. **Servers and Applications** タブで *administration_domain_namehost_name* → Server GroupDirectory Server(*instance_name*) に移動し、**Open** をクリックします。

3. **Tasks** タブで、実行するタスクをクリックします。

4. **Yes** をクリックして確定します。

タスクが終了すると、操作が成功したか、失敗した場合に、コンソールにメッセージが表示されます。

1.5. DIRECTORY SERVER 管理サーバーサービスの起動と停止

管理コンソールは、Directory Server コンソール（Directory Server を管理する GUI）を提供します。

1.5.1. コマンドラインを使用した管理サーバーサービスの起動と停止

systemctl ユーティリティーを使用して、管理サーバーサービスを起動、停止、または再起動します。

- サービスを起動するには、以下を実行します。

```
# systemctl start dirsrv-admin
```

- サービスを停止するには、以下を実行します。

```
# systemctl stop dirsrv-admin
```

- サービスを再起動するには、以下を実行します。

```
# systemctl restart dirsrv-admin
```

必要に応じて、システムの起動時に管理サーバーが自動的に起動するようにします。

```
# systemctl enable dirsrv-admin
```

詳細は、『Red Hat システム管理者』 ガイドのシステムサービスの管理 セクションを参照してください。

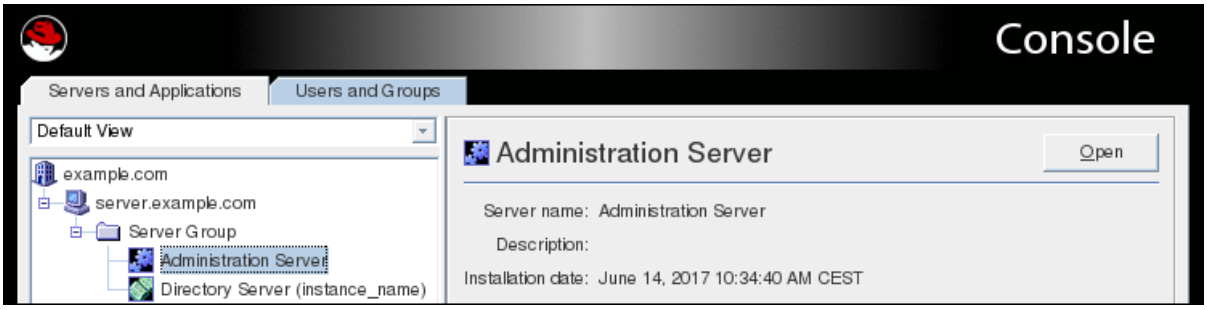
1.5.2. コンソールを使用した管理サーバーのサービスの再起動および停止

管理サーバーサービスを再起動または停止するには、以下を実行します。

1. Directory Server コンソールを起動し、**cn=Directory Manager** ユーザー名を使用してログインします。

   詳細は、「管理コンソールを開く」 を参照してください。

2. **Servers and Applications** タブで *administration_domain_namehost_name* → **Server Group** → **Administration Server** に移動し、**Open** をクリックします。



3. **Tasks** タブで、実行するタスクをクリックします。

4. **Yes** をクリックして確定します。

タスクが終了すると、操作が成功したか、失敗した場合に、コンソールにメッセージが表示されます。

1.6. LDAPI の有効化

IPC(**Inter-process communication**)は、Unix マシン上のプロセスや、ネットワークが互いに直接通信するための方法です。LDAPI により、LDAP 接続は IPC 接続で実行できます。つまり、LDAP 操作は Unix ソケット上で実行できます。これらの接続は、通常の LDAP 接続よりもはるかに高速で、より安全です。

LDAPI は、2 つの設定属性を使用して有効になります。

- ***nsslapd-ldapilisten*** Directory Server の LDAPI を有効にするには、以下を実行します。
- ***nsslapd-ldapifilepath*** Unix ソケットファイルを参照する

LDAPI を有効にするには、以下を実行します。

1. ***nsslapd-ldapilisten*** を変更して LDAPI をオンにし、ソケットファイル属性を追加します。

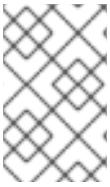
```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=config
changetype: modify
replace: nsslapd-ldapilisten
nsslapd-ldapilisten: on
-
add: nsslapd-ldapifilepath
nsslapd-ldapifilepath: /var/run/slapd-example.socket
```

2. サーバーを再起動して、新しい設定を適用します。

```
# systemctl restart dirsrv@instance
```

1.7. DIRECTORY SERVER ポート番号の変更

Directory Server が使用する標準およびセキュアな LDAP ポート番号は、Directory Server Console で変更するか、**dse.ldif** の **cn=config** エントリー下の ***nsslapd-port*** または ***nsslapd-secureport*** 属性の値を変更することで変更できます。

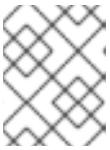


注記

o=NetscapeRoot サブツリーを維持する Configuration Directory Server の標準またはセキュアなポート番号を変更することは、Directory Server コンソールを介して行う必要があります。

1.7.1. 標準ポート番号の変更

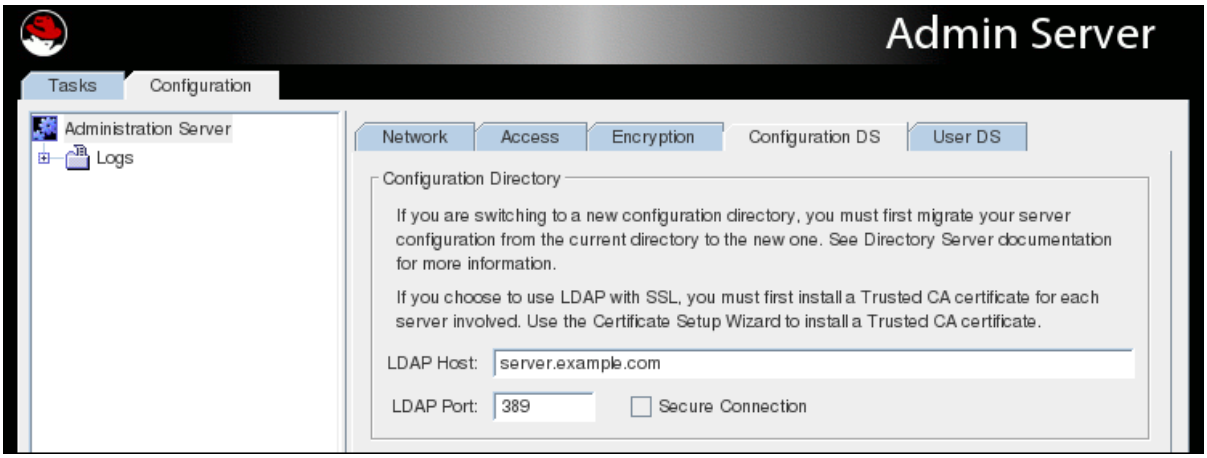
1. Directory Server コンソールの **Configuration** タブを選択し、左側のペインのナビゲーションツリーでトップエントリーを選択します。
2. 右側のペインで **Settings** タブを選択します。
3. ポート番号を変更します。**Port** フィールドで TLS 以外の通信に使用するサーバーのポート番号。デフォルト値は **389** です。
4. **Save** をクリックします。
5. コンソールは警告を返します。**設定ディレクトリーのポート番号を変更します。これは、このディレクトリーを使用するすべての管理サーバーに影響が及ぶため、新しいポート番号で更新する必要があります。ポート番号を変更してもよろしいですか？Yes** をクリックします。
6. その後、サーバーが再起動するまで変更が反映されないことを示すダイアログが表示されます。**OK** をクリックします。



注記

この時点で Directory Server を再起動しないでください。有効にすると、コンソールを介して管理コンソールに必要な変更を加えることはできません。

7. 管理コンソールを開きます。
8. **Configuration タブで Configuration DS** タブを選択します。



9. **LDAP ポート** フィールドで、Directory Server インスタンスの新しい LDAP ポート番号を入力します。

10. Directory Server ポートの SELinux ラベルを変更し、新しいポート番号を Directory Server ポリシーで使用できるようにします。以下に例を示します。

```
# semanage port -a -t ldap_port_t -p tcp 1389
```

警告

SELinux ラベルがリセットされない場合、Directory Server は再起動できなくなります。

11. Directory Server コンソールの **Tasks** タブで、**Restart Directory Server** をクリックします。サーバーを再起動することを確認するダイアログ。**Yes** をクリックします。

12. 管理コンソールの **Configuration DS** タブを開き、**Save** を選択します。

    ダイアログが表示され、読み取り **Directory Server 設定が変更されました。変更を有効にするには、管理サーバーとサーバーグループのすべてのサーバーをシャットダウンする必要があります。OK** をクリックします。

13. 管理コンソールの **Tasks** タブで、**Restart Admin Server** をクリックします。管理サーバーが正常に再起動したことを示すダイアログが開きます。**Close** をクリックします。

注記

コンソールで その他の操作を行う前に、コンソールを閉じて再度開く必要があります。更新してもコンソールを更新できず、何も実行しようとすると、**Unable to contact LDAP server という警告が表示されます。**

1.7.2. LDAPS ポート番号の変更

設定ディレクトリーまたはユーザーディレクトリーのポートを変更するか、またはポート番号の変更には以下のような要件があります。

- また、管理サーバー設定で Directory Server のポート番号も更新する必要があります。
- 設定またはユーザーディレクトリーを参照するその他の Directory Server インスタンスがある場合は、これらのサーバーを更新して新しいポート番号を指定します。

LDAPS ポートを変更するには、以下を実行します。

1. Directory Server インスタンスの証明書を発行するために使用する CA 証明書が Administration Server 証明書データベースにあることを確認します。Administration Server の CA 証明書のインポートは、「CA 証明書のインストール」 で説明されている Directory Server プロセスと同じです。

2. 「標準ポート番号の変更」 のプロセスと同様に、Directory Server Console を使用してセキュアなポートを設定できます（ **暗号化ポート** フィールドに値のみの設定のみ）。ただし、同じマ

シンに複数の Directory Server インスタンスがある場合など、Directory Server コンソールからポート番号を変更できない場合があります。**ldapmodify を使用してポート番号を変更** することが推奨されます。

以下に例を示します。

```
# ldapmodify -x -h server.example.com -p 1389 -D "cn=Directory Manager" -W
dn: cn=config
replace: nsslapd-securePort
nsslapd-securePort: 1636
```

3. Administration Server 設定(**o=netscaperoot**)で Directory Server インスタンスの対応するポート設定を編集します。

   まず、現在の設定を検索します。

```
# ldapsearch -x -h config-ds.example.com -p 389 -D "cn=Directory Manager" -W -b
"cn=slapd-ID,cn=389 Directory Server,cn=Server
Group,cn=server.example.com,ou=example.com,o=NetscapeRoot" -s base "(objectclass=*)"
nsSecureServerPort

dn: cn=slapd-ID,cn=389 Directory Server,cn=Server
Group,cn=server.example.com,ou=example.com,o=NetscapeRoot
nsSecureServerPort: 636
```

   次に、設定を編集します。

```
# ldapmodify -x -h config-ds.example.com -p 389 -D "cn=Directory Manager" -W

dn: cn=slapd-ID,cn=389 Directory Server,cn=Server
Group,cn=server.example.com,ou=example.com,o=NetscapeRoot
replace: nsSecureServerPort
nsSecureServerPort: 1636
```

4. インスタンスの Directory Server コンソールを起動し、新しい LDAPS ポート番号が **Configuration** タブに一覧表示されていることを確認します。

5. 必要に応じて、**Console で SSL を使用する** チェックボックスを選択します。

6. Directory Server ポートの SELinux ラベルを変更し、新しいポート番号を Directory Server ポリシーで使用できるようにします。以下に例を示します。

```
# semanage port -a -t ldap_port_t -p tcp 1636
```

警告

SELinux ラベルがリセットされない場合、Directory Server は再起動できなくなります。

7. Directory Server インスタンスを再起動します。

1.8. DIRECTORY SERVER インスタンスの管理

1.8.1. 新規 Directory Server インスタンスの作成

詳細は、『Red Hat Directory Server インストールガイド』の該当するセクションを参照してください。

- コマンドラインを使用した新規インスタンスの作成
- コンソールを使用した新規インスタンスの作成

1.8.2. Directory Server インスタンスの削除

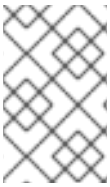
1.8.2.1. コマンドラインを使用した Directory Server インスタンスの削除

その他のインスタンスをすべてアンインストールしたり、管理サーバーインスタンスを削除したり、パッケージを削除したりせずに Directory Server の 1 つのインスタンスを削除できます。

```
# remove-ds.pl -i slapd-instance_name -a
```

remove-ds.pl スクリプトは、- **a** (all)オプションが指定されている場合は、関連するファイルおよびディレクトリーを削除します。ただし、Directory Server インスタンスは、設定 Directory Server から登録解除されません。

デフォルトでは、**鍵と証明書 ファイルは** インスタンス設定ディレクトリーに残り、設定ディレクトリーの名前が **slapd-instance-name.removed になります**。（以下に示すように） **-a** オプションを使用すると、セキュリティーデータベースも削除されます。



注記

インストールの失敗やサーバーを再起動するなど、Directory Server に問題が発生した場合には、**remove-ds.pl** スクリプトの実行に失敗します。この場合は、- **f** オプションを試して、強制的に削除プロセスを実行します。

1.8.2.1.1. Directory Server インスタンスおよび管理サーバーの削除

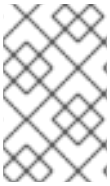
Directory Server と管理サーバーの両方を削除できます（同じシステムに設定されている場合）。

削除操作を実行するには、- **y** オプションが必要です。それ以外の場合は、**remove-ds-admin.pl** スクリプトはドライランを実行しますが、サーバーは削除されません。

-a オプションは必須ではありませんが、Directory Server または管理 Server インスタンスが後で再設定できる場合は推奨されます。デフォルトでは、すべてのセキュリティーデータベースは削除スクリプトで保持されます。**-a** オプションは、セキュリティーデータベースも削除します。

注記

サーバーにバインドするスクリプトには、Directory Server インスタンスが実行している必要があります。

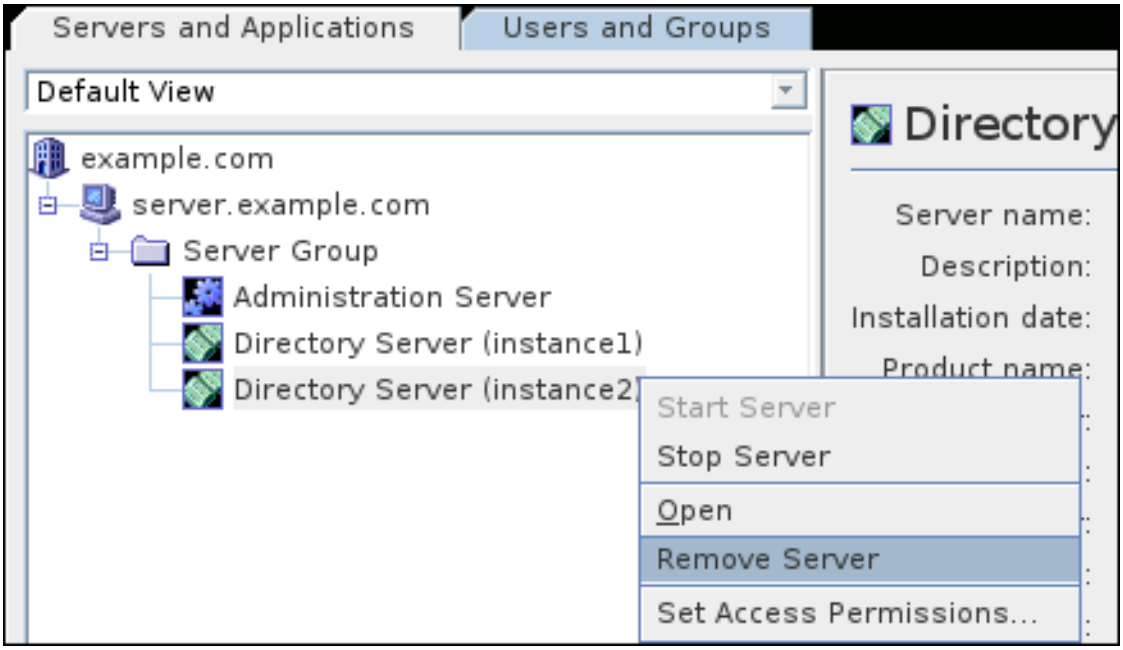


注記

インストールの失敗やサーバーを再起動するなど、Directory Server に問題が発生した場合には、**remove-ds-admin.pl** スクリプトの実行に失敗します。この場合は、- **f** オプションを試して、強制的に削除プロセスを実行します。

1.8.3. コンソールを使用した Directory Server インスタンスの削除

1. Directory Server コンソールを開きます。詳細は、「Directory Server コンソールを開く」 を参照してください。
2. サーバーインスタンスを右クリックし、**Remove Server** を選択します。



3. **Yes** をクリックして確定します。

1.9. DIRECTORY SERVER プラグインの使用

Directory Server には、レプリケーション、サービスのクラス、属性構文などのコア Directory Server 機能を設定するデフォルトプラグインが多数含まれています。コアプラグインはデフォルトで有効になり、完全に設定されます。

他のデフォルトプラグインは、一貫したユーザー定義で、DNA、属性の一意性、および属性リンクを提供することにより、Directory Server の機能を拡張します。これらのプラグインは利用可能ですが、すべてのプラグインがデフォルトで有効または設定される訳ではありません。

プラグインを使用すると、Directory Server を簡単に拡張できるため、お客様は独自のサーバープラグインを作成してデプロイし、特定のデプロイメントに必要なディレクトリー操作を実行することができます。

詳細は、以下を参照してください。

- 「Directory Server プラグインの使用」
- 『Red Hat Directory 『Server 設定、コマンド、およびファイルリファレンス のプラグイン実装サーバー機能リファレンス』 』セクション
- 『Red Hat Directory Server プラグインガイド』

1.9.1. プラグインを動的に有効化

Directory Server は、Directory Server を再起動せずに有効にできる動的プラグインをサポートします。動的な有効化されたプラグインを可能にすると、サーバーの管理がより容易になります。動的プラグインを使用すると、サーバーを複数回再起動して、プラグインをインストールおよび設定することができます。これにより、Directory Server のソフトウェアアプリケーションをデプロイする方がはるかに速くなります。

各プラグインは、***nsslapd-pluginEnabled*** 属性の値を切り替えることで有効または無効にできます。以下に例を示します。

```
# ldapmodify -x -D 'cn=Directory Manager' -W
dn: cn=Plug-in_name,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

cn=config エントリーで ***nsslapd-dynamic-plugins*** スイッチを指定した場合、プラグインを再設定するときに Directory Server を再起動する必要はありません。動的プラグイン機能を有効にするには、***nsslapd-dynamic-plugins*** 属性を **on** に設定します。

```
dn: cn=config
nsslapd-dynamic-plugins: on
```

動的プラグイン機能を無効にするには、***nsslapd-dynamic-plugins*** 属性を **off** に設定します。

```
dn: cn=config
nsslapd-dynamic-plugins: off
```

デフォルトでは、***nsslapd-dynamic-plugins*** は **off** に設定されます。

1.9.2. プラグインの有効化

1.9.2.1. コマンドラインでプラグインの有効化

コマンドラインでプラグインを無効化または有効にするには、**ldapmodify** ユーティリティーを使用して ***nsslapd-pluginEnabled*** 属性の値を編集します。以下に例を示します。

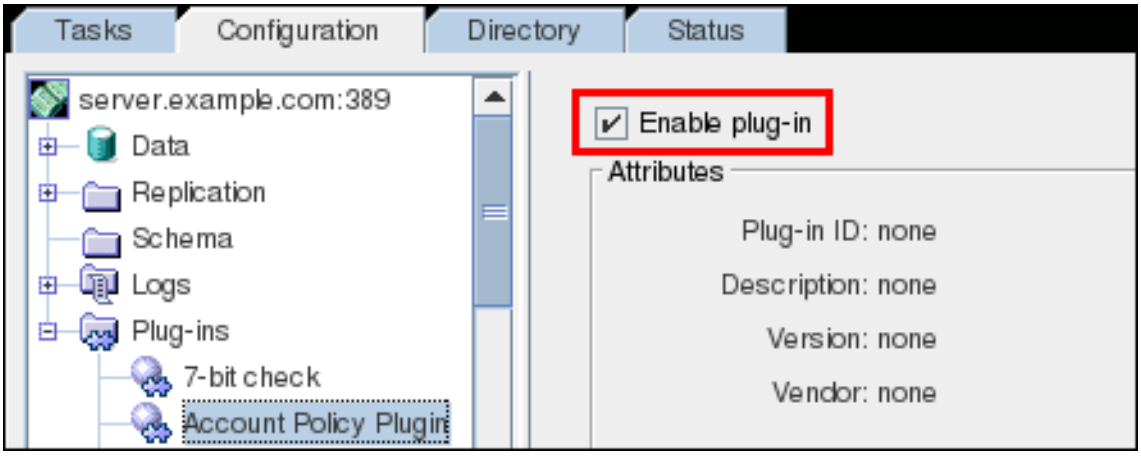
```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=ACL Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

1.9.2.2. Directory Server コンソールでプラグインの有効化

Directory Server コンソールを使用してプラグインを有効または無効にするには、以下を実行します。

1. Directory Server コンソールで、**Configuration** タブを選択します。
2. ナビゲーションツリーの **Plugins** フォルダーをダブルクリックします。

3. **Plugins** 一覧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 の **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. **Save** をクリックします。
6. Directory Server を再起動します。

```
# systemctl restart dirsrv@instance
```

注記

プラグインが無効になっていると、そのバージョンやベンダーなど、プラグインに関する詳細はすべて Directory Server コンソールに表示されません。すべての詳細フィールドには **NONE** が表示されます。

プラグインを有効にすると、Directory Server が再起動（新しいプラグイン設定のロード）後、Directory Server コンソールが更新されるまで、これらの詳細がコンソールに表示されません。

1.9.3. プラグインの設定

Directory Server 9 以前では、***nsslapd-pluginarg**** 属性を使用してプラグインを設定しました。Directory Server 10 は、特定のプラグインに特定の設定属性のサポートを追加しました。

重要

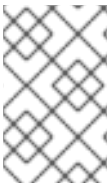
プラグイン固有の設定属性と非推奨の ***nsslapd-pluginarg**** 属性がプラグインの設定に設定されている場合、Directory Server はプラグイン固有の属性の設定のみを使用します。

以下の 2 つの例は、**Referential Integrity** プラグインで同じ設定を使用しますが、異なる設定オプションを使用します。

例1.1 設定属性を使用したプラグイン設定

```
referint-update-delay: 0
referint-logfile: /var/log/dirsrv/slapd-localhost/referint
referint-logchanges: 0
referint-membership-attr: member
```

```
referint-membership-attr: uniquemember
referint-membership-attr: owner
referint-membership-attr: seeAlso
```



注記

Red Hat は、設定プラグイン固有の属性のみを使用することを推奨します。プラグイン固有の属性については、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンスの該当するセクションを参照してください』。

例1.2 プラグイン引数属性を使用したプラグイン設定（非推奨）

```
nsslapd-pluginarg0: 0
nsslapd-pluginarg1: /var/log/dirsrv/slapd-localhost/referint
nsslapd-pluginarg2: 0
nsslapd-pluginarg3: member
nsslapd-pluginarg4: uniquemember
nsslapd-pluginarg5: owner
nsslapd-pluginarg6: seeAlso
```

1.9.3.1. コマンドラインでプラグインの設定

ldapmodify ユーティリティーを使用してプラグインを設定するには、以下を実行します。

1. プラグイン設定の識別名(DN)を特定します。詳細は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス の該当するセクションを参照してください』。
2. 新しい値を設定します。たとえば、**Referential Integrity** プラグインの更新遅延を **0 に設定するには、次のコマンドを実行します。**

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=referential integrity postoperation,cn=plugins,cn=config
changetype: modify
replace: referint-update-delay
referint-update-delay: 0
```

3. Directory Server インスタンスを再起動します。

```
# systemctl restart dirsrv@instance_name
```

1.9.3.2. コンソールを使用したプラグインの設定

Directory Server コンソールを使用してプラグインを設定するには、以下を実行します。

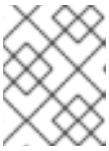
1. Directory Server コンソールを起動し、**cn=Directory Manager** ユーザー名を使用してログインします。

   詳細は、「管理コンソールを開く」 を参照してください。

2. **Servers and Applications** タブで *administration_domain_namehost_name* → Server GroupDirectory Server(*instance_name*) に移動し、**Open** をクリックします。

3. **プラグイン に移動し、** 設定するプラグインを選択します。

4. 右側のパネルで **Advanced** ボタンをクリックします。



注記

Red Hat は、プラグイン固有の属性を使用する **Property Editor** を使用してプラグインを設定することを推奨します。

5. プラグイン固有の属性を設定します。

6. **OK** をクリックして、**Property Editor** を閉じます。

7. Directory Server を再起動します。詳細は、「コンソールを使用した管理サーバーのサービスの再起動および停止」 を参照してください。

1.9.4. プラグインの優先順位の設定

プラグインの優先順位は、プラグインの実行順序にある優先順位です。操作前および操作後のプラグインでは、次のプラグインの開始前に 1 つのプラグインを実行して完了することができます。これにより、2 番目のプラグインが最初のプラグインの結果を活用できるようになります。

プラグインの設定エントリーの優先順位は、プラグインの設定エントリーの ***nsslapd-pluginPrecedence*** 属性で設定されます。この属性の値は、1（最も高い優先度）から 99（最も低い優先度）です。属性が設定されていない場合、デフォルト値は 50 になります。

重要

Red Hat サポートから指示されない限り、デフォルトの Directory Server プラグインにプラグインの優先順位を設定しないでください。プラグインの優先順位属性は、主にコア Directory Server **プラグイン** の動作を変更しない、カスタムプラグインの動作を管理することです。

nsslapd-pluginPrecedence 属性は **ldapmodify** コマンドを使用して設定されます。以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=My Example Plugin,cn=plugins,cn=config
```

```
changetype: modify
replace: nsslapd-pluginPrecedence
nsslapd-pluginPrecedence: 1
```

1.10. サーバー設定属性

Directory Server は、**cn=config** エントリーで維持される設定を **/etc/dirsrv/slapd-*instance_name*/dse.ldif** ファイルに保存します。新規インスタンスを設定すると、Directory Server はこのファイルで変更された設定属性のみを保存します。一覧にない属性には、デフォルト値を使用します。

これを使用すると、以下が可能になります。

- **/etc/dirsrv/slapd-*instance_name*/dse.ldif** ファイルを表示して、このインスタンスに設定したすべての設定パラメーターを特定します。
- パラメーターを削除してデフォルト値を復元します。

  設定パラメーターを削除すると、このパラメーターは **/etc/dirsrv/slapd-*instance_name*/dse.ldif** ファイルに一覧表示されなくなりました。ただし、LDAP プロトコルを使用して **cn=config** エントリーのパラメーターを検索すると、パラメーターとそのデフォルト値が表示されます。

  表1.1「削除できない設定属性」 に記載のパラメーターを削除して、デフォルトにリセットすることはできません。削除を試みると、サーバーは要求を拒否し、Server **is unwilling to perform(53)** エラーを発生させます。
- 新しい Directory Server バージョンで提供される最新のデフォルト値を使用します。

  多くの場合、新しいバージョンは最適化された設定を提供し、セキュリティーが強化されます。たとえば、***passwordStorageScheme*** 属性を設定しないと、Directory Server は、サポートされていて利用可能な、そして最も強力なパスワードストレージスキームを使用します。今後の更新で、セキュリティーを向上させるためにデフォルト値を変更すると、パスワードを設定する際に、新しいストレージスキームを使用してパスワードが自動的に暗号化されます。

注記

パラメーターを手動でデフォルトと同じ値に設定すると、値は更新されません。これは、新しいバージョンのデフォルト値が異なる場合に発生します。

表1.1 削除できない設定属性

nsslapd-accesslog	***nsslapd-auditlog***	***nsslapd-bakdir***
nsslapd-certdir	***nsslapd-certmap-basedn***	***nsslapd-conntablesize***
nsslapd-errorlog	***nsslapd-instancedir***	***nsslapd-ldifdir***
nsslapd-localhost	***nsslapd-localuser***	***nsslapd-lockdir***
nsslapd-rootpw	***nsslapd-referral***	***nsslapd-referralmode***

nsslapd-rundir	***nsslapd-saslpath***	***nsslapd-schemadir***
nsslapd-tmpdir	***nsslapd-workingdir***	

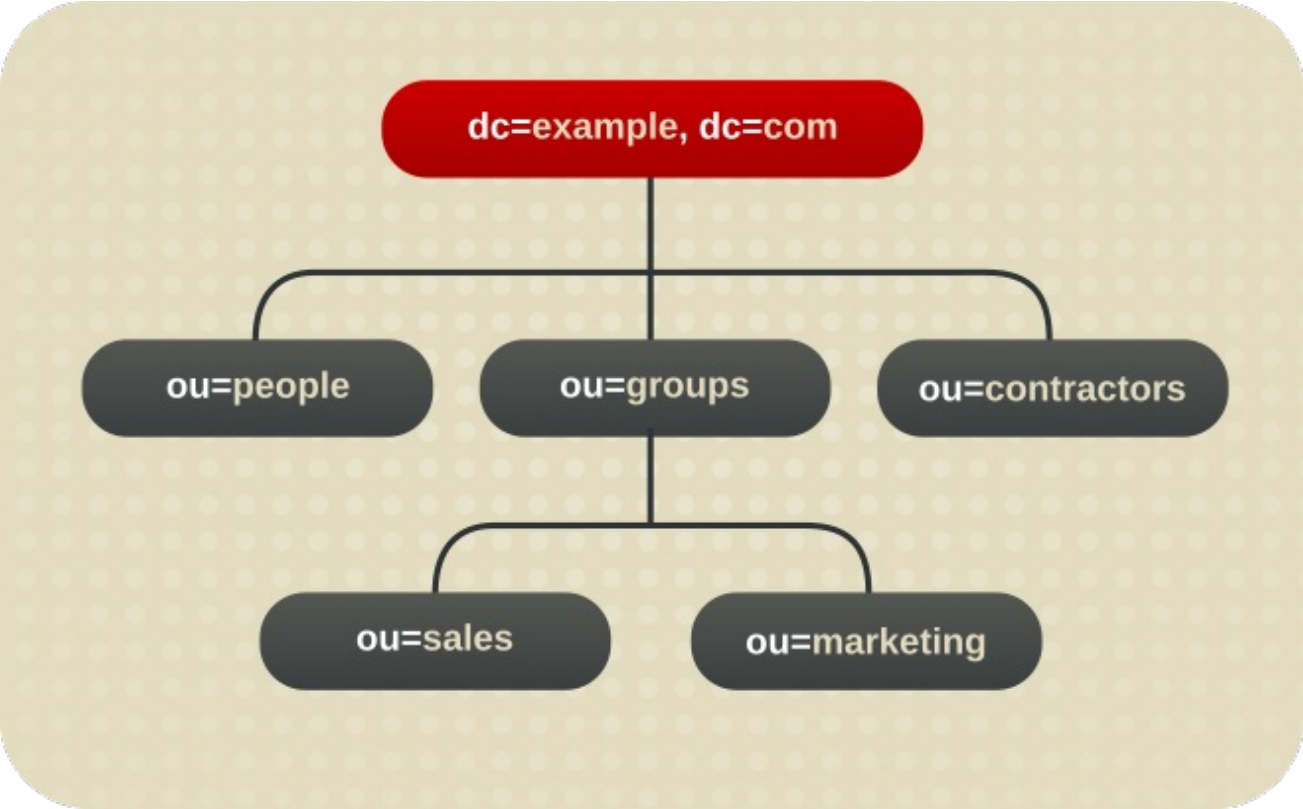
第2章 ディレクトリーデータベースの設定

ディレクトリーはデータベースに保存され、ディレクトリーツリーはデータベース全体に分散されます。本章では、**接尾辞** の作成、ディレクトリーツリーの分岐点、および各接尾辞に関連付けられたデータベースの作成方法を説明します。本章では、リモートサーバーでデータベースを参照するデータベースリンクを作成する方法と、参照を使用してクライアントにディレクトリーデータの外部ソースを指定する方法も説明します。

2.1. 接尾辞の作成および維持

ディレクトリーツリーのさまざまな部分をさまざまなデータベースに保存でき、そのデータベースを複数のサーバーに分散できます。ディレクトリーツリーには、**ノード** と呼ばれる分岐点が含まれます。このノードはデータベースに関連付けられている可能性があります。接尾辞は、特定のデータベースに関連するディレクトリーツリーのノードです。たとえば、簡単なディレクトリーツリーは、図2.1「1 つのルート接尾辞があるディレクトリーツリー」 のように表示されます。

図2.1 1つのルート接尾辞があるディレクトリーツリー



ou=people 接尾辞と、その下のすべてのエントリーおよびノードは 1 つのデータベースに保存され、**ou=groups** 接尾辞は別のデータベースに保存され、**ou=contractors** 接尾辞はまた別のデータベースに保存される可能性があります。

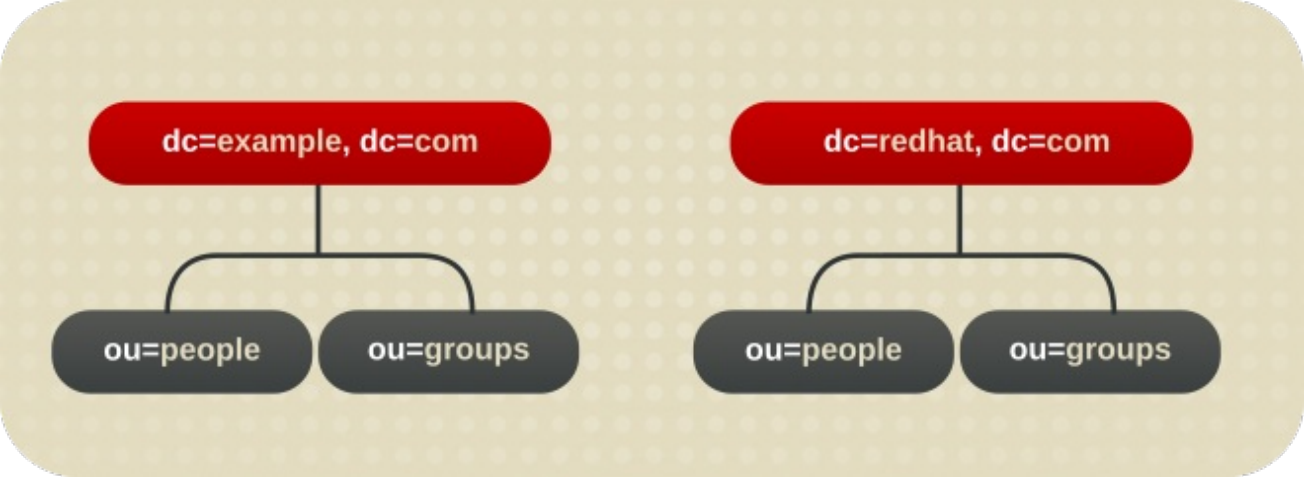
2.1.1. 接尾辞の作成

root 接尾辞は、 サブ接尾辞の親です。これは、Directory Server 用に設計された大規模なツリーの一部になります。**サブ接尾辞は**、root 接尾辞の下にあるブランチです。root 接尾辞とサブ接尾辞はどちらも、ディレクトリーツリーのコンテンツを整理するために使用されます。root 接尾辞およびサブ接尾辞のデータはデータベースに含まれます。

ディレクトリーには、複数のルート接尾辞が含まれる場合があります。たとえば、ISP は複数の Web サイト（example.com 用）と **redhat .com 用など**、複数の Web サイトをホストする場合があります。ここでは、2 つのルート接尾辞が必要です。1 つは **dc=example,dc=com** 命名コンテキストに対応

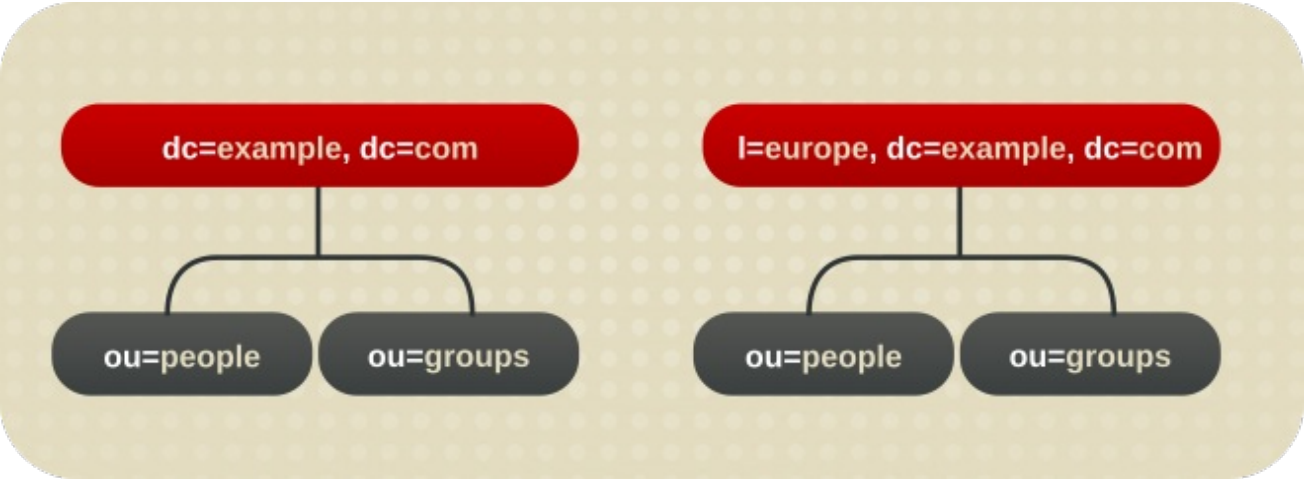
し、図2.2「2 つのルート接尾辞があるディレクトリーツリー」 のように、dc **=redhat,dc=com** 命名コンテキストに対応するものになります。

図2.2 2 つのルート接尾辞があるディレクトリーツリー



また、検索操作からディレクトリーツリーの一部を除外するために、ルート接尾辞を作成することもできます。たとえば、Example Corporation は、一般的な Example Corporation ディレクトリーの検索から、ヨーロッパのオフィスを除外します。これを実行するには、2 つのルート接尾辞を作成します。1 つのルート接尾辞は、一般的な Example Corporation ディレクトリーツリー **dc=example,dc=com** に対応します。また、1 つのルート接尾辞は、ディレクトリーツリーのヨーロッパブランチ **l=europe,dc=example,dc=com** に対応します。クライアントアプリケーションの観点では、ディレクトリーツリーは 図2.3「検索操作に対するルート接尾辞の Off 制限があるディレクトリーツリー」 で説明されています。

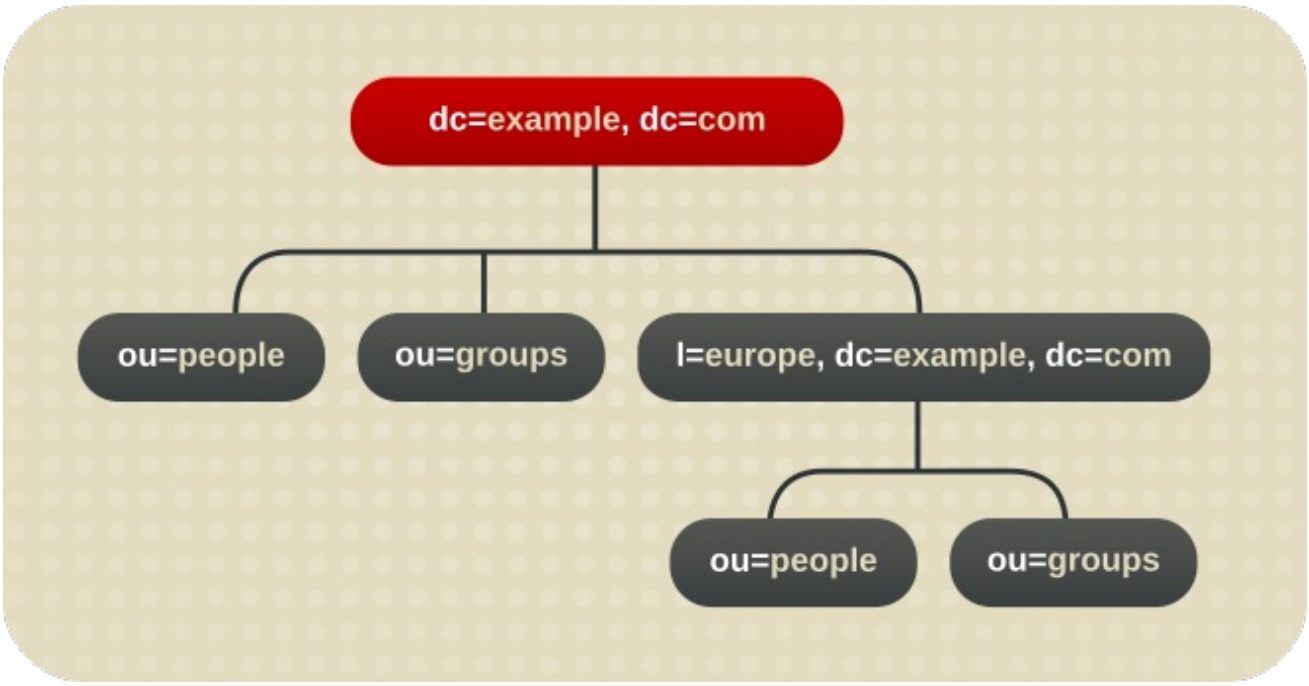
図2.3 検索操作に対するルート接尾辞の Off 制限があるディレクトリーツリー



クライアントアプリケーションがディレクトリーの **dc=example,dc=com** ブランチで検索を実行すると、ディレクトリーの **l=europe,dc=example,dc=com** ブランチは別のルート接尾辞となるため、エントリーを返しません。

一般的な検索で、ディレクトリーツリーのヨーロッパブランチにエントリーを含める場合は、ヨーロッパのブランチに、一般的なブランチのサブ接尾辞を指定します。これには、Example Corporation、dc **=example,dc=com** のルート接尾辞を作成し、ヨーロッパのディレクトリーエントリー **l=europe,dc=example,dc=com** の下にサブ接尾辞を作成します。クライアントアプリケーションの観点からは、ディレクトリーツリーは 図2.4「従属接尾辞が含まれるディレクトリーツリー」 に示されるように表示されます。

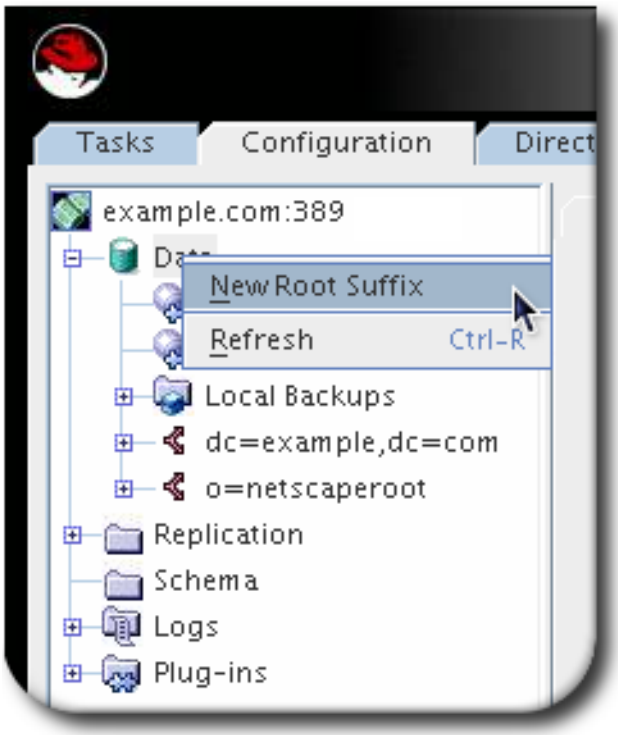
図2.4 従属接尾辞が含まれるディレクトリーツリー



本セクションでは、Directory Server Console またはコマンドラインのいずれかを使用して、ディレクトリーの root およびサブ接尾辞を作成する方法を説明します。

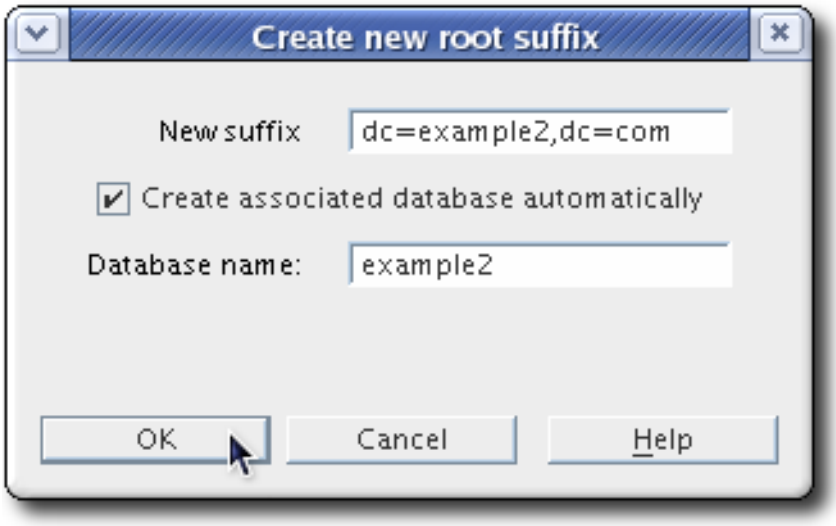
2.1.1.1. コンソールを使用した新規ルート接尾辞の作成

1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のナビゲーションペインで **Data** を右クリックし、ポップアップメニューから **New Root Suffix** を選択します。



3. **New suffix フィールドに一意の接尾辞** を入力します。

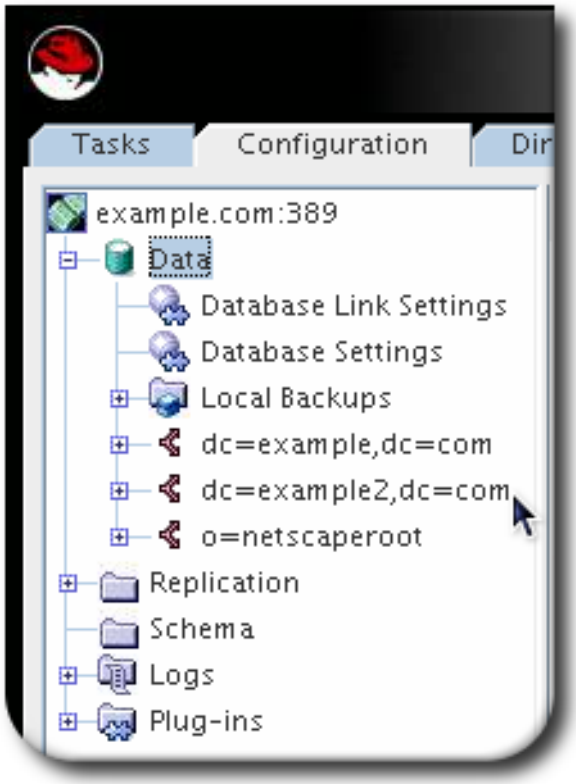
   接尾辞は、dc **=example,dc=com などの dc 命名規則を持つ行に指定する必要があります。**



4. **Create associated database automatically** を選択して、新しいルート接尾辞と同時にデータベースを作成し、**example2** などの **Database name** フィールドに新規データベースの一意の名前を入力します。名前は、英数字、ダッシュ(-)、およびアンダースコア(_)の組み合わせになります。他の文字は使用できません。

   チェックボックスの選択を解除して、後で新しいルート接尾辞のデータベースを作成します。このオプションは、データベースが作成されるディレクトリーを指定します。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ります。

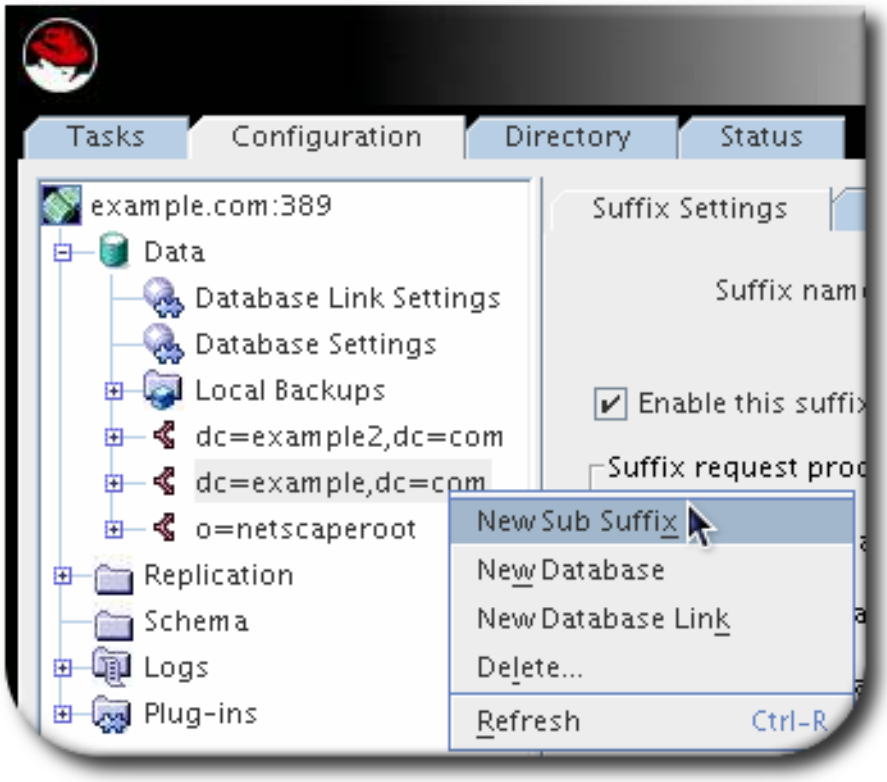
新しいルート接尾辞が **Data フォルダーに表示されます**。



2.1.1.2. コンソールを使用した新しい従属接尾辞の作成

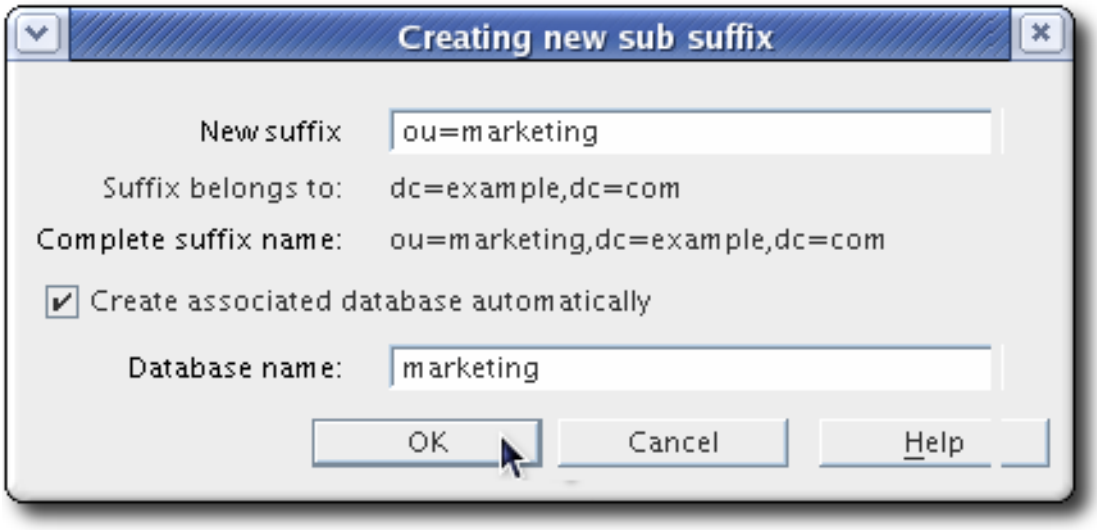
1. Directory Server コンソールで、**Configuration** タブを選択します。

2. 左側のナビゲーションペインの **Data** で、新しいサブ接尾辞を追加する接尾辞を選択します。接尾辞を右クリックし、ポップアップメニューから **New Sub Suffix** を選択します。



Create new sub suffix ダイアログボックスが表示されます。

3. **New suffix** 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 **の命名規則**（例： **ou=groups）を持つ行に指定する必要があります。**

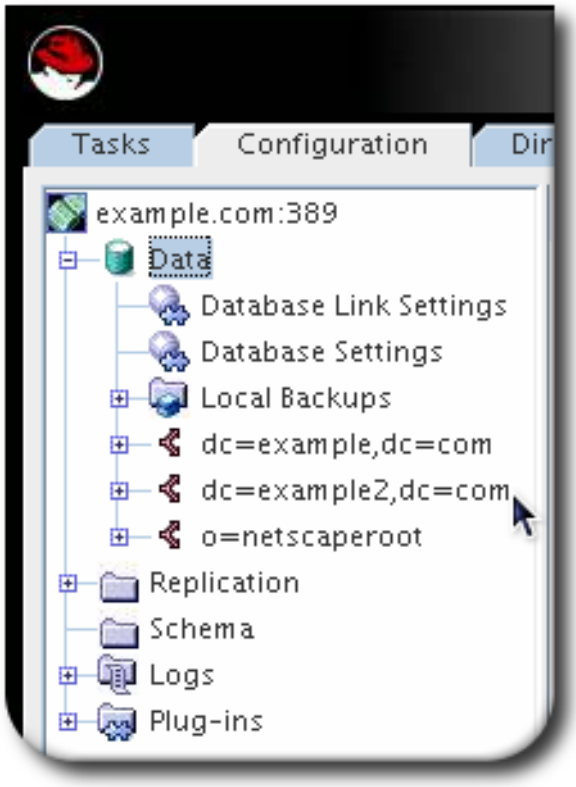


root 接尾辞は名前に自動的に追加されます。たとえば、サブ接尾辞 **ou=groups** が **dc=example,dc=com** 接尾辞の下に作成される場合、コンソールには **ou=groups,dc=example,dc=com という名前が自動的に付けられます。**

4. **Create associated database automatically** チェックボックスを選択し、新しいサブ接尾辞と同時にデータベースを作成し、**example2** などの **Database name** フィールドに新規データベースの一意の名前を入力します。名前は、英数字、ダッシュ(-)、およびアンダースコア**(_)**の組み合わせになります。他の文字は使用できません。

   このチェックボックスを選択しないと、新しいサブ接尾辞のデータベースよりも後で作成する必要があります。データベースが作成されるまで、新しいサブ接尾辞が無効になります。

接尾辞は、左側のナビゲーションペインの **Data** ツリーのルート接尾辞の下に自動的に表示されます。



2.1.1.3. コマンドラインでのルート接尾辞およびサブ接尾辞の作成

接尾辞の設定情報は **cn=mapping tree,cn=config** エントリーに保存されます。**ldapmodify** ユーティリティーを使用して、新しい接尾辞をディレクトリーに追加します。

接尾辞の作成時に設定できるすべてのパラメーターの一覧は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンスの該当するセクションを参照してください』。

ルート接尾辞の作成
たとえば、dc **=example,dc=com** のルート接尾辞を追加するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

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

従属接尾辞の作成
サブ接尾辞の作成は、root 接尾辞の作成と同様です。違いは、***nsslapd-parent-suffix*** に親接尾辞を設定する点です。

たとえば、dc= **example,dc =com ルート接尾辞の下に ou= groups** サブ接尾辞を作成するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
```

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

2.1.2. 接尾辞の維持

2.1.2.1. デフォルトの命名コンテキストの表示

命名コンテキストは接尾辞に類似しており、命名ディレクトリーエントリーのルート構造です。ディレクトリーとデータ構造によっては、複数の命名コンテキストが存在する場合があります。たとえば、標準の Directory Server 設定には、dc **=example,dc=com**、**cn=config** の設定接尾辞である **o=netscaperoot** などのユーザー接尾辞があります。

多くのディレクトリーツリーには複数の命名コンテキストがあり、異なるタイプのエントリーや論理データ分割で使用されます。Directory Server にアクセスするクライアントは、使用する必要がある命名コンテキストを認識しない場合があります。Directory Server には、デフォルトの命名コンテキストが他に認識されていない場合に、デフォルトの命名コンテキストがクライアントに通知するサーバー設定属性があります。

デフォルトの命名コンテキストは、**cn=config** の ***nsslapd-defaultnamingcontext*** 属性に設定されます。この値はルート DSE (Directory Server Agent Service Entry) に伝播され、ルート DSE の ***defaultnamingcontext*** 属性を確認してクライアントが匿名でクエリーできます。

```
# ldapsearch -p 389 -h server.example.com -x -b "" -s base | egrep namingcontext
namingContexts: dc=example,dc=com
namingContexts: dc=example,dc=net
namingContexts: dc=redhat,dc=com
defaultnamingcontext: dc=example,dc=com
```

重要

設定の整合性を維持するには、***nsslapd-allowed-to-delete-attrs*** 一覧から ***nsslapd-defaultnamingcontext*** 属性を削除しないでください。

デフォルトでは、***nsslapd-defaultnamingcontext*** 属性は、***nsslapd-allowed-to-delete-attrs*** 属性に削除 **できる** 属性の一覧に含まれます。これにより、現在のデフォルトの接尾辞を削除してから、適切にサーバー設定を更新できます。

何らかの理由で削除可能な設定属性の一覧から ***nsslapd-defaultnamingcontext*** 属性を削除すると、その属性への変更は保持されません。デフォルトの接尾辞を削除すると、その変更はサーバー設定に伝播できません。つまり、***nsslapd-defaultnamingcontext*** 属性は、空白 (削除) ではなく古い情報を保持することを意味します。これは正しい現在の設定です。

2.1.2.2. 接尾辞の無効化

特定の状況では、ディレクトリーの接尾辞を無効にする必要があります。接尾辞が無効になっていると、その接尾辞に関連するデータベースのコンテンツは、クライアントがアクセスできなくなります。

2.1.2.2.1. コマンドラインでの接尾辞の無効化

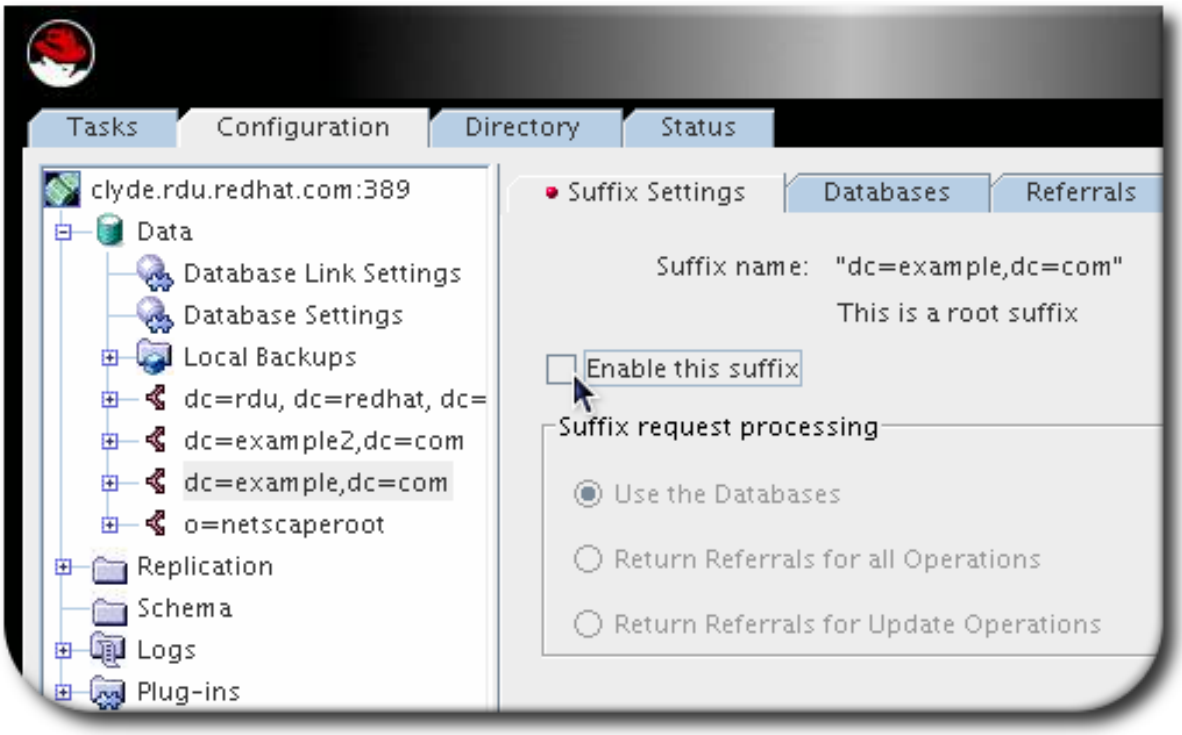
コマンドラインで接尾辞を無効にするには、対応する接尾辞エントリーの ***nsslapd-state*** 属性を **disabled** に設定します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=suffix_DN,cn=mapping tree,cn=config
changetype: modify
replace: nsslapd-state
nsslapd-state: disabled
```

2.1.2.2.2. コンソールを使用した接尾辞の無効化

コンソールを使用して接尾辞を無効にするには、以下を実行します。

1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のナビゲーションペインで **Data** で、接尾辞をクリックして無効にします。
3. **Suffix Setting** タブをクリックし、**Enable this suffix** チェックボックスの選択を解除します。



2.1.2.3. 接尾辞の削除

接尾辞が不要になった場合は、その接尾辞をデータベースから削除します。

警告

接尾辞を削除すると、その接尾辞に関連するデータベースエントリーおよびレプリケーション情報もすべて削除されます。

2.1.2.3.1. コマンドラインを使用した接尾辞の削除

コマンドラインで接尾辞を削除するには、以下を実行します。

1. マッピングツリーから接尾辞を削除します。

```
# ldapdelete -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
"cn="suffix_DN",cn=mapping tree,cn=config"
```

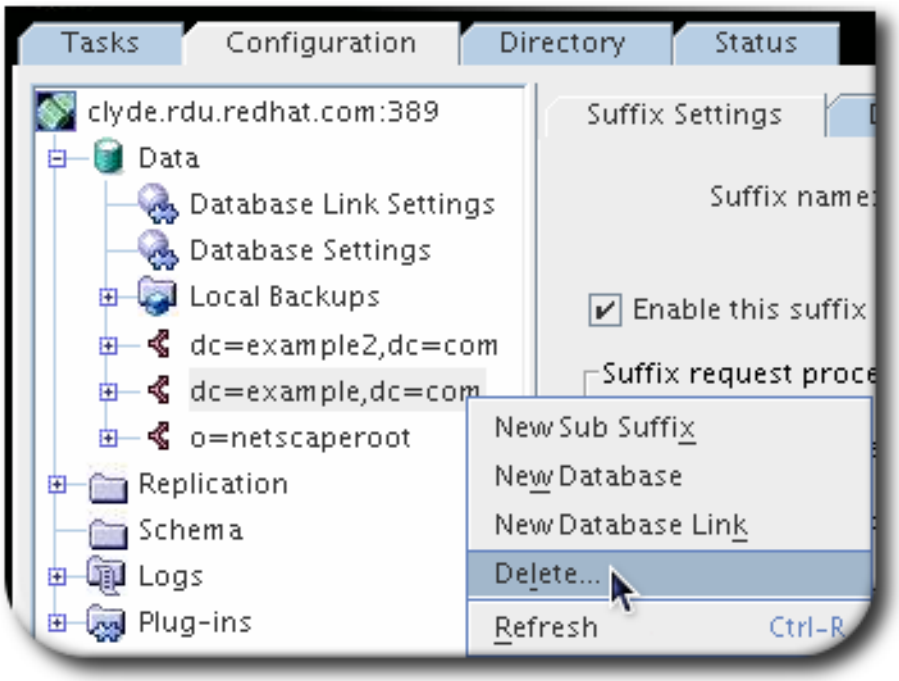
2. 接尾辞が別のデータベースを使用する場合は、データベースを削除します。

```
# ldapdelete -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
"cn=database_name,cn=ldbm database,cn=plugins,cn=config"
```

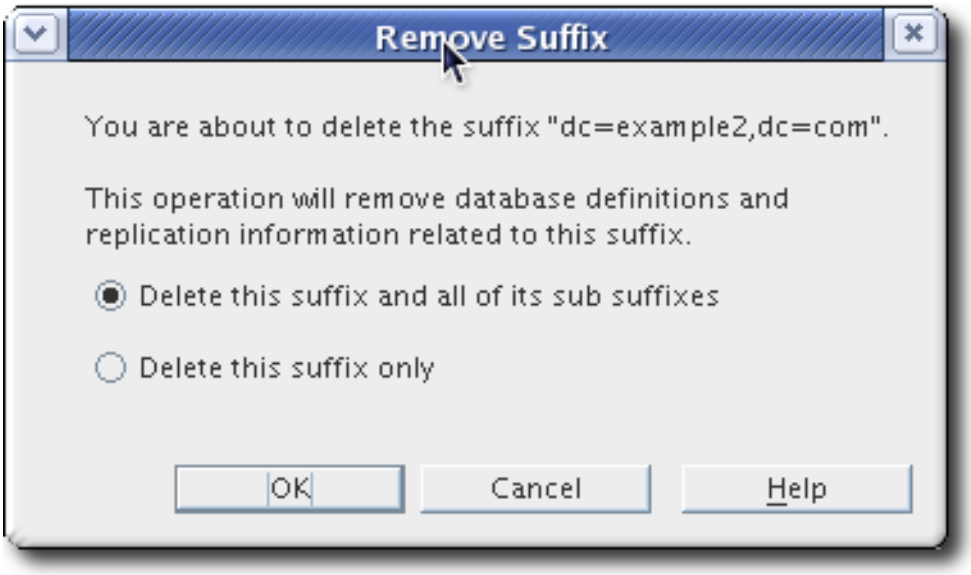
2.1.2.3.2. コンソールを使用した接尾辞の削除

コンソールを使用して接尾辞を削除するには、以下を行います。

1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のナビゲーションペインで **Data** の下で、削除する接尾辞を選択します。
3. 接尾辞を右クリックし、メニューから **Delete** を選択します。



4. **Delete this suffix およびそのサブ接尾辞のすべてを選択 するか 、この接尾辞のみを削除します。**



2.2. データベースの作成および維持

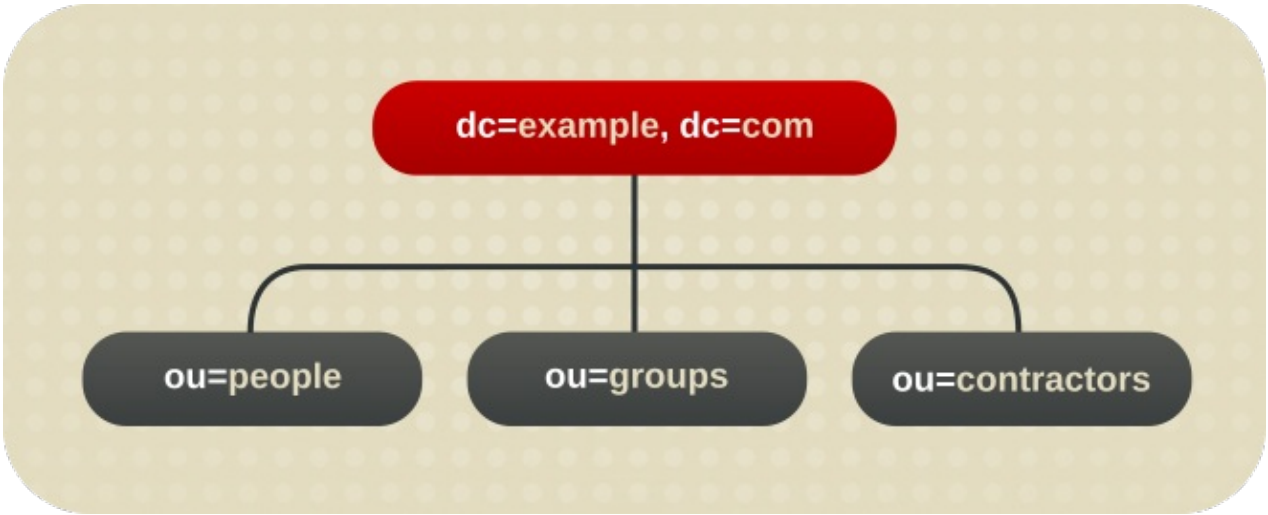
ディレクトリーデータを整理するための接尾辞を作成したら、そのディレクトリーのデータを含むデータベースを作成します。

2.2.1. データベースの作成

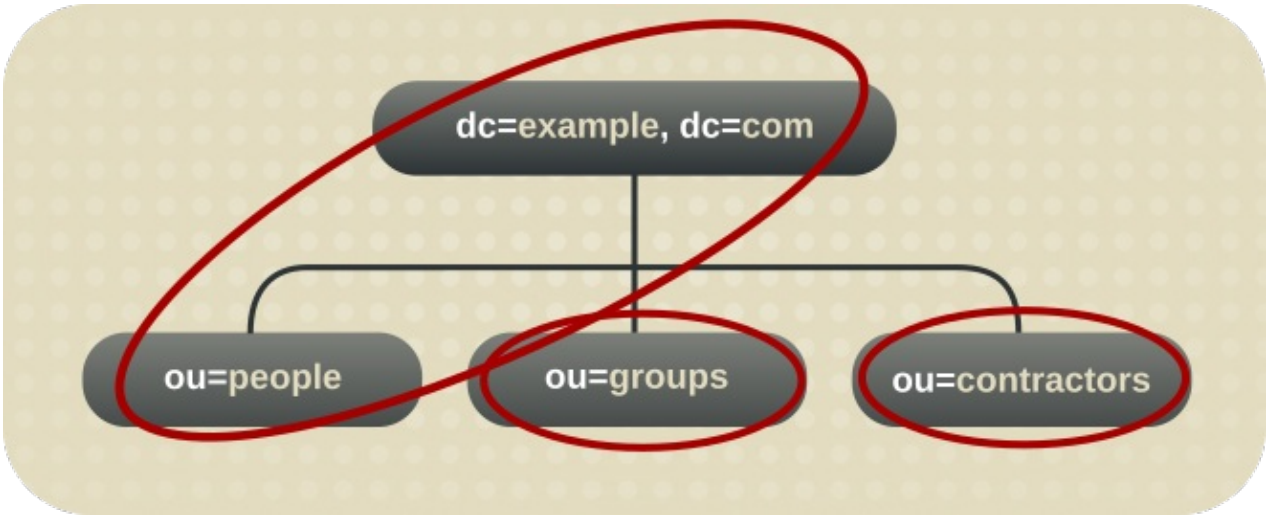
ディレクトリーツリーは、複数の Directory Server データベースに配布できます。複数のデータベースにデータを分散する方法は 2 つあります。

各接尾辞に 1 つのデータベース各接尾辞のデータは個別のデータベースに含まれます。

個別の接尾辞に含まれるデータを格納するために、3 つのデータベースが追加されます。



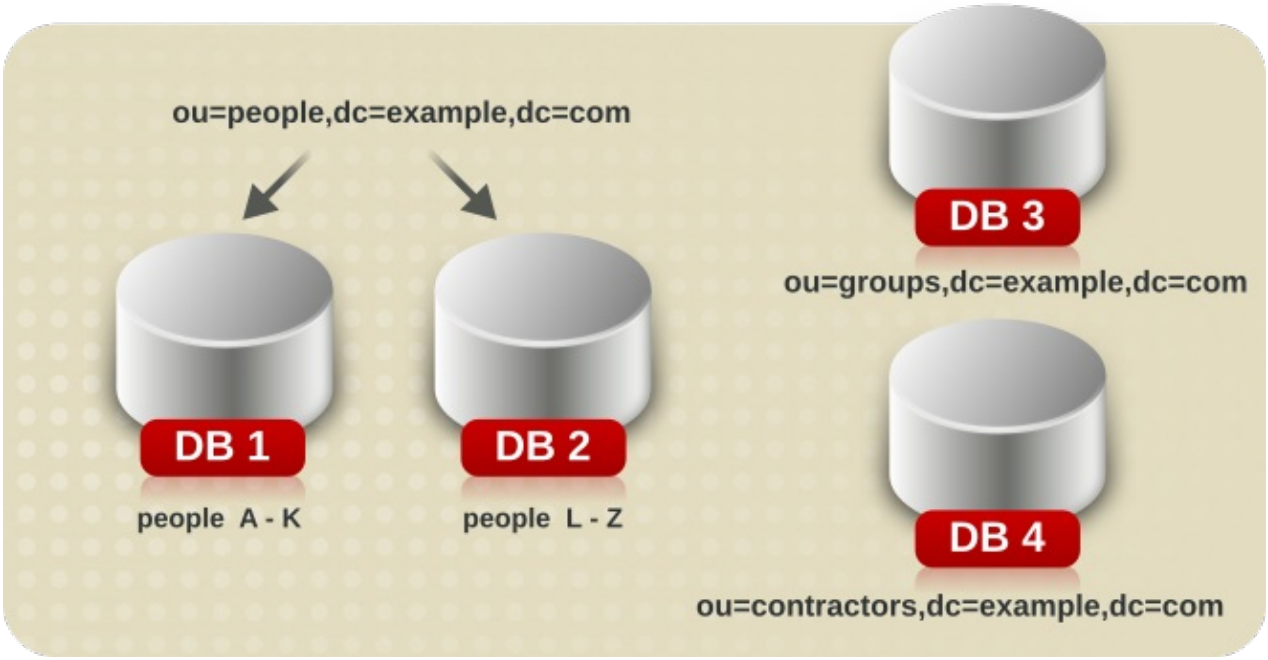
このツリーユニットの分割は、たとえば次の 3 つのデータベースに対応しています。



この例では、DB1 には **ou=people** のデータおよび **dc=example,dc=com** のデータが含まれ、クライアントが **dc=example,dc=com** に基づいて検索を実行できるようにします。ただし、DB2 には **ou=groups** のデータのみが含まれ、DB3 には **ou=contractors** のデータのみが含まれます。

1つの接尾辞に複数のデータベースがあります。

ディレクトリーツリーの **ou=people** ブランチ内のエントリー数が非常に大きくなると、2 つのデータベースを格納しなければならないとします。この場合、**ou=people** に含まれているデータは 2 つのデータベースに分散できます。

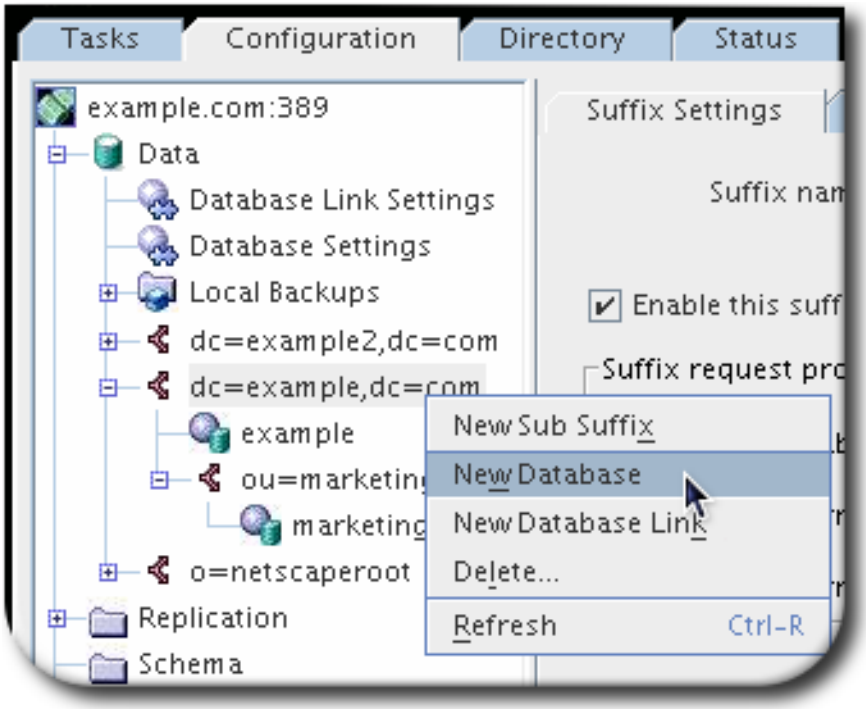


DB1 には **A-K** からの名前の人が含まれ、DB2 には **L-Z** からの名前が含まれます。DB3 には **ou=groups** のデータが含まれ、DB4 には **ou=contractors** のデータが含まれます。

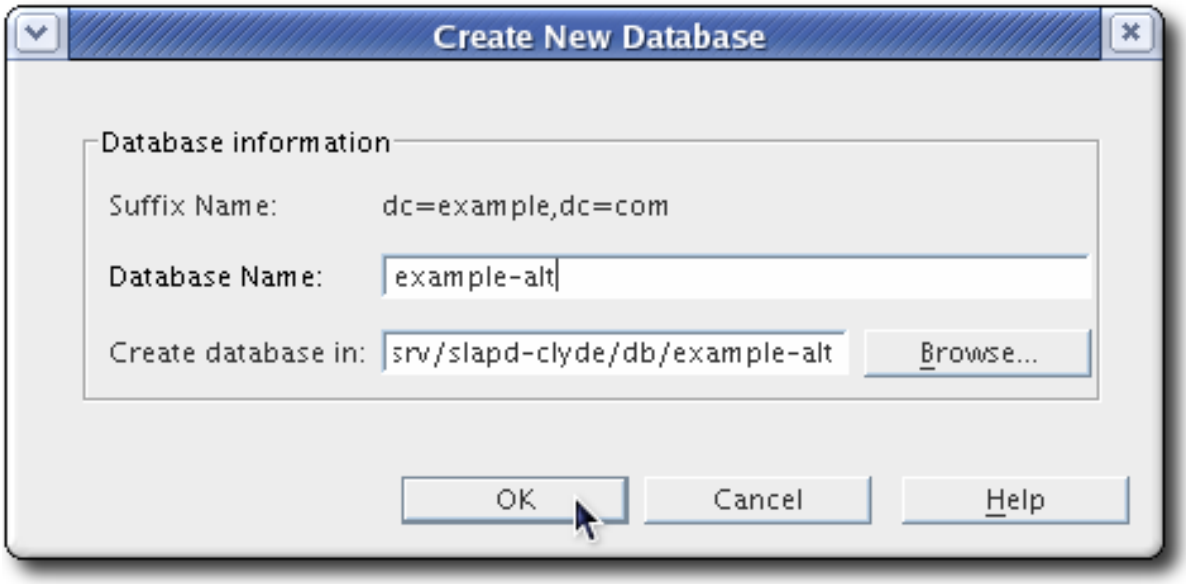
カスタムプラグインは、複数のデータベースにまたがってデータを単一の接尾辞から分散します。Directory Server のディストリビューションロジックの作成方法は、Red Hat コンサルティングにお問い合わせください。

2.2.1.1. コンソールを使用した既存の接尾辞の新規データベースの作成

1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のペインで **Data** を展開し、新しいデータベースを追加する接尾辞をクリックします。
3. 接尾辞を右クリックし、ポップアップメニューから **New Database** を選択します。



4. **example2** などのデータベースの一意の名前を入力します。データベース名は、英数字、ダッシュ(**-**)、およびアンダースコア(_)の組み合わせになります。



Create database in フィールドには、デフォルトのデータベースディレクトリー (**/var/lib/dirsrv/slapd-*instance*/db**)と新規データベースの名前が自動的に入力されます。また、別のディレクトリーの場所を入力またはブラウズすることもできます。

2.2.1.2. コマンドラインから単一の接尾辞用の新規データベースの作成

ldapmodify コマンドラインユーティリティーを使用して、ディレクトリー設定ファイルに新しいデータベースを追加します。データベース設定情報は **cn=ldbm database,cn=plugins,cn=config** エントリーに保存されます。たとえば、新しいデータベースをサーバー **example1** に追加します。

1. **ldapmodify** を実行して、新規データベースのエントリーを作成します。

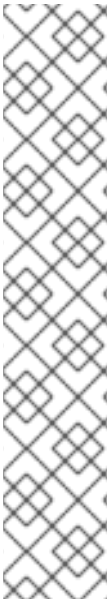
```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

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

追加されたエントリーは、root またはサブ接尾辞 **ou=people,dc=example,dc=com** のデータが含まれる **UserData** という名前のデータベースに対応します。

2. 「コマンドラインでのルート接尾辞およびサブ接尾辞の作成」 の説明に従って、ルートまたは従属接尾辞を作成します。DN 属性で指定されるデータベース名は、接尾辞エントリーの ***nsslapd-backend*** 属性の値に対応している必要があります。

2.2.1.3. 単一の接尾辞に複数のデータベースの追加

1 つの接尾辞は、複数のデータベースに分散できます。ただし、接尾辞を配布するには、ディレクトリーを拡張するためにカスタムディストリビューション機能を作成する必要があります。カスタムディストリビューション機能の作成に関する詳細は、Red Hat コンサルティングにお問い合わせください。



注記

エントリーが分散されたら、再分散できません。以下の制限が適用されます。

- ディストリビューション機能は、エントリーディストリビューションのデプロイ後は変更できません。
- エントリーを異なるデータベースに分散させる可能性がある場合は、LDAP **modrdn** 操作を使用してエントリーの名前を変更することができません。
- 分散ローカルデータベースは複製できません。
- エントリーを異なるデータベースに分散させる可能性がある場合は、**ldapmodify** 操作を使用してエントリーを変更することができません。

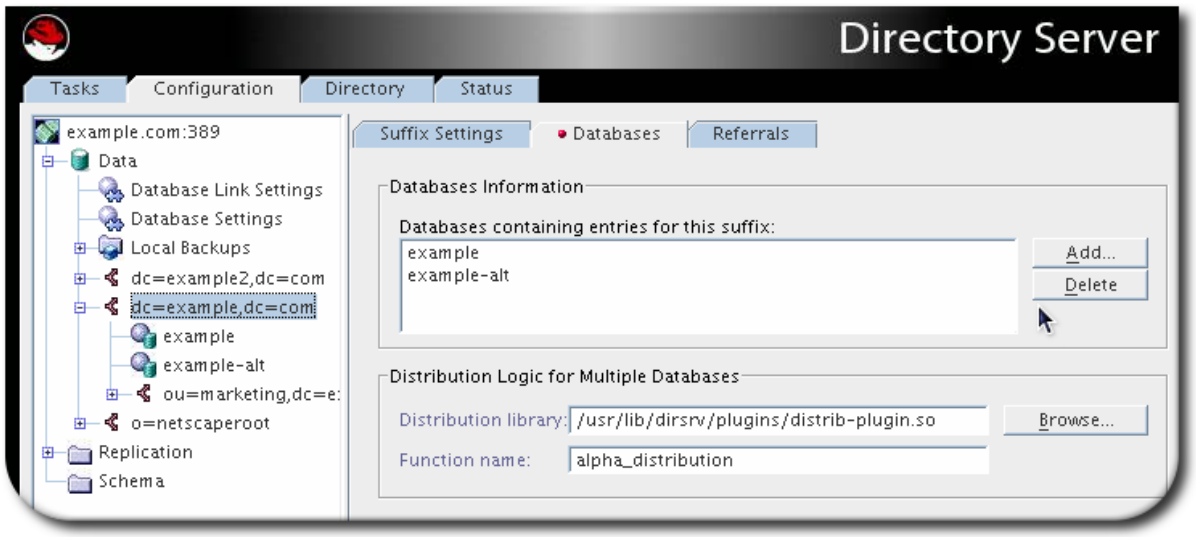
これらの制限に違反すると、Directory Server はエントリーを正しく特定して返さないようにします。

カスタムディストリビューションロジックプラグインを作成したら、そのプラグインをディレクトリーに追加します。

ディストリビューションロジックは、接尾辞で宣言された関数です。この関数は、この接尾辞に到達するすべての操作に対して呼び出されます。これには、接尾辞の前に開始するサブツリー検索操作が含まれます。ディストリビューション機能は、コンソールとコマンドラインインターフェースの両方を使用して接尾辞に挿入できます。

2.2.1.3.1. Directory Server コンソールを使用したカスタムディストリビューション機能の接尾辞への追加

1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のナビゲーションペインで **Data** を展開します。ディストリビューション機能を適用する接尾辞を選択します。
3. 右側のウィンドウで **Databases** タブを選択します。



4. 接尾辞に関連付けられているデータベースは、**Databases** タブにすでにリストされています。**Add** をクリックして、追加のデータベースを接尾辞に関連付けます。
5. ディストリビューションライブラリーへのパスを入力します。

6. **Function name** フィールドにディストリビューション機能の名前を入力します。

2.2.1.3.2. コマンドラインを使用したカスタムディストリビューション機能の接尾辞への追加

1. **ldapmodify** を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
```

2. 以下の属性を接尾辞エントリー自体に追加し、カスタムディストリビューションロジックに関する情報を提供します。

```
dn: suffix
changetype: modify
add: nsslapd-backend
nsslapd-backend: Database1
-
add: nsslapd-backend
nsslapd-backend: Database2
-
add: nsslapd-backend
nsslapd-backend: Database3
-
add: nsslapd-distribution-plugin
nsslapd-distribution-plugin: /full/name/of/a/shared/library
-
add: nsslapd-distribution-funct
nsslapd-distribution-funct: distribution-function-name
```

nsslapd-backend 属性は、この接尾辞に関連付けられたすべてのデータベースを指定します。***nsslapd-distribution-plugin*** 属性は、プラグインが使用するライブラリーの名前を指定します。***nsslapd-distribution-funct*** 属性は、ディストリビューション機能自体の名前を提供します。

ldapmodify コマンドラインユーティリティーの使用に関する詳細は、「コマンドラインでエントリーの管理」 を参照してください。

2.2.2. Directory データベースの維持

- 「読み取り専用モードでのデータベースの配置」
- 「データベースの削除」
- 「トランザクションログディレクトリーの変更」

2.2.2.1. 読み取り専用モードでのデータベースの配置

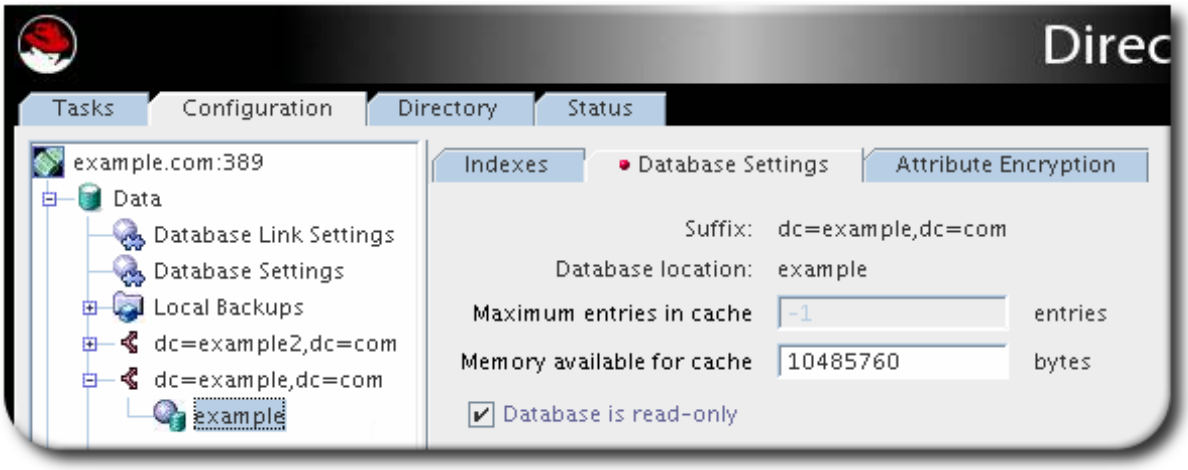
データベースが読み取り専用モードの場合は、エントリーを作成、変更、または削除することはできません。読み取り専用モードが役立つ状況の 1 つは、コンシューマーを手動で初期化する場合や、Directory Server からデータをバックアップまたはエクスポートする前です。読み取り専用モードは、特定の時点でのこれらのデータベースの状態の正確なイメージを保証します。

Directory Server コンソールおよびコマンドラインユーティリティーは、エクスポート操作またはバックアップ操作の前にディレクトリーを読み取り専用モードに自動的に配置しません。これは、ディレクトリーの更新で利用できなくなるためです。ただし、マルチマスターレプリケーションでは、これは問題ではない可能性があります。

- 「コンソールを使用したデータベースの読み取り専用の設定」
- 「コマンドラインからデータベースの読み取り専用の設定」
- 「読み取り専用モードでの Directory Server の配置」

2.2.2.1.1. コンソールを使用したデータベースの読み取り専用の設定

1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のペインで **Data** を展開します。データベースが含まれる接尾辞を展開して、読み取り専用モードにします。
3. 読み取り専用モードに設定するデータベースを選択します。
4. 右側のペインで、**Database Settings** タブを選択します。



5. **データベースが読み取り専用** チェックボックスにチェックマークを入れます。

この変更は即座に有効になります。

データベースをインポートまたは復元する前に、操作の影響を受けるデータベースが読み取り専用モード **ではないことを確認してください。**

読み取り専用モードを無効にするには、Directory Server Console でデータベースを再度開き、**データベースが読み取り専用** チェックボックスの選択を解除します。

2.2.2.1.2. コマンドラインからデータベースの読み取り専用の設定

データベースを読み取り専用モードに手動で配置するには、以下を実行します。

1. **ldapmodify** を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
```

2. read-only 属性を **on に変更します。**

```
dn: cn=database_name,cn=ldbm database,cn=plugins,cn=config
changetype: modify
replace: nsslapd-readonly
nsslapd-readonly: on
```

注記

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の名前は **userRoot** です。

2.2.2.1.3. 読み取り専用モードでの Directory Server の配置

Directory Server が複数のデータベースを維持し、すべてのデータベースを読み取り専用モードで配置する必要がある場合は、1 回の操作で実行できます。

警告

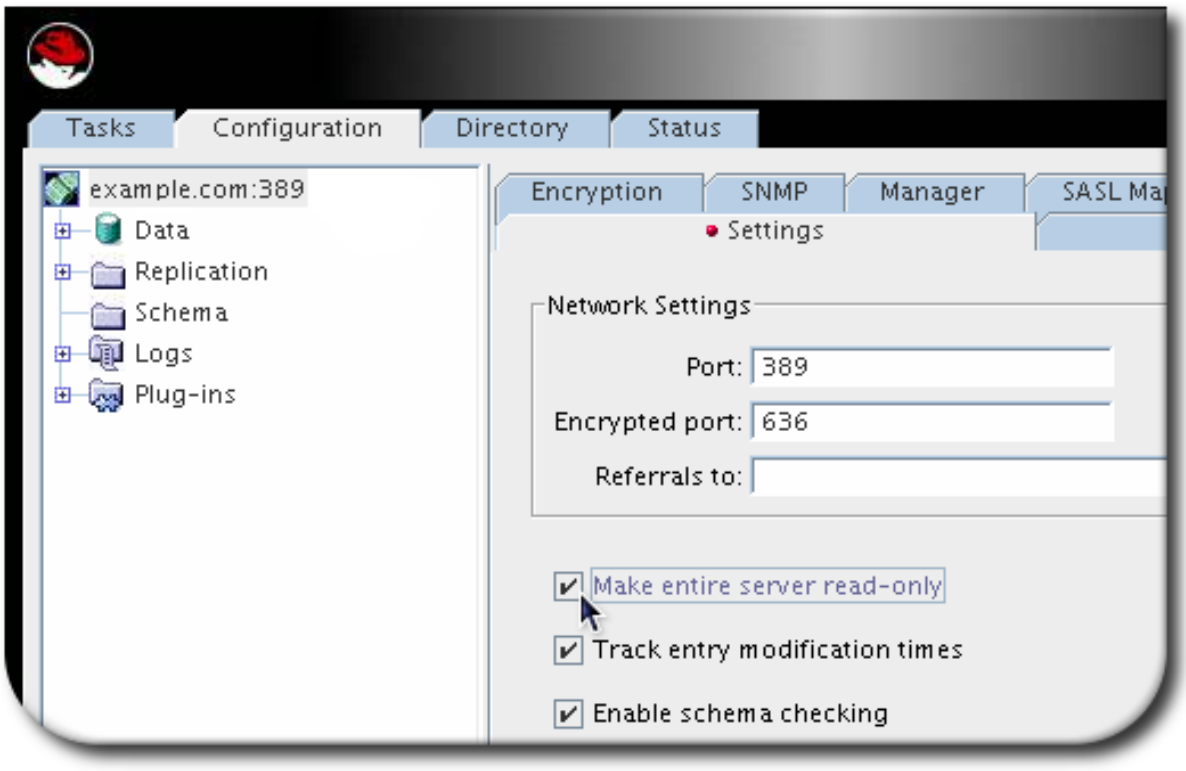
この操作は、Directory Server 設定が読み取り専用であるため、サーバー設定の更新、プラグインの有効化または無効化、読み取り専用モードの場合は Directory Server を再起動することはできません。読み取り専用モードを有効にすると、コンソールから元に戻す **ことはできません**。設定ファイルを変更する必要があります。

注記

Directory Server にレプリカが含まれている場合は、レプリケーションを無効にするため、読み取り専用モードを使用 **しないでください**。

Directory Server を読み取り専用モードにするには、以下を実行します。

1. Directory Server コンソールの **Configuration** タブを選択し、左側のペインのナビゲーションツリーでトップエントリーを選択します。
2. 右側のペインで **Settings** タブを選択します。

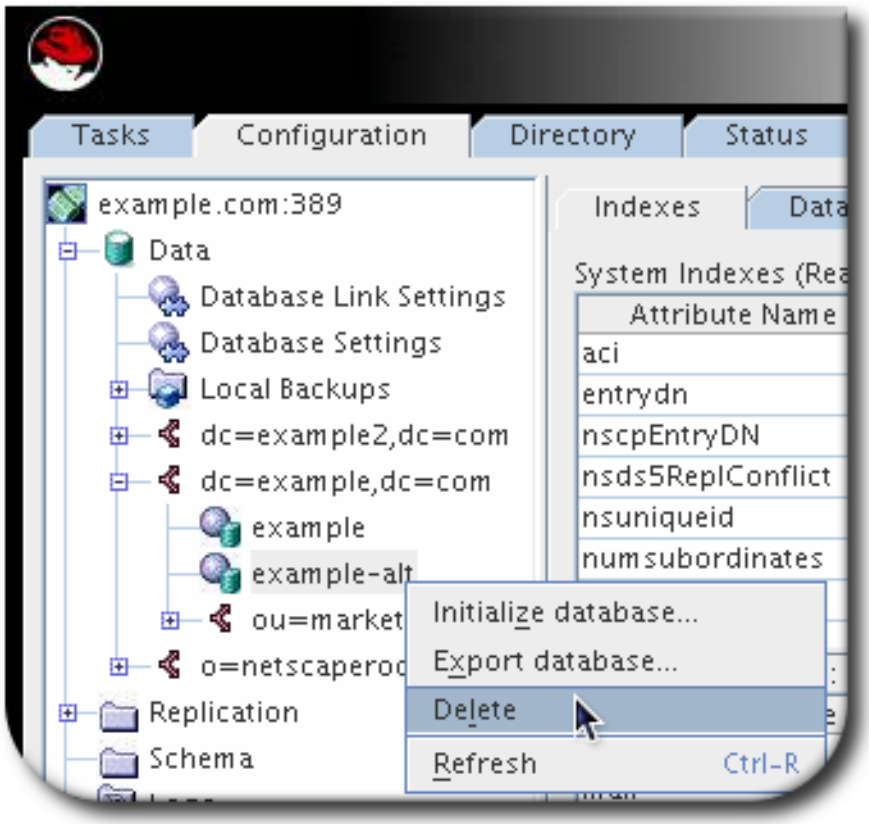


3. **Make Entire Server Read-Only** チェックボックスを選択します。
4. **Save** をクリックし、サーバーを再起動します。

2.2.2.2. データベースの削除

データベースを削除すると、そのデータベースの設定情報とエントリーのみが削除され、物理データベース自体は削除されません。

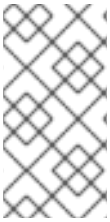
1. Directory Server コンソールで、**Configuration** タブを選択します。
2. **Data** フォルダーを展開し、接尾辞を選択します。
3. 削除するデータベースを選択します。
4. データベースを右クリックし、ポップアップメニューから **Delete** を選択します。



5. **Delete Database** ダイアログボックスでデータベースを削除する必要があることを確認します。

2.2.2.3. トランザクションログディレクトリーの変更

トランザクションログにより、Directory Server は、インスタンスが予期せずにシャットダウンした後にデータベースを復元できます。特定の状況では、管理者はトランザクションログへのパスを変更したい場合があります。たとえば、Directory Server データベースとは異なる物理ディスクに保存するには、以下のコマンドを実行します。



注記

パフォーマンスを向上させるには、場所を変更する代わりに、トランザクションログが含まれるディレクトリーに高速ディスクをマウントします。詳細は、『Red 『Hat Directory Server パフォーマンスチューニングガイド』の該当するセクションを参照してください』。

トランザクションログディレクトリーの場所を変更するには、以下を行います。

1. Directory Server インスタンスを停止します。

```
# systemctl stop dirsrv@instance_name
```

2. トランザクションログ用に新しい場所を作成します。以下に例を示します。

```
# mkdir -p /srv/dirsrv/instance_name/db/
```

3. Directory Server のみがディレクトリーにアクセスできるように、パーミッションを設定します。

```
# chown dirsrv:dirsrv /srv/dirsrv/instance_name/db/
# chmod 770 /srv/dirsrv/instance_name/db/
```

4. 以前のトランザクションログディレクトリーからすべての **__db.*** ファイルを削除します。以下に例を示します。

```
# rm /var/lib/dirsrv/slapd-instance_name/db/__db.*
```

5. 以前のトランザクションログディレクトリーから新しいトランザクションログディレクトリーに、すべての **log.*** ファイルを移動します。以下に例を示します。

```
# mv /var/lib/dirsrv/slapd-instance_name/db/log.* \
    /srv/dirsrv/instance_name/db/
```

6. SELinux が **enforcing** モードで実行している場合は、ディレクトリーに ***dirsrv_var_lib_t*** コンテキストを設定します。

```
# semanage fcontext -a -t dirsrv_var_lib_t /srv/dirsrv/instance_name/db/
# restorecon -Rv /srv/dirsrv/instance_name/db/
```

7. **/etc/dirsrv/slapd-*instance_name*/dse.ldif** ファイルを編集し、**cn=config,cn=ldbm database,cn=plugins,cn=config** エントリーの ***nsslapd-db-logdirectory*** パラメーターを更新します。以下に例を示します。

```
dn: cn=config,cn=ldbm database,cn=plugins,cn=config
...
nsslapd-db-logdirectory: /srv/dirsrv/instance_name/db/
```

8. インスタンスを起動します。

```
# systemctl start dirsrv@instance_name
```

2.3. データベースリンクの作成および維持

チェーン とは、サーバーがクライアントアプリケーションの代わりに他のサーバーに接続し、組み合わせた結果を返すことを意味します。チェーンは **データベースリンク** を介して実装され、リモートで保存されたデータを参照します。クライアントアプリケーションがデータベースリンクからデータを要求すると、データベースリンクは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。

- 「新規データベースリンクの作成」
- 「シャーシポリシーの設定」
- 「データベースリンクの維持」
- 「データベースリンクのデフォルトの設定」
- 「データベースリンクの削除」
- 「データベースリンクおよびアクセス制御評価」

チェーンに関する一般的な情報は、『Red 『Hat Directory Server デプロイメントガイド』の「ディレクトリートポロジーの設計」の章を参照してください』。「データベースリンクアクティビティーの監視」では、データベースリンクアクティビティーを監視する方法を説明します。

2.3.1. 新規データベースリンクの作成

基本的なデータベースリンクの設定には、以下の 4 つの情報が必要です。

- **接尾辞の情報。** 接尾辞は、通常のデータベースではなく、データベースリンクが管理するディレクトリーツリーに作成されます。この接尾辞は、データが含まれるリモートサーバーの接尾辞に対応します。
- **バインド認証情報。** データベースリンクがリモートサーバーにバインドされると、ユーザーのなりすましが行われ、リモートサーバーとバインドするために使用する各データベースリンクの DN および認証情報を指定します。
- **LDAP URL。** これは、データベースリンクが接続するリモートサーバーの LDAP URL を提供します。URL はプロトコル (ldap または ldaps)、サーバーのホスト名または IP アドレス (IPv4 または IPv6)、およびポートで構成されます。
- **フェイルオーバーサーバーの一覧。** これは、障害発生時にデータベースリンクが接続するための代替サーバーの一覧を提供します。この設定項目は任意です。

注記

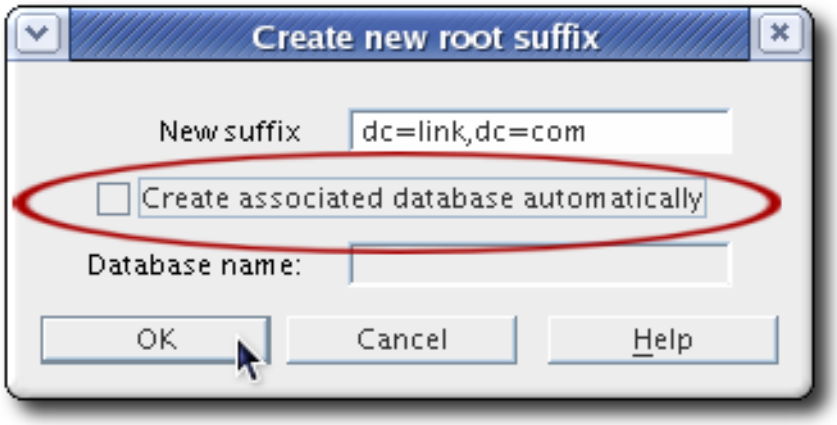
シンプルなパスワード認証 (「セキュアなバインドの要求」) にセキュアなバインドが必要な場合は、セキュアな接続で行われる場合を除き、チェーン操作は失敗します。セキュアな接続（TLS および Start TLS 接続または SASL 認証）の使用が推奨されます。

2.3.1.1. コンソールを使用した新規データベースリンクの作成

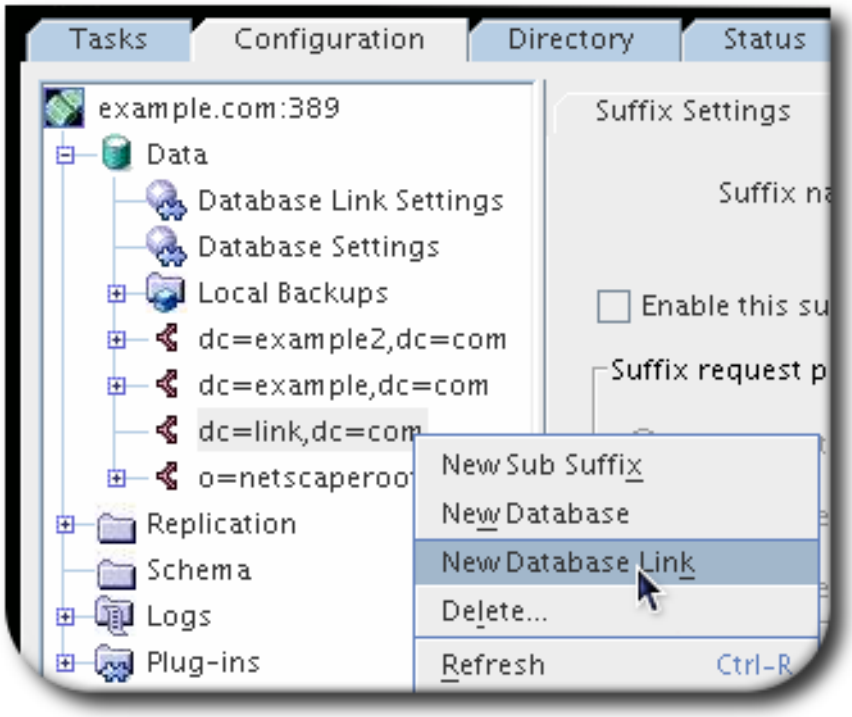
1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 「接尾辞の作成」の説明に従って、新しい接尾辞を作成します。

   Create associated database automatically のチェックボックスの選択を解除します。データベースとデータベースリンクにディレクトリーデータを配布するカスタムディストリビュー

ション機能が必要になるため、データベースとデータベースリンクに、接尾辞にデータベースリンクを設定するのが簡単になります。

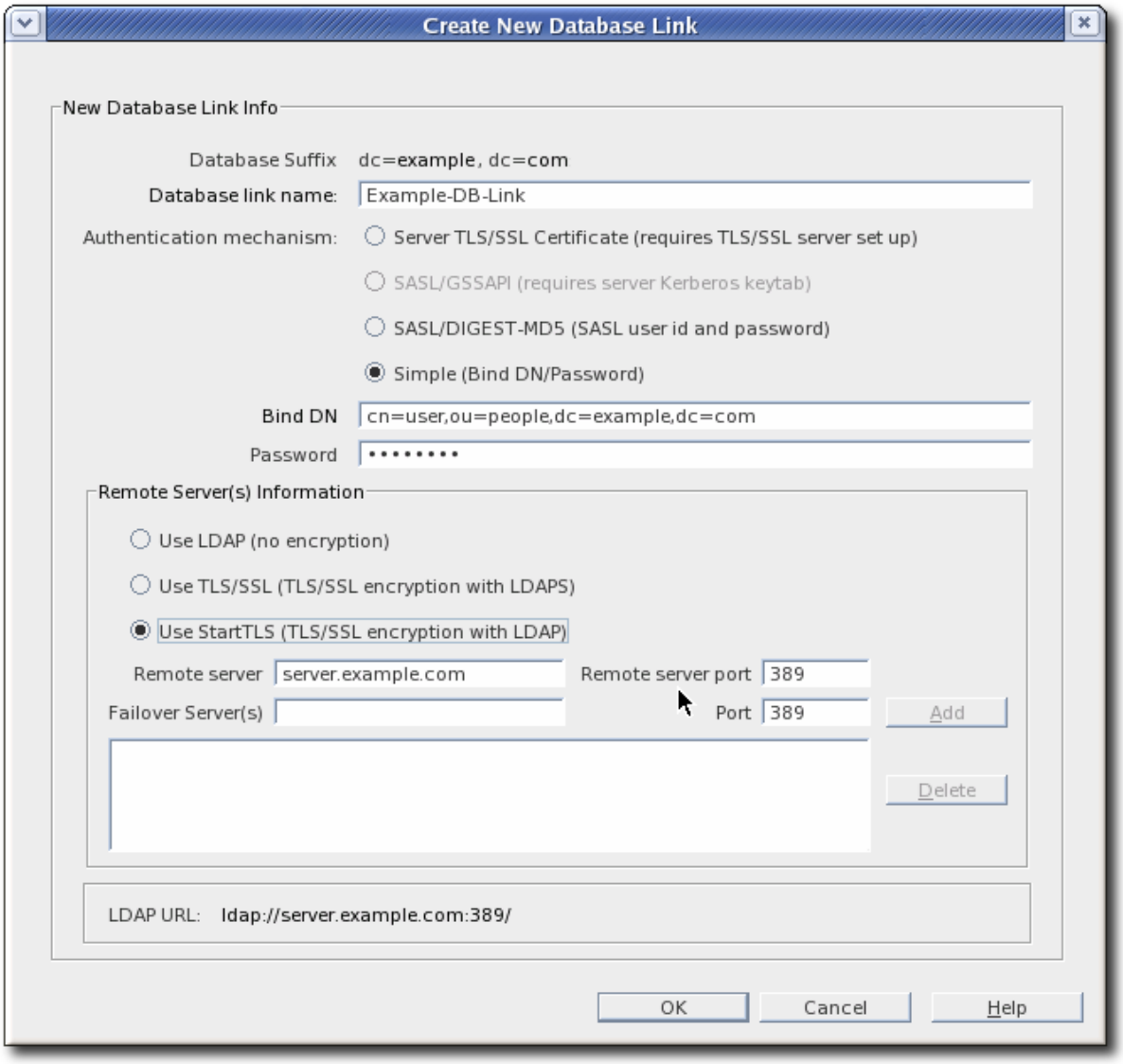


3. 左側のペインで、新しい接尾辞を右クリックし、ポップアップメニューから **New Database Link** を選択します。



4. データベースリンク名を入力します。名前は、英数字、ダッシュ(-)、およびアンダースコア(_)の組み合わせになります。スペースなどの他の文字は使用できません。

5. 認証に適切な方法にラジオボタンを設定します。

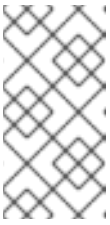


認証方法は 4 つあります。

- **simple** は、サーバーが、暗号化なしで標準ポートで接続することを意味します。必要な情報は、サーバーがリモートサーバーに接続するユーザーのバインド DN およびパスワードです。

- サーバーの **TLS/SSL 証明書は**、ローカルサーバーの TLS 証明書を使用して、リモートサーバーに対して認証します。証明書は、証明書ベースの認証用にローカルサーバーにインストールされ、リモートサーバーに証明書マッピングが設定され、ローカルサーバーの証明書のサブジェクト DN を対応するユーザーエントリーにマッピングできるように、証明書マッピングを設定する必要があります。

  TLS および証明書マッピングの設定については、「TLS の有効化」 を参照してください。



注記

データベースリンクとリモートサーバーが TLS を使用して通信するよう設定されている場合は、操作要求を行うクライアントアプリケーションも TLS を使用して通信する必要があるわけではありません。クライアントは、通常のポートを使用してバインドできます。

- **SASL/DIGEST-MD5** では、認証を行うためにバインド DN およびパスワードのみが必要になります。

- **SASL/GSSAPI** では、ローカルサーバーに Kerberos キータブ（「KDC サーバーおよびキータブの概要」にあるように）があり、リモートサーバーにローカルサーバーのプリンシパルを実際のユーザーエントリーにマッピングする SASL マッピングが必要です。「コンソールからの SASL アイデンティティーマッピングの設定」

6. **Remote Server Information** セクションで、ローカルサーバーがリモートサーバーへの接続に使用する接続タイプを選択します。以下の 3 つのオプションがあります。

   - **LDAP を使用します**。これにより、標準の暗号化されていない接続が設定されます。
   - **TLS/SSL を使用します**。これは、**636** などのサーバーのセキュアな LDAPS ポートを介したセキュアな接続を使用します。この設定は、TLS/TLS を使用するために必要です。

     TLS を使用する場合は、リモートサーバーのポート番号がセキュアなポートに設定されていることを確認します。
   - **Start TLS を使用します**。Start TLS を使用して、サーバーの標準ポートでセキュアな接続を確立します。

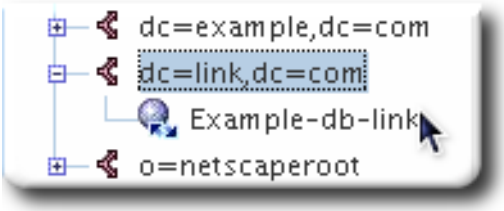
注記

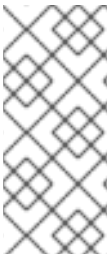
シンプルなパスワード認証 (「セキュアなバインドの要求」) にセキュアなバインドが必要な場合は、セキュアな接続で行われる場合を除き、チェーン操作は失敗します。セキュアな接続（TLS および Start TLS 接続または SASL 認証）の使用が推奨されます。

7. **Remote Server Information** セクションで、リモートサーバーの名前（ホスト名、IPv4 アドレス、または IPv6 アドレス）とポート番号を入力します。

   フェイルオーバーサーバーの場合は、ホスト名とポート番号を入力し、**追加** ボタンをクリックします。フェイルオーバーサーバーはバックアップサーバーであるため、プライマリーリモートサーバーが失敗すると、データベースリンクはフェイルオーバーサーバー一覧の最初のサーバーに接続し、サーバーがアクセスされるまでリストを循環します。

新しいデータベースリンクは、データベースの代わりに接尾辞の下に一覧表示されます。





注記

コンソールは、正常にバインドするためにデータベースリンクが **リモートサーバー** に存在する必要がある情報のチェックリストを提供します。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、Authentication タブをクリックし **ます**。このチェックボックスは、リモートサーバー **のチェックボックスにチェックを記入**します。

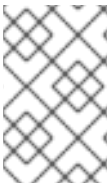
2.3.1.2. コマンドラインからのデータベースリンクの作成

1. **ldapmodify** コマンドラインユーティリティーを使用して、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 エントリーに配置する必要があり** ます。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
```

2. データベースリンクの設定情報を指定します。

```
dn: cn=examplelink,cn=chaining database,cn=plugins,cn=config
changetype: add
objectclass: top
objectclass: extensibleObject
objectclass: nsBackendInstance
nsslapd-suffix: ou=people,dc=example,dc=com suffix being chained
nsfarmserverurl: ldap://people.example.com:389/  LDAP URL to remote server
nsMultiplexorBindDN: cn=proxy admin,cn=config bind DN
nsMultiplexorCredentials: secret bind password
cn: examplelink
```



注記

シンプルなパスワード認証 (「セキュアなバインドの要求」) にセキュアなバインドが必要な場合は、セキュアな接続で行われる場合を除き、チェーン操作は失敗します。セキュアな接続（TLS および Start TLS 接続または SASL 認証）の使用が推奨されます。

デフォルト設定属性は **cn=default instance config,cn=chaining database,cn=plugins,cn=config** エントリーに含まれます。これらの設定属性は、作成時にすべてのデータベースリンクに適用されます。デフォルト設定の変更は、新しいデータベースリンクにのみ影響します。既存のデータベースリンクのデフォルト設定属性を変更することはできません。

各データベースリンクには、データベースリンクエントリー自体(cn= database_**link , cn=chaining database,cn=plugins,cn= config**)で保存される独自の設定情報が含まれます。設定属性の詳細は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス を参照してください』。

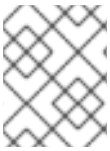
- 「接尾辞情報の提供」
- 「バインド認証情報の提供」
- 「LDAP URL の提供」
- 「フェイルオーバーサーバーの一覧の提供」
- 「異なるバインドメカニズムの使用」
- 「データベースリンクの設定属性の概要」
- 「データベースリンクの設定例」

2.3.1.2.1. 接尾辞情報の提供

nsslapd-suffix 属性を使用して、データベースリンクが管理する接尾辞を定義します。たとえば、データベースリンクが会社のリモートサイトの人情報を参照する場合は、以下の接尾辞情報を入力します。

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

接尾辞情報は、**cn=database_link, cn=chaining database,cn=plugins,cn=config** エントリーに保存されます。



注記

データベースリンクの作成後、***nsslapd-nsslapd-suffix*** 属性の変更は、データベースリンクを含むサーバーが再起動しないと適用されません。

2.3.1.2.2. バインド認証情報の提供

クライアントアプリケーションからの要求がリモートサーバーにチェーンされるようにするには、クライアントアプリケーションに特別なバインド認証情報を指定できます。これにより、リモートサーバーでチェーン操作に必要なプロキシー認証権限が付与されます。バインド認証情報がないと、データベースリンクは **anonymous** としてリモートサーバーにバインドされます。

バインド認証情報を指定するには、以下の手順が必要です。

1. リモートサーバーで以下を行います。
   - データベースリンクの管理ユーザーを作成します。

     エントリーの追加に関する詳細は、「3章 *ディレクトリーエントリーの管理*」を参照してください。
   - データベースリンクによってチェーンされたサブツリーの手順 1 で作成した管理ユーザーのプロキシーアクセス権限を指定します。

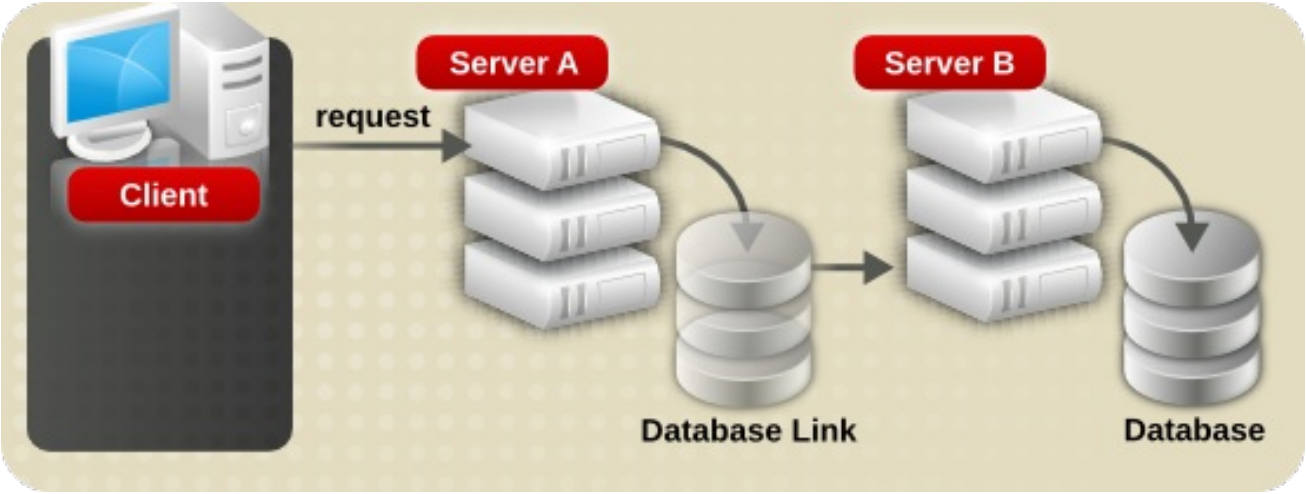
     ACI の設定に関する詳細は、「18章 *アクセス制御の管理*」を参照してください。
2. データベースリンクを含むサーバーで **ldapmodify** を使用して、cn= database_link **, cn=chaining database,cn= plugins,cn=config**エントリーの ***nsMultiplexorBindDN*** 属性にデータベースリンクのユーザー DN を提供します。

警告

nsMultiplexorBindDN は、Directory Manager に含めることはできません。

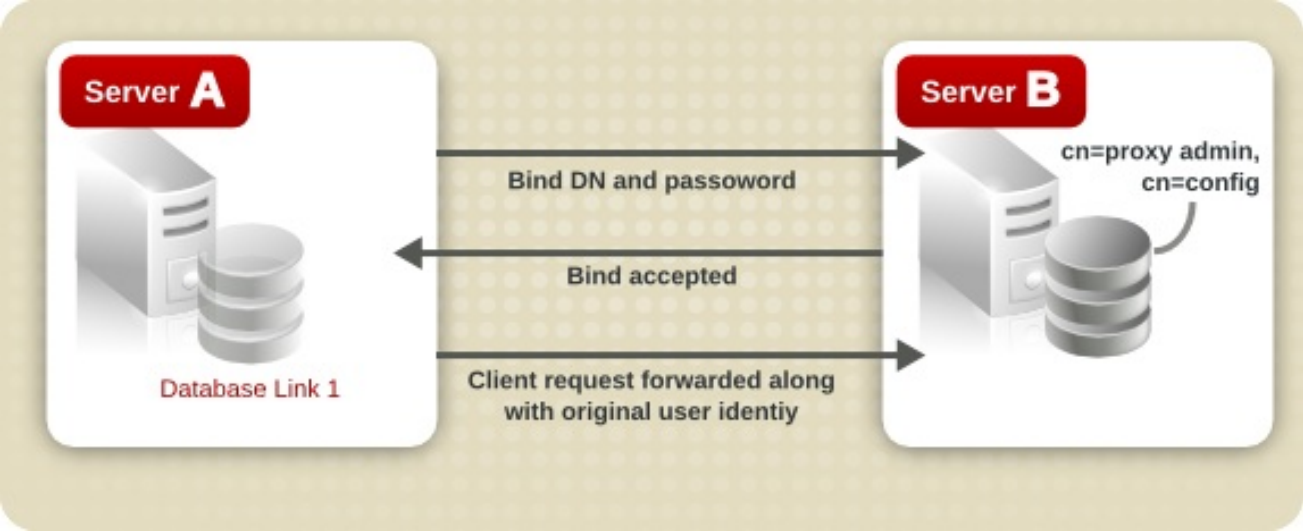
ldapmodify を使用して、cn= database_link **, cn=chaining database,cn= plugins,cn=config**エントリーの ***nsMultiplexorCredentials*** 属性に、データベースリンクのユーザーパスワードを提供します。

たとえば、クライアントアプリケーションは要求をサーバー A に送信します。サーバー A には、サーバー B のデータベースに要求をチェーンするデータベースリンクが含まれています。



サーバー A のデータベースリンクは、***nsMultiplexorBindDN*** 属性で定義されている特別なユーザーと ***nsMultiplexorCredentials*** 属性で定義されているユーザーパスワードを使用してサーバー B にバインドされます。この例では、サーバー A は以下のバインド認証情報を使用します。

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



サーバー B には ***nsMultiplexorBindDN*** に対応するユーザーエントリーが含まれ、このユーザーのプロキシー認証権限を設定する必要があります。プロキシー承認を正しく設定するには、プロキシー ACI をその他の ACI として設定します。

警告

ディレクトリーの制限された領域へのアクセス権限を付与しないようにチェーンを有効にする場合は、アクセス制御を慎重に検討します。たとえば、ブランチにデフォルトのプロキシー ACI が作成されると、データベースリンクを使用して接続するユーザーは、そのブランチの下にあるすべてのエントリーを表示できるようになります。ユーザーがすべてのサブツリーを表示する必要がない場合もあります。セキュリティーホールを回避するには、追加の ACI を作成して、サブツリーへのアクセスを制限します。

ACI の詳細は、「18章 *アクセス制御の管理*」を参照してください。

注記

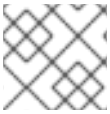
エントリーの作成または変更にクライアントアプリケーションでデータベースリンクを使用する場合、***creatorsName*** 属性および ***modifiersName*** 属性にはエントリーの実際の作成者または修正者が反映されません。これらの属性には、リモートデータサーバーでプロキシー認証権限が付与されている管理ユーザーの名前が含まれています。

2.3.1.2.3. LDAP URL の提供

データベースリンクが含まれるサーバーで、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ーを特定します。標準の LDAP URL 形式とは異なり、リモートサーバーの URL は接尾辞を指定しません。**ldap:**//server:port の形式を取ります。ここで、**サーバーは** ホスト名、IPv4 アドレス、または IPv6 アドレスになります。

nsFarmServerURL 属性を使用したリモートサーバーの URL は、設定ファイルの **cn=database_link, cn=chaining database,cn=plugins,cn=config** エントリーに設定されます。

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



注記

URL の最後で末尾のスラッシュ(/)を使用することを忘れないでください。

データベースリンクが TLS 経由で LDAP を使用してリモートサーバーに接続する場合、リモートサーバーの LDAP URL は URL の LDAP ではなくプロトコル LDAP を使用し、サーバーのセキュアなポートを参照します。 以下に例を示します。

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

注記

TLS を介してチェーンするには、ローカル Directory Server とリモート Directory Server でTLSを有効にする必要があります。TLS の有効化に関する詳細は、「TLS の有効化」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ーが TLS を使用して通信するよう設定されている場合は、操作要求を行うクライアントアプリケーションも TLS を使用して通信する必要があるわけではありません。クライアントは、通常のポートを使用してバインドできます。

2.3.1.2.4. フェイルオーバーサーバーの一覧の提供

障害発生時に使用するサーバーには、追加の LDAP URL が存在する可能性があります。***nsFarmServerURL*** 属性に代替サーバーを追加し、空白で区切って追加します。

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

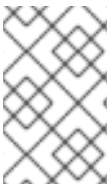
このサンプル LDAP URL では、データベースリンクは最初に標準ポートで **example.com** サーバーに問い合わせ、操作を処理します。応答しない場合、データベースリンクはポート **389** でサーバー **us.example.com** に問い合わせます。このサーバーが失敗した場合は、ポート **1000** で **africa.example.com** にお問い合わせください。

2.3.1.2.5. 異なるバインドメカニズムの使用

ローカルサーバーは、複数の異なる接続タイプと認証メカニズムを使用してリモートサーバーに接続できます。

ローカルサーバーがリモートサーバーに接続する方法は 3 つあります。

- 標準の LDAP ポートを使用する場合
- 専用の LDAPS ポートを使用する場合
- Start TLS（標準ポートでのセキュアな接続）の使用



注記

シンプルなパスワード認証 (「セキュアなバインドの要求」) にセキュアなバインドが必要な場合は、セキュアな接続で行われる場合を除き、チェーン操作は失敗します。セキュアな接続（TLS および Start TLS 接続または SASL 認証）の使用が推奨されます。

最終的に 2 つの接続設定があります。TLS オプションでは、サーバーの両方が TLS 経由の接続を実行および許可するように設定されますが、TLS を強制する別の設定属性はありません。

接続タイプは ***nsUseStartTLS*** 属性で識別されます。これを有効にすると、サーバーは標準ポート **で** Start TLS 接続を開始します。これが **オフ** の場合、サーバーは ***nsFarmServerURL*** 属性のリモートサーバーに設定されたものに応じて LDAP ポートまたは LDAPS ポートを使用します。

たとえば、Start TLS を使用するには、以下を実行します。

```
nsUseStartTLS: on
```

たとえば、標準接続または TLS 接続を使用するには、以下を実行します。

```
nsUseStartTLS: off
```

ローカルサーバーがファームサーバーへの認証に使用できる 4 つの方法があります。

- **empty.**バインドメカニズムが設定されていない場合、サーバーは簡易認証を実行し、バインド情報を付与する ***nsMultiplexorBindDN*** 属性および ***nsMultiplexorCredentials*** 属性を必要とします。
- **EXTERNAL**。これは TLS 証明書を使用して、ファームサーバーをリモートサーバーに認証します。ファームサーバーをセキュアな URL (**ldaps**) に設定するか、***nsUseStartTLS*** 属性を **on** に設定する必要があります。

  さらに、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス の『certmap.conf』 セクションで説明されているように、ファームサーバーの証明書をバインド ID にマッピングするようにリモートサーバーを設定する必要があります。』
- **DIGEST-MD5**。これは、DIGEST-MD5 暗号化での SASL 認証を使用します。単純な認証と同様に、バインド情報を付与するには ***nsMultiplexorBindDN*** 属性および ***nsMultiplexorCredentials*** 属性が必要です。
- **GSSAPI.**SASL 上で Kerberos ベースの認証を使用します。

  ファームサーバーは Kerberos キータブで設定する必要があるため、リモートサーバーには、そのファームサーバーのバインド ID に対して定義された SASL マッピングが必要です。Kerberos

キータブおよび SASL マッピングの設定は、「SASL Identity マッピングの設定」に記載されています。

注記

SASL 接続は、標準の接続または TLS 接続で確立できます。

以下に例を示します。

```
nsBindMechanism: EXTERNAL
```

注記

SASL を使用する場合は、SASL およびパスワードポリシーコンポーネントをチェーンするようにローカルサーバーを設定する必要もあります。「シャーシポリシーの設定」で説明されているように、データベースリンク設定のコンポーネントを追加します。以下に例を示します。

```
ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=config,cn=chaining database,cn=plugins,cn=config
changetype: modify
add: nsActiveChainingComponents
nsActiveChainingComponents: cn=password policy,cn=components,cn=config
-
add: nsActiveChainingComponents
nsActiveChainingComponents: cn=sasl,cn=components,cn=config
^D
```

2.3.1.2.6. データベースリンクの設定属性の概要

以下の表は、データベースリンクの設定に使用できる属性を表しています。これらの属性の一部は、先のセクションで説明されました。すべてのインスタンス属性は、**cn=database_link, cn =chaining database,cn=plugins,cn=config** エントリーで定義されます。

特定のデータベースリンクに定義されている値は、グローバル属性値よりも優先されます。

表2.1 データベースリンクの設定属性

属性	値
nsTransmittedControls [†]	リモートデータサーバーへデータベースリンクによって転送される LDAP コントロールの OID を指定します。
nsslapd-suffix	データベースリンクで管理される接尾辞。エントリーの作成後にこの属性への変更は、データベースリンクを含むサーバーが再起動しないと反映されません。
nsslapd-timelimit	データベースリンクの検索時間制限のデフォルト（秒単位）。デフォルト値は **3600** 秒です。

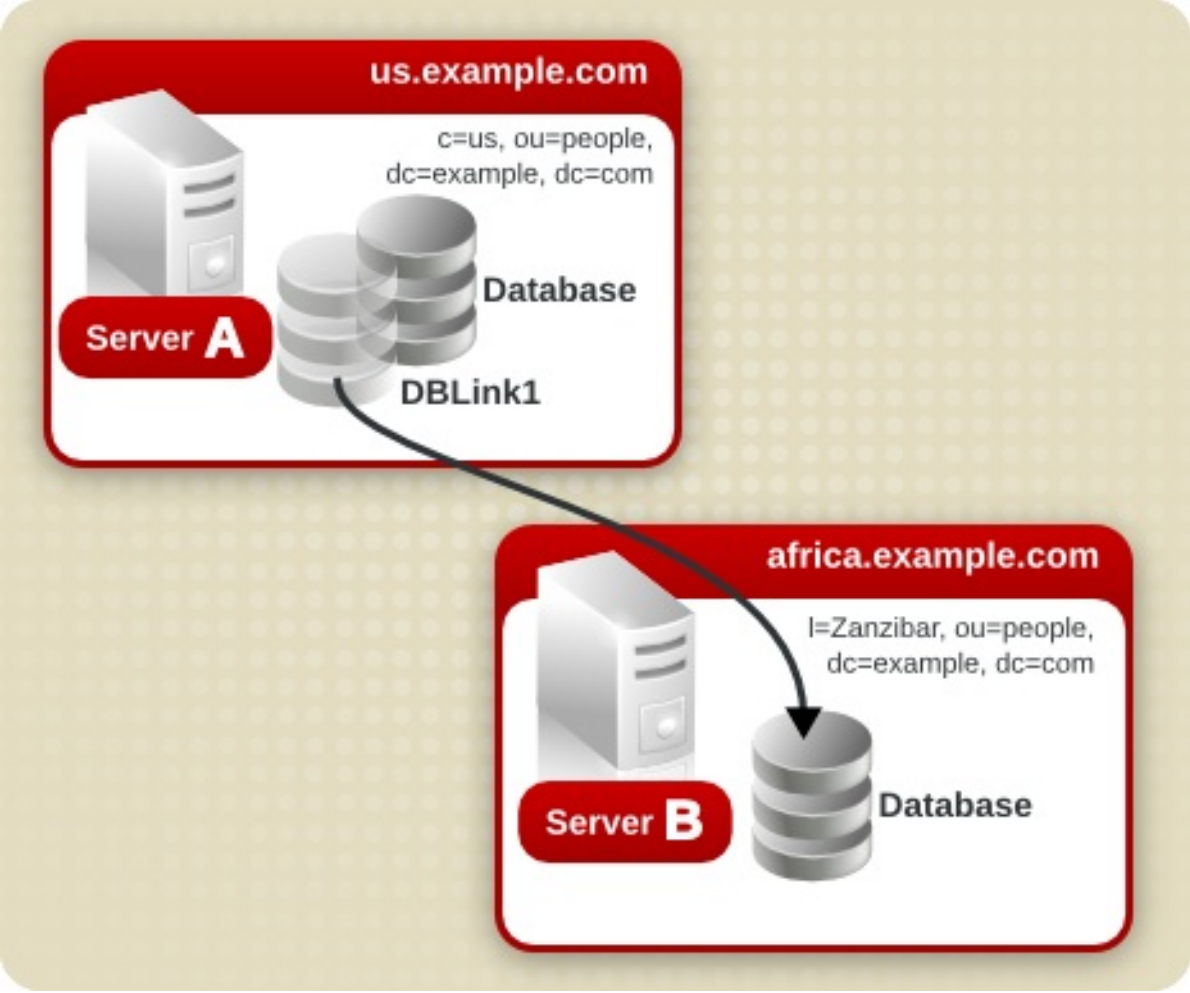
属性	値
nsslapd-sizelimit	エントリーの数で指定するデータベースリンクのデフォルトサイズ制限。デフォルト値は **2000** エントリーです。
nsFarmServerURL	データが含まれるリモートサーバー（またはファームサーバー）の LDAP URL を指定します。この属性には、空白で区切られた、フェイルオーバーのオプションサーバーを含めることができます。カスケード連鎖を使用する場合、この URL は別のデータベースリンクを参照できます。
nsUseStartTLS	Start TLS を使用して標準ポートでセキュアな接続を確立するかどうかを設定します。デフォルトは **off** で、単純な（標準）接続と TLS 接続の両方に使用されます。
nsBindMechanism	リモートサーバーの認証（バインド）に使用する認証方法を設定します。空の値を設定する場合は、単純なバインド(LDAP_SASL_SIMPLE)が使用されます。
nsMultiplexorBindDN	リモートサーバーと通信するために使用される管理エントリーの DN。属性名の ***multiplexor*** という用語は、データベースリンクが含まれ、リモートサーバーと通信するサーバーを意味します。このバインド DN は Directory Manager にすることはできません。この属性が指定されていない場合、データベースリンクは **anonymous** としてバインドします。
nsMultiplexorCredentials	管理ユーザーのパスワード（プレーンテキストで指定されます）。パスワードが提供されない場合、ユーザーは **anonymous** としてバインドできることを意味します。パスワードは設定ファイルで暗号化されます。
nsCheckLocalACI	高度な使用のために予約されます。ACI がデータベースリンクとリモートデータサーバーで評価されるかどうかを制御します。**on** または **off** の値を取ります。この属性への変更は、サーバーの再起動後にのみ行われます。デフォルト値は **off** です。
nsProxiedAuthorization	高度な使用のために予約されます。プロキシー化された承認を無効にします。**off** を指定すると、プロキシー認証が無効になります。デフォルト値は **on** です。
nsActiveChainingComponents [†]	は、チェーンを使用するコンポーネントを一覧表示します。コンポーネントとは、サーバー内の機能的な単位です。データベースリンクインスタンスのこの属性の値は、グローバル設定属性の値を上書きします。特定のデータベースインスタンスでチェーンを無効にするには、値 **none** を使用します。デフォルトのポリシーはチェーンを許可しません。詳細は、「コンポーネントの動作の連鎖」 を参照してください。
nsReferralOnScopedSearch	スコープ指定の検索によって参照を返すかどうかを制御します。この属性は、スコープ指定された検索に対して応答された参照を返す方がより効率的であるため、ディレクトリーを最適化します。**on** または **off** の値を取ります。デフォルト値は **off** です。
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。デフォルト値は **10** です。

属性	値
[†] グローバル属性およびインスタンス属性の両方を指定できます。このグローバル設定属性は、cn **=config,cn=chaining database,cn=plugins,cn=config** エントリーにあります。グローバル属性は動的であるため、その属性への変更はディレクトリー内のデータベースリンクのすべてのインスタンスに対して自動的に有効になります。	

詳細は、Red『Hat Directory Server の設定、コマンド、およびファイルリファレンス のパラメーターの説明を参照してください』。

2.3.1.2.7. データベースリンクの設定例

us.example.com ドメイン内のサーバーに、データベースのサブツリー **l=Walla Walla,ou=people,dc=example,dc=com** が含まれており、**l=Zanzibar,ou=people,dc=example,dc=com** サブツリーの操作要求を、**africa.example.com** ドメインの別のサーバーにチェーンする必要があります。



1. **ldapmodify** を実行して、サーバー A にデータベースリンクを追加します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
```

2. データベースリンクの設定情報を指定します。

```
dn: cn=DBLink1,cn=chaining database,cn=plugins,cn=config
changetype: add
objectclass: top
objectclass: extensibleObject
objectclass: nsBackendInstance
nsslapd-suffix: c=africa,ou=people,dc=example,dc=com
nsfarmserverurl: ldap://africa.example.com:389/
nsMultiplexorBindDN: cn=proxy admin,cn=config
nsMultiplexorCredentials: secret
cn: DBLink1

dn: cn="c=africa,ou=people,dc=example,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=example,dc=com
cn: c=africa,ou=people,dc=example,dc=com
```

最初のエントリーでは、***nsslapd-suffix*** 属性には、サーバー A からチェーンするサーバー B の接尾辞が含まれます。***nsFarmServerURL*** 属性には、サーバー B の LDAP URL が含まれます。

2 番目のエントリーは新しい接尾辞を作成し、サーバーが新しいデータベースリンクに行われた要求をルーティングできるようにします。***cn*** 属性には、データベースリンクの ***nsslapd-suffix*** 属性に指定された同じ接尾辞が含まれます。***nsslapd-backend*** 属性には、データベースリンクの名前が含まれます。***nsslapd-parent-suffix*** 属性は、この新しい接尾辞 **ou=people,dc=example,dc=com** の親を指定します。

3. 以下のように、サーバー B で管理ユーザーを作成します。

```
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
```

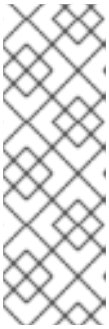
警告

Directory Manager ユーザーは、リモートサーバーのプロキシー管理ユーザーとして使用しないでください。これにより、セキュリティーホール が作成されます。

4. サーバー B の **l=Zanzibar,ou=people,dc=example,dc=com** エントリーに、以下のプロキシー承認 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、**l=Zanzibar,ou=people,dc=example,dc=com** サブツリー内に含まれるリモートサーバーに含まれるデータに、プロキシー admin ユーザーの読み取り専用アクセスのみを提供します。



注記

ユーザーがデータベースリンクにバインドすると、ユーザーのアイデンティティーがリモートサーバーに送信されます。アクセス制御は、常にリモートサーバーで評価されます。ユーザーがリモートサーバーにデータを変更または書き込みできるようにするには、リモートサーバーに正しいアクセス制御を設定します。チェーン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の詳細は、「データベースリンクおよびアクセス制御評価」を参照してください。

2.3.2. シャーシポリシーの設定

この手順では、クライアントアプリケーションによって行われた要求を、データベースリンクを含む Directory Serverに、Directory Server が連鎖させる方法の構成を説明します。このチェーンポリシーは、Directory Server で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。

2.3.2.1. コンポーネントの動作の連鎖

コンポーネントは、内部操作を使用するサーバーの機能ユニットです。たとえば、プラグインはフロントエンドの機能のようにコンポーネントとみなされます。ただし、プラグインは実際には複数のコンポーネントで構成される場合があります (例: ACI プラグイン)。

一部のコンポーネントは、ローカルデータのみにアクセスできることを想定し、内部 LDAP 要求をサーバーに送信します。このようなコンポーネントの場合、チェーンポリシーを制御し、コンポーネントが操作を正常に完了できるようにします。一例として、証明書の検証機能が挙げられます。証明書をチェックするために関数によって行われる LDAP 要求を連鎖させると、リモートサーバーが信頼されていることを意味します。リモートサーバーが信頼されていない場合は、セキュリティーの問題があります。

デフォルトでは、すべての内部操作はチェーンされず、チェーンにはコンポーネントは許可されませんが、これは上書きできます。

また、指定したプラグインがリモートサーバーで操作を実行できるようにするには、リモートサーバーに ACI を作成する必要があります。ACI は、データベースリンクに割り当てられた **接尾辞** に存在している必要があります。

以下は、コンポーネント名、内部操作をチェーンできるようにする可能性のある副次的な影響、およびリモートサーバーの ACI で必要なパーミッションを一覧表示します。

ACI プラグイン

このプラグインはアクセス制御を実装します。ACI 属性の取得および更新に使用される操作は、ローカルおよびリモートの ACI 属性を混在することは安全ではないため、連鎖されません。ただし、ユーザーエントリーの取得に使用されるリクエストは、チェーンコンポーネント属性を設定することでチェーンすることができます。

```
nsActiveChainingComponents: cn=ACI Plugin,cn=plugins,cn=config
```

権限: 読み取り、検索、および比較

リソース制限コンポーネント

このコンポーネントは、ユーザーバインド DN に応じてサーバー制限を設定します。リソース制限コンポーネントを連鎖させることが可能であれば、リソース制限をリモートユーザーに適用できます。リソース制限コンポーネント操作を連鎖させるには、連鎖コンポーネント属性を追加します。

```
nsActiveChainingComponents: cn=resource limits,cn=components,cn=config
```

権限: 読み取り、検索、および比較

証明書ベースの認証チェックコンポーネント

このコンポーネントは、外部バインドメソッドが使用される場合に使用します。リモートサーバーのデータベースからユーザー証明書を取得します。このコンポーネントの連鎖を許可すると、証明書ベースの認証がデータベースリンクと連携できることを意味します。このコンポーネントの動作を連鎖させるには、チェーンコンポーネント属性を追加します。

```
nsActiveChainingComponents: cn=certificate-based authentication,cn=components,cn=config
```

権限: 読み取り、検索、および比較

パスワードポリシーコンポーネント

このコンポーネントは、リモートサーバーへの SASL バインドを許可するために使用されます。SASL 認証の形式によっては、ユーザー名とパスワードを使用した認証が必要になります。パスワードポリシーを有効にすると、サーバーは要求された特定の認証方法を検証および実装し、適切なパスワードポリシーを適用できます。このコンポーネントの動作を連鎖させるには、チェーンコンポーネント属性を追加します。

```
nsActiveChainingComponents: cn=password policy,cn=components,cn=config
```

権限: 読み取り、検索、および比較

SASL コンポーネント

このコンポーネントは、リモートサーバーへの SASL バインドを許可するために使用されます。このコンポーネントの動作を連鎖させるには、チェーンコンポーネント属性を追加します。

```
nsActiveChainingComponents: cn=password policy,cn=components,cn=config
```

権限: 読み取り、検索、および比較

参照整合性プラグイン

このプラグインは、DN を含む属性の更新が、属性へのポインターを含むすべてのエントリーに伝播されるようにします。たとえば、グループのメンバーであるエントリーが削除されると、そのエントリーはグループから自動的に削除されます。このプラグインをチェーンで使用すると、グループメンバーが静的グループ定義にリモートになる場合に静的グループの管理を簡素化できます。このコンポーネントの動作を連鎖させるには、チェーンコンポーネント属性を追加します。

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

権限: 読み取り、検索、および比較

Attribute Uniqueness プラグイン

このプラグインは、指定された属性のすべての値が一意 (重複なし) のものであることを確認します。このプラグインが連鎖されている場合は、データベースリンクで変更された属性でも属性値が一意であることを確認します。このコンポーネントの動作を連鎖させるには、チェーンコンポーネント属性を追加します。

```
nsActiveChainingComponents: cn=attribute uniqueness,cn=plugins,cn=config
```

権限: 読み取り、検索、および比較

ロールのコンポーネント

このコンポーネントは、データベースのエントリーのロールおよびロール割り当てを連鎖させます。このコンポーネントの連鎖は、連鎖されたデータベースであってもロールを維持します。このコンポーネントの動作を連鎖させるには、チェーンコンポーネント属性を追加します。

```
nsActiveChainingComponents: cn=roles,cn=components,cn=config
```

権限: 読み取り、検索、および比較

注記

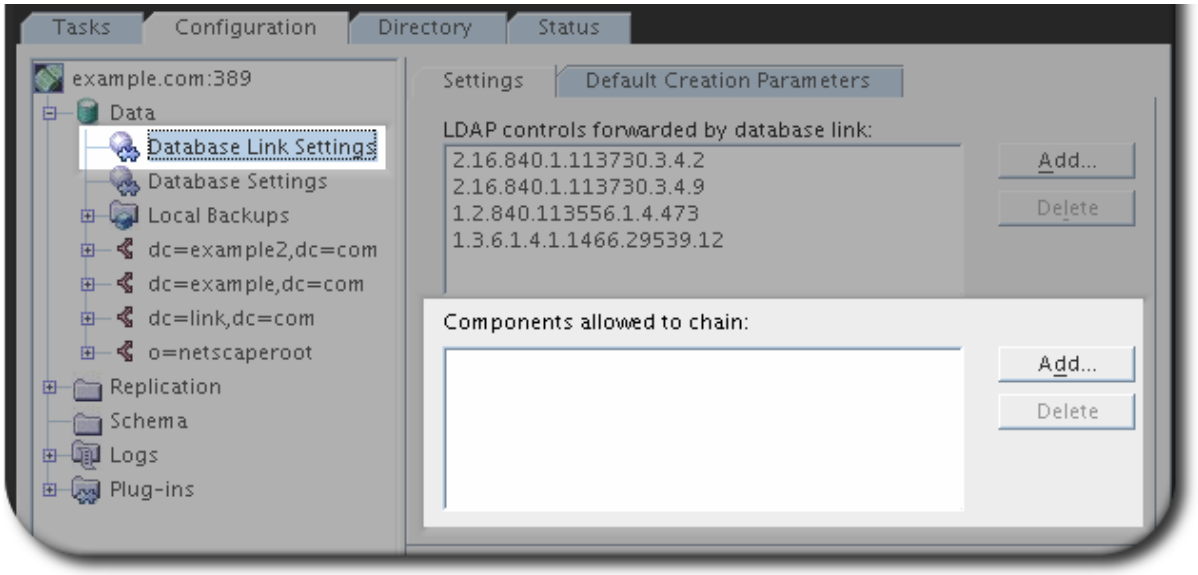
以下のコンポーネントはチェーンできません。

- Roles 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
- 参照整合性プラグイン

連鎖要求を発行しているサーバーで Referential Integrity プラグインを有効にする場合は、パフォーマンス、リソース、時間のニーズ、整合性のニーズを分析してください。整合性チェックは時間がかかり、メモリーと CPU を浪費する可能性があります。ACI および連鎖関連の制限の詳細は、「ACI の制限」を参照してください。

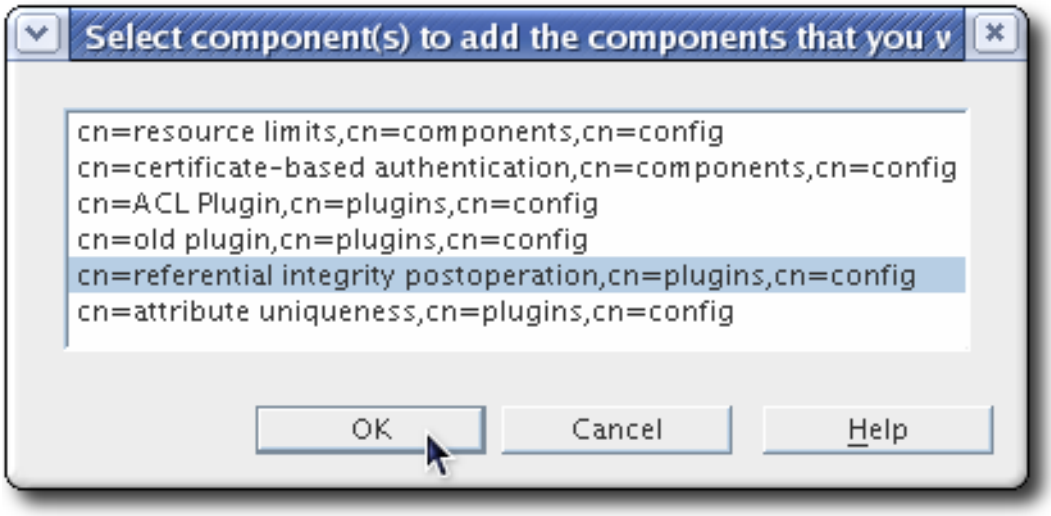
2.3.2.1.1. コンソールを使用したコンポーネントの操作の連鎖

1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のペインで **Data** を展開し、**Database Link Settings** をクリックします。
3. 右側のウィンドウで **Settings** タブを選択します。



4. **チェーン可能な Components** の **Add** ボタンをクリックします。

5. 一覧からチェーンするコンポーネントを選択し、**OK** をクリックします。



6. 変更を反映するためにサーバーを再起動します。

コンポーネントが連鎖した後、操作が連鎖されるリモートサーバーの接尾辞に ACI を作成します。たとえば、これにより Referential Integrity プラグインの ACI が作成されます。

```
aci: (targetattr "*")(target="ldap:///ou=customers,l=us,dc=example,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";)
```

2.3.2.1.2. コマンドラインでのコンポーネントの操作の連鎖

1. 設定ファイルの **cn=config,cn=chaining database,cn=plugins,cn=config** エントリーの ***nsActiveChainingComponents*** 属性を使用して、チェーンに追加するコンポーネントを指定します。

   たとえば、参照整合性コンポーネントがチェーン操作を実行できるようにするには、以下をデータベースリンク設定ファイルに追加します。

   -

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

連鎖が可能なコンポーネントの一覧は、「コンポーネントの動作の連鎖」を参照してください。

2. 変更を有効にするためにサーバーを再起動します。

```
# systemctl restart dirsrv@instance_name
```

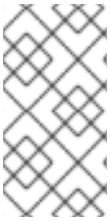
3. 操作を連鎖させるリモートサーバーの接尾辞に ACI を作成します。たとえば、これにより Referential Integrity プラグインの ACI が作成されます。

```
aci: (targetattr "*")(target="ldap:///ou=customers,l=us,dc=example,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";)
```

2.3.2.2. LDAP 制御チェーン

LDAP 制御による操作リクエストを連鎖させることは **できません**。デフォルトでは、以下の制御で行われる要求は、データベースリンクによってリモートサーバーに転送されます。

- **仮想リストビュー (VLV)。**この制御は、すべてのエントリー情報を返すのではなく、エントリーの一部のリストを提供します。
- **サーバー側のソート。**この制御では、通常は特定のマッチングルールを使用して、エントリーを属性値に従ってソートます。
- **逆参照。**この制御は、検索内のエントリー属性の参照を上書きし、参照されるエントリーから指定された属性情報をプルし、残りの検索結果とともに返します。
- **管理 DSA。**この制御は、参照に従うのではなく、スマート参照をエントリーとして返します。そのため、スマートの参照自体は変更または削除できます。
- **ループ検出。**この制御では、別のサーバーとのサーバー連鎖の回数を追跡します。数が設定された数に達すると、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が送信されます。この制御の使用方法は、「ループの検出」を参照してください。

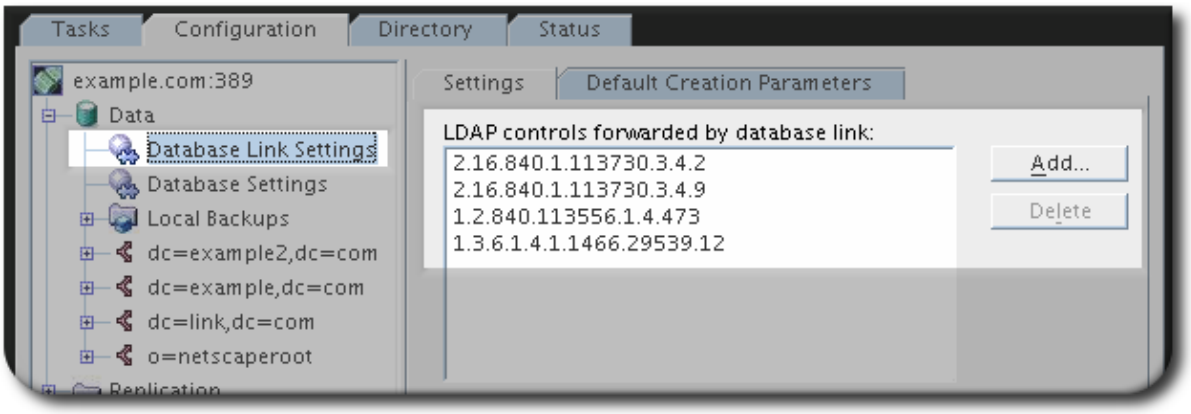


注記

サーバー側のソートおよび VLV 制御は、1 つのデータベースにクライアントアプリケーション要求が行われている場合にのみサポートされます。データベースリンクは、クライアントアプリケーションが複数のデータベースに要求を行う場合に、これらの制御をサポートしません。

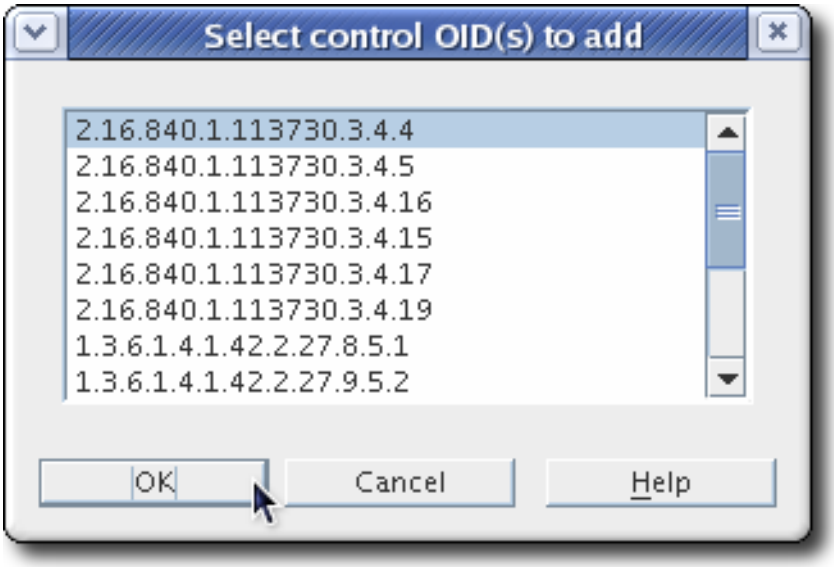
2.3.2.2.1. コンソールを使用した LDAP 制御の連鎖

1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のペインで **Data** フォルダーを展開し、**Database Link Settings** をクリックします。
3. 右側のウィンドウで **Settings** タブを選択します。



4. **データベースリンクセクションによって転送される LDAP Controls** の **Add** ボタンをクリックして、LDAP コントロールを一覧に追加します。

5. リストに追加するコントロールの OID を選択し、**OK** をクリックします。



2.3.2.2.2. コマンドラインでの LDAP 制御の連鎖

チェーン制御には、cn=config,cn= **chaining database,cn=plugins,cn=config** エントリーの ***nsTransmittedControls*** 属性を変更して、データベースリンクが転送するコントロールを変更します。たとえば、仮想リストビュー制御を転送するには、以下を設定ファイルのデータベースリンクエントリーに追加します。

```
nsTransmittedControls: 2.16.840.1.113730.3.4.9
```

さらに、Directory Server のクライアントが独自の制御を作成し、その操作をリモートサーバーにチェーンする必要がある場合は、***nsTransmittedControls*** 属性にカスタム制御の OID を追加します。

チェーン可能な LDAP 制御とその OID を以下の表に示します。

表2.2 LDAP コントロールとその OID

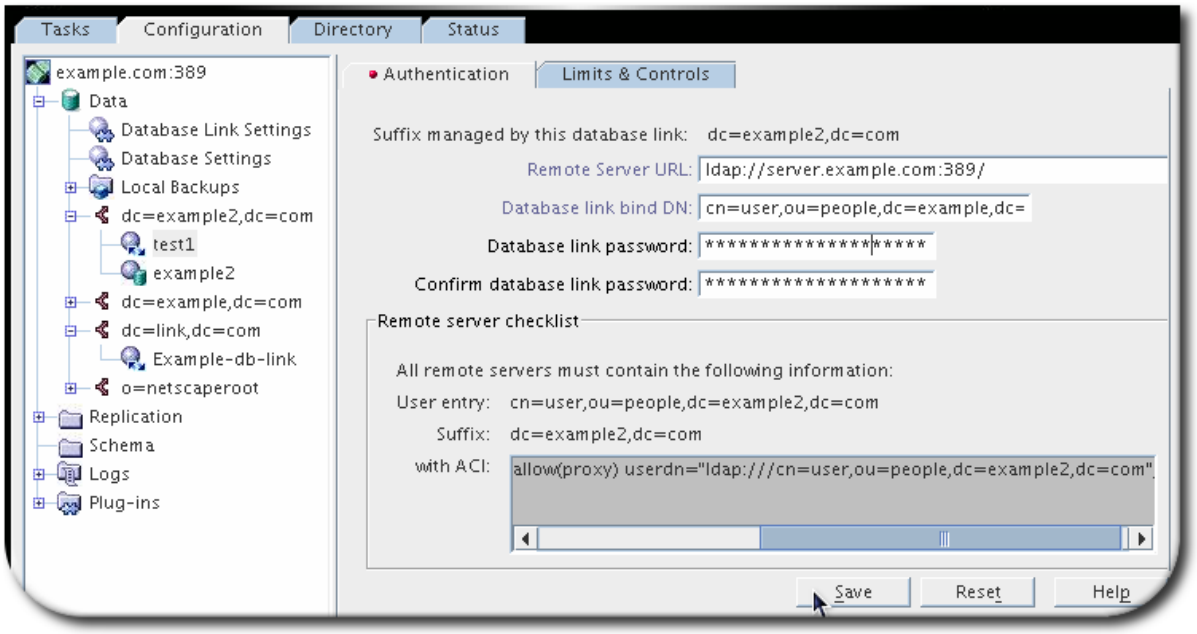
コントロール名	OID
仮想リストビュー (VLV)	2.16.840.1.113730.3.4.9

コントロール名	OID
サーバー側のソート	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
検索の逆参照	1.3.6.1.4.1.4203.666.5.16

2.3.3. データベースリンクの維持

リモートサーバーへの接続用のデータベースリンクのすべての情報。

1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のペインで、**Data** フォルダーを展開し、接尾辞の下にあるデータベースリンクを選択します。
3. 右側のペインで、Authentication タブをクリックし ます。



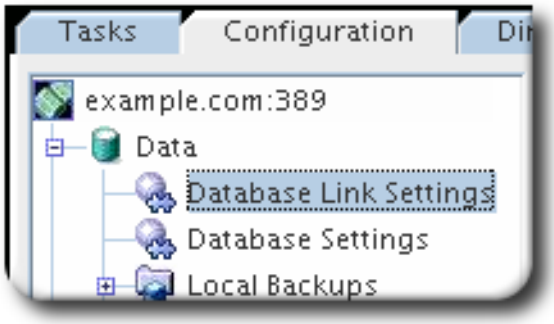
4. 接続情報を変更します。
   - リモートサーバーの LDAP URL。[]
   - リモートサーバーにバインドするためにデータベースリンクで使用されるバインド DN およびパスワード。

2.3.4. データベースリンクのデフォルトの設定

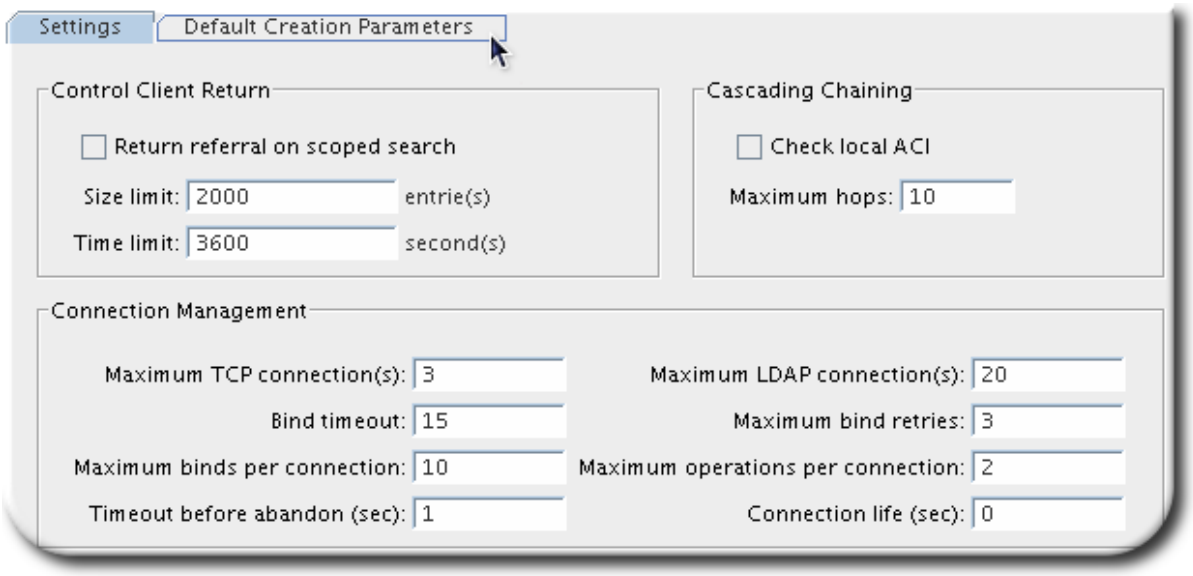
データベースリンクのデフォルト設定は、カスケード連鎖に使用される設定（クライアント要求に許可されるホップの数）、リモートサーバーの接続ルール、およびサーバーがクライアントリクエストに応答する方法を定義します。

1. **Configuration** タブを選択します。

2. 左側のペインで **Data** フォルダーを展開し、**Database Link Settings** をクリックします。**デフォルトの作成パラメーター** タブを開きます。



3. 新しい設定パラメーターを入力します。



注記

データベースリンクのデフォルト設定への変更は、再度適用されません。デフォルト設定に変更を加えた後に作成されたデータベースリンクのみが変更を反映します。

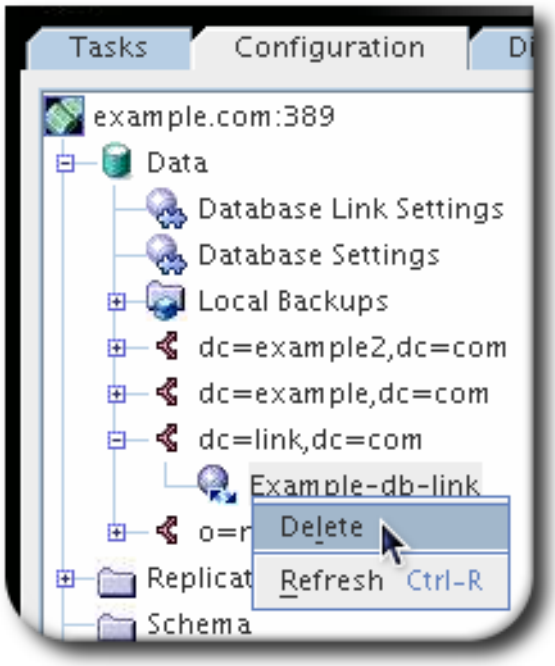
2.3.5. データベースリンクの削除

データベースリンクを削除するには、データベースリンクを右クリックし、ポップアップメニューから **Delete** を選択します。プロンプトが表示されたら、削除を確認します。

1. Directory Server コンソールで、**Configuration** タブを選択します。

2. 左側のナビゲーションペインで **Data** の下で接尾辞を開き、削除するデータベースリンクを選択します。

3. データベースリンクを右クリックし、メニューから **Delete** を選択します。



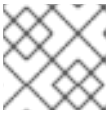
2.3.6. データベースリンクおよびアクセス制御評価

ユーザーがデータベースリンクを含むサーバーにバインドすると、データベースリンクがユーザーの ID をリモートサーバーに送信します。アクセス制御は、常にリモートサーバーで評価されます。リモートサーバーで評価されるすべての LDAP 操作は、プロキシーが設定された承認コントロールを使用して渡されたクライアントアプリケーションの元の ID を使用します。ユーザーがリモートサーバーに含まれるサブツリーに正しいアクセス制御がある場合に限り、リモートサーバーで操作に成功します。これには、いくつかの制限があるリモートサーバーに通常のアクセス制御を追加する必要があります。

- すべての種類のアクセス制御を使用できるわけではありません。

  たとえば、ロールベースまたはフィルターベースの ACI はユーザーエントリーへのアクセスを必要とします。データベースリンクを介してデータにアクセスするため、プロキシーコントロールのデータのみが検証できます。ユーザーエントリーがユーザーのデータと同じデータベースに配置されるように、ディレクトリーを設計することを検討してください。

- クライアントの IP アドレスまたは DNS ドメインに基づくすべてのアクセス制御は、チェーン中にクライアントの元のドメインが失われるためです。リモートサーバーは、クライアントアプリケーションをデータベースリンクと同じ IP アドレスと、同じ DNS ドメインにある表示します。



注記

Directory Server は、IPv4 と IPv6 の IP アドレスの両方に対応します。

データベースリンクで使用される ACI には、以下の制限が適用されます。

- ACI は、使用する任意のグループと共に配置する必要があります。グループが動的である場合は、グループ内のすべてのユーザーが ACI およびグループで配置される必要があります。グループが静的である場合は、リモートユーザーにリンクされます。

- ACI は、使用するロール定義とそれらのロールを持つユーザーに対して配置する必要があります。

- ユーザーのエントリーの値 (例: **userattr** サブジェクトルール) にリンクする ACI は、ユーザーがリモートの場合に機能します。

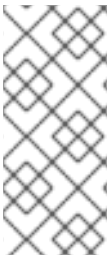
アクセス制御は常にリモートサーバーで評価されますが、データベースリンクとリモートサーバーの両方が含まれるサーバーでも評価できます。これにはいくつかの制限があります。

- アクセス制御の評価時に、ユーザーエントリーの内容は必ずしも利用できるとは限りません (たとえば、データベースリンクを含むサーバーでアクセス制御が評価され、エントリがリモートサーバーにある場合)。

  パフォーマンス上の理由から、クライアントはリモート問い合わせを実行してアクセス制御を評価することはできません。

- データベースリンクは、クライアントアプリケーションによって変更されるエントリーに必ずしもアクセスできるとは限りません。

  変更操作を実行する場合、データベースリンクはリモートサーバーに保存されている全エントリーにアクセスできません。削除操作を実行すると、データベースリンクはエントリーの DN のみを認識します。アクセス制御が特定の属性を指定する場合、データベースリンクを介して実行すると削除操作は失敗します。



注記

デフォルトでは、データベースリンクが含まれるサーバーに設定されたアクセス制御は評価されません。このデフォルトを上書きするには、**cn=database_link, cn=chaining database,cn=plugins,cn=config** エントリーの ***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ーでアクセス制御を評価することは、カスケード連鎖を使用する場合を除いて推奨されません。

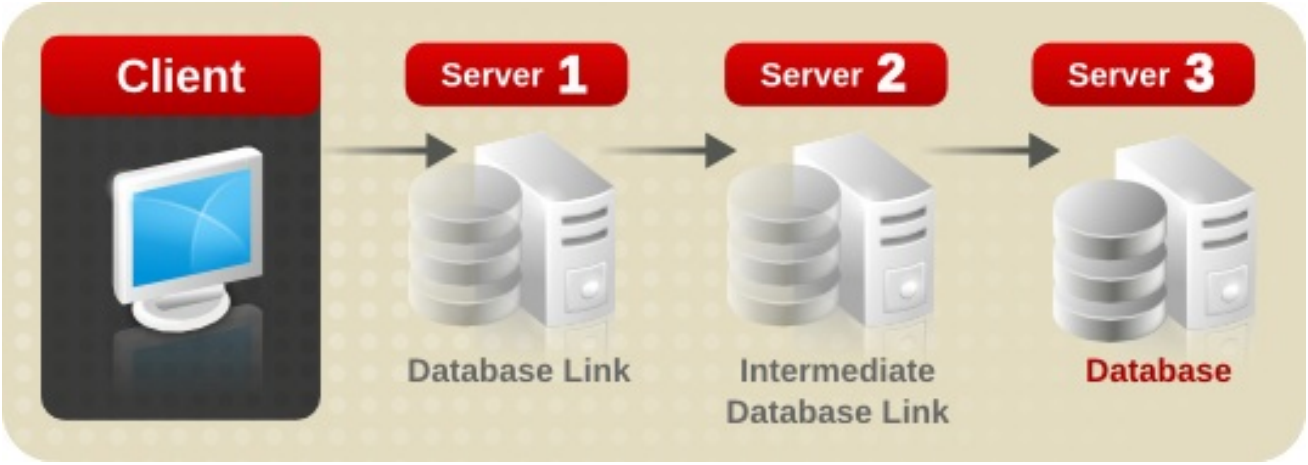
2.4. カスケード連鎖の設定

データベースリンクは、別のデータベースリンクに指定するように設定でき、カスケード連鎖操作を作成します。ディレクトリーツリー内の全データにアクセスするのに、複数のホップが必要になるとカスケード連鎖がいつでも発生します。

- 「カスケード連鎖の概要」
- 「コンソールを使用したカスケード連鎖の設定」
- 「コマンドラインからのカスケード連鎖の設定」
- 「ループの検出」
- 「カスケード連鎖設定属性の概要」
- 「カスケード連鎖設定の例」

2.4.1. カスケード連鎖の概要

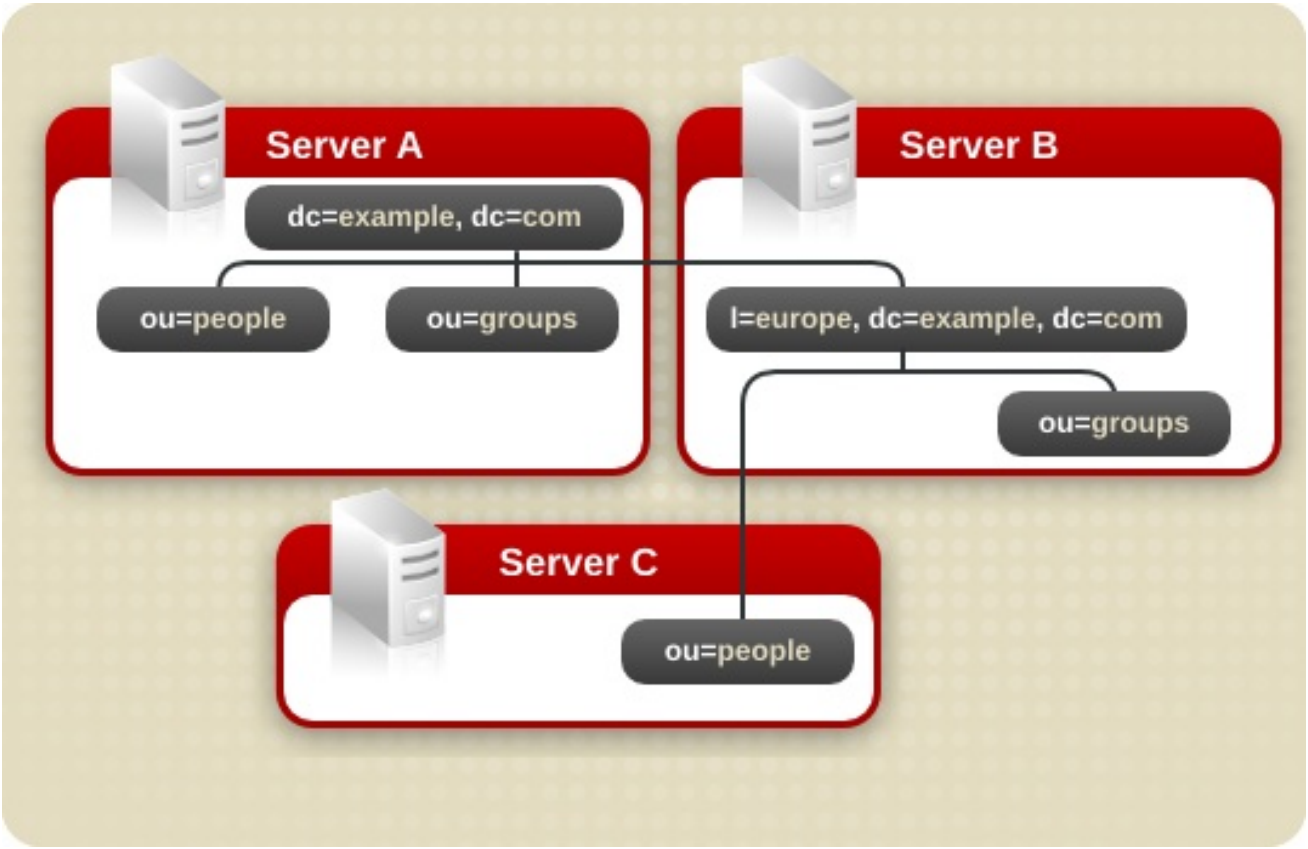
ディレクトリーがクライアントアプリケーションの要求を処理するために必要なホップが複数使用されると、カスケード連鎖が発生します。



クライアントアプリケーションは変更要求を Server 1 に送信します。Server 1 には、別のデータベースリンクが含まれる Server 2 に操作を転送するデータベースリンクが含まれています。Server 2 のデータベースリンクは、クライアントがデータベースに変更するデータが含まれる Server 3 に転送します。クライアントが変更するデータの一部にアクセスするには、2 つのホップが必要になります。

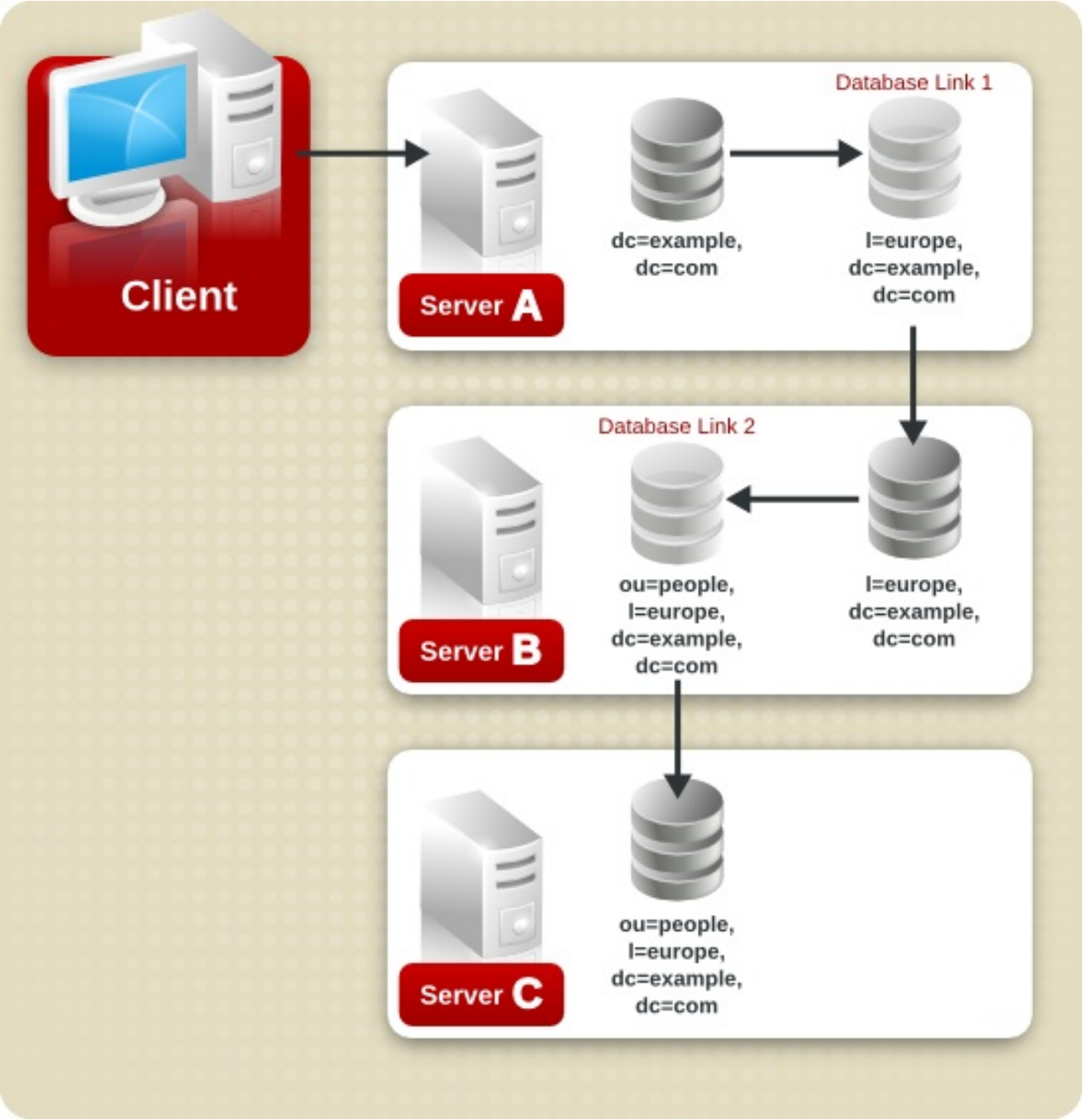
通常の操作要求時に、クライアントはサーバーにバインドし、そのクライアントに適用された ACI が評価されます。カスケード連鎖では、クライアントバインド要求は Server 1 で評価されますが、上の例の Server 2 クライアントに適用される ACI は、要求が宛先サーバーへチェーンされた後にのみ評価されます。

たとえば、サーバー A では、ディレクトリーツリーが分割されます。



ルート接尾辞 **dc=example,dc=comおよび ou=people** と **ou=groups** サブサフィックスは、サーバー A に保存されます。**l=europe,dc=example,dc=com** および **ou=groups** 接尾辞は Server B に保存され、**l=europe,dc=example,dc=com 接尾辞の ou= people** ブランチはサーバー C に保存されます。

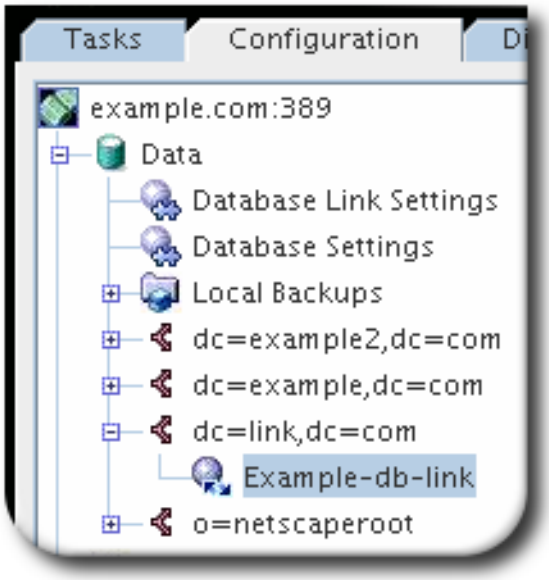
サーバー A、B、および C に設定されたカスケードでは、以下のように **ou=people,l=europe,dc=example,dc=com** エントリーでターゲットとなるクライアント要求が以下のようにルーティングされます。



まず、クライアントは、Database Link 1 を使用して Server A にバインドし、Server B に連鎖します。その後、Server B は、Database Link 2 を使用して **ou=people,l=europe,dc=example,dc=com** ブランチのデータにアクセスする Server C のターゲットデータベースに連鎖されます。ディレクトリーがクライアント要求を処理するには少なくとも 2 つのホップが必要であるため、これはカスケード連鎖とみなされます。

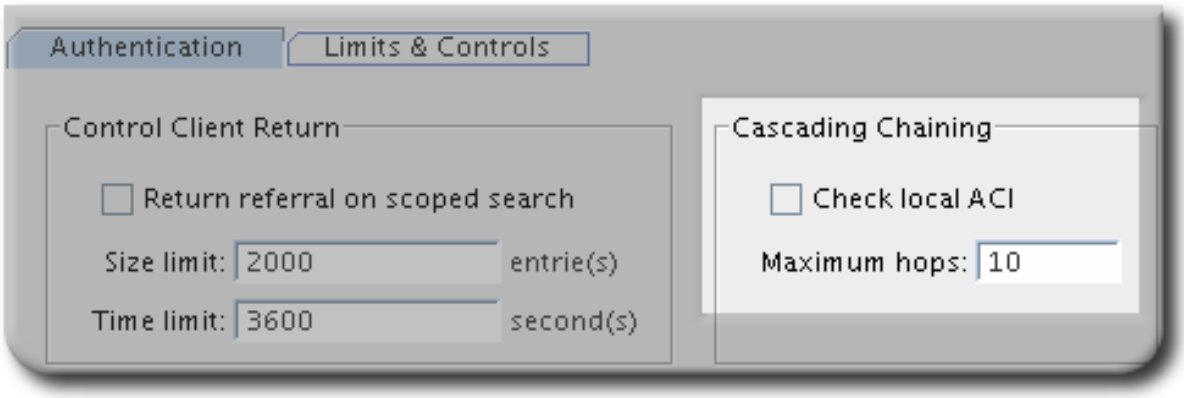
2.4.2. コンソールを使用したカスケード連鎖の設定

1. **Configuration** タブを選択します。左側のペインで **Data** フォルダーを展開し、接尾辞を選択してからデータベースリンクを選択します。



2. 右側のペインで **Limits and Controls** タブをクリックします。

3. **Check local ACI** チェックボックスを選択して、カスケードチェーンに関連する中間データベースリンクのローカル ACI の評価を有効にします。このチェックボックスを選択すると、適切なローカル ACI をデータベースリンクに追加する必要がある場合があります。



4. **Maximum hops** フィールドで、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最大回数を入力します。

   デフォルトでは、最大値は 10 ホップです。10 ホップ後、サーバー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。

2.4.3. コマンドラインからのカスケード連鎖の設定

コマンドラインでデータベースリンクのカスケードを設定するには、以下を実行します。

1. 中間データベースリンクが含まれるサーバーの URL に 1 つのデータベースリンクを指定します。

   カスケード連鎖を作成するには、あるデータベースリンクの ***nsFarmServerURL*** 属性には、別のデータベースリンクが含まれるサーバーの URL が含まれている必要があります。**example1.com と呼ばれるサーバーのデータベースリンクが、africa.example.com** と呼ばれるサーバーのデータベースリンクを参照するとします。たとえば、Server 1 のデータベースリンクの **cn= database_link, cn=chaining database,cn=plugins,cn=config**エントリーには以下が含まれます。

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

2. プロキシー認証制御を送信するように、中間データベースリンクまたはリンク（例： Server 2）を設定します。

   デフォルトでは、データベースリンクは Proxy Authorization Control を送信しません。ただし、1つのデータベースリンクが別の接続する場合、このコントロールは最終的な送信先サーバーに必要な情報を送信するために使用されます。中間データベースリンクはこの制御を送信する必要があります。プロキシー承認制御を送信するようにデータベースリンクを設定するには、中間データベースリンクの **cn=config,cn=chaining database,cn=plugins,cn=config** エントリーに以下を追加します。

   ```
   nsTransmittedControls: 2.16.840.1.113730.3.4.12
   ```

   OID 値は Proxy Authorization Control を表します。LDAP 制御チェーンの詳細は、「LDAP 制御チェーン」 を参照してください。

3. すべての中間データベースリンクに、プロキシー管理ユーザー ACI を作成します。

   ACI は、要求を別のサーバーに変換する前に、最初のデータベースリンクの権限を確認する中間データベースリンクが含まれるサーバーに存在している必要があります。たとえば、Server 2 が Server 1 の認証情報を確認しない場合、ユーザーは **匿名** としてバインドし、プロキシー認証制御を渡し、適切よりも多くの管理権限を許可できます。プロキシー ACI はこのセキュリティー違反を防ぎます。

   a. 中間データベースリンクが含まれるサーバーにデータベースがない場合は、データベースを作成します。このデータベースには、admin ユーザーエントリーおよび ACI が含まれます。データベースの作成に関する詳細は、「データベースの作成」 を参照してください。

   b. データベースの管理ユーザーに対応するエントリーを作成します。

   c. 適切な接尾辞をターゲットとする管理ユーザーの ACI を作成します。これにより、管理者はデータベースリンクの接尾辞にのみアクセスできます。以下に例を示します。

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

      この ACI は、簡単なチェーンの設定時にリモートサーバーに作成された ACI と似ています。

警告

ディレクトリーの制限された領域へのアクセス権限を付与しないようにチェーンを有効にする場合は、アクセス制御を慎重に検討します。たとえば、ブランチにデフォルトのプロキシー ACI が作成されると、データベースリンク経由で接続するユーザーは、ブランチの下にあるすべてのエントリーを表示できます。ユーザーがすべてのサブツリーを表示する必要がない場合もあります。セキュリティーホールを回避するには、追加の ACI を作成して、サブツリーへのアクセスを制限します。

4. すべての中間データベースリンクで、ローカル ACI 評価を有効にします。

   プロキシー管理 ACI が使用されていることを確認するには、チェーンに関連するすべての中間データベースリンクのローカル ACI の評価を有効にします。以下の属性を、各中間データベースリンクの **cn=database_link, cn=chaining database,cn=plugins,cn=config** エントリーに追加します。

   ```
   nsCheckLocalACI: on
   ```

   cn =default instance config,cn=chaining database,cn=plugins,cn=config エントリーでこの属性を **on** に設定すると、すべての新規データベースリンクインスタンスが cn= database **_link, cn=chaining database,** cn=plugins,cn=config エントリーで ***nsCheckLocalACI*** 属性が設定されることを意味します。

5. すべての中間データベースリンクおよび最終的な宛先データベースにクライアント ACI を作成します。

   ローカル ACI の評価が有効になっているため、適切なクライアントアプリケーション ACI をすべての中間データベースリンクおよび最終的な宛先データベースに作成する必要があります。中間データベースリンクでこれを行うには、まず最終的な宛先接尾辞のルート接尾辞を表す接尾辞が含まれるデータベースを作成します。

   たとえば、c **=africa,ou=people,dc=example,dc=com** 接尾辞に対して行われたクライアントの要求がリモートサーバーにチェーンされている場合、すべての中間データベースリンクに **dc=example,dc=com** 接尾辞に関連付けられたデータベースを含める必要があります。

   この上位接尾辞エントリーにクライアント ACI を追加します。以下に例を示します。

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

   この ACI により、Server 1 の **cn=config** エントリーに ***uid*** を持つクライアントアプリケーションが、サーバー 3 の **ou=people,dc=example,dc=com** 接尾辞の下にあるデータに対して、あらゆる種類の操作を実行できます。

2.4.4. ループの検出

Directory Server に含まれる LDAP 制御により、ループが回避されます。サーバーは、最初にチェーンを試行するとき、この制御を、許可されたホップの最大数またはチェーン接続の最大数に設定します。後続の各サーバーでカウントが減ります。サーバーが **0** の数を受信すると、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。

許可されるホップ数は、***nsHopLimit*** 属性を使用して定義されます。指定されていない場合、デフォルト値は **10** になります。

コントロールを使用するには、cn=config,cn= **chaining database,cn=plugins,cn=config** エントリーの ***nsTransmittedControl*** 属性に以下の 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設定ファイルに制御がない場合、ループ検出は実装されません。

2.4.5. カスケード連鎖設定属性の概要

以下に、カスケードチェーンで中間データベースリンクを設定するために使用される属性を説明します。

nsFarmServerURL

カスケードチェーンに次のデータベースリンクが含まれるサーバーの URL。

nsTransmittedControls

カスケードチェーンに関連するデータベースリンクに以下の OID を入力します。

```
nsTransmittedControls: 2.16.840.1.113730.3.4.12
nsTransmittedControls: 1.3.6.1.4.1.1466.29539.12
```

aci

この属性には以下の ACI が含まれる必要があります。

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

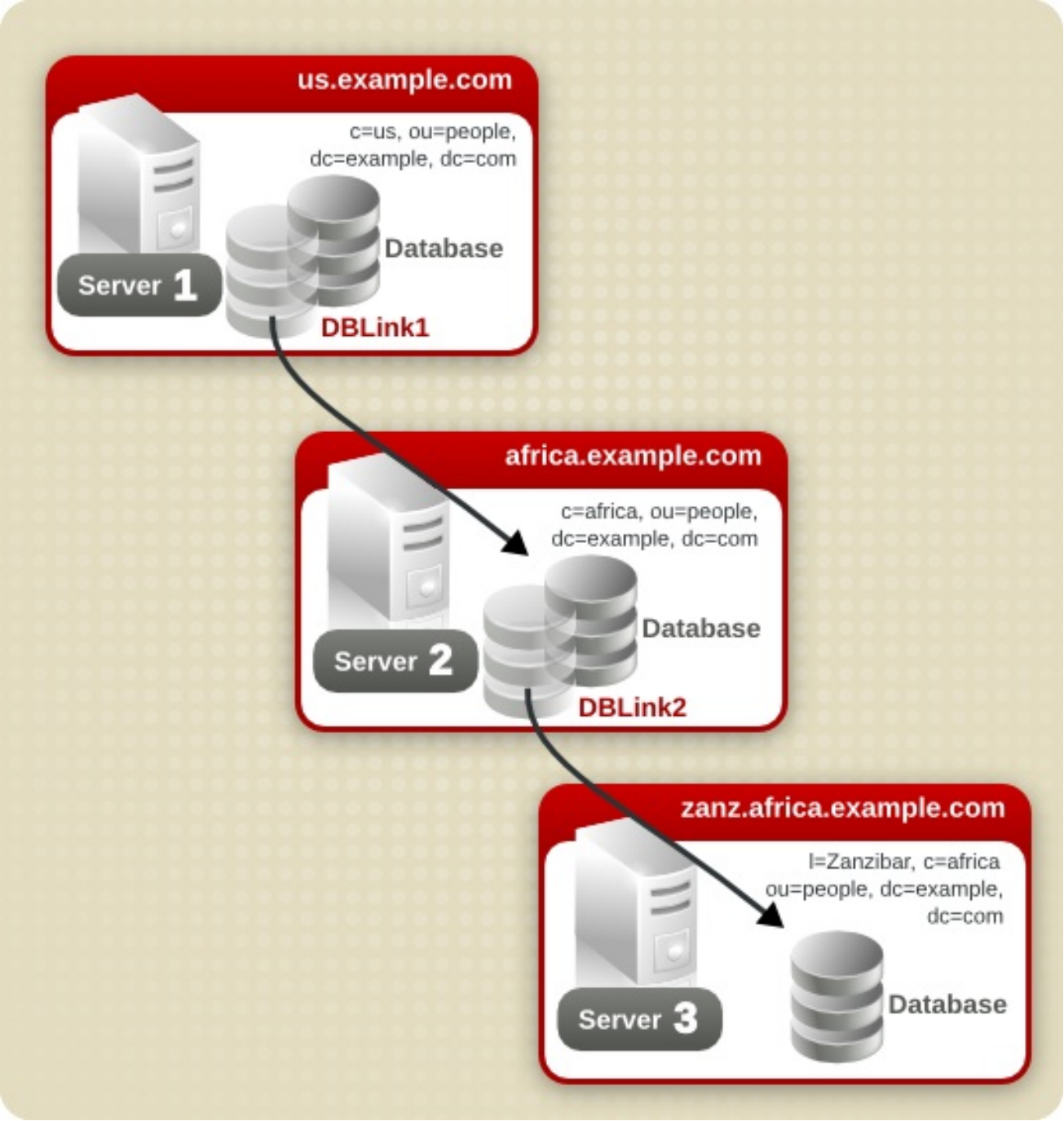
nsCheckLocalACI

チェーン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、以下のようにローカルの ACI 評価を実行します。

```
nsCheckLocalACI: on
```

2.4.6. カスケード連鎖設定の例

以下の図のように 3 つのサーバーに関連するカスケード連鎖を作成するには、チェーンコンポーネントを 3 つのサーバーすべてに設定する必要があります。



- 「サーバーを 1 台設定」
- 「Server Two の設定」
- 「サーバーの 3 つの設定」

2.4.6.1. サーバーを 1 台設定

1. **ldapmodify** を実行し、サーバー 1 でデータベースリンク DBLink1 の設定情報を指定します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=DBLink1,cn=chaining database,cn=plugins,cn=config
changetype: add
objectclass: top
objectclass: extensibleObject
```

```
objectclass: nsBackendInstance
nsslapd-suffix: c=africa,ou=people,dc=example,dc=com
nsfarmserverurl: ldap://africa.example.com:389/
nsMultiplexorBindDN: cn=server1 proxy admin,cn=config
nsMultiplexorCredentials: secret
cn: DBLink1
nsCheckLocalACI:off

dn: cn="c=africa,ou=people,dc=example,dc=com",cn=mapping tree,cn=config
changetype: add
objectclass: nsMappingTree
nsslapd-state: backend
nsslapd-backend: DBLink1
nsslapd-parent-suffix: ou=people,dc=example,dc=com
cn: c=africa,ou=people,dc=example,dc=com
```

最初のセクションでは、DB **Link1 に関連付けられた** エントリーを作成します。2 つ目のセクションでは、新しい接尾辞を作成し、サーバーが正しいサーバーにデータベースリンクに行われた要求を指示できるようにします。***nsCheckLocalACI*** 属性は、ローカル ACI をチェックするように設定する必要はありません。これは、Server 2 のデータベースリンク **DBLink2** でのみ必要です。

2. ループ検出を実装するには、Server 1 の **cn=config,cn=chaining database,cn=plugins,cn=config** エントリーに保存されている ***nsTransmittedControl*** 属性にループ検出制御の OID を指定します。

```
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、ループ検出 OID **1.3.6.1.4.1.1466.29539.12** の値でデフォルトで設定されているので、事前に確認してから、その属性が存在するかどうかを確認します。存在する場合は、この手順は必要ありません。

2.4.6.2. Server Two の設定

1. Server 2 でプロキシー管理ユーザーを作成します。この管理ユーザーは、サーバー 1 がバインドし、サーバー 2 への認証を許可するために使用されます。Server 1 に固有のプロキシー管理ユーザー名を選択すると便利です。これは、サーバー 2 **にバインドできる** プロキシー管理ユーザーです。以下のようにプロキシー管理ユーザーを作成します。

```
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

Directory Manager または管理者 ID ユーザーをリモートサーバーのプロキシー管理ユーザーとして使用しないでください。これにより、セキュリティーホール が作成されます。

2. Server 2 でデータベースリンク **DBLink2** を設定します。

```
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=example,dc=com
nsfarmserverurl: ldap://zanz.africa.example.com:389/
nsMultiplexorBindDN: cn=server2 proxy admin,cn=config
nsMultiplexorCredentials: secret
cn: DBLink2
nsCheckLocalACI:on

dn: cn="l=Zanzibar,c=africa,ou=people,dc=example,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=example,dc=com
cn: l=Zanzibar,c=africa,ou=people,dc=example,dc=com
```

データベースリンク DBLink2 はカスケード連鎖設定の中間データベースリンクであるため、***nsCheckLocalACI*** 属性を **on** に設定して、クライアントとプロキシーの管理ユーザーによるデータベースリンクへのアクセスを許可するかどうかをサーバーが確認できるようにします。

3. Server 2 のデータベースリンクは、プロキシー承認制御とループ検出制御を送信するように設定する必要があります。プロキシー認証制御とループ検出制御を実装するには、対応する OID の両方を指定します。以下の情報を Server 2 の **cn=config,cn=chaining database,cn=plugins,cn=config** エントリーに追加します。

```
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
```

nsTransmittedControl: 2.16.840.1.113730.3.4.12 は、プロキシー承認コントロールの OID です。**nsTransmittedControl: 1.3.6.1.4.1.1466.29539.12** はループ検出制御になります。

ループ検出制御が設定されているかどうかについて確認し、それに応じて上記のコマンドを調整します。

4. ACI を設定します。Server 2 で、**l=Zanzibar,c=africa,ou=people,dc=example,dc=com** 接尾辞の上に接尾辞が存在することを確認し、以下のアクションが利用できるようにします。

   - データベースリンク接尾辞の追加
   - サーバー 2 で作成されたプロキシー認証ユーザーを使用してサーバー 1 が接続できるようにローカルプロキシー認証 ACI を追加します。
   - ローカルクライアントの ACI を追加し、クライアント操作が Server 2 で成功し、サーバー 3 に転送できます。ローカルの ACI チェックが **DBLink2** データベースリンクに対して有効になっているため、このローカル ACI が必要です。

   どちらの ACI も **c=africa,ou=people,dc=example,dc=com** 接尾辞が含まれるデータベースに配置されます。

   注記

   これらの ACI を作成するには、エントリーを保持するために、c **=africa,ou=people,dc=example,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必要があります。このデータベースは、各データベースリンクの ***nsslapd-suffix*** 属性で指定された接尾辞上の接尾辞と関連付ける必要があります。つまり、最終的な宛先サーバーの接尾辞は、中間サーバーで指定された接尾辞のサブ接尾辞になります。

   a. ローカルプロキシー認証 ACI を **c=africa,ou=people,dc=example,dc=com** エントリーに追加します。

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

   b. 次に、ACI チェックが有効になっているとクライアント操作をサーバー 2 で成功できるようにするローカルクライアント ACI を追加します。この ACI は、**l=Zanzibar,c=africa,ou=people,dc=example,dc=com** ブランチへのアクセスを提供するために、宛先サーバーで作成された ACI と同じです。**c=us,ou=people,dc=example,dc=com** 内のすべてのユーザーには、サーバー 3 の **l=Zanzibar,c=africa,ou=people,dc=example,dc=com** のエントリーへの更新アクセス権が必要になる場合があります。**c=africa,ou=people,dc=example,dc=com** 接尾辞に以下の ACI を作成し、これを許可します。

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

   この ACI は、Server 1 の **c=us,ou=people,dc=example,dc=com** の UID を持つクライアントが、サーバーの **l=Zanzibar,c=africa,ou=people,dc=example,dc=com** 接尾辞ツリーであらゆる種類の操作を実行できます。Server 2 に、サーバー 3 に追加の権限を必要とする別の接尾辞にユーザーがある場合は、Server 2 に追加のクライアント ACI を追加する必要がある場合があります。

2.4.6.3. サーバーの 3 つの設定

1. Server 2 をプロキシー承認に使用する 3 つのサーバーで管理ユーザーを作成します。

```
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
```

2. 次に、同じローカルプロキシー承認 ACI を Server 2 にある 3 つのサーバーに追加します。以下のプロキシー承認 ACI を **l=Zanzibar,ou=people,dc=example,dc=com** エントリーに追加します。

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

この ACI は、**l=Zanzibar,ou=people,dc=example,dc=com** サブツリー内に含まれるデータに、Server 2 プロキシー管理者にのみ読み取り専用アクセスできるようにします。

3. 元のクライアントアプリケーションに対応する **l=Zanzibar,ou=people,dc=example,dc=com** サブツリーにローカルクライアント ACI を作成します。Server 2 でクライアントに作成されたものと同じ ACI を使用します。

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

これで、カスケード連鎖は設定されました。このカスケード構成により、ユーザーは Server 1 にバインドし、Server 3 の **l=Zanzibar,c=africa,ou=people,dc=example,dc=com** ブランチの情報を変更できます。セキュリティーのニーズによっては、より詳細なアクセス制御を提供する必要がある場合があります。

2.5. 参照の使用

リファーラルは、特定の情報についてどのサーバーに接続するかをクライアントアプリケーションに通知します。このリダイレクトは、クライアントアプリケーションが、ローカルサーバーに存在しないディレクトリーエントリーを要求するか、メンテナンスのためにデータベースがオフラインになったときに発生します。本セクションでは、リファーラルに関する以下の情報を提供します。

- 「リファーラルモードでのサーバーの起動」
- 「デフォルト参照の設定」
- 「スマートリファーラルの作成」
- 「バグ修正参照の作成」

ディレクトリーでリファーラル部分を使用する方法に関する概念情報は、『Red Hat Directory Server デプロイメントガイド』 を参照してください。

2.5.1. リファーラルモードでのサーバーの起動

リファーラルは、現在のサーバーが利用できない場合や、クライアントが現在のサーバーに保持されな

い情報を要求する場合に、クライアントアプリケーションを別のサーバーにリダイレクトするために使用されます。 たとえば、Directory Server の設定変更がある間に Directory Server を起動すると、そのサーバーが利用できない場合にすべてのクライアントを別のサプライヤーに参照します。リファーラルモードで Directory Server を起動するには、**refer** コマンドを使用します。

refer オプションを指定して **nsslapd** を実行します。

```
# ns-slapd refer -D /etc/dirsrv/slapd-instance_name [-p port] -r referral_url
```

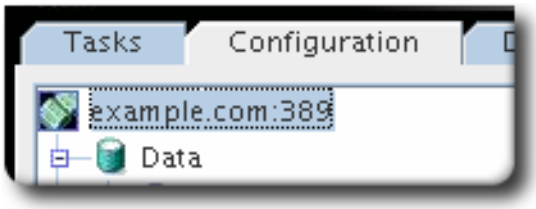
- **/etc/dirsrv/slapd-*instance_name*/** は、Directory Server 設定ファイルがあるディレクトリーです。これは、Red Hat Enterprise Linux 7 上のデフォルトの場所です。
- **port** は、参照モードで開始する Directory Server のオプションのポート番号です。
- **referral_url** は、クライアントに返される参照先です。LDAP URL の形式は、「付録C *LDAP URL*」を参照してください。

2.5.2. デフォルト参照の設定

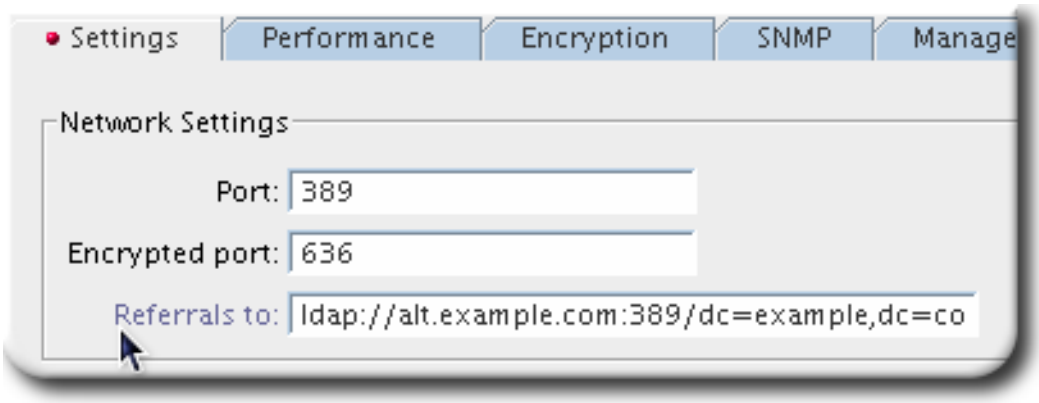
デフォルトの参照は、ディレクトリーが維持する接尾辞内に含まれていない DN に操作を送信するクライアントアプリケーションに返されます。以下の手順では、コンソールおよびコマンドラインユーティリティーを使用して、ディレクトリーのデフォルトリファーラルを設定する方法を説明します。

2.5.2.1. コンソールを使用したデフォルトのリファーラルの設定

1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のペインで、ナビゲーションツリーでトップエントリーを選択します。



3. 右側のペインで **Settings** タブを選択します。
4. 参照の LDAP URL を入力します。



複数の参照 URL をスペースで区切って入力します。

```
"ldap://dir1.example.com:389/dc=example,dc=com" "ldap://dir2.example.com/"
```

LDAP URL の詳細は、付録C *LDAP URL* を参照してください。

2.5.2.2. コマンドラインからのデフォルトリファーラルの設定

ldapmodify は、ディレクトリーの設定ファイルの **cn=config** エントリーにデフォルトの参照を追加できます。たとえば、1 つの Directory Server の **dir1.example.com から dir2.example.com** という名前のサーバーに新しいデフォルトリファーラルを追加するには、新しい行を **cn=config** エントリーに追加します。

1. **ldapmodify** ユーティリティーを実行し、デフォルトの参照を **dir2.example.com** サーバーに追加します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=config
changetype: modify
replace: nsslapd-referral
nsslapd-referral: ldap://dir2.example.com/
```

ディレクトリーの **cn=config** エントリーにデフォルトの参照を追加した後、ディレクトリーはクライアントアプリケーションによるリクエストに対してデフォルトの参照を返します。Directory Server を再起動する必要はありません。

2.5.3. スマートリファーラルの作成

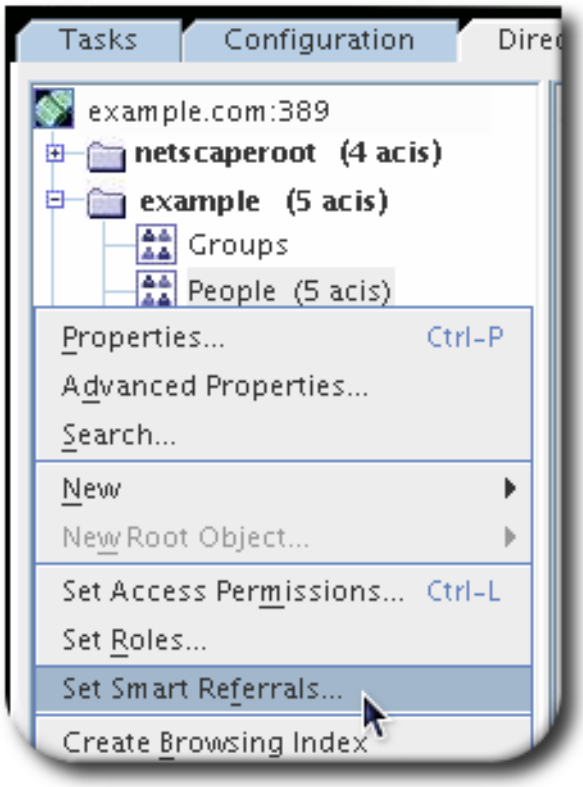
スマートリファーラルは、ディレクトリーエントリーまたはディレクトリーツリーを特定の LDAP URL にマッピングします。スマートリファーラルを使用すると、クライアントアプリケーションは特定のサーバーまたは特定のサーバーの特定のエントリーを参照できます。

たとえば、クライアントアプリケーションは、ディレクトリーエントリー **uid=jdoe,ou=people,dc=example,dc=com** を要求します。サーバー **directory.europe.example.com** のエントリー **cn=john doe,o=people,l=europe,dc=example,dc=com** を参照するクライアントにスマートリファーラルが返されます。

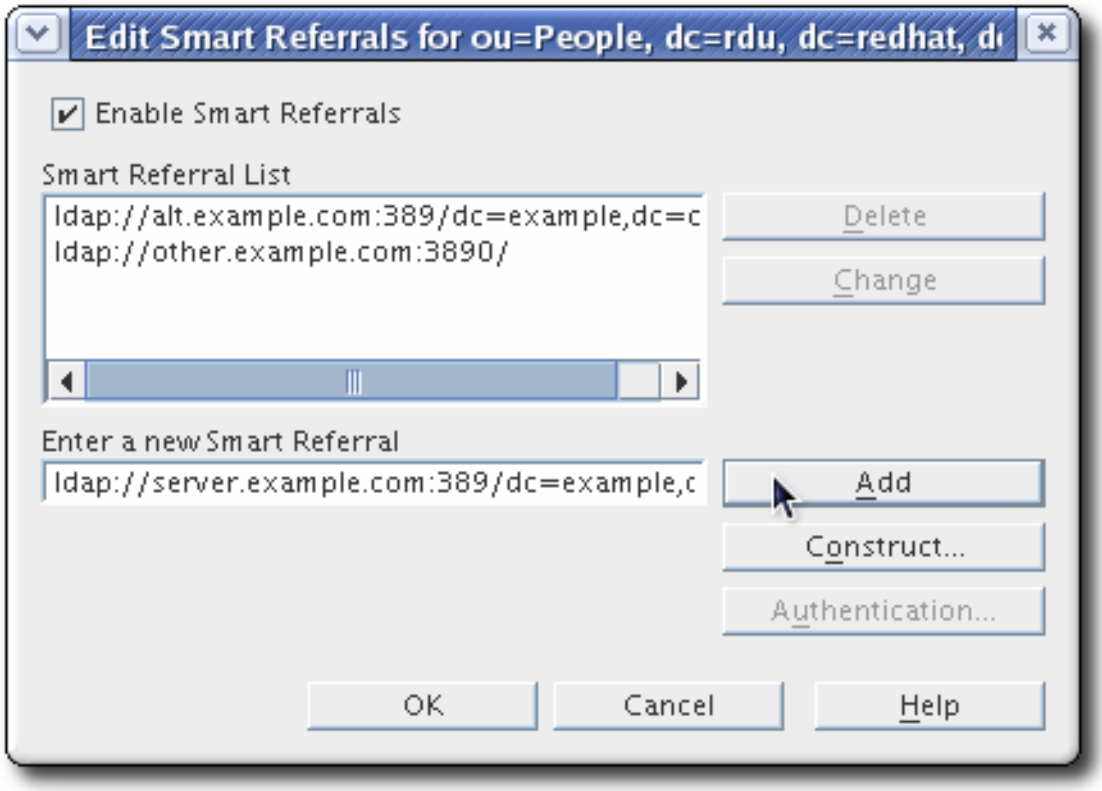
ディレクトリーがスマートリファーラルを使用する方法は、RFC 2251 セクション 4.1.11 で指定された標準仕様に準拠します。RFC は、http://www.ietf.org/rfc/rfc2251.txt でダウンロードできます。

2.5.3.1. Directory Server コンソールを使用したスマートリファーラルの作成

1. Directory Server コンソールで、Directory **タブを選択します**。
2. 左側のナビゲーションペインでツリーを参照して、参照を追加するエントリーを選択します。
3. エントリーを右クリックし、**Set Smart Referrals** を選択します。



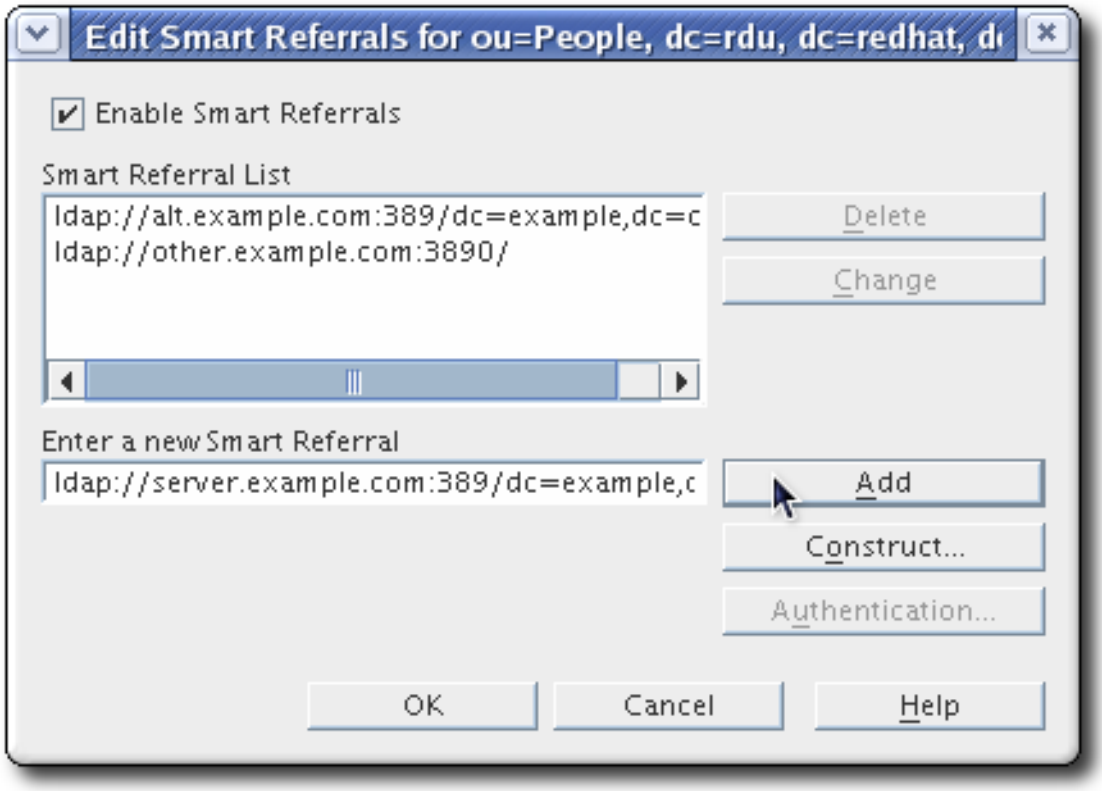
4. **Enable Smart Referral** チェックボックスを選択します。（オプションにチェックを入れると、エントリーからすべてのスマート参照を削除し、エントリーから **参照** オブジェクトクラスを削除します。）



5. **Enter a new Smart Referral** フィールドに、LDAP URL 形式でリファーラルを入力し、 **Add** をクリックします。LDAP URL は以下の形式である必要があります。

```
ldap://server:port/[optional_dn]
```

Server は、**サーバーのホスト名**、IPv4 アドレス、または IPv6 アドレスになります。**optional_dn** は、サーバーが要求するクライアントアプリケーションに戻るための明示的な DN です。

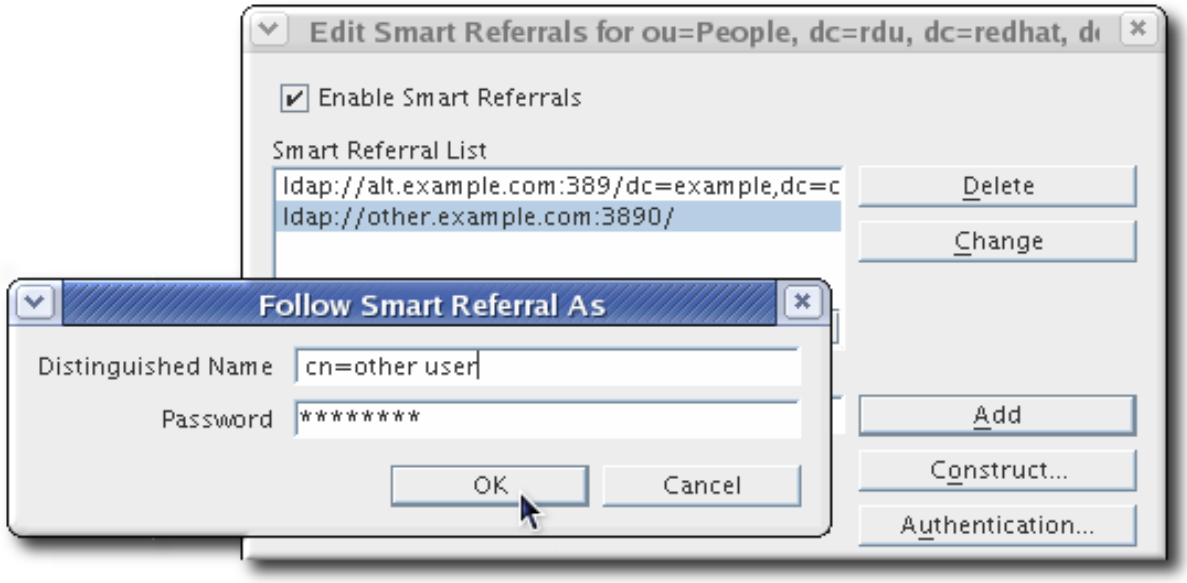


構成は、参照の追加プロセスを指示するウィザードが開きます。

スマート参照 **リスト は、選択したエントリーに対して現在参照情報を一覧表示します**。参照のリスト全体が、**すべてのオペレーションの場合は Return Referrals リクエストに対応してクライアントアプリケーションに返されます。また、Suffix Settings** タブの **Update Operations オプションの場合は Return Referrals** が返されます。これは **Configuration** タブで利用できます。

一覧を変更するには、**Edit** をクリックして選択したリファーラルを編集するか、**Delete** をクリックして選択した参照を削除します。

6. 別の認証情報を使用するように参照を設定するには、Authentication をクリックし 、 適切な DN とパスワードを指定します。この認証は、コンソールが閉じられるまでのみ有効です。その後、コンソールへのログインに使用されるのと同じ認証にリセットされます。



2.5.3.2. コマンドラインからのスマートリファーラルの作成

ldapmodify コマンドラインユーティリティーを使用して、コマンドラインからスマートリファーラルを作成します。

スマートリファーラルを作成するには、関連するディレクトリーエントリーを作成し、**参照** 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は単一の属性 ***ref*** を許可します。***ref*** 属性には LDAP URL が含まれている必要があります。

たとえば、以下を追加して、既存のエントリー **uid=jdoe** のスマートリファーラルを返します。

```
dn: uid=jdoe,ou=people,dc=example,dc=com
objectclass: referral
ref: ldap://directory.europe.example.com/cn=john%20doe,ou=people,l=europe,dc=example,dc=com
```

注記

LDAP URL のスペースに続く情報は、サーバーで無視されます。このため、参照として使用する LDAP URL 内の領域の代わりに **%20** を使用します。

directory.europe .example.com にリファーラルを持つ uid=jdoe,ou=people,dc= example,dc=com エントリーを追加するには、インポート前に LDIF ファイルに以下を追加します。

```
dn: uid=jdoe,ou=people,dc=example,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
objectclass: referral
cn: john doe
sn: doe
uid: jdoe
ref: ldap://directory.europe.example.com/cn=john%20doe,ou=people,l=europe,dc=example,dc=com
```

DN パスにすでにリファーラルがある場合は、**ldapmodify** で **-M** オプションを使用します。スマートリファーラルの詳細は、『『Red Hat Directory Server デプロイメントガイド』を参照してください』。

2.5.4. バグ修正参照の作成

以下の手順では、**接尾辞** にリファーラルを作成する方法を説明します。これは、接尾辞のプロセスがデータベースまたはデータベースリンクではなく、リファーラルを使用して処理されることを意味します。

警告

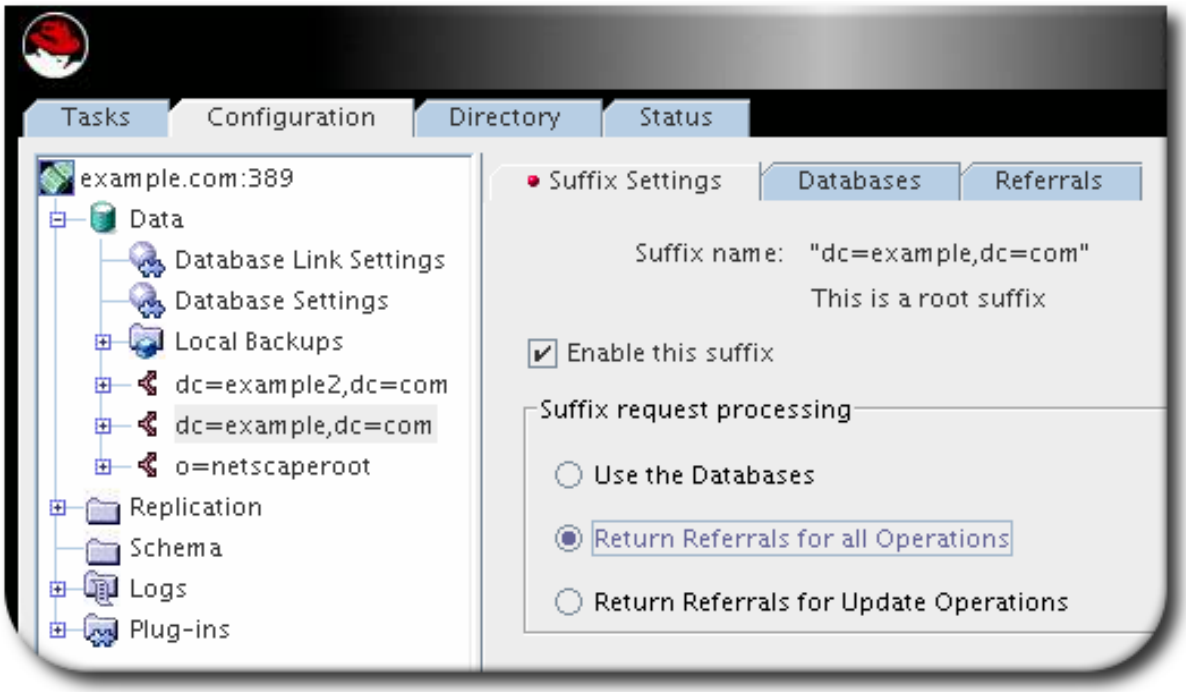
リファーラルを返すように接尾辞を設定すると、接尾辞に関連付けられたデータベースに含まれる ACI は無視されます。

2.5.4.1. コンソールを使用した接尾辞リファーラルの作成

参照を使用して、クライアントアプリケーションを別のサーバーに一時的にポイントできます。たとえば、接尾辞にリファーラルを追加して、接尾辞と関連したデータベースを、Directory Server データベースのユーザーに影響を与えずに、メンテナンスのために、接尾辞に関連付けられたデータベースをオフにできます。

接尾辞でリファーラルを設定するには、以下を実行します。

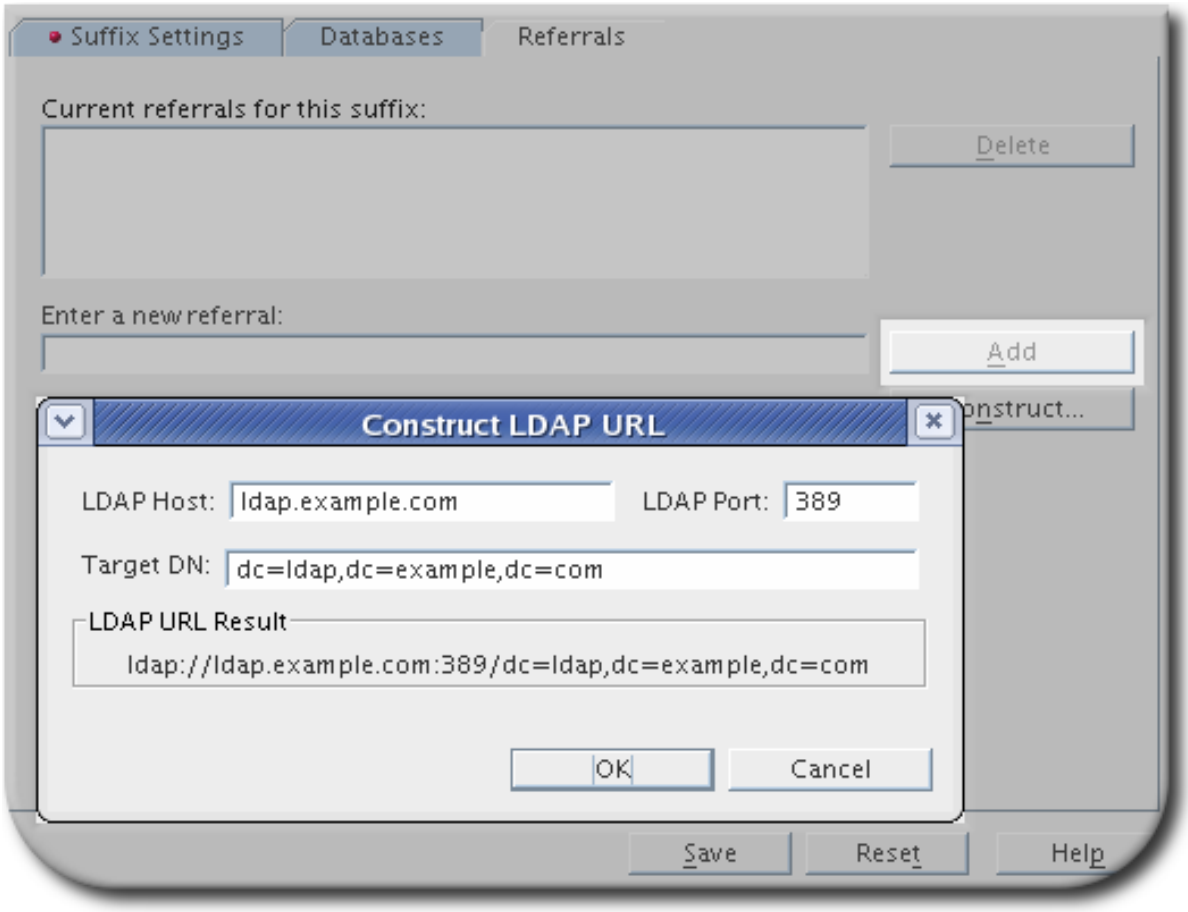
1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のペインの **Data** で、参照を追加する接尾辞を選択します。
3. **Suffix Settings** タブをクリックし、**Return Referrals for ... を選択します。operations** ラジオボタン。



Update Operations で Return Referrals を選択すると、ディレクトリーは更新および書き込みリクエストのみを読み取り専用データベースにリダイレクトします。たとえば、ディレクトリーデータのローカルコピーがあり、そのデータは検索に利用できますが、更新には利用でき

ないため、複数のサーバーに複製されます。Directory Server が更新要求に対してのみ参照を有効にすると、クライアントがエントリーの更新を要求すると、クライアントはデータを所有するサーバー（変更要求を続行できる）と呼ばれます。

4. **Referrals** タブをクリックします。LDAP URL を入力します。[1] **Enter a new referral** フィールドで、Construct をクリックして LDAP URL **を作成します**。



5. **Add** をクリックして、参照を一覧に追加します。

   複数の参照を入力できます。ディレクトリーは、クライアントアプリケーションからのリクエストに対応して参照のリスト全体を返します。

2.5.4.2. コマンドラインからの接尾辞リファーラルの作成

cn=mapping tree,cn=config ブランチ下のディレクトリー設定ファイルの root またはサブ接尾辞エントリーに接尾辞リファーラルを追加します。

ldapmodify を実行し、接尾辞リファーラルを **ou=people,dc=example,dc=com** root 接尾辞に追加します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=ou=people,dc=example,dc=com,cn=mapping tree,cn=config
changetype: add
objectclass: extensibleObject
objectclass: nsMappingTree
nsslapd-state: referral
nsslapd-referral: ldap://zanzibar.com/
```

nsslapd-state 属性は reference に設定されます。つまり 、 この接尾辞に作成されたリクエストに対して参照が返されます。***nsslapd-referral*** 属性には、接尾辞によって返される参照の LDAP URL が含まれます（この場合は、za **nzibar.com サーバーへの参照** ）。

nsslapd-state 属性は、**更新時に参照に** 設定することもできます。つまり、更新リクエスト以外のすべての操作に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更新時に **参照セットに設定された接尾辞の更新リクエストを行うと、** クライアントは参照を受け取ります。

接尾辞設定属性の詳細は、「コマンドラインで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。

[] 標準の LDAP URL 形式とは異なり、リモートサーバーの URL は接尾辞を指定しません。これには **ldap://server:port/** の形式があり、**server** にはホスト名、IPv4 アドレス、または IPv6 アドレスを指定できます。

[1] 付録C *LDAP URL* には、LDAP URL の構造に関する詳細が記載されています。

第3章 ディレクトリーエントリーの管理

本章では、Directory Server Console および **ldapmodify** および **ldapdelete** コマンドラインユーティリティーを使用して、ディレクトリーの内容を変更する方法を説明します。

Active Directory に保存されているエントリーは、Windows Sync を介して Directory Server に追加できます。Windows User Sync で同期されたエントリーを追加または変更する方法は、16章*Red Hat Directory Server と Microsoft Active Directory の同期* を参照してください。

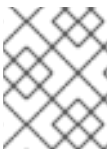
3.1. コマンドラインでエントリーの管理

コマンドラインを使用して LDAP 操作を実行するには、openldap-clients パッケージをインストールします。このパッケージによりインストールされるユーティリティーを使用すると、以下が可能になります。

- 新規エントリーの追加
- 既存のエントリーへの新規属性の追加
- 既存のエントリーおよび属性の更新
- エントリーからエントリーおよび属性を削除します
- 一括操作の実行

openldap-clients パッケージをインストールするには、以下を実行します。

```
# yum install openldap-clients
```



注記

LDAP 操作を実行するには、適切なパーミッションが必要です。アクセス制御の詳細は、「18章 *アクセス制御の管理*」を参照してください。

3.1.1. ldapadd、ldapmodify、および ldapdelete ユーティリティーへの入力の提供

ディレクトリー内のエントリーまたは属性を追加、更新、または削除する場合は、ユーティリティーのインタラクティブモードを使用して LDAP データ交換形式 (LDIF) ステートメントを入力するか、LDIF ファイルをこれらのファイルに渡します。

LDIF の詳細は、「LDIF ファイルの形式の概要」を参照してください。

3.1.1.1. インタラクティブモードでの入力の提供

インタラクティブモードでは、**ldapadd**、**ldapmodify**、および **ldapdelete** ユーティリティーはコマンドラインから入力を読み取ります。インタラクティブモードを終了するには、**Ctrl**+**D** (**^D**) のキーの組み合わせを押して End Of File (EOF) エスケープシーケンスを送信します。

インタラクティブモードでは、ユーティリティーは、**Enter** を 2 回押したときに、または EOF シーケンスを送信するときに、ステートメントを LDAP サーバーに送信します。

対話型モードを使用します。

- ファイルを作成せずに LDIF ステートメントに入るには、次を実行します。

例3.1 ldapmodify インタラクティブモードを使用した LDIF ステートメントの入力

以下の例では、インタラクティブモードで **ldapmodify** を開始し、***telephoneNumber*** 属性を削除し、**cn=*manager_name*,ou=people,dc=example,dc=com** 値を持つ manager 属性を **uid=*user*,ou=people,dc=example,dc=com** エントリーに追加します。最後のステートメントの後に **Ctrl**+**D** を押して、インタラクティブモードを終了します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: uid=user,ou=people,dc=example,dc=com
changetype: modify
delete: telephoneNumber
-
add: manager
manager: cn=manager_name,ou=people,dc=example,dc=com
^D
```

- 別のコマンドで出力される LDIF ステートメントを Directory Server にリダイレクトするには、次を実行します。

例3.2 リダイレクトされたコンテンツでの ldapmodify インタラクティブモードの使用

以下の例では、**command_that_outputs_LDIF** コマンドの出力を **ldapmodify** にリダイレクトします。対話モードは、リダイレクトされたコマンドの終了後に自動的に終了します。

```
# command_that_outputs_LDIF | ldapmodify -D "cn=Directory Manager" \
    -W -p 389 -h server.example.com -x
```

3.1.1.2. LDIF ファイルを使用した入力の提供

インタラクティブモードでは、**ldapadd**、**ldapmodify**、および **ldapdelete** ユーティリティーは、ファイルから LDIF ステートメントを読み取ります。このモードを使用して、多数の LDIF ステートメントを Directory Server に送信します。

例3.3 LDIF ステートメントを持つファイルを ldapmodify に渡す

1. LDIF ステートメントでファイルを作成します。たとえば、以下のステートメントで ~/**example.ldif** ファイルを作成します。

```
dn: uid=user,ou=people,dc=example,dc=com
changetype: modify
delete: telephoneNumber
-
add: manager
manager: cn=manager_name,ou=people,dc=example,dc=com
```

この例では、***telephoneNumber*** 属性を削除し、**cn=*manager_name*,ou=people,dc=example,dc=com** 値を持つ manager 属性を **uid=*user*,ou=people,dc=example,dc=com** エントリーに追加します。

2. **-f *file_name*** オプションを使用して、ファイルを **ldapmodify** コマンドに渡します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x \
    -f ~/example.ldif
```

3.1.2. 継続的操作モード

複数の LDIF ステートメントを Directory Server に送信し、1回の操作に失敗すると、プロセスは停止します。ただし、エラーが発生する前に処理されるエントリーは、正常に追加、変更、または削除されています。

エラーを無視してバッチでさらに LDIF ステートメントの処理を続けるには、**-c** オプションを **ldapadd** および **ldapmodify** に渡します。以下に例を示します。

```
# ldpamodify -c -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
```

3.1.3. エントリーの追加

新しいエントリーをディレクトリーに追加するには、**ldapadd** ユーティリティーまたは **ldapmodify** ユーティリティーを使用します。**ldapadd** は **/bin/ldapmodify** へのシンボリックリンクであることに注意してください。そのため、**ldapadd** は **ldapmodify -a** と同じ操作を実行します。

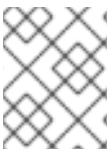
注記

親エントリーがすでに存在する場合のみ、新しいディレクトリーエントリーを追加できます。たとえば、 **ou=people,dc=example,dc=com** の親エントリーが存在しない場合は、**cn=user,ou=people,dc=example,dc=com** エントリーを追加できません。

3.1.3.1. ldapadd を使用したエントリーの追加

ldapadd ユーティリティーを使用して、たとえば **cn=user,ou=people,dc=example,dc=com** ユーザーエントリーを追加するには、以下を実行します。

```
# ldapadd -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: uid=user,ou=People,dc=example,dc=com
uid: user
givenName: given_name
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
sn: surname
cn: user
```



注記

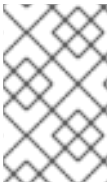
ldapadd を実行すると、**changetype: add** 操作が自動的に実行されます。そのため、LDIF ステートメントで **changetype: add** を指定する必要はありません。

コマンドで使用されるパラメーターの詳細は、ldapadd(1) の man ページを参照してください。

3.1.3.2. ldapmodify を使用したエントリーの追加

ldapmodify ユーティリティーを使用して、たとえば **cn=user,ou=people,dc=example,dc=com** ユーザーエントリーを追加するには、以下を実行します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: uid=user,ou=People,dc=example,dc=com
uid: user
givenName: given_name
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
sn: surname
cn: user
```



注記

-a オプションを **ldapmodify** コマンドに渡すと、ユーティリティーは **changetype: add** 操作を自動的に実行します。そのため、LDIF ステートメントで **changetype: add** を指定する必要はありません。

コマンドで使用されるパラメーターの詳細は、ldapmodify(1) の man ページを参照してください。

3.1.3.3. ルートエントリーの作成

dc=example,dc=com などのデータベース接尾辞のルートエントリーを作成するには、**cn=Directory Manager** ユーザーとしてバインドし、エントリーを追加します。

DN は、データベースのルートまたは従属接尾辞の DN に対応します。

たとえば、**dc=example,dc=com** 接尾辞を追加するには、次のコマンドを実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: dc=example,dc=com
changetype: add
objectClass: top
objectClass: domain
dc: example
```

注記

ルートオブジェクトは、接尾辞に 1 つのデータベースがある場合にのみ追加できます。複数のデータベースに保存されている接尾辞を作成する場合は、**-n *back_end*** オプションを指定して **ldif2db** ユーティリティーを使用し、新しいエントリーを保持するデータベースを設定する必要があります。詳細は、「コマンドラインからのインポート」 を参照してください。

3.1.4. ディレクトリーエントリーの更新

ディレクトリーエントリーを変更する場合は、**changetype: modify** ステートメントを使用します。change 操作に応じて、エントリーから属性を追加、変更、または削除できます。

ldapmodify ユーティリティーを使用して、LDIF ステートメントを Directory Server に送信します。たとえば、インタラクティブモードでは、以下のようになります。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
```

ldapmodify コマンドで使用されるパラメーターの詳細は、ldapmodify(1) の man ページを参照してください。

3.1.4.1. エントリーへの属性の追加

エントリーに属性を追加するには、**add** 操作を使用します。

たとえば、**555-1234567** の値を持つ ***telephoneNumber*** 属性を **uid=*user*,dc=people,dc=example,dc=com** エントリーに追加するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: uid=user,dc=people,dc=example,dc=com
changetype: modify
add: telephoneNumber
telephoneNumber: 555-1234567
```

属性が多値である場合、属性名を複数回指定して、1 つの操作ですべての値を追加できます。たとえば、**uid=*user*,dc=people,dc=example,dc=com** に 2 つの ***telephoneNumber*** 属性を同時に追加するには、以下のコマンドを実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: uid=user,dc=people,dc=example,dc=com
changetype: modify
add: telephoneNumber
telephoneNumber: 555-1234567
telephoneNumber: 555-7654321
```

3.1.4.2. 属性の値の更新

属性の値を更新する手順は、属性が単値であるか多値であるかによって異なります。

単値属性の更新
単値属性を更新する場合は、**replace** 操作を使用して既存の値を上書きします。以下のコマンドは、**uid=*user*,dc=people,dc=example,dc=com** エントリーの ***manager*** 属性を更新します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: uid=user,dc=people,dc=example,dc=com
changetype: modify
replace: manager
manager: uid=manager_name,dc=people,dc=example,dc=com
```

多値属性の特定値の更新

多値属性の特定の値を更新するには、最初に置き換えるエントリーを削除してから、新しい値を追加する必要があります。以下のコマンドは、**uid=*user*,dc=people,dc=example,dc=com** エントリーの現在 **555-1234567** に設定されている ***telephoneNumber*** 属性だけを更新します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: uid=user,dc=people,dc=example,dc=com
changetype: modify
delete: telephoneNumber
telephoneNumber: 555-1234567
-
add: telephoneNumber
telephoneNumber: 555-9876543
```

3.1.4.3. エントリーからの属性の削除

エントリーから属性を削除するには、**delete** 操作を実行します。

属性の削除
たとえば、**uid=*user*,dc=people,dc=example,dc=com** エントリーから ***manager*** 属性を削除するには、以下のコマンドを実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: uid=user,dc=people,dc=example,dc=com
changetype: modify
delete: manager
```

注記

属性に複数の値が含まれる場合、この操作によりすべての値が削除されます。

多値属性から特定の値の削除
多値属性から特定の値を削除する場合は、LDIF ステートメントに属性とその値を一覧表示します。たとえば、**uid=*user*,dc=people,dc=example,dc=com** エントリーから **555-1234567** に設定されている ***telephoneNumber*** 属性だけを削除するには、以下のコマンドを実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: uid=user,dc=people,dc=example,dc=com
changetype: modify
delete: telephoneNumber
telephoneNumber: 555-1234567
```

3.1.5. エントリーの削除

エントリーを削除すると、ディレクトリーからエントリーが削除されます。

注記

子エントリーのないエントリーのみを削除できます。たとえば、**uid=*user*,ou=People,dc=example,dc=com** エントリーがまだ存在している場合は、**ou=People,dc=example,dc=com** エントリーを削除できません。

3.1.5.1. ldapdelete を使用したエントリーの削除

ldapdelete ユーティリティーを使用すると、1 つまたは複数のエントリーを削除できます。たとえば、**uid=*user*,ou=People,dc=example,dc=com** エントリーを削除するには、次のコマンドを実行します。

```
# ldapdelete -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
"uid=user,ou=People,dc=example,dc=com"
```

1 つの操作で複数のエントリーを削除するには、コマンドに追加します。以下に例を示します。

```
# ldapdelete -D "cn=Directory Manager" -W -p 389 -h server.example.com -x \
    "uid=user1,ou=People,dc=example,dc=com" \
    "uid=user2,ou=People,dc=example,dc=com"
```

使用されるパラメーターの詳細は、ldapdelete(1) の man ページを参照してください。

3.1.5.2. ldapmodify を使用したエントリーの削除

ldapmodify ユーティリティーを使用してエントリーを削除するには、**changetype: delete** 操作を使用します。たとえば、**uid=*user*,ou=People,dc=example,dc=com** エントリーを削除するには、次のコマンドを実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: uid=user,dc=people,dc=example,dc=com
changetype: delete
```

3.1.6. エントリーの名前変更および変更

エントリーの名前変更時に、**ldapmodify** ユーティリティーを使用して LDIF ステートメントを Directory Server に送信します。たとえば、インタラクティブモードでは、以下のようになります。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
```

ldapmodify コマンドで使用されるパラメーターの詳細は、ldapmodify(1) の man ページを参照してください。

注記

moddn アクセス制御リスト (ACL) を使用して、エントリーを移動するパーミッションを付与します。詳細は 「ソースおよび宛先 DN のターゲット」を参照してください。

3.1.6.1. 名前変更操作のタイプ

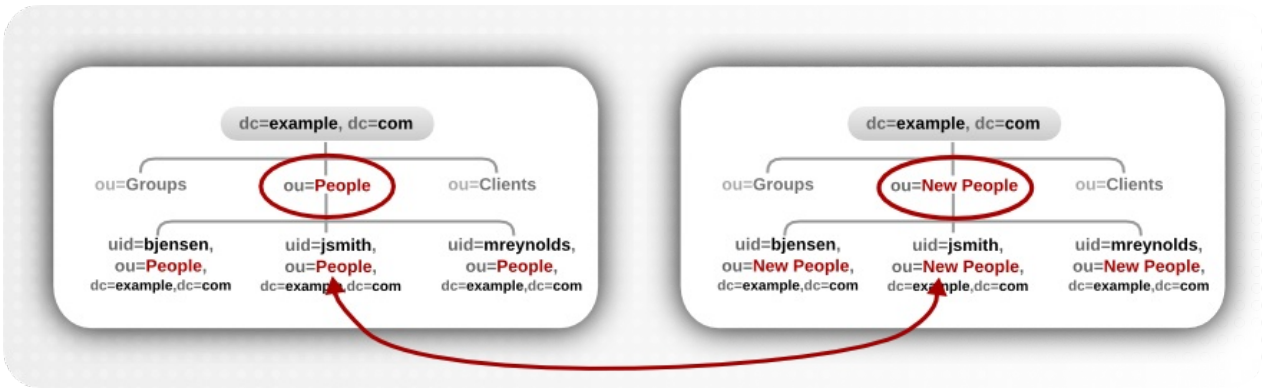
以下の名前変更操作が存在します。

エントリーの名前変更

エントリーの名前を変更すると、**modrdn** 操作はエントリーの RDN (Relative Distinguished Name) を変更します。

サブエントリーの名前変更

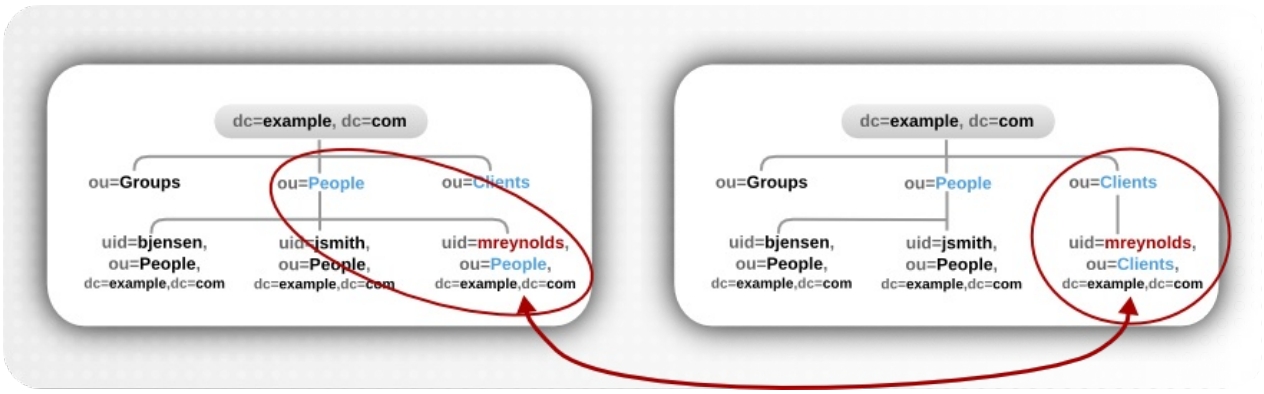
サブツリーエントリーの場合、**modrdn** 操作はサブツリーと子エントリーの DN コンポーネントの名前を変更します。



大規模なサブツリーでは、このプロセスに多くの時間とリソースが必要になる可能性があることに注意してください。

エントリーを新しい親へ移動

サブツリーの名前を変更する同様のアクションは、エントリーをあるサブツリーから別のサブツリーに移動することです。これは、**modrdn** 操作の拡張タイプで、エントリーの名前を同時に変更し、***newSuperior*** 属性を設定して、エントリーを別の親に移動します。



3.1.6.2. エントリーの名前変更に関する考慮事項

名前変更の操作を実行する場合は、以下の点に留意してください。

- root 接尾辞の名前を変更することはできません。
- サブツリー名前変更操作によるレプリケーションへの影響は最小限に抑えられます。レプリカ合意は、データベースのサブツリーではなく、データベース全体に適用されます。そのため、サブツリーの名前変更操作ではレプリカ合意の再設定は必要ありません。サブツリーの名前変更操作後のすべての名前の変更は、通常どおり複製されます。
- サブツリーの名前を変更し、同期合意を再設定する必要がある場合があります。同期合意は、接尾辞またはサブツリーレベルで設定されます。そのため、サブツリーの名前を変更すると、同期が中断する可能性があります。
- サブツリーの名前を変更するには、サブツリーに設定されたサブツリーレベルのアクセス制御命令 (ACI) を手動で再設定し、サブツリーの子エントリーに設定されたエントリーレベルの ACI (エントリーレベルの ACI) を手動で再設定する必要があります。
- ***ou*** から ***dc*** への移行など、サブツリーのコンポーネントを変更しようとすると、スキーマ違反で失敗する可能性があります。たとえば、**organizationalUnit** オブジェクトクラスには ***ou*** 属性が必要です。この属性がサブツリーの名前の一部として削除されると、操作は失敗します。
- グループを移動すると、MemberOf プラグインは ***memberOf*** 属性を自動的に更新します。ただし、グループが含まれるサブツリーを移動する場合は、**cn=memberof task** エントリーでタスクを手動で作成するか、**fixup-memberof.pl** を使用して関連する ***memberOf*** 属性を更新する必要があります。

  memberOf 属性参照のクリーンアップに関する詳細は、「memberOf 値の同期」 を参照してください。

3.1.6.3. *deleteOldRDN* パラメーター

エントリーの名前を変更すると、***deleteOldRDN*** パラメーターは古い RDN が削除されるかどうかを制御します。

***deleteOldRDN*: 0**

既存の RDN は、新しいエントリーの値として保持されます。生成されるエントリーには、古い属性と新しい共通名 (CN) を持つ 2 つの ***cn*** 属性が含まれます。

たとえば、以下の属性は、**deleteOldRDN: 0** パラメーターを設定して、**cn=old_group,dc=example,dc=com** から**cn=new_group,dc=example,dc=com** に名前を変更したグループに属しています。

```
dn: cn=new_group,ou=Groups,dc=example,dc=com
objectClass: top
objectClass: groupOfUniqueNames
cn: old_group
cn: new_group
```

***deleteOldRDN*: 1**

Directory Server は古いエントリーを削除し、新しい RDN を使用して新しいエントリーを作成します。新しいエントリーには、新しいエントリーの ***cn*** 属性のみが含まれます。

たとえば、以下のグループは、**deleteOldRDN: 1** パラメーターを設定して、**cn=new_group,dc=example,dc=com** に名前を変更しました。

```
dn: cn=new_group,ou=Groups,dc=example,dc=com
objectClass: top
objectClass: groupofuniquenames
cn: new_group
```

3.1.6.4. エントリーまたはサブツリーの名前変更

エントリーまたはサブツリーの名前変更には、**changetype: modrdn** 操作を使用し、***newrdn*** 属性に新しい RDN を設定します。

たとえば、**cn=old_group,ou=Groups,dc=example,dc=com エントリーの名前を cn=new_group,ou=Groups,dc=example,dc=com** に変更するには、次のコマンドを実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=old_group,ou=Groups,dc=example,dc=com
changetype: modrdn
newrdn: cn=new_group
deleteOldRDN: 1
```

deleteOldRDN の詳細は、「***deleteOldRDN*** パラメーター」を参照してください。

3.1.6.5. エントリーを新しい親へ移動

エントリーを新しい親に移動するには、**changetype: modrdn** 操作を使用して、以下の属性を設定します。

newrdn

移動したエントリーの RDN を設定します。RDN が同じままであっても、このエントリーを設定する必要があります。

newSuperior

新しい親エントリーの DN を設定します。

たとえば、**uid=*ユーザーエントリー*を ou= Engineering,ou=People,dc=example,dc=com から ou=Marketing,ou=People,dc=example,dc=com** に移動するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: uid=user,ou=Engineering,ou=People,dc=example,dc=com
changetype: modrdn
newrdn: uid=user
newSuperior= ou=Marketing,ou=People,dc=example,dc=com
deleteOldRDN: 1
```

deleteOldRDN の詳細は、「***deleteOldRDN*** パラメーター」を参照してください。

3.1.7. 特殊文字の使用

コマンドラインを使用する場合は、スペース ( )、アスタリスク (*)、バックスラッシュ (\) などのコマンドラインインタープリターに特別な意味を持つ文字を引用符で囲みます。コマンドラインインタープリターに応じて、一重引用符または二重引用符を使用します。

たとえば、**cn=Directory Manager** ユーザーとして認証するには、ユーザーの DN を引用符で囲みます。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
```

また、DN にコンポーネントのコンマが含まれる場合は、バックスラッシュを使用してエスケープします。たとえば、**uid=*user*,ou=People,dc=example.com Chicago, IL** ユーザーとして認証するには、次のコマンドを実行します。

```
# ldapmodify -a -D "cn=uid=user,ou=People,dc=example.com Chicago\, IL" \
    -W -p 389 -h server.example.com -x
```

3.1.8. Binary 属性の使用

特定の属性は、***jpegPhoto*** 属性などのバイナリー値をサポートします。このような属性を追加または更新すると、ユーティリティーはファイルから属性の値を読み取ります。このような属性を追加または更新するには、**ldapmodify** ユーティリティーを使用できます。

たとえば、**uid=*user*,ou=People,dc=example,dc=com** エントリーに ***jpegPhoto*** 属性を追加し、**/home/*user_name*/photo.jpg** ファイルから属性の値を読み取るには、次のコマンドを実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: uid=user,ou=People,dc=example,dc=com
changetype: modify
add: jpegPhoto
jpegPhoto:< file:///home/user_name/photo.jpg
```

重要

: と **<** の間には、スペースがないことに注意してください。

3.1.9. 国際化されたディレクトリーにおけるエントリーの更新

属性の値を英語以外の言語で使用するには、属性の値を言語タグに関連付けます。

ldapmodify を使用して言語タグが設定されている属性を更新する場合は、値と言語タグを正確に一致させる必要があります。そうでないと、操作は失敗します。

たとえば、**lang-fr** 言語タグが設定された属性値を変更するには、**modify** 操作にタグを追加します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: uid=user,ou=People,dc=example,dc=com
changetype: modify
replace: homePostalAddress;lang-fr
homePostalAddress;lang-fr: 34 rue de Seine
```

3.2. ディレクトリーコンソールを使用したエントリーの管理

Directory Server コンソールの **Directory** タブおよび **Property Editor** を使用して、エントリーを個別に追加、変更、または削除できます。

複数のエントリーを同時に追加するには、「コマンドラインでエントリーの管理」 で説明されているコマンドラインユーティリティーを使用します。

- 「ルートエントリーの作成」
- 「ディレクトリーエントリーの作成」
- 「ディレクトリーエントリーの変更」
- 「ディレクトリーエントリーの削除」

注記

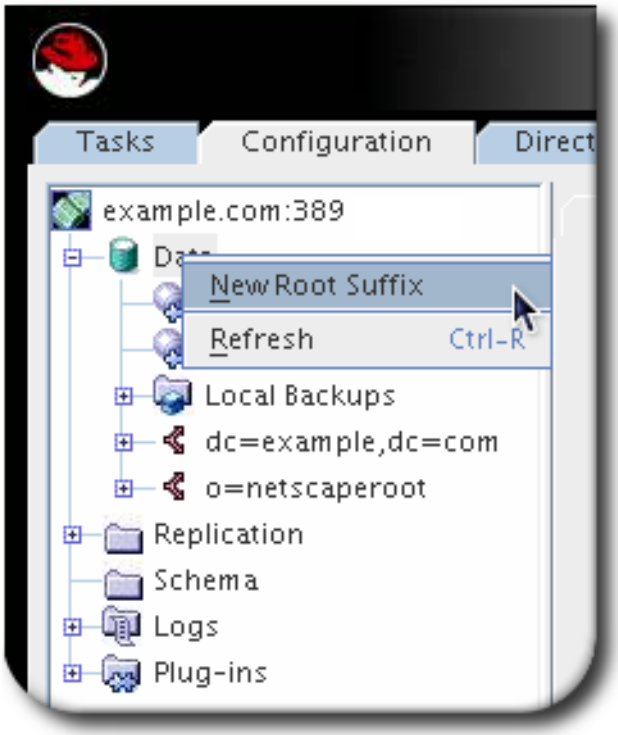
適切なアクセス制御ルールが設定されない限り、ディレクトリーは変更できません。ディレクトリーのアクセス制御ルールを作成する方法は、18章 *アクセス制御の管理* を参照してください。

3.2.1. ルートエントリーの作成

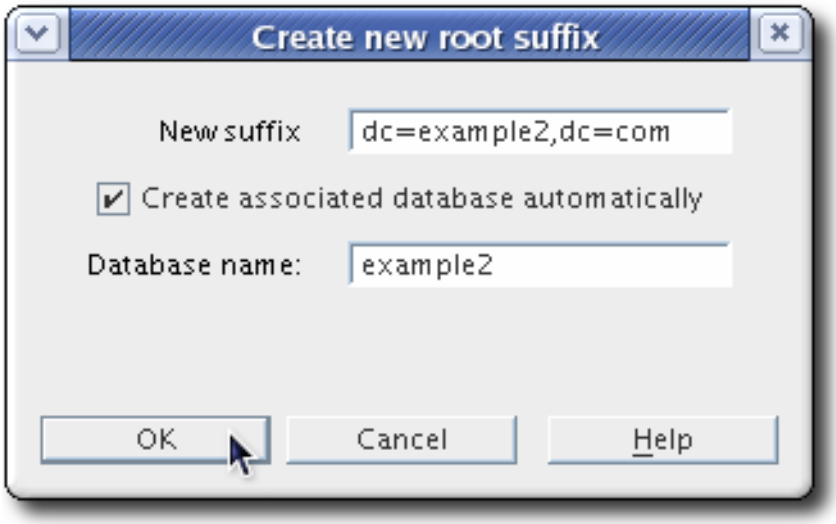
新しいデータベースが作成されるたびに、データベースに保存される接尾辞に関連付けられます。その接尾辞を表すディレクトリーエントリーは自動的に作成されません。

データベースの root エントリーを作成するには、以下を実行します。

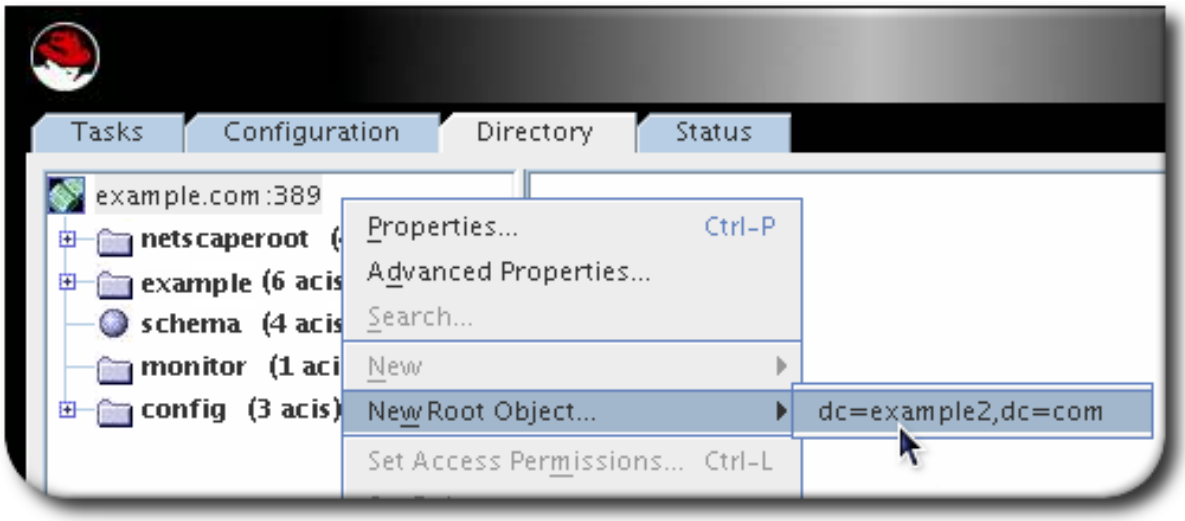
1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のメニューの **Data** エントリーを右クリックし、メニューから **New Root Suffix** を選択します。



3. 新しい接尾辞およびデータベース情報を入力します。

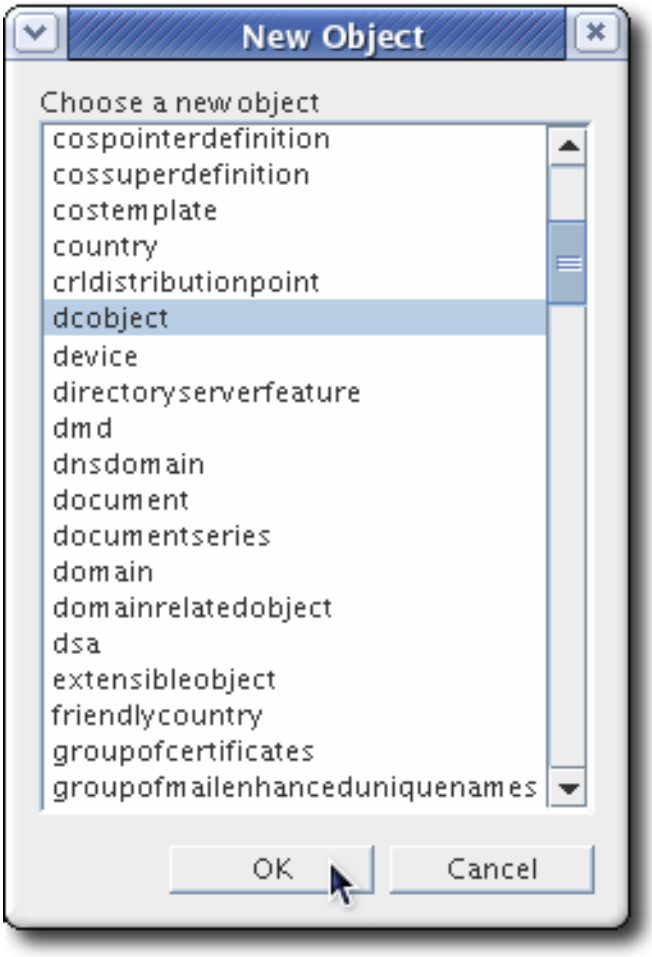


4. **Directory** タブで、Directory Server を表す最上位オブジェクトを右クリックし、**New Root Object** を選択します。



New Root Object のセカンダリーメニューは、対応するディレクトリーエントリーのない新しい接尾辞を表示します。作成するエントリーに対応する接尾辞を選択します。

5. **New Object** ウィンドウで、新規エントリーに対応するオブジェクトクラスを選択します。



オブジェクトクラスには、接尾辞に名前を付けるために使用される属性が含まれている必要があります。たとえば、エントリーが **ou=people,dc=example,dc=com** の接尾辞に対応している場合は、**または ou** 属性を許可する別のオブジェクトクラスを選択します。

6. New Object ウィンドウで **OK** をクリックします。

新しいエントリーの プロパティーエディターが開きます。「ディレクトリーエントリーの変更」 で説明されているように、**OK をクリックして現在の値** を使用するか、エントリーを変更します。

3.2.2. ディレクトリーエントリーの作成

Directory Server コンソールは、新しいディレクトリーエントリー用に事前に設定したフォームとともに事前定義されたテンプレートを提供します。表3.1「エントリーテンプレートと対応するオブジェクトクラス」 は、各テンプレートに使用されるオブジェクトクラスのタイプを表示します。

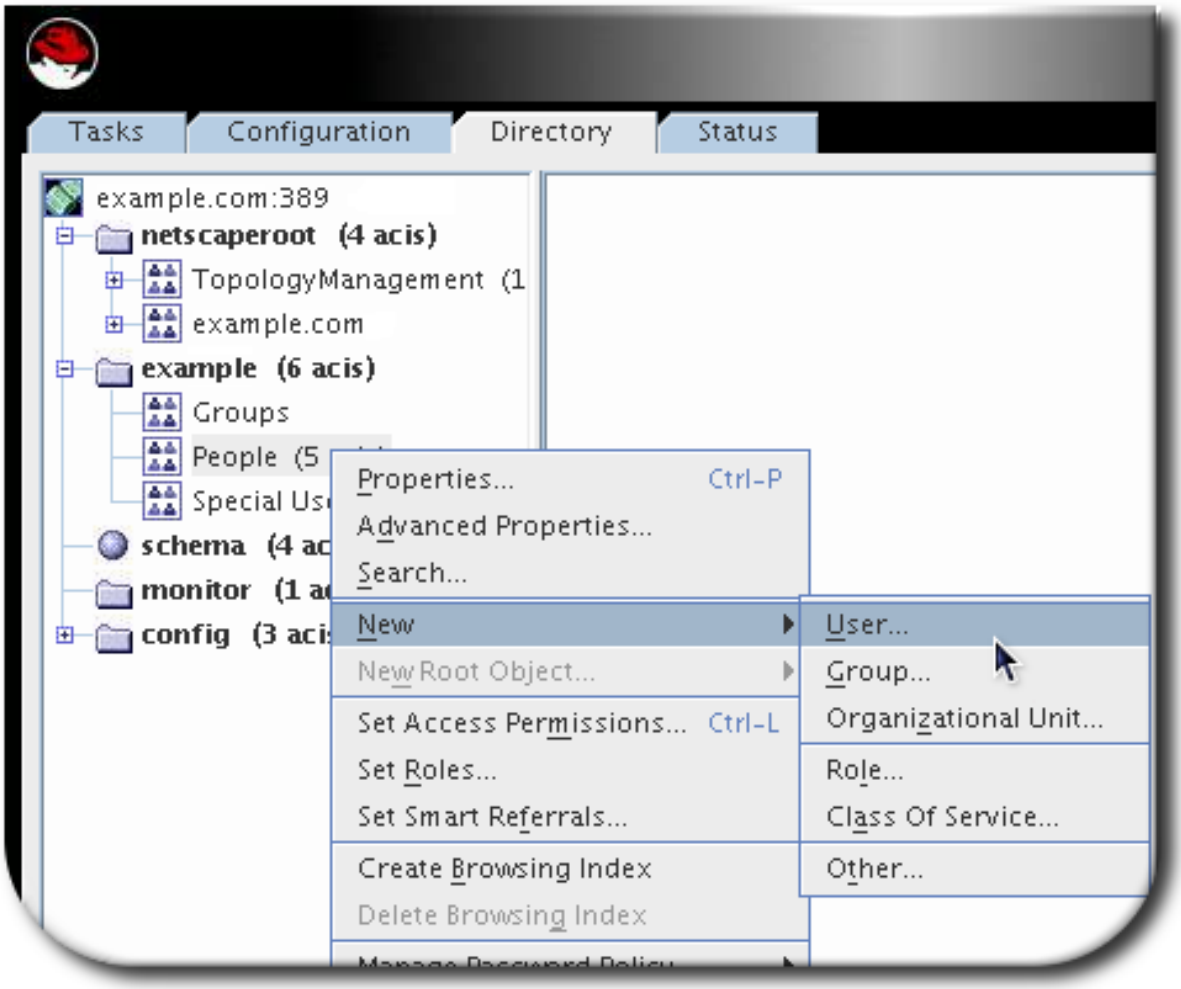
表3.1 エントリーテンプレートと対応するオブジェクトクラス

Template	オブジェクトクラス
ユーザー	inetOrgPerson
グループ	groupOfUniqueNames
組織単位	organizationalUnit
ロール	nsRoleDefinition

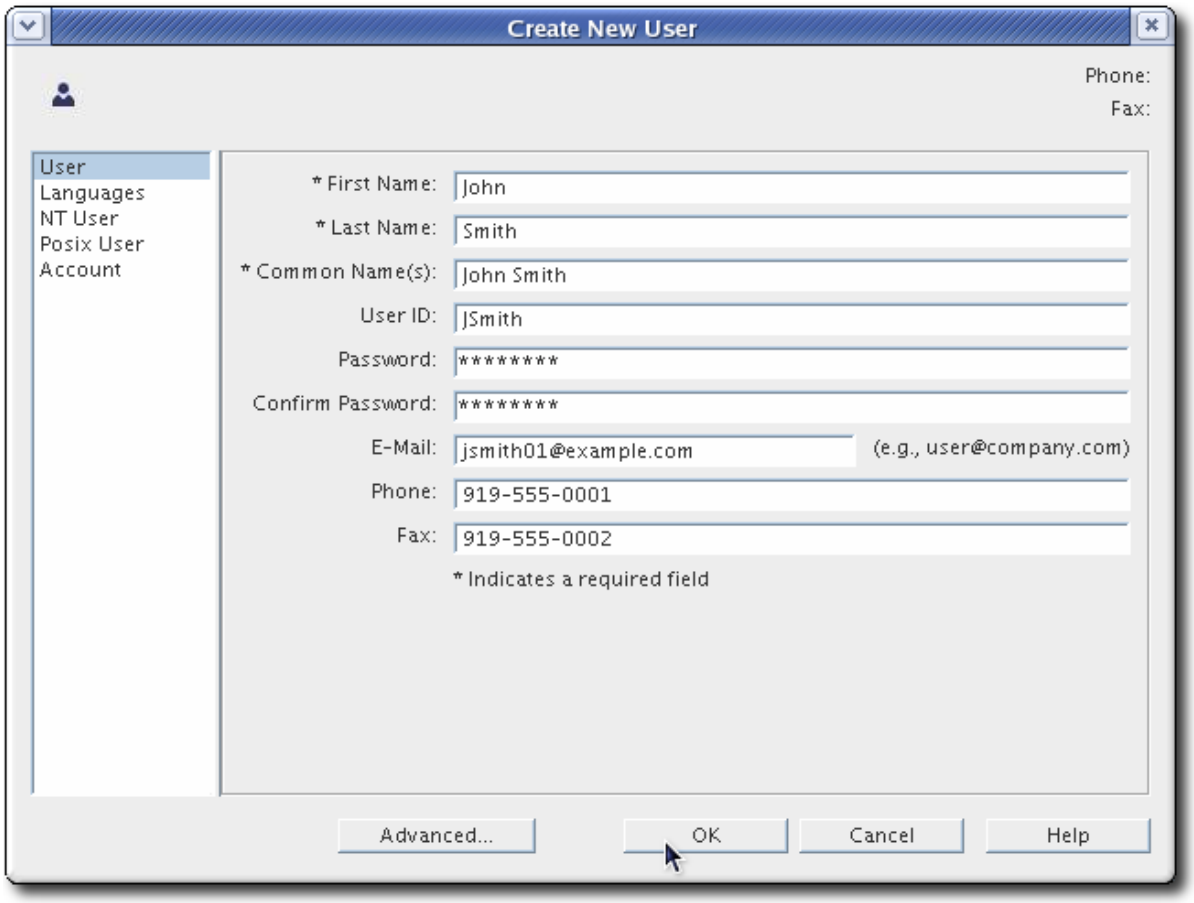
Template	オブジェクトクラス
サービスのクラス	cosSuperDefinition

別のタイプ **Other** では、ユーザーが特定のオブジェクトクラスおよび属性を適用できるようにして、あらゆる種類のエントリーを作成できます。

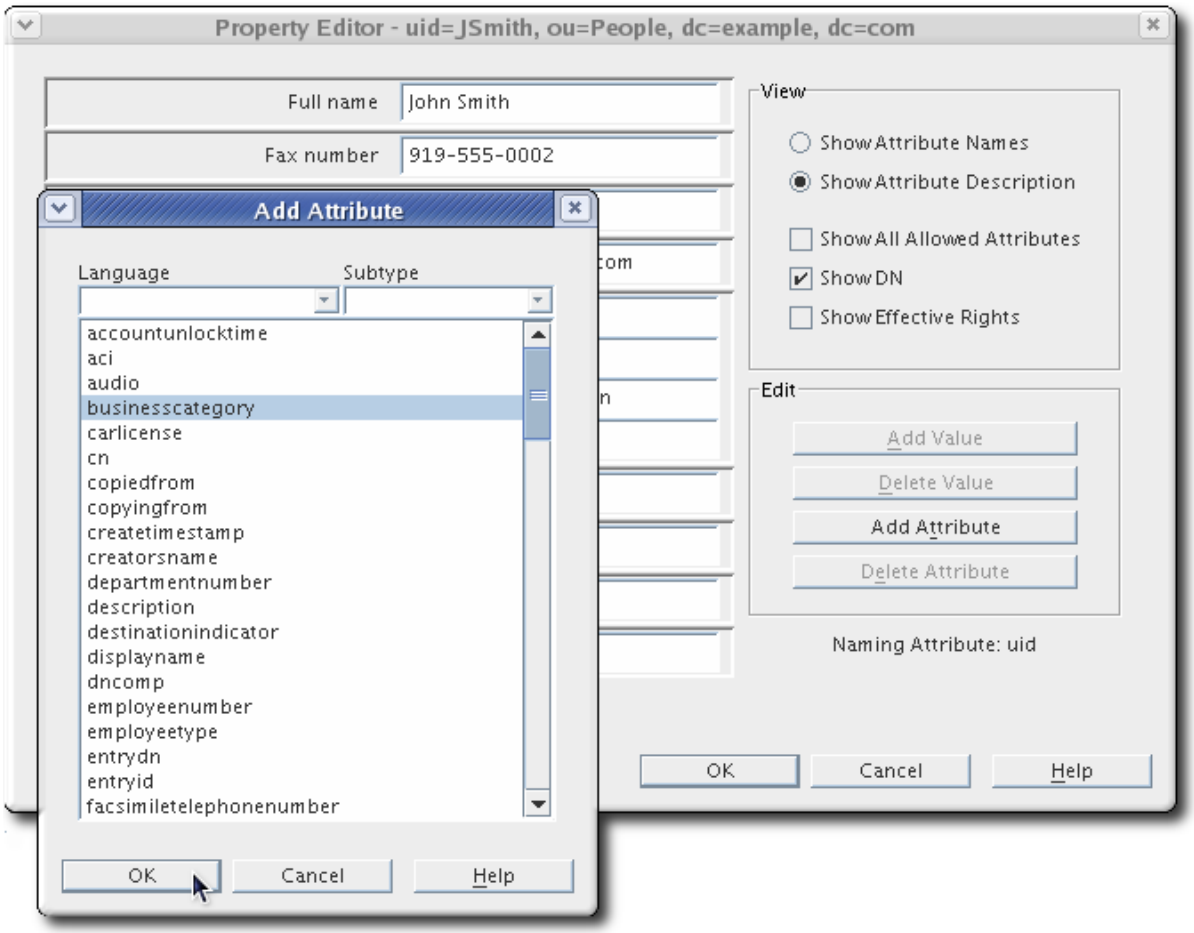
1. Directory Server コンソールで、Directory **タブを選択します。**
2. 左側のペインで、メインエントリーを右クリックして新しいエントリーを追加し、エントリーのタイプ（ **User**、**Group**、OrganizationalUnit、**Role**、**Class of Service**、または **Other** ）を選択します。



3. 新しいエントリータイプが **Other** の場合、オブジェクトクラスの一覧が開きます。一覧からオブジェクトクラスを選択して新規エントリーを定義します。
4. 一覧表示されているすべての属性の値を指定します。必要な属性にはアスタリスク(*)が付いています。

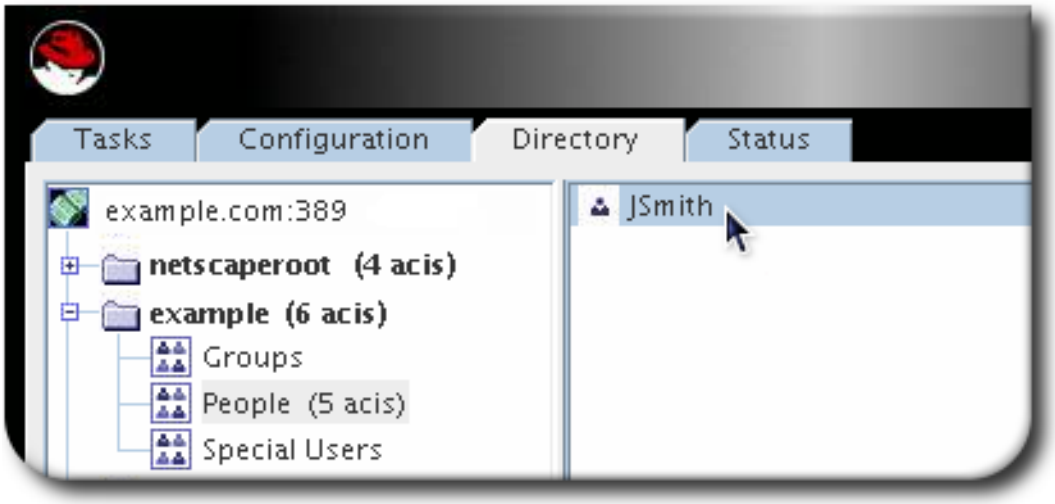


5. オブジェクトクラス（エントリータイプ）で利用可能な属性の詳細一覧を表示するには、**Advanced** ボタンをクリックします。



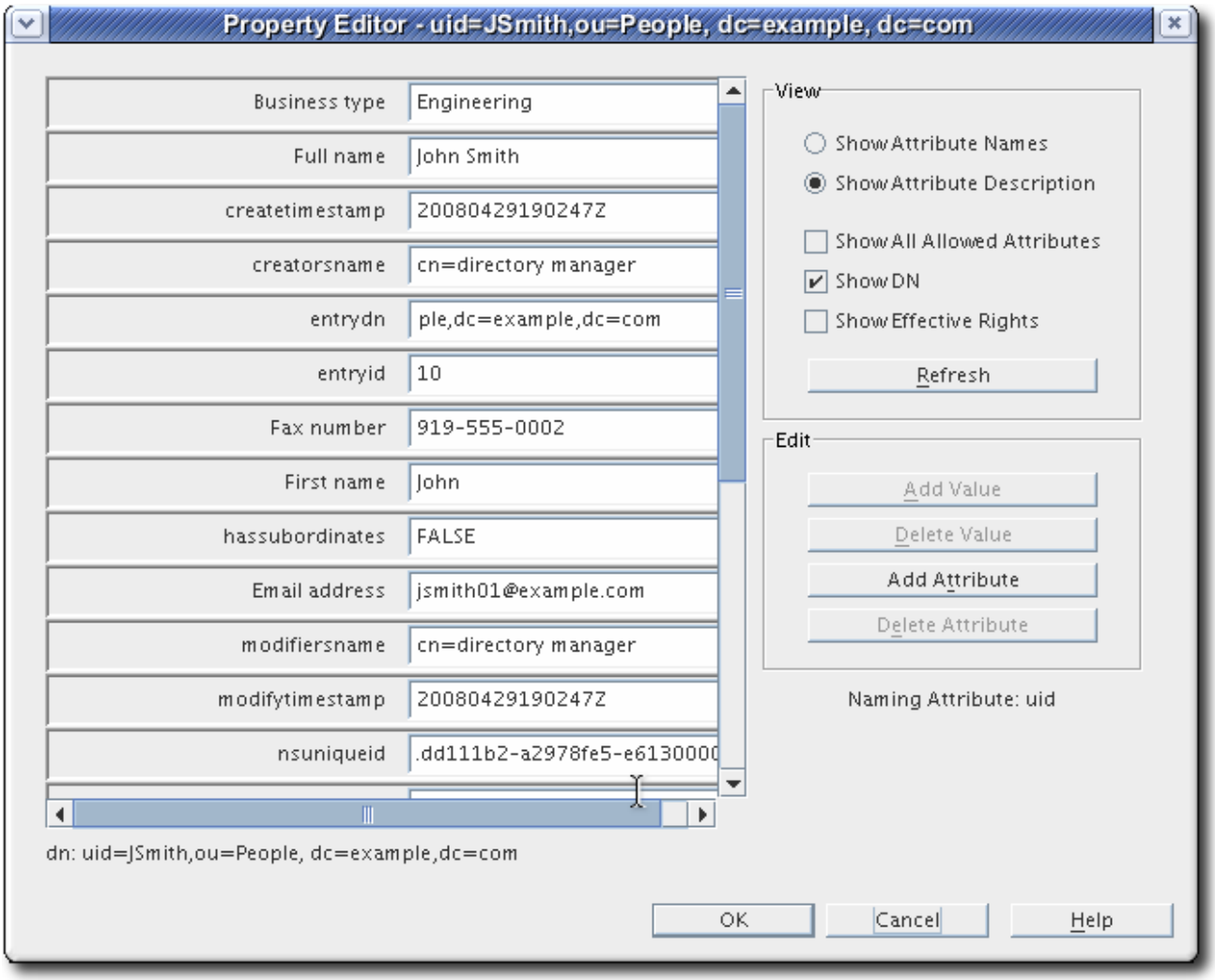
Property Editor で追加の属性を選択し、属性値を入力します。

6. **OK** をクリックしてエントリーを保存します。新しいエントリーが右側のペインに表示されます。



3.2.3. ディレクトリーエントリーの変更

Directory Server コンソールのディレクトリーエントリーを変更するには、**Property Editor** と呼ばれるダイアログウィンドウを使用します。**Property Editor** には、エントリーに属するオブジェクトクラスおよび属性のリストが含まれ、オブジェクトクラス、属性、属性値、および属性サブタイプを追加および削除することで、そのエントリーに属するオブジェクトクラスおよび属性の編集に使用できます。



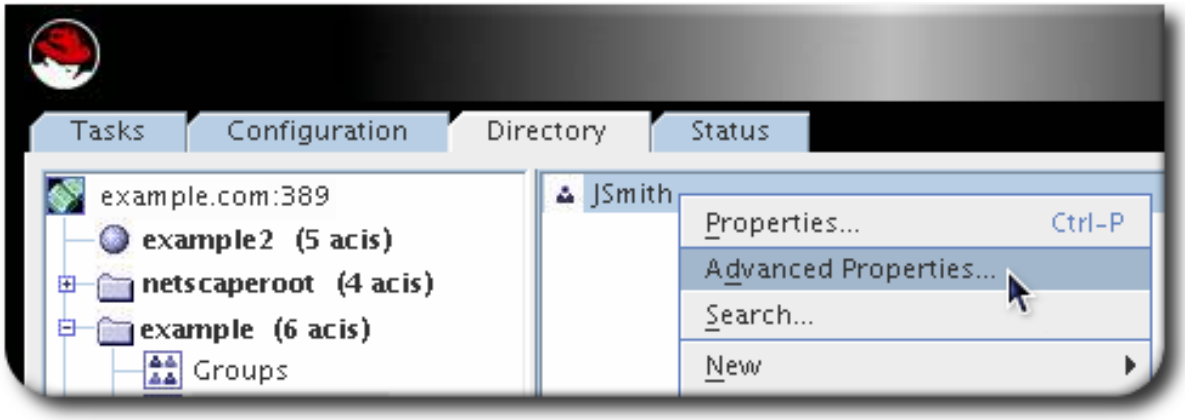
プロパティーエディターは、複数の方法で開くことができます。

- **Directory** タブで、エントリーを右クリックし、ポップアップメニューから **Advanced Properties** を選択します。
- **Directory** タブから、エントリーをダブルクリックして、詳細 **ボタンをクリックします**。
- **Create...** new entry フォームから **Advanced** ボタンをクリックします。
- **OK**をクリックして、新規オブジェクト **ウィンドウから**

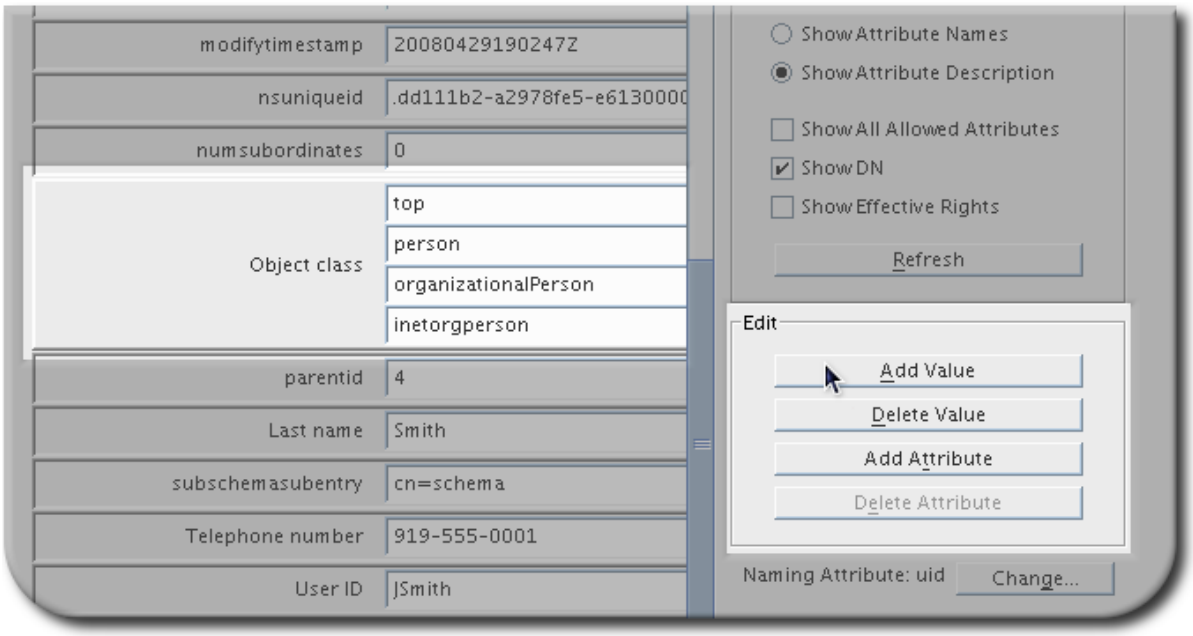
3.2.3.1. オブジェクトクラスのエントリーへの追加または削除

オブジェクトクラスをエントリーに追加するには、以下を実行します。

1. **Directory** Server コンソールの Directory タブで、エントリーを右クリックし、ポップアップメニューから **Advanced** を選択します。

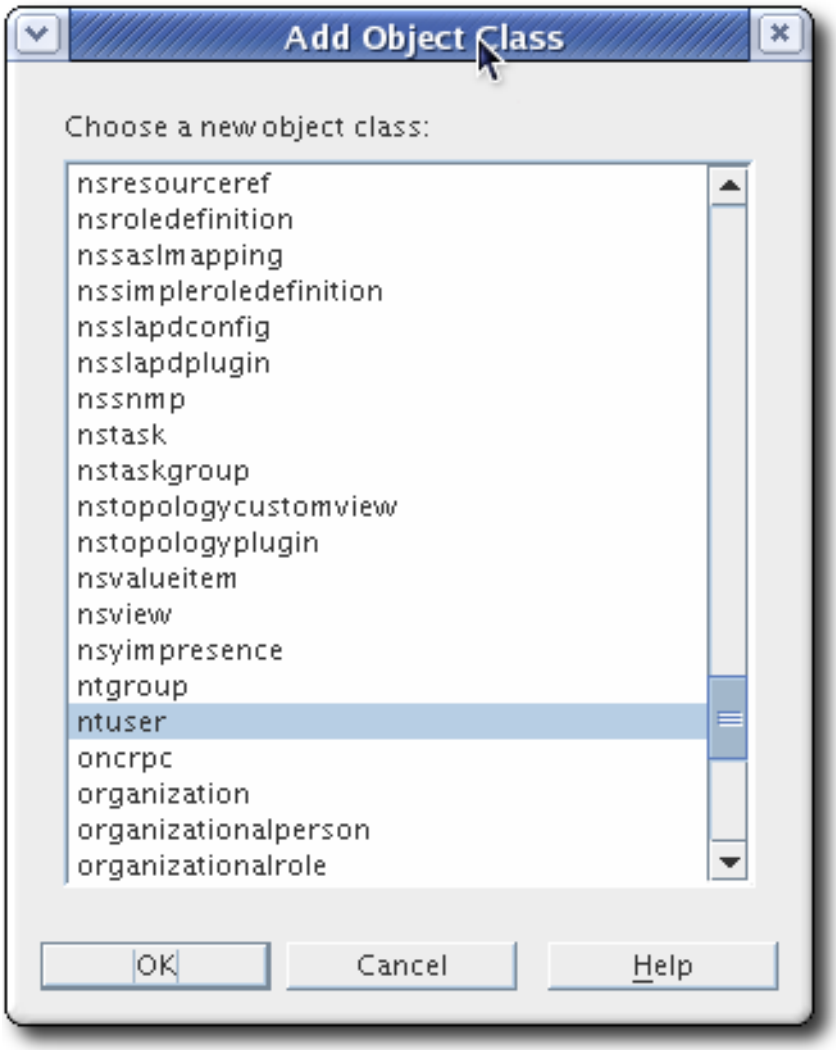


2. オブジェクトクラスフィールドを選択し、**Add Value** をクリックします。



Add Object Class ウィンドウが開きます。これは、エントリーに追加できるオブジェクトクラスの一覧を表示します。

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。



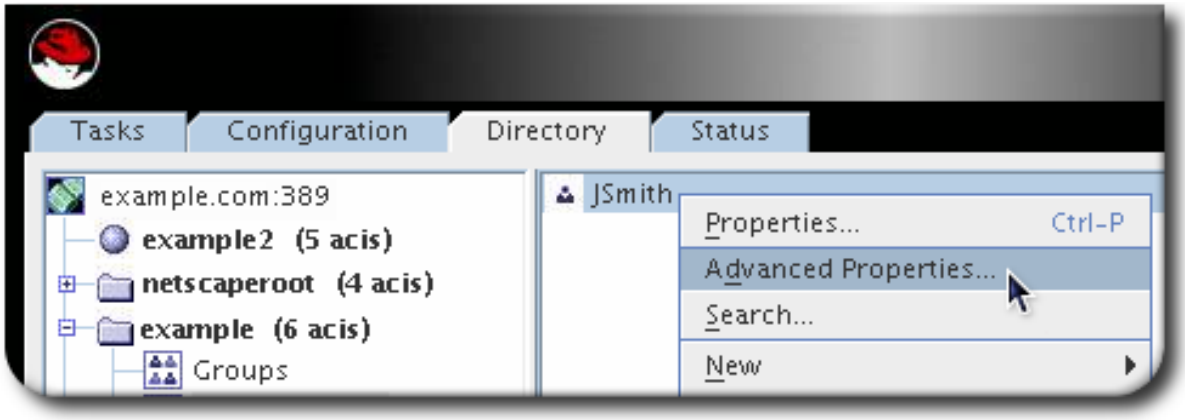
エントリーからオブジェクトクラスを削除するには、削除するオブジェクトクラスのテキストボックスに、**Delete Value** をクリックします。

3.2.3.2. エントリーへの属性の追加

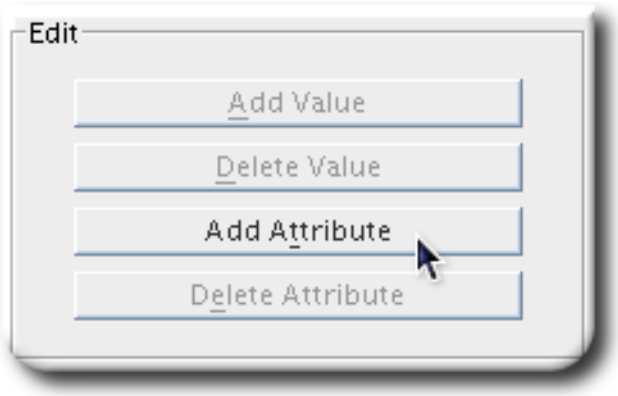
エントリーに属性を追加する前に、エントリーに属性を必要とするか、許可するオブジェクトクラスを含める必要があります。詳細は、「オブジェクトクラスのエントリーへの追加または削除」 および 12 章 *ディレクトリースキーマの管理* を参照してください。

エントリーに属性を追加するには、以下を実行します。

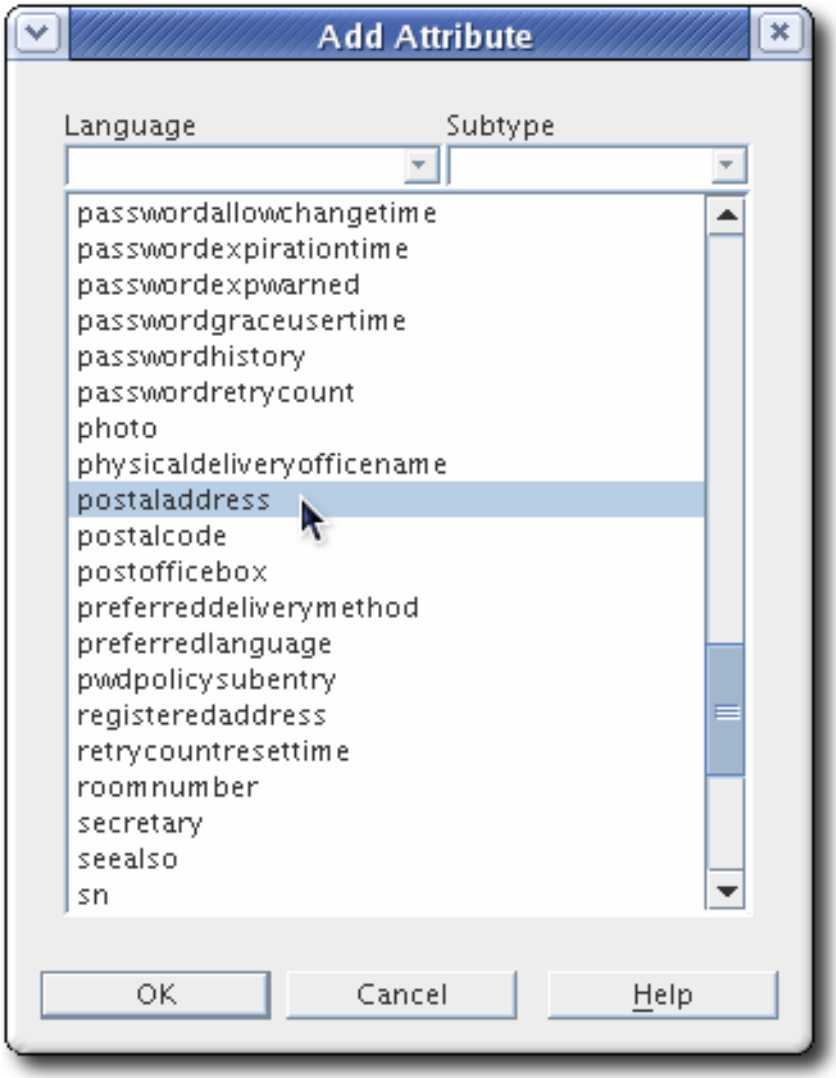
1. **Directory** Server コンソールの Directory タブで、エントリーを右クリックし、ポップアップメニューから **Advanced** を選択します。



2. **Add Attribute** をクリックします。



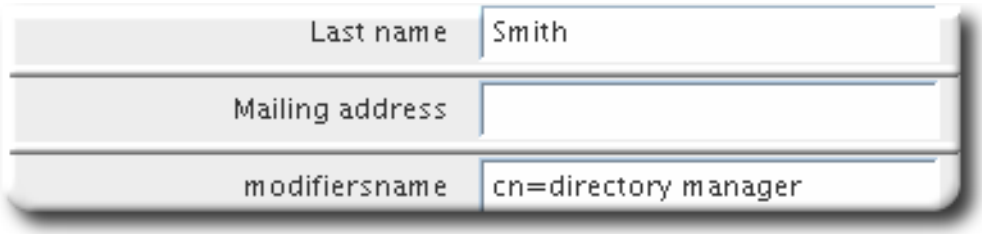
3. 一覧から追加する属性を選択し、**OK** をクリックします。



注記

追加する属性が一覧にない場合は、最初に属性が含まれるオブジェクトクラスを追加し、続いて属性を追加します。オブジェクトクラスの追加方法は、「オブジェクトクラスのエントリーへの追加または削除」 を参照してください。必要な属性を含むオブジェクトクラスが見つからない場合は、『Red Hat Directory Server 10 Configuration, Command, and File Reference』 （この属性を使用するオブジェクトクラスの一覧）の属性を検索します。

4. 属性名の右側にあるフィールドの新しい属性の値を入力します。



エントリーから属性とすべての値を削除するには、**Edit** メニューから **Delete Attribute** を選択します。

3.2.3.3. 大きな属性の追加

設定属性 ***nsslapd-maxbersize*** は LDAP 要求の最大サイズ制限を設定します。Directory Server のデフォルト設定では、この属性を 2 メガバイトに設定します。要求で 2 メガバイトより大きい非常に大きな属性を追加しようとすると、LDAP の追加または修正操作は失敗します。ただし、この制限はレプリケーションプロセスには適用されません。

非常に大きな属性を追加するには、まず ***nsslapd-maxbersize*** 設定属性の設定を、実行する最大の LDAP 要求よりも大きい値に変更します。

設定する値を決定する際には、1 つの属性だけでなく、属性を追加するのに使用される LDAP add および modify 操作 **のすべての要素** を考慮してください。考慮すべき要因には、以下が含まれます。

- リクエストの各属性名のサイズ
- リクエストの各属性の値のサイズ
- 要求内の DN のサイズ
- オーバーヘッド（通常は 10 キロバイト）

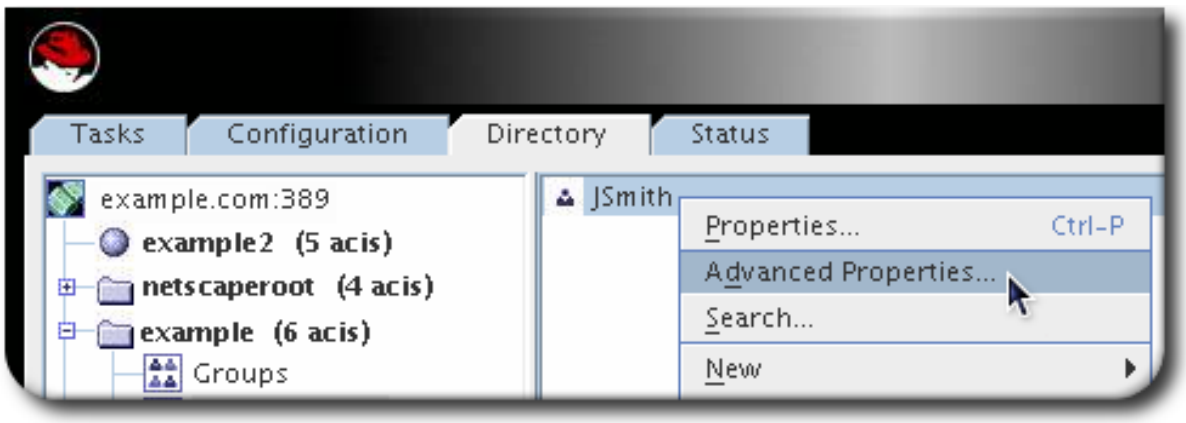
nsslapd-maxbersize 設定を増やす必要がある一般的な問題の 1 つは、CRL 値（***certificateRevocationList***、***authorityRevocationList***、および ***deltaRevocationList*** など）を保持する属性を使用することです。

nsslapd-maxbersize 属性の詳細は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス の該当するセクションを参照してください』。

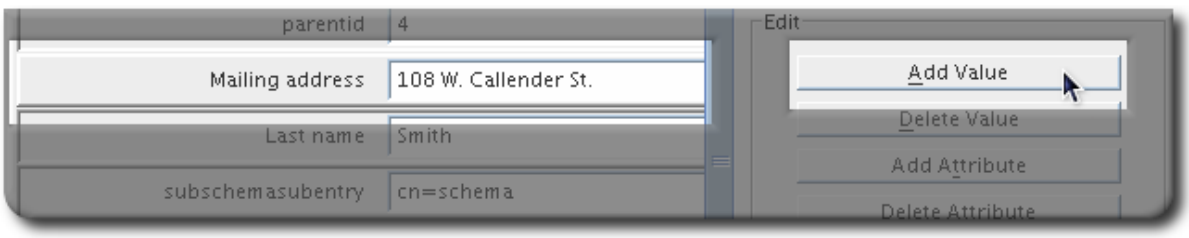
3.2.3.4. 属性値の追加

多値属性は、1 つの属性に対して複数の値をエントリーに追加できます。

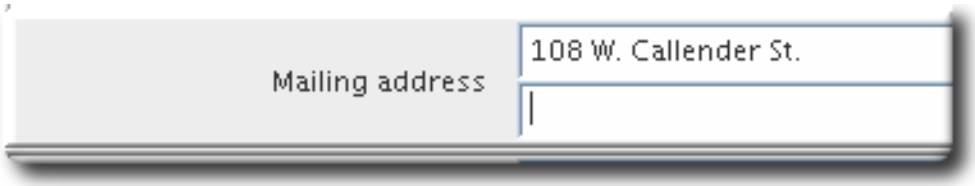
1. **Directory** Server コンソールの Directory タブで、エントリーを右クリックし、ポップアップメニューから **Advanced** を選択します。



2. 値を追加する属性を選択し、値 **の追加 をクリックします。**



3. 新しい属性値を入力します。



エントリーから属性値を削除するには、削除する属性値のテキストボックスに、**Delete Value** をクリックします。

3.2.3.5. 属性サブタイプの追加

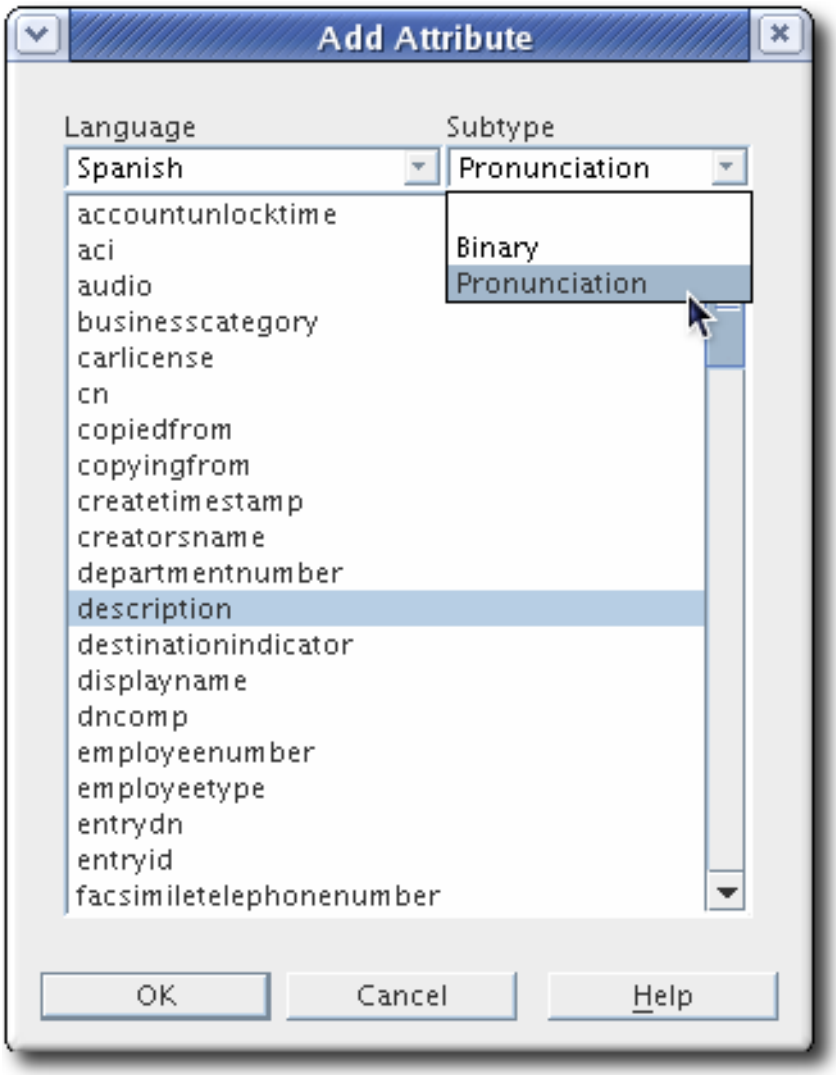
サブタイプでは、外部文字のセットバージョンを提供するなど、同じエントリー値をさまざまな方法で表すことができます。エントリーに追加できる属性には、言語、バイナリー、および発音の 3 種類のサブタイプがあります。

サブタイプをエントリーに追加するには、次のコマンドを実行します。

1. **Directory** Server コンソールの Directory タブで、エントリーを右クリックし、ポップアップメニューから **Properties** を選択します。

2. **Add Attribute** をクリックし、一覧から追加する属性を選択します。

3. **言語** ドロップダウンリストから値を選択して、言語サブタイプを追加します。**Subtype** ドロップダウンリストから値を選択して、バイナリーまたはプローブのサブタイプを追加します。



言語サブタイプ

ユーザーの名前は、デフォルト言語以外の言語の文字でより正確に表現できる場合があります。たとえば、ユーザーである Noriko は日本語の名前を持ち、可能な限り日本語の文字で表現します。指定の name **属性の** 言語サブタイプとして日本語を選択して、他のユーザーが日本語と英語で名前を検索できるようにします。以下に例を示します。

```
givenname;lang-ja
```

属性に言語サブタイプを指定するには、以下のようにサブタイプを属性名に追加します。

```
attribute;lang-subtype:attribute value
```

attribute はエントリーに追加される属性です。**サブタイプ** は、言語の略語の 2 文字です。サポートされる言語のサブタイプは、表D.1「サポートされる言語サブタイプ」 に記載されています。

エントリーの属性 **インスタンスごとに** 1 つの言語サブタイプのみを追加できます。複数の言語のサブタイプを割り当てるには、別の属性インスタンスをエントリーに追加してから、新しい言語のサブタイプを割り当てます。たとえば、以下は不正なものです。

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

代わりに以下を使用します。

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

バイナリーサブタイプ

バイナリーサブタイプを属性に割り当てると、**ユーザー証明書(usercertificate;binary)**などの属性値がバイナリーであることを示します。

バイナリーデータ **はバイナリー** サブタイプ（例： **jpegphoto**）を含まない属性に保存できますが、**バイナリー** サブタイプは、属性タイプの複数のバリアントが存在する可能性があるクライアントに対して指定します。

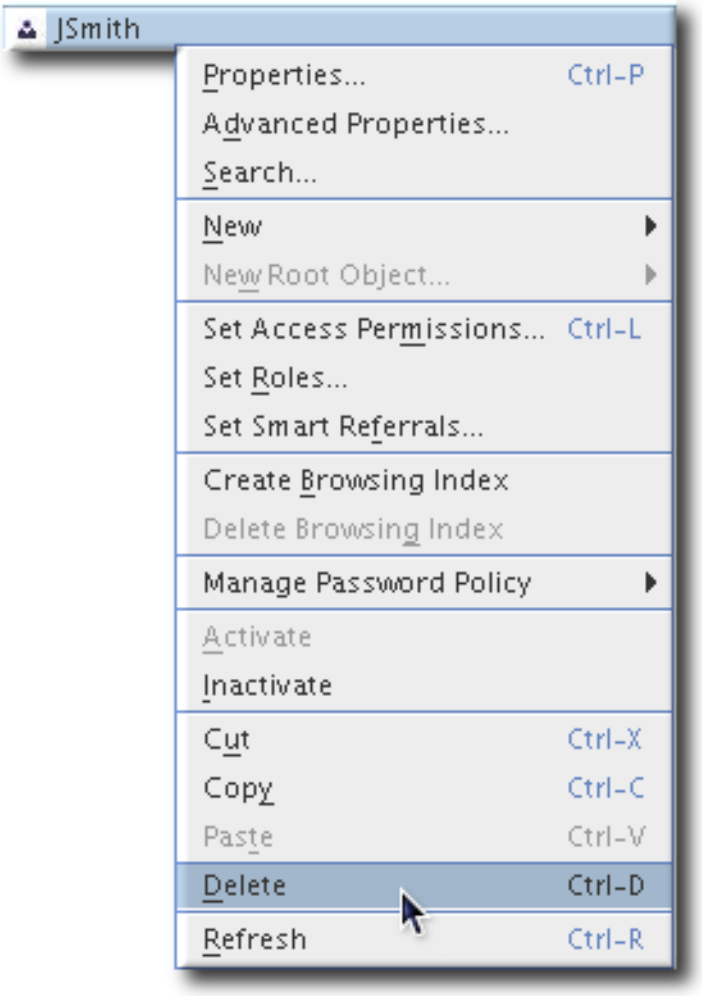
Pronunciation サブタイプ

属性に pronunciation サブタイプを割り当てると、属性値が電話番号であることを示します。サブタイプは、属性名に **attribute;phonetic** として追加されます。このサブタイプは、一般的に、複数のアルファベットを持つ言語サブタイプと組み合わせて使用されます。1 つは電話番号です。

このサブタイプは、**cn** や **givenname** などのユーザー名が含まれることが想定される属性に便利です。たとえば、**anyname;lang-ja;phonetic** は、属性値がユーザーの日本語名の電話番号であることを示します。

3.2.4. ディレクトリーエントリーの削除

1. Directory Server コンソールで、Directory **タブを選択します**。
2. エントリーを右クリックして削除するメニューから **Delete** を選択します。



警告

サーバーは、エントリーまたはエントリーを即座に削除します。削除操作を元に戻す方法はありません。

第4章 ディレクトリーエントリーの変更の追跡

これは、エントリーに変更が加えられるタイミングを追跡するのに役立ちます。Directory Server が追跡するエントリーの変更には、以下の 2 つの側面があります。

- 変更シーケンス番号を使用してデータベースへの変更を追跡します。これは、レプリケーションおよび同期で使用されるシーケンス番号の変更に類似しています。通常のディレクトリー操作はすべて、シーケンス番号がトリガーされます。
- 作成および変更の情報を割り当てます。これらの属性は、エントリーを作成して直近に変更したユーザーの名前と、エントリーの作成および修正時のタイムスタンプを記録します。

注記

エントリー USN、時間および名前の変更、および時間および作成はすべて操作属性であり、通常の **ldapsearch** では返されません。操作属性の検索実行に関する詳細は、「操作属性の検索」を参照してください。

4.1. 更新シーケンス番号でデータベースへの変更の追跡

USN プラグインは、LDAP クライアントがデータベース内のものが変更されたことを通知する方法を提供します。

4.1.1. エントリーシーケンス番号の概要

USN プラグインが有効な場合は、エントリーに対して書き込み操作を実行するたびに、エントリーに割り当てられるシーケンス番号 (USN) を更新します。書き込み操作には、add、modify、modrdn、および delete 操作が含まれます。エクスポート操作などの内部データベース操作は、更新シーケンスでカウントされません。 USN カウンターは、最近割り当てられた USN を追跡します。

4.1.1.1. ローカルおよびグローバルの USN

USN は、単一のエントリーではなく、データベース全体に対してグローバルに評価されます。USN は、データベースまたはディレクトリーの変更を追跡するために単に上向きにチェックするという点で、レプリケーションと同期の変更シーケンス番号に似ています。ただし、エントリー USN は CSN と別個に維持され、USN は複製されません。

このエントリーには、***entryUSN*** オペレーション属性のエントリーへの最後の変更の変更番号が表示されます。（操作属性の検索実行に関する詳細は、「操作属性の検索」 を参照してください。）

例4.1 エントリー USN の例

```
dn: uid=jsmith,ou=People,dc=example,dc=com
mail: jsmith@example.com
uid: jsmith
givenName: John
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
sn: Smith
cn: John Smith
userPassword: {SSHA}EfhKCl4iKl/ipZMsWIlTQatz7v2lUnptxwZ/pw==
entryusn: 1122
```

USN プラグインには、ローカルモードとグローバルモードという 2 つのモードがあります。

- ローカルモードでは、各バックエンドデータベースには、そのバックエンドデータベースに固有の USN カウンターを持つ USN プラグインのインスタンスがあります。これはデフォルト設定です。
- グローバルモードでは、ディレクトリー全体に追加された変更に適用されるグローバル USN カウンターを使用する USN プラグインのグローバルインスタンスがあります。

USN プラグインをローカルモードに設定すると、結果はローカルのバックエンドデータベースに限定されます。USN プラグインをグローバルモードに設定すると、返される結果はディレクトリー全体に対して行われます。

ルート DSE は、***lastusn*** 属性のデータベースのエントリーに割り当てられた最新の USN を表示します。USN プラグインがローカルモードに設定されているので、各データベースに独自のローカル USN カウンターがある場合、***lastUSN*** は、USN が割り当てられているデータベースと、USN の両方を表示します。

```
lastusn;database_name:USN
```

以下に例を示します。

```
lastusn;example1: 2130
lastusn;example2: 2070
```

グローバルモードでは、データベースが共有 USN カウンターを使用する場合、***lastUSN*** 属性は最新の USN のみを表示します。

```
lastusn: 4200
```

4.1.1.2. USN エントリーのインポート

エントリーがインポートされると、USN プラグインは ***nsslapd-entryusn-import-initval*** 属性を使用して、エントリーに USN が割り当てられているかどうかを確認します。***nsslapd-entryusn-import-initval*** の値が数値である場合、インポートされたエントリーはこの数字をエントリーの USN として使用します。***nsslapd-entryusn-import-initval*** の値が数値でない場合、USN プラグインは ***lastUSN*** 属性の値を使用して、インポートしたエントリーの USN で増やします。

4.1.2. USN プラグインの設定

「Directory Server コンソールでプラグインの有効化」 で説明されているように、USN プラグインをエントリーに記録するには有効にする必要があります。プラグインは、Directory Server Console またはコマンドラインを使用して有効にできます。以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -x
dn: cn=USN,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

次にサーバーを再起動して変更を適用します。

4.1.3. グローバル USN の有効化

デフォルト設定では、Directory Server は各バックエンドデータベースに一意の更新シーケンス番号 (USN) を使用します。すべてのバックエンドデータベースで一意の USN を有効にするには、以下を実行します。

1. **USN** プラグインを有効にします。「USN プラグインの設定」 を参照してください。
2. ***nsslapd-entryusn-global*** パラメーターを **on** に設定します。

```
# ldapmodify -D "cn=Directory Manager" -W -x
dn: cn=config
changetype: modify
replace: nsslapd-entryusn-global
nsslapd-entryusn-global: on
```

4.1.4. USN Tombstone エントリーのクリーンアップ

エントリーが削除されると、USN プラグインは、エントリーを tombstone エントリーに移動します。レプリケーションが有効な場合は、USN および Replication プラグインによって個別の tombstone エントリーが保持されます。tombstone エントリーはレプリケーションプロセスで削除されますが、サーバーのパフォーマンスのために、サーバーをレプリカに変換するか、サーバーのメモリーを解放する前に USN tombstones を削除することが有益です。

usn-tombstone-cleanup.pl コマンドは、特定のデータベースバックエンドまたは特定の接尾辞の USN tombstone エントリーを削除します。必要に応じて、特定の USN までの tombstone エントリーをすべて削除できます。以下に例を示します。

```
# /usr/lib64/dirsrv/instance/usn-tombstone-cleanup.pl -D "cn=Directory Manager" -w secret -s
"ou=people,dc=example,dc=com" -m 1100
```

バックエンドは、- **s** オプションを使用して、- **n** オプションまたは接尾辞を使用して指定する必要があります。両方を指定すると、- **s オプション** の接尾辞が使用されます。

usn-tombstone-cleanup.pl コマンドのオプションは、表4.1「USN-tombstone-cleanup.pl オプション」 に一覧表示されます。このツールの詳細は、設定、『コマンド、およびファイルリファレンス を参照してください』。

表4.1 USN-tombstone-cleanup.pl オプション

オプション	詳細
-D **rootdn**	Directory Manager などの **root** 権限でユーザー DN を指定します。デフォルトは、Directory Manager の DN です。これは、**cn=config** 下の ***nsslapd-root*** 属性から読み取られます。
-m **maximum_USN**	削除するエントリーの上限を設定します。指定された最大値(inclusive)までの ***entryUSN*** 値を持つすべての tombstone エントリーは削除されますが、USN 値を超えると削除されます。最大 USN 値が設定されていない場合、すべてのバックエンド tombstone エントリーが削除されます。
-n **backendInstance**	クリーニングするエントリーが含まれるデータベースの名前を指定します（削除）。

オプション	詳細
-s **suffix**	クリーニングするエントリーを含む接尾辞の名前を指定します（削除）。
-w **password**	ユーザー DN に関連付けられたパスワード。

4.2. 操作属性によるエントリー変更の追跡

デフォルト設定を使用すると、Directory Server は、全エントリーで以下の操作属性を追跡します。

- ***creatorsName***: エントリーを最初に作成したユーザーの識別名 (DN) です。
- ***createTimestamp***: エントリーの作成時にグリニッジ標準時 (GMT) 形式のタイムスタンプ。
- ***modifiersName***: エントリーを最後に変更したユーザーの識別名。
- ***modifyTimestamp***: エントリーが最後に修正された時点の GMT 形式のタイムスタンプ。

デフォルトの検索では操作属性が返されないことに注意してください。これらの属性はクエリーで明示的に要求する必要があります。詳細は 「操作属性の検索」を参照してください。

重要

これらの操作属性の追跡は、無効にしないことが推奨されます。無効にすると、エントリーは ***nsUniqueID*** 属性に割り当てられた一意の ID を取得しなくなり、レプリケーションは機能しません。

4.2.1. データベースリンクにより変更されたエントリーまたは作成済みエントリー

データベースリンク上でエントリーが作成または変更されると、***creatorsName*** および ***modifiersName*** 属性には、リモートサーバーのプロキシー認可権限を付与されたユーザー名が含まれます。この場合、属性はエントリーの元の作成者または最新の変更者を表示しません。ただし、アクセスログには、プロキシーユーザー (**dn**) と元のユーザー (**authzid**) の両方が表示されます。以下に例を示します。

```
[23/May/2011:18:13:56.145747965 +051800] conn=1175 op=0 BIND dn="cn=proxy
admin,ou=People,dc=example,dc=com" method=128 version=3
[23/May/2011:18:13:56.575439751 +051800] conn=1175 op=0 RESULT err=0 tag=97 nentries=0
etime=0 dn="cn=proxy admin,ou=people,dc=example,dc=com"
[23/May/2011:18:13:56.744359706 +051800] conn=1175 op=1 SRCH base="dc=example,dc=com"
scope=2 filter="(objectClass=*)" attrs=ALL
authzid="uid=user_name,ou=People,dc=example,dc=com"
```

プロキシー承認の詳細は、「バインド認証情報の提供」を参照してください。

4.2.2. コマンドラインを使用した変更の追跡を有効にする方法

変更追跡はデフォルトで有効になっています。Red Hat は、この機能を無効にしないことを推奨します。コマンドラインでエントリー変更の追跡を再度有効にするには、次のコマンドを実行します。

1. ***nsslapd-lastmod*** を **on** に設定します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=config
nsslapd-lastmod: on
```

2. 必要に応じて、不足している ***nsUniqueID*** 属性を再生成するには、以下を実行します。

   a. データベースを LDAP データ交換形式(LDIF)ファイルにエクスポートします。「コマンドラインを使用した LDIF へのデータベースのエクスポート」 を参照してください。

   b. LDIF ファイルからデータベースをインポートします。「コマンドラインからのインポート」 を参照してください。

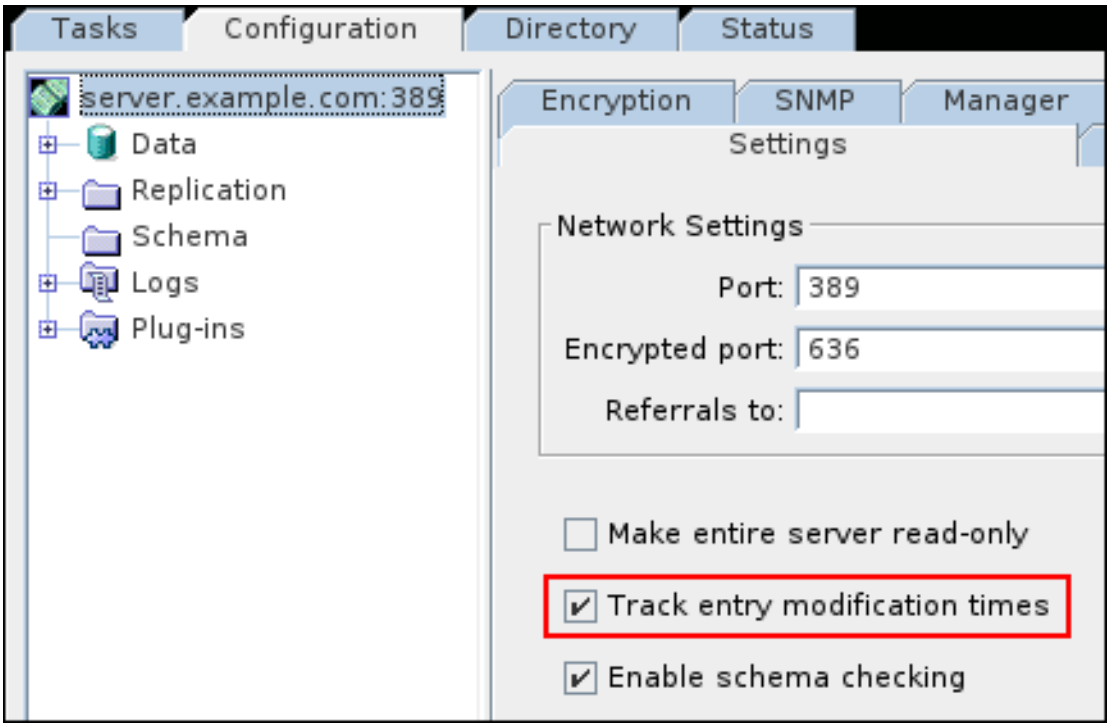
4.2.3. コンソールを使用した変更の追跡を有効にする方法

変更追跡はデフォルトで有効になっています。Red Hat は、この機能を無効にしないことを推奨します。コンソールを使用してエントリー変更の追跡を再度有効にするには、以下を実行します。

1. Directory Server コンソールを開きます。「Directory Server コンソールを開く」 を参照してください。

2. **Configuration** タブでサーバー名を選択します。

3. **Settings** タブで **Track Entry Modification Times** チェックボックスを選択します。



4. 必要に応じて、不足している ***nsUniqueID*** 属性を再生成するには、以下を実行します。

   a. データベースを LDAP データ交換形式(LDIF)ファイルにエクスポートします。「コマンドラインを使用した LDIF へのデータベースのエクスポート」 を参照してください。

   b. LDIF ファイルからデータベースをインポートします。「コマンドラインからのインポート」 を参照してください。

4.3. プラグイン開始更新のバインド DN の追跡

エントリーへの変更の 1 つで、ディレクトリーツリー全体で、他の自動変更をトリガーすることができます。たとえば、ユーザーが削除されると、そのユーザーは **Referential Integrity Postoperation** プラグインが属するグループから自動的に削除されます。

最初のアクションは、サーバーにバインドされているユーザーアカウントによって実行されているものとしてエントリーに表示されますが、関連するすべての更新 (デフォルト) はプラグインによって実行されているものとして表示され、どのユーザーがその更新を開始したかについての情報はありません。たとえば、MemberOf プラグインを使用してグループメンバーシップでユーザーエントリーを更新し、グループアカウントの更新はバインドされたユーザーが実行済みとして表示されますが、ユーザーエントリーの編集は MemberOf プラグインによって実行されると表示されます。

```
dn: cn=my_group,ou=groups,dc=example,dc=com
modifiersname: uid=jsmith,ou=people,dc=example,dc=com

dn: uid=bjensen,ou=people,dc=example,dc=com
modifiersname: cn=memberOf plugin,cn=plugins,cn=config
```

nsslapd-plugin-binddn-tracking 属性により、サーバーは更新操作を開始したユーザーと、実際に実行した内部プラグインを追跡できます。バインドされたユーザーは ***modifiersname*** 操作属性および ***creatorsname*** 操作属性に表示されますが、実行されたプラグインは ***internalModifiersname*** 操作属性および ***internalCreatorsname*** 操作属性に表示されます。以下に例を示します。

```
dn: uid=bjensen,ou=people,dc=example,dc=com
modifiersname: uid=jsmith,ou=people,dc=example,dc=com
internalModifiersname: cn=memberOf plugin,cn=plugins,cn=config
```

nsslapd-plugin-binddn-tracking 属性は、バインドされたユーザーと、その接続に対して実行される更新間の関係を追跡し、維持します。

注記

internalModifiersname 属性および ***internalCreatorsname*** 属性は、常にプラグインをアイデンティティーとして表示します。このプラグインは、MemberOf プラグインなどの追加のプラグインである可能性があります。コア Directory Server が変更を行うと、プラグインはデータベースプラグイン **cn=ldbm database,cn=plugins,cn=config** になります。

nsslapd-plugin-binddn-tracking 属性はデフォルトで無効になっています。サーバーがバインド DN に基づいて操作を追跡できるようにするには、**ldapmodify** を使用してその属性を有効にします。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=config
changetype: modify
replace: nsslapd-plugin-binddn-tracking
nsslapd-plugin-binddn-tracking: on
```

4.4. パスワード変更時間の追跡

パスワードの変更操作は、通常、エントリーに対するその他の変更として扱われるため、更新時間は ***lastModified*** 操作属性に記録されます。ただし、Active Directory 同期におけるパスワードの更新や、他の LDAP クライアントへの接続を容易にするため、パスワード変更の時間は別々に記録しないといけない場合があります。

パスワードポリシー内の ***passwordTrackUpdateTime*** 属性は、エントリーのパスワードが最後に更新された日時のタイムスタンプを記録するようサーバーに指示します。パスワードの変更時間は、***pwdUpdateTime*** (***modifyTimestamp*** または ***lastModified*** の操作属性とは別の) ユーザーエントリーで操作属性として保存されます。

passwordTrackUpdateTime 属性は、パスワード変更時間にアクセスする必要があるクライアントに応じて、グローバルパスワードポリシー、サブツリー、またはユーザーレベルのポリシーの一部として設定できます。パスワードポリシーの設定は、「パスワードポリシーの管理」を参照してください。

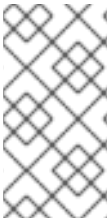
第5章 参照整合性の維持

参照整合性 は、関連するエントリー間の関係を維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、ディレクトリー内の 1 つのエントリーへの更新が、更新されたエントリーを参照するその他のエントリーに適切に反映されるようにします。

たとえば、ユーザーのエントリーがディレクトリーから削除され、参照整合性が有効になると、サーバーはユーザーがメンバーとなるグループからユーザーも削除します。参照整合性が有効になっていないと、ユーザーは管理者が手動で削除するまでグループのメンバーのままになります。これは、ユーザーおよびグループの管理のディレクトリーに依存する他の製品と Directory Server を統合する場合に重要な機能です。

5.1. 参照整合性の仕組み

Referential Integrity Postoperation プラグインを有効にすると、削除または名前変更の操作直後に、指定した属性で整合性の更新を実行します。デフォルトでは、**Referential Integrity Postoperation** プラグインは無効になっています。



注記

プラグインによって生成された操作が複製されるため、マルチマスターレプリケーション環境の 1 つのサプライヤーレプリカでのみ **Referential Integrity Postoperation** プラグインを有効にします。複数のマスターでプラグインを有効にする場合は、サーバーですでに実行された操作を管理し、再適用する必要があります。

ユーザーエントリーまたはグループエントリーがディレクトリー内で削除、更新、名前変更、または移動すると、その操作は参照整合性ログファイルに記録されます。ログファイル内の識別名 (DN) の場合、Directory Server はプラグイン設定に設定された属性を定期的に検索および更新します。

- エントリーの場合は、ログファイルで削除済みと表示され、ディレクトリーの対応する属性が削除されます。
- エントリーの場合は、ログファイルで更新済みと表示され、ディレクトリーの対応する属性が更新されます。
- エントリーの場合は、ログファイルで名前が変更または移動済みと表示され、ディレクトリーの対応する属性の名前が変わります。

デフォルトでは、**Referential Integrity Postoperation** プラグインが有効な場合は、削除または名前変更の操作直後に ***member***、***uniquemember***、***owner***、および ***seeAlso*** 属性で整合性の更新を実行します。ただし、**Referential Integrity Postoperation** プラグインの動作は、ディレクトリーのニーズに応じて複数の方法で設定できます。

- レプリケーション変更ログに整合性の更新を記録します。
- 更新間隔を変更します。
- 参照整合性を適用する属性を選択します。
- 参照整合性を無効にします。

参照整合性で使用される属性はすべて存在と等価性のためにインデックス化する **必要** があります。これらの属性をインデックス化しないと、変更操作および削除操作のためにサーバーのパフォーマンスが低下します。

```
nsIndexType: pres
nsIndexType: eq
nsIndexType: sub
```

インデックスの確認および作成に関する詳細は、「標準インデックスの作成」を参照してください。

5.2. レプリケーションによる参照整合性の使用

レプリケーション環境で **Referential Integrity Postoperation** プラグインを使用する場合は、特定の制限があります。

- 専用のコンシューマーサーバー (読み取り専用レプリカのみが含まれるサーバー) では有効に **しないでください**。
- 読み書きレプリカと読み取り専用レプリカの組み合わせが含まれるサーバーで有効に **しないでください**。
- 読み書きレプリカのみが含まれるサプライヤーサーバーで有効にできます。
- マルチマスターレプリケーションでは、1 つのサプライヤーでプラグインを有効にします。

レプリケーション環境がこれらのすべての条件を満たす場合は、**Referential Integrity Postoperation** プラグインを有効にすることができます。

1. 「参照整合性の有効化および無効化」 の説明に従って、**Referential Integrity Postoperation** プラグインを有効にします。
2. 変更ログに整合性の更新を記録するようにプラグインを設定します。
3. すべてのコンシューマーサーバーで、**Referential Integrity Postoperation** プラグインが無効になっていることを確認します。

注記

サプライヤーサーバーが **Referential Integrity Postoperation** 整合性プラグインの変更をコンシューマーサーバーに送信するため、コンシューマーサーバーで **Referential Integrity Postoperation** プラグインを実行する必要はありません。

5.3. 参照整合性の有効化および無効化

5.3.1. コマンドラインから参照整合性の有効化および無効化

Referential Integrity Postoperation プラグインを有効または無効にするには、プラグインの設定エントリーに ***nsslapd-pluginEnabled*** パラメーターを設定します。

たとえば、プラグインを有効にするには、以下を実行します。

1. ***nsslapd-pluginEnabled*** パラメーターを **on** に設定します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=,cn=plugins,cn=config
```

```
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

2. インスタンスを再起動します。

```
# systemctl restart dirsrv@instance_name
```

5.3.2. コンソールでの参照整合性の有効化および無効化

Referential Integrity Postoperation プラグインを有効にするには、「Directory Server コンソールでプラグインの有効化」 の手順に従います。

5.4. 更新間隔の変更

デフォルトでは、サーバーは、**delete** または **modrdn** の操作直後に参照整合性の更新を実行します。操作の量によっては、パフォーマンスに影響する可能性があります。パフォーマンスへの影響を軽減するために、更新間の時間を増やすことができます。

間隔を秒単位で設定します。あるいは、以下の値を設定できます。

- **0**: 参照整合性の確認は即座に実行されます。
- **-1**: 参照整合性の確認は実行されません。

重要

更新間隔を **0** に設定すると、**Referential Integrity Postoperation** プラグインの更新間隔を **0** に設定する場合に限り、マルチマスターレプリケーション環境のすべてのマスターでプラグインを有効にすることができます。ただし、1 つのマスターに正の値を設定する場合は、レプリケーションループとディレクトリーの不整合を防ぐために、他のマスターでプラグインを有効にしないでください。

マルチマスターレプリケーション環境でプラグインを有効にする場合、Red Hat は更新間隔を **0** に設定し、すべてのマスターでプラグインを有効にすることを推奨します。

5.4.1. コマンドラインを使用した更新間隔の変更

たとえば、コマンドラインを使用して更新間隔を即座更新に設定するには、次を実行します。

1. ***referint-update-delay*** パラメーターで間隔を秒単位で設定します。

```
# ldapmodify -D "cn=Directory Manager" -W -x
dn: cn=referential integrity postoperation,cn=plugins,cn=config
changetype: modify
replace: referint-update-delay
referint-update-delay: 0
```

2. Directory Server インスタンスを再起動します。

```
# systemctl restart dirsrv@instance_name
```

参照整合性 は 1 つのマスターでのみ有効にできます。この間隔を **0** に設定すると、Directory Server は

上記の変更をすべてのコンシューマーに即座に複製します。間隔を **0** よりも大きい値に設定し、**Referential Integrity** が有効なマスターがオフラインである場合は、このマスターが再び起動する前に参照はクリーンアップされません。

5.4.2. コンソールを使用した更新間隔の変更

コンソールを使用して更新間隔を設定するには、以下を行います。

1. **Referential Integrity Postoperation** プラグインの設定で **Property Editor** を開きます。詳細は、「コンソールを使用したプラグインの設定」 を参照してください。
2. ***referint-update-delay*** パラメーターで間隔を秒単位で設定します。
3. Directory Server インスタンスを再起動します。「コンソールを使用した Directory Server インスタンスの起動および停止」 を参照してください。

5.5. 属性一覧の変更

デフォルトでは、参照整合性プラグインは ***member*** 属性、***uniquemember*** 属性、***owner*** 属性、および ***seeAlso*** 属性を確認し、更新します。コマンドラインまたはコンソールを使用して、更新する属性を追加または削除できます。

注記

Referential Integrity プラグインのパラメーターリストに設定される属性には、全データベースで等価インデックスが必要です。そうでない場合、プラグインは削除済みまたは変更された DN に一致するためにデータベースのすべてのエントリーをスキャンします。これにより、パフォーマンスに大きく影響する可能性があります。インデックスの確認および作成に関する詳細は、「標準インデックスの作成」を参照してください。

5.5.1. コンソールを使用した属性一覧の変更

1. **Referential Integrity Postoperation** プラグインの設定で **Property Editor** を開きます。詳細は、「コンソールを使用したプラグインの設定」 を参照してください。
2. ***referint-membership-attr*** 属性の属性を更新します。

   値の追加 **および** 値 **の削除 ボタンを使用して**、値を追加 したり、既存の値を削除したりできます。
3. Directory Server インスタンスを再起動します。「コンソールを使用した Directory Server インスタンスの起動および停止」 を参照してください。

5.5.2. コマンドラインでの属性一覧の設定

1. 属性リストを更新します。
   - プラグインが確認および更新される必要があるその他の属性を追加するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -x

dn: cn=referential integrity postoperation,cn=plugins,cn=config
```

```
add: referint-membership-attr
referint-membership-attr: attribute_name
```

- プラグインが確認および更新されなくなった属性を削除するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -x

dn: cn=referential integrity postoperation,cn=plugins,cn=config
delete: referint-membership-attr
referint-membership-attr: attribute_name
```

2. Directory Server インスタンスを再起動します。

```
# systemctl restart dirsrv@instance_name
```

5.6. 参照整合性のためのスコープの設定

エントリーを削除すると、エントリーへの参照は削除または変更されます。この更新がすべてのエントリーおよびすべてのグループに適用されると、パフォーマンスに影響が及ぶ可能性があり、選択されたサブツリーに参照整合性を制限できなくなる可能性があります。この問題の *スコープアドレス*を定義します。

たとえば、接尾辞 **dc=example,dc=com** に **ou=active users,dc=example,dc=com** と **ou=deleted users,dc=example,dc=com** の 2 つのサブツリーが含まれる場合があります。**deleted users** のエントリーは、参照整合性の確保のために処理しないでください。

Referential Integrity Postoperation プラグイン設定でスコープを定義するために、以下の 3 つの属性を使用できます。

nsslapd-pluginEntryScope 属性

この多値属性は、削除または名前変更のエントリーの範囲を制御します。これは、Referential Integrity Postoperation プラグインがユーザーエントリーの削除操作または名前変更操作を検索するサブツリーを定義します。定義されたサブツリーに存在しないユーザーを削除または名前変更した場合、プラグインは操作を無視します。この属性を使用すると、プラグインが操作を適用するデータベースのブランチを指定できます。

```
nsslapd-pluginEntryScope: dn
```

nsslapd-pluginExcludeEntryScope 属性

この属性は、削除または名前変更のエントリーの範囲も制御します。これは、Referential Integrity Postoperation プラグインがユーザーの削除や名前変更の操作を無視するサブツリーを定義します。

```
nsslapd-pluginExcludeEntryScope: dn
```

nsslapd-pluginContainerScope 属性

この属性は、参照を更新するグループの範囲を制御します。ユーザーの削除後、Referential Integrity Postoperation プラグインはユーザーが属するグループを検索し、それに応じて更新します。この属性は、プラグインがユーザーが属するグループを検索するブランチを指定します。Referential Integrity Postoperation プラグインは、指定のコンテナーブランチにあるグループのみを更新し、その他のグループは更新されないままにします。

nsslapd-pluginContainerScope: *dn*

第6章 DIRECTORY DATABASE への入力

データベースには、Red Hat Directory Server が管理するディレクトリーデータが含まれます。

6.1. データのインポート

Directory Server には、データをインポートする（Directory Server Console またはインポートツールを使用）、またはレプリケーション用にデータベースを初期化することで、以下のいずれかの方法でデータベースにデータを作成できます。

表6.1「インポート方法の比較」では、インポートとデータベースの初期化の相違点を説明します。

表6.1 インポート方法の比較

動作	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	いいえ	はい
LDAP 操作	追加、変更、削除	追加のみ
パフォーマンス	より時間がかかる	速い
パーティション特長	すべてのパーティションで機能	ローカルパーティションのみ
サーバー障害への応答	ベストエフォート (障害が発生した時点までに行われたすべての変更が残る)	アトミック (障害発生後にすべての変更が失われる)
LDIF ファイルの場所	コンソールへのローカル	サーバーに対するコンソールまたはローカルへのローカル
設定情報 (**cn=config**) をインポートする	Yes	いいえ

6.1.1. インポート中の EntryUSN 初期値の設定

エントリーがサーバーからエクスポートされ、別のサーバーにインポートされた場合には、エントリーの更新シーケンス番号 (USN) が保持されません。「更新シーケンス番号でデータベースへの変更の追跡」で説明されているように、エントリー USN はローカルサーバーで発生する操作に割り当てられるため、これらの USN を別のサーバーにインポートすることは意味がありません。

ただし、データベースのインポート時やデータベースの初期化時にエントリーに USN 値を設定することが可能です (レプリカがレプリケーションに対して初期化される場合など)。これは、***nsslapd-entryusn-import-initval*** 属性を設定して行います。これにより、インポートされたすべてのエントリーの USN が開始されます。

nsslapd-entryusn-import-initval には 2 つの値を指定することができます。

- 整数。インポートされたすべてのエントリーに使用される明示的な開始番号です。
- **次に**、インポートされたすべてのエントリーは、インポート操作前にサーバー上の最も大きなエントリー USN 値を使用し、1 つずつ増分します。

nsslapd-entryusn-import-initval が設定されていない場合、すべてのエントリー USN はゼロで始まります。

たとえば、インポートまたは初期化操作の前にサーバーの最大値が 1000 で、***nsslapd-entryusn-import-initval*** の値が next の場合、インポートされたすべてのエントリーには 1001 の USN が割り当てられます。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x "(cn=*)" entryusn

dn: dc=example,dc=com
entryusn: 1001
dn: ou=Accounting,dc=example,dc=com
entryusn: 1001
dn: ou=Product Development,dc=example,dc=com
entryusn: 1001
...
dn: uid=jsmith,ou=people,dc=example,dc=com
entryusn: 1001
...
```

エントリー USN の初期値を設定するには、データをインポートするサーバーまたは初期化を実行するマスターサーバーに ***nsslapd-entryusn-import-initval*** 属性を追加するだけです。

```
# ldapmodify -D "cn=Directory Manager" -W -x -D "cn=directory manager" -W -p 389 -h
server.example.com -x

dn: cn=config
changetype: modify
add: nsslapd-entryusn-import-initval
nsslapd-entryusn-import-initval: next
```

注記

マルチマスターレプリケーションでは、***nsslapd-entryusn-import-initval*** 属性はサーバー間で **複製されません**。つまり、値は、レプリカの初期化に使用しているすべてのサプライヤーサーバーに対して設定する必要があります。

たとえば、Supplier1 に ***nsslapd-entryusn-import-initval*** が **next** に設定され、レプリカの初期化に使用される場合は、インポートされたエントリーのエントリー USN は最も高い値に 1 が追加されます。Supplier2 に ***nsslapd-entryusn-import-initval*** が設定されておらず、レプリカの初期化に使用される場合は、Supplier1 および Supplier 2 にマルチマスターレプリカ合意がある場合でも、インポートされたエントリーのすべてのエントリー USN はゼロで始まります。

6.1.2. コンソールからのデータベースのインポート

Directory Server コンソールからインポート操作を実行すると、**ldapmodify** 操作はデータを追加し、エントリーの変更や削除を行います。この操作は、Directory Server が管理するすべてのデータベースと、Directory Server が設定したデータベースリンクを持つリモートデータベースで実行されます。

インポート操作は、Directory Server コンソールに対してローカルとなるサーバーインスタンスまたは別のホストマシン（リモートインポート操作）で実行できます。

インポートを実行するには、Directory Manager としてログインしている必要があります。

注記

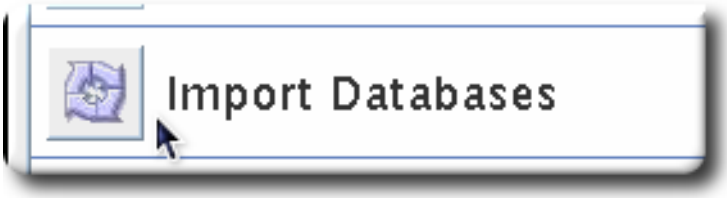
インポート操作に使用される LDIF ファイルは、UTF-8 文字セットエンコーディングを使用する必要があります。インポート操作は、データをローカル文字セットエンコーディングから UTF-8 文字セットエンコーディングに変換しません。

警告

インポートされたすべての LDIF ファイルには、root 接尾辞も含まれている必要があります。

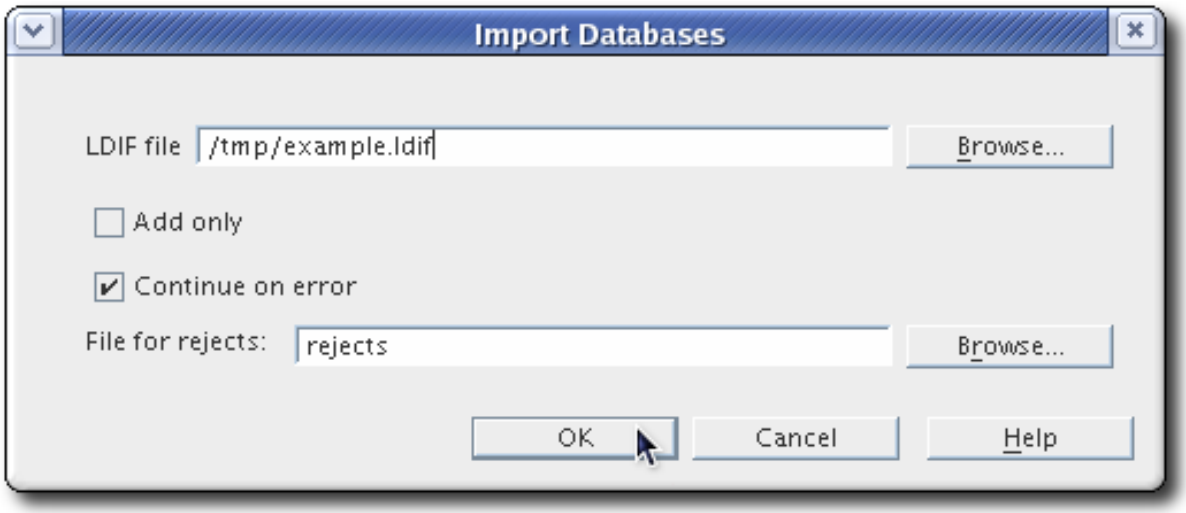
Directory Server コンソールからデータをインポートするには、以下を実行します。

1. **Tasks** タブを選択します。画面の下部までスクロールし、**Import Database** を選択します。



または、**Configuration** タブを開き、**Console** メニューから **Import** を選択します。

2. **Import Database** ダイアログボックスで、LDIF **ファイル** フィールドでインポートする LDIF ファイルへの完全パスを入力するか、**Browse** をクリックしてインポートするファイルを選択します。



コンソールがマシンのリモートで実行している場合、フィールド名は LDIF ファイルとして表示されます（コンソールを実行するマシン上）。**ファイルを参照する場合、Directory Server ホストの現在のディレクトリーは参照されませんが、コンソールを実行しているマシンのファイルシステムも参照しません。**

リモートコンソールを使用してデータベースをインポートする場合 **は、データベースへの相対**

パスを使用しないでください。リモートインポートでは、ファイルに相対パス **が指定されている場合に** Cannot write to file... というエラーで操作に失敗します。リモートインポート操作には常に絶対パスを使用します。

3. **オプション** ボックスで、以下のオプションのいずれかまたは両方を選択します。

   - **のみを追加します。**LDIF ファイルには、デフォルトの追加手順に加えて、変更および削除手順を含めることができます。サーバーが add 以外の操作を無視するには、**Add only** チェックボックスを選択します。

   - **Error に進みます。**エラーが発生した場合でもインポートを続行するには、サーバーの **Continue on error** チェックボックスを選択します。たとえば、このオプションを使用して、新しいものに加えて、データベースにすでに存在するエントリーが含まれる LDIF ファイルをインポートします。サーバーノートでは、すべての新規エントリーの追加中に rejects ファイルの既存のエントリーがあります。

4. **File for Rejects** フィールドに、サーバーがインポートできないすべてのエントリーを記録するファイルへの完全パスを入力するか、**Browse** をクリックして拒否が含まれるファイルを選択します。

   拒否は、データベースにインポートできないエントリーです。たとえば、サーバーは、データベースにすでに存在するエントリーや、親オブジェクトのないエントリーをインポートできません。コンソールは、サーバーによって送信されたエラーメッセージが rejects 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、サーバーは拒否されたエントリーを記録しません。

サーバーはインポートを実行し、インデックスも作成します。

注記

末尾のスペースは、リモートコンソールのインポート時に破棄されますが、ローカルコンソールまたは **ldif2db** インポート操作時に保持されます。

6.1.3. コンソールからのデータベースの初期化

データベース内の既存のデータは、データベースを初期化することで上書きできます。

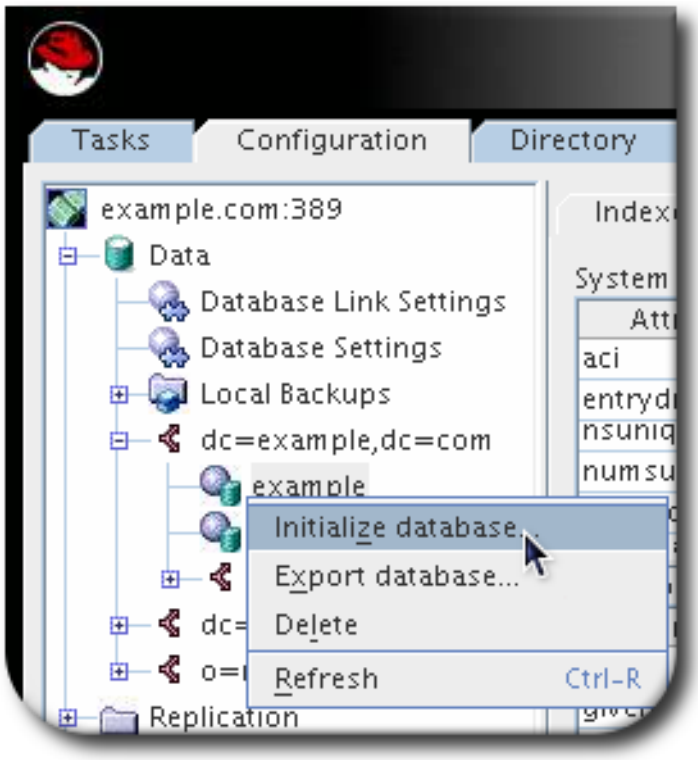
Directory Manager (root DN)を除き、ルートエントリーが含まれる LDIF ファイルをデータベースにインポートできないため、データベースを初期化するために Directory Manager としてログインしている必要があります。Directory Manager のみが、dc **=example,dc=com** などのルートエントリーにアクセスできます。

警告

LDIF ファイルからデータベースを初期化する場合は、データを復元しない限り、**o=NetscapeRoot** 接尾辞を上書きしないように注意してください。それ以外の場合は、データベースの初期化により情報が削除され、Directory Server の再インストールが必要になる場合があります。

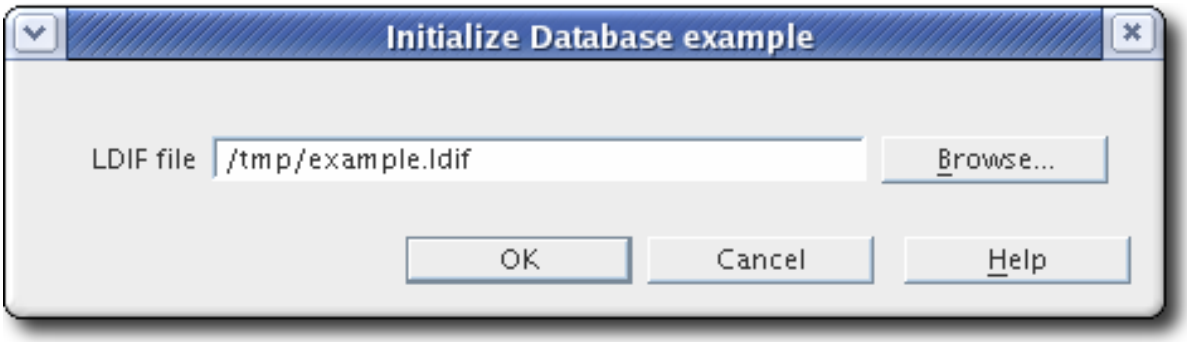
Directory Server コンソールを使用してデータベースを初期化するには、以下を実行します。

1. **Configuration** タブを選択します。
2. 左側のナビゲーションペインで **Data** tree を展開します。初期化するデータベースの接尾辞を展開し、データベース自体をクリックします。
3. データベースを右クリックし、**Initialize Database** を選択します。



または、**Object** メニューから **Initialize Database** を選択します。

4. **LDIF file** フィールドにインポートする LDIF ファイルへの完全パスを入力するか、 **Browse** をクリックします。



5. コンソールが、インポートされているファイルのマシンから実行中の場合は、**OK** をクリックして、即座にインポートに進みます。Console がマシンリモートから LDIF ファイルを含むサーバーに実行している場合は、以下のオプションのいずれかを選択して **OK** をクリックします。

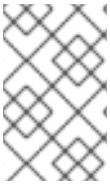
   - **ローカルマシンからいる。**LDIF ファイルがローカルマシンにあることを示します。
   - **サーバーマシン から。**LDIF ファイルがリモートサーバーにあることを示します。

   デフォルトの LDIF ディレクトリーは **/var/lib/dirsrv/slapd-*instance*/ldif** です。

6.1.4. コマンドラインからのインポート

コマンドラインでデータをインポートする方法は 4 つあります。

- **ldif2db の使用。**このインポートメソッドはデータベースの内容を上書きし、サーバーを停止する必要があります。「ldif2db コマンドラインユーティリティーを使用したインポート」 を参照してください。
- **Using ldif2db.pl.**このインポート方法は、サーバーの実行中にデータベースの内容を上書きします。「ldif2db.pl Perl スクリプトを使用したインポート」 を参照してください。
- **ldif2ldap の使用。**このメソッドは、LDAP を介して LDIF ファイルを追加します。このメソッドは、全データベースにデータを追加する場合に便利です。「ldif2ldap コマンドラインスクリプトを使用したインポート」 を参照してください。
- **cn=tasks エントリーの作成。**このメソッドは、インポート操作を自動的に起動する一時タスクエントリーを作成します。これは、**ldif2db** の実行に似ています。「cn=tasks エントリーを使用したインポート」 を参照してください。

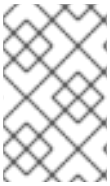


注記

インポート操作に使用される LDIF ファイルは、UTF-8 文字セットエンコーディングを使用する必要があります。インポート操作は、データをローカル文字セットエンコーディングから UTF-8 文字セットエンコーディングに変換しません。

警告

インポートされたすべての LDIF ファイルには、root 接尾辞も含まれている必要があります。



注記

暗号化したデータベースをインポートするには、スクリプトで **-E** オプションを使用します。詳細は、「暗号化したデータベースのエクスポートおよびインポート」 を参照してください。

6.1.4.1. ldif2db コマンドラインユーティリティーを使用したインポート

ldif2db スクリプトは、指定したデータベースのデータを上書きします。また、このスクリプトでは、インポートの開始時に Directory Server を停止する必要があります。

デフォルトでは、スクリプトは最初に保存してから、既存の **o=NetscapeRoot** 設定情報を、インポートするファイルの **o=NetscapeRoot** 設定情報とマージします。

警告

このスクリプトは、データベースのデータを上書きします。

LDIF をインポートするには、以下を行います。

1. サーバーを停止します。

```
# systemctl stop dirsrv@instance
```

2. **ldif2db** コマンドラインスクリプトを実行します。

```
# ldif2db -Z instance_name -n Database1 -i /var/lib/dirsrv/slapd-instance/ldif/demo.ldif -i
/var/lib/dirsrv/slapd-instance/ldif/demo2.ldif
```

この例で使用されるパラメーターの詳細は、Red『Hat Directory Server の設定、コマンド、およびファイルリファレンス の **ldif2db** スクリプトの説明を参照してください』。

警告

-n オプションで指定したデータベースが LDIF ファイルに含まれる接尾辞と一致しない場合は、データベースに含まれるすべてのデータが削除され、インポートに失敗します。データベース名が間違っていることがないことを確認してください。

3. サーバーを起動します。

```
# systemctl start dirsrv@instance
```

6.1.4.2. ldif2db.pl Perl スクリプトを使用したインポート

ldif2db スクリプトと同様に、**ldif2db.pl** スクリプトは、指定されたデータベースのデータを上書きします。このスクリプトでは、インポートを実行するためにサーバーを実行する必要があります。

警告

このスクリプトは、データベースのデータを上書きします。

ldif2db.pl スクリプトを実行します。

```
# ldif2db.pl -Z instance_name -D "cn=Directory Manager" -w secret -i
/var/lib/dirsrv/slapd-instance/ldif/demo.ldif -n Database1
```

この例で使用されるパラメーターの詳細は、Red『Hat Directory Server の設定、コマンド、およびファイルリファレンス の **ldif2db.pl** スクリプトの説明を参照してください』。

注記

スクリプトを実行するには **root** 権限は必要ありませんが、Directory Manager として認証する必要があります。

6.1.4.3. ldif2ldap コマンドラインスクリプトを使用したインポート

ldif2ldap スクリプトは、LDAP を介して LDIF ファイルを追加します。このスクリプトを使用すると、データはすべてのディレクトリーデータベースに同時にインポートされます。**ldif2ldap** を使用してインポートするには、サーバーが稼働している必要があります。

ldif2ldap を使用して LDIF をインポートするには、以下を実行します。

```
[root@server ~]# ldif2ldap -Z instance_name -D "cn=Directory Manager" -w secretpwd
/var/lib/dirsrv/slapd-instance/ldif/demo.ldif
```

ldif2ldap スクリプトでは、管理ユーザーの DN、管理ユーザーのパスワード、およびインポートする LDIF ファイルの絶対パスおよびファイル名が必要です。

この例で使用されるパラメーターの詳細は、Red『Hat Directory Server の設定、コマンド、およびファイルリファレンス の **ldif2ldap** スクリプトの説明を参照してください』。

6.1.4.4. cn=tasks エントリーを使用したインポート

Directory Server 設定の **cn=tasks,cn=config** エントリーは、サーバーがタスクの管理に使用する一時的なエントリーのコンテナーエントリーです。複数の共通ディレクトリータスクには、**cn=tasks,cn=config** の下にコンテナーエントリーがあります。一時タスクエントリーは、**cn=import,cn=tasks,cn=config** の下に作成し、インポート操作を開始できます。

ldif2db および **ldif2db.pl** スクリプトと同様に、**cn=tasks** のインポート操作はデータベースのすべての情報を上書きします。

このタスクエントリーには以下の 3 つの属性が必要です。

- 一意の名前(***cn***)
- インポートする LDIF ファイルのファイル名(***nsFilename***)
- ファイルをインポートするデータベースの名前(***nsInstance***)

ldif2db および ldif2db.pl スクリプトで、- **s** オプションおよび **-x** オプションと同様に、インポートを包含または除外する接尾辞の DN を指定することもできます。

エントリーは、「**ldapmodify** を使用したエントリーの追加」 で説明されているように **ldapmodify** を使用して追加されます。以下に例を示します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=example import,cn=import,cn=tasks,cn=config
```

```
changetype: add
objectclass: extensibleObject
cn: example import
nsFilename: /home/files/example.ldif
nsInstance: userRoot
nsIncludeSuffix: ou=People,dc=example,dc=com
nsExcludeSuffix: ou=Groups,dc=example,dc=com
```

タスクが完了するとすぐに、エントリーはディレクトリー設定から削除されます。

この例で使用される属性と、このエントリーに設定できるその他の属性の詳細は、Red Hat Directory Server 設定、コマンド、およびファイルリファレンス の **cn=import,cn=tasks,cn=config** エントリーの説明を参照してください。『』

6.2. データのエクスポート

LDAP データ交換形式 (LDIF) ファイルは、Directory Server データベースからデータベースエントリーをエクスポートするために使用されます。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、以下の場合に役に立ちます。

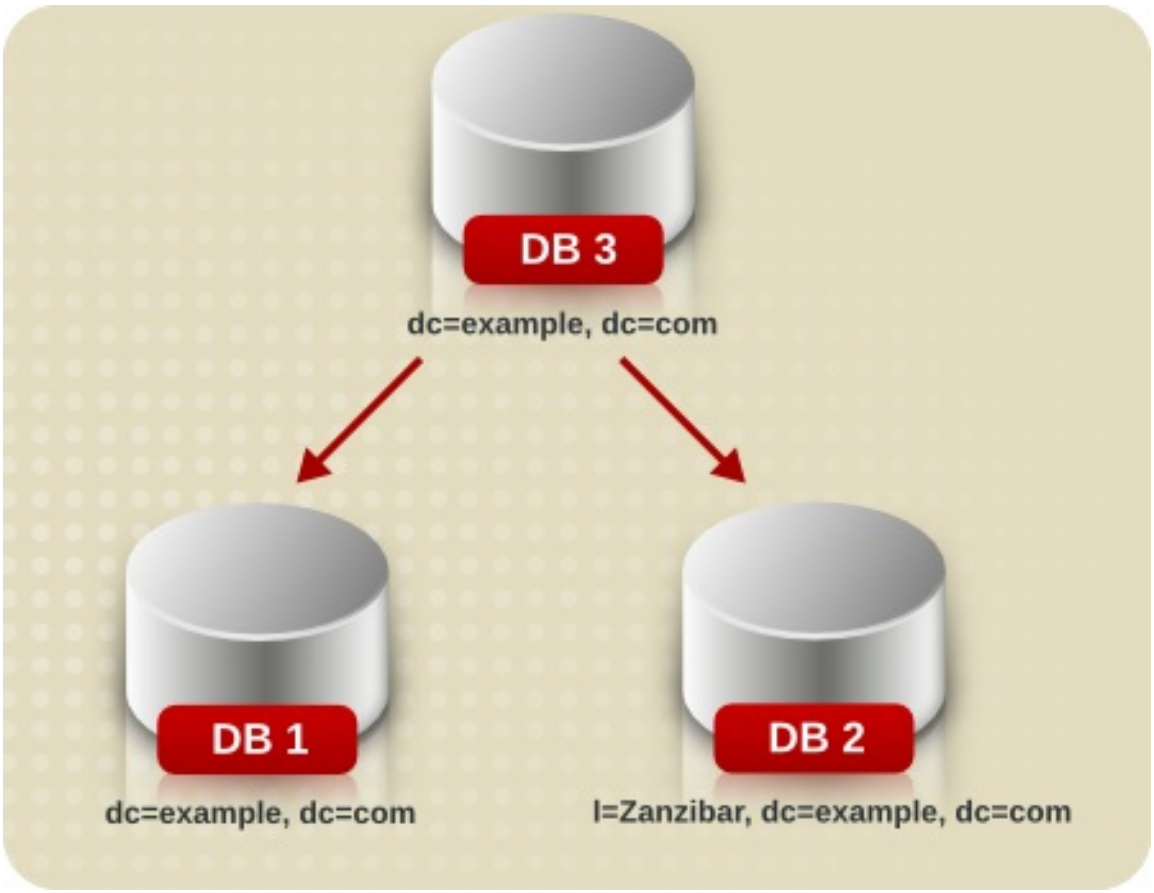
- データベースのデータのバックアップを作成します。
- 別の Directory Server にデータをコピーします。
- 別のアプリケーションへのデータのエクスポート。
- ディレクトリートポロジーの変更後にデータベースを再作成します。

たとえば、ディレクトリーに 1 つのデータベースが含まれ、その内容が 2 つのデータベースに分割されている場合、2 つの新しいデータベースは、古いデータベースの内容をエクスポートし、これを 2 つの新しいデータベースにインポートして、これを 2 つの新しいデータベースにインポートします。図 6.1「データベースコンテンツの 2 つのデータベースへの分割」

注記

エクスポート操作は、設定情報 (**cn=config**)、スキーマ情報 (**cn=schema**)、または監視情報 (**cn=monitor**) をエクスポートしません。

図6.1 データベースコンテンツの 2 つのデータベースへの分割



Directory Server コンソールまたはコマンドラインユーティリティーを使用して、データをエクスポートすることができます。

- 「コンソールを使用したディレクトリーデータの LDIF へのエクスポート」
- 「コンソールを使用した単一データベースの LDIF へのエクスポート」
- 「コマンドラインを使用した LDIF へのデータベースのエクスポート」

警告

エクスポート操作中はサーバーを停止しないでください。

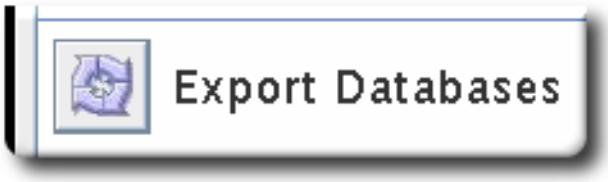
6.2.1. コンソールを使用したディレクトリーデータの LDIF へのエクスポート

最終的なエクスポートされたファイルの場所に応じて、一部またはすべてのディレクトリーデータを LDIF にエクスポートできます。LDIF ファイルがサーバーにある場合は、サーバーにローカルにあるデータベースに含まれるデータのみをエクスポートできます。LDIF ファイルがサーバーへのリモートである場合は、すべてのデータベースおよびデータベースリンクをエクスポートすることができます。

エクスポート操作を実行して、Directory Server Console または別のホストマシン（リモートエクスポート操作）にローカルとなるサーバーインスタンスからデータを取得できます。

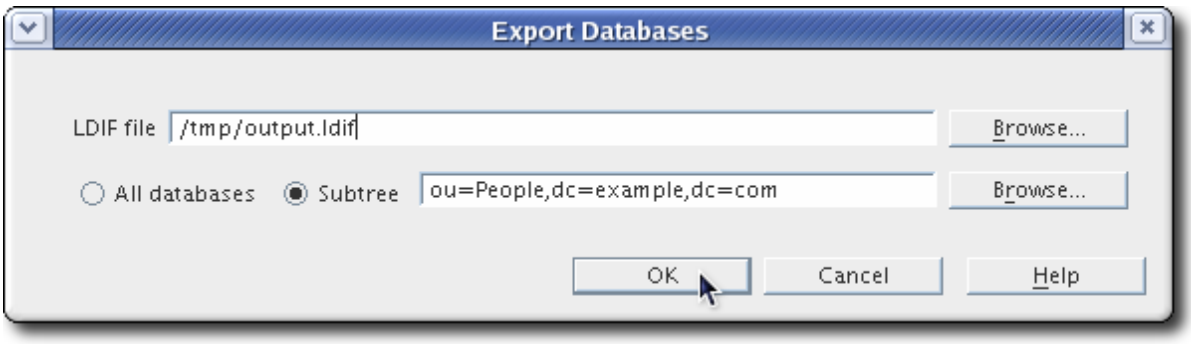
サーバーの実行中に、Directory Server Console から LDIF にディレクトリーデータをエクスポートします。

1. **Tasks** タブを選択します。画面の下部までスクロールし、**Export Database(s)** をクリックします。



   または、**Configuration** タブを選択して、**Console メニューから Export** をクリックします。

2. LDIF **File** フィールドに LDIF ファイルの完全パスおよびファイル名を入力するか、**Browse** をクリックしてファイルを見つけます。



   コンソールがリモートサーバーで実行している場合、**参照** は有効になりません。**参照** ボタンが有効になっていないと、ファイルはデフォルトのディレクトリー **/var/lib/dirsrv/slapd-*instance*/ldif** に保存されます。

3. Console がマシンのリモート上で稼働している場合は、**LDIF File** フィールドの下に 2 つのラジオボタンが表示されます。

   - **To local machine** を選択して、コンソールを実行しているマシンの LDIF ファイルにデータをエクスポートします。
   - **To server machine** to export to an LDIF file in the server's machine を選択します。

4. ディレクトリー全体をエクスポートするには、Entire **データベース** 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単一のサブツリーのみをエクスポートするには、**Subtree** ラジオボタンを選択してから、**Subtree** テキストボックスに接尾辞の名前を入力します。このオプションは、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します。

   または、**Browse** をクリックして接尾辞またはサブツリーを選択します。

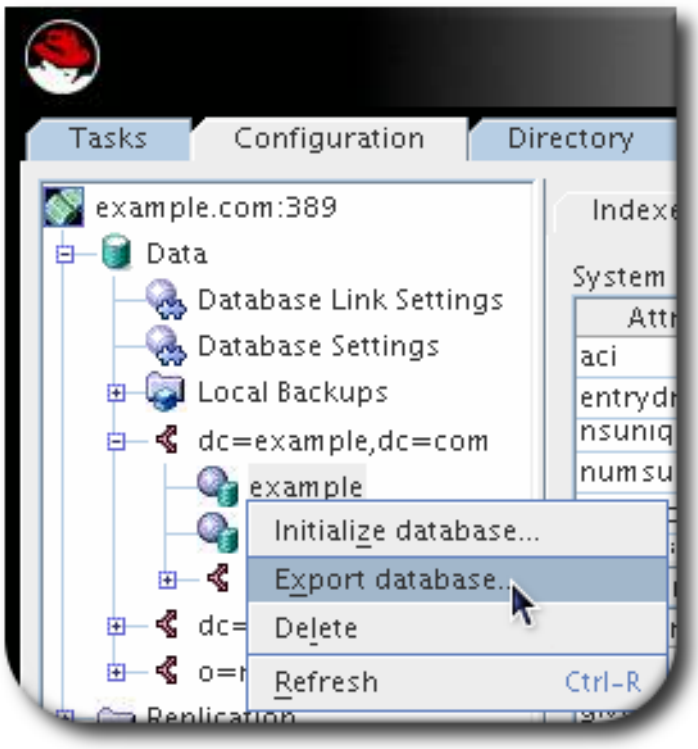
6.2.2. コンソールを使用した単一データベースの LDIF へのエクスポート

1 つのデータベースを LDIF にエクスポートすることもできます。サーバーの実行中に以下を行います。

1. **Configuration** タブを選択します。

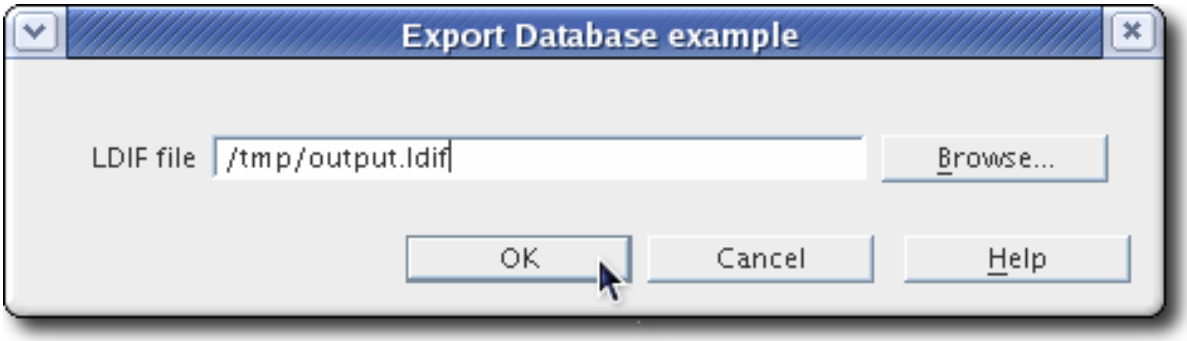
2. 左側のナビゲーションペインで **Data** tree を展開します。接尾辞を展開し、接尾辞の下にあるデータベースを選択します。

3. データベースを右クリックし、**Export Database** を選択します。



または、**Object** メニューから **Export Database** を選択します。

4. **LDIF file** フィールドで、LDIF ファイルへの完全パスを入力するか、**Browse** をクリックします。



参照 ボタンが有効になっていないと、ファイルはデフォルトのディレクトリー **/var/lib/dirsrv/slapd-*instance*/ldif** に保存されます。

6.2.3. コマンドラインを使用した LDIF へのデータベースのエクスポート

Directory Server は、データを LDIF ファイルにエクスポートする方法をサポートします。

6.2.3.1. Directory Server の実行中にデータベースのエクスポート

Directory Server の実行中にデータベースをエクスポートするには、エクスポートタスクを作成します。**db2ldif.pl** スクリプトを使用してこれを作成するか、手動でタスクを作成することができます。タスクが完了すると、Directory Server は、cn **=export,cn=tasks,cn=config** エントリーからタスクエン

トリーを自動的に削除します。

タスクエントリーの属性を設定する **db2ldif.pl** コマンドラインオプションを比較する場合は、『Red Hat Directory Server Configuration, Command, and File Referenceを参照してください』。

6.2.3.1.1. db2ldif.pl スクリプトを使用した データベースのエクスポート

db2ldif.pl スクリプトは、Directory Server の実行中にデータベースをエクスポートするタスクを作成します。たとえば、**userRoot** データベースをエクスポートするには、以下のコマンドを実行します。

```
# db2ldif.pl -Z instance_name -D "cn=Directory Manager" -w - -n userRoot
```

デフォルトでは、スクリプトは、エクスポートされたデータを **/var/lib/dirsrv/slapd-*instance_name*/ldif/** ディレクトリーに保存します。作成されたファイルには ***instance_name -database_or_suffix_name -time_ stamp*.ldif という名前が付けられ** ます。または、- **a *file_name*** オプションをスクリプトに渡して別の場所を設定することもできます。Directory Server ユーザーには、宛先ディレクトリーに書き込みパーミッションが必要になることに注意してください。

利用可能なコマンドラインオプションの詳細は、Red『Hat Directory Server の設定、コマンド、およびファイルリファレンス のスクリプトの説明を参照してください』。

暗号化されたデータベースをエクスポートするには、「暗号化したデータベースのエクスポートおよびインポート」 を参照してください。

6.2.3.1.2. エクスポートタスクの手動作成

db2ldif.pl スクリプトを使用してエクスポートタスクを作成する代わりに、タスクエントリーを手動で作成できます。たとえば、**userRoot** データベースを **/tmp/export.ldif** ファイルにエクスポートするタスクを作成するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=task_name,cn=export,cn=tasks,cn=config
objectclass: extensibleObject
cn: task_name
nsInstance: userRoot
nsFilename: /tmp/export.ldif
```

エクスポートタスクエントリーに使用できる設定の一覧は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス を参照してください』。

6.2.3.2. Directory Server が Stopped 中にデータベースのエクスポート

Directory Server インスタンスが停止している間にデータベースをエクスポートするには、db **2ldif** スクリプトを使用します。このスクリプトは、db **2ldif.pl** スクリプトと同じオプションを取ります。これは、インスタンスの実行中にデータをエクスポートできます。

たとえば、インスタンスの停止中に **userRoot データベースをエクスポート** するには、以下を実行します。

```
# db2ldif -Z instance_name -n userRoot
```

デフォルトでは、スクリプトは、エクスポートされたデータを **/var/lib/dirsrv/slapd-*instance_name*/ldif/** ディレクトリーに保存します。作成されたファイルには

instance_name -database_or_suffix_name -time_ stamp*.ldif という名前が付けられ** ます。または、**-a *file_name オプションをスクリプトに渡して別の場所を設定することもできます。Directory Server ユーザーには、宛先ディレクトリーに書き込みパーミッションが必要になることに注意してください。

利用可能なコマンドラインオプションの詳細は、Red『Hat Directory Server の設定、コマンド、およびファイルリファレンス のスクリプトの説明を参照してください』。

6.3. データのバックアップおよび復元

データベースは、Directory Server コンソールまたはコマンドラインスクリプトを使用してバックアップおよび復元できます。バックアップには以下が含まれます。以下に例を示します。

- それらのデータベース内に格納されているデータを含む、**userRoot** および **NetscapeRoot** などの全データベースファイル
- トランザクションログ
- インデックス

バックアップとは対照的に、「データのエクスポート」 の説明に従ってデータをエクスポートできます。エクスポート機能を使用して、LDAP Data Interchange Format (LDIF) 形式のサーバーからサブツリーなどの特定のデータをエクスポートします。

本セクションでは、以下の手順について説明します。

- 「すべてのデータベースのバックアップ」
- 「dse.ldif 設定ファイルのバックアップ」
- 「すべてのデータベースの復元」
- 「単一データベースの復元」
- 「複製されたエントリーが含まれるデータベースの復元」
- 「dse.ldif 設定ファイルの復元」

警告

バックアップまたは復元操作中にサーバーを停止しないでください。

6.3.1. すべてのデータベースのバックアップ

以下の手順では、Directory Server コンソールとコマンドラインから、ディレクトリー内のすべてのデータベースのバックアップを説明します。

注記

これらのバックアップメソッドは、データベースリンクを使用してチェーンされるリモートサーバーのデータベースに含まれるデータのバックアップには使用できません。

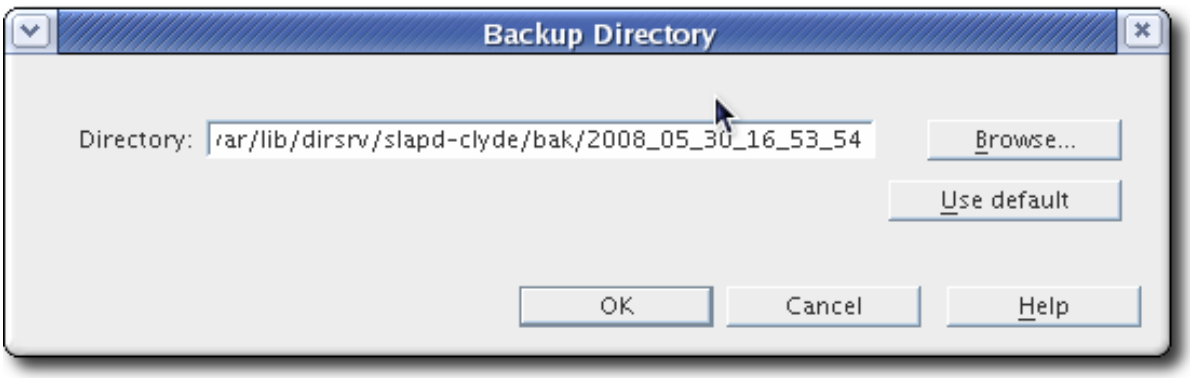
6.3.1.1. コンソールからのすべてのデータベースのバックアップ

Directory Server コンソールからデータベースをバックアップする場合、サーバーはすべてのデータベースコンテンツおよび関連するインデックスファイルをバックアップの場所にコピーします。サーバーの実行中にバックアップを実行できます。

Directory Server コンソールからデータベースをバックアップするには、以下を実行します。

1. **Tasks** 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。

3. **Directory** テキストボックスにバックアップファイルを保存するディレクトリーの完全パスを入力するか、**Use default** をクリックすると、サーバーはバックアップディレクトリーの名前を指定します。

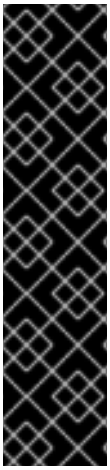


コンソールがディレクトリーと同じマシンで実行されている場合は、**Browse** をクリックしてローカルディレクトリーを選択します。

デフォルトの場所で、バックアップファイルは **/var/lib/dirsrv/slapd-*instance*/bak** に配置されます。デフォルトでは、バックアップディレクトリー名にはサーバーインスタンスの名前が含まれ、バックアップが作成された日時(instance-YYYY_MM_DD_hhmmss)が含まれます。

6.3.1.2. コマンドラインでのすべてのデータベースのバックアップ

データベースは、db2bak コマンドラインスクリプトまたは **db2bak.pl** Perl スクリプトを使用してコマンドラインからバックアップできます。コマンドラインスクリプトは、サーバーの実行時またはサーバーが停止しているときに機能します。Perl スクリプトは、サーバーが実行されているときにのみ使用できます。



重要

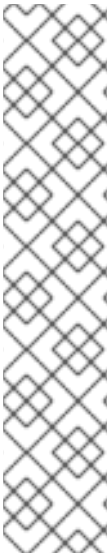
バックアップされるデータベースがマスターデータベース（つまり changelog）である場合は、db **2bak.pl** Perl スクリプトを使用してバックアップするか、サーバーが実行中の場合は Directory Server Console を使用してバックアップする必要があります。changelog は、サーバーのシャットダウン時に RUV エントリーをデータベースに書き込みます。サーバーの実行中に changelog はその変更をメモリーに保持します。Perl スクリプトとコンソールの場合、これらの changelog RUV は、バックアッププロセスの実行前にデータベースに書き込まれます。ただし、この手順はコマンドラインスクリプトでは実行されません。

db2bak は、実行中のマスターサーバーでは実行しないでください。Perl スクリプトを使用するか、またはサーバーを停止してからバックアップを実行します。

このバックアップ方法で設定情報をバックアップ **することはできません**。設定情報をバックアップする方法は、「dse.ldif 設定ファイルのバックアップ」 を参照してください。

db2bak.pl スクリプトを使用してコマンドラインからディレクトリーをバックアップするには、バックアップのファイル名とディレクトリーを指定して、Perl スクリプト **db2bak.pl** を実行します。

```
# db2bak.pl -Z instance_name -D "cn=Directory Manager" -w password -a /var/lib/dirsrv/slapd-example/bak/instance-2020_04_30_16_27_5-custom-name
```



注記

-a オプションを使用して **nsslapd-bakdir** ディレクティブで設定したデフォルトのバックアップディレクトリーを指定する場合は、末尾のスラッシュ("/")を使用しないでください。以下に例を示します。

```
# db2bak.pl -Z instance_name -D "cn=Directory Manager" -w password -a /var/lib/dirsrv/slapd-example/bak
```

slapd-example/bak の後にスラッシュがないことに注意してください。

この制限は、**nsslapd-bakdir** で設定されるディレクトリーと同じディレクトリーを指定する場合にのみ適用されます。デフォルトのバックアップディレクトリー（**bak/custom-name/**など）内であっても、他のディレクトリーは、末尾のスラッシュの有無でも指定できます。

サーバーがバックアップしたデータベースを保存するバックアップディレクトリーは、スクリプトで指定できます。ディレクトリーを指定しないと、バックアップファイルは **/var/lib/dirsrv/slapd-*instance*/bak** に保存されます。デフォルトでは、バックアップディレクトリーは Directory Server インスタンス名とバックアップの日付(**serverID-YYYY_MM_DD_hhmmss**)で名前が付けられます。

ldif2db の詳細は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス のスクリプトの説明を参照してください』。

6.3.1.3. cn=tasks エントリーを使用したデータベースのバックアップ

Directory Server 設定の **cn=tasks,cn=config** エントリーは、サーバーがタスクの管理に使用する一時的なエントリーのコンテナーエントリーです。複数の共通ディレクトリータスクには、**cn=tasks,cn=config** の下にコンテナーエントリーがあります。一時タスクエントリーは、**cn=backup,cn=tasks,cn=config** の下に作成し、バックアップ操作を開始できます。

バックアップタスクエントリーには以下の 3 つの属性が必要です。

- 一意の名前(***cn***)
- バックアップファイルを書き込むディレクトリー(***nsArchiveDir***)。バックアップファイルには、Directory Server インスタンス名とバックアップの日付(**serverID-YYYY_MM_DD_hhmmss**)という名前が付けられます。
- データベースのタイプ(***nsDatabaseType***)。唯一のオプションは **ldbm データベース** です。

エントリーは、「**ldapmodify** を使用したエントリーの追加」 で説明されているように **ldapmodify** を使用して追加されます。以下に例を示します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=example backup,cn=backup,cn=tasks,cn=config
changetype: add
objectclass: extensibleObject
cn: example backup
nsArchiveDir: /export/backups/
nsDatabaseType: ldbm database
```

タスクが完了するとすぐに、エントリーはディレクトリー設定から削除されます。

この例で使用される属性と、このエントリーに設定できるその他の属性の詳細は、Red Hat Directory Server 設定、コマンド、およびファイルリファレンス の **cn=backup,cn=tasks,cn=config** エントリーの説明を参照してください。『』

6.3.2. dse.ldif 設定ファイルのバックアップ

Directory Server は、**dse.ldif** 設定ファイルを自動的にバックアップします。Directory Server が起動すると、ディレクトリーは、**/etc/dirsrv/slapd-*instance*** ディレクトリーの **dse.ldif.startOK** という名前のファイルに dse.ldif ファイルのバックアップ **を自動的に作成します**。

dse.ldif ファイルが変更されると、ファイルが最初に **/etc/dirsrv/slapd-*instance*** ディレクトリーの **dse.ldif.bak** というファイルにバックアップされ、ディレクトリーが **dse.ldif** ファイルに変更が書き込まれます。

6.3.3. すべてのデータベースの復元

以下の手順では、Directory Server コンソールとコマンドラインから、ディレクトリー内のすべてのデータベースを復元する方法を説明します。

注記

バックアップからデータベースを復元すると、changelog も復元します。

重要

データベースの復元中、サーバーが稼働している必要があります。ただし、復元中にデータベースが処理操作の対象になります。

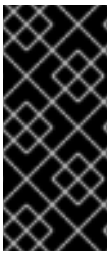
したがって、データベースを復元する前に、すべてのレプリケーションプロセスを停止します。詳細は、「レプリカ合意の無効化および再有効化」 を参照してください。

6.3.3.1. コンソールからのすべてのデータベースの復元

データベースが破損したら、Directory Server コンソールを使用して、以前に生成されたバックアップからデータを復元します。このプロセスは、サーバーを停止し、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の場所からデータベースディレクトリーにコピーすることで構成されます。

警告

データベースの復元は、既存のデータベースファイルを上書きします。



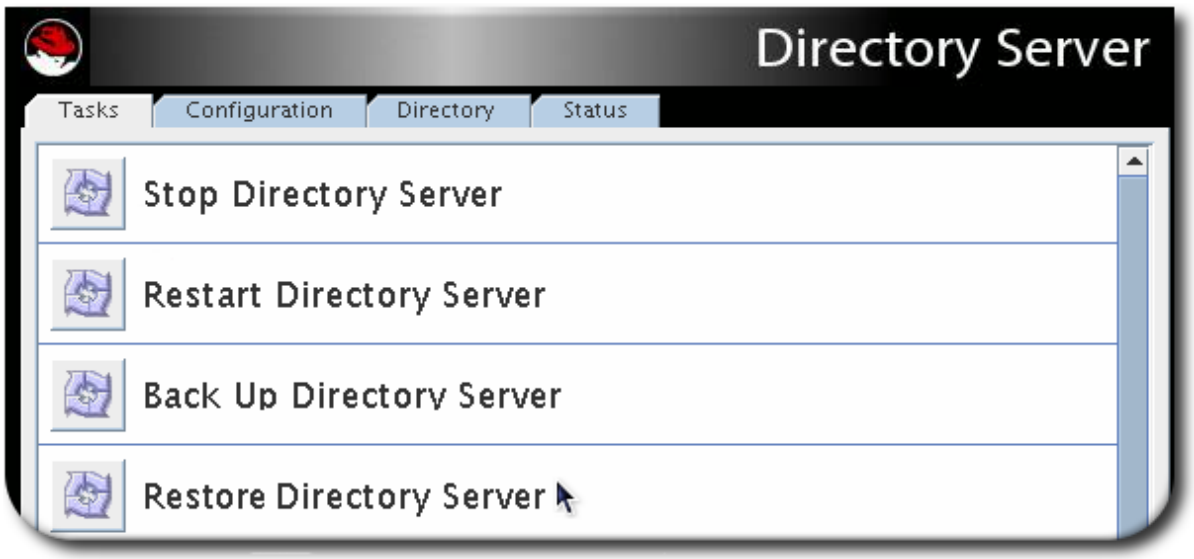
重要

データベースの復元中、サーバーが稼働している必要があります。ただし、復元中にデータベースが処理操作の対象になります。

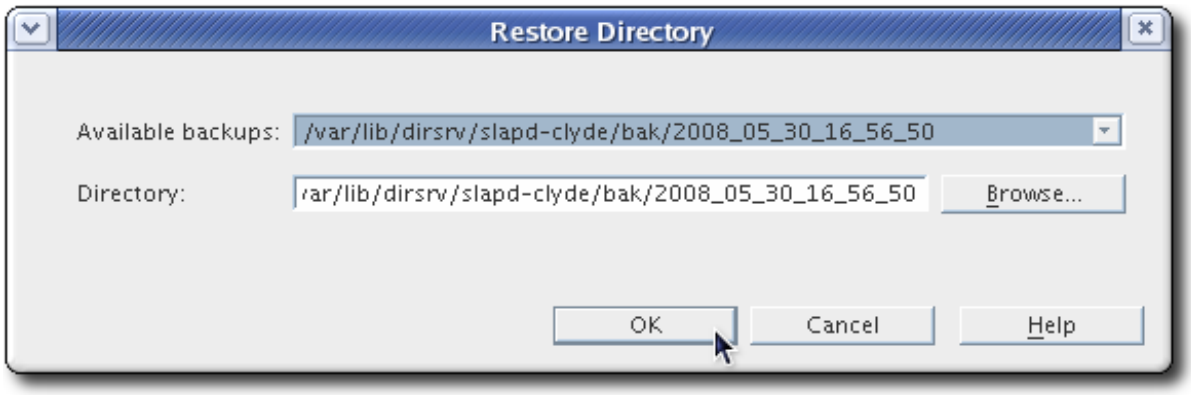
したがって、データベースを復元する前に、すべてのレプリケーションプロセスを停止します。詳細は、「レプリカ合意の無効化および再有効化」 を参照してください。

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、以下を行います。

1. Directory Server コンソールで、Tasks **タブを選択します**。
2. **Directory Server のリストア** をクリックします。



3. **Available Backups** リストからバックアップを選択するか、**Directory** テキストボックスに既存のバックアップへの完全パスを入力します。



利用可能なバックアップの一覧には、デフォルトのディレクトリー **/var/lib/dirsrv/slapd-*instance*/bak/backup_directory** にあるバックアップがすべて表示されます。**backup_directory** は、**serverID-YYYY_MM_DD_hhmmss** 形式の最新のバックアップのディレクトリーです。

6.3.3.2. コマンドラインでのデータベースの復元

コマンドラインからデータベースを復元する方法は 3 つあります。

- **bak2db** コマンドラインスクリプトの使用このスクリプトでは、サーバーをシャットダウンする必要があります。
- Perl スクリプト **bak2db.pl** の使用。このスクリプトは、サーバーの実行中に機能します。
- **cn=restore,cn=tasks,cn=config** に一時的なエントリーを作成します。この方法は、サーバーの実行中に実行することもできます。

重要

データベースの復元中はサーバーが実行されている必要があります（**bak2db** コマンドラインスクリプトの実行を除く）。ただし、復元中にデータベースが処理操作の対象になります。

したがって、データベースを復元する前に、すべてのレプリケーションプロセスを停止します。詳細は、「レプリカ合意の無効化および再有効化」 を参照してください。

6.3.3.2.1. bak2db コマンドラインユーティリティーの使用

1. Directory Server を実行している場合は、停止します。

```
# systemctl stop dirsrv@instance
```

2. **bak2db** コマンドラインスクリプトを実行します。**bak2db** スクリプトには、入力ファイルの完全パスおよび名前が必要です。

```
# bak2db -Z instance_name /var/lib/dirsrv/slapd-instance/bak/instance-
2020_04_30_11_48_30
```

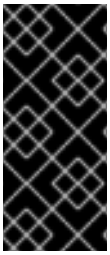
この例で使用されるパラメーターの詳細は、Red 『Hat Directory Server の設定、コマンド、およびファイルリファレンス の **bak2db** スクリプトの説明を参照してください』。

6.3.3.2.2. bak2db.pl Perl スクリプトの使用

bak2db.pl Perl スクリプトを実行します。

```
# bak2db.pl -Z instance_name -D "cn=Directory Manager" -w secret -a
/var/lib/dirsrv/slapd-instance/bak/instance-2020_04_30_11_48_30
```

この例で使用されるパラメーターの詳細は、Red『Hat Directory Server の設定、コマンド、および ファイルリファレンス の **bak2db.pl** スクリプトの説明を参照してください』。



重要

データベースの復元中、サーバーが稼働している必要があります。ただし、復元中にデータベースが処理操作の対象になります。

したがって、データベースを復元する前に、すべてのレプリケーションプロセスを停止します。詳細は、「レプリカ合意の無効化および再有効化」 を参照してください。

6.3.3.2.3. cn=tasks エントリーを使用したデータベースの復元

Directory Server 設定の **cn=tasks,cn=config** エントリーは、サーバーがタスクの管理に使用する一時的なエントリーのコンテナーエントリーです。複数の共通ディレクトリータスクには、**cn=tasks,cn=config** の下にコンテナーエントリーがあります。一時タスクエントリーは、**cn=restore,cn=tasks,cn=config** の下に作成し、復元操作を開始できます。

重要

データベースの復元中、サーバーが稼働している必要があります。ただし、復元中にデータベースが処理操作の対象になります。

したがって、データベースを復元する前に、すべてのレプリケーションプロセスを停止します。詳細は、「レプリカ合意の無効化および再有効化」 を参照してください。

復元タスクエントリーには、バックアップタスクと同じ 3 つの属性が必要です。

- 一意の名前(***cn***)
- バックアップファイルを取得するディレクトリー(***nsArchiveDir***)
- データベースのタイプ(***nsDatabaseType***)。唯一のオプションは **ldbm データベース** です。

エントリーは、「**ldapmodify** を使用したエントリーの追加」 で説明されているように **ldapmodify** を使用して追加されます。以下に例を示します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=example restore,cn=restore,cn=tasks,cn=config
changetype: add
objectclass: extensibleObject
cn: example restore
nsArchiveDir: /export/backups/
nsDatabaseType: ldbm database
```

タスクが完了するとすぐに、エントリーはディレクトリー設定から削除されます。

この例で使用される属性と、このエントリーに設定できるその他の属性の詳細は、Red Hat Directory Server 設定、コマンド、およびファイルリファレンス の **cn=restore,cn=tasks,cn=config** エントリーの説明を参照してください。『』

6.3.4. 単一データベースの復元

Directory Server コンソールではなく、コマンドラインで単一のデータベースを復元することが可能です。単一データベースを復元するには、以下を実行します。

1. Directory Server が実行している場合は停止します。

```
# systemctl stop dirsrv@instance
```

2. **-n** パラメーターを使用してデータベース名を指定し 、 **/var/lib/dirsrv/slapd-*instance*/bak** アーカイブからバックエンドを復元します。以下に例を示します。

```
# bak2db -Z instance_name /var/lib/dirsrv/slapd-instance/bak/backup_file -n userRoot
```

3. Directory Server を再起動します。

```
# systemctl start dirsrv@instance
```

注記

Directory Server の起動に失敗した場合は、**/var/lib/dirsrv/slapd-*instance*/db/log.###** のデータベーストランザクションログファイルを削除してから、サーバーの起動を再試行します。

6.3.5. 複製されたエントリーが含まれるデータベースの復元

サプライヤーサーバーを復元すると、いくつかの状況が発生する可能性があります。

- コンシューマーサーバーも復元します。

  非常にまれな状況では、すべてのデータベースで、(データが同期されるため)、コンシューマーはサプライヤーと同期したままとなり、他に何もする必要はありません。レプリケーションは中断せずに再開します。

- サプライヤーだけが復元します。

  サプライヤーのみが復元された場合や、コンシューマーが別のバックアップから復元された場合は、サプライヤーがコンシューマーを再初期化して、データベースのデータを更新します。サプライヤーのみが復元された場合や、コンシューマーが別のバックアップから復元された場合は、サプライヤーがコンシューマーを再初期化して、データベースのデータを更新します。

- サプライヤーサーバーでチェンジログエントリーの有効期限が切れていません。

  データベースのバックアップの取得後にサプライヤーの変更ログが期限切れになっていない場合は、ローカルコンシューマーを復元し、通常の操作を継続します。この状態は、**cn=changelog5,cn=config** エントリーで、最大 changelog age 属性 ***nsslapd-changelogmaxage*** に設定された値よりも短い期間内にバックアップを取得した場合に限り発生します。このオプションの詳細は、『Red Hat Directory Server 設定、コマンド、およびファイルリファレンス』 を参照してください。

Directory Server は、レプリカとその changelog 間の互換性を自動的に検出します。不一致が検出されると、サーバーは古い changelog ファイルを削除し、空のファイルを新たに作成します。

- changelog エントリーは、ローカルバックアップの時間以降、サプライヤーサーバーで期限切れです。

  changelog エントリーの有効期限が切れている場合は、コンシューマーを再初期化します。コンシューマーの再初期化に関する詳細は、「コンシューマーの初期化」 を参照してください。

例6.1 Directory Server のレプリケーショントポロジーの復元

たとえば、2 つのマスターと 2 つのコンシューマーサーバーで構成されるレプリケーション環境のサーバーをすべて復元するには、以下を実行します。

1. 最初のマスターを復元します。**-r** オプションを指定せずに **ldif2db** ユーティリティーを使用して、データをインポートします。「コマンドラインからのインポート」 を参照してください。

2. レプリケーションを使用して残りのサーバーをオンラインに初期化します。

   a. 最初のマスターから 2 番目のマスターを初期化します。

   b. マスターからコンシューマーを初期化します。

   詳細は、「コンシューマーの初期化」 を参照してください。

3. 各サーバーで ***nsds5replicaLastUpdateStatus*** 属性を表示し、レプリケーションが正しく機能していることを確認します。

   ```
   # ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b
   "cn=example_agreement,cn=replica,cn=dc\=example\,dc\=com,cn=mapping
   tree,cn=config" nsds5replicaLastUpdateStatus
   ```

   可能なステータスの詳細は、Red 『Hat Directory Server の設定、コマンド、およびファイルリファレンス の付録 『レプリカ合意の状況』 を参照してください』。

復元操作中に、復元されたデータベースに関連する changelog が削除されます。再初期化が必要であることを示すメッセージが、サプライヤーサーバーのログファイルに記録されます。

レプリケーションの管理に関する情報は、「15章 *レプリケーションの管理*」を参照してください。

6.3.6. dse.ldif 設定ファイルの復元

ディレクトリーは、**/etc/dirsrv/slapd-*instance*** ディレクトリーの **dse.ldif** ファイルの 2 つのバックアップコピーを作成します。**dse.ldif.startOK** ファイルは、サーバーの起動時に **dse.ldif** ファイルのコピーを記録します。**dse.ldif.bak** ファイルには、**dse.ldif** ファイルへの最新の変更のバックアップが含まれます。最新の変更でバージョンを使用して、ディレクトリーを復元します。

dse.ldif 設定ファイルを復元するには、以下を実行します。

1. サーバーを停止します。

   ```
   # systemctl stop dirsrv@instance
   ```

2. 「単一データベースの復元」 で説明されているようにデータベースを復元し、**dse.ldif** ファイルのバックアップコピーをそのディレクトリーにコピーします。

3. サービスを再起動します。

```
# systemctl restart dirsrv@instance
```

第7章 属性および値の管理

Red Hat Directory Server は、ディレクトリーエントリーで一部の属性タイプを動的かつ自動的に維持するためのさまざまなメカニズムいくつか提供します。これらのプラグインおよび設定オプションを使用すると、ディレクトリーデータの管理やエントリー間の関係の表現が容易になります。

エントリーの特徴の一部は、相互 **関係** です。とうぜん。マネージャーには従業員がいるため、この 2 つのエントリーには関連性があります。グループはメンバーに関連付けられます。共通の物理的な場所を共有するエントリー間のように、あまり明白でない関係もあります。

Red Hat Directory Server は、このようなエントリー間の関係をスムーズにかつ一貫して維持する方法を複数提供します。複数のプラグインは、ディレクトリー内のデータの一部として属性を自動的に適用または生成できます。これには、サービスのクラス、属性のリンク、一意の数値属性値の生成が含まれます。

7.1. 属性の一意性の有効化

ディレクトリーまたはサブツリー全体で属性の値が一意になるように、**Attribute Uniqueness** プラグインを使用します。

複数の属性を一意にしたい場合や、異なる条件を使用する場合は、プラグインに複数の設定レコードを作成します。

7.1.1. Attribute Uniqueness プラグインの新規設定レコードの作成

値が一意である必要がある属性ごとに、**Attribute Uniqueness** プラグインの新しい設定レコードを作成します。

注記

コマンドラインからプラグインの新しい設定レコードのみを作成できます。

***Example* Attribute Uniqueness** という名前のプラグインの設定解除および無効にした新しい設定レコードを作成するには、以下を実行します。

```
# ldapadd -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=Example Attribute Uniqueness,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: Example Attribute Uniqueness
nsslapd-pluginPath: libattr-unique-plugin
nsslapd-pluginInitfunc: NSUniqueAttr_Init
nsslapd-pluginType: betxnpreoperation
nsslapd-pluginEnabled: off
nsslapd-plugin-depends-on-type: database
nsslapd-pluginId: NSUniqueAttr
nsslapd-pluginVersion: none
nsslapd-pluginVendor: 389 Project
nsslapd-pluginDescription: Enforce unique attribute values
uniqueness-attribute-name: uid
```

7.1.2. サフィックスまたはサブツリーにおける属性一意の設定

Attribute Uniqueness プラグインを設定して、特定のサフィックス、サブツリー、またはサフィックスおよびサブツリーで属性の値が一意になるようにすることができます。

7.1.2.1. コマンドラインでサフィックスまたはサブツリーに対する属性一意の設定

たとえば、**mail** 属性に保存される値が一意となるように設定するには、以下を実行します。

1. たとえば **mail Attribute Uniqueness** という名前の Attribute Uniqueness プラグインの新たな設定レコードを作成します。詳細は 「Attribute Uniqueness プラグインの新規設定レコードの作成」を参照してください。

2. プラグイン設定レコードを有効にし、***mail*** 属性に保存される値が内部で一意である必要があります。たとえば、**ou=Engineering,dc=example,dc=com および ou=Sales, dc=example,dc=com** サブツリーなどです。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=mail Attribute Uniqueness,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
-
add: uniqueness-attribute-name
uniqueness-attribute-name: mail
-
add: uniqueness-subtrees
uniqueness-subtrees: ou=Engineering,dc=example,dc=com
uniqueness-subtrees: ou=Sales,dc=example,dc=com
```

3. 必要に応じて、このプラグイン設定レコードに設定されたすべてのサブツリーで一意性を設定するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=mail Attribute Uniqueness,cn=plugins,cn=config
changetype: modify
add: uniqueness-across-all-subtrees
uniqueness-across-all-subtrees: on
```

4. インスタンスを再起動します。

```
# systemctl restart dirsrv@instance_name
```

7.1.2.2. コンソールを使用したサフィックスまたはサブツリーに対する属性一意の設定

たとえば、**mail** 属性に保存される値が一意となるように設定するには、以下を実行します。

1. Attribute Uniqueness プラグインの新しい設定レコードを作成します。「Attribute Uniqueness プラグインの新規設定レコードの作成」 を参照してください。

2. プラグイン設定レコードの設定で Property Editor を開きます。詳細は、「コンソールを使用したプラグインの設定」 を参照してください。

3. プラグインを有効にするには、以下を設定します。

```
nsslapd-pluginEnabled: on
```

4. **mail** 属性を一意である必要があります。

```
uniqueness-attribute-name: mail
```

5. 属性の値が一意である必要があるサブツリーを設定します。

```
uniqueness-subtrees: ou=Engineering,dc=example,dc=com
uniqueness-subtrees: ou=Sales,dc=example,dc=com
```

uniqueness-subtrees 属性の値フィールドを選択し、値を追加 **ボタンをクリックして** 2 番目の ***uniqueness-subtrees*** 属性を追加します。

6. 必要に応じて、このプラグイン設定レコードに設定されたすべてのサブツリーで一意性を設定するには、***uniqueness-across-all-subtrees*** 属性を追加し、これを以下に設定します。

```
uniqueness-across-all-subtrees: on
```

7. **OK をクリックして、Property Editorを閉じます。**

8. Directory Server インスタンスを再起動します。「コンソールを使用した Directory Server インスタンスの起動および停止」 を参照してください。

7.1.3. オブジェクトクラスに対する属性の一意性の設定

Attribute Uniqueness プラグインを設定して、特定のオブジェクトクラスが含まれるサブツリーエントリーで属性の値が一意になるようにすることができます。Directory Server は、更新されたオブジェクトの親エントリーでこのオブジェクトクラスを検索します。Directory Server でオブジェクトクラスが見つからなかった場合、検索はディレクトリーツリーのルートまで次の上位レベルのエントリーで続行されます。オブジェクトクラスが見つかった場合、Directory Server は、*uniqueness-attribute-name* に設定された属性の値がこのサブツリー内で一意であることを確認します。

注記

このシナリオは、コマンドラインのみを使用して設定できます。

たとえば、mail 属性に保存されている値が、nsContainer オブジェクトクラスが含まれるエントリーで一意となるように設定するには、以下を実行します。

1. **たとえば mail Attribute Uniqueness という名前の Attribute Uniqueness プラグインの新たな設定レコードを作成します。詳細は** **「Attribute Uniqueness プラグインの新規設定レコードの作成」を参照してください。**

2. **プラグイン設定レコードを有効にし、*mail* 属性に保存される値は、nsContainer オブジェクトクラスが含まれるエントリーで一意である必要があります。**

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=mail Attribute Uniqueness,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

```
-
add: uniqueness-top-entry-oc
uniqueness-top-entry-oc: nsContainer
```

3. 必要に応じて、チェックされるオブジェクトの範囲を制限できます。サーバーが **nsContainer** オブジェクトクラスを含むエントリーの下にあるエントリーのサブセットのみをチェックするようにするには、***uniqueness-subtree-entries-oc*** パラメーターに追加のオブジェクトクラスを設定します。この追加クラスも存在している必要があります。

   たとえば、**nsContainer** オブジェクトクラスセットが含まれるエントリーにあるすべてのエントリーで ***mail*** 属性を一意にする必要があります。ただし、プラグインは、**inetOrgPerson** など、この属性を提供するオブジェクトクラスが含まれるエントリーで ***mail*** のみを検索します。この場合は、以下を入力します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=mail Attribute Uniqueness,cn=plugins,cn=config
add: uniqueness-subtree-entries-oc
uniqueness-subtree-entries-oc: inetOrgPerson
```

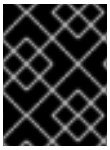
4. インスタンスを再起動します。

```
# systemctl restart dirsrv@instance_name
```

7.1.4. 属性の一意性プラグイン設定パラメーター

Attribute Uniqueness プラグインの設定レコードを設定するには、**cn=*attribute_uniqueness_configuration_record_name*,cn=plugins,cn=config** エントリーでプラグインの設定属性を設定します。

新しいプラグイン固有の属性名(例7.1「プラグイン固有の属性を使用した属性の一意性プラグイン設定」)を使用するか、非推奨の ***nsslapd-plugarg**** 属性(例7.2「***nsslapd-pluginarg**** 属性を使用した属性の一意性プラグイン設定」)を使用して、このプラグインを設定できます。



重要

Red Hat では、プラグイン固有の属性名のみを使用して Attribute Uniqueness プラグインを設定 することを推奨します。

例7.1 プラグイン固有の属性を使用した属性の一意性プラグイン設定

```
dn: cn=Example Attribute Uniqueness,cn=plugins,cn=config
nsslapd-pluginEnabled: on
uniqueness-attribute-name: attribute_name
uniqueness-top-entry-oc: objectclass1
uniqueness-subtree-entries-oc: objectclass2
```

例7.2 *nsslapd-pluginarg 属性を使用した属性の一意性プラグイン設定**

```
dn: cn=Example Attribute Uniqueness,cn=plugins,cn=config
nsslapd-pluginEnabled: on
```

```
nsslapd-pluginarg0: attribute=mail
nsslapd-pluginarg1: markerObjectClass=objectclass1
nsslapd-pluginarg2: requiredObjectClass=objectclass2
```

表7.1 属性の一意性プラグイン設定パラメーター

パラメーター	新規または古い構文	定義
cn	both	Attribute Uniqueness プラグインの設定レコードの名前を設定します。どの文字列も使用できますが、Red Hat では設定レコードの ***attribute_name* Attribute Uniqueness という名前を付けることを推奨します。**
nsslapd-pluginEnabled	both	プラグイン設定レコードを**有効**（オン）または無効（無効）します。
uniqueness-attribute-name	新規	値が一意である必要がある属性の名前を設定します。この属性は多値です。
uniqueness-subtrees	新規	プラグインが属性の値の一意性をチェックする DN を設定します。この属性は多値です。
uniqueness-across-all-subtrees	新規	有効な場合は(**on**)、プラグインは属性セット全体で属性が一意であることを確認します。属性を **off** に設定すると、一意性は更新されたエントリーのサブツリー内でのみ適用されます。
uniqueness-top-entry-oc	新規	Directory Server は、更新されたオブジェクトの親エントリーでこのオブジェクトクラスを検索します。これが見つからない場合、検索はディレクトリーツリーのルートまでの次のレベルエントリーで続行されます。オブジェクトクラスが見つかった場合、Directory Server は、***uniqueness-attribute-name*** に設定された属性の値がこのサブツリー内で一意であることを確認します。
uniqueness-subtree-entries-oc	新規	任意で、***uniqueness-top-entry-oc*** パラメーターを使用する場合は、エントリーにこのパラメーターに設定されたオブジェクトクラスが含まれる場合に限り、Attribute Uniqueness プラグインが属性が一意であるかどうかを確認することができます。詳細は、「オブジェクトクラスに対する属性の一意性の設定」 を参照してください。

パラメーター	新規または古い構文	定義
nsslapd-pluginarg0	old	この ***nsslapd-pluginarg**** パラメーターと同等のプラグイン固有の属性は ***uniqueness-attribute-name*** です。説明については、このパラメーターを参照してください。 属性を attribute ***=attribute_name*** に設定します。
nsslapd-pluginarg[1-9]	old	この ***nsslapd-pluginarg**** パラメーターと同等のプラグイン固有の属性は ***uniqueness-top-entry-oc*** です。説明については、このパラメーターを参照してください。 属性を **markerObjectClass=*object_class*** に設定します。
nsslapd-pluginarg[1-9]	old	同等のプラグイン固有の属性は ***uniqueness-subtree-entries-oc*** です。説明については、このパラメーターを参照してください。 属性を **requiredObjectClass=*object_class*** に設定します。

7.2. サービスのクラスの割り当て

サービス定義 (CoS) は、アプリケーションに透過的な方法でエントリー間で属性を共有します。CoS はエントリー管理を簡素化し、ストレージ要件を削減します。

Directory Server のクライアントは、ユーザーのエントリーの属性を読み取ります。CoS では、一部の属性値はエントリー自体に保存されない可能性があります。代わりに、エントリーがクライアントアプリケーションに送信されるため、これらの属性の値はサービスロジックのクラスによって生成されます。

各 CoS は、ディレクトリー内に 2 種類のエントリーで構成されます。

- **CoS 定義エントリー。CoS 定義エントリーは、使用される CoS のタイプを識別します。ロール定義エントリーと同様に、LDAPsubentry オブジェクトクラスから継承されます。CoS 定義エントリーは、有効なブランチの下にあります。**
- **テンプレートエントリー。CoS テンプレートエントリーには、共有属性値の一覧が含まれます。テンプレートエントリー属性値への変更は、CoS の範囲内のすべてのエントリーに自動的に適用されます。1 つの CoS に、複数のテンプレートエントリーが関連付けられている場合があります。**

CoS 定義エントリーとテンプレートエントリーは、CoS の範囲内で任意のターゲットエントリーに属性情報を提供するために対話します。

7.2.1. CoS 定義エントリーの概要

CoS 定義エントリーは、cosSuperDefinition オブジェクトクラスのインスタンスです。CoS 定義エントリーには、エントリーを生成するために使用するテンプレートエントリーのタイプを指定する 3 つのオブジェクトクラスのいずれか 1 つも含まれています。CoS と対話するターゲットエントリーは、CoS 定義エントリーと同じ親を共有します。

CoS には 3 つのタイプの CoS 定義エントリーを使用して定義されます。

- ポインター CoS。ポインター CoS は、テンプレート DN のみを使用してテンプレートエントリーを特定します。
- 間接的な CoS。間接 CoS は、ターゲットエントリーの属性の 1 つを使用してテンプレートエントリーを識別します。たとえば、間接的な CoS はターゲットエントリーの *manager* 属性を指定する場合があります。次に、*manager* 属性の値を使用してテンプレートエントリーを特定します。

  ターゲットエントリーの属性は単値であり、DN が含まれる必要があります。
- Classic CoS。Classic CoS は、テンプレートエントリーのベース DN とターゲットエントリーの属性の 1 つの値を使用してテンプレートエントリーを特定します。

CoS の各タイプのオブジェクトクラスおよび属性に関する詳細は、「コマンドラインでの CoS の管理」を参照してください。

CoS ロジックが、CoS が値を生成する属性を含むことを検出すると、デフォルトでは CoS が値を生成する属性を検出すると、クライアントアプリケーションにエントリー自体の属性値が提供されます。ただし、CoS 定義エントリーはこの動作を制御できます。

7.2.2. CoS テンプレートエントリーの概要

CoS テンプレートエントリーには、CoS ロジックによって生成された属性の値 (1 つまたは複数) が含まれます。CoS テンプレートエントリーには、一般的なオブジェクトクラス **cosTemplate** が含まれます。特定の CoS テンプレートエントリーは、CoS 定義とともにディレクトリーツリーに保存されます。

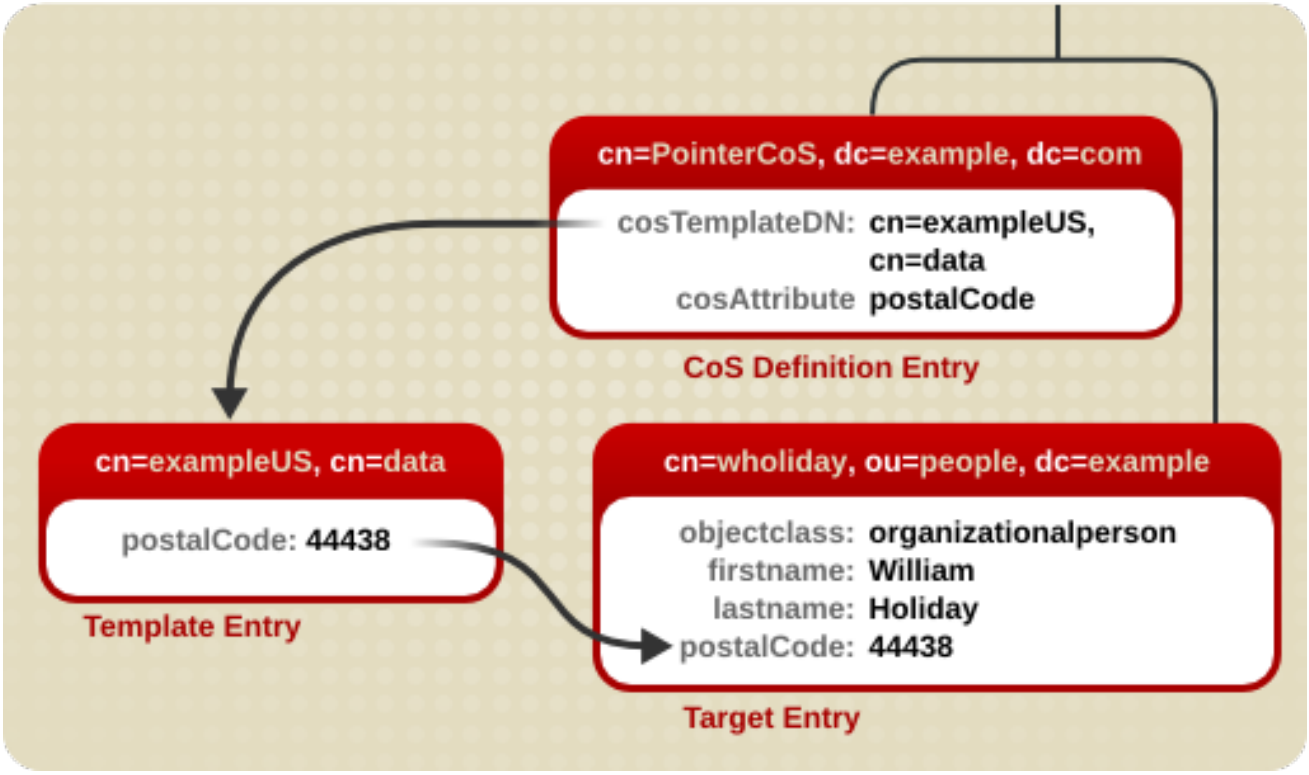
テンプレートエントリーの相対識別名 (RDN) は、以下のいずれかで決定されます。

- テンプレートエントリーの DN のみ。このタイプのテンプレートは Pointer CoS 定義に関連付けられます。
- ターゲットエントリーの属性の 1 つの値。テンプレートエントリーに相対 DN を提供するために使用される属性は、*cosIndirectSpecifier* 属性を使用して CoS 定義エントリーに指定されます。このタイプのテンプレートは、間接 CoS 定義に関連付けられます。
- CoS がテンプレートの 1 つのレベル検索を実行するサブツリーの DN と、ターゲットエントリの属性の1つの値の組み合わせ。このタイプのテンプレートは、Classic CoS 定義に関連付けられます。

7.2.3. Pointer CoS の仕組み

管理者は、**dc=example,dc=com** に保存されているすべてのエントリーと共通の郵便番号を共有するポインター CoS を作成します。図7.1「Pointer CoS のサンプル」に示されるように、この CoS の 3 つのエントリーが表示されます。

図7.1 Pointer CoS のサンプル

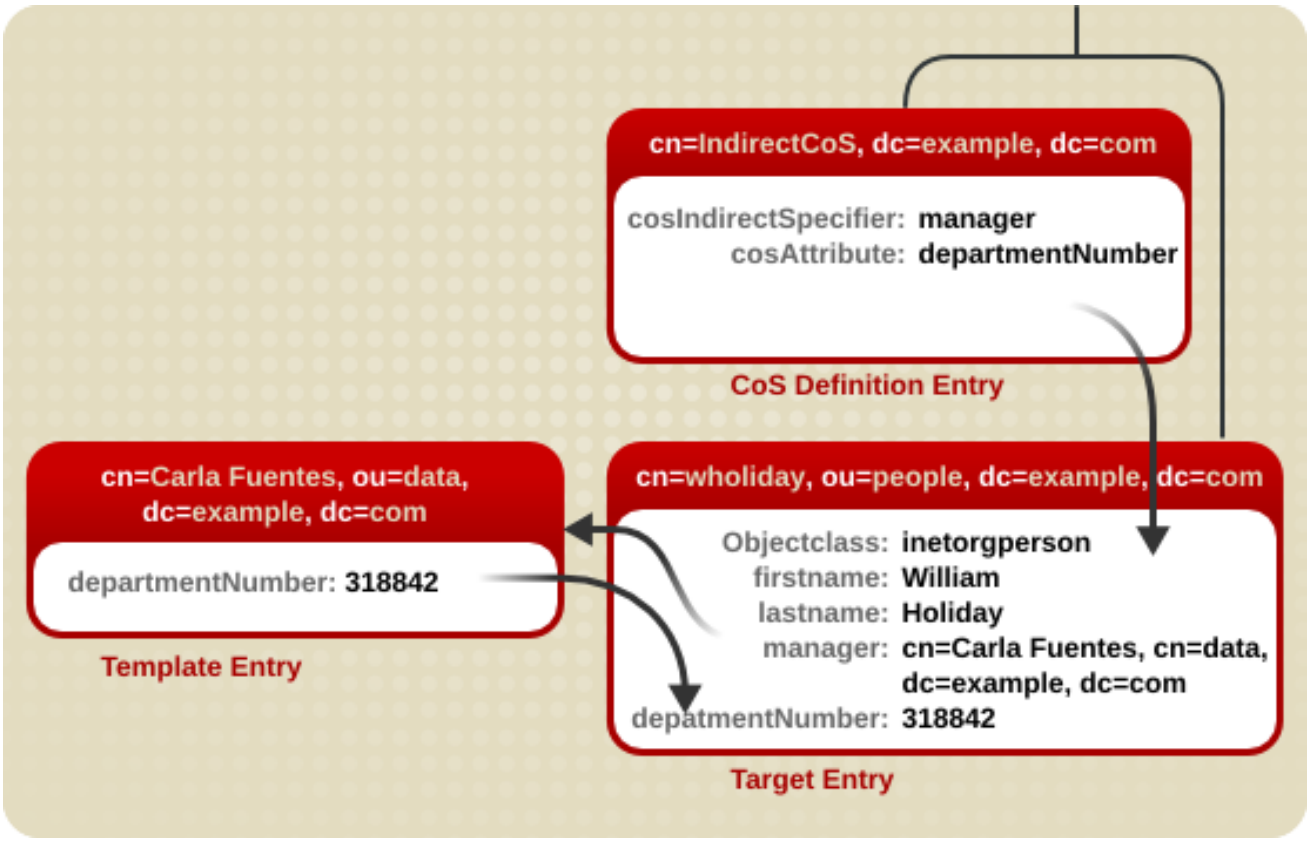


この例では、テンプレートエントリーが CoS 定義エントリーの DN cn=exampleUS,cn=data で識別されます。*postalCode* 属性がエントリー cn=wholiday,ou=people,dc=example,dc=com に対してクエリーされるたびに、Directory Server は、テンプレートエントリー cn=exampleUS,cn=data で使用可能な値を返します。

7.2.4. 間接的な CoS の仕組み

管理者は、ターゲットエントリーの *manager* 属性を使用してテンプレートエントリーを識別する間接的な CoS を作成します。図7.2「間接的な CoS の例」に示されるように、3 つの CoS エントリーが表示されます。

図7.2 間接的な CoS の例

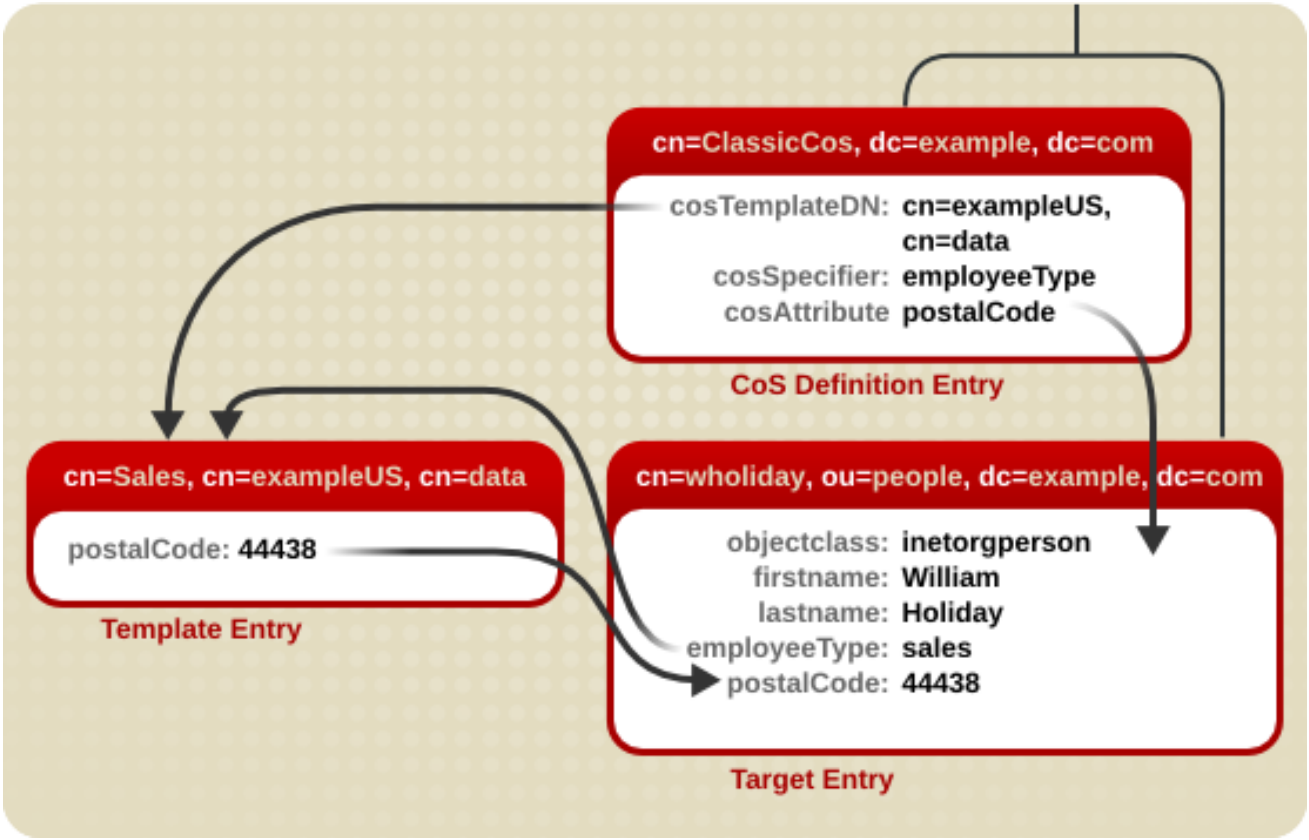


この例では、Wiiam Holiday のターゲットエントリーには間接指定子 (*manager* 属性) が含まれます。William のマネージャーは Carla Fuentes であるため、*manager* 属性にはテンプレートエントリーの DN へのポインター cn=Carla Fuentes,ou=people,dc=example,dc=com が含まれます。テンプレートエントリーは次に、318842 の *departmentNumber* 属性値を提供します。

7.2.5. Classic CoS の仕組み

管理者は、テンプレート DN と CoS 指定子の組み合わせを使用して、有番号を含むテンプレートエントリーを特定します。図7.3「Classic CoS のサンプル」に示されるように、3 つの CoS エントリーが表示されます。

図7.3 Classic CoS のサンプル



この例では、CoS 定義エントリーの ***cosSpecifier*** 属性は、***employeeType*** 属性を指定しています。この属性は、テンプレート DN と組み合わせて、テンプレートエントリーを **cn=sales,cn=exampleUS,cn=data** として特定します。その後、テンプレートエントリーは、***postalCode*** 属性値をターゲットエントリーに提供します。

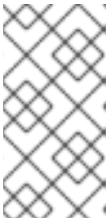
7.2.6. 物理属性値の処理

cosAttribute 属性には、サービスのクラスによって管理される別の属性の名前が含まれます。この属性は、属性値の後に override 修飾子を許可します。属性値の生成時に、CoS がエントリーの既存の属性値を処理する方法を設定します。

> **cosAttribute:** *attribute_name override*

override 修飾子は 4 つあります。

- **default**: エントリーに対応する属性値が格納されてない場合のみ、生成された値を返します。
- **override**: エントリーに値が保存されている場合でも、常に CoS によって生成された値を返します。
- **operational**: 生成された属性が検索で明示的に要求されている場合にのみ、生成された属性を返します。操作属性は、返されるためにスキーマチェックに合格する必要はありません。operational を使用すると、既存の属性値も上書きされます。



注記

属性は、スキーマで操作可能として定義されている場合にのみ操作可能にすることができます。たとえば、CoS が *description* 属性の値を生成する場合、この属性はスキーマで稼働していないため、operational 修飾子を使用することはできません。

- operational-default: エントリーに対応する属性値が格納されておらず、検索で明示的に要求された場合にのみ、生成された値を返します。

修飾子が設定されていない場合は、default と仮定されます。

たとえば、このポインター CoS 定義エントリーは、*postalCode* 属性の値を生成するテンプレートエントリー cn=exampleUS,ou=data,dc=example,dc=com に関連付けられていることを示しています。override 修飾子は、この値が *postalCode* 属性のエントリーによって保存される値よりも優先されることを示しています。

```
dn: cn=pointerCoS,dc=example,dc=com
objectclass: top
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=exampleUS,ou=data,dc=example,dc=com
cosAttribute: postalCode override
```

注記

エントリーに CoS によって生成された属性値が含まれる場合、操作修飾子または上書き修飾子で定義された場合には、属性の値を手動で更新することは できません。

CoS 属性の詳細は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス』 を参照してください。

7.2.7. CoS を使用した多値属性の処理

属性は、サービスのクラスを使用して生成できます。これには複数値の属性が含まれます。これにより、混乱を生じさせる可能性があります。どの CoS が値を提供しますか。これらのいずれか、またはすべてですか。競合 CoS テンプレートからどのように値が選択されていますか。生成された属性は単値または多値を使用しますか。

これを解決する方法は 2 つあります。

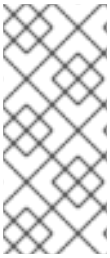
- 複数の CoS が生成する属性をターゲットエントリーにマージするルールを作成。これにより、ターゲットエントリーの値が複数表示されます。
- 競合する CoS 定義の中から 1 つの CoS 値を選択するように優先度を設定。これにより、ターゲットエントリーに 1 つの値が生成されます。

注記

Indirect CoS は *cosPriority* 属性をサポートしません。

CoS が CoS 属性の複数の値を処理する方法は、merge-schemes 修飾子を使用するかどうかで定義されます。

cosAttribute: *attribute override merge-schemes*



注記

merge-schemes 修飾子は、CoS による物理属性値や override 修飾子の処理方法に影響を与えません。競合する CoS テンプレートまたは定義が複数ある場合は、競合するすべての CoS 定義のすべての ***cosAttribute*** に、同じ merge-schemes と override 修飾子を設定する必要があります。それ以外の場合は、1 つの組み合わせが可能なすべての CoS 定義から任意に選択されます。

merge-schemes 修飾子を使用すると、CoS に、管理対象属性に対して複数の値を生成する、または生成できることを通知します。多値 CoS 属性を持つシナリオは 2 つあります。

- 1 つの CoS テンプレートエントリーに管理 CoS 属性のインスタンスが複数含まれるため、ターゲットエントリーに多値が作成されます。以下に例を示します。

```
dn: cn=server access template,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: cosTemplate
accessTo: mail.example.com
accessTo: irc.example.com
```

注記

このメソッドは、Classic CoS でのみ動作します。

- 複数の CoS 定義が同じターゲット属性にサービスクラスを定義する可能性があるため、複数のテンプレートエントリーがあります。以下に例を示します。

```
dn: cn=mail template,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: cosTemplate
accessTo: mail.example.com

dn: cn=chat template,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: cosTemplate
accessTo: irc.example.com
```

ただし、CoS 定義が複数ある場合でも、属性には 1 つの値のみが生成されることがあります。CoS 定義が複数ある場合、値は任意に選択されます。これは予測不可で、意図しないオプションです。どの CoS テンプレートを使用するかを制御する方法は、テンプレートに順位 (優先順位) を設定することであり、優先順位の高い CoS が常に「勝ち」、値を提供します。

値を提供するために複数のテンプレートが完成することはかなり一般的です。たとえば、CoS 定義エントリーには多値の ***cosSpecifier*** 属性を使用できます。テンプレートの優先度は、***cosPriority*** 属性を使用して設定します。この属性は、特定のテンプレートのグローバル優先度を表します。0 の優先度が最も優先されます。

たとえば、部門番号を生成する CoS テンプレートエントリーは、以下のようになります。

```
dn: cn=data,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: cosTemplate
departmentNumber: 71776
cosPriority: 0
```

このテンプレートエントリーには、*departmentNumber* 属性の値が含まれます。優先度はゼロであるため、このテンプレートは、別の *departmentNumber* 値を定義する他の競合するテンプレートよりも優先されます。

cosPriority 属性が含まれないテンプレートは、優先度が最も低いとみなされます。2 つ以上のテンプレートが属性値を提供し、優先度が同じ (または優先順位がない) 場合は、値が任意に選択されます。

注記

負の *cosPriority* 値の動作は Directory Server では定義されないため、負の値を入力しないでください。

7.2.8. CoS 指定の属性の検索

CoS 定義はエントリーで属性の値を指定します。たとえば、CoS はサブツリー内のすべてのエントリーに *postalCode* 属性を設定できます。ただし、これらの CoS 定義属性に対する検索は、通常のエントリーに対する検索のように動作しません。

CoS が定義する属性が、あらゆる種類のインデックス (存在を含む) でインデックス化されている場合、CoS によって設定される値を持つ属性は検索で返されません。以下に例を示します。

- Ted Morris の *postalCode* 属性は、CoS によって定義されます。
- Barbara Jensen の *postalCode* 属性は、Barbara Jensen 自身のエントリーに設定されます。
- *postalCode* 属性はインデックス化されます。

ldapsearch コマンドでフィルター (postalCode=*) が使用される場合、Barbara Jensen のエントリーが返されますが、Tedris のエントリーは返されません。

CoS が定義されている属性がインデックス化されて いない 場合には、属性の値がローカルで設定されるか、CoS と設定されている場合にかかわらず、一致するすべてのエントリーが検索で返されます。以下に例を示します。

- Ted Morris の *postalCode* 属性は、CoS によって定義されます。
- Barbara Jensen の *postalCode* 属性は、Barbara Jensen 自身のエントリーに設定されます。
- *postalCode* 属性はインデックス化 されません。

ldapsearch コマンドでフィルター (postalCode=*) を使用する場合は、Barbara Jensen のエントリーと Ted Morris のエントリーの両方が返されます。

CoS は override を許可します。これは、CoS エントリーの *cosAttribute* 属性に指定された ID です。つまり、属性のローカル値は CoS 値を上書きできます。CoS に上書きが設定されていると、エントリーのローカル値がある限り、ldapsearch 操作は属性がインデックス化された場合でもエントリーの値を返します。CoS を持ち、ローカル値を持たない他のエントリーは、ldapsearch 操作では返されません。

CoS で定義した属性で LDAP 検索要求を実行する際に問題が発生する可能性があるため、CoS を使用して生成する属性を決定する際には注意してください。

7.2.9. アクセス制御と CoS

サーバーは、通常の保存属性と同じように、CoS が生成した属性へのアクセスを制御します。ただし、CoS によって生成された属性の値によってはアクセス制御ルールは機能しません。これは、検索フィルターで CoS が生成する属性の使用に適用される制限と同じです。

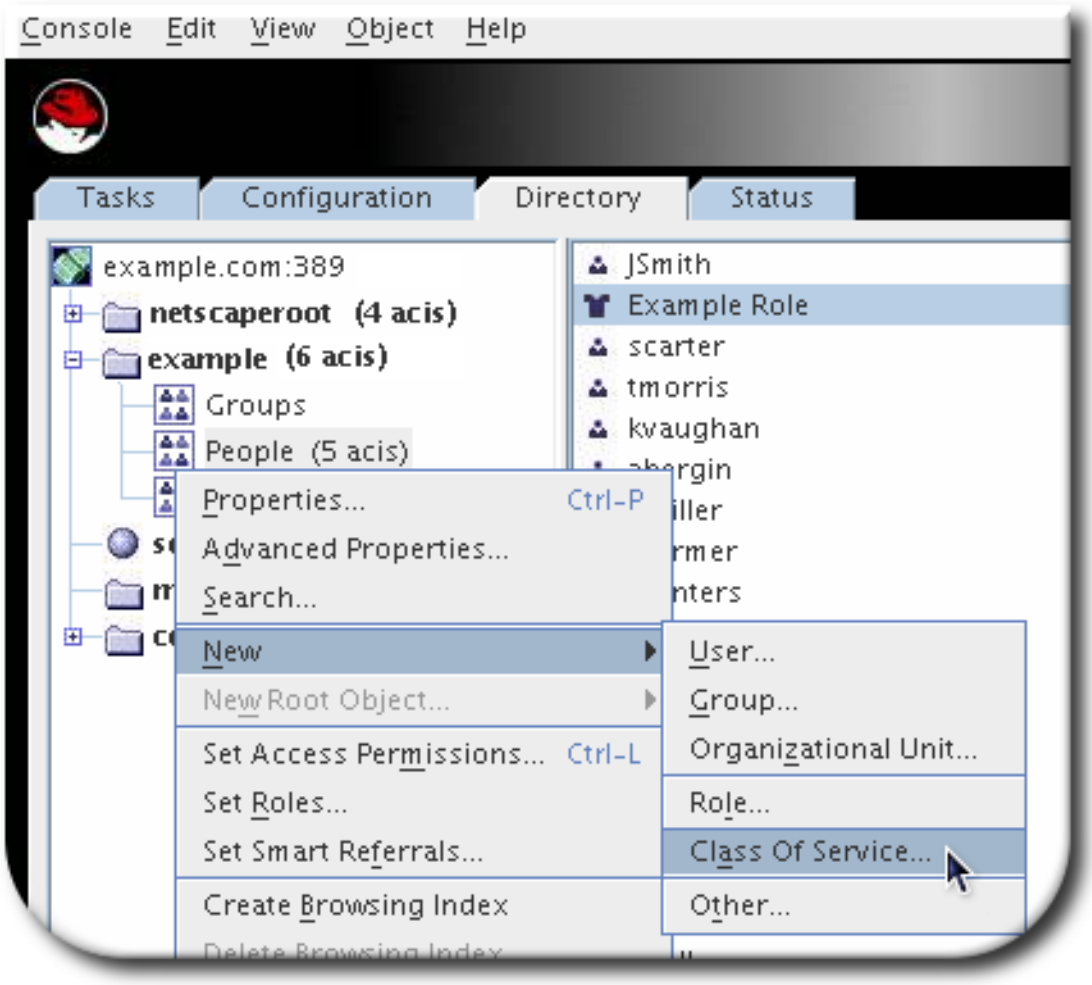
7.2.10. コンソールを使用した CoS の管理

本セクションでは、Directory Server コンソールを使用して CoS を作成し、編集する方法を説明します。

- 「新規 CoS の作成」
- 「CoS テンプレートエントリーの作成」

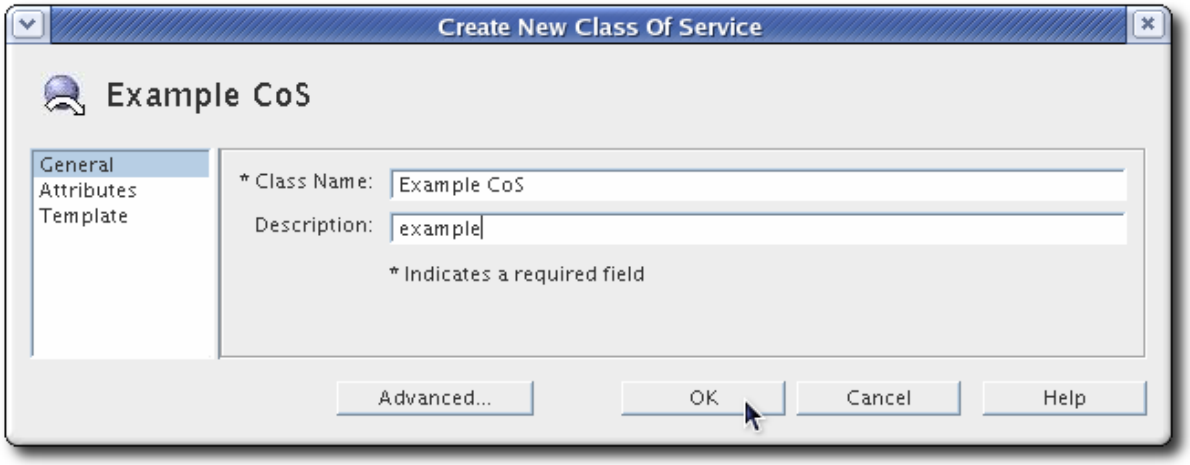
7.2.10.1. 新規 CoS の作成

1. Directory Server コンソールで、Directory タブを選択します。
2. 左側のナビゲーションペインでツリーを参照し、新しいクラスの親エントリーを選択します。
3. **Object** メニューに移動し、**New > Class of Service** を選択します。



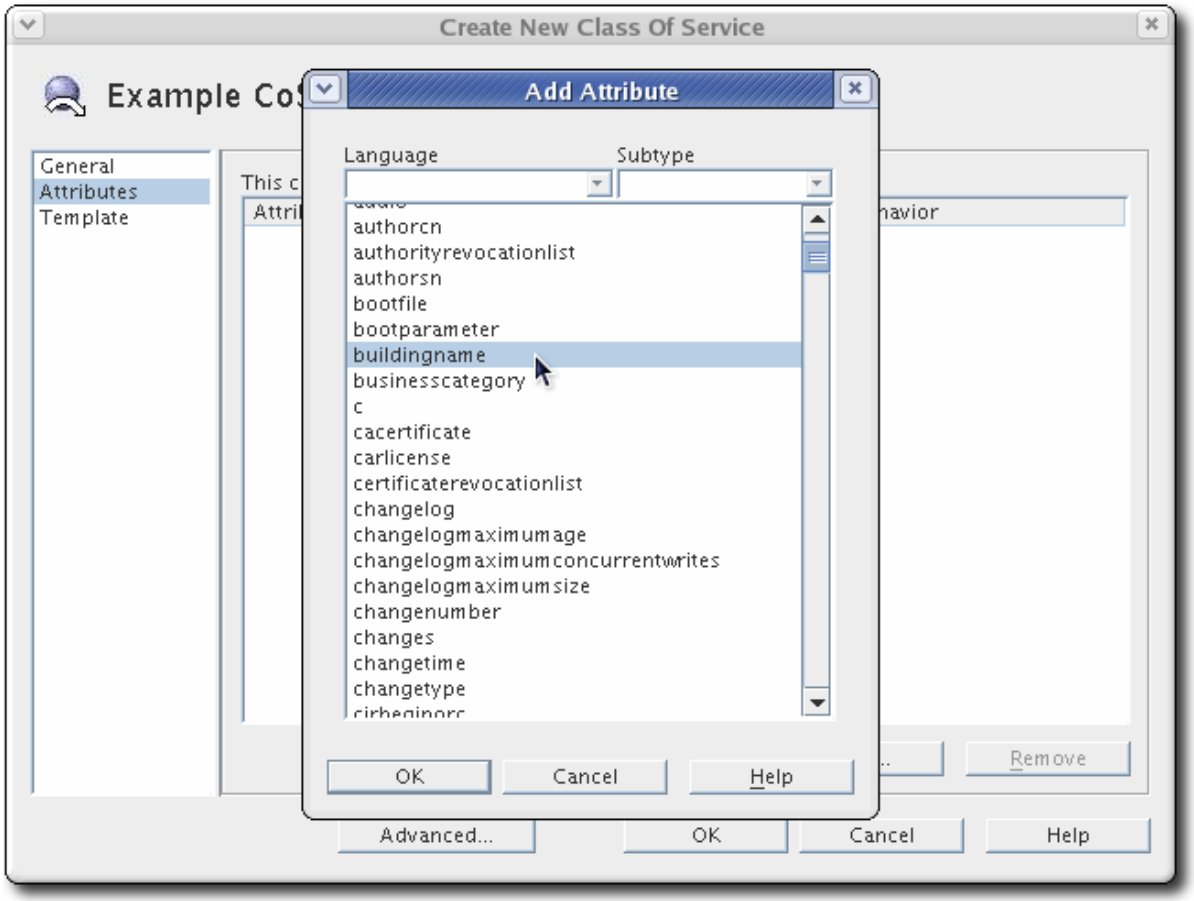
または、エントリーを右クリックし、**New > Class of Service** を選択します。

4. 左側のペインで **General** を選択します。右側のペインで、**Class Name** フィールドに新しいクラスのサービスの名前を入力します。**Description** フィールドにクラスの説明を入力します。

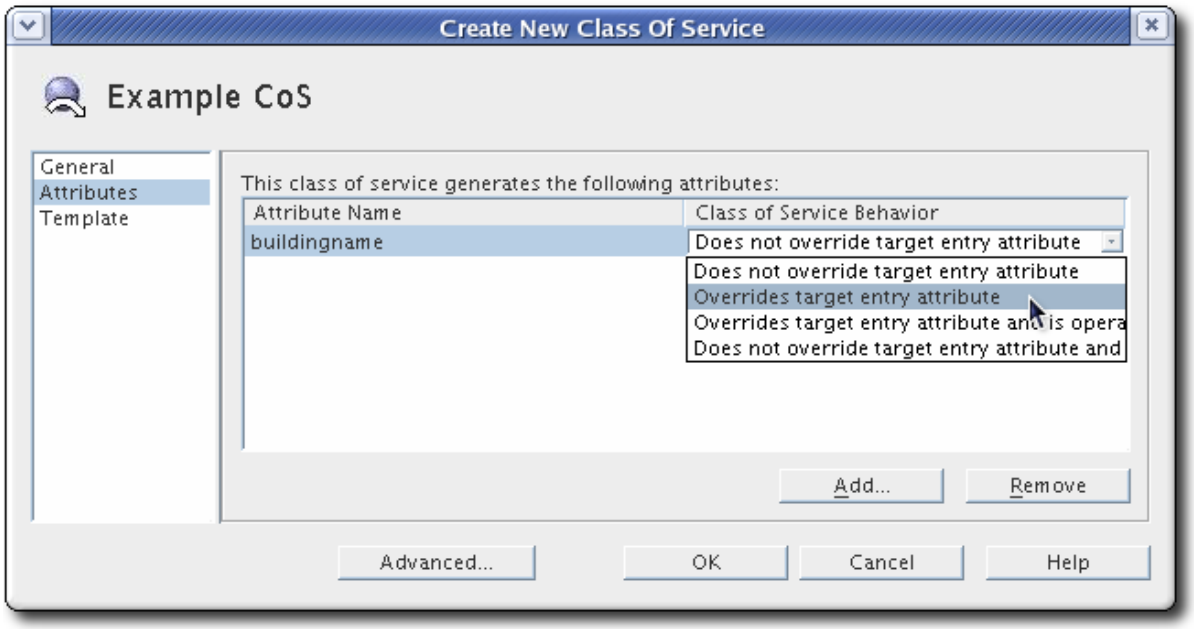


5. 左側のペインで **属性** をクリックします。右側のペインには、ターゲットエントリーで生成された属性の一覧が表示されます。

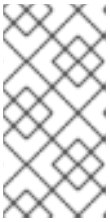
   Add をクリックして、可能な属性の一覧を参照し、一覧に追加します。



6. リストに属性が追加されると、**サービス の動作** 列にドロップダウンリストが表示されます。



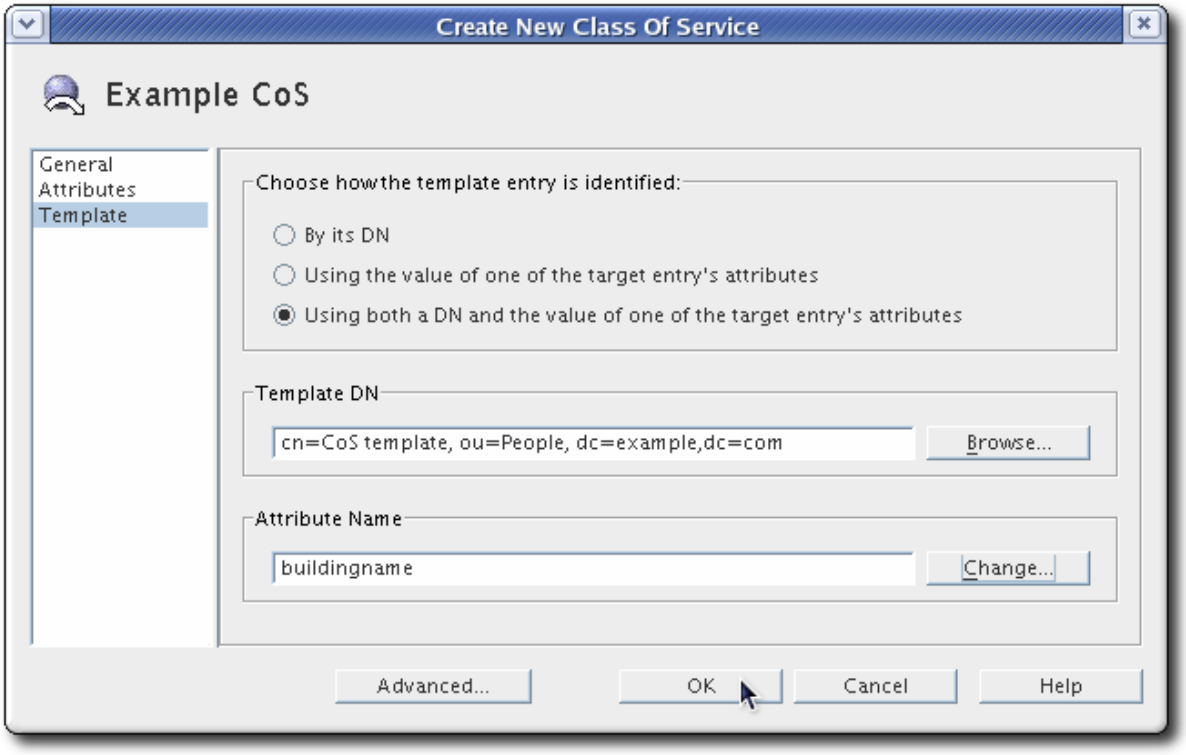
- **Does not override target entry attribute を選択して、エントリーとともに対応する属性値が保存されていない場合に限り、ディレクトリーに対して生成された値を返すよう指示します。**
- **Overrides target entry attribute を選択して、CoS が生成した属性の値がローカル値をオーバーライドします。**
- **Overrides target entry attribute を選択し、 属性がローカル値を上書きして、属性が機能しないようにするため、明示的に要求されない限り、クライアントアプリケーションに表示されないようにすることができます。**
- **Does not override target entry attribute を選択し、エントリーに対応する属性値が格納されておらず、属性が機能しなくなった（明示的に要求されない限りクライアントアプリケーションに表示されない）場合にのみ、生成された値を返すようにディレクトリーに指示する 機能です。**



注記

属性は、スキーマで操作可能としても定義されている場合にのみ操作できます。たとえば、CoS が *description* 属性の値を生成する場合、Overrides ターゲットエントリー属性を選択することはできません。この属性はスキーマで稼働していないためです。

7. **左側のペインで Template をクリックします。右側のペインで、テンプレートエントリーの特定方法を選択します。**



- DN です。DN（ポインター CoS）のみで識別されるテンプレートエントリーを指定するには、**Template DN** フィールドにテンプレートの **DN** を入力します。**Browse** をクリックして、ローカルサーバーで DN を見つけます。これは、**cn=CoS template,ou=People,dc=example,dc=com** などの **DN** の正確な **DN** になります。

- ターゲットエントリーの属性の 1 つの値の使用。ターゲットエントリーの属性の 1 つ（間接的な CoS）の値で識別されるテンプレートエントリーが識別されるようにするには、**Attribute Name** フィールドに属性名を入力します。**Change** をクリックして、利用可能な属性の一覧から別の属性を選択します。

- その DN とターゲットエントリーの属性の 1 つの値の両方を使用します。DN とターゲットエントリーの属性のいずれかの値(classic CoS)の両方によって識別されるテンプレートエントリーを設定するには、テンプレート DN と属性名の両方を入力します。Classic CoS のテンプレート DN はポインター CoS に対してより一般的です。テンプレートエントリーが存在するサフィックスまたは従属接尾辞を参照します。Classic CoS には複数のテンプレートが存在する場合があります。

8. **OK** をクリックします。

7.2.10.2. CoS テンプレートエントリーの作成

ポインター CoS または従来の CoS の場合は、サービスのクラスの作成時に設定されたテンプレート DN に従い、テンプレートエントリーが必要です。***cosTemplateDn*** 属性がその DN を反映しますが、テンプレートエントリーを CoS 自体に配置するのが最善の方法です。

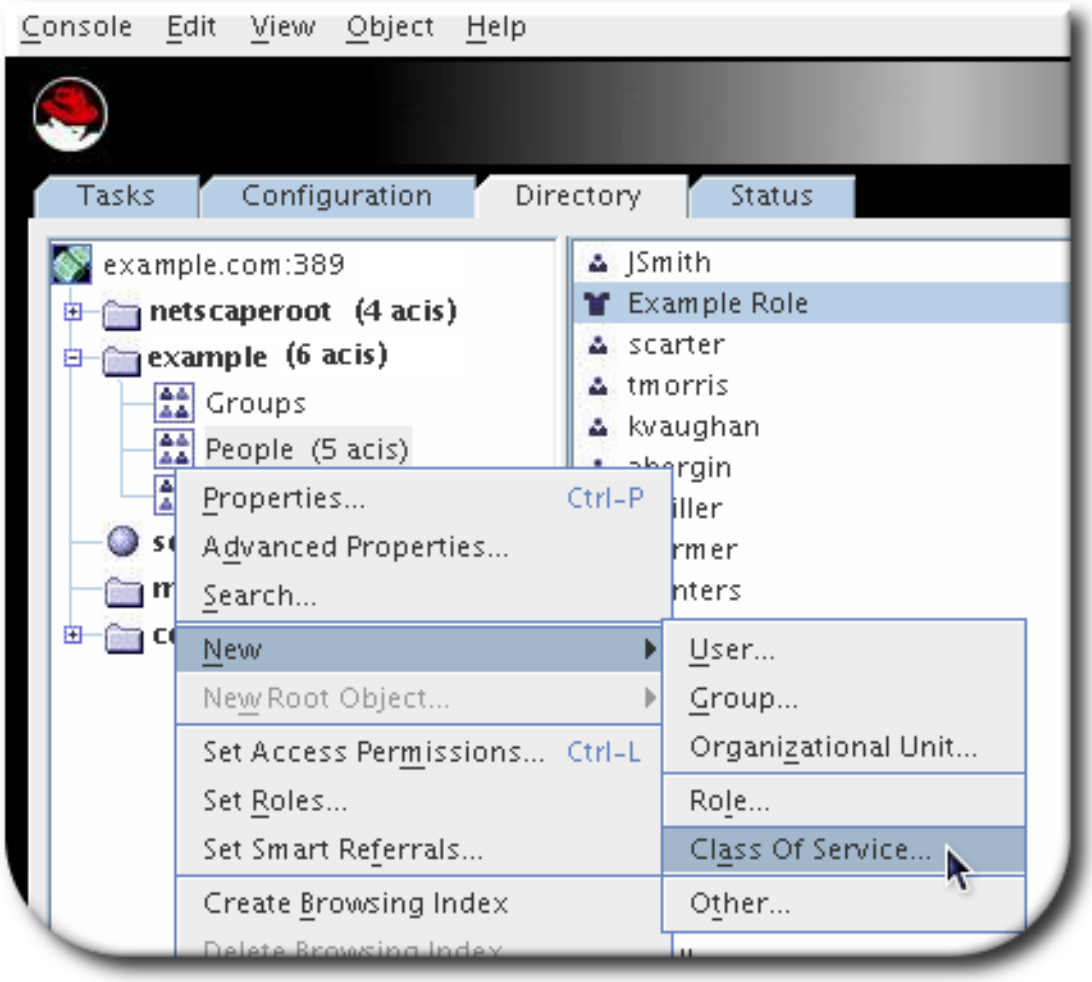
- ポインター CoS の場合は、このエントリーが CoS の作成時に与えられた正確な DN を反映していることを確認します。

- Classic CoS の場合、テンプレート DN は、テンプレートのベース接尾辞として CoS エントリー自体を参照する再帰的である必要があります。

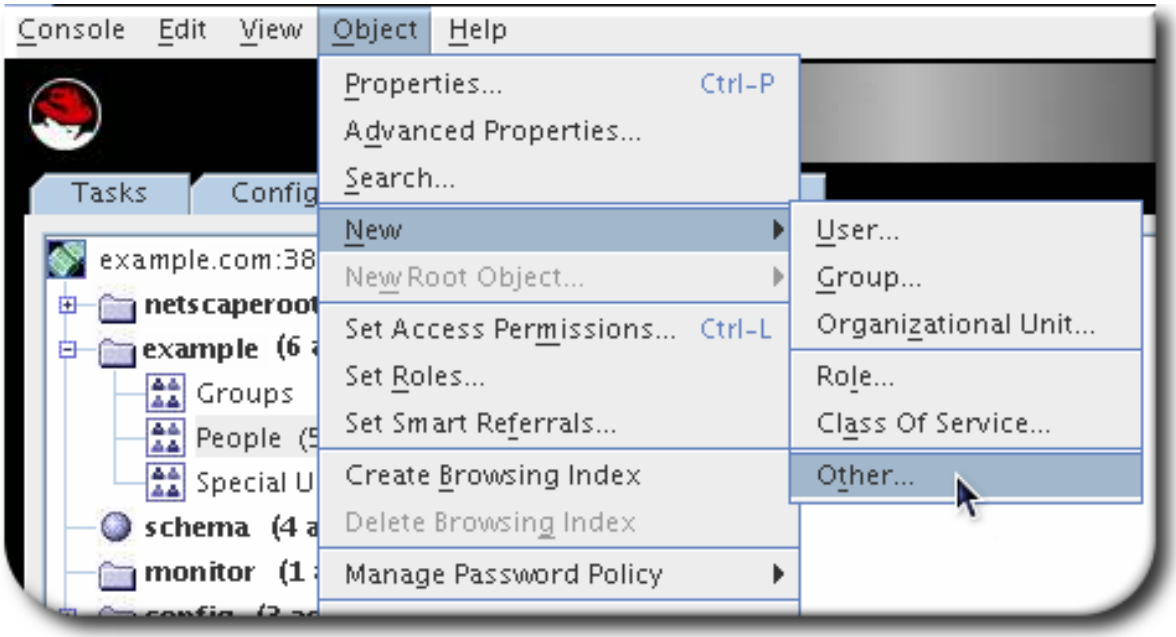
1. Directory Server コンソールで、Directory タブを選択します。

2. 左側のナビゲーションペインでツリーを参照し、サービスのクラスが含まれる親エントリーを選択します。

   CoS が他のエントリーと共に右側のペインに表示されます。

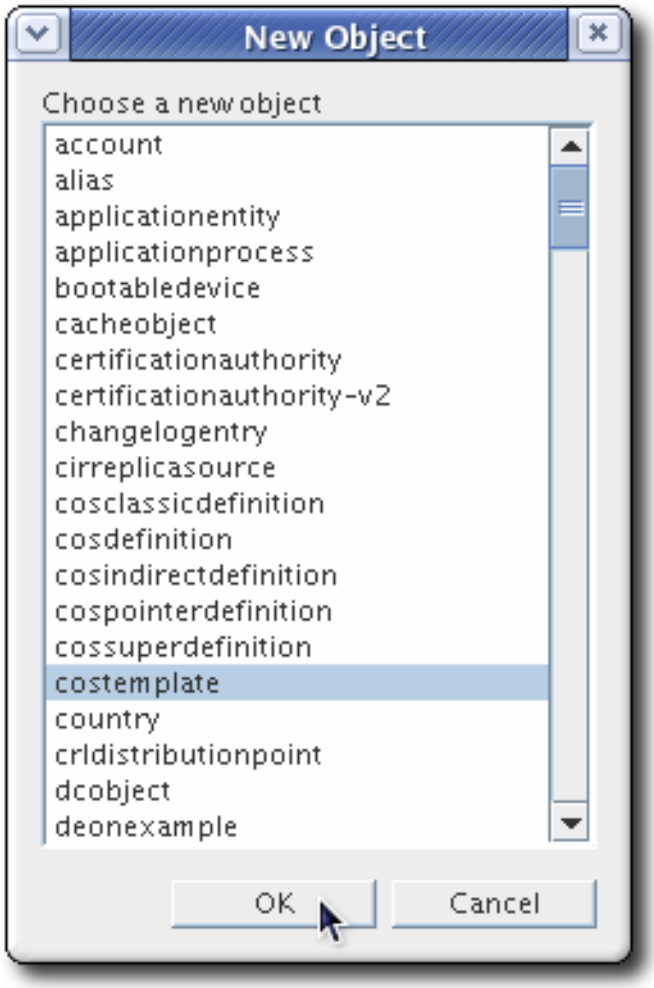


3. CoS を右クリックし、New > Other の順に選択します。



または、右側のペインで CoS を選択し、上部のメニューの Object をクリックし、New > Other を選択します。

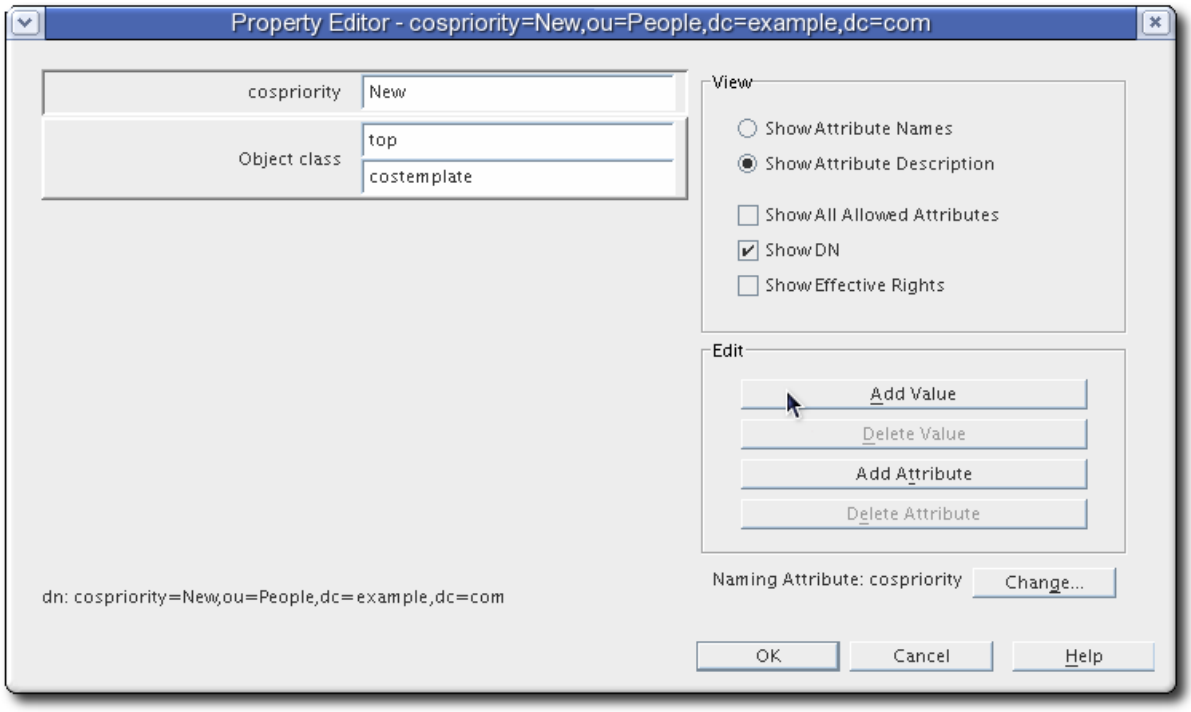
4. オブジェクトクラスの一覧から cosTemplate を選択します。



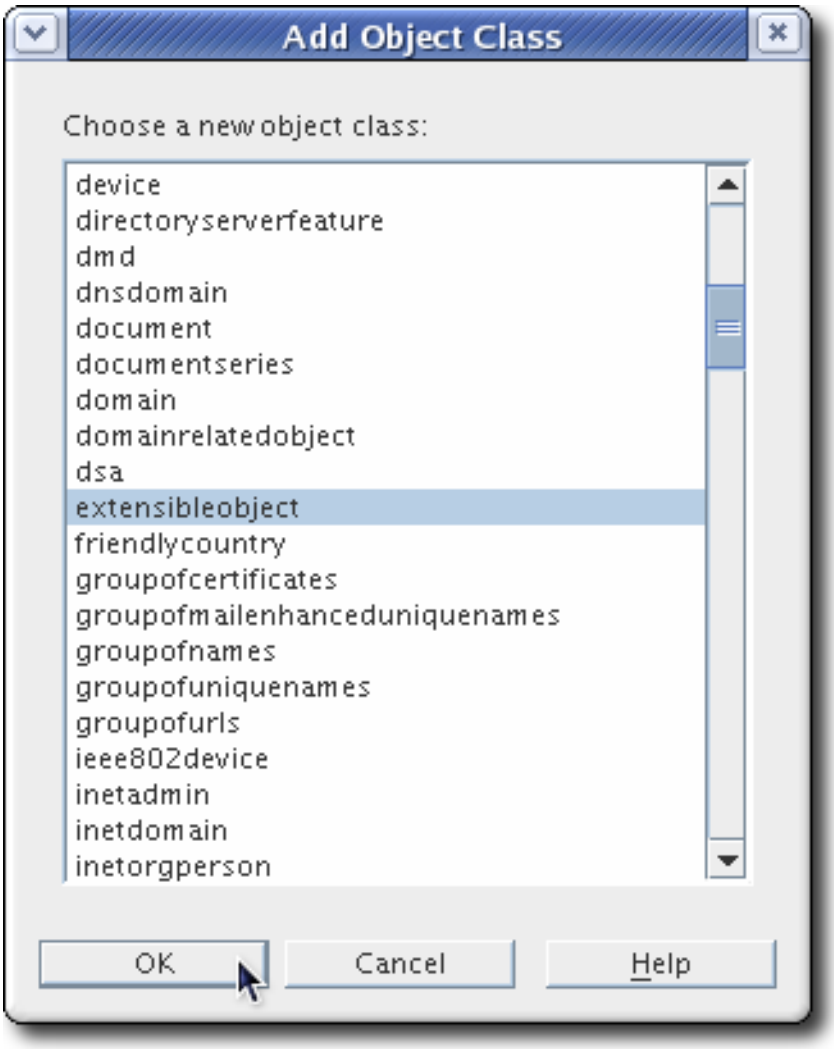
注記

LDAPsubentry オブジェクトクラスは新規テンプレートエントリーに追加できます。CoS テンプレートエントリーを LDAPsubentry オブジェクトクラスのインスタンスに設定すると、通常の検索を設定エントリーによって妨げられません。ただし、テンプレートエントリーがすでに存在し、その他（ユーザーエントリーである場合など）に使用される場合は、LDAPsubentry オブジェクトクラスをテンプレートエントリーに追加する必要はありません。

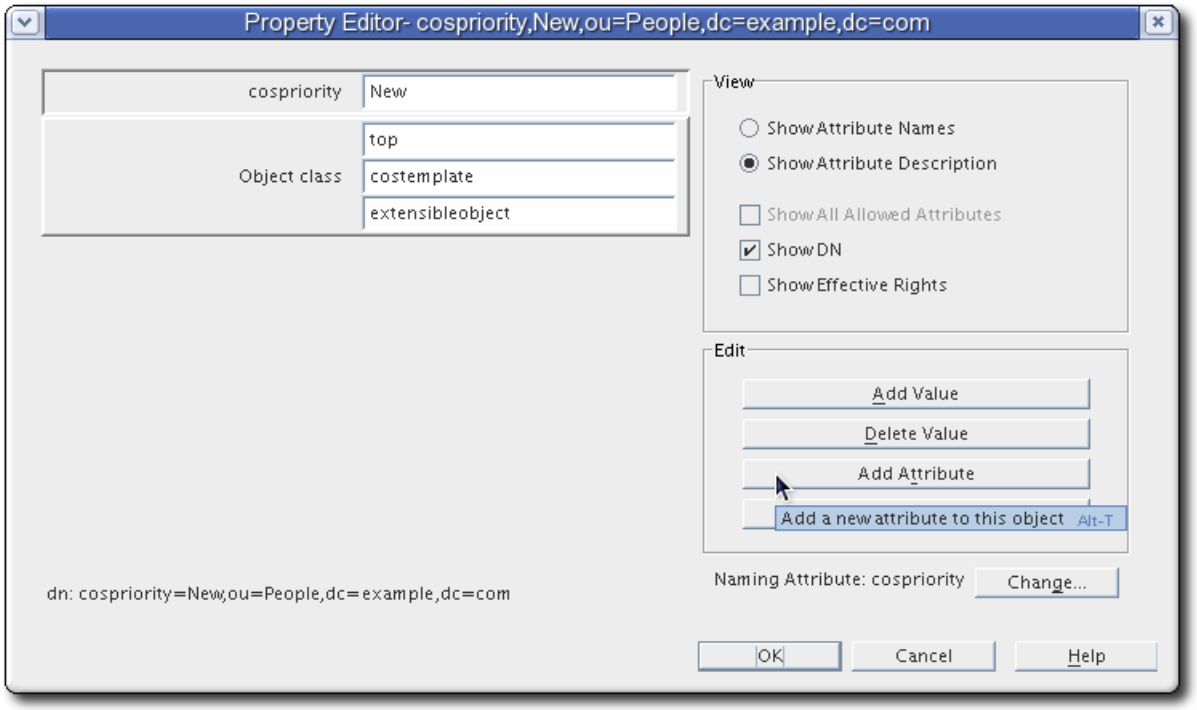
5. オブジェクトクラス属性を選択し、Add Value をクリックします。



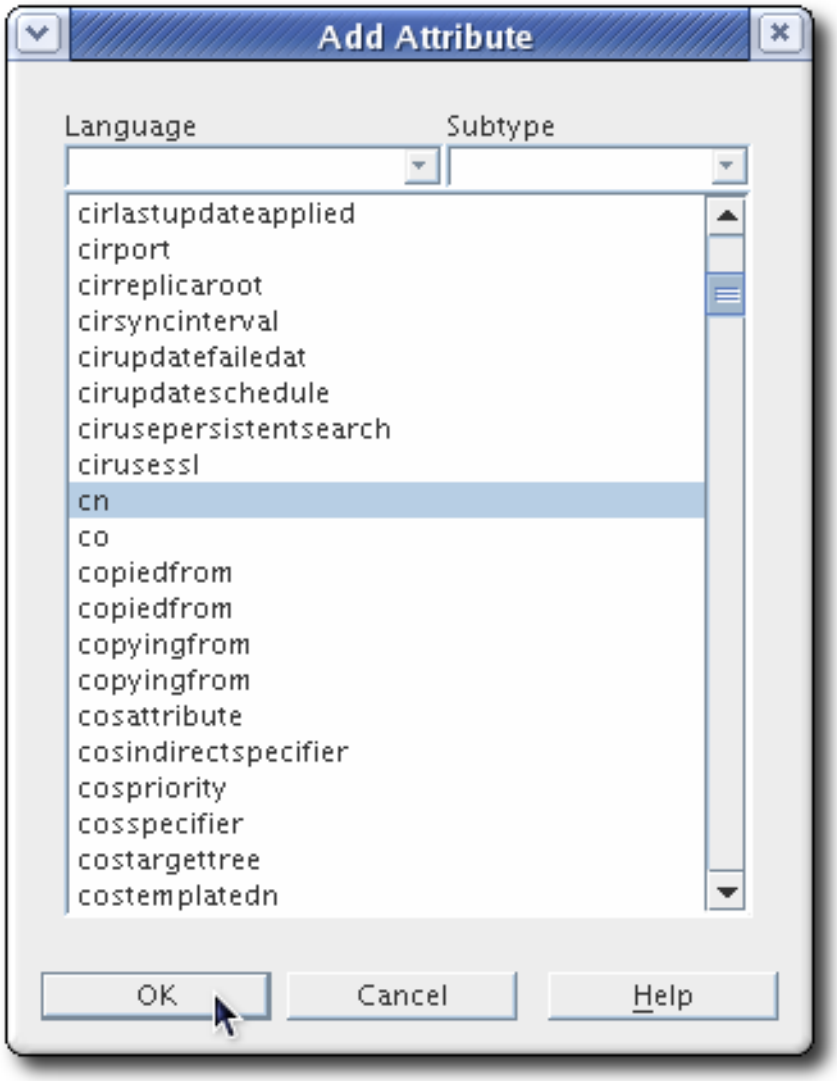
6. **extensibleObject オブジェクトクラス を追加します。これにより、ディレクトリーで利用可能な属性を追加できます。**



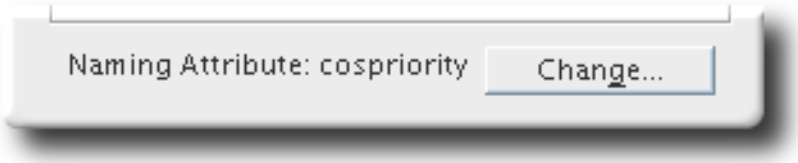
7. **Add Attribute ボタンをクリックします。**



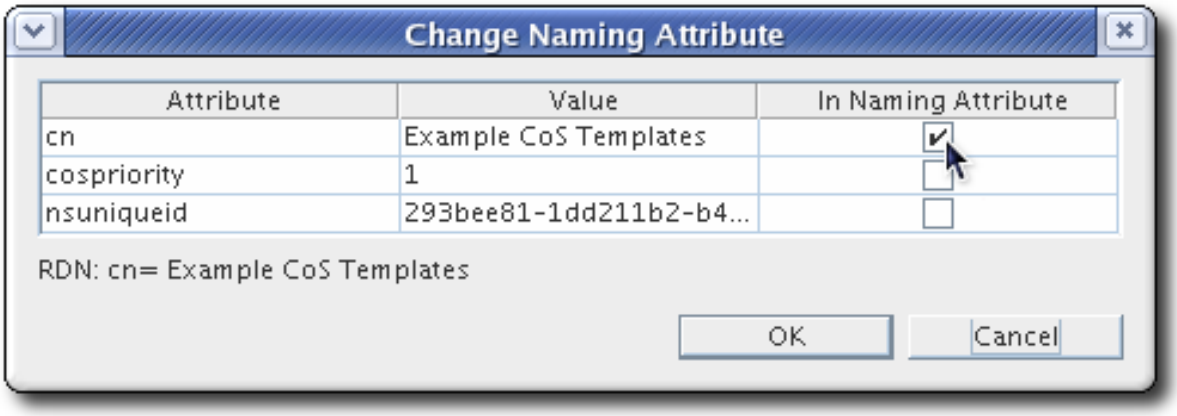
8. ***cn*** **属性を追加し、ターゲットエントリーの属性値に対応する値を指定します。たとえば、従来の CoS の値を設定するために** ***manager*** **属性を使用する場合は、** ***cn*** **に uid=bparker,ou=people,dc=example,dc=com などのマネージャーの DN の値を指定します。または、cn=QA Role,dc=example,dc=com や通常の属性値などのロールに設定します。たとえば、** ***employeeType*** **属性が選択された場合は、完全な時間または 一時的 になります。**



9. 右下の **Change** ボタンをクリックして naming 属性を変更します。

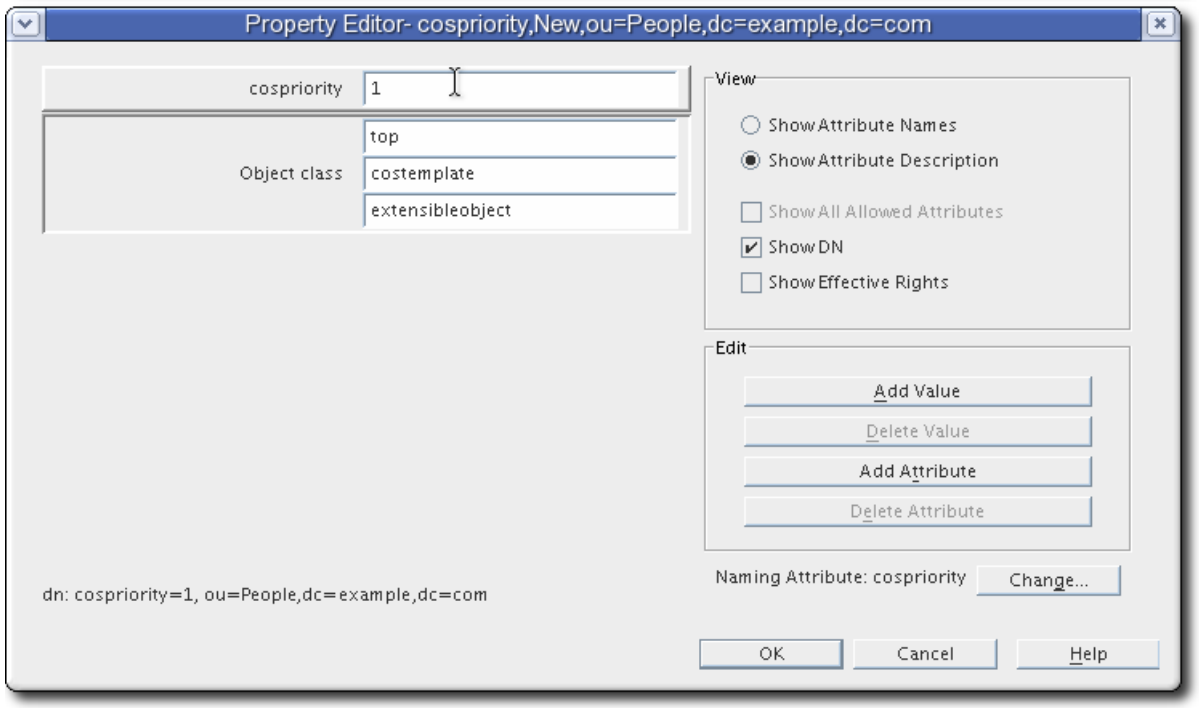


10. エントリーの ***cospriority*** を、***cn*** ではなく naming 属性として使用します。



11. **Add Attribute** ボタンをクリックし、CoS にリストされている属性を追加します。ここで使用する値は、ターゲットエントリーのディレクトリー全体で使用されます。

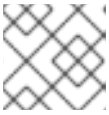
12. ***cospriority*** を設定します。エントリーの特定の属性に適用される CoS が複数ある場合があります。***cospriority*** 属性は、その特定の CoS の重要性をランク付けします。***cospriority*** が高いものは競合に優先します。最も高い優先度は 0 です。



cosPriority 属性が含まれないテンプレートは、優先度が最も低いとみなされます。2 つ以上のテンプレートが属性値を指定し、優先度が同じ（または優先順位がない）場合、値は任意に選択されます。

注記

負の ***cosPriority*** 値の動作は Directory Server では定義されないため、負の値を入力しないでください。



注記

cosPriority 属性は、間接的な CoS ではサポートされていません。

CoS は、その下にエントリーがある場合に、左側のナビゲーションペインに表示されます。Classic CoS の場合、属性指定子の潜在的な値に応じて、複数のエントリーが存在する可能性があります。

既存の CoS のターゲットエントリーで生成された説明または属性を編集するには、**Directory** タブに一覧表示されている CoS エントリーをダブルクリックして、エディターウィンドウで適切な変更を加えます。

7.2.11. コマンドラインでの CoS の管理

すべての設定情報とテンプレートデータはディレクトリーにエントリーとして格納されるため、標準の LDAP ツールを使用して CoS 設定および管理に使用できます。

- 「コマンドラインでの CoS 定義エントリーの作成」

- 「コマンドラインでの CoS テンプレートエントリーの作成」
- 「Pointer CoS の例」
- 「間接的な CoS の例」
- 「Classic CoS の例」
- 「CoS エントリーの検索」

7.2.11.1. コマンドラインでの CoS 定義エントリーの作成

各タイプの CoS では、定義エントリーに特定のオブジェクトクラスを指定する必要があります。すべての CoS 定義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。

ポインター CoS は **cosPointerDefinition** オブジェクトクラスを使用します。例7.3「Pointer CoS エントリーの例」に示されるように、このオブジェクトクラスは、***cosTemplateDn*** 属性で指定されたエントリー DN 値を使用してテンプレートエントリーを識別します。

例7.3 Pointer CoS エントリーの例

```
dn: cn=pointerCoS,dc=example,dc=com
objectclass: top
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn:DN_string
cosAttribute:list_of_attributes qualifier
cn: pointerCoS
```

間接 CoS は **cosIndirectDefinition** オブジェクトクラスを使用します。このタイプの CoS は、***cosIndirectSpecifier*** 属性で指定されているターゲットエントリーの属性のいずれかの値に基づいてテンプレートエントリーを識別します。これは 例7.4「間接的な CoS エントリーの例」で説明されています。

例7.4 間接的な CoS エントリーの例

```
dn: cn=indirectCoS,dc=example,dc=com
objectclass: top
objectclass: cosSuperDefinition
objectclass: cosIndirectDefinition
cosIndirectSpecifier:attribute_name
cosAttribute:list_of_attributes qualifier
cn: indirectCoS
```

クラス CoS は **cosClassicDefinition** オブジェクトクラスを使用します。これは、テンプレートエントリーの DN (***cosTemplateDn*** 属性に設定) と、ターゲットエントリーの属性のいずれかの値 (***cosSpecifier*** 属性に設定) の両方を使用してテンプレートエントリーを特定します。これは 例7.5「Classic CoS エントリーの例」で説明されています。

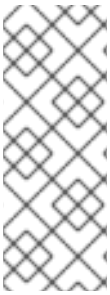
例7.5 Classic CoS エントリーの例

```
dn: cn=classicCoS,dc=example,dc=com
objectclass: top
objectclass: cosSuperDefinition
objectclass: cosClassicDefinition
cosTemplateDn:DN_string
cosSpecifier:attribute_name
cosAttribute:list_of_attributes qualifier
cn: classicCoS
```

サービスのクラスの場合、オブジェクトクラスは CoS のタイプを定義し、サポート属性は CoS テンプレートを定義することで影響を受けるディレクトリーエントリーを識別します。各 CoS には、定義できる属性が 1 つあります (*cosAttribute*)。CoS の目的は、複数のエントリーに属性値を提供することです。*cosAttribute* 属性は、CoS が生成する属性を定義します。

7.2.11.2. コマンドラインでの CoS テンプレートエントリーの作成

各テンプレートエントリーは、**cosTemplate** オブジェクトクラスのインスタンスです。



注記

新しいテンプレートエントリーに **LDAPsubentry** オブジェクトクラスを追加することを検討してください。CoS テンプレートエントリーを **LDAPsubentry** オブジェクトクラスのインスタンスに設定すると、通常の検索を設定エントリーに妨げられずに実行することができます。ただし、テンプレートエントリーがすでに存在し、ユーザーエントリーなどの他のものに使用される場合は、**LDAPsubentry** オブジェクトクラスをテンプレートエントリーに追加する必要はありません。

CoS テンプレートエントリーには、CoS (CoS 定義エントリーの *cosAttribute* 属性で指定) によって生成された属性とその属性の値も含まれます。

たとえば、*postalCode* 属性の値を提供する CoS テンプレートエントリーは、以下のようになります。

```
dn:cn=exampleUS,ou=data,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: cosTemplate
postalCode: 44438
```

以下のセクションでは、テンプレートエントリーの例と、各タイプの CoS 定義エントリーの例を示します。

- 「Pointer CoS の例」
- 「間接的な CoS の例」
- 「Classic CoS の例」

7.2.11.3. Pointer CoS の例

Corporation の管理者の例では、**dc=example,dc=com** ツリー内のすべてのエントリーと共通の郵便番号コードを共有するポインター CoS を作成します。

1. **ldapmodify を使用して、新しい pointer CoS 定義エントリーを dc=example,dc=com 接尾辞に追加します。**

```
dn: cn=pointerCoS,dc=example,dc=com
changetype: add
objectclass: top
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=exampleUS,ou=data,dc=example,dc=com
cosAttribute: postalCode
```

2. **テンプレートエントリーを作成します。**

```
dn: cn=exampleUS,ou=data,dc=example,dc=com
changetype: add
objectclass: top
objectclass: extensibleObject
objectclass: cosTemplate
postalCode: 44438
```

CoS テンプレートエントリー (cn=exampleUS,ou=data,dc=example,dc=com) は、 *postalCode* 属性に保存された値を dc=example,dc=com サフィックスに置かれているエントリーに提供します。これらのエントリーはターゲットエントリーです。

7.2.11.4. 間接的な CoS の例

この間接 CoS は、ターゲットエントリーの *manager* 属性を使用して CoS テンプレートエントリーを識別します。これは、属性の値によって異なります。

1. **ldapmodify を使用して、dc=example,dc=com サフィックスに新しい間接 CoS 定義エントリーを追加します。**

```
dn: cn=indirectCoS,dc=example,dc=com
changetype: add
objectclass: top
objectclass: cosSuperDefinition
objectclass: cosIndirectDefinition
cosIndirectSpecifier: manager
cosAttribute: departmentNumber
```

ディレクトリーまたはマネージャーエントリーに *departmentNumber* 属性がすでに含まれる場合は、他の属性をマネージャーエントリーに追加する必要はありません。この属性は定義エントリーの *cosIndirectSpecifier* 属性で指定されるので、定義エントリーは *manager* 属性が含まれるエントリーのターゲットサフィックス (dc=example,dc=com のエントリー) を検索します。次に、一覧表示された manager エントリーの *departmentNumber* 値を確認します。*departmentNumber* 属性の値は、*manager* 属性を持つマネージャーの下位すべてに自動的にリレーされます。*departmentNumber* の値は、異なるマネージャーのエントリーに記載されている部門番号によって異なります。

7.2.11.5. Classic CoS の例

Corporation の例の管理者は、テンプレート DN と *cosSpecifier* 属性で指定された属性の組み合わせを使用して、郵便番号コードを自動的に生成するClassic CoS を作成します。

1. ldapmodify を使用して、新しいクラス CoS 定義エントリーを dc=example,dc=com サフィックスに追加します。

```
dn: cn=classicCoS,dc=example,dc=com
changetype: add
objectclass: top
objectclass: cosSuperDefinition
objectclass: cosClassicDefinition
cosTemplateDn: cn=classicCoS,dc=example,dc=com
cosSpecifier: businessCategory
cosAttribute: postalCode override
```

2. 営業部門およびマーケティング部門のテンプレートエントリーを作成します。CoS 属性をテンプレートエントリーに追加します。テンプレートの *cn* は、ターゲットエントリーの *businessCategory* 属性の値を設定し、テンプレートの値に応じて属性を追加または上書きされます。

```
dn: cn=sales,cn=classicCoS,dc=example,dc=com
changetype: add
objectclass: top
objectclass: extensibleObject
objectclass: cosTemplate
postalCode: 44438

dn: cn=marketing,cn=classicCoS,dc=example,dc=com
changetype: add
objectclass: top
objectclass: extensibleObject
objectclass: cosTemplate
postalCode: 99111
```

Classic CoS 定義エントリーは、dc=example,dc=com サフィックスの下にあるすべてのエントリーに適用されます。エントリーにある *businessCategory* 属性と *cosTemplateDn* 属性の組み合わせによって、2 つのテンプレートのいずれかに到達することができます。1つ目は営業テンプレートで、営業部門に従業員固有の郵便番号コードを提供します。マーケティングテンプレートは、マーケティング部門の従業員固有の郵便番号コードを提供します。

7.2.11.6. CoS エントリーの検索

CoS 定義エントリーは 運用上の エントリーで、デフォルトでは通常の検索では返されません。つまり、CoS が ou=People,dc=example,dc=com に定義されている場合、以下の ldapsearch コマンドはこれらを返しません。

```
ldapsearch -x -s sub -b ou=People,dc=example,dc=com "(objectclass=*)"
```

CoS 定義エントリーを返すには、ldapSubEntry オブジェクトクラスを CoS 定義エントリーに追加します。以下に例を示します。

```
dn: cn=pointerCoS,ou=People,dc=example,dc=com
objectclass: top
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
```

```
objectclass: ldapSubEntry
cosTemplateDn: cn=exampleUS,ou=data,dc=example,dc=com
cosAttribute: postalCode override
```

次に、検索に特別な検索フィルター (objectclass=ldapSubEntry) を使用します。このフィルターは、OR（| を使用して他の検索フィルターに追加できます）：

```
ldapsearch -x -s sub -b ou=People,dc=example,dc=com "(|(objectclass=*)
(objectclass=ldapSubEntry))"
```

この検索は、ou=People,dc=example,dc=com サブツリーの CoS 定義エントリーに加えて、すべての通常のエントリーを返します。

注記

コンソールには、CoS エントリーが自動的に表示されます。

7.2.12. ロールベースの属性の作成

Classic CoS スキームは、エントリーが所有するロールに基づいてエントリーの属性値を生成します。たとえば、ロールベースの属性を使用して、エントリーごとにサーバーのルックスルー制限を設定できます。

ロールベースの属性を作成するには、Classic CoS の CoS 定義エントリーで *nsRole* 属性を *cosSpecifier* として使用します。*nsRole* 属性が多値指定できるため、複数のテンプレートエントリーを持つ CoS スキームを定義できます。使用するテンプレートエントリーの曖昧さを解決するには、CoS テンプレートエントリーに *cosPriority* 属性を追加します。

たとえば、この CoS は manager ロールのメンバーが標準のメールボックスクォータを超過できるようにします。マネージャーのロールエントリーは次のとおりです。

```
dn: cn=ManagerRole,ou=people,dc=example,dc=com
objectclass: top
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: ManagerRole
nsRoleFilter: ou=managers
Description: filtered role for managers
```

重要

nsRoleFilter 属性は仮想属性の値を受け付けません。

Classic CoS 定義エントリーは以下のようになります。

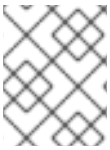
```
dn: cn=managerCOS,dc=example,dc=com
objectclass: top
objectclass: cosSuperDefinition
objectclass: cosClassicDefinition
cosTemplateDn: cn=managerCOS,dc=example,dc=com
cosSpecifier: nsRole
cosAttribute: mailboxquota override
```

cosTemplateDn 属性は、*cosSpecifier* 属性で指定された属性 (ターゲットエントリーの *nsRole* 属性) とともに CoS テンプレートエントリーを識別する値を提供します。CoS テンプレートエントリーは、*mailboxquota* 属性の値を提供します。override の他の修飾子は、ターゲットエントリーの既存の *mailboxquota* 属性値を上書きするよう CoS に指示します。

対応する CoS テンプレートエントリーは以下のようになります。

```
dn:cn="cn=ManagerRole,ou=people,dc=example,dc=com",cn=managerCOS,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: cosTemplate
mailboxquota: 1000000
```

テンプレートは、*mailboxquota* 属性の値 1000000 を指定します。



注記

ロールエントリーと CoS 定義およびテンプレートエントリーは、ディレクトリーツリーの同じレベルにある必要があります。

7.3. 属性値の管理属性のリンク

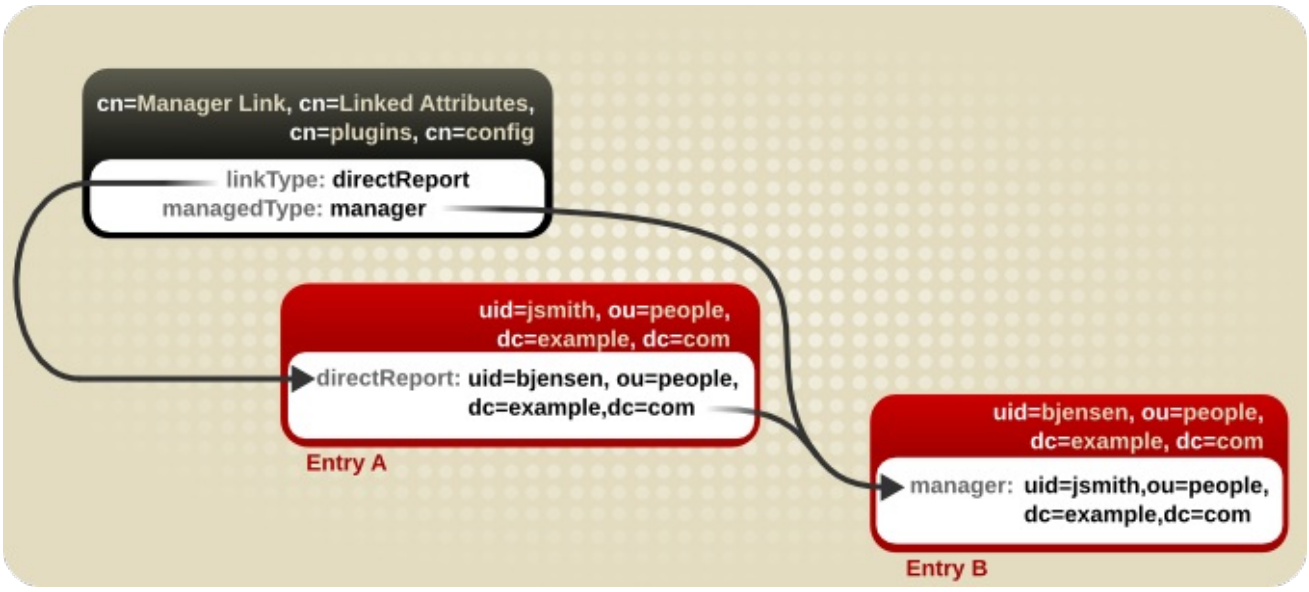
サービスクラスは、住所、郵便番号、主なオフィス番号など、すべてが 同じ値 の属性を持つエントリーに対して、属性値を動的に提供します。これらは共有属性値であり、単一のテンプレートエントリーで更新されます。

多くの場合、エントリー間には、それらの間のリンクを表現する方法が必要な関係がありますが、その関係を表現する値 (および場合によっては属性) は異なります。Red Hat Directory Server は、指定された属性を繋ぎ合わせる方法を提供するため、1 つのエントリーの属性が変更すると、関連するエントリーの対応する属性が自動的に更新されます。(リンクおよび管理属性の両方に DN の値があります。リンク属性の値には、更新するプラグインのエントリーの DN が含まれます。2 つ目のエントリーの管理属性には、元のリンクエントリーを参照する DN の値があります。)

7.3.1. リンク属性の概要

リンク先属性プラグインは、プラグインの複数のインスタンスを許可します。各インスタンスは、管理者によって手動で維持される 1 つの属性 (*linkType*) およびプラグインによって自動的に維持される 1 つの属性 (*managedType*) を設定します。

図7.4 リンク先属性の基本設定

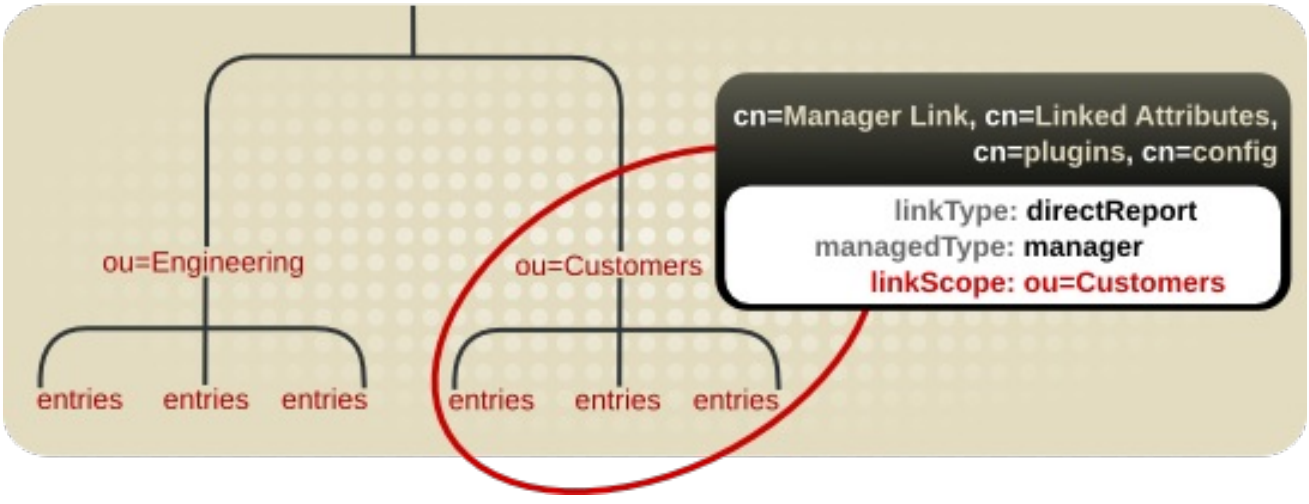


注記

データの一貫性を維持するには、プラグインプロセスのみが管理属性を維持する必要があります。管理属性へのすべての書き込みアクセスを制限する ACI の作成を検討してください。ACI の設定に関する詳細は、「ACI の追加」を参照してください。

リンク先属性プラグインインスタンスは、ディレクトリー内の単一のサブツリーに制限できます。これにより、属性の組み合わせと影響を受けるエントリーのより柔軟なカスタマイズが可能になります。スコープが設定されていない場合、プラグインはディレクトリー全体で動作します。

図7.5 リンク先属性プラグインを特定のサブツリーに制限



リンク先属性プラグインインスタンスを設定する場合は、特定の設定が必要です。

- 管理属性とリンク先属性の両方で、属性定義で識別名の構文が必要です。リンク先属性は基本的にクロス参照で管理されます。プラグインがこれらの相互参照を処理する方法は、属性値からエントリーの DN をプルすることで行われます。

  カスタムスキーマ要素のプランニングに関する情報は、「12章 *ディレクトリースキーマの管理*」を参照してください。

- 各リンク先属性プラグインインスタンスはローカルで、管理 属性は一部レプリケーションを使用してレプリケーションからブロックする必要があります。

  あるサプライヤーに加えられた変更は、プラグインが自動的に発生し、対応するディレクトリーエントリーの値を管理するため、データはサーバー全体で一貫性を維持します。ただし、リンクされたエントリー間でデータを一貫性を保つには、プラグインインスタンスで管理属性を維持する必要があります。つまり、管理属性の値は、マルチマスターレプリケーション環境であってもレプリケーションプロセスではなく、プラグインプロセスによってのみ維持される必要があります。

  一部レプリケーションの使用方法は、「一部レプリケーションを使用した属性のサブセットの複製」を参照してください。

7.3.2. リンク元属性プラグイン構文の確認

デフォルトのリンク先属性プラグインエントリーは、各プラグインインスタンスのコンテナーエントリーです。これは、次のセクションの password 構文プラグインや DNA プラグインに類似します。このコンテナーエントリーの下にある各エントリーは、異なるリンク管理属性ペアを定義します。

新しいリンク元属性ペアを作成するには、コンテナーエントリーの下に新しいプラグインインスタンスを作成します。基本的なリンク元属性プラグインインスタンスでは、以下の 2 つの項目を定義する必要があります。

- *linkType* 属性において、管理者が手動で管理する属性
- *managedType* 属性に含まれる、プラグインによって動的に作成される属性
- 必要に応じて、プラグインを *linkScope* 属性のディレクトリーツリーの特定の部分に制限するスコープ

例7.6 リンク先属性のプラグインインスタンスエントリーの例

```
dn: cn=Manager Link,cn=Linked Attributes,cn=plugins,cn=config
objectClass: top
objectClass: extensibleObject
cn: Manager Link
linkType: directReport
managedType: manager
linkScope: ou=people,dc=example,dc=com
```

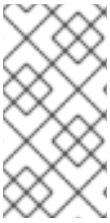
リンク先属性プラグインインスタンスのインスタンスで利用可能な属性はすべて、表7.2「リンクされたプラグインインスタンス属性」 に記載されています。

表7.2 リンクされたプラグインインスタンス属性

プラグイン属性	詳細
cn	プラグインインスタンスの一意の名前を指定します。
linkScope	プラグインインスタンスの機能を制限する接尾辞の DN が含まれます。

プラグイン属性	詳細
linkType	管理者が維持する属性を指定します。この属性は手動で維持され、プラグインの参照として使用されます。この属性には DN 値の形式が必要です。属性を追加、変更、または削除されると、その値には、更新するプラグインのターゲットエントリーの DN が含まれます。
managedType	プラグインによって維持される属性を指定します。この属性は、ターゲットエントリーで作成され、更新されます。この属性には DN 値の形式が必要です。属性がエントリーに追加されると、その値は管理エントリーへの相互参照として表されます。

7.3.3. 属性リンクの設定



注記

リンク先属性プラグインインスタンスは、Directory Server Console に作成できますが、必要な属性をすべて手動で追加することで、ディレクトリーエントリーの Advanced Property Editor のみを使用して、コマンドラインでエントリーを手動で作成します。

1. これが有効になっていない場合は、「Directory Server コンソールでプラグインの有効化」または「プラグインを動的に有効化」の説明に従って、リンク先属性プラグインを有効にします。
2. プラグインインスタンスを作成します。*managedType* および *linkType* 属性の両方が必要です。プラグインの構文については、「リンク元属性プラグイン構文の確認」を参照してください。以下の例は、ldapmodify を使用して作成したプラグインインスタンスを示しています。

```
dn: cn=Manager Link,cn=Linked Attributes,cn=plugins,cn=config
changetype: add
objectClass: top
objectClass: extensibleObject
cn: Manager Link
linkType: directReport
managedType: manager
```

3. *nsslapd-dynamic-plugins* を使用して動的プラグインを有効にするためにサーバーが設定されていない場合は、サーバーを再起動して新しいプラグインインスタンスを適用します。

```
# systemctl restart dirsrv.target
```

7.3.4. 属性リンクのクリーンアップ

管理属性とリンク先属性は同期しなくなる可能性があります。たとえば、リンク先属性をサーバーにインポートまたは複製できましたが、リンク属性が適切に構成されていなかったため、対応する管理対象属性はインポートされませんでした。管理属性とリンク先属性ペアは、スクリプト (fix-linkedattrs.pl) を実行するか、修正タスクを起動することで修正できます。

修正タスクは、参照エントリーに対応するリンク属性 (管理者が管理する属性) のない管理属性 (プラグインによって管理される属性) を削除します。逆に、エントリーにリンク属性が存在する場合は、タスクで不明な管理属性が追加されます。

7.3.4.1. fixup-linkedattrs.pl を使用したリンク先属性の再生成

fixup-linkedattrs.pl スクリプトは特殊なタスクを起動し、ディレクトリーエントリー上の管理属性とリンク属性のペアをすべて再生成します。特定の状況では、1つまたはもう1つが失われる可能性があります。リンク属性がエントリーに存在する場合、タスクは利用可能な属性でクロス参照されている DN を追跡し、参照されたエントリーで対応する管理属性を作成します。対応するリンク属性のない管理属性が存在する場合は、管理属性値が削除されます。

プラグインのスコープに設定されたリンク属性ペアをすべて修復するには、Directory Manager で以下のコマンドを実行します。

```
# fixup-linkedattrs.pl -D "cn=Directory Manager" -w password
```

-I オプションを使用してターゲットプラグインインスタンス DN を指定し、修正タスクを単一のリンク管理属性ペアに制限することもできます。

```
# fixup-linkedattrs.pl -D "cn=Directory Manager" -w password -I "cn=Manager Link,cn=Linked Attributes,cn=plugins,cn=config"
```

例で使用するパラメーターの詳細は、Red『Hat Directory Server の設定、コマンド、およびファイルリファレンス の fixup-linkedattrs.pl スクリプトの説明を参照してください』。

7.3.4.2. ldapmodify を使用したリンク先属性の再生成

リンクされた属性の修復は、特別なタスク設定エントリーで管理できるタスクの1つです。タスクエントリーは、dse.ldif ファイルの cn=tasks 設定エントリーで発生するため、ldapmodify を使用してエントリーを追加してタスクを開始することもできます。タスクが完了すると、エントリーはディレクトリーから削除されます。

このタスクは、実行時に fixup-linkedattrs.pl スクリプトによって自動的に作成されるタスクと同じです。

リンク付きの属性修正タスクを開始するには、cn=fixup linked attributes,cn=tasks,cn=config エントリーの下にエントリーを追加します。必要な属性は特定タスクの *cn* のみですが、*ttl* 属性にはタイムアウト期間を設定できます。ldapmodify の使用:

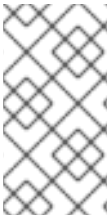
```
dn: cn=example,cn=fixup linked attributes,cn=tasks,cn=config
changetype: add
cn:example
ttl: 5
```

タスクが完了すると、エントリーは dse.ldif 設定から削除されるため、同じタスクエントリーを継続的に再利用できます。

cn=fixup linked attributes タスク設定は『、設定、コマンド、およびファイルリファレンス を参照してください』。

7.4. 一意の数値属性値の割り当ておよび管理

エントリー属性によっては、*uidNumber* や *gidNumber* などの一意の番号が必要です。Directory Server は、DNA (Distributed Numeric Assignment) プラグインを使用して、指定された属性に一意の番号を自動的に生成して提供できます。



注記

属性の一意性 は、DNA プラグインで維持されるとは限りません。プラグインは、重複しない範囲のみを割り当てますが、管理属性に手動で数字を割り当てることができ、手動で割り当てられた番号が一意であることを検証したり要求したりすることはありません。

一意の数字を割り当てる問題は、数字の生成ではなく、レプリケーションの競合を回避することで問題となります。DNA プラグインは、単一 のバックエンド全体に一意の番号を割り当てます。マルチマスターレプリケーションの場合、各マスターがローカル DNA プラグインインスタンスを実行しているときに、各インスタンスが真に一意の番号セットを使用していることを確認する方法が必要です。これは、割り当てる各サーバーに異なる 範囲 を割り当てることで行われます。

7.4.1. 動的番号の割り当ての概要

サーバーの DNA プラグインは、インスタンスが発行することのできる利用可能な番号の範囲を割り当てます。範囲の定義は非常にシンプルで、サーバーの次に利用可能な番号 (範囲の下限) と最大値 (範囲の最後) の 2 つの属性で設定されます。初期の下限範囲は、プラグインインスタンスを設定する際に設定されます。その後、下部の値はプラグインによって更新されます。利用可能な数を各レプリカの複数の範囲に分割することで、サーバーはすべて、互いに重複することなく、継続的に番号を割り当てることができます。

7.4.1.1. フィルター、検索、およびターゲットエントリー

(「標準インデックスの作成」で説明されているように) サーバーは、内部的にソートされた検索を実行し、次に指定された範囲がすでに取得されているかどうかを確認し、管理属性に適切な順序のマッチングルールと同じインデックスを割り当てる必要があります。

DNA プラグインは、常にディレクトリーツリーの特定領域 (スコープ) と、そのサブツリー内の特定のエントリータイプ (フィルター) に適用されます。

重要

DNA プラグインは 1 つのバックエンドでのみ機能します。複数のデータベースの番号割り当ては管理できません。DNA プラグインは、DNA プラグイン以外で値がすでに割り当てられているかどうかを確認する際に、ソートコントロールを使用します。この検証は、ソートコントロールを使用して単一のバックエンドでのみ機能します。

7.4.1.2. 範囲および割り当て番号

Directory Server が属性値の生成を処理する方法は複数あります。

- 最も単純なケースでは、属性がない場合に、unique-number 属性を必要とするオブジェクトクラスを持つディレクトリーにユーザーエントリーが追加されます。管理属性に値を持たないエントリーを追加すると、DNA プラグインによる値の割り当てが発生します。このオプションは、一意の値を 1 つの属性に割り当てるように DNA プラグインが設定されている場合に限り機能します。

- 同様の管理可能なオプションは、マジック番号 を使用することです。このマジックナンバーは、管理対象属性のテンプレート値であり、サーバーの範囲外のもの、数字、または単語でさえあり、プラグインは新しい割り当て値に置き換える必要があると認識します。マジック値でエントリーが追加され、エントリーが設定された DNA プラグインの範囲およびフィルター内にある場合は、プラグインでマジック番号を使用した新しい値の生成が自動的に発生します。下の例では、ldapmodify の使用に基づいて、0 をマジックナンバーとして追加します。

```
dn: uid=jsmith,ou=people,dc=example,dc=com
changetype: add
objectClass: top
objectClass: person
objectClass: posixAccount
uid: jsmith
cn: John Smith
uidNumber: 0
gidNumber: 0
....
```

DNA プラグインは、新規の一意の値のみを生成します。DNA プラグインが制御する属性に特定の値を使用するためにエントリーを追加または変更した場合には、指定した番号が使用されます。DNA プラグインは、その番号を上書きしません。

7.4.1.3. 同じ範囲の複数の属性

DNA プラグインは、1つの属性タイプに、または1つの範囲の一意の番号から複数の属性タイプに一意の番号を割り当てることができます。

これにより、属性に一意の数字を割り当てるためのオプションが複数提供されます。

- 一意の番号の1つの範囲から、1つの属性タイプに割り当てられた1つの番号。
- 1つのエントリーの 2 つの属性に割り当てられた同じ一意の番号。
- 2 つの異なる属性は、同じ範囲の一意の数字から 2 つの異なる数字を割り当てていました。

多くの場合は、属性タイプごとに一意の番号を割り当てるだけで十分です。新しい従業員エントリーに *employeeID* を割り当てる際には、各従業員エントリーに一意の *employeeID* が割り当てられます。

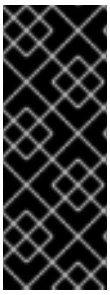
ただし、同じ範囲の数字から複数の属性に一意の番号を割り当てることが役に立つ場合もあります。たとえば、*uidNumber* と *gidNumber* を posixAccount エントリーに割り当てると、DNA プラグインは同じ数を両方の属性に割り当てます。これを行うには、マジック値を指定して、両方の管理属性を変更操作に渡します。ldapmodify の使用:

```
#  ldapmodify -D "cn=Directory Manager" -W -x

dn: uid=jsmith,ou=people,dc=example,dc=com
changetype: modify
add: uidNumber
uidNumber: 0
-
add:gidNumber
gidNumber: 0
```

DNA プラグインで複数の属性を処理する場合は、オブジェクトクラスが1つしか許可されない場合、プラグインはこれらの属性の1つにのみ一意の値を割り当てることができます。たとえば、posixGroup オブジェクトクラスは *uidNumber* 属性を許可しませんが、*gidNumber* を許可します。DNA プラグインが *uidNumber* と *gidNumber* の両方を管理する場合は、posixGroup エントリーが作成されると、*gidNumber* の一意の番号が *uidNumber* 属性および *gidNumber* 属性と同じ範囲から割り当てられます。プラグインが管理するすべての属性に同じプールを使用すると、一意の数字の割り当てを維持し、異なるエントリーの *uidNumber* と *gidNumber* が異なる範囲から割り当てられ、同じ 一意 の番号になる状況を防ぐことができます。

複数の属性が DNA プラグインで処理される場合は、1つの変更操作のエントリーで指定の管理属性のすべてに同じ値が割り当てられます。同じ範囲から別の数字を割り当てるには、別の変更操作を実施する必要があります。以下の例では、ldapmodify を使用してこれを行います。

```
# ldapmodify -D "cn=Directory Manager" -W -x
dn: uid=jsmith,ou=people,dc=example,dc=com
changetype: modify
add: uidNumber
uidNumber: 0
^D

# ldapmodify -D "cn=Directory Manager" -W -x
dn: uid=jsmith,ou=people,dc=example,dc=com
changetype: modify
add: employeeId
employeeId: magic
```



重要

DNA プラグインが一意の数字を複数の属性に割り当てるように設定する場合は、一意の番号を必要とする各属性にマジック値を指定する必要があります。1つの属性に一意の番号を提供するように DNA プラグインが設定されている場合には、これは必須ではありませんが、複数の属性に必要です。エントリーが範囲に対して定義された各タイプの属性を許可しない場合や、さらに重要なことに、定義されたすべての属性タイプをエントリーが許可しますが、属性のサブセットのみが一意の値を必要とする場合があります。

例7.7 DNA および一意の銀行口座番号

銀行の例では、顧客の *primaryAccount* 属性および *customerID* 属性に同じ一意の番号を使用します。銀行の例の管理者は、DNA プラグインが同じ範囲から両方の属性に一意の値を割り当てるよう設定していました。

また、銀行では、顧客 ID とプライマリーの口座番号と同じ範囲のセカンダリー口座に番号を割り当てますが、これらの数字をプライマリーの口座番号と同じにすることはできません。銀行の例の管理者は、DNA プラグインも *secondaryAccount* 属性を管理するように設定しますが、エントリーの作成、*secondaryAccount* 属性および *primaryAccount* 属性の割り当ての後 にのみ *customerID* 属性を追加します。これにより、*primaryAccount* および *customerID* は、同じ一意の番号を共有し、*secondaryAccount* 番号は完全に一意ですが、それでも同じ範囲の番号からのものです。

7.4.2. DNA プラグイン構文の確認

DNA プラグイン自体は、パスワードストレージスキームプラグインと同様のコンテナーエントリーです。DNA プラグインエントリーの下にある各 DNA エントリーは、DNA プラグインの新しい管理範囲を定義します。

DNA プラグインの新しい管理範囲を設定するには、コンテナーエントリーの下にエントリーを作成します。

最も基本的な設定は、1台のサーバーで分散数値の割り当てを設定することです。つまり、その範囲はサーバー間で共有されず、転送されません。基本的な DNA 設定エントリーでは、以下の 4 つの項目を定義します。

- 値が管理される属性で、*dnaType* 属性に設定されます。

- *dnaScope* 属性に設定される、エントリーを検索するためのベースとして使用するエントリー DN
- *dnaFilter* 属性に設定される、管理するエントリーの特定に使用する検索フィルター
- *dnaNextValue* 属性に設定される、次に割り当てることができる値 (エントリーの作成後、プラグインにより処理される)

cn=*DNA_config_entry*,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config エントリーでサポートされる属性の一覧は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス を参照してください。』

1つの属性タイプに対して、1台のサーバーに分散数値割り当てを設定するには、以下を実行します。

```
dn: cn=Account UIDs,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
objectClass: top
objectClass: dnaPluginConfig
cn: Account UIDs
dnatype: uidNumber
dnafilter: (objectclass=posixAccount)
dnascope: ou=people,dc=example,dc=com
dnaNextValue: 1
```

複数のサプライヤーが分散数値の割り当てに設定されている場合は、エントリーに範囲を転送するために必要な情報が含まれます。

- サーバーが割り当てることができる最大数。これにより、範囲の上限が設定されます。これは、複数のサーバーが番号を割り当てるときに論理的に必要です。これは *dnaMaxValue* 属性で設定されます。
- *dnaThreshold* 属性に設定された範囲転送を発生させるのに範囲が十分低いしきい値です。これが設定されていない場合、デフォルト値は 1 になります。
- *dnaRangeRequestTimeout* 属性に設定される、サーバーが転送を待ってハングしないようにするためのタイムアウト期間。これを設定しないと、デフォルト値は 10 (10 秒) になります。
- *dnaSharedCfgDN* 属性に設定した各サプライヤーの範囲情報を保存するすべてのサプライヤーサーバー間で共有される設定エントリー DN。

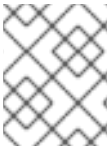
サーバーで割り当て可能な特定の数字の範囲は、*dnaNextRange* 属性で定義されます。これは、次に利用可能な範囲を表示し、サーバーが範囲が割り当てられているか、または使用しているため、プラグインにより自動的に管理されます。この範囲は「デッキ上」にあります。 別のサーバーには割り当てられておらず、ローカルの Directory Server が使用する場合は引き続き利用できます。

```
dn: cn=Account UIDs,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
objectClass: top
objectClass: dnaPluginConfig
cn: Account UIDs
dnatype: uidNumber
dnafilter: (objectclass=posixAccount)
dnascope: ou=People,dc=example,dc=com
dnanextvalue: 1
dnaMaxValue: 1300
dnasharedcfgdn: cn=Account UIDs,ou=Ranges,dc=example,dc=com
```

```
dnathreshold: 100
dnaRangeRequestTimeout: 60
dnaNextRange: 1301-2301
```

dnaNextRange 属性は、個別の特定範囲を他のサーバーに割り当てる必要がある場合にのみ明示的に設定する必要があります。*dnaNextRange* 属性に設定した範囲は、重複を避けるために、他のサーバーで利用可能な範囲から一意でなければなりません。他のサーバーからの要求がなく、*dnaNextRange* が設定されているサーバーが設定 *dnaMaxValue* に到達した場合は、次に値 (*dnaNextRange*の一部) がこのデッキから割り当てられます。

dnaNextRange 割り当ては、DNA 設定に設定されている *dnaThreshold* 属性によっても制限されます。*dnaNextRange* 用に別のサーバーに割り当てられる範囲は、範囲が *dnaNextRange* のデッキで利用可能であっても、サーバーのしきい値に違反できません。



注記

dnaNextRange 属性が明示的に設定されていない場合は、内部的に処理されます。自動的に処理される場合、*dnaMaxValue* 属性は次の範囲の上限として機能します。

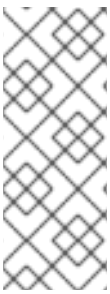
各サプライヤーは、範囲およびその接続設定に関する情報を含む、現在の範囲を別の設定エントリーで追跡します。このエントリーは、*dnasharedcfgdn* の場所の子です。設定エントリーは他のすべてのサプライヤーに複製されるため、各サプライヤーが設定を確認して、新しい範囲で問い合わせるサーバーを見つけることができます。以下に例を示します。

```
dn: dnaHostname=ldap1.example.com+dnaPortNum=389,cn=Account
UIDs,ou=Ranges,dc=example,dc=com
objectClass: dnaSharedConfig
objectClass: top
dnahostname: ldap1.example.com
dnaPortNum: 389
dnaSecurePortNum: 636
dnaRemainingValues: 1000
```

7.4.3. 一意の番号割り当ての設定

一意の数字分散は、DNA プラグインの異なるインスタンスを作成して設定されます。これらの DNA プラグインインスタンスは、コマンドラインからのみ作成できますが、Directory Server コンソールから編集できます。

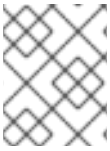
7.4.3.1. 一意の番号割り当ての設定



注記

一意の番号が割り当てられている属性には、同等インデックスが設定されている必要があります。*dnaNextvalue* がすでに取得されているかどうかを確認するために、サーバーは、内部的にソートされた検索を実行する必要があります。これには、適切な順序のマッチングルールとともに整数属性で等価インデックスが必要であるかを確認します。

インデックスの作成については、「標準インデックスの作成」を参照してください。



注記

すべてのサプライヤーサーバーに DNA プラグインを設定し、数字の範囲の値と重複しないように注意してください。

1. 複製されたサブツリーに共有コンテナーエントリーを作成します。以下の例では、ldapmodify を使用してこれを行います。

```
dn: ou=Ranges,dc=example,dc=coma
changetype: add
objectclass: top
objectclass: extensibleObject
objectclass: organizationalUnit
ou: Ranges


dn: cn=Account UIDs,ou=Ranges,dc=example,dc=coma
changetype: add
objectclass: top
objectclass: extensibleObject
cn: Account UIDs
```

2. DNA プラグインを有効にし、これを動的として設定します。デフォルトでは、（コンテナーエントリーである）プラグインエントリーは無効になっています。動的プラグインの設定に関する詳細は、「プラグインを動的に有効化」 を参照してください。

3. コンテナーエントリーの下に、新しい DNA プラグインインスタンスを作成します。以下に例を示します。

注記

一意の番号割り当て(*dnaType*)を持つエントリー属性を設定するプラグイン属性は多値です。同じプラグインインスタンスに複数の属性が設定されている場合、その番号の割り当ては同じ範囲から取得されます。異なる範囲を使用するには、異なるプラグインインスタンスを設定します。

ldapmodify の使用:

```
dn: cn=Account UIDs,cn=Distributed Numeric Assignment
Plugin,cn=plugins,cn=config
changetype: add
objectClass: top
objectClass: dnaPluginConfig
cn: Account UIDs
dnatype: uidNumber
dnafilter: (objectclass=posixAccount)
dnascope: ou=People,dc=example,dc=com
dnanextvalue: 1
dnaMaxValue: 1300
dnasharedcfgdn: cn=Account UIDs,ou=Ranges,dc=example,dc=com
dnathreshold: 100
dnaRangeRequestTimeout: 60
dnaMagicRegen: magic
```

cn=*DNA_config_entry*,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config エントリーでサポートされる属性の一覧は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス を参照してください。』

4. マルチマスターレプリケーションのサーバーでは、接続情報および範囲を指定するホストの設定エントリーを作成します。

   エントリーの DN は、ホスト名とポート番号の組み合わせです (dnaHostname+dnaPortNum)。

   ldapmodify の使用:

   ```
   dn: dnaHostname=ldap1.example.com+dnaPortNum=389,cn=Account
   UIDs,ou=Ranges,dc=example,dc=com
   changetype: add
   objectClass: dnaSharedConfig
   objectClass: top
   dnahostname: ldap1.example.com
   dnaPortNum: 389
   dnaSecurePortNum: 636
   dnaRemainingValues: 1000
   ```

5. サーバーが動的プラグインを有効にするために設定されていない場合は、サーバーを再起動して新しいプラグインインスタンスを読み込みます。

   ```
   # systemctl restart dirsrv@instance
   ```

7.4.3.2. コンソールでの DNA プラグインの編集

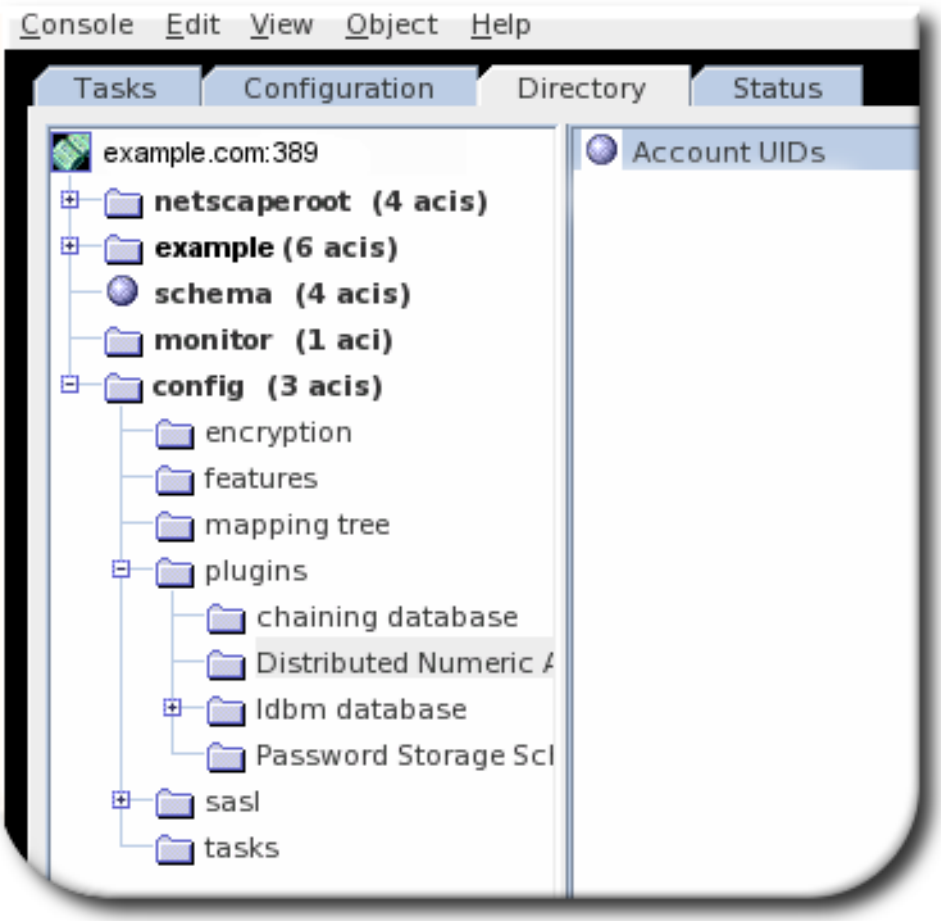
注記

一意の番号が割り当てられている属性には、同等インデックスが設定されている必要があります。*dnaNextvalue* がすでに取得されているかどうかを確認するために、サーバーは、内部的にソートされた検索を実行する必要があります。これには、適切な順序のマッチングルールとともに整数属性で等価インデックスが必要であるかを確認します。

インデックスの作成については、「標準インデックスの作成」を参照してください。

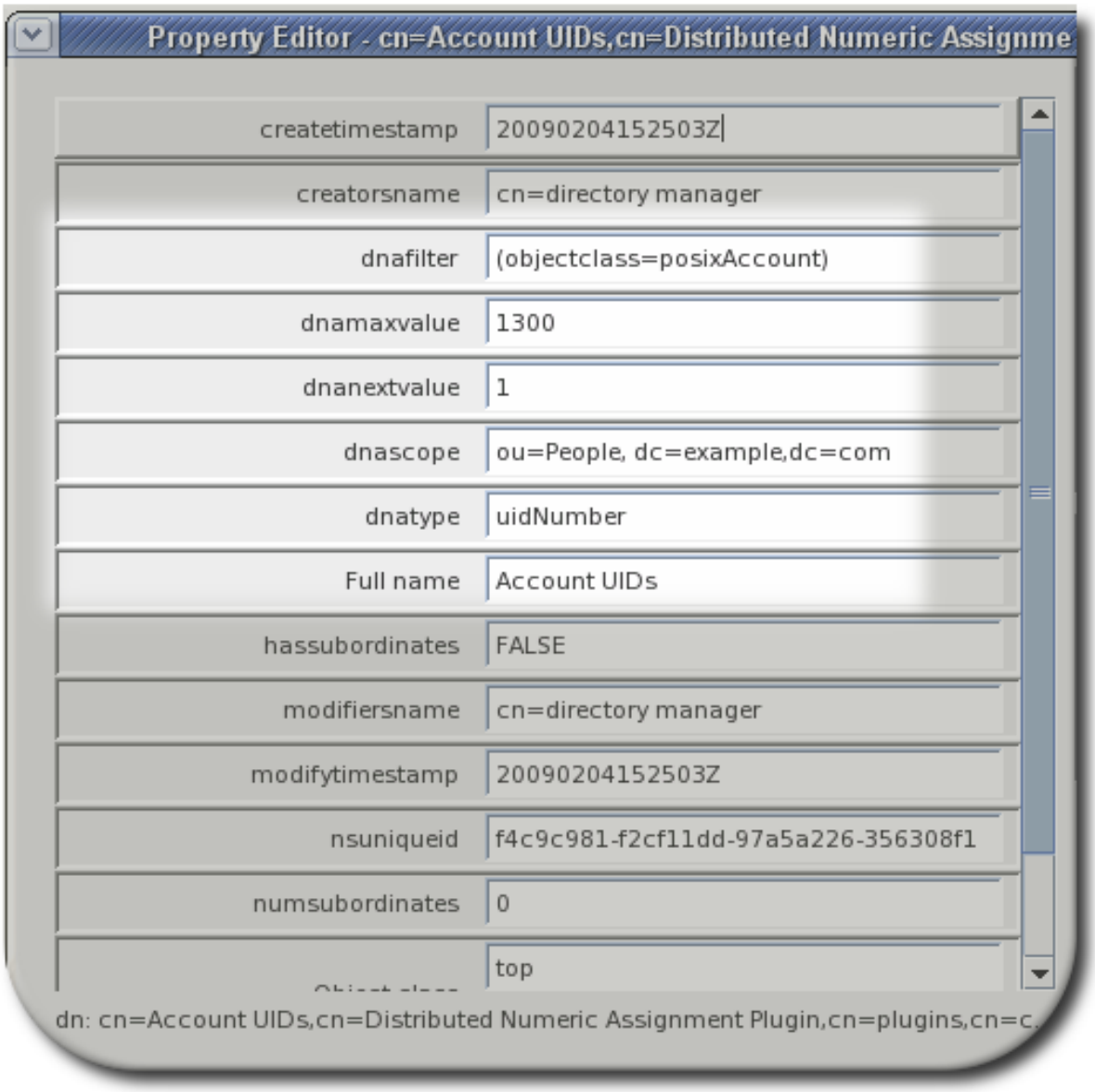
Directory Server コンソールを使用して、DNA プラグインインスタンスを編集できます。

1. **Directory** タブをクリックします。
2. **config** ディレクトリーを開き、**plugins** フォルダーを展開します。
3. **Distributed Numeric Assignment** プラグインフォルダーをクリックします。すべての DNA プラグインインスタンスがメインウィンドウに一覧表示されます。



4. DNA インスタンスエントリーを強調表示し、**Advanced** リンクを右クリックして、プロパティーエディターを開きます。

5. DNA 関連の属性を編集します。



7.4.4. Distributed Number Assignment プラグインのパフォーマンスに関する注意事項

DNA 構成が動的に変更されると、スレッドロックの問題が発生する可能性があります。そのため、新しい構成のスレッドは解放されないため、DNA 構成にアクセスする新しい操作 (DNAタスクやDNA構成への追加の変更など) は古い構成にアクセスします。これにより、操作が古い設定を使用するか、操作がハングする可能性があります。

これを回避するには、動的 DNA 設定の変更の間隔を 35 秒に設定します。これは、DNA 設定の変更と、DNA プラグイン操作を発生させるディレクトリーエントリーの変更の両方の間にスリープ状態または遅延があることを意味します。

第8章 エントリーの編成とグループ化

ディレクトリーに含まれるエントリーは、ユーザーアカウントの管理を簡素化するために、さまざまな方法でグループ化できます。Red Hat Directory Server は、エントリーのグループ化やエントリー間で属性を共有するさまざまな方法に対応します。ロールおよびサービスのクラスによって提供される機能を完全に活用するには、ディレクトリーのデプロイメントを計画するときにディレクトリートポロジーを決定します。

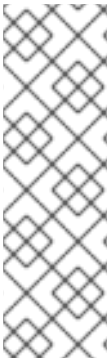
8.1. グループの使用

オペレーティングシステムと同様に、Directory Server のグループにユーザーを追加できます。グループはロールとして他の方法で機能します。ロールを使用している場合には、割り当てられたロールの DN はユーザーオブジェクトの ***nsRoleDN*** 属性に保存されます。グループを使用する場合は、このグループのメンバーであるユーザーの DN は、グループオブジェクトの ***member*** 属性に保存されます。memberOf プラグインを有効にした場合、次にユーザーがメンバーであるグループは、ユーザーオブジェクトの ***memberOf*** 属性に追加で保存されます。このプラグインを有効にすると、グループにもロールの利点があり、ロールの使用時と同様にユーザーのグループメンバーシップを一覧表示できます。また、グループはロールよりも高速です。

memberOf プラグインの使用方法は、「ユーザーエントリーにおけるグループメンバーシップの一覧表示」を参照してください。

8.1.1. コンソールで静的グループの作成

静的グループは、任意の数のユーザーの DN 属性に同じグループ値を指定してエントリーを整理します。

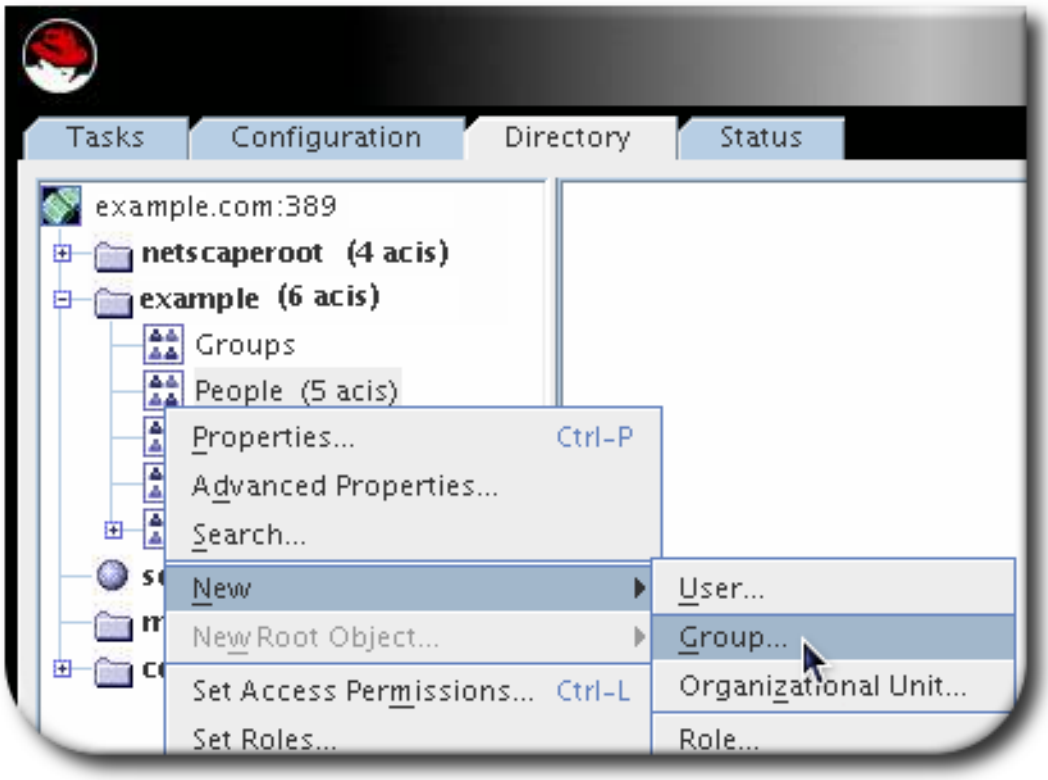


注記

ユーザーがリモート Directory Server にエントリー（チェーンデータベースなど）がある場合は、静的グループを定義するエントリーを持つ Directory Server とは異なる場合は、Referential Integrity プラグインを使用して、削除されたユーザーエントリーが静的グループから自動的に削除されるようにします。

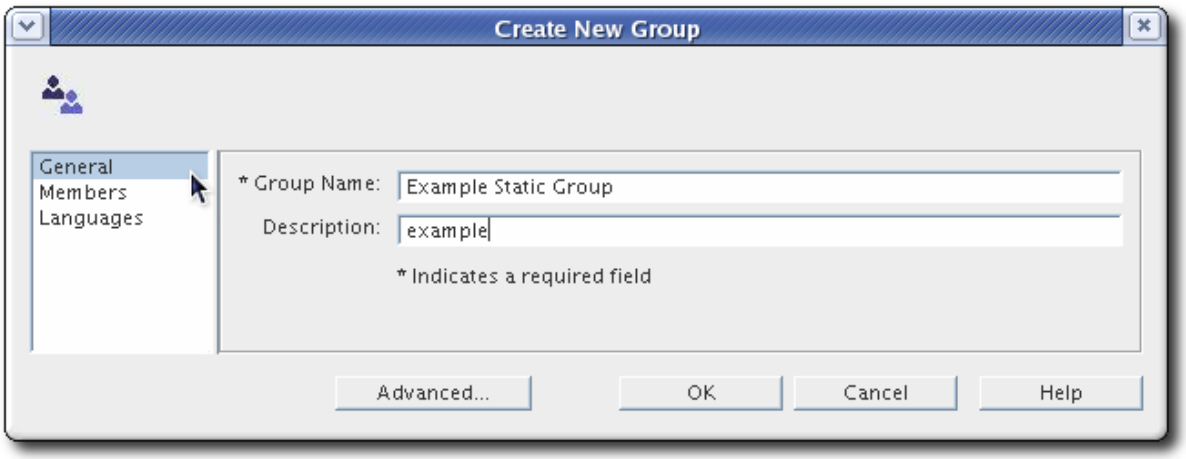
Referential Integrity プラグインには、パフォーマンスとアクセス制御に関する考慮事項がいくつかあります。チェーンで参照整合性を使用する方法は、「シャーシポリシーの設定」 を参照してください。

1. Directory Server コンソールで、Directory タブを選択します。
2. 左側のペインで、新しいグループを追加するエントリーを右クリックし、**New > Group** を選択します。



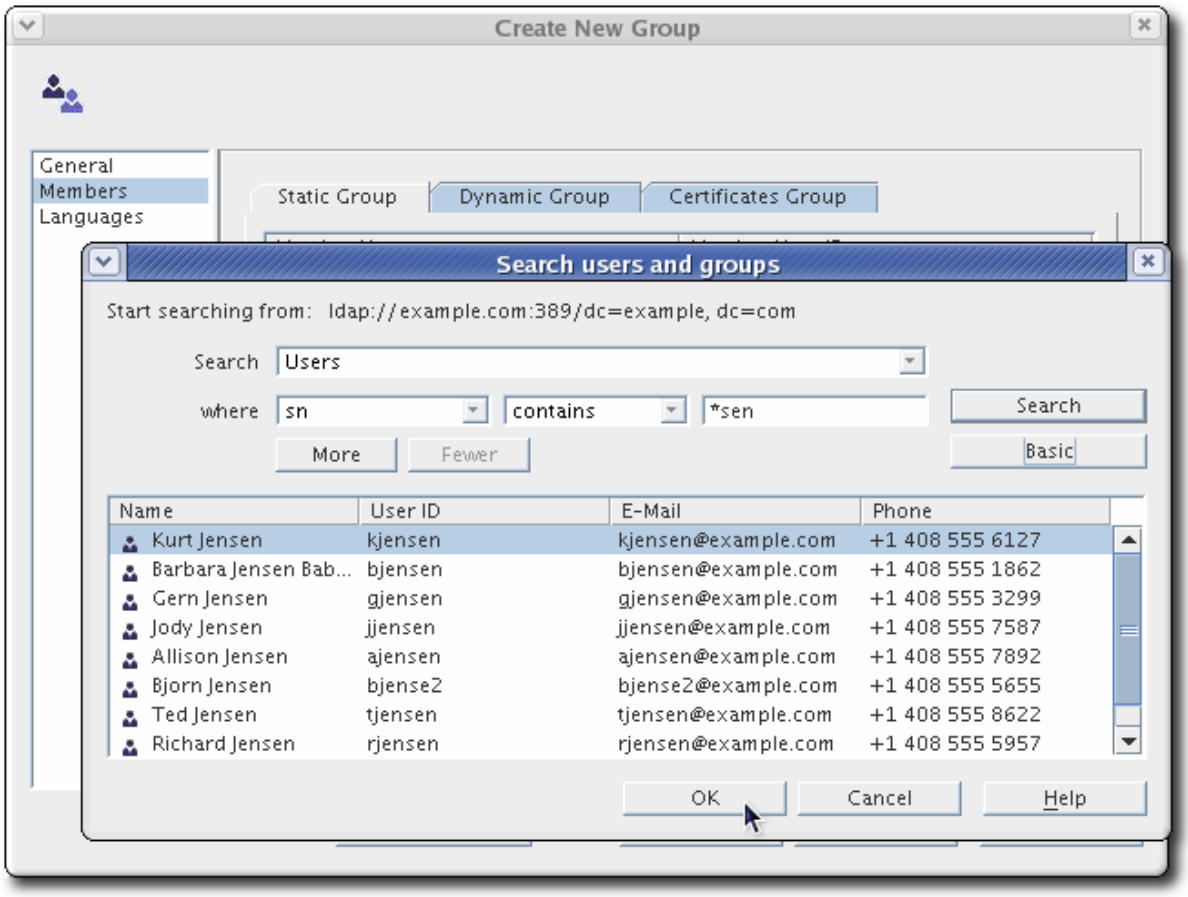
または、**Object** メニューに移動し、**New** > **Group** を選択します。

3. 左側のペインで **General** をクリックします。**Group Name** フィールドに新規グループの名前を入力します（名前は必須です）、**Description** フィールドに新しいグループの説明を入力します。

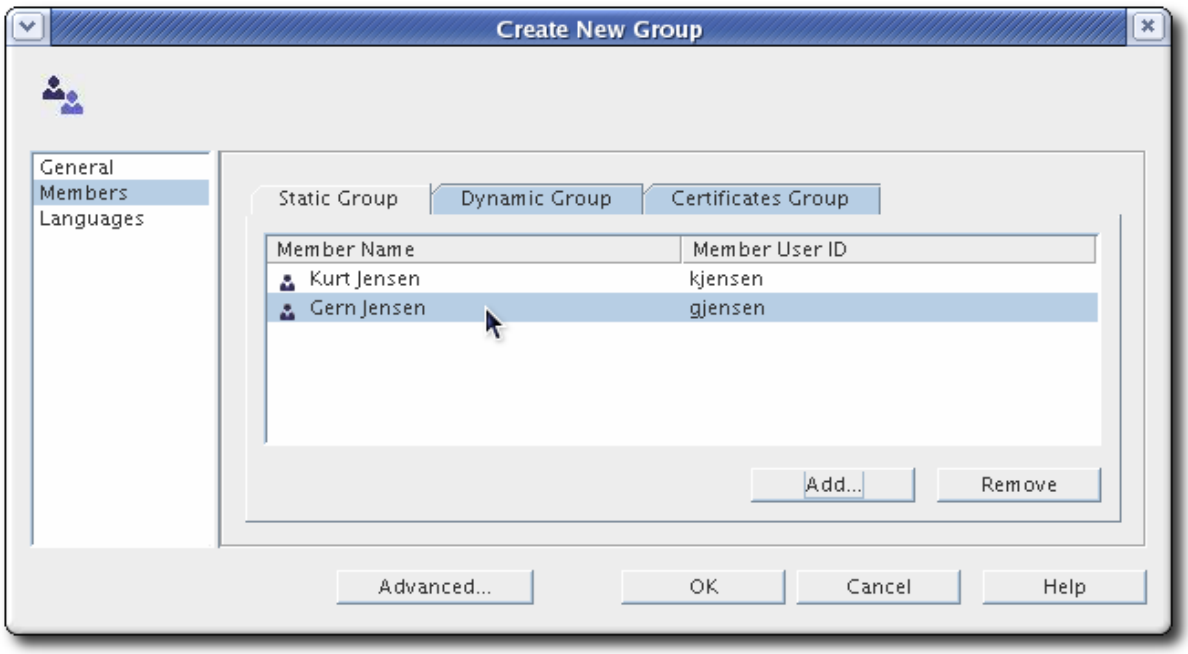


4. 左側のペインで **Members** をクリックします。右側のペインで、静的グループ タブを選択します。**Add** をクリックして、新しいメンバーをグループに追加します。

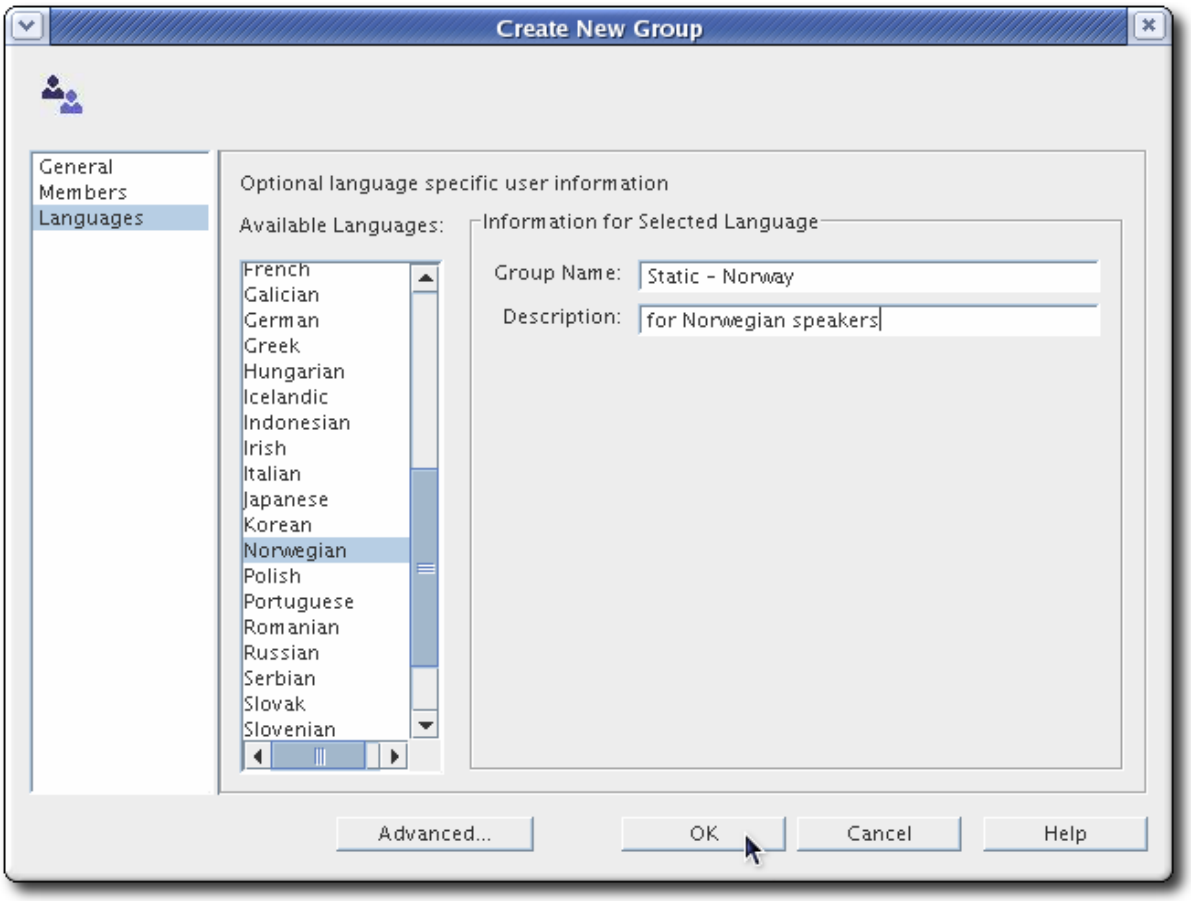
5. **Search** ドロップダウンリストで、検索するエントリーのソート（ユーザー、グループ、またはその両方）を選択してから **Search** をクリックします。



6. 返されたエントリーからメンバーを選択し、OK をクリックします。

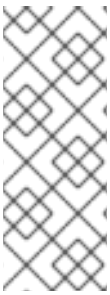


7. 左側のペインで Languages をクリックし、そのグループに言語固有の情報を追加します。



8. **OK** をクリックして新規グループを作成します。右側のペインに表示されます。

静的グループを編集するには、グループエントリーをダブルクリックして、エディターウィンドウで変更を行います。変更を表示するには、**View** メニューに移動し、**Refresh** を選択します。



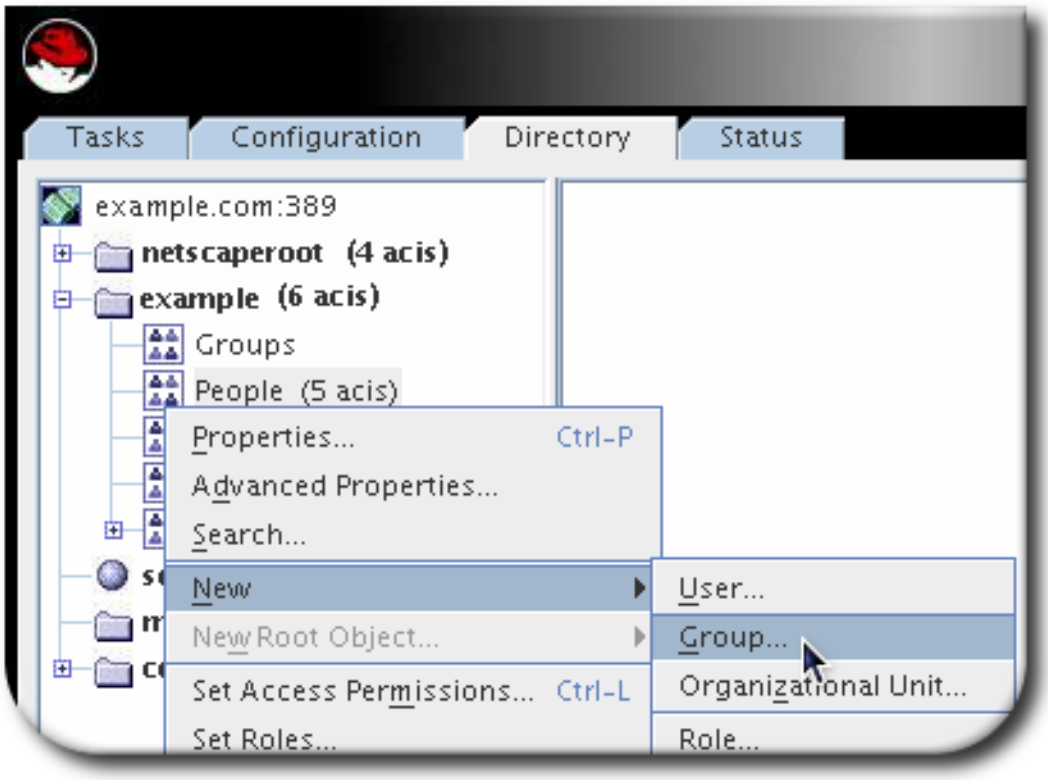
注記

静的グループを管理するコンソールは、ユーザーの検索の VLV インデックスがない場合に、検索操作中に可能なすべての選択を表示できない場合があります。この問題は、ユーザーの数が 1000 以上で、検索用の VLV インデックスがない場合にだけ発生します。この問題を回避するには、フィルター (**objectclass=person**) および scope サブツリーを使用して、user 接尾辞の VLV インデックスを作成します。「コマンドラインから参照インデックスの作成」 を参照してください。

8.1.2. コンソールでの動的グループの作成

動的グループは、DN に基づいてユーザーをフィルタリングし、それらを 1 つのグループに含めます。

1. Directory Server コンソールで、Directory タブを選択します。
2. 左側のペインで、新しいグループを追加するエントリーを右クリックし、**New > Group** を選択します。

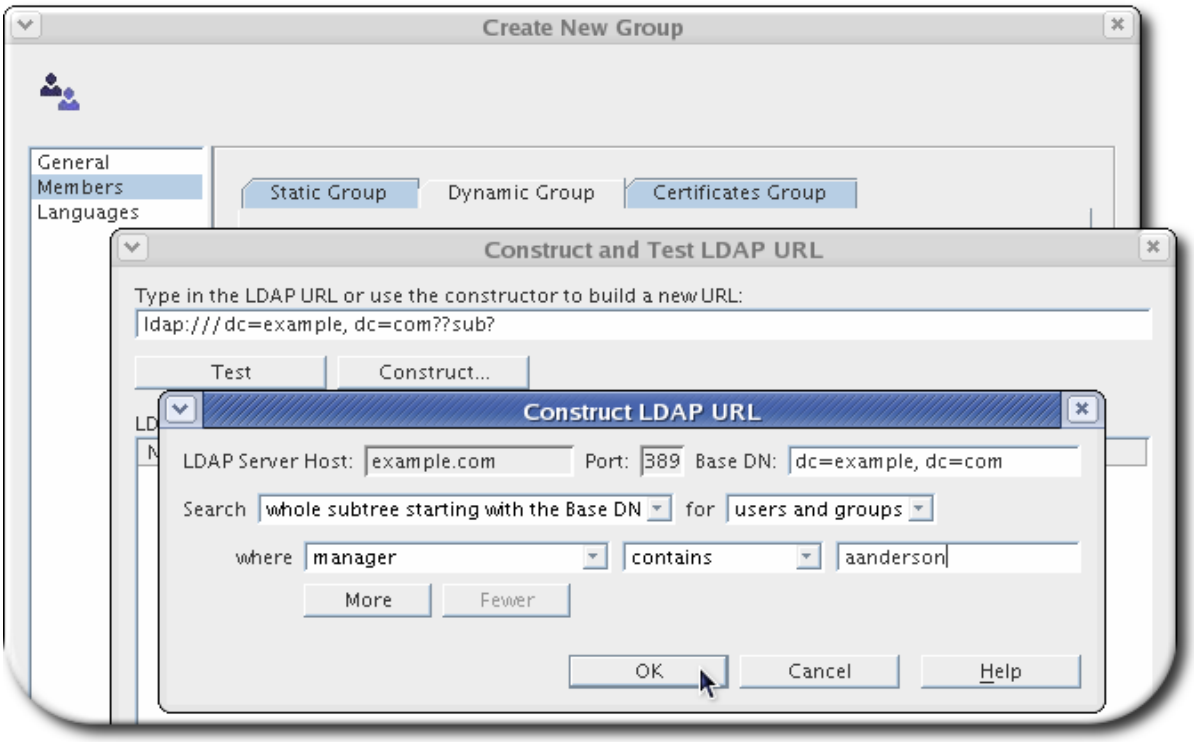


または、**Object** メニューに移動し、**New** > **Group** を選択します。

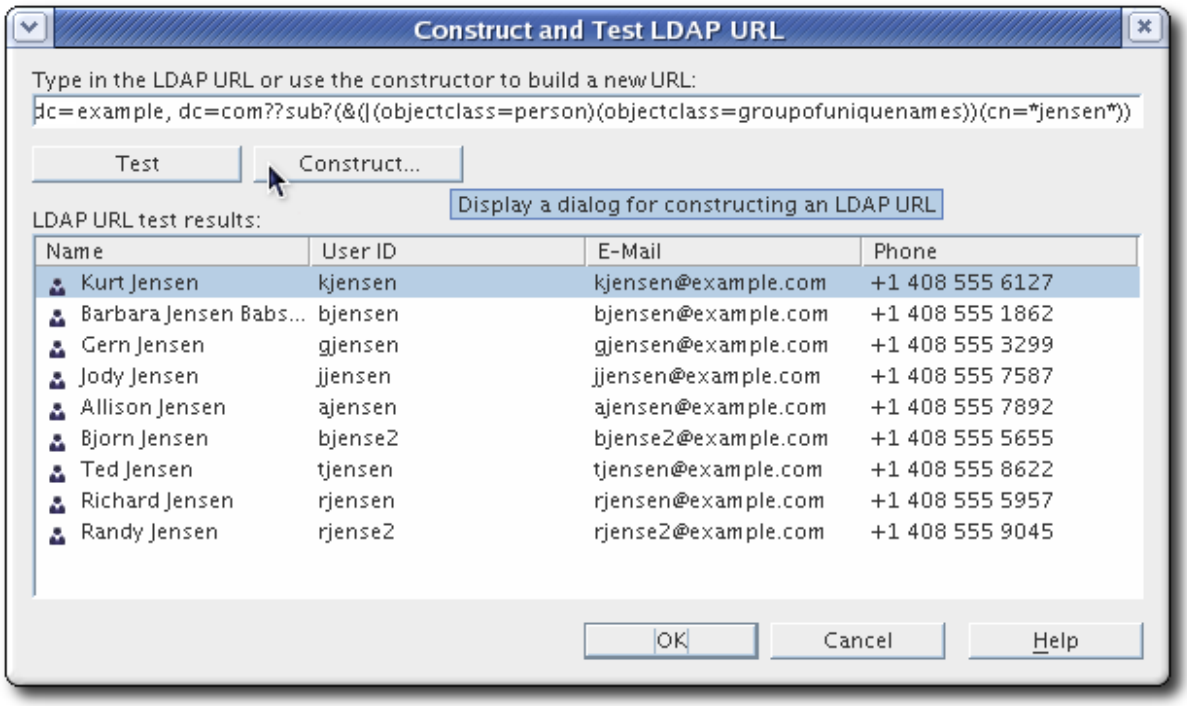
3. 左側のペインで **General** をクリックします。**Group Name** フィールドに新規グループの名前を入力します（名前は必須です）、**Description** フィールドに新しいグループの説明を入力します。

4. 左側のペインで **Members** をクリックします。右側のペインで、**Dynamic Group** タブを選択します。**Add** をクリックして、データベースのクエリーを行うための LDAP URL を作成します。

5. テキストフィールドに LDAP URL を入力するか、LDAP URL の構成 で説明する **Construct to** を選択します。



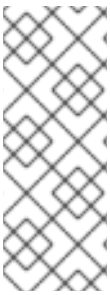
この結果には、フィルターに対応する現在のエントリー（グループメンバー）が表示されます。



6. 左側のペインで **Languages** をクリックし、そのグループに言語固有の情報を追加します。

7. **OK** をクリックします。右側のペインに新しいグループが表示されます。

動的グループを編集するには、グループエントリーをダブルクリックしてエディターウィンドウを開き、動的グループに変更を加えます。グループの変更を表示するには、**View** メニューに移動し、**Refresh** を選択します。



注記

動的グループを管理するコンソールは、ユーザーの検索の VLV インデックスがない場合に、検索操作中に可能なすべての選択を表示できない場合があります。この問題は、ユーザーの数が 1000 以上で、検索用の VLV インデックスがない場合に発生する可能性があります。この問題を回避するには、フィルター **(objectclass=person)** および scope サブツリーを使用して、user 接尾辞の VLV インデックスを作成します。「コマンドラインから参照インデックスの作成」 を参照してください。

8.1.3. コマンドラインでのグループの作成

コマンドラインから静的グループと動的グループの両方を作成するプロセスは、同様のプロセスです。グループエントリーには、グループ名、グループの種類、およびメンバー属性が含まれます。

グループタイプにはいくつかのオプションがあります。詳細は、『Red Hat Directory Server 10 の設定、コマンド、およびファイルリファレンス』 を参照してください。この場合の グループのタイプは、所有するメンバー属性を定義するタイプを指します。

- **groupOfNames** (推奨) は、任意のエントリーの追加を可能にする単純なグループです。これのメンバーを判断するために使用される属性は *member* です。

- **groupOfNames** などの **groupOfUniqueNames** は、ユーザー DN をメンバーとして一覧表示しますが、メンバーは一意でなければなりません。これにより、ユーザーをグループメンバーとして複数回追加しないようにできます。これは、セルフ参照グループメンバーシップを防ぐ 1

つの方法です。これのメンバーを判断するために使用される属性は *uniqueMember* です。

- **groupOfURLs** は、LDAP URL の一覧を使用して、メンバーシップの一覧をフィルタリングして生成します。このオブジェクトクラスはすべての動的グループに必要で、**groupOfNames** および **groupOfUniqueNames** と共に使用できます。

- **groupOfCertificates** は、グループメンバーを識別するための証明書 (実際には証明書名) を検索して識別するために LDAP フィルターを使用するという点で、**groupOfURLs** と似ています。これは、グループに特別なアクセスパーミッションを付与できるため、グループベースのアクセス制御に役立ちます。これのメンバーを判断するために使用される属性は *memberCertificate* です。

表8.1「動的および静的のグループスキーマ」 コマンドラインから作成されるグループのデフォルト属性を一覧表示します。

表8.1 動的および静的のグループスキーマ

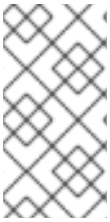
グループのタイプ	グループオブジェクトクラス	member 属性
静的	groupOfUniqueNames	uniqueMember
動的	groupOfUniqueNames groupOfURLs	memberURL

静的グループエントリーは、グループの特定のメンバーを一覧表示します。たとえば、**ldapmodify** を使用するには、以下を実行します。

```
dn: cn=static group,ou=Groups,dc=example,dc=com
changetype: add
objectClass: top
objectClass: groupOfUniqueNames
cn: static group
description: Example static group.
uniqueMember: uid=mwhite,ou=People,dc=example,dc=com
uniqueMember: uid=awhite,ou=People,dc=example,dc=com
```

動的グループは、少なくとも 1 つの LDAP URL を使用して、グループに属するエントリーを識別し、複数の LDAP URL を指定できます。または、**groupOfUniqueNames** のような別のグループオブジェクトクラスで使用する場合は、動的 LDAP URL とともにいくつかのグループメンバーを明示的に一覧表示できます。たとえば、**ldapmodify** を使用するには、以下を実行します。

```
dn: cn=dynamic group,ou=Groups,dc=example,dc=com
changetype: add
objectClass: top
objectClass: groupOfUniqueNames
objectClass: groupOfURLs
cn: dynamic group
description: Example dynamic group.
memberURL: ldap:///dc=example,dc=com??sub?(&(objectclass=person)(cn=*sen*))
```



注記

memberOf プラグインは、動的に生成されるグループメンバーシップをサポートしません。属性にグループメンバーを一覧表示する代わりに *memberURL* 属性を設定すると、memberOf プラグインはフィルターに一致するユーザーオブジェクトに *memberOf* 属性を追加しません。

8.1.4. ユーザーエントリーにおけるグループメンバーシップの一覧表示

グループに属するエントリーは、グループエントリー自体で定義されます。これにより、グループを確認し、そのメンバーを確認し、グループメンバーシップを一元管理できるようになります。ただし、1 つのユーザーが属するグループを確認することは適切な方法ではありません。ロールがあるので、メンバーシップを示すユーザーエントリーには何もありません。

MemberOf プラグインは、グループメンバーシップの一覧を対応するユーザーエントリーに関連付けます。

MemberOf プラグインはグループエントリーのメンバー属性を分析し、メンバーのエントリーに対応する *memberOf* 属性を自動的に書き込みます。(デフォルトでは、これにより *member* 属性を確認しますが、複数の属性インスタンスを使用して複数の異なるグループタイプをサポートすることができます。)

メンバーシップが変更になると、プラグインはユーザーエントリーの *memberOf* 属性を更新します。MemberOf プラグインは、ネスト化されたグループメンバーシップを含むエントリーを確認して、ユーザーが属するグループを表示する方法を提供します。ネスト化されたグループを介してメンバーシップを追跡することは非常に困難ですが、MemberOf プラグインには、すべてのグループの (直接および間接的な) メンバーシップが表示されます。

MemberOf プラグインは、動的グループや循環グループではなく、静的グループのメンバー属性を管理します。

8.1.4.1. memberOf プラグインを使用する場合の考慮事項

本セクションでは、memberOf プラグインを使用する際に重要な考慮事項を説明します。

レプリケーショントポロジーでの memberOf プラグインの使用

レプリケーショントポロジーで *memberOf* 属性を管理する方法は 2 つあります。

- トポロジー内のすべてのマスターで memberOf プラグインおよび読み取り専用レプリカサーバーを有効にします。この場合、すべてのレプリカ合意で、レプリケーションから *memberOf* 属性を除外する必要があります。属性の除外に関する詳細は、「一部レプリケーションを使用した属性のサブセットの複製」を参照してください。
- memberOf プラグインは、トポロジー内のすべてのマスターサーバーでのみ有効にします。そのためには、以下を実行します。
  - レプリカ合意ですべての書き込みが有効なマスターに対する *memberOf* 属性のレプリケーションを無効にする必要があります。属性の除外に関する詳細は、「一部レプリケーションを使用した属性のサブセットの複製」を参照してください。
  - *memberOf* 属性のレプリケーションを、すべての読み取り専用レプリカに対して、レプリカ合意で有効にする必要があります。
  - 読み取り専用レプリカでは、memberOf プラグインを有効にしないでください。

分散データベースでの memberOf プラグインの使用

「データベースの作成」で説明されているように、ディレクトリーのサブツリーを個別のデータベースに保存できます。デフォルトでは、memberOf プラグインはグループと同じデータベース内に保存されるユーザーエントリーのみを更新します。プラグインがグループとして異なるデータベースのユーザーも更新できるようにするには、*memberOfAllBackends* パラメーターを on に設定する必要があります。「コンソールからの MemberOf プラグインの編集」を参照してください。

8.1.4.2. memberOf プラグインで必要なオブジェクトクラス

デフォルトでは、memberOf プラグインは nsMemberOf オブジェクトクラスをオブジェクトに追加し、*memberOf* 属性を提供します。このオブジェクトクラスは、この目的のために任意のオブジェクトに安全に追加でき、このプラグインを正しく動作させるためにこれ以上のアクションは必要ありません。代わりに、inetUser オブジェクトクラスまたは inetAdmin オブジェクトクラスが含まれるユーザーオブジェクトを作成できます。どちらのオブジェクトクラスも *memberOf* 属性をサポートします。

ネスト化されたグループを設定するには、グループは extensibleObject オブジェクトクラスを使用する必要があります。

注記

ディレクトリーエントリーに必要な属性をサポートするオブジェクトクラスが含まれていない場合、操作は以下のエラーで失敗します。

```
LDAP: error code 65 - Object Class Violation
```

8.1.4.3. MemberOf プラグイン構文

MemberOf プラグインインスタンスは、ポーリングするグループメンバー属性 (*memberOfGroupAttr*) と、メンバーのユーザーエントリー (*memberOfAttr*) で作成および管理する属性の 2 つの属性を定義します。

memberOfGroupAttr 属性は多値です。異なるタイプのグループは異なるメンバー属性を使用するため、複数の *memberOfGroupAttr* 属性を使用すると、プラグインで複数のグループタイプを管理できます。

プラグインインスタンスは、MemberOf プラグインを識別するためのプラグインパスと関数も提供し、プラグインを有効にするための状態設定が含まれます。これらの両方は、すべてのプラグインに必要です。デフォルトの MemberOf プラグインが、例8.1「デフォルトの MemberOf プラグインエンティティー」に表示されます。

例8.1 デフォルトの MemberOf プラグインエンティティー

```
dn: cn=MemberOf Plugin,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: MemberOf Plugin
nsslapd-pluginPath: libmemberof-plugin
nsslapd-pluginInitfunc: memberof_postop_init
nsslapd-pluginType: postoperation
nsslapd-pluginEnabled: on
nsslapd-plugin-depends-on-type: database
memberOfGroupAttr: member
```

```
memberOfGroupAttr: uniqueMember
memberOfAttr: memberOf
memberOfAllBackends: on
nsslapd-pluginId: memberOf
nsslapd-pluginVersion: X.Y.Z
nsslapd-pluginVendor: Red Hat, Inc.
nsslapd-pluginDescription: memberOf plugin
```

例で使用するパラメーターの詳細と、設定可能なパラメーターの詳細は、『Red Hat Directory Server コマンド、設定、およびファイルリファレンス』 の『MemberOf プラグイン属性』 セクションを参照してください。

注記

1つのメンバー属性 (デフォルトでは *member*) のみを許可した古いバージョンの Directory Server との後方互換性を維持するために、プラグイン設定で使用される新しいメンバー属性に加えて、*member* グループ属性または以前のメンバー属性を含める必要がある場合があります。

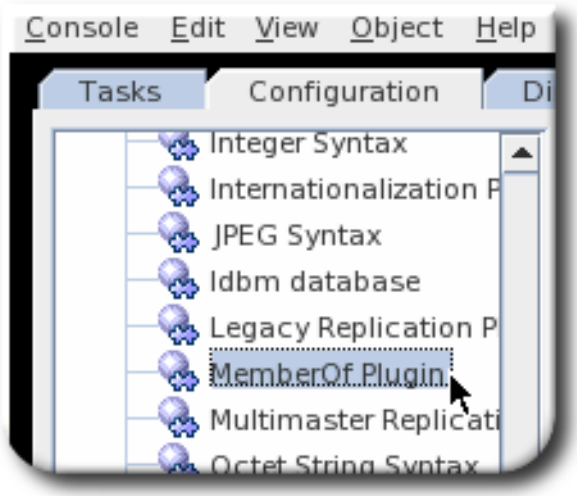
```
memberOfGroupAttr: member
memberOfGroupAttr: uniqueMember
```

8.1.4.4. MemberOf プラグインのインスタンスの設定

ディレクトリーで使用されるグループのタイプに応じて、MemberOf プラグインで定義された属性を変更できます。

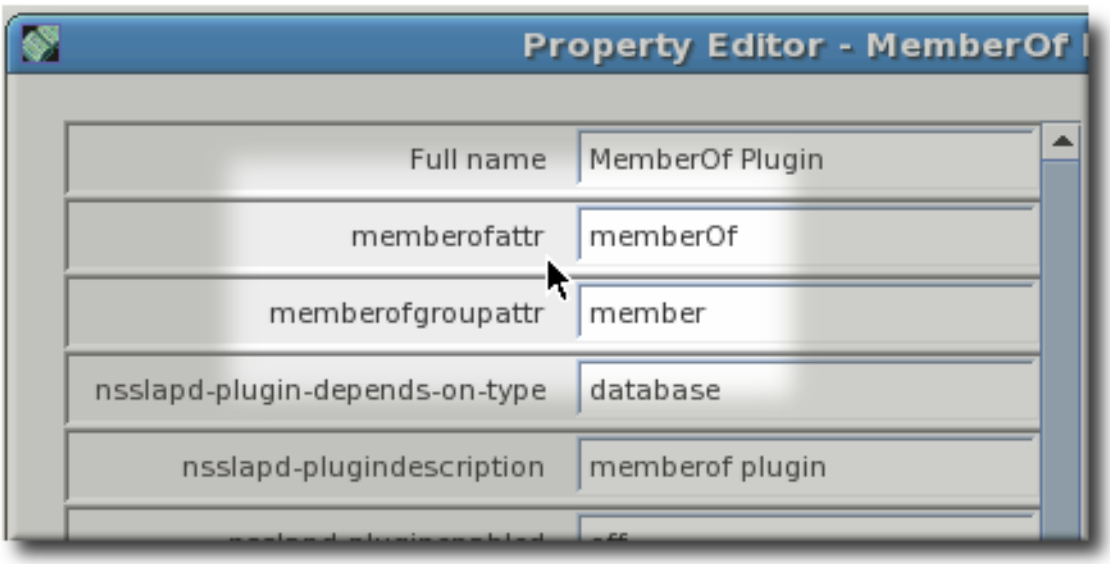
8.1.4.4.1. コンソールからの MemberOf プラグインの編集

1. **Configuration** タブを選択し、**Plugins** フォルダーに展開します。
2. **Memberof Plugin** エントリーまでスクロールします。



3. プラグインが有効化されていることを確認します。これはデフォルトで無効にされます。
4. **Advanced** ボタンをクリックして **Advanced Properties Editor** を開きます。
5. *memberOfGroupAttr* 属性は、サーバーがメンバーエントリーを識別するために使用するグ

ループエントリーの属性を設定します。この属性は、異なるグループ/メンバータイプに対して複数回使用できます。*memberOfAttr*属性は、プラグインがユーザーエントリーで作成および管理する属性を設定します。



6. 変更を保存します。

7. Directory Server が動的プラグインを有効にするために設定されていない場合は、サーバーを再起動してプラグインを更新します。

8.1.4.4.2. コマンドラインでの MemberOf プラグインの編集

1. MemberOf プラグインを有効にします。**ldapmodify** の使用:

```
dn: cn=MemberOf Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

2. グループメンバーエントリー属性に使用する属性を設定します。デフォルトの属性は *member* で、**replace** コマンドを使用して変更できます。*memberOfGroupAttr*属性は多値であるため、追加のメンバータイプを定義に追加できます。たとえば、**ldapmodify** を使用するには、以下を実行します。

```
dn: cn=MemberOf Plugin,cn=plugins,cn=config
changetype: modify
add: memberOfGroupAttr
memberOfGroupAttr: uniqueMember

add: memberOfGroupAttr
memberOfGroupAttr: customMember-
```

3. グループメンバーシップを表示するようにユーザーエントリーに設定する属性を設定します。たとえば、**ldapmodify** を使用するには、以下を実行します。

```
dn: cn=MemberOf Plugin,cn=plugins,cn=config
changetype: modify
replace: memberOfAttr
```

```
memberOfAttr: memberOf
```

4. オプション。デプロイメントで分散データベースを使用する場合は、*memberOfAllBackends* 属性を有効にして、ユーザーエントリーについてローカルのデータベースだけではなく、すべてのデータベースを検索します。ldapmodify の使用:

```
dn: cn=MemberOf Plugin,cn=plugins,cn=config
changetype: modify
replace: memberOfAllBackends
memberOfAllBackends: on
```

5. Directory Server が動的プラグインを有効にするために設定されていない場合は、サーバーを再起動して変更した新しいプラグインインスタンスを読み込みます。

8.1.4.5. memberOf プラグイン共有の設定

プラグイン設定を複製すると、ネットワーク上で一貫した設定を維持するのに役立ちます。これは特に大規模なデプロイメントで役に立ちます。マスターレプリケーションサーバーの設定のみを更新し、変更が他のすべてのサーバーに複製されます。

memberOf プラグイン設定は、cn=config 接尾辞以外の、バックエンドまたは接尾辞の共有設定エントリーに保存できます。

プラグインエントリーでは、nsslapd-pluginConfigArea 属性を使用して共有設定の場所を指定します。

```
nsslapd-pluginConfigArea: entry_DN
```

nsslapd-pluginConfigArea 属性をすべてのレプリカの同じプラグインエントリーに設定した後、レプリケーションは今後の設定変更をすべて処理します。

以下の表は、共有設定エントリーで使用できる属性を示しています。

表8.2 memberOf プラグイン共有設定の属性

設定属性	値	例
memberOfAttr (required)	属性名	memberOf
memberOfGroupAttr (required)	属性名	uniqueMember
memberOfAllBackends	on|off	off
memberOfEntryScope	エントリー DN	ou=people,dc=example,dc=com
memberOfSkipNested	on|off	on
memberOfEntryScopeExcludeSubtree	エントリー DN	ou=other,dc=example,dc=com

以下の例では、nsslapd-pluginConfigArea が設定されています。そのため、プラグインエントリーの設定は無視されます。

```
dn: cn=MemberOf Plugin,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: MemberOf Plugin
nsslapd-pluginPath: libmemberof-plugin
nsslapd-pluginInitfunc: memberof_postop_init
nsslapd-pluginType: postoperation
nsslapd-pluginEnabled: on
nsslapd-plugin-depends-on-type: database
memberOfGroupAttr: member
memberOfAttr: memberOf
nsslapd-pluginConfigArea: cn=memberOf plugin configuration,dc=example,dc=com
```

この例では、**memberOf** プラグインは **member** ではなく **uniquemember** グループ属性を使用します。

```
dn: cn=memberOf plugin configuration,dc=example,dc=com
objectClass: top
objectClass: extensibleObject
cn: MemberOf Plugin Configuration
memberOfGroupAttr: uniquemember
memberOfAttr: memberOf
```

8.1.4.6. MemberOf プラグインのスコープの設定

複数のバックエンドまたは複数のネストされたサフィックスを設定した場合は、***memberOfEntryScope*** パラメーターおよび ***memberOfEntryScopeExcludeSubtree*** パラメーターを使用して、**MemberOf** プラグインが動作するサフィックスを設定できます。

ユーザーをグループに追加する場合、**MemberOf** プラグインは、ユーザーおよびグループの両方がプラグインのスコープにある場合に限り ***memberOf*** 属性をグループに追加します。たとえば、***dc=example,dc=com*** 内のすべてのエントリーで機能するように **MemberOf** プラグインを設定し、***ou=private,dc=example,dc=com*** のエントリーを除外するには、以下を設定します。

```
memberOfEntryScope: dc=example,dc=com
memberOfEntryScopeExcludeSubtree: ou=private,dc=example,dc=com
```

memberOfEntryScope パラメーターで設定したスコープからユーザーエントリーを移動した場合には、以下を実行します。

- ***member*** などのメンバーシップ属性は、グループエントリーで更新され、ユーザー DN 値が削除されます。
- ***memberOf*** 属性は、グループ DN 値を削除するためにユーザーエントリーで更新されます。

注記

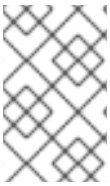
memberOfEntryScopeExcludeSubtree パラメーターに設定した値は、***memberOfEntryScope*** に設定した値よりも高くなります。両方のパラメーターで設定したスコープが重複する場合、**MemberOf** プラグインは、非オーバーラッピングディレクトリーエントリーでのみ機能します。

8.1.4.7. memberOf 値の同期

MemberOf プラグインは、グループエントリー自体の設定に基づいて、グループメンバーエントリーで *memberOf* 属性を自動的に管理します。ただし、*memberOf* 属性がすでに設定されているサーバーに、ユーザーエントリーで *memberOf* 属性を直接編集することも、新しいエントリーをインポートまたは複製できます。このような状況では、サーバープラグインによって管理される *memberOf* 設定と、エントリーに定義された実際のメンバーシップとの間に不整合が生じます。

Directory Server には、プラグインを手動で実行する *memberOf* 修復タスクがあり、適切な *memberOf* 属性がエントリーに設定されていることを確認します。このタスクをトリガーする方法は 3 つあります。

- Directory Server コンソール
- **fixup-memberof.pl** スクリプトの使用
- **cn=memberof task,cn=tasks,cn=config** タスクエントリーの実行



注記

memberOf 再生成タスクは、エントリー自体が複製されていてもローカルで実行されます。つまり、更新されたエントリーが複製されるまで、他のサーバーのエントリーの *memberOf* 属性は更新されません。

8.1.4.7.1. fixup-memberof.pl を使用した memberOf 属性の初期化および再生成

fixup-memberof.pl は、*memberOf* の説明に従って「ldapmodify を使用した memberOf 属性の初期化および再生成」 属性を再生成するために使用される Perl スクリプトラッパーです。

詳細は、**man fixup-memberof.pl** も参照してください。

8.1.4.7.2. ldapmodify を使用した memberOf 属性の初期化および再生成

memberOf 属性を再生成することは、特別なタスク設定エントリーで管理できるタスクの 1 つです。タスクエントリーは、**dse.ldif** ファイルの **cn=tasks** 設定エントリーで発生するため、**ldapmodify** を使用してエントリーを追加してタスクを開始することもできます。タスクが完了するとすぐに、エントリーはディレクトリーから削除されます。

fixup-memberof.pl スクリプトは、*memberOf* 属性を再生成する Directory Server インスタンスに特別なタスクエントリーを作成します。

memberOf 修正タスクを開始するには、**cn=memberof** タスク、**cn= tasks, cn=config** エントリーの下にエントリーを追加します。必要な属性は、特定タスクの *cn* のみです。**ldapmodify** の使用:

```
dn: cn=example memberOf,cn=memberof task,cn=tasks,cn=config
changetype: add
cn:example memberOf
```

タスクが完了するとすぐに、エントリーは **dse.ldif** 設定から削除 されるため、同じタスクエントリーを継続的に再利用できます。

cn=memberof タスク 設定は、設定、『コマンド、およびファイルリファレンス を参照してください』。

8.1.5. 指定したグループへのエントリーの自動追加

- 「Automembership ルールの構造の確認」

- 「Automembership ルールの例」
- 「Automembership 定義の作成」

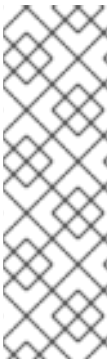
グループ管理は、特に Directory Server データおよび組織を使用するクライアントや、グループを使用してエントリーに機能を適用するクライアントなど、ディレクトリーデータを管理する上で重要な要素となります。グループにより、ディレクトリー全体で一貫して、信頼できるポリシーの適用が容易になります。パスワードポリシー、アクセス制御リスト、その他のルールはすべてグループメンバーシップに基づいて設定できます。

アカウントの作成時に、新しいエントリーをグループに自動的に割り当てることができるため、管理者の介入なしに、適切なポリシーと機能がそれらのエントリーに即座に適用されるようにします。

動的グループは、一致するエントリーがグループに自動的に含まれるため、グループを作成してメンバーを自動的に割り当てる 1 つの方法です。Directory Server のポリシーおよび設定を適用するには、これで十分です。ただし、LDAP アプリケーションとクライアントには、通常、必要な操作を行うためにグループメンバーの静的リストおよび明示的なリストが必要です。静的グループのすべてのメンバーは、このグループに手動で追加する必要があります。

静的グループ自体は動的グループのようなメンバーを検索できませんが、静的グループに自動的にメンバーを追加できるようになります (Auto Membership プラグイン)。

自動メンバーシップにより、基本的に、静的グループが動的グループのように動作できるようにします。異なる自動メンバー定義により、すべての新規ディレクトリーエントリーで自動的に実行される検索が作成されます。自動メンバールールは、動的検索フィルターと同様に、一致するエントリーを検索し、特定します。次に、これらのエントリーをメンバーとして静的グループに追加します。

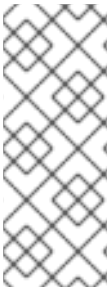


注記

デフォルトでは、**cn=Auto Membership Plugin,cn=plugins,cn=config** エントリーの ***autoMemberProcessModifyOps*** パラメーターは **on** に設定されます。この設定では、Automembership プラグインは、ユーザーエントリーを編集して管理者が別のグループにユーザーを移動する際にグループメンバーシップも更新します。

autoMemberProcessModifyOps を **off** に設定すると、Directory Server は、ユーザーにグループエントリーを追加する場合にのみプラグインを起動し、グループメンバーシップを更新するために手動で修正タスクを実行する必要があります。

Automembership は、ディレクトリーに保存されているオブジェクトタイプ (ユーザー、マシン、ネットワークデバイス、顧客データ、またはその他のアセット) をターゲットにすることができます。



注記

Auto Membership プラグインは、定義された基準に基づいて既存のグループに新しいエントリーを追加します。新しいエントリー用のグループは作成されません。

特定タイプの新規エントリーの作成時に対応するグループエントリーを作成するには、Managed Entries プラグインを使用します。詳細は、「デュアルエントリーの自動作成」を参照してください。

8.1.5.1. Automembership ルールの構造の確認

Auto Membership プラグイン自体は、**cn=plugins,cn=config** のコンテナーエントリーです。グループ割り当ては、子エントリーで定義されます。

8.1.5.1.1. Automembership 設定エントリー

自動メンバー割り当ては、Automembership プラグインエントリーの子であるメインの定義エントリーを使用して作成されます。各定義エントリーは 3 つの要素を定義します。

- 検索スコープと検索フィルターの両方を含むエントリーを識別する LDAP 検索 (*autoMemberScope* および *autoMemberFilter*)
- メンバーエントリーを追加するデフォルトグループ (*autoMemberDefaultGroup*)
- メンバーエントリーの形式: *member* などのグループエントリーの属性、および *dn* などの属性値 (*autoMemberGroupingAttr*)

定義は、automember ルールの基本設定です。必要な情報をすべて識別し、一致するメンバーエントリーとそのメンバーの所属先のグループを特定します。

たとえば、以下の定義は、すべての Windows ユーザーを **cn=windows-users** グループに割り当てます。

```
dn: cn=Windows Users,cn=Auto Membership Plugin,cn=plugins,cn=config
objectclass: autoMemberDefinition
autoMemberScope: ou=People,dc=example,dc=com
autoMemberFilter: objectclass=ntUser
autoMemberDefaultGroup: cn=windows-group,cn=groups,dc=example,dc=com
autoMemberGroupingAttr: member:dn
```

この例に使用される属性と、このエントリーに設定できるその他の属性の詳細は、Red『Hat Directory Server Configuration , Command, and File Reference』の cn=Auto Membership Plugin,cn=plugins,cn=config エントリーの説明を参照してください。

8.1.5.1.2. 追加の正規表現エントリー

ユーザーグループのように、一致するすべてのエントリーをメンバーとして追加する必要がある場合は、単純な定義で十分です。ただし、他の属性の値によっては、LDAP 検索フィルターに一致するエントリーを異なるグループに追加する必要がある場合があります。たとえば、IP アドレスや物理的な場所に応じて、異なるグループにマシンを追加しないといけない場合があります。ユーザーは、従業員 ID 番号に応じて異なるグループに置かなければならない場合があります。

automember 定義では、正規表現を使用して、グループから含めたり、除外したりするエントリーに追加の条件を提供したり、選択したエントリーを追加する新しい特定のグループを作成したりできます。

たとえば、automember 定義は、汎用ホストグループに追加されるすべてのマシンを設定します。

例8.2 ホストグループの automember 定義

```
dn: cn=Hostgroups,cn=Auto Membership Plugin,cn=plugins,cn=config
objectclass: autoMemberDefinition
cn: Hostgroups
autoMemberScope: dc=example,dc=com
autoMemberFilter: objectclass=ipHost
autoMemberDefaultGroup: cn=systems,cn=hostgroups,dc=example,dc=com
autoMemberGroupingAttr: member:dn
```

指定の範囲内に完全修飾ドメイン名を持つマシンが Web サーバーグループに追加されるように、正規表現ルールが追加されます。

例8.3 Web サーバーグループの正規表現条件

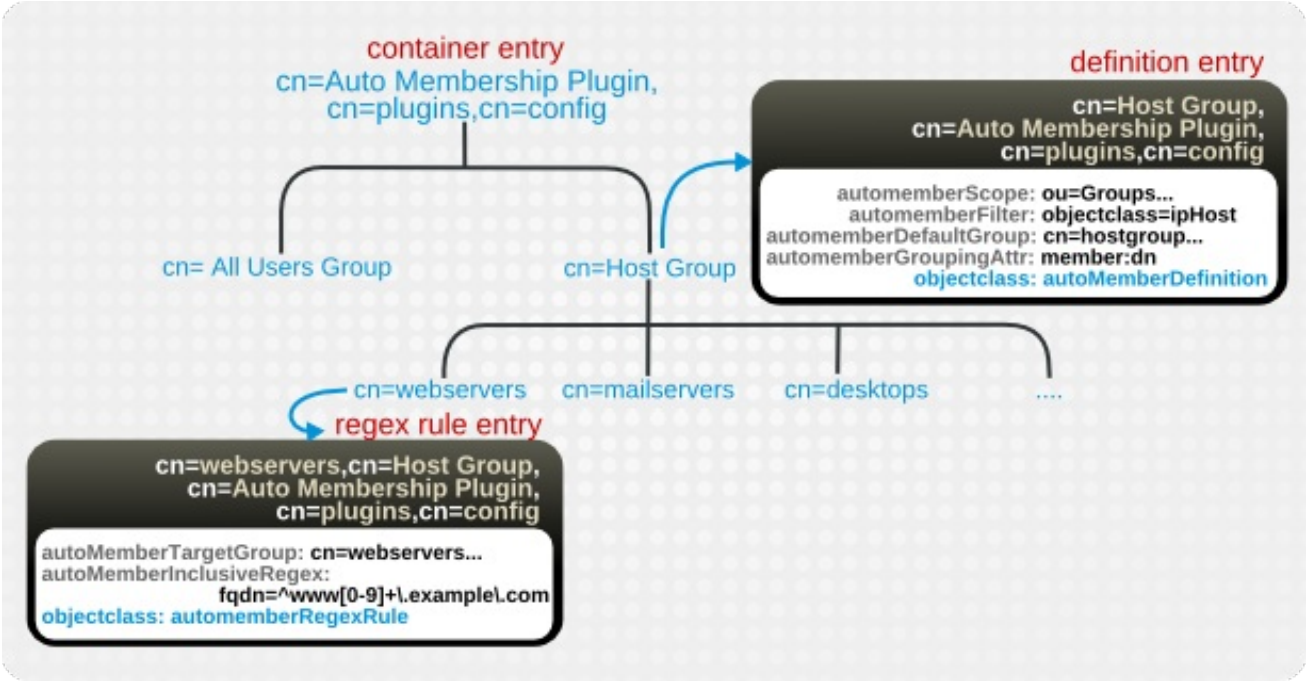
```
dn: cn=webservers,cn=Hostgroups,cn=Auto Membership Plugin,cn=plugins,cn=config
objectclass: autoMemberRegexRule
description: Group for webservers
cn: webservers
autoMemberTargetGroup: cn=webservers,cn=hostgroups,dc=example,dc=com
autoMemberInclusiveRegex: fqdn=^www\.web[0-9]+\.example\.com
```

そのため、www.web1.example.com などの式 ^www\.web[0-9]+\.example\.com に一致する完全修飾ドメイン名で追加されたホストマシンは、正確な正規表現に定義される cn=webservers グループに追加されます。LDAP フィルター objectclass=ipHost に一致するが別のタイプの完全修飾ドメイン名を持つその他のマシンエントリーは、メインの定義エントリーで定義される一般的なホストグループ cn=systems に追加されます。

したがって、定義内のグループは、一般的な定義に一致するが、正規表現ルールの条件を満たさないエントリーのフォールバックです。

正規表現ルールは、自動メンバー定義の子エントリーです。

図8.1 正規表現の条件



各ルールには、複数の包含および除外の式を含めることができます。(除外は最初に評価されます。) エントリーが包含ルールと一致する場合は、グループに追加されます。

正規表現ルールに指定できるターゲットグループは 1 つだけです。

表8.3 正規表現の条件属性

属性	説明
autoMemberRegexRule (必須オブジェクトクラス)	正規表現ルールとしてエントリーを識別します。このエントリーは、automember 定義 (**objectclass: autoMemberDefinition**) の子である必要があります。
autoMemberInclusiveRegex	含めるエントリーを識別するために使用する正規表現を設定します。一致するエントリーのみがグループに追加されます。複数の正規表現を使用できます。エントリーがこれらの式のいずれかと一致する場合は、グループに含まれます。 式の形式は、Perl と互換性のある正規表現 (PCRE) です。PCRE パターンの詳細は、**pcresyntax**(3) の man ページを参照してください。 これは多値属性です。
autoMemberExclusiveRegex	除外するエントリーを識別するために使用する正規表現を設定します。エントリーが除外条件と一致する場合は、グループに 含まれません。複数の正規表現を使用できます。エントリーがこれらの式のいずれかと一致する場合は、グループで除外されます。 式の形式は、Perl と互換性のある正規表現 (PCRE) です。PCRE パターンの詳細は、**pcresyntax**(3) の man ページを参照してください。 これは多値属性です。  注記 除外条件は最初に評価され、包含条件よりも優先されます。
autoMemberTargetGroup	正規表現の条件を満たす場合に、エントリーをメンバーとして追加するグループを設定します。

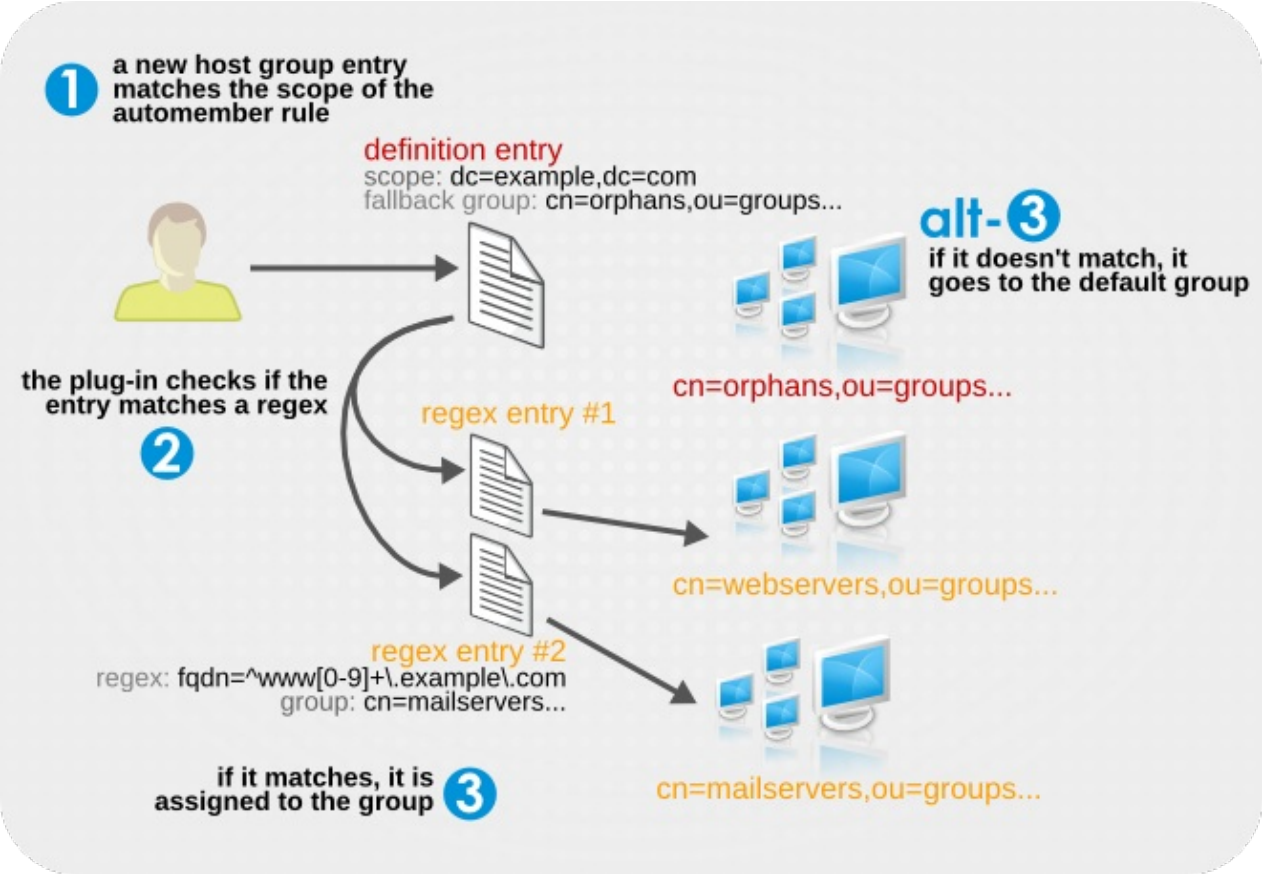
8.1.5.2. Automembership ルールの例

Automembership ルールは通常、ユーザーとマシンに適用されます (ただし、どのタイプのエントリーにも適用することができます)。自動メンバーシップルールの計画に役立つ便利な例がいくつかあります。

- IP アドレスに基づく異なるホストグループ
- Windows ユーザーグループ
- 従業員の ID に基づく異なるユーザーグループ

> 例8.4 IP アドレス別のホストグループ
>
> automember ルールは、まずルールのスコープとターゲットを定義します。「追加の正規表現エントリー」の例では、設定グループを使用してフォールバックグループと正規表現エントリーを定義し、一致するエントリーをソートします。

スコープは、全 ホストエントリーの検索に使用されます。その後、プラグインは正規表現エントリーで繰り返し処理します。エントリーが包含する正規表現と一致する場合は、そのホストグループに追加されます。グループに一致しない場合は、デフォルトグループに追加されます。



実際のプラグイン設定エントリーは、定義エントリーに、ホストを Web サーバーグループまたはメールサーバーグループにフィルターを設定する 2 つの正規表現エントリーに対して設定されます。

configuration entry
dn: cn=Hostgroups,cn=Auto Membership Plugin,cn=plugins,cn=config
objectclass: autoMemberDefinition
cn: Hostgroups
autoMemberScope: dc=example,dc=com
autoMemberFilter: objectclass=bootableDevice
autoMemberDefaultGroup: cn=orphans,cn=hostgroups,dc=example,dc=com
autoMemberGroupingAttr: member:dn

regex entry #1
dn: cn=webservers,cn=Hostgroups,cn=Auto Membership Plugin,cn=plugins,cn=config
objectclass: autoMemberRegexRule
description: Group placement for webservers
cn: webservers
autoMemberTargetGroup: cn=webservers,cn=hostgroups,dc=example,dc=com
autoMemberInclusiveRegex: fqdn=^www[0-9]+\.example\.com
autoMemberInclusiveRegex: fqdn=^web[0-9]+\.example\.com
autoMemberExclusiveRegex: fqdn=^www13\.example\.com
autoMemberExclusiveRegex: fqdn=^web13\.example\.com

regex entry #2
dn: cn=mailservers,cn=Hostgroups,cn=Auto Membership Plugin,cn=plugins,cn=config
objectclass: autoMemberRegexRule

```
description: Group placement for mailservers
cn: mailservers
autoMemberTargetGroup: cn=mailservers,cn=hostgroups,dc=example,dc=com
autoMemberInclusiveRegex: fqdn=^mail[0-9]+\.example\.com
autoMemberInclusiveRegex: fqdn=^smtp[0-9]+\.example\.com
autoMemberExclusiveRegex: fqdn=^mail13\.example\.com
autoMemberExclusiveRegex: fqdn=^smtp13\.example\.com
```

例8.5 Windows ユーザーグループ

「Automembership 設定エントリー」に表示されている基本的なユーザーグループは、*posixAccount* 属性を使用して新規ユーザーをすべて識別します。Directory Server 内に作成された新規ユーザーはすべて、*posixAccount* 属性を使用して作成されます。したがって、新しい Directory Server ユーザーにとって安全なキャッチオールです。ただし、ユーザーアカウントを Windows ドメインから Directory Server に同期すると、Windows ユーザーアカウントは *posixAccount* 属性 なし で作成されます。

Windows ユーザーは *ntUser* 属性で識別されます。基本的な all-users グループルールは、特に目的の Windows ユーザーに変更できます。これは、デフォルトの all-users グループまたは Windows 固有のグループに追加できます。

```
dn: cn=Windows Users,cn=Auto Membership Plugin,cn=plugins,cn=config
objectclass: autoMemberDefinition
autoMemberScope: dc=example,dc=com
autoMemberFilter: objectclass=ntUser
autoMemberDefaultGroup: cn=Windows Users,cn=groups,dc=example,dc=com
autoMemberGroupingAttr: member:dn
```

例8.6 従業員タイプによるユーザーグループ

Auto Membership プラグインはカスタム属性で機能します。これは、他のアプリケーションが管理するエントリーに役立ちます。たとえば、人的リソースアプリケーションは、カスタムの *employeeType* 属性で従業員タイプをもとにユーザーを作成してから参照できます。

例8.4「IP アドレス別のホストグループ」と同様、ユーザータイプのルールは 2 つの正規表現フィルターを使用して、すべての時間および一時従業員をソートます。この例では、実際の正規表現ではなく明示的な値を使用します。その他の属性については、従業員の ID 番号範囲に基づいてフィルターを作成するなど、正規表現を使用することが推奨されます。

```
configuration entry
dn: cn=Employee groups,cn=Auto Membership Plugin,cn=plugins,cn=config
objectclass: autoMemberDefinition
cn: Hostgroups
autoMemberScope: ou=employees,ou=people,dc=example,dc=com
autoMemberFilter: objectclass=inetorgperson
autoMemberDefaultGroup: cn=general,cn=employee
groups,ou=groups,dc=example,dc=com
autoMemberGroupingAttr: member:dn

regex entry #1
dn: cn=full time,cn=Employee groups,cn=Auto Membership Plugin,cn=plugins,cn=config
objectclass: autoMemberRegexRule
```

```
description: Group for full time employees
cn: full time
autoMemberTargetGroup: cn=full time,cn=employee
groups,ou=groups,dc=example,dc=com
autoMemberInclusiveRegex: employeeType=full

regex entry #2
dn: cn=temporary,cn=Employee groups,cn=Auto Membership Plugin,cn=plugins,cn=config
objectclass: autoMemberRegexRule
description: Group placement for interns, contractors, and seasonal employees
cn: temporary
autoMemberTargetGroup: cn=temporary,cn=employee
groups,ou=groups,dc=example,dc=com
autoMemberInclusiveRegex: employeeType=intern
autoMemberInclusiveRegex: employeeType=contractor
autoMemberInclusiveRegex: employeeType=seasonal
```

8.1.5.3. Automembership 定義の作成

1. 必要に応じて、Auto Membership プラグインを有効にします。**ldapmodify** の使用:

   ```
   dn: cn=Auto Membership Plugin,cn=plugins,cn=config
   changetype: replace
   replace: nsslapd-pluginEnabled
   nsslapd-pluginEnabled: on
   ```

2. **cn=Auto Membership Plugin,cn=plugins,cn=config** コンテナーエントリーの下に、新しいプラグインインスタンスを作成します。このエントリーは、autoMember **Definition** オブジェクトクラスに属している必要があります。**ldapmodify** の使用:

   ```
   dn: cn=Example Automember Definition,cn=Auto Membership
   Plugin,cn=plugins,cn=config
   objectclass: autoMemberDefinition
   ...
   ```

   定義に必要な属性は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス』 に記載されています。

3. 定義のスコープおよびフィルターを設定します。これは、一致するエントリーの初期検索に使用されます。

   たとえば、ou=People サブツリーに追加され、*ntUser* 属性が含まれる新しいエントリーの場合：

   ```
   autoMemberScope: ou=People,dc=example,dc=com
   autoMemberFilter: objectclass=ntUser
   ```

4. （デフォルトまたはフォールバックグループとして）一致するエントリーを追加するグループと、そのグループタイプのメンバーエントリーの形式を設定します。

   ```
   autoMemberDefaultGroup: cn=windows-group,cn=groups,dc=example,dc=com
   autoMemberGroupingAttr: member:dn
   ```

5. オプション。包含または排他的な正規表現フィルターを作成し、これらのフィルターに一致するエントリーに使用するグループを設定します。

   正規表現条件の属性は、表8.3「正規表現の条件属性」 に記載されています。

   正規表現の条件は、automember 定義の子として追加されます。これらの条件は **autoMemberRegexRule** オブジェクトクラスに属している必要があります。

   ldapmodify の使用:

   ```
   dn: cn=Example Regex,cn=Example Automember Definition,cn=Auto Membership
   Plugin,cn=plugins,cn=config
   objectclass: autoMemberRegexRule
   ...
   ```

   次に、ターゲットグループ名と包含的または排他的な正規表現を追加します。include および exclude 条件の両方を使用でき、両方のタイプの式を複数使用できます。

   ```
   autoMemberTargetGroup: cn=windows-admin-group,cn=groups,dc=example,dc=com
   autoMemberInclusiveRegex: cn=\.* Administrator \*
   ```

   新規エントリーが正規表現条件と一致する場合は、automember 定義に設定されたデフォルトグループ の代わりに、そのグループに追加されます。

6. Directory Server が動的プラグインを有効にするために設定されていない場合は、サーバーを再起動して変更した新しいプラグインインスタンスを読み込みます。

8.1.5.4. 自動メンバー定義の既存のエントリーの更新

Auto Member プラグインは、新規エントリーがディレクトリーに追加される場合にのみ実行されます。プラグインは、automembership ルールに一致するように編集される既存のエントリーまたはエントリーを無視します。

自動メンバールールに対して既存のエントリーをチェックし、それに応じてグループメンバーシップを更新するために実行できるディレクトリータスク操作があります。このタスク(**cn=automember rebuild membership**)では、LDAP 検索パラメーターに基づいて、処理する既存のエントリーを特定するには、3 つの要素を実行する必要があります。

- 検索フィルター
- 検索範囲
- 検索を開始するベース DN

特定のタスク実行にも名前が必要です。

タスクエントリーは **ldapmodify** を使用して作成できます。タスクが完了すると、エントリーが自動的に削除されます。以下に例を示します。

```
dn: cn=my rebuild task, cn=automember rebuild membership,cn=tasks,cn=config
objectClass: top
objectClass: extensibleObject
cn: my rebuild task
```

```
basedn: dc=example,dc=com
filter: (uid=*)
scope: sub
```

8.1.5.5. 自動メンバー定義のテスト

Auto Member プラグインの各インスタンスは、定義と正規表現に関連してはいますが別々のエントリーのセットであるため、ユーザーがグループにどのようにマップされるかを正確に確認するのは難しい場合があります。これは、ユーザーの異なるサブセットをターゲットとする複数のルールがあると、より困難になります。

ドライランタスクが 2 つあり、すべての Auto Member プラグイン定義が設計通りにグループを適切に割り当てているかどうかを判断するのに役立ちます。

既存のエントリーを使用したテスト

cn=automember export updates が、ディレクトリー内の 既存のエントリー に対して実行し、ルールに基づいてユーザーの追加結果をエクスポートします。これは、既存のルールをテストして、実際のデプロイメントの実行方法を確認するのに役立ちます。

このタスクには、**cn=automember rebuild membership** タスク (検索、検索フィルター、および検索スコープのベース DN) と同じ情報が必要です。また、推奨されるエントリーの更新を記録するエクスポート LDIF ファイルを指定する追加パラメーターがあります。

ldapmodify の使用:

```
dn: cn=test export, cn=automember export updates,cn=tasks,cn=config
objectClass: top
objectClass: extensibleObject
cn: test export
basedn: dc=example,dc=com
filter: (uid=*)
scope: sub
ldif: /tmp/automember-updates.ldif
```

Import LDIF でのテスト

cn=automember map updates タスクは、新規ユーザーのインポート LDIF を取得してから、現在の自動メンバールールに対して新規ユーザーを実行します。これは、(実際の) 新規または既存のユーザーエントリーに適用する前に、新しいルールをテストする場合に非常に役立ちます。

これは、提案された新しいエントリーの変更を、既存のルールにマッピングまたは関連付けるため、マップタスクと呼ばれます。

このタスクには、入力 LDIF (少なくとも一部のユーザーエントリーを含む) の場所と、提案されたエントリー更新を書き込む出力 LDIF ファイルの 2 つの属性のみが必要です。入力および出力の LDIF ファイルの両方がローカルマシンの絶対パスです。

たとえば、**ldapmodify** を使用するには、以下を実行します。

```
dn: cn=test mapping, cn=automember map updates,cn=tasks,cn=config
objectClass: top
objectClass: extensibleObject
cn: test mapping
ldif_in: /tmp/entries.ldif
ldif_out: /tmp/automember-updates.ldif
```

8.2. ロールの使用

ロールは、前述のセクションで説明されている静的および動的グループを統一するエントリーグループ化メカニズムです。ロールは、アプリケーションにより効率的で使いやすいように設計されています。たとえば、アプリケーションはグループを選択し、複数のグループのメンバー一覧を参照するのではなく、エントリー自体をクエリーしてメンバーであるロールの一覧を取得できます。

8.2.1. ロールの概要

Red Hat には 2 種類のグループがあります。静的グループ には、有限で、定義されたメンバーの一覧があります。動的グループ は、フィルターを使用してどのエントリーがグループのメンバーかを認識するため、グループメンバーシップはグループフィルターの変更に一致するエントリーとして常に変更されます。(両方の種類グループが、「グループの使用」で説明されています。)

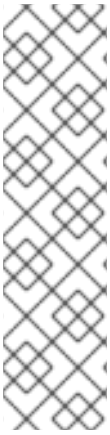
ロール はハイブリッドグループで、静的グループと動的グループの両方として機能します。グループを使用すると、エントリーはメンバーとしてグループエントリーに追加されます。ロールを使用すると、role 属性がエントリーに追加され、その属性はロールエントリー内のメンバーを自動的に識別するために使用されます。

ロール メンバー は、ロールを持つエントリーです。メンバーは、明示的に、または動的に指定できます。ロールのメンバーシップの指定方法は、ロールのタイプによって異なります。Directory Server は、以下の 3 種類のロールをサポートします。

- 管理対象ロール には、メンバーの明示的な列挙リストがあります。
- フィルターされたロール には、LDAP フィルターで指定される各エントリーに含まれる属性に応じて、エントリーがロールに割り当てられます。フィルターに一致するエントリーはロールを持ちます。
- ネストされたロール は、他のロールが含まれるロールです。

管理対象ロールは、通常、静的グループで実行可能なものをすべて実行できます。ロールメンバーは、動的グループによるフィルタリングと同様に、フィルターされたロールを使用してフィルタリングできます。ロールはグループよりも使いやすく、実装に柔軟性が高まり、クライアントの複雑さが軽減されます。

ロールの作成時には、ユーザーがロールから追加できるか、またはロールから削除できるかどうかを判断します。ロールおよびアクセス制御の詳細は、「セキュアなロールの使用」を参照してください。



注記

サーバーがクライアントアプリケーションに対して機能するため、Directory Server ではロールの評価がグループを評価するよりもリソース集約されます。ロールを使用すると、クライアントアプリケーションは *nsRole* 属性を検索してロールのメンバーシップを確認することができます。*nsRole* 属性は、エントリーが属するロールを識別する計算属性です。*nsRole* 属性はエントリー自体に保存されません。クライアントアプリケーションの観点からは、メンバーシップを確認する方法は統一されており、サーバー側で実行されます。

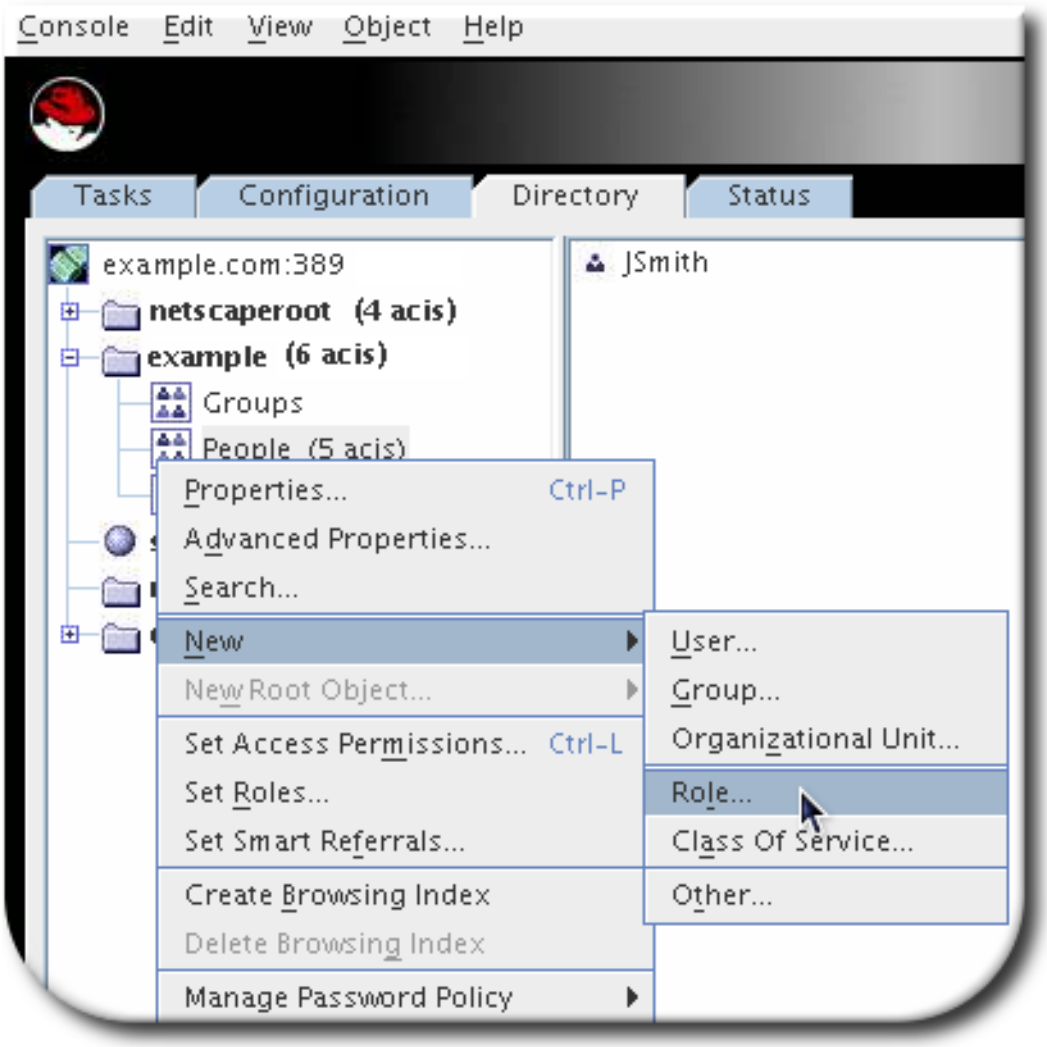
ロールの使用に関する考慮事項は、『Red Hat Directory Server デプロイメントガイド』 で説明しています。

8.2.2. 管理ロールの作成

管理対象ロールには、メンバーの明示的な列挙リストがあります。エントリーに *nsRoleDN* 属性を追加して、管理ロールがエントリーに追加されます。

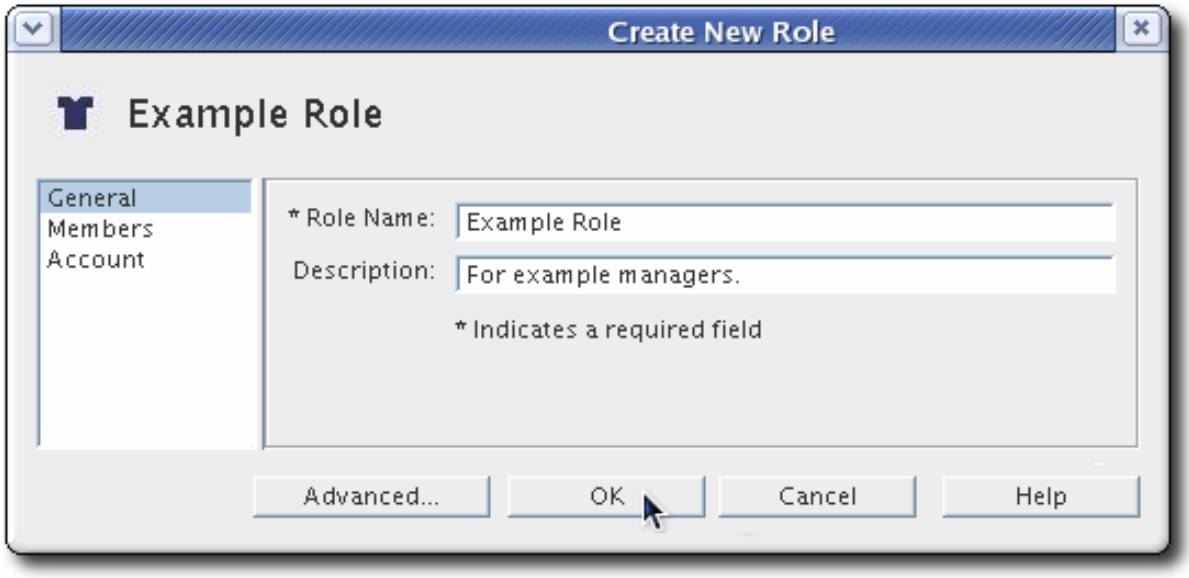
8.2.2.1. コンソールでの管理ロールの作成

1. Directory Server コンソールで、Directory タブを選択します。
2. 左側のナビゲーションペインでツリーを参照し、新規ロールの親エントリーを選択します。
3. オブジェクト メニューに移動し 、 **New > Role** を選択します。

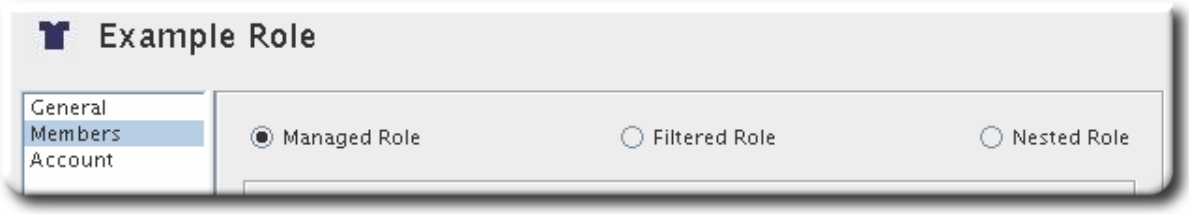


または、エントリーを右クリックし、**New > Role** を選択します。

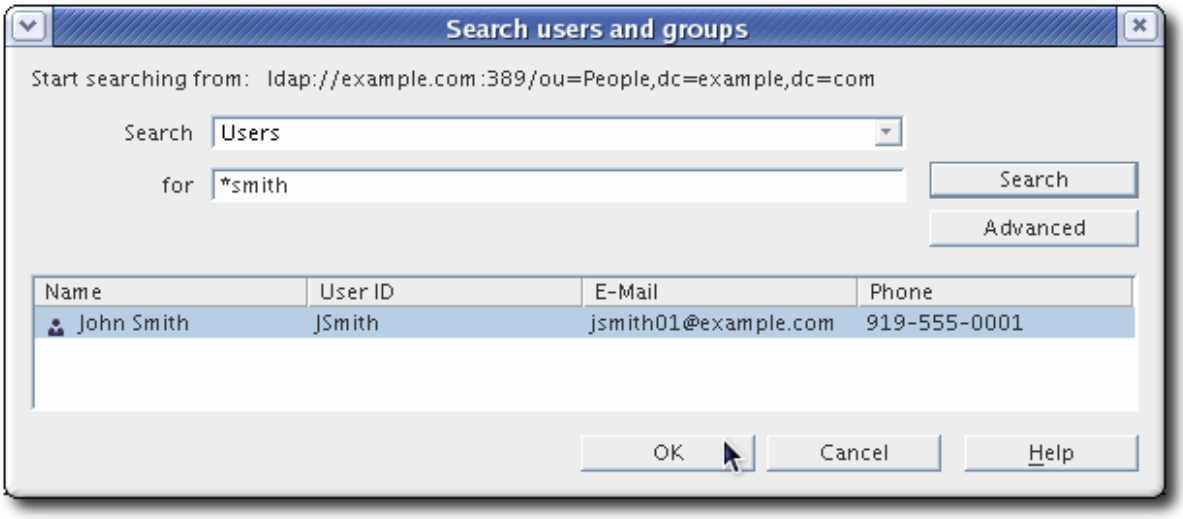
4. 左側のペインで **General** をクリックします。**Role Name** フィールドに新規ロールの名前を入力します。ロール名が必要です。



5. **Description** フィールドに、新規ロールの説明を入力します。
6. 左側のペインで **Members** をクリックします。
7. 右側のペインで、Managed Role を選択します。**Add** をクリックして、新しいエントリーをメンバーの一覧に追加します。



8. **Search** ドロップダウンリストで、**Search** ドロップダウンリストから **Users** を選択します。返されたエントリーのいずれかを選択し、**OK** をクリックします。



9. エントリーを追加したら、**OK** をクリックします。

8.2.2.2. コマンドラインでの管理ロールの作成

ロールは、ITU X.509 標準で定義されている ldapsubentry オブジェクトクラスから継承されます。さらに、各管理ロールには、nsRoleDefinition オブジェクトクラスから継承する 2 つのオブジェクトクラスが必要です。

- nsSimpleRoleDefinition
- nsManagedRoleDefinition

管理ロールでは、任意の *description* 属性も許可されます。

管理ロールのメンバーは、エントリーに *nsRoleDN* 属性を持ちます。

この例では、マーケティング部門に割り当てることのできるロールを作成します。

1. -a オプションで ldapmodify を使用して、管理ロールエントリーを追加します。新しいエントリーには、nsManagedRoleDefinition オブジェクトクラスが含まれ、その後に LdapSubEntry、nsRoleDefinition、および nsSimpleRoleDefinition のオブジェクトクラスを継承します。

```
dn: cn=Marketing,ou=people,dc=example,dc=com
objectclass: top
objectclass: LdapSubEntry
objectclass: nsRoleDefinition
objectclass: nsSimpleRoleDefinition
objectclass: nsManagedRoleDefinition
cn: Marketing
description: managed role for marketing staff
```

2. ldapmodify を使用して、ロールをマーケティングスタッフメンバーに 1 つずつ割り当てます。

```
dn: cn=Bob,ou=people,dc=example,dc=com
changetype: modify
add: nsRoleDN
nsRoleDN: cn=Marketing,ou=people,dc=example,dc=com
```

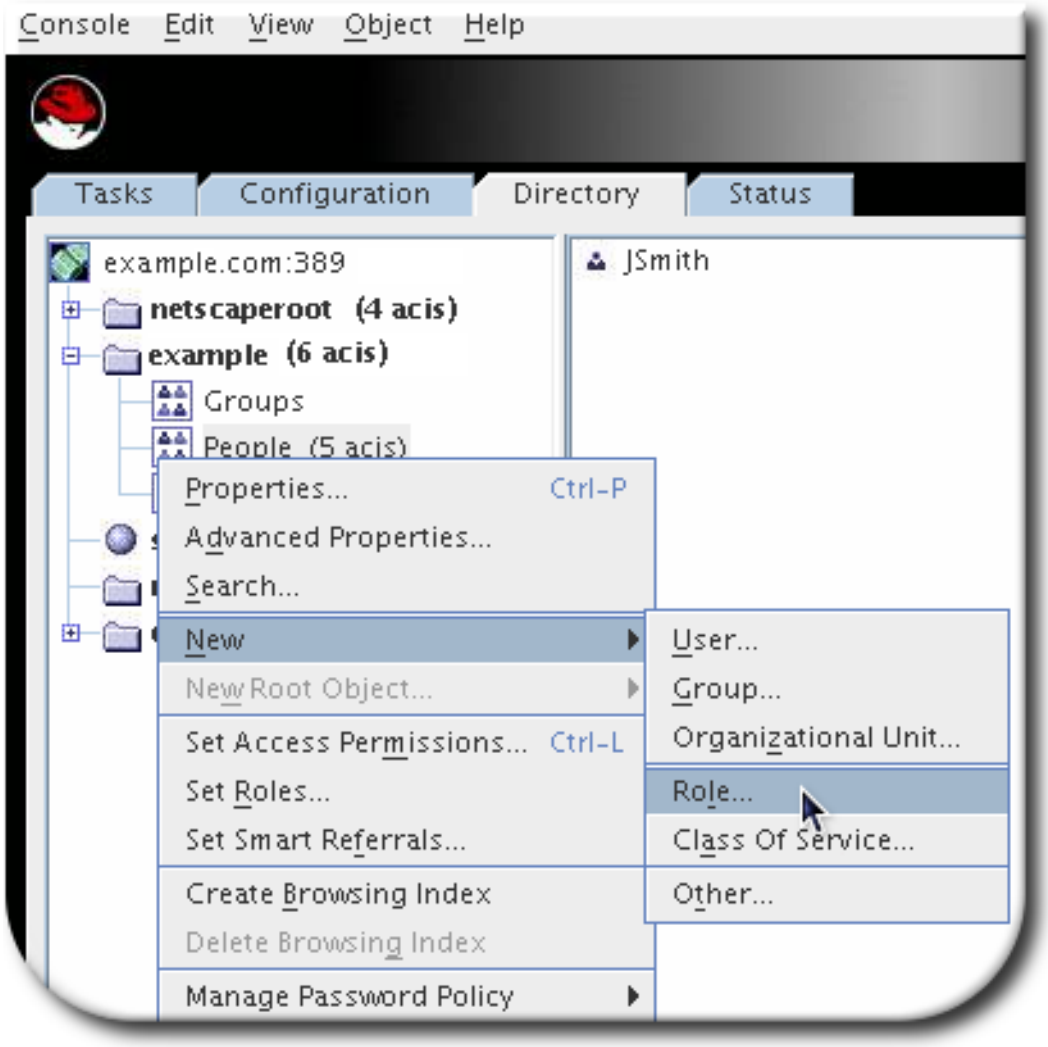
エントリーの *nsRoleDN* 属性は、エントリーが管理ロール cn=Marketing,ou=people,dc=example,dc=com のメンバーであることを示します。

8.2.3. フィルター設定されたロールの作成

エントリーは、エントリーがロールに定義されている特定の属性を持つかどうかに応じて、フィルターされたロールに割り当てられます。ロール定義は、ターゲット属性の LDAP フィルターを指定します。フィルターに一致するエントリーは、ロールを持ちます (ロールのメンバーです)。

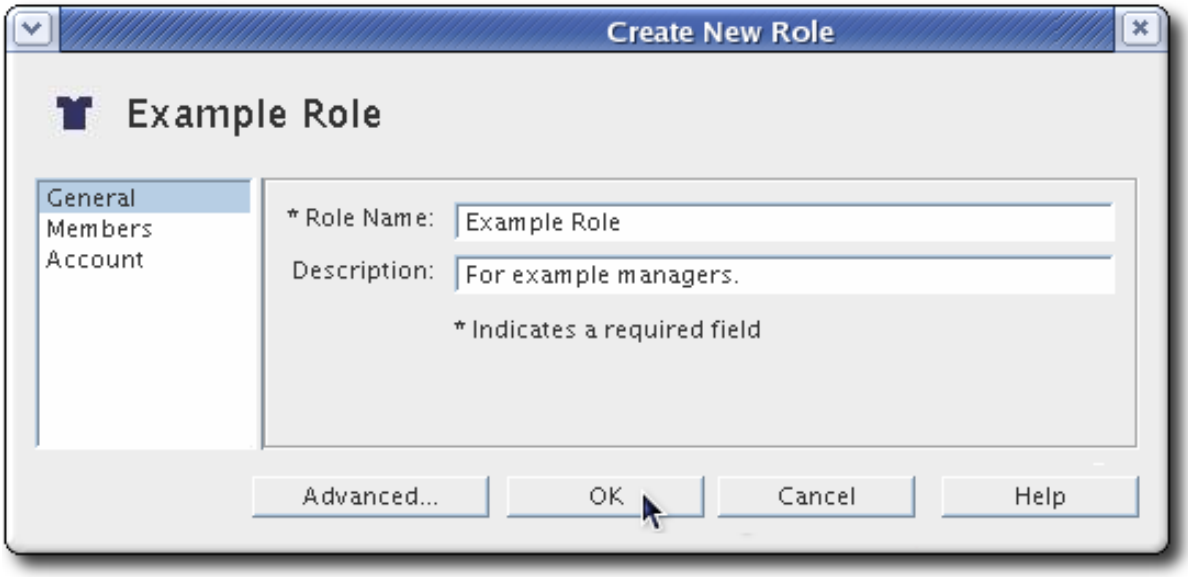
8.2.3.1. コンソールでのフィルターロールの作成

1. Directory Server コンソールで、Directory タブを選択します。
2. 左側のナビゲーションペインでツリーを参照し、新規ロールの親エントリーを選択します。
3. オブジェクト メニューに移動し 、 New > Role を選択します。



または、エントリーを右クリックし、**New > Role** を選択します。

4. 左側のペインで **General** をクリックします。**Role Name** フィールドに新規ロールの名前を入力します。ロール名が必要です。

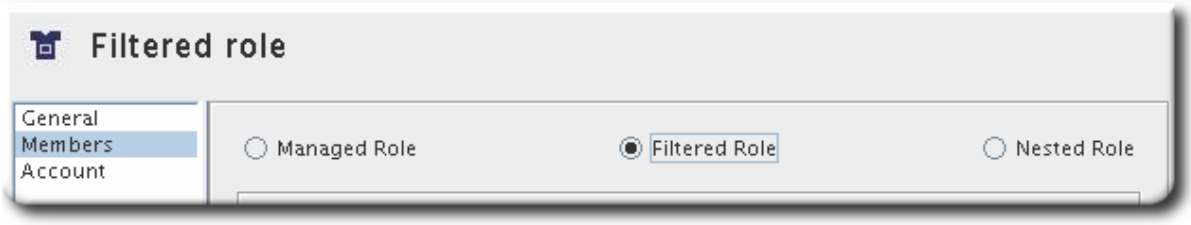


5. **Description** フィールドに、新規ロールの説明を入力します。

6. 左側のペインで **Members** をクリックします。

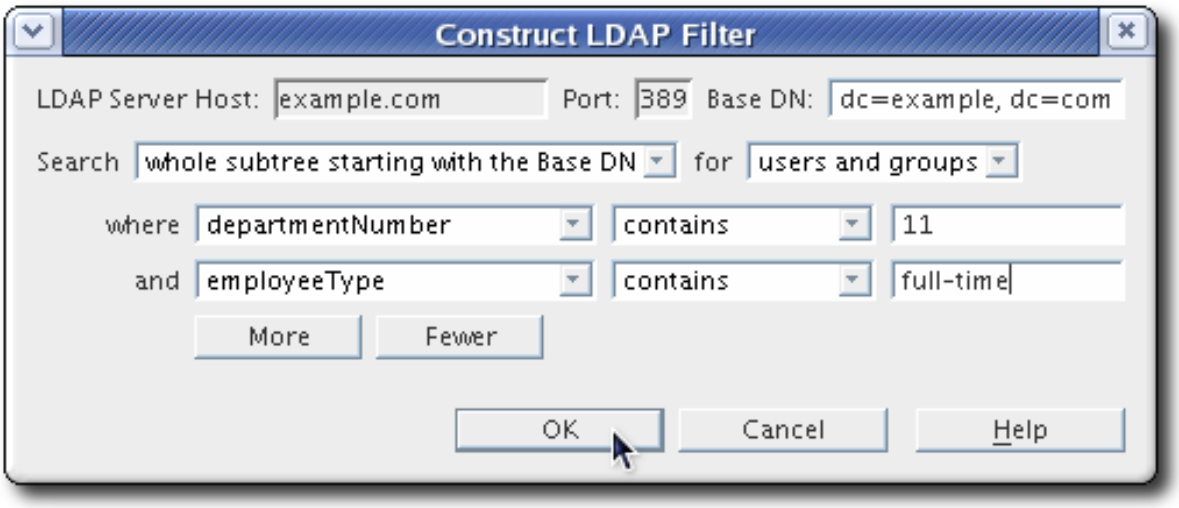
検索ダイアログボックスが簡単に表示されます。

7. 右側のペインで、Filtered **Role** を選択します。



8. テキストフィールドに LDAP フィルターを入力するか、または LDAP フィルター の構成 を介して、Construct to be struct をクリックします。

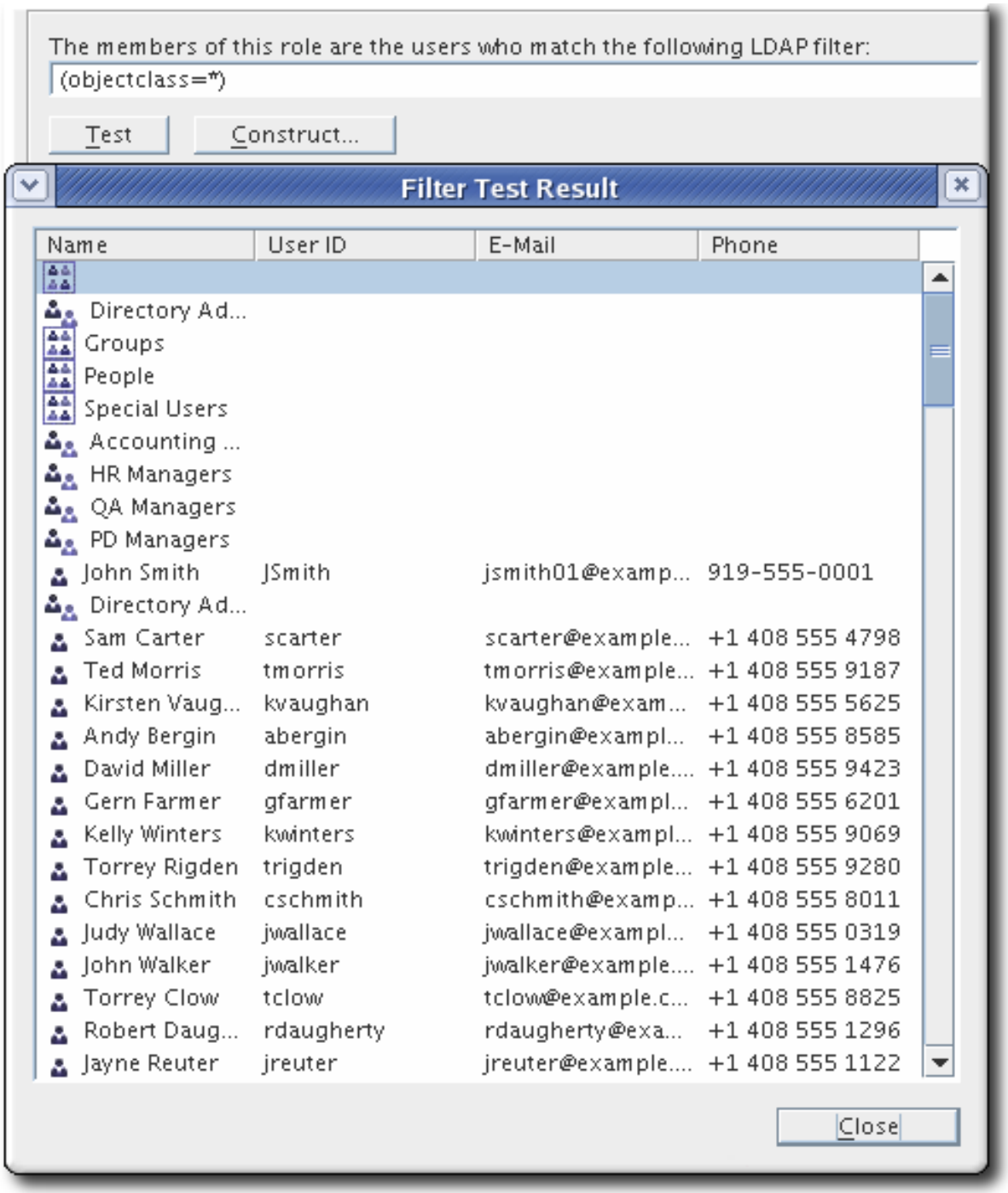
コンストラクト は標準の LDAP URL 構成ダイアログを開きます。**LDAP Server Host**、**Port**、**Base DN**、および **Search** のフィールドを無視します（検索スコープはフィルターされたロール定義を設定できません）。



- For ドロップダウンリストから、フィルターするエントリーの種類 を選択します。エントリーは、ユーザー、グループ、またはその両方になります。

- **Where** ドロップダウンリストから属性を選択します。2 つのフィールドに続く 2 つのフィールドは、ドロップダウンリストから修飾子(include)、not、 または **not** のいずれかを選択して検索を改良して検索を改良します。テキストボックスに属性値を入力します。その他のフィルターを追加するには、詳細 をクリックします。不要なフィルターを削除するには、F **ewer** をクリックします。

9. **Test** をクリックしてフィルターを試行します。



10. **OK** をクリックします。

8.2.3.2. コマンドラインでフィルターされたロールの作成

ロールは、ITU X.509 標準で定義されている **ldapsubentry** オブジェクトクラスから継承されます。さらに、フィルターが設定された各ロールには、**nsRoleDefinition** オブジェクトクラスから継承される 2 つのオブジェクトクラスが必要です。

- nsComplexRoleDefinition
- nsFilteredRoleDefinition

フィルターが設定されたロールエントリーには、ロールメンバーを判断するために LDAP フィルターを定義する ***nsRoleFilter*** 属性も必要です。任意で、ロールは ***description*** 属性を取ることができます。

フィルターが設定されたロールのメンバーは、*nsRoleFilter* 属性で指定されたフィルタに一致するエントリーです。

この例では、すべての営業マネージャーに適用される、フィルターが設定されたロールを作成します。

1. **-a** オプションを指定して **ldapmodify** を実行して、新規エントリーを追加します。
2. フィルターされたロールエントリーを作成します。

   ロールエントリーには **nsFilteredRoleDefinition** オブジェクトクラスがあり、これは、オブジェクトクラス **LdapSubEntry**、**nsRoleDefinition**、および **nsComplexRoleDefinition** から継承されます。

   nsRoleFilter 属性は、**sales managers** の値が含まれる *o* (組織) 属性にフィルターを設定します。

```
dn: cn=SalesManagerFilter,ou=people,dc=example,dc=com
changetype: add
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: SalesManagerFilter
nsRoleFilter: o=sales managers
Description: filtered role for sales managers
```

以下のエントリーはフィルター (**sales managers** 値を持つ *o* 属性) と一致するため、このフィルターが自動的に設定されたロールのメンバーになります)。

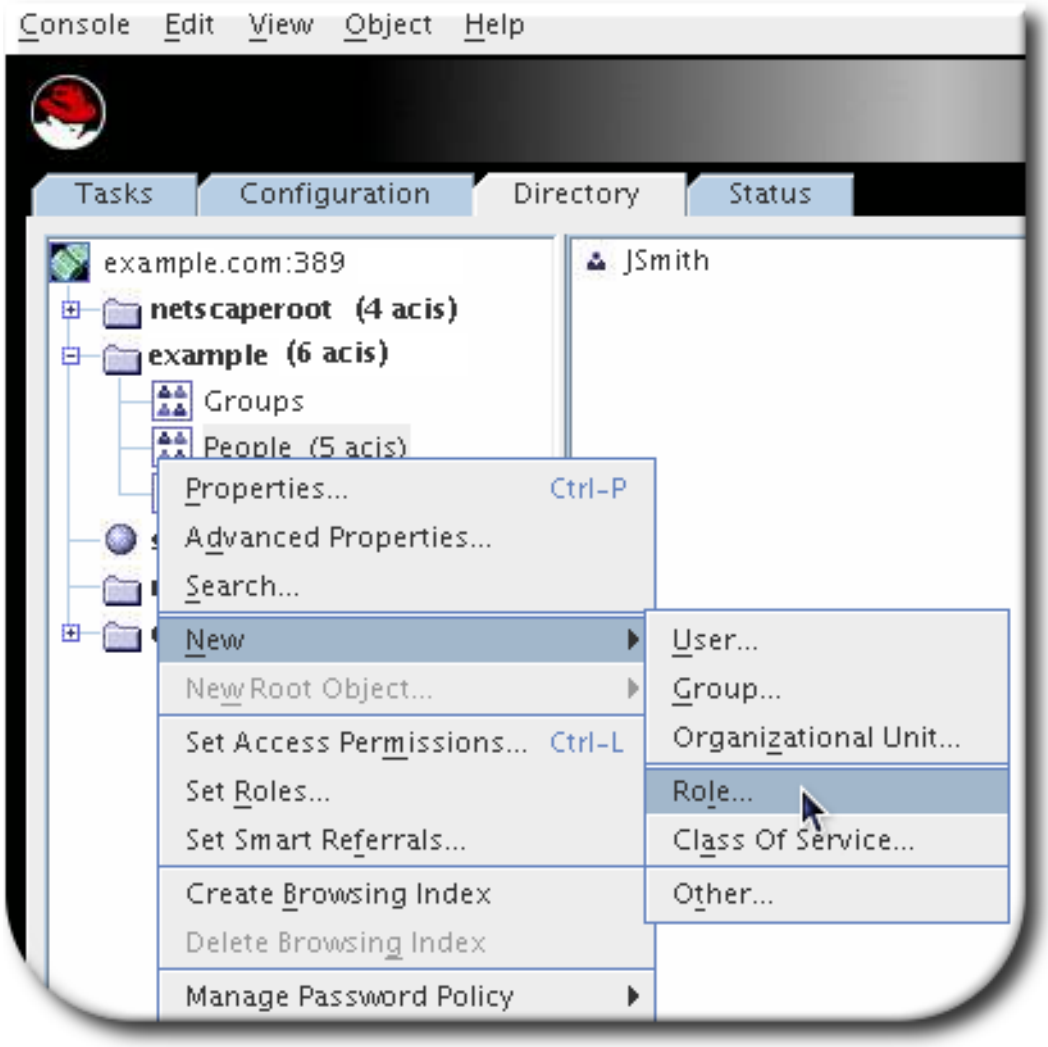
```
dn: cn=Pat Smith,ou=people,dc=example,dc=com
objectclass: person
cn: Pat
sn: Smith
userPassword: secret
o: sales managers
```

8.2.4. ネスト化されたロールの作成

ネストされたロールは、他のロールが含まれるロールです。ネストされたロールを作成する前に、別のロールが存在している必要があります。ネストされたロールが作成されると、コンソールにはネスト化可能なロールの一覧が表示されます。ネストされたロール内でネストされたロールは、*nsRoleDN* 属性を使用して指定します。

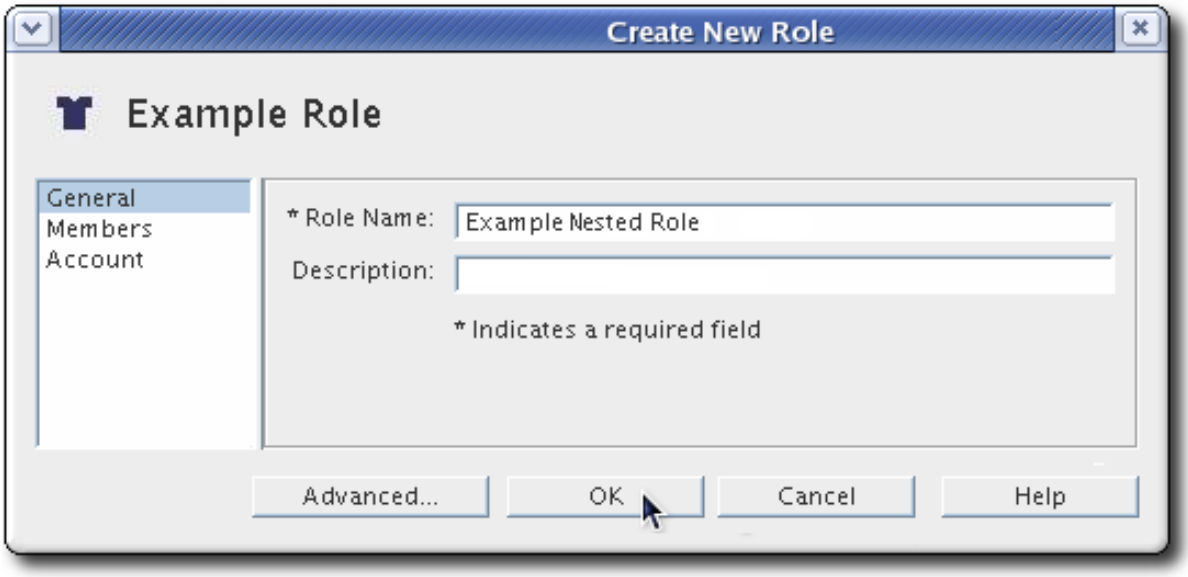
8.2.4.1. コンソールでのネスト化されたロールの作成

1. Directory Server コンソールで、Directory タブを選択します。
2. 左側のナビゲーションペインでツリーを参照し、新規ロールの親エントリーを選択します。
3. オブジェクト メニューに移動し 、 **New > Role** を選択します。



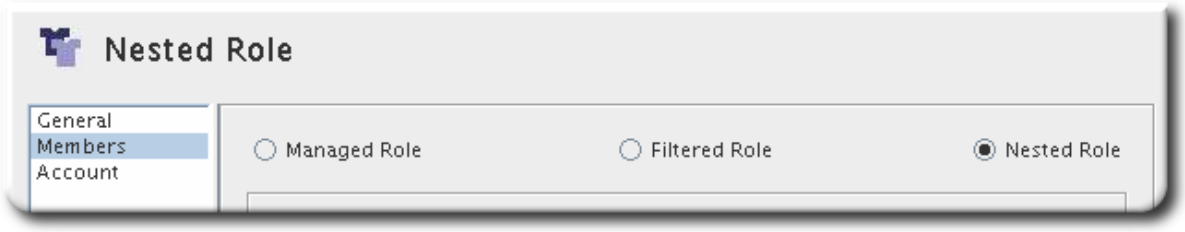
または、エントリーを右クリックし、**New > Role** を選択します。

4. 左側のペインで **General** をクリックします。**Role Name** フィールドに新規ロールの名前を入力します。ロール名が必要です。



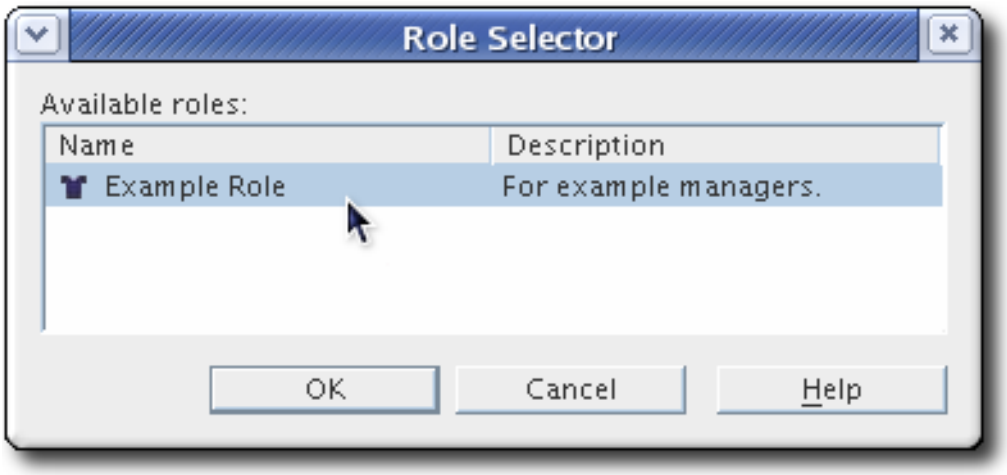
5. 左側のペインで **Members** をクリックします。

6. 右側のペインで、**Nested Role** を選択します。



7. **Add** をクリックして、ロールを一覧に追加します。ネストされたロールのメンバーは、他の既存ロールのメンバーです。

8. **Available roles** リストからロールを選択し、**OK** をクリックします。



8.2.4.2. コマンドラインでのネスト化されたロールの作成

ロールは、ITU X.509 標準で定義されている **ldapsubentry** オブジェクトクラスから継承されます。さらに、ネスト化された各ロールには、**nsRoleDefinition** オブジェクトクラスから継承する 2 つのオブジェクトクラスが必要です。

- nsComplexRoleDefinition
- nsNestedRoleDefinition

ネストされたロールエントリーには、コンテナーロール内でネスト化するロールを識別するための ***nsRoleDN*** 属性も必要です。任意で、ロールは ***description*** 属性を取ることができます。

ネスト化されたロールのメンバーは、ネストされたロール定義エントリーの ***nsRoleDN*** 属性で指定されたロールのメンバーです。

この例では、管理されたマーケティングのロールとフィルター処理されたセールスマネージャーのロールから 1 つのロールを作成します。

1. **-a** オプションを指定して **ldapmodify** を実行して、新規エントリーを追加します。

2. ネストされたロールエントリーを作成します。ネストされたロールには 4 つのオブジェクトクラスがあります。

   - **nsNestedRoleDefinition**
   - **LDAPsubentry (継承)**
   - **nsRoleDefinition (継承)**

- nsComplexRoleDefinition (継承)

nsRoleDN 属性には、マーケティング管理ロールと営業マネージャーのフィルターが設定されたロールの両方の DN が含まれます。

```
dn: cn=MarketingSales,ou=people,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsNestedRoleDefinition
cn: MarketingSales
nsRoleDN: cn=SalesManagerFilter,ou=people,dc=example,dc=com
nsRoleDN: cn=Marketing,ou=people,dc=example,dc=com
```

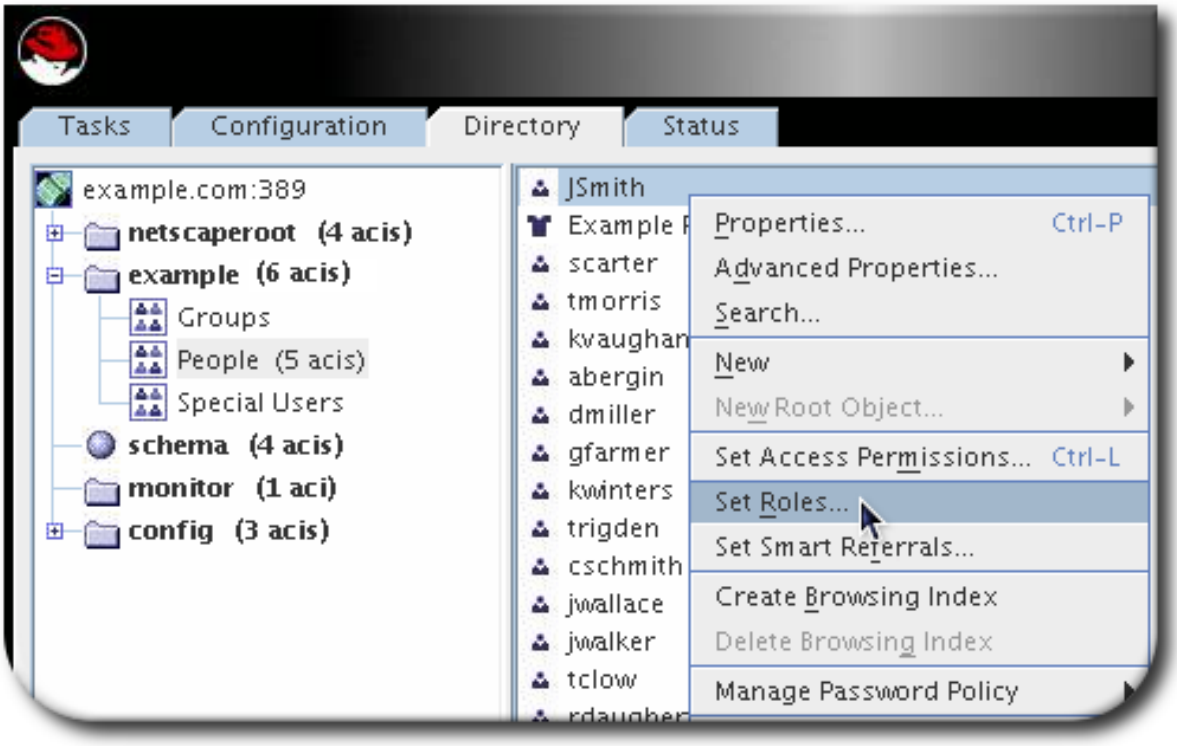
以前の例のユーザー Bob および Pat はどちらも、この新しいネストされたロールのメンバーです。

8.2.5. エントリーへのロールの編集と割り当て

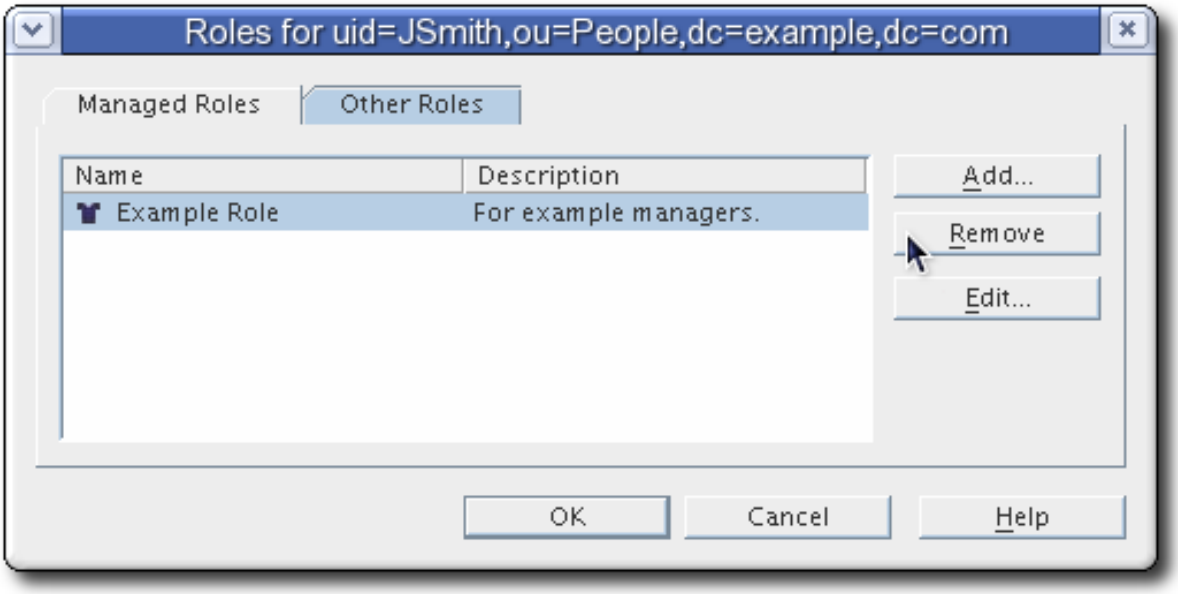
ロールに属するエントリーは、ロールエントリー自体に割り当てられます。管理ロールの場合、ユーザーエントリーは明示的に追加されます。フィルターされたロールの場合、LDAP フィルターの結果により追加されます。

ユーザーエントリーは、メンバーとしてエントリーを追加するか、フィルターを調整することで、コマンドラインでロールに割り当てられます。ただし、Directory Server コンソールには、必要なユーザーエントリーをアクティブに編集してロールにエントリーを追加するショートカットがあります（ただし、機能的にはロールエントリーを編集します）。

1. Directory タブを選択します。
2. 左側のナビゲーションペインでツリーを参照し、ロールを表示または編集するエントリーを選択します。
3. Object メニューから Set Roles を選択します。



4. **Managed Roles** タブを選択して、このエントリーが属する管理ロールを表示します。

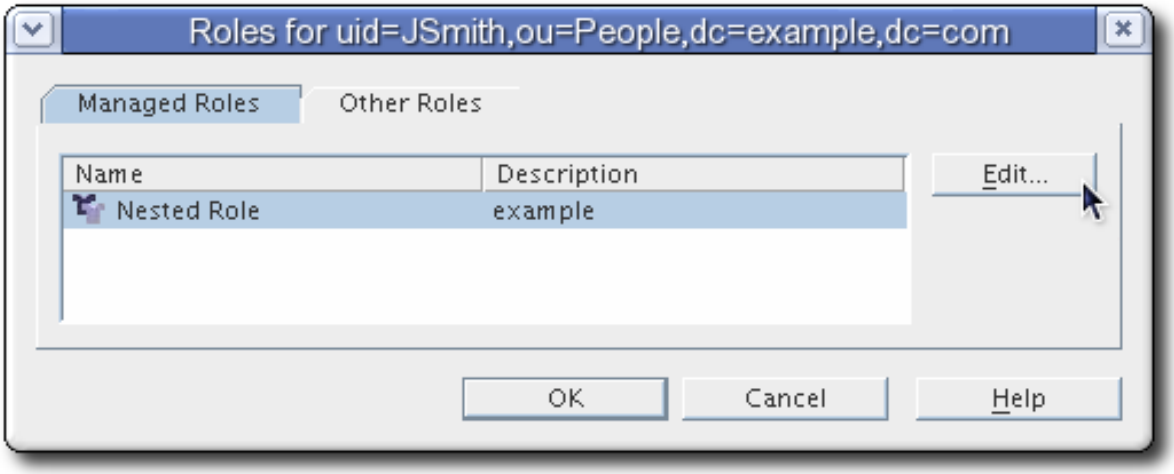


5. 新しい管理ロールを追加するには、**Add** をクリックし、**Role Selector** ウィンドウから利用可能なロールを選択します。

注記

エントリーに関連付けられている管理ロールの設定は、編集 ボタンをクリックして編集できます。**Edit Entry** ダイアログボックスが開き、ロールの一般的な情報またはメンバーを変更できます。

6. **Other Roles** タブを選択して、このエントリーが属するフィルターされたロールまたはネストされたロールを表示します。



Edit をクリックして、エントリーに関連付けられたフィルターまたはネストされたロールを変更します。

8.2.6. コマンドラインでエントリーのロールの表示

Directory Server コンソールに表示されると、エントリーに対するロール割り当てが常に表示されます。ただし、ロールの割り当てはコマンドラインから自動的に返されません。

nsRole 属性は操作の属性です。LDAP では、操作属性を明示的に要求する必要があります。デフォルトでは、エントリーのスキーマに通常の属性が返されません。単一操作属性の一覧を表示することで、明示的に要求するか、**+** を使用して結果オブジェクトの運用上の属性をすべて出力することができます。たとえば、この **ldapsearch** コマンドは、エントリーの通常の属性に加えて **uid=scarter** がメンバーであるロールのリストを返します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b
"dc=example,dc=com" -s sub -x "(uid=scarter)" \* nsRole

dn: uid=scarter,ou=people,dc=example,dc=com
objectClass: inetorgperson
objectClass: top
objectClass: person
objectClass: organizationalPerson
uid: scarter
cn: Sam Carter
sn: Carter
givenName: Sam
mail: scarter@example.com
userPassword: {SSHA}6BE31mhTfcYylQF60kWlnEL8slvPZ59hvFTRKw==
manager: uid=lbrown,ou=people,dc=example,dc=com
nsRole: cn=Role for Managers,dc=example,dc=com
nsRole: cn=Role for Accounting,dc=example,dc=com
```

重要

ロールメンバーシップを評価するには、***nsRole*** 属性ではなく、***nsRoleDN*** 属性を使用するようにしてください。

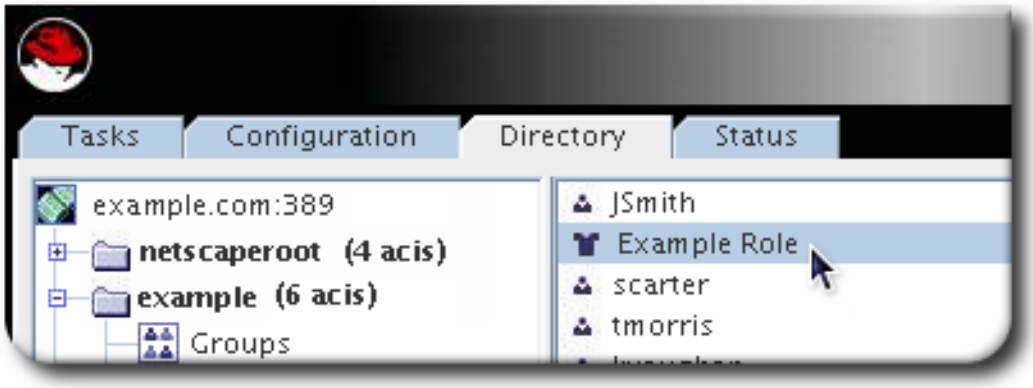
8.2.7. ロールのアクティブまたはアクティブ作成

ロールのアクティベート/非アクティブ化の概念により、エントリーのグループ全体を1つの操作でアクティブまたは非アクティブにすることができます。つまり、あるロールのメンバーを、そのメンバーが属するロールを非アクティブにすることで、一時的に無効にすることができます。

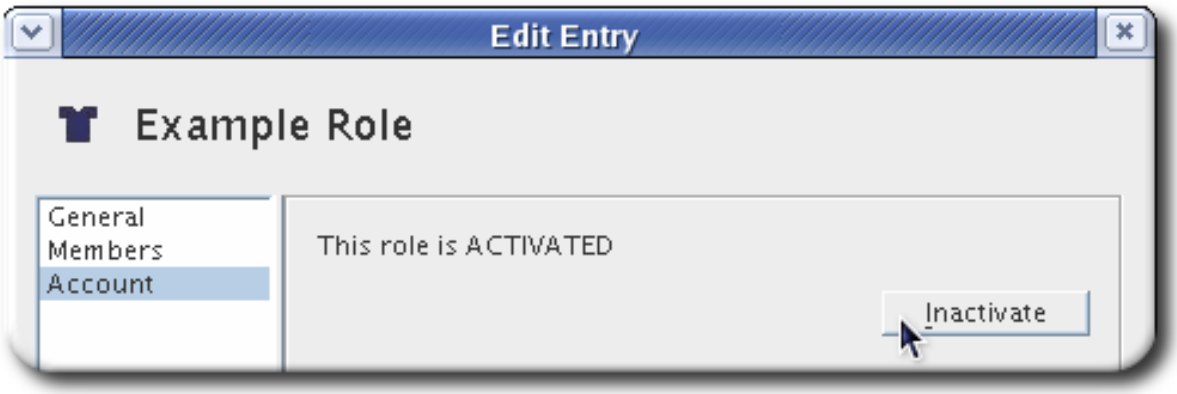
ロールが非アクティブになっても、そのロールエントリーを使用してユーザーをサーバーにバインドできなくなるわけではありません。非アクティブ化されたロールの意味は、そのロールに属するエントリーを使用してユーザーをサーバーにバインドできないということです。非アクティブ化されたロールに属するエントリーは、*nsAccountLock* 属性が true に設定されます。

ロールのメンバーは、所属するロールを非アクティブにすることで、一時的に無効にできます。ロールを非アクティブ化し、ロール自体ではなく、ロールが所有するエントリーを非アクティブにします。

1. Directory タブを選択します。
2. 左側のペインでナビゲーションツリーを参照し、ロールのベース DN を見つけます。ロールは、他のエントリーを含む右側のペインに表示されます。



3. ロールをダブルクリックして、Account タブを開き、Inactivate ボタンをクリックします。



または、ロールを選択します。ロールを右クリックし、メニューから Inactivate を選択します。

ロールは非アクティブです。

無効化されたロールを再度アクティブにするには、ロール設定を再度開くか、Object メニューを開き、Activate を選択します。ロールのすべてのメンバーが再度有効化されます。

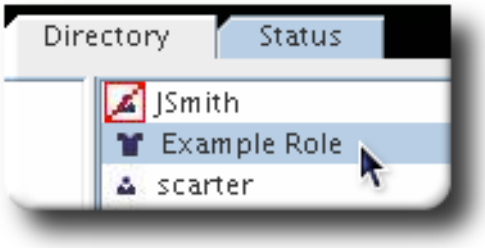
8.2.8. エントリーのアクティベーションステータスの表示

ネスティングされたロールが非アクティブになると、ネスティングされたロール内の任意のロールのメンバーである場合、ユーザーをサーバーにバインドできません。ネスティングされたロールの直接的ま

たは間接的なメンバーであるロールに属するすべてのエントリーは、*nsAccountLock* が true に設定されます。何層にもなったネスティングロールが可能です。ネスティング内のどのネスティングされたロールを非アクティブにしても、その下にあるすべてのロールとユーザーが非アクティブになります。

Directory Server コンソールは、エントリーの active または inactive ステータスを自動的に表示します。

非アクティブエントリーを表示するには、View メニューから Inactivation State を選択します。非アクティブロールのメンバーには、そのロールを介して赤色があります。たとえば、John Smith は、非アクティブな Example ロールのメンバーです。



nsAccountLock 属性は操作属性で、検索属性のリストの検索コマンドで明示的に要求するか、すべての操作属性を要求するために + を指定する必要があります。以下に例を示します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b
"dc=example,dc=com" -s sub -x "(uid=scarter)" nsAccountLock
```

8.2.9. ロールの削除の概要

ロールを削除するとロールエントリーが削除されますが、各ロールメンバーの *nsRoleDN* 属性は削除されません。各ロールメンバーの *nsRoleDN* 属性を削除するには、Referential Integrity プラグインを有効にし、*nsRoleDN* 属性を管理するように設定します。Referential Integrity プラグインの詳細は、「5章 *参照整合性の維持*」を参照してください。

8.2.10. セキュアなロールの使用

すべてのロールがセキュリティーコンテキストでの使用に適しているわけではありません。新しいロールを作成するときは、そのロールをエントリーに割り当てたり、エントリーから削除したりするのがどれほど簡単かを考慮してください。ユーザーが自身をロールに簡単に追加または削除できることが適切な場合があります。たとえば、Mountain Biking と呼ばれるグループロールがある場合は、関心のあるユーザーは自身を追加したり、それ自体を簡単に削除したりできます。

ただし、セキュリティー状況によっては、このようなオープンなロールを持つことは不適切です。潜在的なセキュリティーリスクの 1 つは、ロールを非アクティブ化してユーザー アカウントを非アクティブ化することです。非アクティブなロールには、接尾辞に特別な ACI が定義されます。管理者により、ロールを自由に追加および削除することが許可されている場合、状況によっては、アカウントがロックされないように、非アクティブなロールから自身を削除できる場合があります。

たとえば、ユーザー A には管理ロール MR があります。MR ロールは、アカウントの非アクティブ化を使用してロックされています。これは、*nsAccountLock* 属性はそのユーザーの true として計算されるため、ユーザー A はサーバーにバインドできないことを意味します。ただし、ユーザー A がすでに Directory Server にバインドされ、MR ロールでロックされたことに気付くと、ユーザーは自分のエントリーから *nsRoleDN* 属性を削除して、自身を妨げる ACI がない場合は自身のロックを解除します。

ユーザーが *nsRoleDN* 属性を削除しないようにするには、使用されているロールのタイプに応じて以下の ACI を使用します。

- 管理対象ロール。管理ロールのメンバーであるエントリーの場合は、以下の ACI を使用して適切な *nsRoleDN* を削除して、ユーザーが自身をロック解除しないようにします。

```
aci: (targetattr="nsRoleDN") (targattrfilters= add=nsRoleDN:(!
(nsRoleDN=cn=AdministratorRole,dc=example,dc=com)), del=nsRoleDN:(!
(nsRoleDN=cn=nsManagedDisabledRole,dc=example,dc=com))) (version3.0;acl "allow
mod of nsRoleDN by self but not to critical values"; allow(write) userdn=ldap:///self;)
```

- フィルターが設定されたロール。フィルターに含まれる属性は保護されるべきです。これにより、ユーザーは属性を変更してフィルターされたロールを再取得できません。ユーザーは、フィルター処理されたロールによって使用される属性を追加、削除、または変更することを許可されるべきではありません。フィルター属性の値が計算された場合、フィルター属性の値を変更できるすべての属性を同じ方法で保護する必要があります。

- ネストされたロール。ネストされたロールはフィルターされたロールと管理対象のロールで構成されているため、両方の ACI はネストされたロールを構成するロールの属性 (*nsRoleDN* またはその他) の変更について考慮する必要があります。

アカウントのアクティブ化に関する詳しい情報は、「ユーザーおよびロールの手動による非アクティブ化」を参照してください。

8.3. デュアルエントリーの自動作成

一部のクライアントおよび Red Hat Directory Server と統合には、2 つのエントリーが必要です。たとえば、Posix システムには、通常、各ユーザーにグループがあります。Directory Server の 管理エントリープラグイン は、適切な作成元のエントリーが作成されるたびに、属性の正確な値と特定の値で新しい管理エントリーが自動的に作成されます。

8.3.1. 管理対象エントリー

管理エントリープラグインの背後にある基本的な概念は、エントリー A の作成時に、関連する属性値を含むエントリー B が自動的に配置される必要があることです。たとえば、Posix ユーザー (posixAccount エントリー) が作成されると、対応するグループエントリー (posixGroup エントリー) も作成する必要があります。管理エントリープラグインのインスタンスは、プラグインが、新しいエントリー (管理エントリー) を自動的に生成するエントリー (作成元のエントリー) を識別します。

プラグインは、ディレクトリーツリーの定義範囲内で機能し、指定した検索フィルターに一致するエントリーのみが管理エントリー操作をトリガーします。

サービスのクラスを設定するのと同様に、管理エントリーは 2 つのエントリーで設定されます。

- プラグインインスタンスおよび使用するテンプレートの範囲を特定する定義エントリー

- 最終的な管理エントリーをモデル化するテンプレートエントリー

8.3.1.1. インスタンス定義エントリーの概要

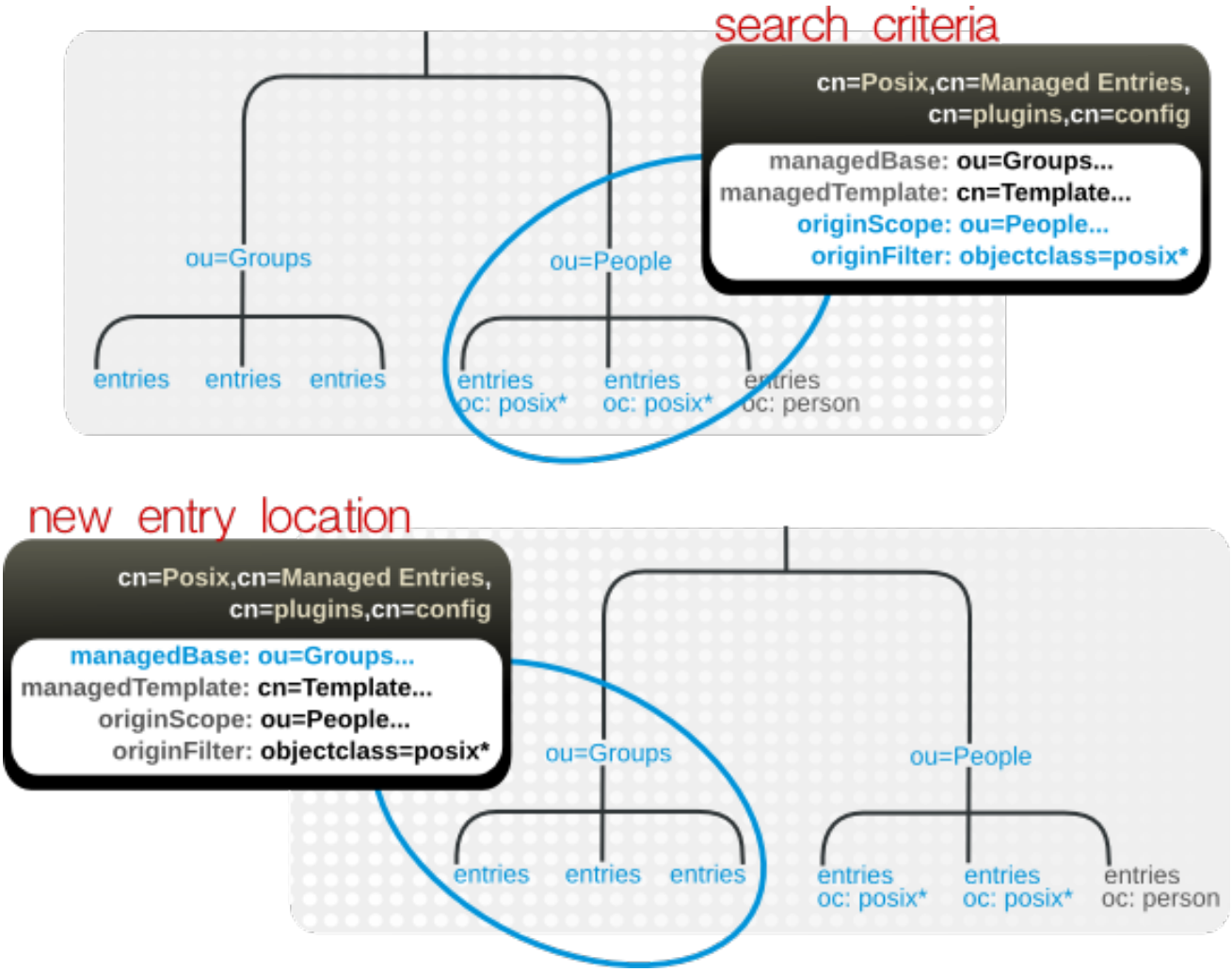
リンク先属性および DNA プラグインと同様に、Managed Entries プラグインには cn=plugins,cn=config にコンテナーエントリーがあり、プラグインの各固有の設定インスタンスには、そのコンテナーの下に定義エントリーがあります。

管理エントリープラグインのインスタンスは、以下の 3 つを定義します。

- (検索範囲と検索フィルターを使用する) 作成元のエントリーを識別する検索基準

- 管理エントリーを作成するサブツリー (新しいエントリーの場所)
- 管理エントリーに使用するテンプレートエントリー

図8.2 管理エントリーの定義



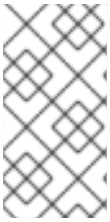
以下に例を示します。

```
dn: cn=Posix User-Group,cn=Managed Entries,cn=plugins,cn=config
objectclass: extensibleObject
cn: Posix User-Group
originScope: ou=people,dc=example,dc=com
originFilter: objectclass=posixAccount
managedBase: ou=groups,dc=example,dc=com
managedTemplate: cn=Posix User-Group Template,ou=Templates,dc=example,dc=com
```

作成元のエントリーには、管理エントリーを作成するために特別な構成または設定は必要ありません。プラグインの範囲内に作成し、指定の検索フィルターと一致させる必要があります。

8.3.1.2. テンプレートエントリーの概要

プラグインの各インスタンスは、管理エントリー設定を定義するテンプレートエントリーを使用します。テンプレートは、オブジェクトクラスからエントリーの値にエントリーを効果的に配列します。



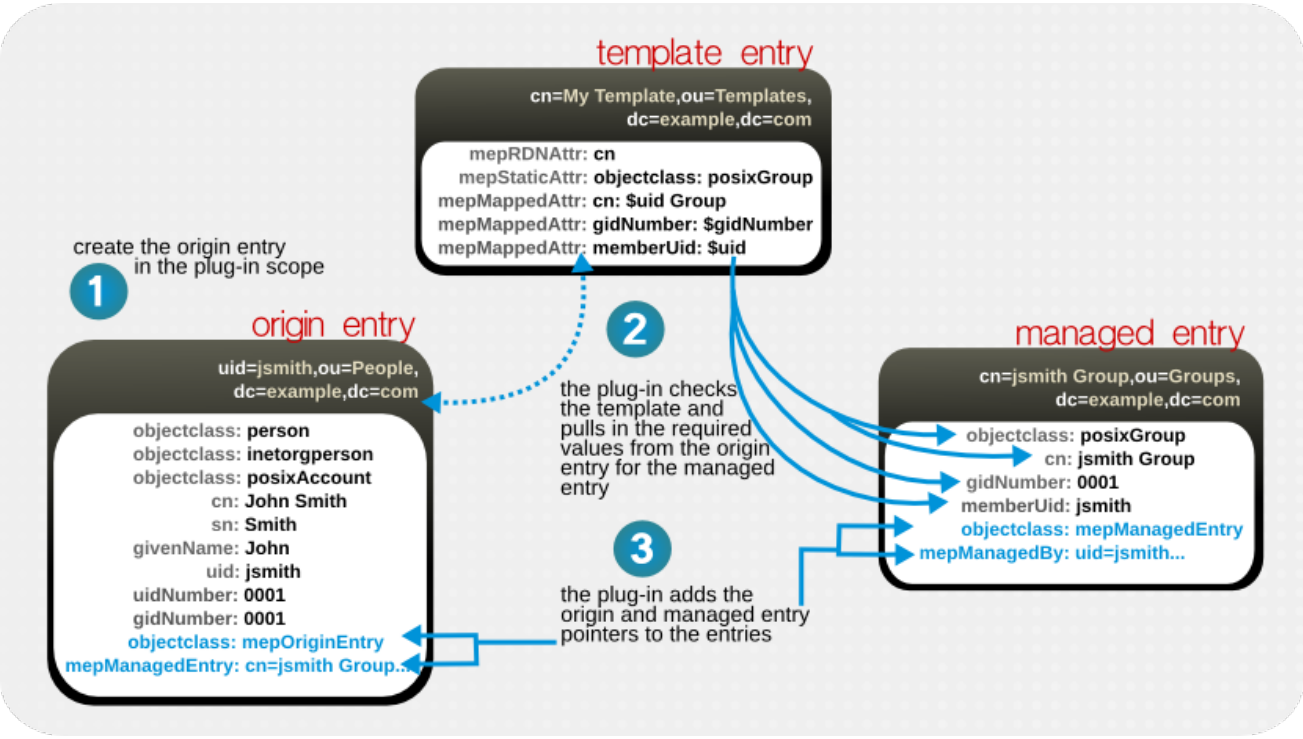
注記

テンプレートは定義エントリーで参照されるため、ディレクトリーのどこにでも配置できます。ただし、テンプレートエントリーを複製された接尾辞の下に置くことが推奨されます。これにより、マルチマスターレプリケーションの他のマスターはすべて、管理エントリープラグインのローカルインスタンスに同じテンプレートを使用します。

テンプレートエントリーの概念は、CoS で使用されるテンプレートに似ていますが、重要な相違点があります。管理対象エントリーテンプレートは、サービスのクラスに使用されるテンプレートのタイプとは若干異なります。サービスのクラスの場合、テンプレートには、その CoS に属するすべてのエントリーに読み込まれる特定の値を持つ 1 つの属性が含まれます。このようなエントリーの CoS 属性は、エントリーに設定した属性ではなく、仮想属性であるため、サービスのクラスへの変更は直ちに関連エントリーに反映されます。

一方、管理エントリープラグインのテンプレートエントリーは、関連するエントリーに値を提供する中央エントリーではありません。これは真のテンプレートです。エントリーの内容をレイアウトします。テンプレートエントリーには、静的属性 (CoS のように事前定義の値を持つもの) とマップされた属性 (作成元のエントリーからの値または値の一部をプルする属性) の両方を含めることができます。テンプレートは、管理エントリーが作成され、作成元のエントリーが変更され、テンプレートがこれらの更新を適用するために再度評価される場合に のみ 管理エントリーに適用されます。

図8.3 テンプレート、管理対象エントリー、およびアーティファクトエントリー



テンプレートは、テンプレートに static 属性を使用して、管理エントリーの属性に特定の値を指定できます。テンプレートは、作成元のエントリーの一部の属性から派生する値を使用することもできるため、この値はエントリーへのエントリーとは異なる場合があります。これは、値ではなく作成元のエントリーの属性タイプを参照するため、マップされた 属性とは異なる場合があります。

マップされた値は、トークン (動的な値) と静的な値の組み合わせを使用しますが、マップされた属性ごとに 1 つのトークン しか使用できません。

```
dn: cn=Posix User-Group Template,ou=Templates,dc=example,dc=com
objectclass: mepTemplateEntry
cn: Posix User-Group Template
mepRDNAttr: cn
```

```
mepStaticAttr: objectclass: posixGroup
mepMappedAttr: cn: $cn Group Entry
mepMappedAttr: gidNumber: $gidNumber
mepMappedAttr: memberUid: $uid
```

テンプレートのマップされた属性は、先頭にドル記号 ($) を追加して作成元のエントリーから値をプルして管理エントリーで使用するトークンを使用します。ドル記号が管理属性値で実際に定義されている場合は、ドル記号を 2 つ使用してドル記号をエスケープできます。

マッピングされた属性の定義は、**Attr: ${cn}test** など、中括弧で囲んで引用することができます。トークン名の直後にスペースやコンマなど属性名として有効な文字が使用される場合は、トークンの値を引用符で囲む必要はありません。たとえば、**$cn test** は属性名の直後に続くため、属性定義では使用できますが、管理エントリープラグインは作成元のエントリーで **cntest** という名前の属性を検索しようとするため、**$cntest** は有効ではありません。中括弧を使用すると、属性トークン名を識別します。

注記

静的属性およびマップされた属性に値が必要な属性の構文に準拠するようにしてください。

8.3.1.3. 管理エントリープラグインにより書き込まれるエントリー属性

作成元のエントリーと管理エントリーの両方には、管理エントリープラグインのインスタンスによって管理されていることを示す特別な管理エントリー属性があります。作成元のエントリーでは、プラグインは関連付けられた管理エントリーへのリンクを追加します。

```
dn: uid=jsmith,ou=people,dc=example,dc=com
objectclass: mepOriginEntry
objectclass: posixAccount
...
sn: Smith
mail: jsmith@example.com
mepManagedEntry: cn=jsmith Posix Group,ou=groups,dc=example,dc=com
```

管理エントリーでは、プラグインはテンプレートで定義された属性の他に、作成元のエントリーを参照する属性を追加します。

```
dn: cn=jsmith Posix Group,ou=groups,dc=example,dc=com
objectclass: mepManagedEntry
objectclass: posixGroup
...
mepManagedBy: uid=jsmith,ou=people,dc=example,dc=com
```

特別な属性を使用して、管理および作成元のエントリーを示すことで、関連するエントリーを簡単に特定し、管理エントリープラグインによる変更を評価できます。

8.3.1.4. 管理エントリープラグインおよび Directory Server 操作

管理エントリープラグインでは、追加操作および削除操作と同様に、Directory Server が一般的な操作を実行する方法に影響します。

表8.4 管理エントリープラグインおよび Directory Server 操作

演算子	管理対象エントリープラグインによる効果
Add	すべての追加操作では、サーバーは新しいエントリーが管理エントリープラグインインスタンスの範囲内にあるかどうかを確認します。作成元エントリーの基準を満たすと、管理エントリーが作成され、管理エントリー関連の属性が元のエントリーと管理エントリーの両方に追加されます。
Modify	作成元のエントリーが変更すると、管理エントリーを更新するプラグインが誘発されます。ただし、**テンプレート** エントリーを変更しても、自動的に管理エントリーを更新しません。テンプレートエントリーへの変更は、次に作成元エントリーが変更するまで、管理エントリーに反映されません。 管理エントリー **内** でマップされた管理属性は、管理エントリープラグインでのみ手動で変更することができません。管理対象エントリーの他の属性 (管理エントリープラグインで追加された静的属性を含む) は手動で変更できます。
Delete	作成元のエントリーが削除されると、管理エントリープラグインもそのエントリーに関連付けられた管理エントリーを削除します。削除できるエントリーにはいくつかの制限があります。 ● テンプレートエントリーは、プラグインインスタンス定義で現在参照されている場合は削除できません。 ● 管理エントリーは、管理エントリープラグイン以外では削除できません。
Rename	元のエントリーの名前を変更した場合は、プラグインは対応する管理エントリーを更新します。エントリーがプラグインスコープの **外** に移動すると、管理エントリーが削除されますが、エントリーがプラグインスコープの **中** に移動した場合は、追加操作のように処理され、新しい管理エントリーが作成されます。削除操作と同様に、名前変更または移動できるエントリーに制限があります。 ● 設定定義エントリーは、コンテナーエントリーの管理エントリープラグインを外に移動できません。エントリーが削除されると、そのプラグインインスタンスが非アクティブになります。 ● エントリーが管理エントリープラグインのコンテナーエントリー **内** に移動する場合、これは検証され、アクティブな設定定義として処理されます。 ● テンプレートエントリーの名前を変更したり、プラグインインスタンス定義で現在参照している場合は移動したりすることはできません。 ● 管理エントリープラグインの名前を変更または移動することはできません。
レプリケーション	管理エントリープラグイン操作は **レプリケーションの更新によって開始しません**。プラグインスコープのエントリーの追加または修正操作が別のレプリカに複製される場合、その操作はレプリカで管理エントリープラグインインスタンスを発生させず、エントリーを作成または更新しません。管理エントリーの更新を複製する唯一の方法は、最終管理エントリーをレプリカに複製することです。

8.3.2. 管理対象エントリーテンプレートエントリーの作成

最初に作成するエントリーはテンプレートエントリーです。テンプレートエントリーには、生成される管理エントリーに必要なすべての設定を含める必要があります。これは、テンプレート内の静的およびマップされた属性に属性値表明を設定して行います。

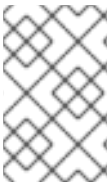
```
mepStaticAttr: attribute: specific_value
mepMappedAttr: attribute: $token_value
```

静的属性は明示的な値を設定し、マップされた属性は、元のエントリーから一部の値をプルし、指定の属性を提供するためにそれを使用します。これらの属性の値は、attribute: $attr のトークンになります。属性の拡張トークンの構文が必要な属性構文に違反しない限り、他の用語や文字列を属性で使用できます。以下に例を示します。

```
mepMappedAttr: cn: Managed Group for $cn
```

管理エントリーについて、その後に設定する必要がある構文ルールがいくつかあります。

- **マップされた値は、トークン (動的な値) と静的な値の組み合わせを使用しますが、マップされた属性ごとに 1 つのトークン しか使用できません。**
- **テンプレートのマップされた属性は、先頭にドル記号 ($) を追加して作成元のエントリーから値をプルして管理エントリーで使用するトークンを使用します。ドル記号が管理属性値で実際に定義されている場合は、ドル記号を 2 つ使用してドル記号をエスケープできます。**
- **マッピングされた属性の定義は、Attr: ${cn}test など、中括弧で囲んで引用することができます。トークン名の直後にスペースやコンマなど属性名として有効な文字が使用される場合は、トークンの値を引用符で囲む必要はありません。たとえば、$cn test は属性名の直後に続くため、属性定義では使用できますが、管理エントリープラグインは作成元のエントリーで cntest という名前の属性を検索しようとするため、$cntest は有効ではありません。中括弧を使用すると、属性トークン名を識別します。**
- **静的属性およびマップされた属性に値が必要な属性の構文に準拠するようにしてください。**



注記

静的属性およびマップされた属性に値が必要な属性の構文に準拠するようにしてください。たとえば、マップされた属性のいずれかが *gidNumber* の場合、マップされた値は整数にする必要があります。

表8.5 管理エントリーテンプレートの属性

属性	説明
mepTemplateEntry (オブジェクトクラス)	テンプレートとしてエントリーを識別します。
cn	エントリーの共通名を指定します。
mepMappedAttr	プラグインは、元のエントリーから取得した値で管理エントリーの属性を作成するために使用する属性とトークンのペアが含まれます。
mepRDNAttr	管理エントリーで naming 属性として使用する属性を指定します。RDN として使用される属性は、設定のためにマップされた属性である **必要があります**。
mepStaticAttr	管理エントリーに指定された値とともに使用される属性と値のペアが含まれます。

テンプレートエントリーを作成するには、以下を実行します。

1. **ldapmodify** を実行してエントリーを追加します。このエントリーは、ディレクトリーツリーの任意の場所に置くことができます。

```
dn: cn=Posix User Template,ou=templates,dc=example,dc=com
cn: Posix User Template
...
```

「ディレクトリーエントリーの作成」で説明されているように、Directory Server コンソールを使用して、エントリーを作成することもできます。

2. **mepTemplateEntry** オブジェクトクラスに、これがテンプレートエントリーであることを示します。

```
objectClass: top
objectclass: mepTemplateEntry
...
```

3. エントリーの属性を設定します。これらは、表8.5「管理エントリーテンプレートの属性」で説明されています。RDN 属性(***mepRDNAttr***)が必要です。属性パラメーターは任意で、値はプラグインが作成するエントリーのタイプによって異なります。naming 属性に使用する属性がマップされた属性としてテンプレートエントリーに含まれていることを確認してください。

注記

エントリーのオブジェクトクラスと同様に、各管理エントリーに対して同じ属性。値を手動で設定する必要があります。***mepStaticAttr***

```
mepRDNAttr: cn
mepStaticAttr: objectclass: posixGroup
mepMappedAttr: cn: $cn Group Entry
mepMappedAttr: gidNumber: $gidNumber
mepMappedAttr: memberUid: $uid
```

8.3.3. 管理対象エントリーインスタンス定義の作成

テンプレートエントリーが作成されると、そのテンプレートを参照する定義エントリーを作成できます。定義エントリーは、管理エントリープラグインのインスタンスです。

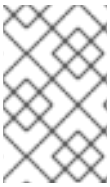
注記

定義が作成されると、サーバーは指定されたテンプレートエントリーが存在するかどうかを確認します。サーバーは、テンプレートが存在しないと、定義設定が無効であるという警告を返します。

定義エントリーは、潜在的な元のエントリーと管理エントリーを作成する情報を認識するためにパラメーターを定義する必要があります。プラグインインスタンスに使用できる属性は、表8.6「管理エントリー定義エントリーの属性」に記載されています。

表8.6 管理エントリー定義エントリーの属性

属性名	説明
originFilter	検索に使用する検索フィルターで、管理エントリーを必要とするサブツリーのエントリーを特定します。構文は、通常の検索フィルターと同じです。
originScope	監視するプラグインの潜在的な元のエントリーを含むベースサブツリー。
managedTemplate	管理エントリーの作成に使用するテンプレートエントリーを特定します。このエントリーは、ディレクトリーツリーの任意の場所に置くことができます。
managedBase	管理エントリーを作成するサブツリー。



注記

管理エントリープラグインはデフォルトで有効です。このプラグインが無効になっている場合は、「Directory Server コンソールでプラグインの有効化」 の説明に従って再度有効にします。

インスタンスを作成するには、以下を実行します。

1. **ldapmodify を使用して、cn=Managed Entries,cn=plugins,cn=config コンテナーエントリーの下に新しいプラグインインスタンスを作成します。**

```
dn: cn=instance,cn=Managed Entries,cn=plugins,cn=config
...
```

2. **作成元のエントリー検索、新しい管理エントリーの場所、使用するテンプレートエントリーの場所を設定します。これらの必須属性は、表8.6「管理エントリー定義エントリーの属性」 に一覧表示されます。**

```
objectClass: top
objectClass: extensibleObject
cn: Posix User-Group
originScope: ou=people,dc=example,dc=com
originFilter: objectclass=posixAccount
managedBase: ou=groups,dc=example,dc=com
managedTemplate: cn=Posix User-Group Template,ou=Templates,dc=example,dc=com
```

3. **Directory Server が動的プラグインを有効にするために設定されていない場合は、サーバーを再起動して変更した新しいプラグインインスタンスを読み込みます。**

8.3.4. 複製されたデータベースへの管理エントリープラグイン設定の追加

「管理対象エントリー」強調表示されているように、管理エントリープラグインの異なるインスタンスが、cn=plugins,cn=com のコンテナープラグインエントリーの下にある子として作成されます。(これは、複数のインスタンスを許可するプラグインに共通です。 この欠点は、cn=plugins,cn=com の設定エントリーが複製されないため、各 Directory Server インスタンスに設定を再作成する必要があります。

管理エントリープラグインエントリーでは、*nsslapd-pluginConfigArea* 属性が許可されます。この属性は、プラグインインスタンスエントリーが含まれるメインのデータベースエリアにおける、別のコンテナーエントリーへの属性です。このコンテナーエントリーは、複製されたデータベースで使用することができます。これにより、プラグイン設定を複製できます。

1. **ldapmodify** を使用して、レプリケートされるサブツリーにコンテナーエントリーを作成します。

```
dn: cn=managed entries container,ou=containers,dc=example,dc=com
objectclass: top
objectClass: extensibleObject
objectClass: nsContainer
cn: managed entries container
```

2. **ldapmodify** を使用して、コンテナーエントリーを参照する管理エントリープラグインエントリーに *nsslapd-pluginConfigArea* 属性を追加します。

```
dn: cn=Managed Entries,cn=plugins,cn=config
changetype: modify
add: nsslapd-pluginConfigArea
nsslapd-pluginConfigArea: cn=managed entries
container,ou=containers,dc=example,dc=com
```

3. 新規コンテナーエントリーの下にある定義 (「管理対象エントリーインスタンス定義の作成」) エントリーおよびテンプレート (「管理対象エントリーテンプレートエントリーの作成」) エントリーを移動または作成します。

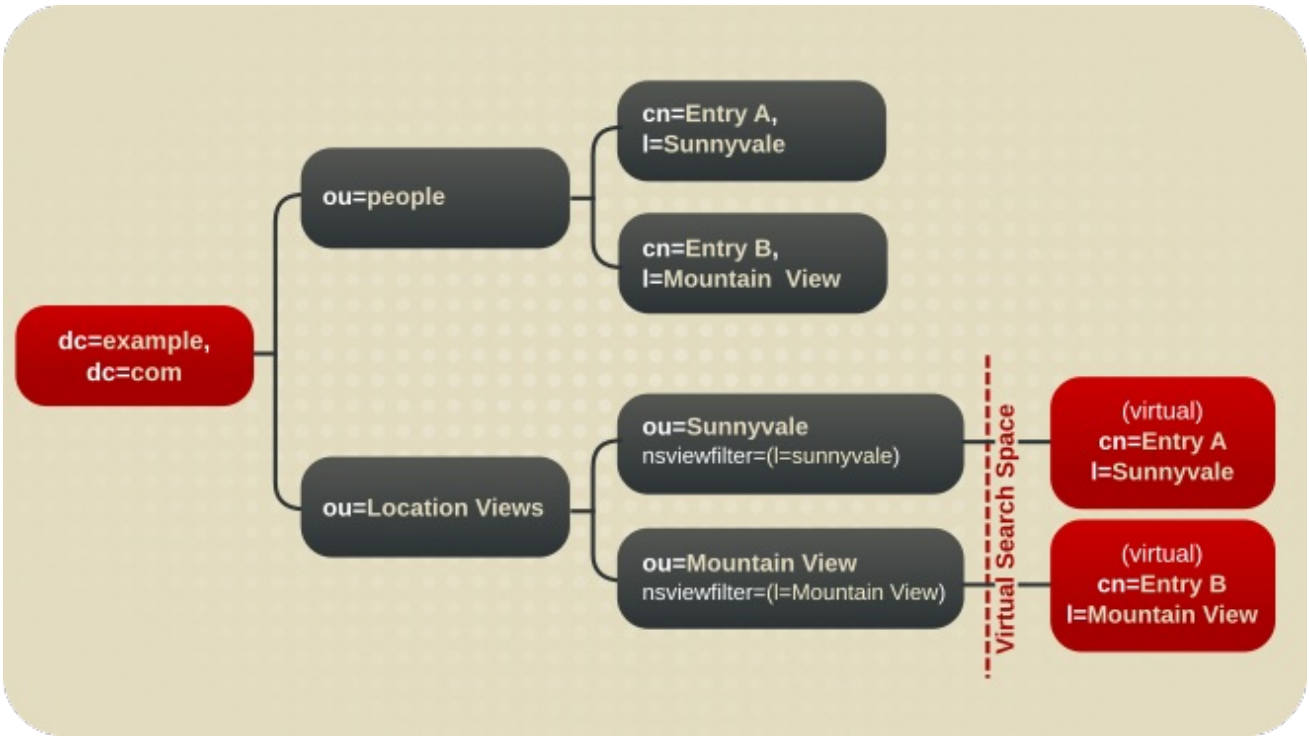
8.4. ビューの使用

仮想ディレクトリーツリービュー、または ビュー は、、仮想ディレクトリー階層を作成するため、これらのエントリーが特定の場所に物理的に存在していることを確認せずに、エントリーの移動が容易になります。ビューは、フィルターされたロールまたは動的グループのメンバーと同様に、エントリーの情報を使用してビュー階層に置きます。ビューは一連のエントリーに DIT 階層を重ね合わせ、クライアントアプリケーションには、ビューは通常のコンテナ階層として表示されます。

8.4.1. ビューの概要

ビューはサブツリーの組織単位エントリーを使用するなど、通常の階層と同様にディレクトリーツリーを作成しますが、ビューには追加のオブジェクトクラス (nsview) およびフィルター属性 (*nsviewfilter*) が含まれており、このビューに属するエントリーのフィルターを設定します。ビューコンテナーエントリーを追加すると、ビューフィルターと一致するすべてのエントリーが即座にビューに入力されます。対象となるエントリーは、ビューの中に存在している ように見える だけで、実際の場所は変わりません。たとえば、ビューは **ou=Location Views** として作成され、フィルターが **l=Mountain View** 設定されます。**cn=Jane Smith,l=Mountain View,ou=People,dc=example,dc=com** どのすべてのエントリーは **ou=Location Views** エントリーで直ちに一覧表示されますが、実際の **cn=Jane Smith** ントリーは **ou=People,dc=example,dc=com** サブツリーに残ります。

図8.4 仮想 DIT ビュー階層を含むディレクトリーツリー



仮想 DIT ビューは、サブツリーまたは 1 レベルの検索が、想定された結果で返されることで、通常の DIT と同様に動作します。

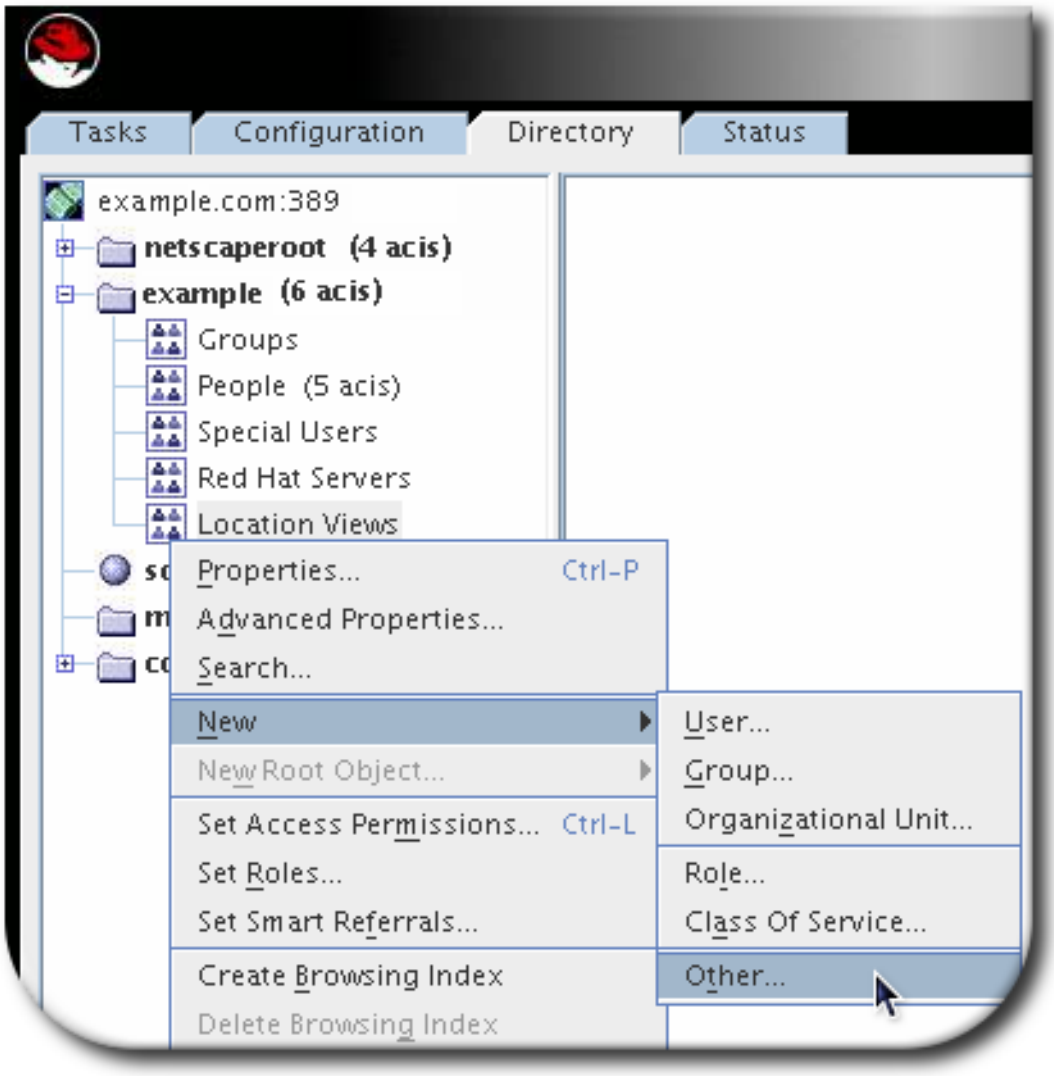
注記

Directory Server でインストールされるビューエントリーの例 Example-views.ldif を含む LDIF ファイルのサンプルがあります。このファイルは、Red Hat Enterprise Linux 7 の /usr/share/dirsrv/data/ ディレクトリーにあります。本章のセクションは、Example-views.ldif がサーバーにインポートされることを前提としています。

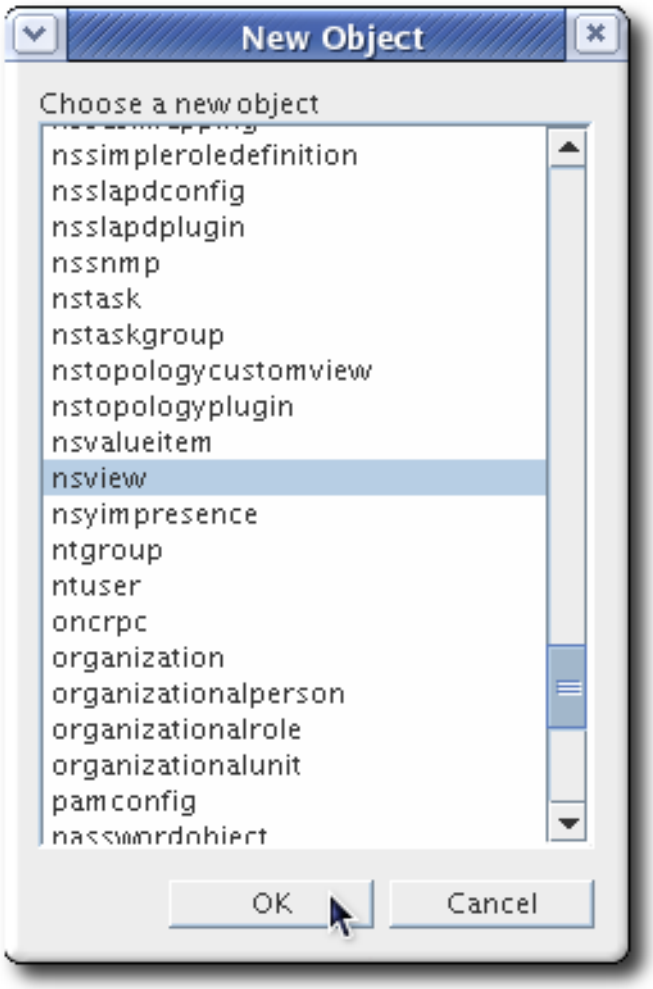
『Red Hat Directory Server デプロイメントガイド』 には、ディレクトリーツリー階層とビューを統合する方法の詳細が記載されています。

8.4.2. コンソールでビューの作成

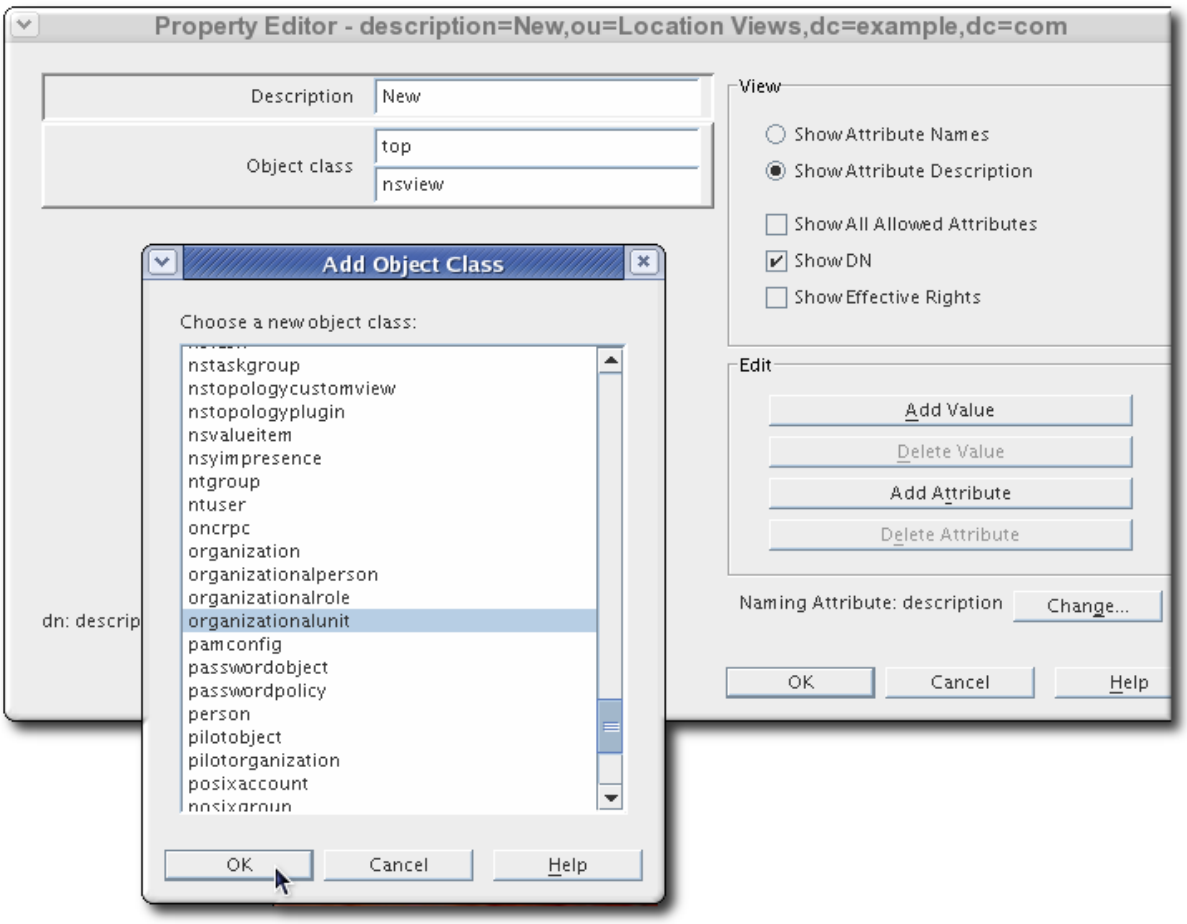
1. Directory タブを選択します。
2. 左側のナビゲーションツリーで、ビューを保持する組織単位の接尾辞を作成します。たとえば、ローカリティー(l)属性に基づいたビューの場合は、この組織単位 の場所ビューに名前を付けます。サブ接尾辞の作成については、「コンソールを使用した新しい従属接尾辞の作成」 を参照してください。
3. ou=Location Views を右クリックし、New > Other の順に選択します。



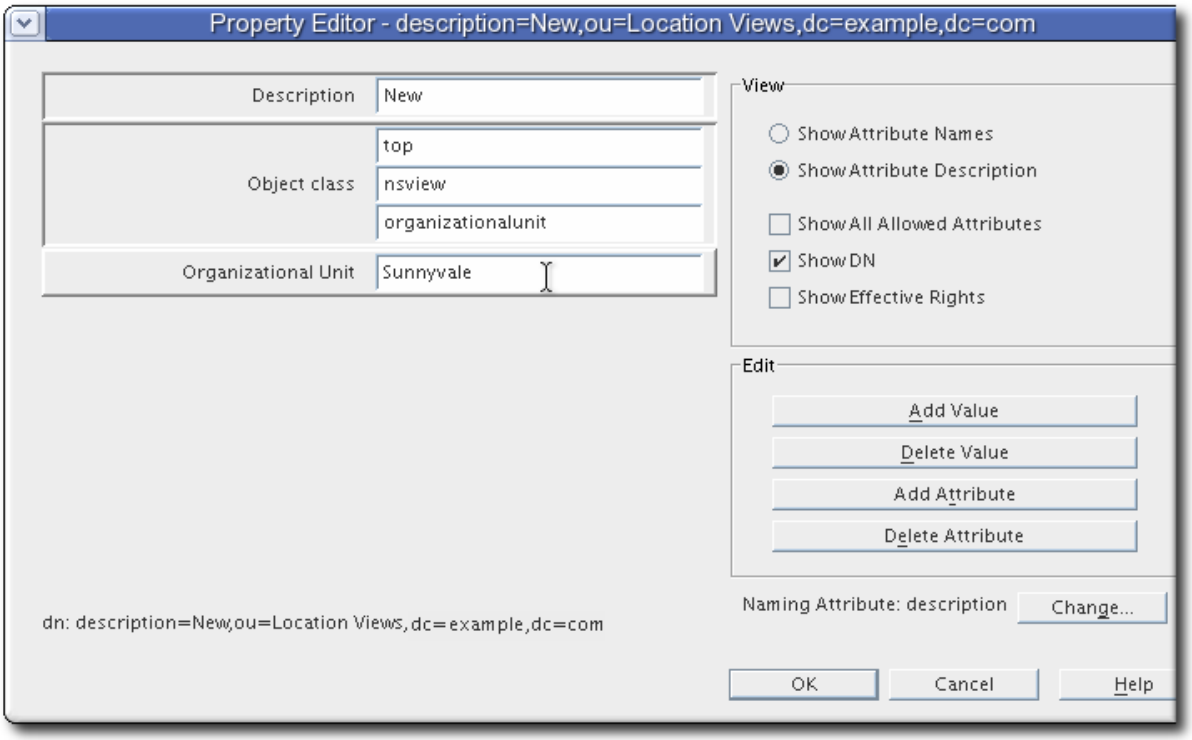
4. **New Object メニューから nsview を選択し、OK をクリックします。**



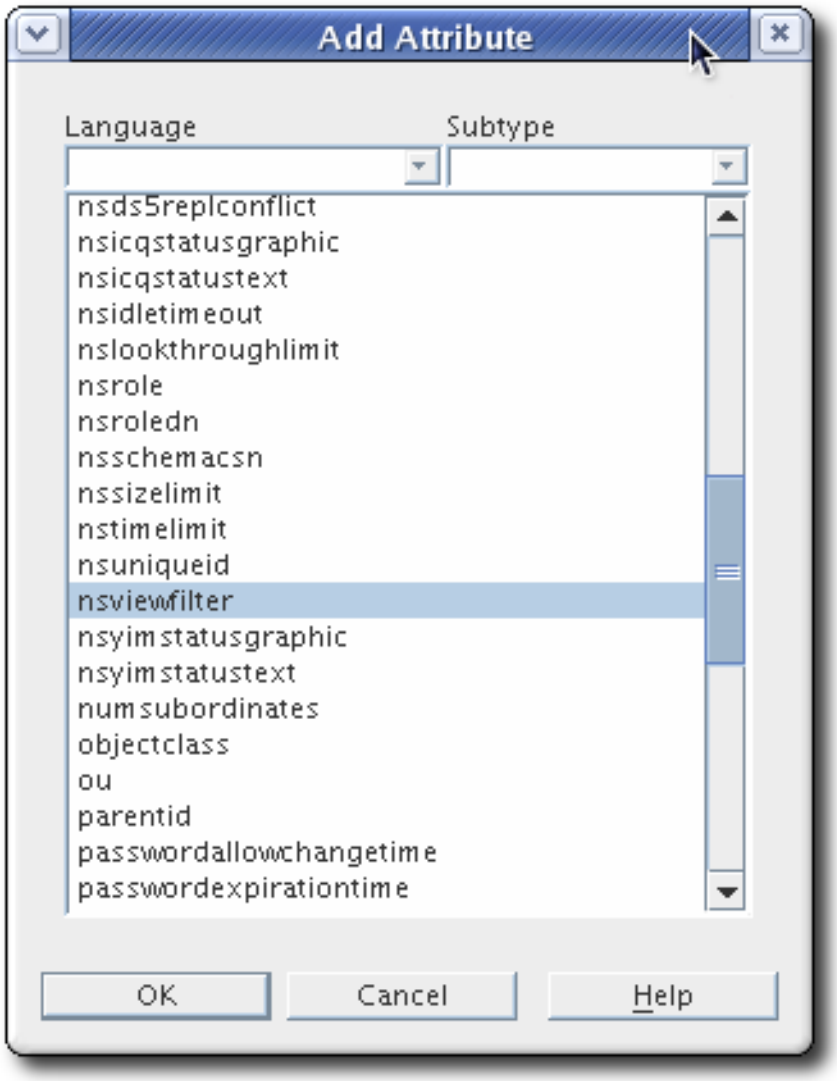
5. **Property Editor ウィンドウで Add Value ボタンをクリックし、組織のユニットオブジェクトクラスを追加します。**



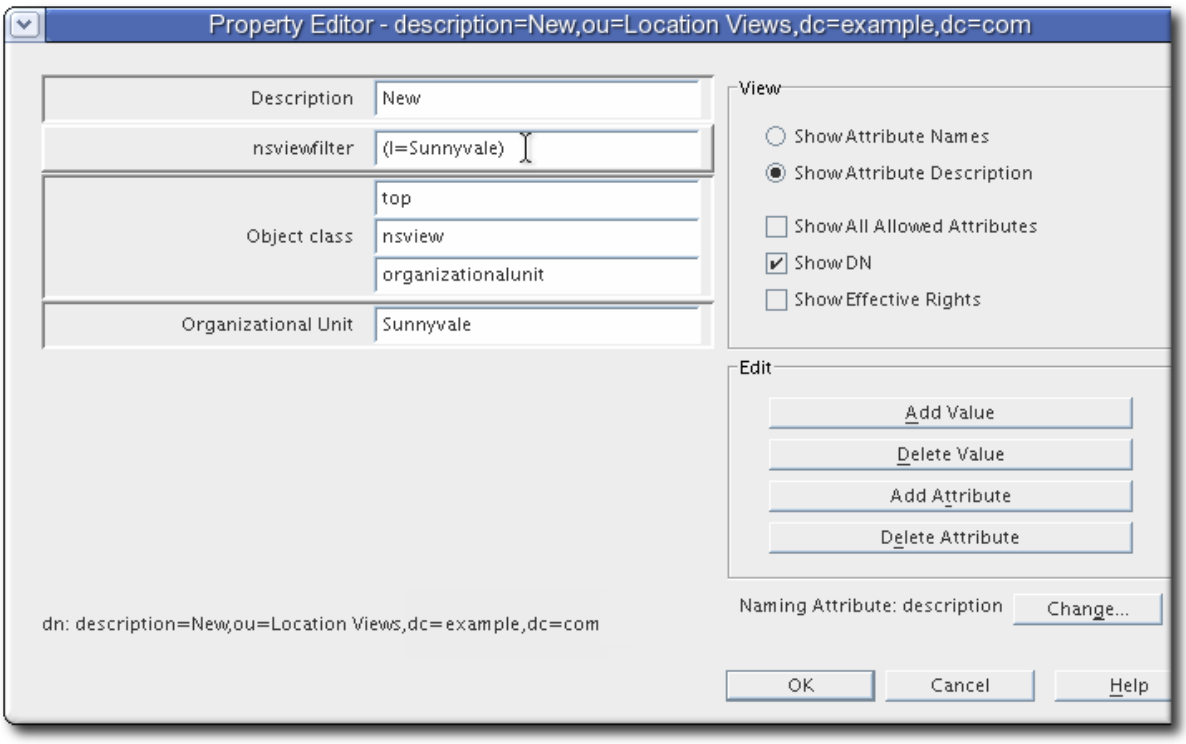
6. **ビューを整理する方法は、組織ユニットに名前を付けます。たとえば、ou=Sunnyvale です。*ou*属性を naming 属性にします。**



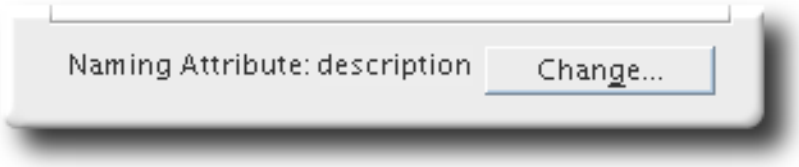
7. **Add Attribute ボタンをクリックし、nsviewfilter 属性を追加します。**



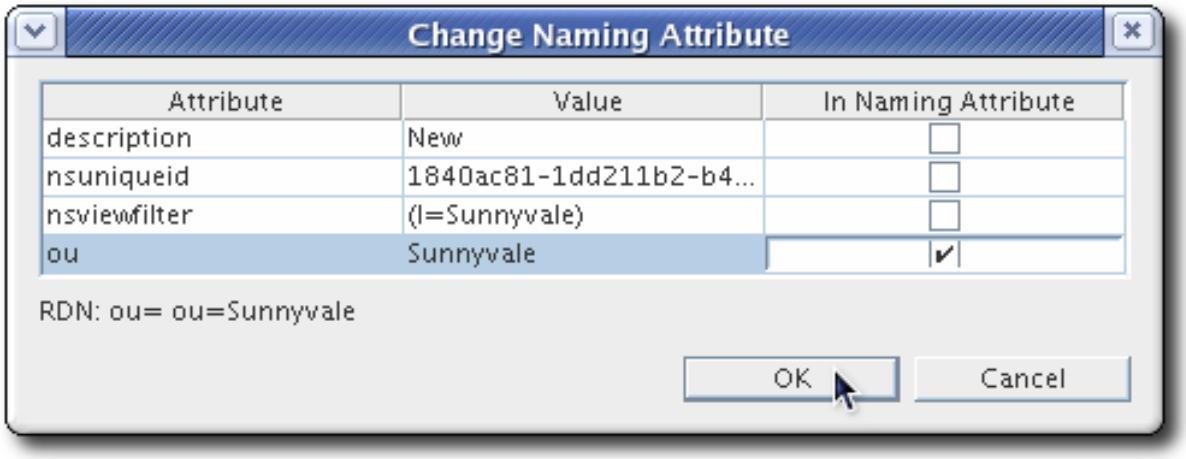
8. **（l=Sunnyvale など）ビューを反映するフィルターを作成します。**



9. **右下の Change ボタンをクリックして naming 属性を変更します。**



エントリーの *description* を、*ou* ではなく naming 属性として使用します。



10. **OK をクリックして属性ボックスを閉じ、再度 OK をクリックして新しいビューエントリーを保存します。**

新しいビューには、検索フィルターに一致するエントリーが自動的に入力され、ディレクトリーに追加された新しいエントリーがビューに自動的に含まれます。

8.4.3. コマンドラインでのビューの作成

1. **ldapmodify ユーティリティーを使用してサーバーにバインドし、新しいビューエントリーを設定ファイルに追加する準備を行います。**

2. **Example-views.ldif ファイルから、ビューコンテナーの ou=Location Views,dc=example,dc=com ファイルを、Directory Server にアサインします。この例では、root 接尾辞 dc=example,dc=com の下に新しい views コンテナーエントリーを追加します。このエントリーには、nsview ブジェクトクラスおよび *nsViewFilter* 属性が必要です。*nsViewFilter* 属性は、ビューに属するエントリーを識別する属性値を設定します。**

```
dn: ou=Mountain View,ou=Location Views,dc=example,dc=com
changetype: add
objectClass: top
objectClass: organizationalUnit
objectClass: nsview
ou: Mountain View
nsViewFilter: l=Mountain View
description: views categorized by location
```

8.4.4. ビューのパフォーマンスの向上

「ビューの概要」の説明通りに、ビューは指定のフィルターに基づいて検索結果から派生します。フィルターの一部は *nsViewFilter* 属性で定義される属性です。フィルターの残りの部分はエントリー階層に基づいており、ビューに含まれる実際のエントリーの *entryid* と *parentid* を探します。

```
(|(parentid=search_base_id)(entryid=search_base_id)
```

searched-for 属性 (*entryid*、*parentid*、または *nsViewFilter* に設定された属性) のいずれかがインデックス化されない場合、views 操作は一致するエントリーのツリー全体を検索するため、ビューの検索はインデックスなしの検索になります。

views パフォーマンスを改善するには、*entryid*、*parentid*、および *nsViewFilter* で設定した属性の等価インデックスを作成します。

等価インデックスの作成については「標準インデックスの作成」で説明されています。また、既存のインデックスを新しい属性を含めるように更新する方法は、「既存のデータベースへの新規インデックスの生成」で説明されています。

第9章 セキュアな接続の設定

デフォルトでは、クライアントおよびユーザーは、標準接続で Red Hat Directory Server に接続します。標準接続では暗号化が使用されないため、サーバーとクライアントの間で情報が平文でやり取りされます。

Directory Server は TLS 接続、**StartTLS** 接続、および SASL 認証をサポートします。これは、傍受されていても、ディレクトリーデータを保護する暗号化およびセキュリティーの層を提供します。

9.1. セキュアな接続の要求

Directory Server は、暗号化された接続を使用する次の方法を提供します。

LDAPS

LDAPS プロトコルを使用すると、接続は暗号化を使用して開始し、成功または失敗します。ただし、暗号化されていないデータがネットワーク経由で送信されることはありません。このため、暗号化されていない LDAP で **StartTLS** を使用する代わりに、LDAPS の使用が推奨されます。

LDAP 上の STARTTLS

クライアントは LDAP プロトコルで暗号化されていない接続を確立し、**StartTLS** コマンドを送信します。コマンドに成功すると、それ以降の通信はすべて暗号化されます。

警告

StartTLS コマンドが失敗し、クライアントが接続をキャンセルしないと、認証情報を含むすべてのデータが暗号化されずにネットワーク上に送信されます。

SASL

Simple Authentication and Security Layer (SASL) を使用すると、Kerberos などの外部認証方法を使用してユーザーを認証できます。詳細は「SASL Identity マッピングの設定」を参照してください。

9.2. 最小強度係数の設定

追加のセキュリティーを確保するために、Directory Server は、接続を許可する前に特定の暗号化レベルを必要とするように設定できます。Directory Server は、すべての接続に特定のセキュリティー 強度係数 (SFF) を定義し、要求できます。SSF は、接続または操作に対するキー強度によって定義される最小限の暗号化レベルを設定します。

すべてのディレクトリー操作に最小限の SSF を必要とするには、***nsslapd-minssf*** 設定属性を設定します。最小 SSF を適用する場合、Directory Server は操作で使用可能な各暗号化タイプ (TLS または SASL) を調べ、どちらの SSF 値が高いかを判断し、高い値を最小 SSF と比較します。SASL 認証と TLS は、レプリケーションなどのサーバー間の接続に対して、SASL 認証と TLS の両方を設定できます。

注記

または、*nsslapd-minssf-exclude-rootdse* 設定属性を使用します。これにより、ルート DSE に対するクエリーを 除き、Directory Server へのすべての接続の最小 SSF 設定が設定されます。クライアントは、操作を開始する前に、デフォルトの命名コンテキストなどのサーバー設定に関する情報を取得しないといけない場合があります。*nsslapd-minssf-exclude-rootdse* 属性を使用すると、クライアントは最初にセキュアな接続を確立しなくてもその情報を取得できます。

接続の最初の操作が開始すると、接続の SSF が評価されます。これにより、2 つの接続が通常の接続を開始した場合でも、StartTLS および SASL バインドは成功します。TLS セッションまたは SASL セッションが開かれると、SSF が評価されます。SSF 要件を満たさない接続は、LDAP がエラーを実行することを拒否して閉じられます。

最小の SSF を設定して、セキュアでない接続がディレクトリーへの接続を無効にします。

警告

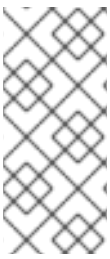
SASL を使用せずに暗号化されていない LDAP プロトコルを使用してディレクトリーに接続する場合、最初の LDAP メッセージにはバインド要求を含めることができます。この場合、SSF は設定された最小値を満たしていないため、サーバーが接続をキャンセルする前に、認証情報がネットワーク経由で暗号化されずに送信されます。

LDAPS プロトコルまたは SASL バインドを使用して、認証情報を暗号化せず送信しないようにします。

デフォルトの *nsslapd-minssf* 属性値は 0 です。これは、サーバー接続の最小 SSF がないことを意味します。値は、適切な正の整数に設定できます。値は、セキュアな接続に必要な鍵強度を表します。

以下の例では、*nsslapd-minssf* 属性を cn=config エントリーに追加します。

```
# ldapmodify -D "cn=Directory Manager" -W -x

dn: cn=config
changetype: modify
replace: nsslapd-minssf
nsslapd-minssf: 128
```



注記

ACI は、「接続に一定レベルのセキュリティーの要求」にあるように、特定タイプの操作に SSF を必要とするように設定できます。

「セキュアなバインドの要求」にあるように、*nsslapd-require-secure-binds* 属性をオンにすることで、バインド操作にセキュアな接続が必要になる場合があります。

9.3. DIRECTORY SERVER が使用する NSS データベースの管理

TLS 暗号化または証明書ベースの認証を設定する場合は、Network Security Services(NSS)に保存されている証明書を管理する必要があります。本セクションでは、Directory Server の NSS データベース管理に関する最も一般的なアクションを説明します。

9.3.1. Directory Server インスタンスの NSS データベースの作成

Directory Server は、証明書を NSS データベース **/etc/dirsrv/slapd-*instance_name*/** ディレクトリーに保存します。証明書を管理する前に、データベースを作成する必要があります。

注記

セキュリティー上の理由から、Red Hat はデータベースの強固なパスワードを設定することを推奨します。

9.3.1.1. コマンドラインを使用した NSS データベースの作成

コマンドラインを使用して NSS データベースを作成するには、以下を実行します。

1. NSS データベースを作成し、パスワードを設定します。

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -N
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
```

2. パーミッションを設定します。

```
# chown dirsrv:dirsrv /etc/dirsrv/slapd-instance_name/*.db
# chown dirsrv:dirsrv /etc/dirsrv/slapd-instance_name/pkcs11.txt
# chmod 600 /etc/dirsrv/slapd-instance_name/*.db
# chmod 600 /etc/dirsrv/slapd-instance_name/pkcs11.txt
```

9.3.1.2. コンソールを使用した NSS データベースの作成

Directory Server コンソールで、Directory Server コンソールで **Manage Certificates** タスクエントリーを初めて開くと、Directory Server は NSS データベースを自動的に作成します。

Manage Certificates タスクエントリーを開くには、以下を実行します。

1. Directory Server コンソールを開きます。
2. **Tasks** タブで **Manage Certificates** をクリックし、データベースを保護するパスワードを設定します。

9.3.2. 証明書署名要求の作成

証明書署名要求 (CSR) は、サーバーの鍵を署名するための認証局 (CA) への要求です。このセクションでは、秘密鍵を含む CSR を作成する方法を説明します。

9.3.2.1. コマンドラインを使用した証明書署名要求の作成

キーおよび CSR を作成するには、certutil ユーティリティーを使用します。

```
# certutil -d instance_directory -R -g key_size -a \
    -o output_file -8 FQDN -s "certificate_subject"
```

例9.1 単一ホスト名の秘密鍵および CSR の作成

以下のコマンドは、server.example.com ホストの 4096 ビット秘密鍵を生成し、CSR を /root/*instance_name*.csr ファイルに保存します。

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -R -g 4096 -a \
    -o /root/instance_name.csr -8 server.example.com \
    -s "CN=server.example.com,O=example_organization,OU=IT,ST=North Carolina,C=US"
```

-8 *server.example.com* オプションは、DNS:server.example.com エントリーを持つサブジェクト代替名(SAN)の拡張を CSR に追加します。*-s* パラメーターで指定した文字列は、RFC 1485 に従って有効なサブジェクト名である必要があります。CN フィールドが必要で、サーバーの完全修飾ドメイン名(FQDN)に設定する必要があります。その他のフィールドは任意です。

例9.2 マルチホームホストの秘密鍵および CSR の作成

Directory Server ホストに複数の名前がある場合は、CSR の SAN 拡張で、すべてのホスト名を持つ CSR を作成します。以下のコマンドは、4096 ビット秘密鍵と、server .example.com および server. example.net ホスト名の CSR を生成します。このコマンドは、CSR を /root/*instance_name*.csr ファイルに保存します。

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -R -g 4096 -a \
    -o /root/instance_name.csr -8 server.example.com,server.example.net \
    -s "CN=server.example.com,O=example_organization,OU=IT,ST=North Carolina,C=US"
```

-8 *server.example.com,server.example.net* オプションは、DNS:server.example.com、DNS:server.example.net エントリーで SAN 拡張を CSR に追加します。*-s* パラメーターで指定した文字列は、RFC 1485 に従って有効なサブジェクト名である必要があります。CN フィールドが必要で、サーバーの FQDN のいずれかに設定する必要があります。その他のフィールドは任意です。

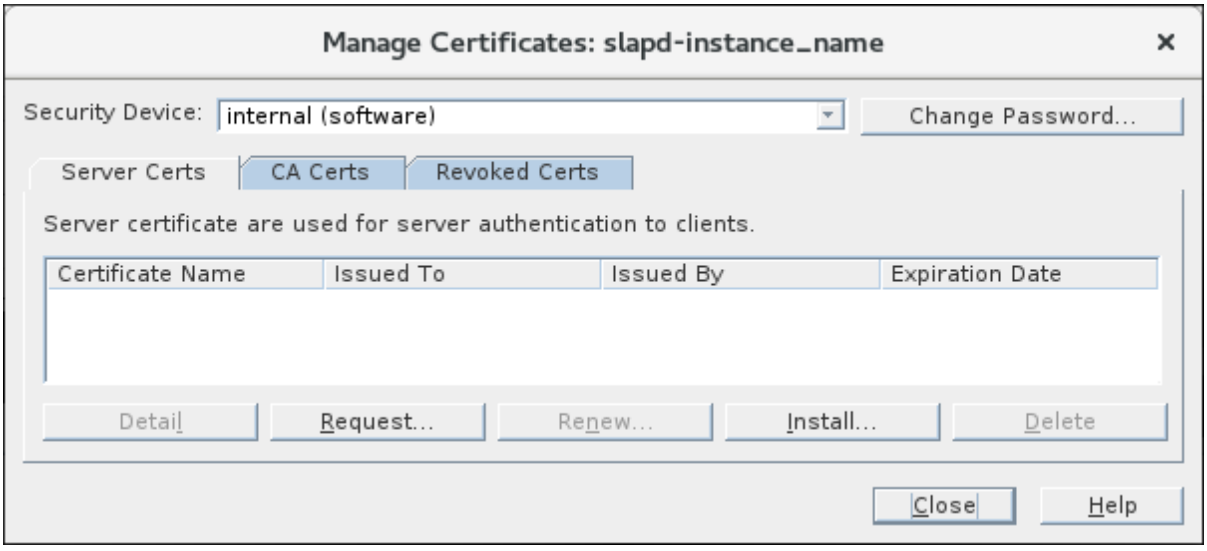
certutil および拡張使用方法の詳細は、certutil(1) の man ページを参照してください。

CSR を生成した後、それを CA に送信し、発行された証明書を取得します。詳細は、CA のドキュメントを参照してください。

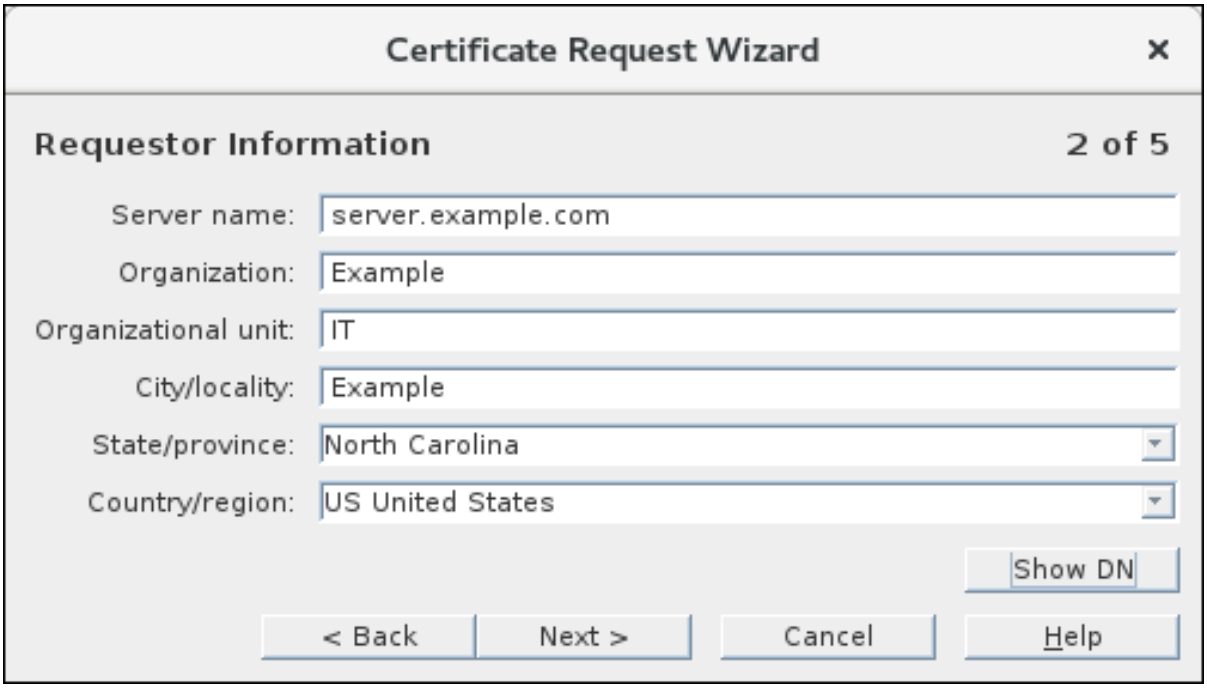
9.3.2.2. コンソールを使用した証明書署名要求の作成

コンソールを使用してキーおよび CSR を作成するには、以下を実行します。

1. **Directory Server コンソールを開きます。**
2. **Tasks タブで、Manage Certificates をクリックします。**
3. **Server Certs タブで、Request ボタンをクリックします。**



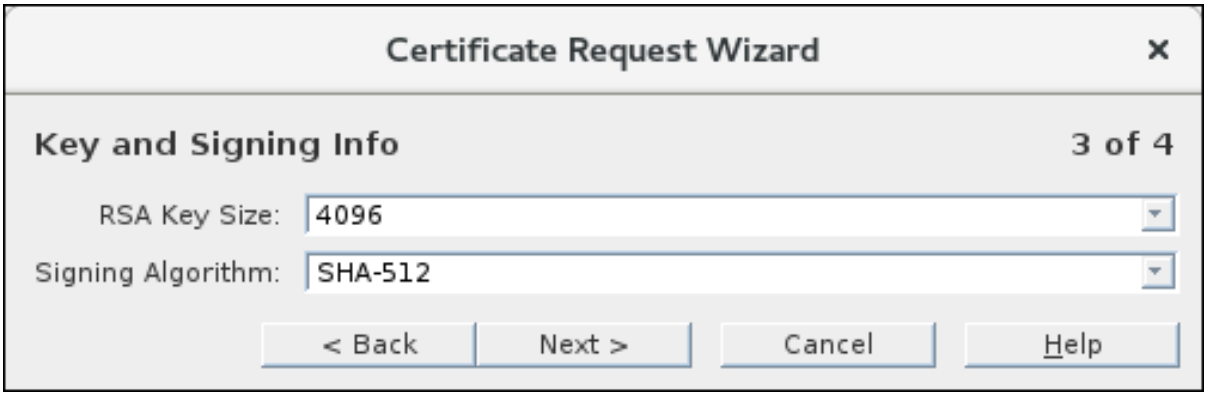
4. **証明書を手動で要求するか、または表示される CA のいずれかから証明書を要求するかどうかを選択し、Next をクリックします。**
5. **要求された情報を入力し、Next をクリックします。**



重要

Server name フィールドにサーバーの完全修飾ドメイン名(FQDN)を入力します。

6. **キーサイズおよび署名アルゴリズムを選択します。Next をクリックします。**



セキュリティー上の理由から、以下のようになります。

- **RSA キーサイズが 2048 ビット以上のもの**
- **SHA-256 以降などの強力な署名アルゴリズム**

7. Network Security Services(NSS)データベースのパスワードを入力し、**Done** をクリックします。

Hardware Security Module(HSM)を使用して証明書を保存する場合、デバイスは接続され、「ハードウェアセキュリティーモジュールの使用」 の説明どおりにモジュールがインストールされていると、モジュールは **Active Encryption Token** メニューで利用できます。

8. CSR をクリップボードにコピーするか、またはファイルに保存します。
9. 完了 をクリックします。

CSR を生成した後に、これを CA に送信し、発行された証明書を取得します。詳細は、CA のドキュメントを参照してください。

9.3.3. CA 証明書のインストール

Directory Server が認証局 (CA) を信頼できるようにするには、CA の証明書を Network Security Services (NSS) データベースにインストールする必要があります。このプロセスでは、CA が発行する証明書を信頼すべてきかどうかを設定する必要があります。

表9.1 CA 信頼オプション

コンソールオプション	certutil オプション	詳細
クライアントからの接続を許可（クライアント認証）	T,,	サーバーは、TLS **EXTERNAL** バインドに適したクライアント証明書を発行するためにこの CA 証明書を信頼します。
他のサーバーへの接続の許可（サーバー認証）	C,,	サーバーは、レプリケーションパートナーへの暗号化された接続を確立するために使用する証明書を検証し、信頼できる CA により発行されていることを確認します。

CA に両方のオプションを設定できます。certutil を使用する場合は、*-T "CT,,"* パラメーターをユーティリティーに渡します。

9.3.3.1. コマンドラインを使用した CA 証明書のインストール

Directory Server の NSS データベースに CA 証明書をインストールするには、**certutil** ユーティリティーを使用します。たとえば、**/etc/pki/CA/nss/ca.crt** ファイルに保存されている CA 証明書をインポートするには、以下を実行します。

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -A -n "certificate_nickname" \
    -t "C,," -i /etc/pki/CA/nss/ca.crt
```

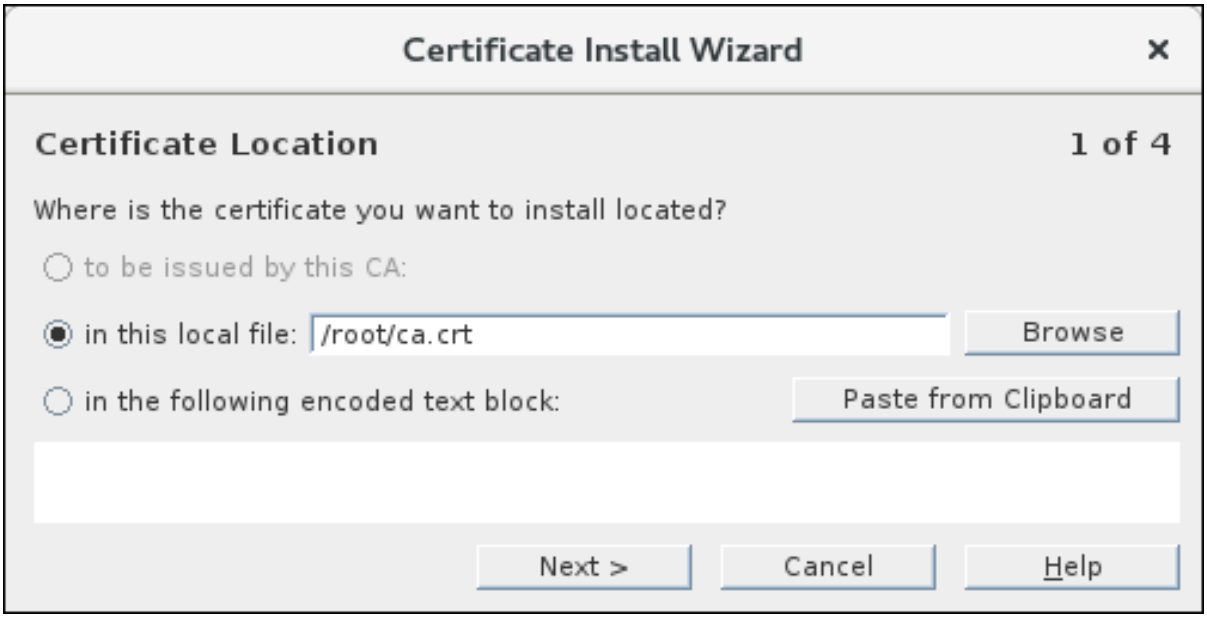
-t trust_options パラメーターは、CA が発行する証明書を信頼する証明書を設定します。表9.1「CA 信頼オプション」を参照してください。

上記のコマンドで使用したパラメーターの詳細は、certutil(1) の man ページを参照してください。

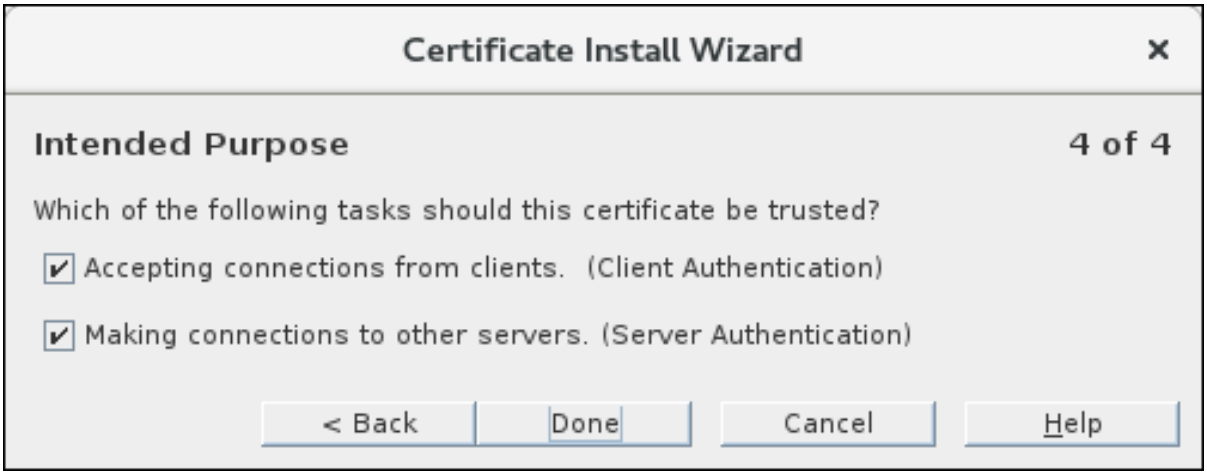
9.3.3.2. コンソールを使用した CA 証明書のインストール

Directory Server コンソールを使用して CA 証明書をインストールするには、以下を実行します。

1. Directory Server コンソールを開きます。
2. **Tasks** タブで、**Manage Certificates** をクリックします。
3. **CA Certs** タブを選択し、**Install** ボタンをクリックします。
4. サーバー証明書が含まれるファイルを選択するか、またはフィールドに証明書を貼り付けます。**Next** をクリックします。



5. 証明書の詳細を確認し、**Next** をクリックします。
6. 証明書のニックネームを確認し、**Next** をクリックします。
7. CA が発行する証明書を信頼する証明書を設定します。オプションのいずれかまたは両方を選択できます。表9.1「CA 信頼オプション」を参照してください。



9.3.4. 証明書のインストール

認証局 (CA) が要求された証明書を発行したら、Network Security Services (NSS) データベースにインストールする必要があります。

9.3.4.1. コマンドラインを使用したサーバー証明書のインストール

Directory Server の NSS データベースにサーバー証明書をインストールするには、certutil ユーティリティーを使用します。以下に例を示します。

1. **CA 証明書をインストールします。「CA 証明書のインストール」を参照してください。**
2. **証明書をインポートします。たとえば、/root/*instance_name*.crt ファイルに保存されている証明書をインポートするには、次のコマンドを実行します。**

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -A \
    -n "server-cert" -t ",," -a -i /root/instance_name.crt
```

3. **必要に応じて、証明書を確認します。**

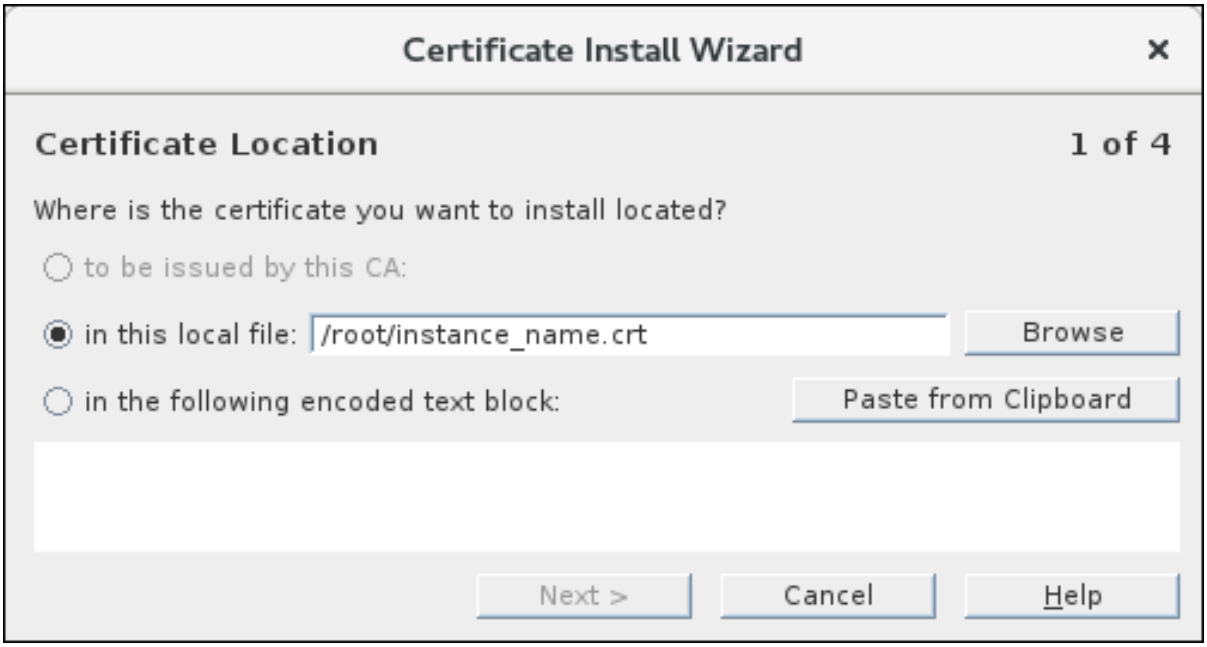
```
# certutil -d /etc/dirsrv/slapd-instance_name/ -V -n "server-cert" -u V
```

以前の certutil コマンドで使用されるパラメーターの詳細は、certutil(1) の man ページを参照してください。

9.3.4.2. コンソールを使用した証明書のインストール

コンソールを使用してサーバー証明書をインストールするには、以下を行います。

1. **CA 証明書をインストールします。「CA 証明書のインストール」を参照してください。**
2. **Directory Server コンソールを開きます。**
3. **Tasks タブで、Manage Certificates をクリックします。**
4. **Install ボタンをクリックします。**
5. **サーバー証明書が含まれるファイルを選択するか、またはフィールドに証明書を貼り付けます。Next をクリックします。**



6. 証明書の詳細を確認し、**Next** をクリックします。
7. 証明書のニックネームを設定し、**Next** をクリックします。

注記

Directory Server コンソールは、既存のニックネームと同じニックネームを使用する証明書のインストールには対応していません。この問題を回避するには、コマンドラインを使用して証明書をインストールします。「コマンドラインを使用したサーバー証明書のインストール」を参照してください。

8. NSS データベースのパスワードを入力し、**Done** をクリックします。

9.3.5. 自己署名証明書の生成およびインストール

特定の状況では、管理者は、Directory Server への暗号化された接続に自己署名証明書を使用します。

注記

この操作は、コマンドラインを使用した場合のみ実行できます。

自己署名証明書を作成してインストールするには、以下を実行します。

1. Network Security Services(NSS)データベースがすでに初期化されているかどうかを確認します。

```
# certutil -d /etc/dirsrv/slapd-instance_name -L
```

コマンドが失敗した場合は、データベースを初期化します。詳細は、「Directory Server インスタンスの NSS データベースの作成」を参照してください。

2. ランダムなデータで関心のあるファイルを生成します。たとえば、サイズが 4096 ビットのあるファイルを生成するには、次のコマンドを実行します。

```
# openssl rand -out /tmp/noise.bin 4096
```

3. 自己署名証明書を作成し、NSS データベースに追加します。

```
# certutil -S -x -d /etc/dirsrv/slapd-instance_name/ -z /tmp/noise.bin \
    -n "server-cert" -s "CN=$HOSTNAME" -t "CT,C,C" -m $RANDOM \
    --keyUsage digitalSignature,nonRepudiation,keyEncipherment,dataEncipherment
```

Red Hat Enterprise Linux は、***$HOSTNAME*** 変数を自動的に完全修飾ドメイン名 (FQDN) に置換え、***$RANDOM*** を無作為に生成した番号に置き換えます。先のコマンドで使用したパラメーターの詳細は、certutil(1) の man ページを参照してください。

4. 必要に応じて、生成された証明書が自己署名されていることを確認します。

```
# certutil -L -d /etc/dirsrv/slapd-instance_name/ -n "server-cert" | egrep "Issuer|Subject"
        Issuer: "CN=server.example.com"
        Subject: "CN=server.example.com"
```

このコマンドの出力には、証明書の発行者とサブジェクトの両方について Directory Server ホストの FQDN が表示されるはずです。

9.3.6. 証明書の更新

証明書がまもなく期限切れになる場合は、セキュアな接続の確立を継続するのに期間で証明書を更新する必要があります。

9.3.6.1. コマンドラインでの証明書の更新

証明書を更新するには、以下を実行します。

1. キーサイズ、ホスト名、サブジェクトなど、同じオプションで新しい証明書署名要求 (CSR) を作成します。CSR の作成に関する詳細は、「コマンドラインを使用した証明書署名要求の作成」を参照してください。

2. CA から発行した証明書を取得したら、同じニックネームを使用してデータベースにインストールします。「コマンドラインを使用した CA 証明書のインストール」を参照してください。

Directory Server は、新たに発行した証明書を自動的に使用します。

9.3.6.2. コンソールを使用した証明書の更新

更新のプロセスは、証明書署名要求(CSR)の生成に似ています。「コンソールを使用した CA 証明書のインストール」 の手順に従いますが、**Manage Certificates** タスクの **Request** ボタンの代わりに **Renew** をクリックします。

9.3.7. 証明書の削除

たとえば、証明書が公開されていないため、証明書が必要なくなった場合は、その証明書をデータベースから削除します。

9.3.7.1. コマンドラインで証明書の削除

コマンドラインで証明書を削除するには、以下を行います。

1. 秘密鍵を削除します。「秘密鍵の削除」 を参照してください。

2. 必要に応じて、データベースの証明書を表示します。

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -L
Certificate Nickname                    Trust Attributes
                                        SSL,S/MIME,JAR/XPI

Example CA                              CT,,
server-cert                             u,u,u
```

3. 証明書を削除します。たとえば、*server-cert* ニックネームで証明書を削除するには、次のコマンドを実行します。

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -D -n "server-cert"
```

9.3.7.2. コンソールを使用した証明書の削除

コンソールを使用して証明書を削除するには、以下を実行します。

1. Directory Server コンソールを開きます。
2. **Tasks** タブで、**Manage Certificates** をクリックします。
3. **Server Certs** タブで証明書を選択し、**Delete** ボタンをクリックします。
4. **Yes** をクリックして確定します。

9.3.8. 秘密鍵の削除

たとえば、強力な鍵を作成したなどのため、秘密鍵が必要なくなった場合は、データベースから削除します。

警告

秘密鍵を削除すると、この鍵に基づく証明書は機能しなくなります。

9.3.8.1. コマンドラインでの秘密鍵の削除

秘密鍵を削除するには、次を実行します。

1. 削除する鍵に基づいてすべての証明書を削除します。「証明書の削除」を参照してください。
2. 必要に応じて、データベースのキーを表示します。

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -K
certutil: Checking token "NSS Certificate DB" in slot "NSS User Private Key and Certificate
Services"
Enter Password or Pin for "NSS Certificate DB":
< 0> rsa      7a2fb6c269d83c4036eac7e4edb6aaf2ed08bc4a   server-cert
< 1> rsa      662b826aa3dd4ca7fd7e6883558cf3866c42f4e2   example-cert
```

3. 秘密鍵を削除します。たとえば、*example-cert* ニックネームで秘密鍵を削除するには、次を実行します。

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -F -n "example-cert"
```

9.3.8.2. コンソールを使用した秘密鍵の削除

コンソールを使用した秘密鍵の削除はサポートされていません。ただし、「コンソールを使用した証明書署名要求の作成」 に従ってコンソールを使用して新しい証明書を要求すると、コンソールは新しい秘密鍵を自動的に生成し、これを使用します。

9.3.9. CA 信頼オプションの変更

特定の状況では、認証局 (CA) の trust オプションを更新する必要があります。本セクションでは、この手順を説明します。

9.3.9.1. コマンドラインを使用した CA 信頼オプションの変更

CA の信頼オプションを変更するには、*-t* パラメーターの新しいオプションを certutil ユーティリティーに渡します。

たとえば、Directory Server が **example-CA** という名前の CA が発行するクライアント証明書のみを信頼するように設定するには、以下を実行します。

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -M -t "T,," -n "example-CA"
```

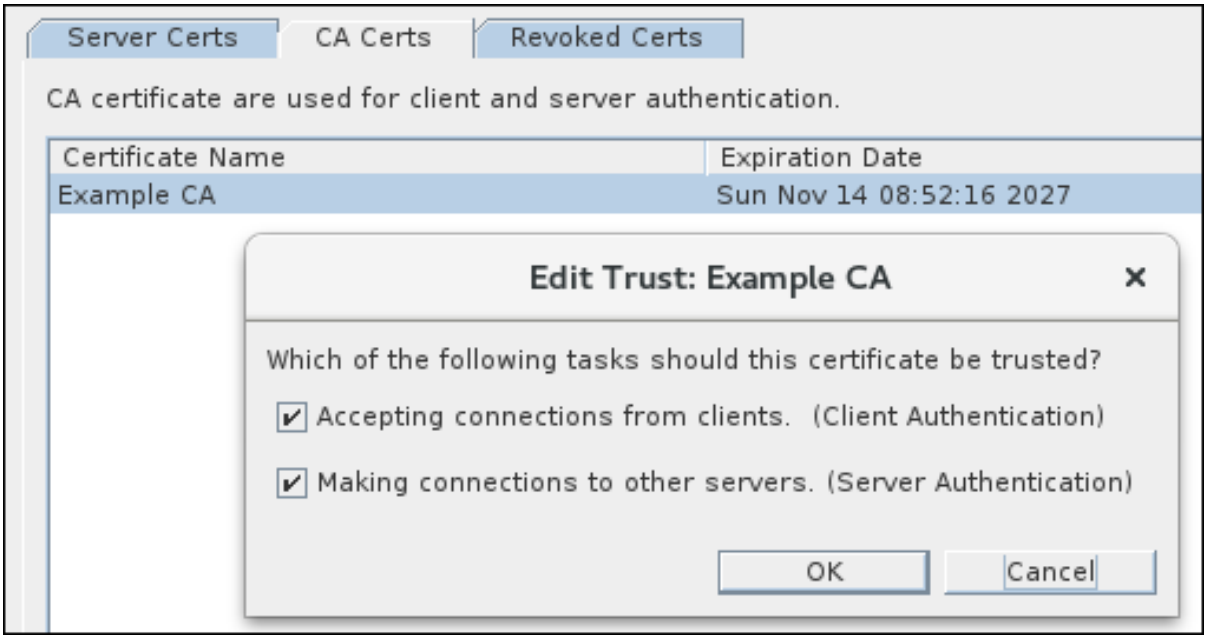
-t trust_options パラメーターは、CA が発行する証明書を信頼する証明書を設定します。表9.1「CA 信頼オプション」を参照してください。

パラメーターおよび信頼オプションの詳細は、certutil(1) の man ページを参照してください。

9.3.9.2. コンソールを使用した CA 信頼オプションの変更

コンソールを使用して CA の信頼オプションを変更するには、以下を実行します。

1. Directory Server コンソールを開きます。
2. **Tasks** タブで、**Manage Certificates** をクリックします。
3. **CA Certs** タブを選択します。
4. 編集する CA を選択し、**Edit Trust** ボタンをクリックし、CA が発行する証明書を信頼する証明書を設定します。オプションのいずれかまたは両方を選択できます。表9.1「CA 信頼オプション」を参照してください。



9.3.10. NSS データベースのパスワードの変更

特定の状況では、管理者が Network Security Services (NSS) データベースのパスワードを変更します。本セクションでは、この手順を説明します。

重要

パスワードファイルを使用して Directory Server が Network Security Services (NSS) データベースを自動的に開くようにするには、新しいパスワードの設定後にファイルを更新する必要があります。「Directory Server のパスワードファイルの作成」を参照してください。

9.3.10.1. コマンドラインを使用した NSS データベースのパスワードの変更

NSS データベースのパスワードを変更するには、次を実行します。

```
# certutil -d /etc/dirsrv/slapd-instance_name -W
Enter Password or Pin for "NSS Certificate DB":
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
Password changed successfully.
```

9.3.10.2. コンソールを使用した NSS データベースのパスワードの変更

コンソールを使用して NSS データベースのパスワードを変更するには、以下を実行します。

1. **Directory Server コンソールを開きます。**
2. **Tasks タブで、Manage Certificates をクリックします。**
3. **パスワードの変更 ボタンをクリックします。**

4. 現在のパスワードを入力し、OKをクリックします。

9.3.11. 証明書失効リストの追加

認証局(CA)が証明書を取り消すと、CA は証明書を証明書失効リスト(CRL)に追加します。Directory Server は、この一覧を使用して、CA によって信頼されなくなった証明書を識別し、アクセスを拒否することができます。

9.3.11.1. コマンドラインを使用した証明書失効リストの追加

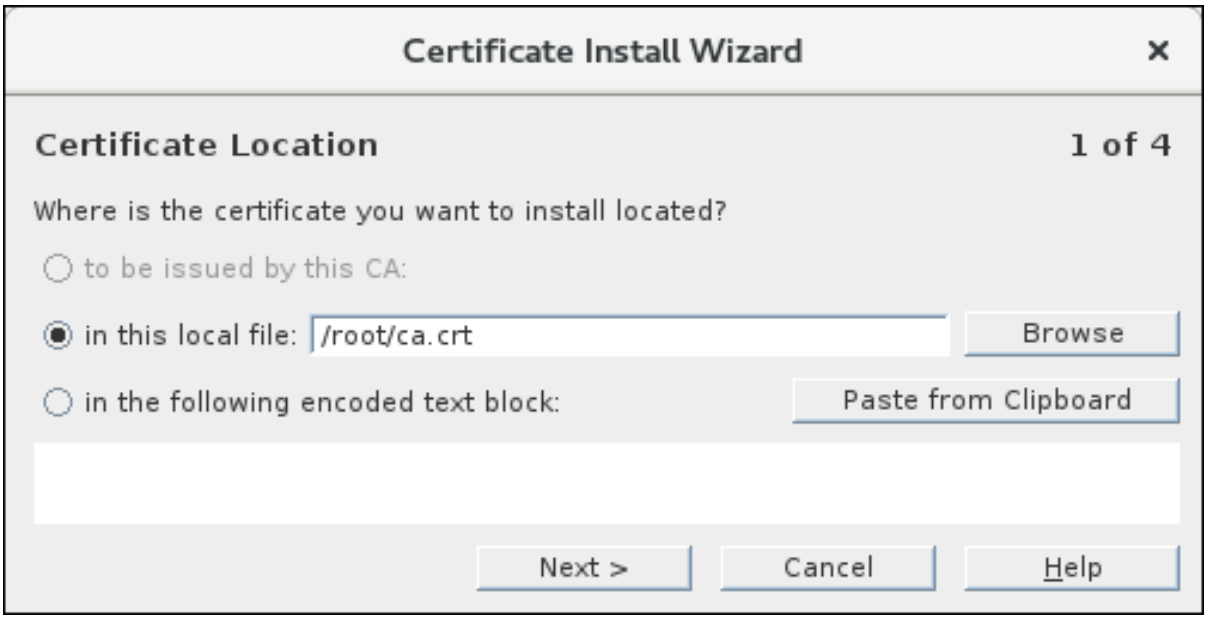
certutil を使用して CRL を追加するには、CA 証明書のインストール時に *-4 URL_to_CRL_file* パラメーターをユーティリティーに渡します。

CA 証明書のインストールの詳細は、「コマンドラインを使用した CA 証明書のインストール」 を参照してください。

9.3.11.2. コンソールを使用した証明書失効リストの追加

コンソールを使用して CRL を追加するには、以下を実行します。

1. Directory Server コンソールを開きます。
2. **Tasks** タブで、**Manage Certificates** をクリックします。
3. **Revoked Certs** タブを選択し、**Add** ボタンをクリックします。
4. ファイルへのパスを入力し、リストの形式を選択して **OK** をクリックします。



9.4. TLS の有効化

Directory Server は、クライアントとサーバーとの間で暗号化された接続と、レプリケーション環境でののサーバー間の暗号化をサポートします。このため、Directory Server は以下に対応します。

- LDAPS プロトコル: TLS 暗号化は接続が確立された後に直接使用されます。
- LDAP プロトコルの **STARTTLS** コマンド: 接続は、クライアントが **STARTTLS** コマンドを送信するまで暗号化されません。

重要

セキュリティー上の理由から、Red Hat は TLS 暗号化を有効にすることを推奨します。

バインド識別名 (DN) およびパスワード、または証明書ベースの認証を使用して、簡易認証で TLS を使用できます。

Directory Server の暗号化サービスは、Mozilla Network Security Services (NSS) (TLS およびベース暗号化機能のライブラリー) によって提供されます。NSS には、連邦情報処理標準 (FIPS) 140-2 認定であるソフトウェアベースの暗号化トークンが含まれています。

9.4.1. Directory Server での TLS の有効化

本セクションでは、Directory Server で TLS を有効にする方法を説明します。

9.4.1.1. コマンドラインを使用した Directory Server での TLS の有効化

コマンドラインで TLS を有効にするには、以下を実行します。

1. Directory Server の NSS データベースがすでに存在しているかどうかを確認します。

```
# ls -1 /etc/dirsrv/slapd-instance_name/*.db
```

データベースが存在しない場合は作成します。「コマンドラインを使用した NSS データベースの作成」 を参照してください。

2. 証明書を要求してインストールします。
   - 認証局 (CA) が発行する証明書の場合:
     1. Certificate Signing Request (CSR) を生成します。「コマンドラインを使用した証明書署名要求の作成」を参照してください。
     2. CA 証明書をインポートします。「コマンドラインを使用した CA 証明書のインストール」を参照してください。
     3. CA が発行するサーバー証明書をインポートします。「コマンドラインを使用したサーバー証明書のインストール」を参照してください。
   - 自己署名証明書は、「自己署名証明書の生成およびインストール」を参照してください。
3. TLS を有効にし、LDAPS ポートを設定します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=config
changetype: modify
replace: nsslapd-securePort
```

```
nsslapd-securePort: 636
-
replace: nsslapd-security
nsslapd-security: on
```

4. NSS データベースのサーバー証明書のニックネームを表示します。

```
# certutil -L -d /etc/dirsrv/slapd-instance_name/
Certificate Nickname                 Trust Attributes
                                     SSL,S/MIME,JAR/XPI

Example CA                           CT,,
server-cert                          u,u,u
```

次の手順でニックネームが必要です。

5. RSA 暗号ファミリーを有効にするには、NSS データベースセキュリティーデバイスおよびサーバー証明書のニックネームを設定し、次のエントリーをディレクトリーに追加します。

```
# ldapadd -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=RSA,cn=encryption,cn=config
cn: RSA
objectClass: top
objectClass: nsEncryptionModule
nsSSLToken: internal (software)
nsSSLPersonalitySSL: server-cert
nsSSLActivation: on
```

注記

デフォルトでは、NSS データベース内のセキュリティーデバイスの名前は **internal (software)** です。

cn=RSA,cn=encryption,cn=config エントリーがすでに存在しているため、上記のコマンドが失敗する場合は、対応する属性を更新します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=RSA,cn=encryption,cn=config
changetype: modify
replace: nsSSLToken
nsSSLToken: internal (software)
-
replace: nsSSLPersonalitySSL
nsSSLPersonalitySSL: server-cert
-
replace: nsSSLActivation
nsSSLActivation: on
```

6. 必要に応じて、Directory Server がサポートする暗号化の一覧を更新します。詳細は「コマンドラインを使用した Directory Server が使用する暗号の表示および設定」を参照してください。

7. 必要に応じて、証明書ベースの認証を有効にします。詳細は「証明書ベースのクライアント認証の使用」を参照してください。

8. 必要に応じて、パスワードファイルを作成して、NSS データベースのパスワードを要求せずに Directory Server が起動するようにします。詳細は「Directory Server のパスワードファイルの作成」を参照してください。

9. Directory Server インスタンスを再起動します。

```
# systemctl restart dirsrv@instance_name
```

NSS データベースにパスワードを設定し、パスワードファイルを作成しないと、Directory Server は NSS データベースのパスワードを要求します。詳細は「パスワードファイルなしで Directory Server の起動」を参照してください。

10. 必要に応じて、サーバーへの接続時に Directory Server Console が TLS を使用するようにします。「コマンドラインを使用したコンソールから Directory Server への接続に対する TLS の有効化」 を参照してください。

11. 必要に応じて、Red Hat Identity Management Consoleが TLS を使用するように TLS を有効にします。「管理サーバーでの TLS の有効化」 を参照してください。

9.4.1.2. コンソールを使用した Directory Server での TLS の有効化

コンソールを使用して Directory Server で TLS を有効にするには、以下を実行します。

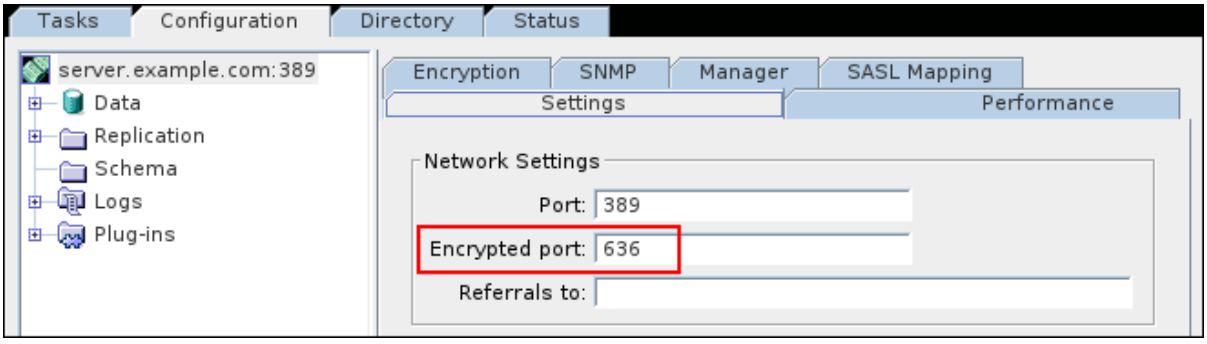
1. CSR を作成します。「コンソールを使用した証明書署名要求の作成」 を参照してください。

2. 認証局 (CA) 証明書をインポートします。「コンソールを使用した CA 証明書のインストール」を参照してください。

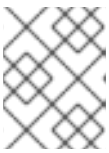
3. CA が発行するサーバー証明書をインポートします。「コンソールを使用した証明書のインストール」 を参照してください。

4. Directory Server コンソールを開き、**Configuration** タブでホスト名を選択します。

5. 右側のペインの **Settings** タブで LDAPS ポートを **Encrypted port** フィールドに入力し、**Save** ボタンをクリックします。



LDAPS のデフォルトポートは 636 です。



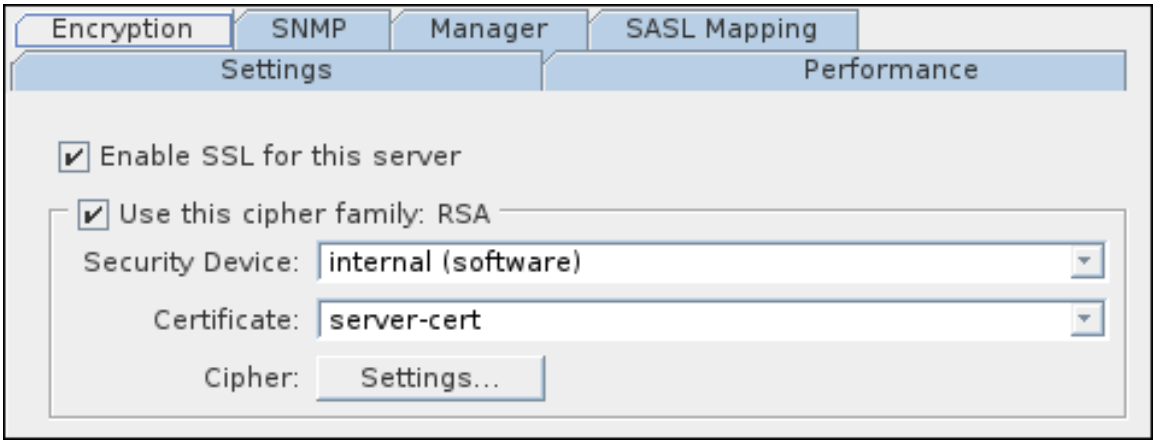
注記

LDAPS ポートは、ポート フィールドの 暗号化されていない接続に設定されたセットとは異なる必要があります。

6. 右側のペインの **Encryption** タブで、以下を実行します。

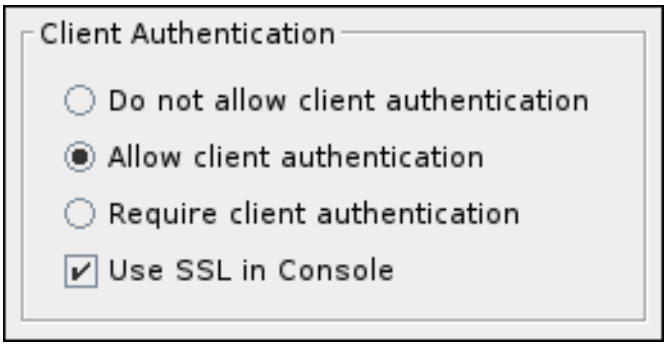
a. このサーバーの **Enable SSL** を選択します。

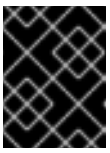
b. **Use this cipher family: RSA。** 一覧からセキュリティーデバイスおよび証明書を選択します。



c. 必要に応じて、**Settings** ボタンをクリックして、Directory Server がサポートする暗号化の一覧を更新します。詳細は、「コンソールを使用した Directory Server が使用する暗号の表示および設定」 を参照してください。

d. 必要に応じて、ユーザーが証明書を使用して認証できるようにします。詳細は「証明書ベースのクライアント認証の使用」を参照してください。

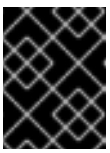




重要

TLS が Directory Server でのみ有効で、Directory Server コンソールではない場合は、**Require client authentication** を選択しないでください。

e. アウトバウンド SSL 接続オプションに対して証明書の名前に対して **Check host name** を選択し、認証用にクライアントが提示する証明書のサブジェクト名の *cn* 属性と一致することを確認します。



重要

Red Hat は、中間者攻撃(MITM)に対して発信 TLS 接続を保護するために、レプリケーション環境でこのオプションを有効にすることを推奨します。

f. **Use SSL in Console** オプションが選択されていないことを確認します。

警告

この手順を終了する前に、Use SSL in Console オプションを有効にしないでください。設定を保存すると直ちに反映されるためです。これにより、コンソールがサーバーへの接続に失敗します。

このオプションを誤って有効にし、コンソールがサーバーへの接続に失敗した場合には、コマンドラインを使用してオプションを無効にします。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h
server.example.com -x
dn: cn=slapd-instance_name,cn=Red Hat Directory Server,
 cn=Server
Group,cn=server.example.com,ou=example.com,o=NetscapeRoot
changetype: modify
replace: nsServerSecurity
nsServerSecurity: off
```

g. **Save をクリックします。**

7. **必要に応じて、パスワードファイルを作成して、NSS データベースのパスワードを要求せずに Directory Server が起動するようにします。詳細は「Directory Server のパスワードファイルの作成」を参照してください。**

8. **Directory Server インスタンスを再起動します。**

```
# systemctl restart dirsrv@instance_name
```

NSS データベースにパスワードを設定し、パスワードファイルを作成しないと、Directory Server は NSS データベースのパスワードを要求します。詳細は「パスワードファイルなしで Directory Server の起動」を参照してください。

9. **必要に応じて、サーバーへの接続時に Directory Server Console が TLS を使用するようにします。「コンソールを使用したコンソールから Directory Server への接続に対する TLS の有効化」 を参照してください。**

10. **必要に応じて、Red Hat Identity Management Consoleが TLS を使用するようにします。「管理サーバーでの TLS の有効化」 を参照してください。**

9.4.1.3. 暗号化暗号の設定

Directory Server は異なる暗号に対応し、その暗号化を有効または無効にできます。暗号化は、暗号化で使用されるアルゴリズムです。クライアントがサーバーとの TLS 接続を開始すると、クライアントは情報の暗号化を好む暗号をサーバーに指示します。サーバーがこれらの暗号のいずれかに対応する場合は、このアルゴリズムを使用して暗号化された接続を確立できます。

「TLS の有効化」に従って暗号化を有効にすると、Directory Server が使用する暗号化を表示および更新できます。

9.4.1.3.1. コマンドラインを使用した Directory Server が使用する暗号の表示および設定

利用可能なすべての暗号の表示
Directory Server で対応可能なすべての暗号の一覧を表示するには、次のコマンドを実行します。

```
# ldapsearch -xLLL -H ldap://server.example.com:389 -D "cn=Directory Manager" -W \
    -b 'cn=encryption,cn=config' -s base nsSSLSupportedCiphers -o ldif-wrap=no

dn: cn=encryption,cn=config
nsSSLSupportedCiphers: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256::AES-
GCM::AEAD::128
...
nsSSLSupportedCiphers: SSL_CK_RC2_128_CBC_EXPORT40_WITH_MD5::RC2::MD5::128
```

これは、有効または無効にできる暗号だけ含まれる一覧になります。この一覧には、Directory Server が現在使用している暗号は表示されません。

使用する暗号ディレクトリーサーバーの表示
Directory Server が現在使用中の暗号は、*nsSSLEnabledCiphers* の読み取り専用属性に保存されます。それらを表示するには、次のコマンドを実行します。

```
# ldapsearch -xLLL -H ldap://server.example.com:389 -D "cn=Directory Manager" -W \
    -b 'cn=encryption,cn=config' -s base nsSSLEnabledCiphers -o ldif-wrap=no

dn: cn=encryption,cn=config
nsSSLEnabledCiphers: TLS_RSA_WITH_AES_256_CBC_SHA::AES::SHA1::256
nsSSLEnabledCiphers: TLS_RSA_WITH_AES_128_CBC_SHA::AES::SHA1::128
...
```

さらに、有効/無効にするように設定された暗号を表示できます。

```
# ldapsearch -xLLL -H ldap://server.example.com:389 -D "cn=Directory Manager" -W \
    -b 'cn=encryption,cn=config' -s base nsSSL3Ciphers -o ldif-wrap=no

dn: cn=encryption,cn=config
nsSSL3Ciphers: -all,+tls_rsa_aes_128_sha,+tls_rsa_aes_256_sha,...
```

重要

Directory Server は、*nsSSL3Ciphers* 属性からの設定を使用して、実際に使用されている暗号の一覧を生成します。ただし、*nsSSL3Ciphers* で弱い暗号化を有効にし、*allowWeakCiphers* パラメーターをデフォルトの off に設定した場合、Directory Server は強力な暗号化のみを使用し、*nsSSLSupportedCiphers* 読み取り専用属性に表示します。

有効な暗号リストの更新
有効な暗号の一覧を更新するには、次のコマンドを実行します。

1. 現在有効な暗号の一覧を表示します。「使用する暗号ディレクトリーサーバーの表示」を参照してください。

2. 特定の暗号のみを有効にするには、*nsSSL3Ciphers* 属性を更新します。たとえば、TLS_RSA_WITH_AES_128_GCM_SHA256 暗号のみを有効にするには、次のコマンドを実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=encryption,cn=config
changetype: modify
add: nsSSL3Ciphers
nsSSL3Ciphers: -all,+TLS_RSA_WITH_AES_128_GCM_SHA256
```

3. Directory Server インスタンスを再起動します。

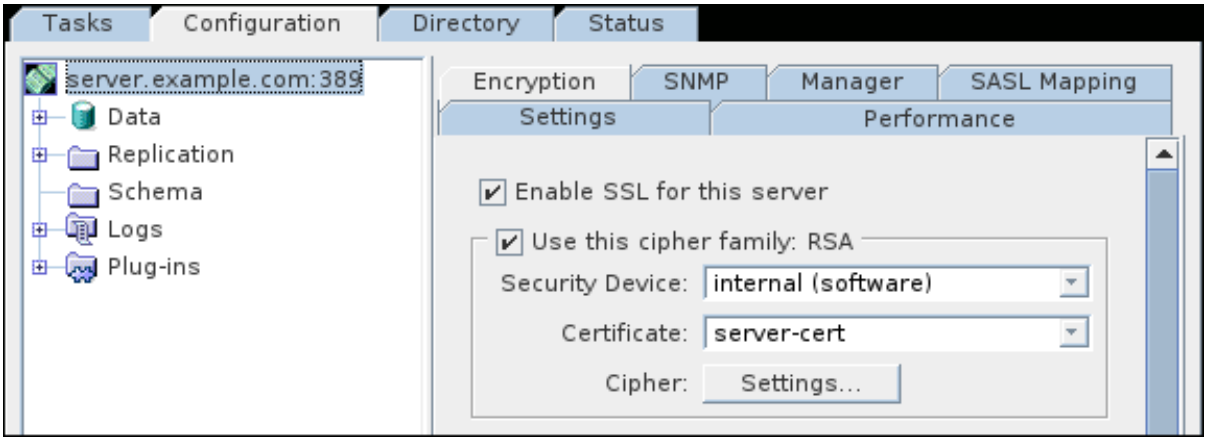
```
# systemctl restart dirsrv@instance_name
```

4. 必要に応じて、有効な暗号の一覧を表示して、結果を確認します。「使用する暗号ディレクトリーサーバーの表示」を参照してください。

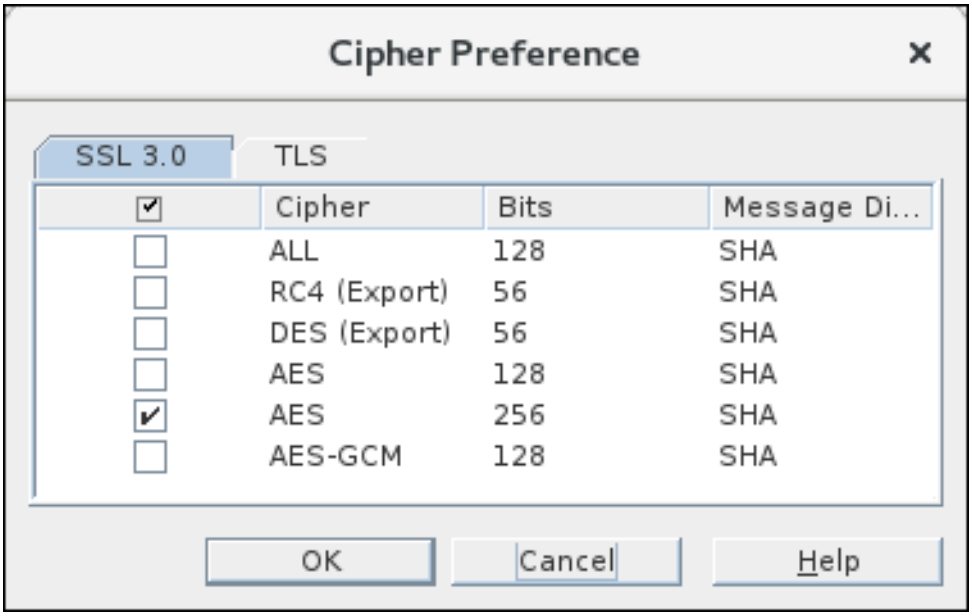
9.4.1.3.2. コンソールを使用した Directory Server が使用する暗号の表示および設定

コンソールを使用して暗号を選択し、必要に応じて更新するには、以下を実行します。

1. Directory Server コンソールを開きます。
2. **Configuration** タブでサーバー名を選択します。
3. 右側のペインで **Encryption** タブを選択し、**Settings** ボタンをクリックします。



4. 必要に応じて、暗号の一覧を更新します。以下に例を示します。



5. **OK** をクリックします。
6. **Save** をクリックします。
7. 暗号の一覧を更新した場合は、Directory Server インスタンスを再起動します。

```
# systemctl restart dirsrv@instance_name
```

9.4.1.4. パスワードファイルなしで Directory Server の起動

暗号化を有効にし、NSS データベースに設定したパスワードを使用して Directory Server を起動する場合は、以下を行います。

- **systemctl** コマンドで **ns-slapd** Directory Server プロセスが起動すると、**systemd** はパスワードを求めるプロンプトを表示し、その入力内容を **systemd-tty-ask-password-agent** ユーティリティーに自動的に渡します。以下に例を示します。

```
# systemctl start dirsrv
Enter PIN for Internal (Software) Token:
```

- まれに、**ns-slapd** Directory Server プロセスが **systemctl** ユーティリティーにより開始されず、ターミナルから切り離されていると、**wall** コマンドを使用してすべての端末にメッセージを送信します。以下に例を示します。

```
Broadcast message from root@server (Fri 2017-01-01 06:00:00 CET):

Password entry required for 'Enter PIN for Internal (Software) Token:' (PID 1234).
Please enter password with the systemd-tty-ask-password-agent tool!
```

パスワードを入力するには、次を実行します。

```
# systemd-tty-ask-password-agent
Enter PIN for Internal (Software) Token:
```

9.4.1.5. Directory Server のパスワードファイルの作成

暗号化が有効で、NSS データベースに設定したパスワードがあると、サービスの起動時に Directory Server はこのパスワードを要求します。「パスワードファイルなしで Directory Server の起動」を参照してください。

このプロンプトを省略するには、NSS データベースパスワードを /etc/dirsrv/slapd-*instance_name*/pin.txt ファイルに保存できます。これにより、このパスワードを要求せずに Directory Server が自動的に起動できます。

警告

パスワードはクリアテキストで保存されます。サーバーがセキュアでない環境で実行している場合は、パスワードファイルを使用しないでください。

パスワードファイルを作成するには、以下を実行します。

1. 以下の内容で /etc/dirsrv/slapd-*instance_name*/pin.txt ファイルを作成します。
   - NSS ソフトウェア暗号モジュールを使用する場合は、以下になります。

     ```
     Internal (Software) Token:password
     ```

   - Hardware Security Module (HSM) を使用する場合:

     ```
     name_of_the_token:password
     ```

2. パーミッションを設定します。

```
# chown dirsrv:dirsrv /etc/dirsrv/slapd-instance_name/pin.txt
# chmod 400 /etc/dirsrv/slapd-instance_name/pin.txt
```

9.4.1.6. 証明書の有効期限が切れた場合の Directory Server の動作の管理方法

デフォルトでは、暗号化が有効で、証明書の有効期限が切れると、Directory Server は警告をログに記録し、サービスを起動します。この動作を変更するには、cn=config エントリーの ***nsslapd-validate-cert*** 属性を設定します。以下の値を設定できます。

- **warn**: Directory Server インスタンスが起動し、期限切れの証明書に関する警告を /var/log/dirsrv/slapd-*instance_name*/error ログファイルに記録します。これはデフォルト設定です。
- **on**: Directory Server は証明書を検証し、証明書の有効期限が切れると、インスタンスの起動に失敗します。
- **off**: Directory Server は証明書の有効期限を検証しません。インスタンスが起動し、警告は記録されません。

例9.3 証明書の有効期限が切れると Directory Server が起動しないようにする

証明書の有効期限が切れている場合は Directory Server が起動しないようにするには、次を実行します。

1. ***nsslapd-validate-cert*** 属性を **on** に設定します。

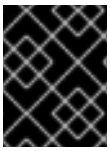
```
# ldapmodify -D "cn=Directory Manager" -W -p 636 -h server.example.com -x
dn: cn=config
changetype: modify
replace: nsslapd-validate-cert
nsslapd-validate-cert: on
```

2. Directory Server インスタンスを再起動します。

```
# systemctl restart dirsrv@instance_name
```

9.4.2. コンソールから Directory Server への接続に TLS を有効化

本セクションでは、TLS を使用してディレクトリーにアクセスするために Directory Server コンソールを設定する方法を説明します。



重要

コンソールで TLS を有効にする前に、「Directory Server での TLS の有効化」に従って Directory Server で暗号化を有効にし、インスタンスを再起動します。

Red Hat Identity Management コンソールへの暗号化された接続を設定するには、「管理サーバーでの TLS の有効化」を参照してください。

9.4.2.1. コマンドラインを使用したコンソールから Directory Server への接続に対する TLS の有効化

コンソールから Directory Server への接続に TLS を有効にするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 636 -h server.example.com -x
dn: cn=slapd-instance_name,cn=Red Hat Directory Server,
 cn=Server Group,cn=server.example.com,ou=example.com,o=NetscapeRoot
changetype: modify
replace: nsServerSecurity
nsServerSecurity: on
```

次回コンソールを開始すると、Directory Server への接続に TLS を自動的に使用します。

9.4.2.2. コンソールを使用したコンソールから Directory Server への接続に対する TLS の有効化

コンソールから Directory Server への接続に TLS を有効にするには、以下を実行します。

1. Directory Server コンソールを開き、**Configuration** タブでホスト名を選択します。
2. 右側のペインの **Encryption** タブで、以下を実行します。
   a. コンソールで **SSL の使用** を選択します。

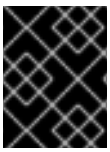
b. **Save**をクリックします。

3. Directory Server コンソールを再起動します。

9.4.3. 管理サーバーでの TLS の有効化

本セクションでは、以下を行う方法を説明します。

- Red Hat Identity Management コンソールアプリケーションへの接続時の HTTPS プロトコルの有効化
- Administration Server が、Directory Server への暗号化された接続 を使用してデータを **o=NetscapeRoot** エントリーに保存するように設定します。
- Red Hat Identity Management Consoleアプリケーションを有効にして、LDAPS プロトコルを使用して、ディレクトリーに保存されているユーザーおよびグループを管理します。



重要

この機能を有効にする前に、「Directory Server での TLS の有効化」の説明に従って Directory Server で暗号化を有効にし、インスタンスを再起動します。

管理サーバーで TLS を有効にするには、以下を実行します。

1. 必要な証明書をインポートします。以下のいずれかの方法で選択します。
   - Directory Server と同じ秘密鍵と証明書を使用するには、「管理サーバーの Directory Server プライベートキーおよび証明書の使用」 を参照してください。
   - 管理サーバーに別の鍵と証明書を使用するには、以下を参照してください。
     1. 「証明書署名要求の作成」
     2. 「CA 証明書のインストール」
     3. 「証明書のインストール」

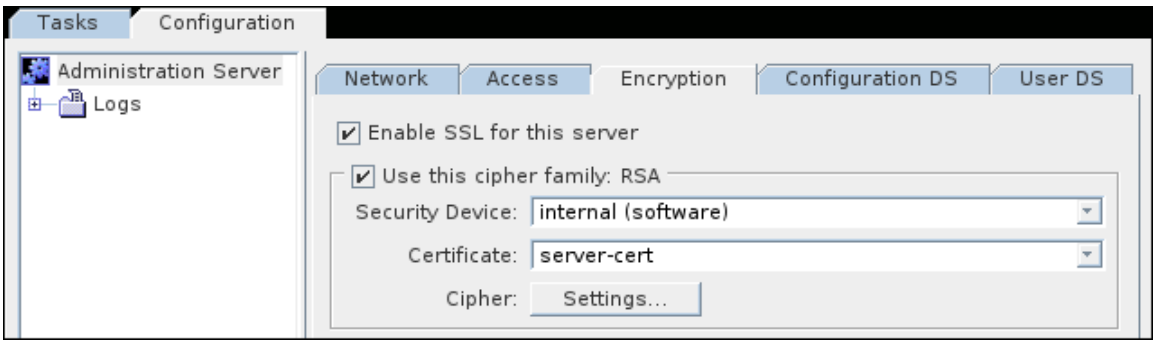
     

     重要

     Directory Server コンソールではなく、管理コンソールの **Manage Certificates** メニューの手順を実行します。

   管理サーバーと Directory Server は、他の共有証明書を信頼するために、少なくとも 1 つの CA 証明書を共有する必要があります。
2. 管理コンソールを開きます。
3. **Configuration** タブで、左側のペインで **Administration Server** エントリーを選択します。
4. 右側のペインで **Encryption** タブを選択して、Red Hat Identity Management Consoleの暗号化を有効にします。
   a. このサーバーの **Enable SSL** を選択します。

b. **Use this cipher family: RSA。** 一覧からセキュリティーデバイスおよび証明書を選択します。



c. **必要に応じて、Settings ボタンをクリックして、Administration Server がサポートする暗号の一覧を更新します。**

d. **必要に応じて、証明書を使用してクライアント認証を有効にします。詳細は「証明書ベースのクライアント認証の使用」を参照してください。**

e. **Save をクリックします。**

5. **右側のペインで Configuration DS タブを選択して、LDAP プロトコルを使用して Administration Server が o=NetscapeRoot エントリーにデータを格納するように設定します。**

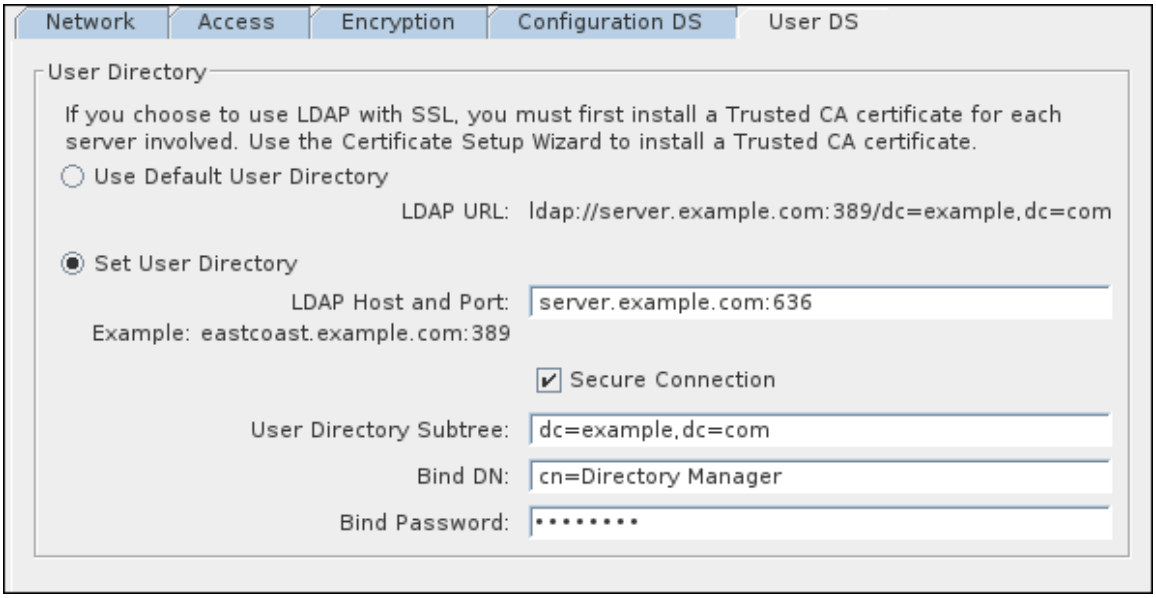
a. **o=NetscapeRoot エントリーを保存する Directory Server インスタンスの LDAPS ポートを設定します。デフォルトでは、LDAPS は 636 ポートを使用します。**

b. **Secure Connection を選択します。**

c. **Save をクリックします。**

6. **右側のペインで ユーザー DS タブを選択して、Red Hat Identity Management Consoleが暗号化された接続を使用してユーザーおよびグループを管理するように設定します。**

a. **Set User Directory を選択し、フィールドに入力します。暗号化された接続では、Secure Connections オプションが選択され、LDAP Host および Port フィールドで指定されたポートポートは LDAPS をサポートする必要があります。**



b. **Save** をクリックします。

7. 必要に応じて、コンソールから ~/.redhat-idm-console/Console. version.Login.preferences ファイルの接続の最小および最大の TLS*バージョン* を設定します。以下に例を示します。

```
sslVersionMin: TLS1.1
sslVersionMax: TLS1.2
```

8. 必要に応じて、パスワードファイルを作成し、Network Security Services(NSS)データベースのパスワードを要求せずに管理サーバーが起動するようにします。詳細は、「管理サーバーのパスワードファイルの作成」 を参照してください。

9. 管理サーバーを再起動します。

```
# systemctl restart dirsrv-admin
```

パスワードファイルを作成していない場合、システムは NSS データベースのパスワードを要求します。

10. コンソールが証明書を信頼するように設定するには、「Directory Server コンソールが使用する証明書の管理」 を参照してください。

この手順を完了したら、HTTPS プロトコルを使用して Red Hat Identity Management Consoleに接続できます。以下に例を示します。

```
# redhat-idm-console -a https://server.example.com:9830
```

9.4.3.1. Directory Server コンソールが使用する証明書の管理

サーバーが使用する証明書およびキーは、/etc/dirsrv/slapd-*instance_name*/ ディレクトリーの NSS セキュリティーデータベースに保存されます。Directory Server コンソール自体は、TLS 接続に証明書と鍵も使用します。これらの証明書は、ユーザーのホームディレクトリーにある別のデータベースに保存されます。Directory Server Console を使用して TLS 経由で Directory Server の複数のインスタンスに接続する場合は、すべての Directory Server インスタンスで証明書を発行したすべての CA を信頼する必要があります。

Directory Server Console に対して TLS が有効になっている場合、Directory Server Console には、サーバーのクライアント証明書を信頼するために、発行する CA 証明書のコピーが必要です。それ以外の場合は、コンソールは証明書を発行した CA を信頼しないエラーを返します。

注記

サーバーの証明書を発行した CA の CA 証明書のみが必要です。Directory Server コンソールには、独自のクライアント証明書は必要ありません。

Linux でコンソールを使用する場合の CA 証明書のインポート
たとえば、**/root/ca.crt** ファイルに保存されている CA 証明書をデータベースに追加するには、以下を実行します。

```
# certutil -d ~/.redhat-idm-console/ -A -n "Example CA" -t CT,, -a -i /root/ca.crt
```

Windows でコンソールを使用する場合の CA 証明書のインポート
たとえば、**C :\ca.crt** ファイルに保存されている CA 証明書をデータベースに追加するには、以下を実行します。

```
> cd C:\Program Files\Red Hat Identity Management Console\
> certutil.exe -d "C:\Documents and Settings\user_name\.389-console\" -A -n "Example CA" -t
CT,, -a -i C:\ca.crt
```

9.4.4. Directory Server が使用する CA 証明書の Red Hat Enterprise Linux のトラストストアへの追加

Directory Server で TLS 暗号化を有効にすると、CA が発行した証明書を使用するようにインスタンスを設定します。クライアントが LDAPS プロトコルまたは LDAP 上の **STARTTLS** コマンドを使用してサーバーへの接続を確立する場合、Directory Server はこの証明書を使用して接続を暗号化します。クライアントユーティリティーは CA 証明書を使用して、サーバーの証明書が有効であるかどうかを確認します。デフォルトでは、これらのユーティリティーは、サーバーの証明書を信頼していない場合に接続を取り消します。

例9.4 クライアントユーティリティーが CA 証明書を使用しない場合の接続エラー

クライアントユーティリティーが CA 証明書を使用しない場合、ユーティリティーは TLS 暗号化の使用時にサーバーの証明書を検証できません。これにより、サーバーへの接続に失敗します。以下に例を示します。

```
# ldapsearch -H ldaps://server.example.com:636 -D "cn=Directory Manager" -W -b
"dc=example,dc=com" -x
Enter LDAP Password:
ldap_sasl_bind(SIMPLE): Can't contact LDAP server (-1)
```

Red Hat Enterprise Linux でクライアントユーティリティーを有効にして Directory Server が使用する証明書を検証するには、オペレーティングシステムのトラストストアに CA 証明書を追加します。

1. Directory Server が使用する CA 証明書のローカルコピーがない場合は、以下を実行します。
   a. サーバーの NSS データベースの証明書を一覧表示します。

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -L
```

```
Certificate Nickname                    Trust Attributes
                                        SSL,S/MIME,JAR/XPI

Example CA                              C,,
server-cert                             u,u,u
```

b. **NSS データベースの CA 証明書のニックネームを使用して、CA 証明書をエクスポートします。**

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -L -n "Example CA" -a > /tmp/ds-ca.crt
```

2. **CA 証明書を /etc/pki/ca-trust/source/anchors/ ディレクトリーにコピーします。以下に例を示します。**

```
# cp /tmp/ds-ca.crt /etc/pki/ca-trust/source/anchors/
```

3. **CA 信頼データベースを再構築します。**

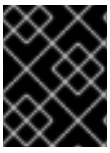
```
# update-ca-trust
```

9.5. DIRECTORY SERVER で有効な暗号化プロトコルの表示

Directory Server で有効な暗号化プロトコルを表示するには、次のコマンドを実行します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x \
    -s base -b 'cn=encryption,cn=config' sslVersionMin sslVersionMax

dn: cn=encryption,cn=config
sslVersionMin: TLS1.0
sslVersionMax: TLS1.2
```

***sslVersionMin* パラメーターおよび *sslVersionMax* パラメーターは、Directory Server が使用する暗号化プロトコルを制御します。デフォルトでは、プロトコルの TLS 1.0 以降のバージョンのみが有効になっています。**

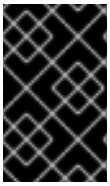


重要

セキュリティー上の理由から、パラメーターはセキュアでない SSL2 または SSL 3 プロトコルバージョンに設定することができません。

9.6. 暗号化プロトコルバージョンの設定

***sslVersionMin* パラメーターおよび *sslVersionMax* パラメーターを更新して、Directory Server が使用する暗号化プロトコルを設定します。**



重要

***sslVersionMax* パラメーターでサポートされる最強の暗号化プロトコルバージョンを常に使用するには、このパラメーターを設定しないでください。「*sslVersionMax* パラメーターにおける強固なプロトコルを自動的に使用」 を参照してください。**

たとえば、TLS 1.1 および 1.2 のみを有効にするには、以下を実行します。

1. ***sslVersionMin*** パラメーターおよび ***sslVersionMax*** パラメーターを更新します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=encryption,cn=config
changetype: modify
replace: sslVersionMin
sslVersionMin: TLS1.1
-
replace: sslVersionMax
sslVersionMax: TLS1.2
```

2. Directory Server インスタンスを再起動します。

```
# systemctl restart dirsrv@instance_name
```

9.6.1. *sslVersionMax* パラメーターにおける強固なプロトコルを自動的に使用

sslVersionMax パラメーターが設定されていない場合（デフォルト）、Directory Server は、このパラメーターに最も強力な暗号化プロトコルバージョンを使用します。これにより、更新後に常に最も強力なプロトコルバージョンを有効にできます。

sslVersionMax が設定されていない場合の特定
sslVersionMax が設定されていない場合でも、パラメーターが検索で返されます。パラメーターが設定されていないかどうかを特定するには、次のコマンドを実行します。

```
# grep sslVersionMax /etc/dirsrv/slapd-instance_name/dse.ldif
```

このコマンドで出力が表示されない場合、パラメーターは設定されず、デフォルト（最も強力な暗号化プロトコル）を使用します。

sslVersionMax パラメーターの削除
sslVersionMax パラメーターを削除して、デフォルトの設定を使用します。

1. ***sslVersionMax*** パラメーターを削除します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=encryption,cn=config
changetype: modify
delete: sslVersionMax
```

2. Directory Server インスタンスを再起動します。

```
# systemctl restart dirsrv@instance_name
```

9.7. ハードウェアセキュリティーモジュールの使用

セキュリティーモジュールは、Directory Server と TLS レイヤーとの間のメディアとして機能します。モジュールは、暗号化および復号に使用される鍵および証明書を保存します。これらのモジュールを定義する標準は Public Key Cryptography Standard (PKCS) #11 で、これらのモジュールは PKCS#11 モジュールです。

デフォルトでは、Directory Server はビルトインのセキュリティーデータベース **key3.db** および **cert8.db** を使用して、サーバーが使用する鍵と証明書を保存します。

外部のセキュリティーデバイスを使用して Directory Server 証明書および鍵を保存することもできます。Directory Server が外部の PKCS#11 モジュールを使用するには、モジュールのドライバーを Directory Server にインストールする必要があります。

詳細は、ハードウェアのセキュリティーモジュールのドキュメントを参照してください。

9.8. 証明書ベースのクライアント認証の使用

Directory Server は、LDAP クライアントの証明書ベースの認証と、レプリケーションなどのサーバー間接続をサポートします。

証明書ベースの認証を有効にしている場合は、設定によっては、クライアントが証明書を使用して認証したり、認証する必要があります。証明書を検証した後に、サーバーは証明書の **subject** フィールドの属性に基づいて、ディレクトリー内のユーザーを検索します。検索でユーザーエントリーを 1 つだけ返すと、Directory Server はこのユーザーを使用してすべての操作を行います。必要に応じて、認証に使用される証明書を、ユーザーの ***userCertificate*** 属性に保存されている Distinguished Encoding Rules(DER)形式の証明書と一致するように設定できます。

証明書ベースの認証を使用する利点:

- 効率が改善されました。証明書データベースのパスワードに一度要求されたアプリケーションを使用し、その証明書を後続のバインドまたは認証操作に使用すると、バインド DN およびパスワードを継続的に提供するよりも効率的です。
- セキュリティーが改善されました。証明書ベースの認証は、証明書ベースの認証では公開鍵の暗号化が使用されるため、証明書以外のバインド操作よりも安全です。バインド認証情報はネットワーク全体で傍受することはできません。証明書やデバイスが失われた場合は、PIN なしで使用しないため、フィッシング攻撃などのサードパーティーの干渉の影響を受けません。

9.8.1. 証明書ベースの認証の設定

証明書ベースの認証を有効にするには、以下を行います。

1. 暗号化された接続を有効にします。詳細は「TLS の有効化」を参照してください。
2. CA 証明書をインストールし、クライアントとサーバーの接続の信頼オプションを設定します。「CA 証明書のインストール」を参照してください。
3. 必要に応じて、クライアントおよびサーバーの **CT,,** 信頼オプションが CA 証明書に設定されていることを確認します。

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -L
Certificate Nickname                Trust Attributes
                                    SSL,S/MIME,JAR/XPI

Example CA                          CT,,
```

4. **/etc/dirsrv/slapd-*instance_name*/certmap.conf** ファイルを作成し、証明書から Directory Server ユーザーへ情報をマッピングします。以下に例を示します。

```
certmap default         default
default:DNComps         dc
```

```
default:FilterComps     mail,cn
default:VerifyCert      on

certmap example         o=Example Inc.,c=US
example:DNComps
```

これは、この発行者には *DNComps* パラメーターが空に設定されているため、o=Example Inc.,c=US 発行者識別名 (DN) セットを持つ証明書を使用するユーザーを認証するため、Directory Server が証明書のサブジェクトからベース DN を生成しないように設定されています。また、*FilterComps* および *VerifyCert* の設定も、デフォルトのエントリーから継承されます。

指定の証明書とは異なる発行者 DN を持つ証明書は default エントリーの設定を使用し、証明書のサブジェクトの *cn* 属性に基づいてベース DN を生成します。これにより、ディレクトリー全体を検索せずに、Directory Server が特定の DN で検索を開始できます。

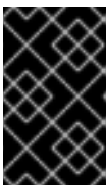
すべての証明書について、Directory Server は、証明書のサブジェクトの *mail* 属性および *cn* 属性を使用して検索フィルターを生成します。ただし、*mail* がサブジェクトに存在しない場合は、Directory Server はサブジェクトで証明書の *e* 属性の値を自動的に使用します。

利用可能なパラメーターの詳細と説明は、Red『Hat Directory Server の設定、コマンド、およびファイルリファレンス の certmap.conf ファイルの説明を参照してください』。

5. クライアント認証を有効にします。たとえば、クライアント認証を任意に設定するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -Z

dn: cn=encryption,cn=config
changetype: modify
replace: nsSSLClientAuth
nsSSLClientAuth: allowed
```

または、*nsSSLClientAuth* パラメーターを required に設定して、クライアントが認証に使用する必要のある証明書を設定します。



重要

Directory Server コンソールは、クライアント認証に対応していません。*nsSSLClientAuth* を required に設定すると、コンソールを使用してインスタンスを管理することはできません。

6. /etc/dirsrv/slapd-*instance_name*/certmap.conf ファイルで *alias_name*:VerifyCert on を設定して、認証証明書がユーザーの *userCertificate* 属性に保存されている証明書と一致する必要がある場合は、その証明書をユーザーエントリーに追加します。「ユーザーへの証明書の追加」を参照してください。

9.8.2. ユーザーへの証明書の追加

証明書ベースの認証を設定する際に、認証に使用する証明書が、ユーザーの *userCertificate* バイナリー属性に保存されている証明書と一致する必要があるように設定できます。/etc/dirsrv/slapd-*instance_name*/certmap.conf ファイルに *alias_name*:VerifyCert on を設定してこの機能を有効にした場合は、影響を受けるユーザーの証明書をディレクトリーエントリーに追加する必要があります。

重要

証明書を、*userCertificate* 属性の識別名エンコーディングルール (DER) 形式で保存する必要があります。

ユーザーの *userCertificate* 属性に証明書を保存するには、以下を行います。

1. 証明書が DER 形式ではない場合は、これを変換します。以下に例を示します。

```
# openssl x509 -in /root/certificate.pem -out /root/certificate.der -outform DER
```

2. 証明書をユーザーの *userCertificate* 属性に追加します。以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: uid=user_name,ou=People,dc=example,dc=com
changetype: modify
add: userCertificate
userCertificate: < /root/example.der
```

バイナリー属性の使用に関する詳細は、「Binary 属性の使用」を参照してください。

9.8.3. バインドリクエストの EXTERNAL SASL メカニズムの強制

TLS セッションの開始時に、クライアントは証明書をサーバーに送信します。次に、バインド要求を送信します。ほとんどのクライアントは、SASL メカニズム **EXTERNAL** を使用してバインド要求を実行します。これは、バインド要求の認証情報ではなく、バインドの証明書で ID を使用する必要があることを Directory Server に通知します。

ただし、クライアントが簡単な認証または匿名の認証情報を使用する場合は、この情報がありません。この場合は、証明書および証明書のクライアント ID が有効であっても、TLS セッションが無効な認証情報で失敗します。

Directory Server を設定してクライアントが SASL メカニズム **EXTERNAL** を使用し、要求内の他のバインドメソッドを無視するには、以下を行います。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=config
changetype: modify
replace: nsslapd-force-sasl-external
nsslapd-force-sasl-external: on
```

9.8.4. 証明書を使用した認証

OpenLDAP クライアントツールを使用して、証明書を使用して認証に対応する Directory Server インスタンスに対して認証します。

1. CA 証明書、ユーザーキー、およびユーザー証明書の対応するパスに、以下の環境変数を設定します。以下に例を示します。

```
LDAPTLS_CACERT=/home/user_name/CA.crt
LDAPTLS_KEY=/home/user_name/user.key
LDAPTLS_CERT=/home/user_name/user.crt
```

あるいは、 ~/.ldaprc ファイルに *TLS_CACERT* パラメーター、*TLS_KEY* パラメーター、および *TLS_CERT* パラメーターを設定します。詳細は、ldap.conf(5) の man ページの 『TLS OPTIONS』 セクションを参照してください。

2. サーバーに接続します。以下に例を示します。

```
# ldapwhoami -H ldaps://server.example.com:636
```

別のクライアントを使用する場合は、証明書ベースの認証を使用して接続する方法は、クライアントアプリケーションのドキュメントを参照してください。

9.9. SASL IDENTITY マッピングの設定

Red Hat Directory Server は、一部のアプリケーションが情報を安全に共有できるように、TLS の代替手段である Simple Authentication and Security Layer(SASL)による LDAP クライアント認証をサポートします。

Simple Authentication and Security Layer(SASL) は、LDAP などのプロトコルと GSS-API などの認証方法との間の抽象化レイヤーであり、SASL と対話できるプロトコルが SASL と連携できる認証メカニズムを利用できるようにします。簡単に言えば、SASL は、異なるメカニズムを使用して、アプリケーションに対して認証できるようにする仲介者です。SASL は、クライアントとサーバーとの間で暗号化されたセッションを確立するためにも使用できます。

SASL フレームワークでは、クライアントアプリケーションとサーバーアプリケーションの両方で有効になっているメカニズムに応じて、サーバーに対してユーザーを認証するためにさまざまなメカニズムを使用できます。SASL は、暗号化された (セキュアな) セッションのレイヤーも作成します。GSS-API を使用すると、Directory Server は Kerberos チケットを使用してセッションを認証し、データを暗号化します。

9.9.1. SASL Identity マッピングの概要

SASL バインド要求の処理時に、サーバーは、サーバー内に格納されている LDAP エントリーで Directory Server に対して認証するために使用される SASL 認証 ID を照合またはマップします。Kerberos を使用する場合、通常 SASL ユーザー ID の形式は userid@REALM になります (例: **scarter@EXAMPLE.COM**)。この ID は、**uid=scarter,ou=people,dc=example,dc=com** など、ユーザーの Directory Server エントリーの DN に変換する必要があります。

認証 ID が個人の LDAP エントリーに明確に対応する場合は、Directory Server が認証 ID を自動的にエントリー DN にマッピングするように設定できます。Directory Server には、最も一般的な構成を処理する事前構成済みのデフォルトマッピングがいくつかあり、カスタマイズされたマップを作成できます。デフォルトでは、バインド試行時に SASL マッピングフォールバックが有効ではない場合は、最初に一致するマッピングルールのみが適用されます。SASL マッピングフォールバックの詳細は、「SASL マッピングフォールバックの有効化」を参照してください。

1つのマッピングルールのみが認証文字列と一致するように、SASL マップを設定するようにしてください。

SASL マッピングは、コンテナーエントリー下のエントリーによって設定されます。

```
dn: cn=sasl,cn=config
objectClass: top
objectClass: nsContainer
cn: sasl
```

SASL アイデンティティーマッピングエントリーは、以下のエントリーの子です。

```
dn: cn=mapping,cn=sasl,cn=config
objectClass: top
objectClass: nsContainer
cn: mapping
```

マッピングエントリーは以下の属性で定義されます。

- ***nsSaslMapRegexString***: 指定した *authid* の要素をマップするために使用される正規表現。
- ***nsSaslMapFilterTemplate***: DN を作成する *nsSaslMapRegexString* の要素を適用するテンプレート。
- ***nsSaslMapBaseDNTemplate***: 構築した DN と照合する検索ベースまたは特定のエントリー DN を指定します。
- オプション: ***nsSaslMapPriority***: この SASL マッピングの優先度を設定します。***nsslapd-sasl-mapping-fallback*** が *cn=config* で有効になっている場合は、優先度値が使用されます。詳細は「SASL マッピングの優先度の設定」を参照してください。

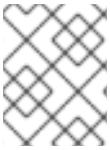
詳細は、Red 『Hat Directory Server の設定、コマンド、およびファイルリファレンス の該当するセクションを参照してください』。

以下に例を示します。

```
dn: cn=mymap,cn=mapping,cn=sasl,cn=config
objectclass:top
objectclass:nsSaslMapping
cn: mymap
nsSaslMapRegexString: \(.*\)@\(.*\)\.\(.*\)
nsSaslMapFilterTemplate: (objectclass=inetOrgPerson)
nsSaslMapBaseDNTemplate: uid=\1,ou=people,dc=\2,dc=\3
```

nsSaslMapRegexString 属性は、検索中にテンプレート属性に埋め込まれたバインド ID に対し、\1、\2、\3 形式の変数を設定します。この例では、inetOrgPerson オブジェクトクラスに属する ou=People,dc=example,dc=com サブツリーに含まれるユーザーに対して SASL アイデンティティーマッピングを設定します。

Directory Server が、**mconnors@EXAMPLE.COM** をユーザー ID (***authid***) として使用する SASL バインド要求を受け取ると、正規表現は **uid=mconnors,ou=people,dc=EXAMPLE,dc=COM** をユーザー ID として使用するベース DN テンプレートに入力し、認証がそこから続行します。



注記

dc の値は大文字と小文字を区別しないため、dc=EXAMPLE と dc=example は同じです。

Directory Server では、以下のようなより包含されたマッピングスキームも使用できます。

```
dn: cn=example map,cn=mapping,cn=sasl,cn=config
objectclass: top
objectclass: nsSaslMapping
cn: example map
nsSaslMapRegexString: \(.*\)
nsSaslMapBaseDNTemplate: ou=People,dc=example,dc=com
nsSaslMapFilterTemplate: (cn=\1)
```

これは任意のユーザー ID に一致し、フィルター cn=userId を満たす ou=People,dc=example,dc=com サブツリー下でエントリーをマップします。

nsSaslMapRegexString 属性にレルムを指定すると、マッピングを 1 つのレルムに制限することができます。以下に例を示します。

```
dn: cn=example map,cn=mapping,cn=sasl,cn=config
objectclass: top
objectclass: nsSaslMapping
cn: example map
nsSaslMapRegexString: \(.*\)@US.EXAMPLE.COM
nsSaslMapBaseDNTemplate: ou=People,dc=example,dc=com
nsSaslMapFilterTemplate: (cn=\1)
```

このマッピングは以前のマッピングと同じですが、US.EXAMPLE.COM レルムから認証されるユーザーにのみ適用されます。(レルムは「プリンシパルおよびレルムについて」で説明されています。)

レプリケーション時やチェーンでなど、別のサーバーに接続する場合、デフォルトのマッピングはアイデンティティーを適切にマッピングしません。これは、あるサーバーのプリンシパル (SASL ID) が、認証が実行するサーバー上のプリンシパルと一致しないため、マッピングエントリーに一致しないためです。

サーバーが SASL を使用したサーバー認証を使用できるようにするには、特定のサーバープリンシパルの特定ユーザーエントリーへのマッピングを作成します。たとえば、このマッピングは ldap1.example.com サーバーを cn=replication manager,cn=config エントリーに一致します。マッピングエントリー自体は 2 番目のサーバーに作成されます (例: ldap2.example.com)。

```
dn: cn=z,cn=mapping,cn=sasl,cn=config
objectclass: top
objectclass: nsSaslMapping
cn: z
nsSaslMapRegexString: ldap/ldap1.example.com@EXAMPLE.COM
nsSaslMapBaseDNTemplate: cn=replication manager,cn=config
nsSaslMapFilterTemplate: (objectclass=*)
```

レルム名は、SASL GSS-API 設定のプリンシパル名に含まれていないことがあります。2 つ目のマッピングは、プリンシパル名にレルムを指定せずに、最初のマッピングと同じ 2 番目のマッピングを作成できます。以下に例を示します。

```
dn: cn=y,cn=mapping,cn=sasl,cn=config
objectclass: top
objectclass: nsSaslMapping
cn: y
nsSaslMapRegexString: ldap/ldap1.example.com
nsSaslMapBaseDNTemplate: cn=replication manager,cn=config
nsSaslMapFilterTemplate: (objectclass=*)
```

レルムが指定されていないため、2 番目のマッピングはより一般的です (つまり、最初のマッピングよりも多くのエントリーと一致する可能性があります)。ベストプラクティスは、より具体的なマッピングを最初に処理し、より一般的なマッピングに徐々に進めていく方法です。

nsSaslMapPriority パラメーターを使用して SASL マッピングに優先度が設定されていない場合は、マッピングが処理される順序を指定する方法はありません。ただし、SASL マッピングの処理方法 (名前) を制御する方法もあります。Directory Server は、ASCII の逆順で SASL マッピングを処理しま

す。過去 2 つの例では、cn=z マッピング (最初の例) が最初に処理されます。一致する場合、サーバーは cn=y マッピングを処理します (2 番目の例)。

注記

LDIF ファイルでマッピングを指定し、ConfigFile ディレクティブで LDIF ファイルを追加すると、サイレントインストール中にインスタンスが作成される時に SASL マッピングを追加できます。サイレントインストールの使用方法は、『インストールガイド』 で説明しています。

9.9.2. Directory Server のデフォルトの SASL マッピング

Directory Server には、最も一般的な使用法のいくつかを処理するための事前定義された SASL マッピングルールがあります。

Kerberos UID マッピング

これは、user@example.com などの 2 部分レルムを使用して Kerberos プリンシパルと一致します。レルムは、検索ベースの定義に使用され、ユーザー ID (*authid*) はフィルターを定義します。検索ベースは dc=example,dc=com と (uid=user) のフィルターです。

```
dn: cn=Kerberos uid mapping,cn=mapping,cn=sasl,cn=config
objectClass: top
objectClass: nsSaslMapping
cn: Kerberos uid mapping
nsSaslMapRegexString: \(.*\)@\(.*\)\.\(.*\)
nsSaslMapBaseDNTemplate: dc=\2,dc=\3
nsSaslMapFilterTemplate: (uid=\1)
```

RFC 2829 DN 構文

このマッピングは、dn: で始まる有効な DN (RFC 2829 で定義) である *authid* と一致します。*authid* は、指定された DN に直接マッピングします。

```
dn: cn=rfc 2829 dn syntax,cn=mapping,cn=sasl,cn=config
objectClass: top
objectClass: nsSaslMapping
cn: rfc 2829 dn syntax
nsSaslMapRegexString: ^dn:\(.*\)
nsSaslMapBaseDNTemplate: \1
nsSaslMapFilterTemplate: (objectclass=*)
```

RFC 2829 U 構文

このマッピングは、u: の接頭辞が付いた UID の *authid* と一致します。プレフィックスの後に指定された値は (uid=value) のフィルターを定義します。検索ベースは、デフォルトの userRoot データベースの接尾辞になるようにハードコーディングされます。

```
dn: cn=rfc 2829 u syntax,cn=mapping,cn=sasl,cn=config
objectClass: top
objectClass: nsSaslMapping
cn: rfc 2829 u syntax
nsSaslMapRegexString: ^u:\(.*\)
nsSaslMapBaseDNTemplate: dc=example,dc=com
nsSaslMapFilterTemplate: (uid=\1)
```

UID マッピング

このマッピングは、他のデフォルトのマッピングルールに一致しない平文文字列である *authid* と一致します。この値を使用して (uid=value) のフィルターを定義します。検索ベースは、デフォルトの userRoot データベースの接尾辞になるようにハードコーディングされます。

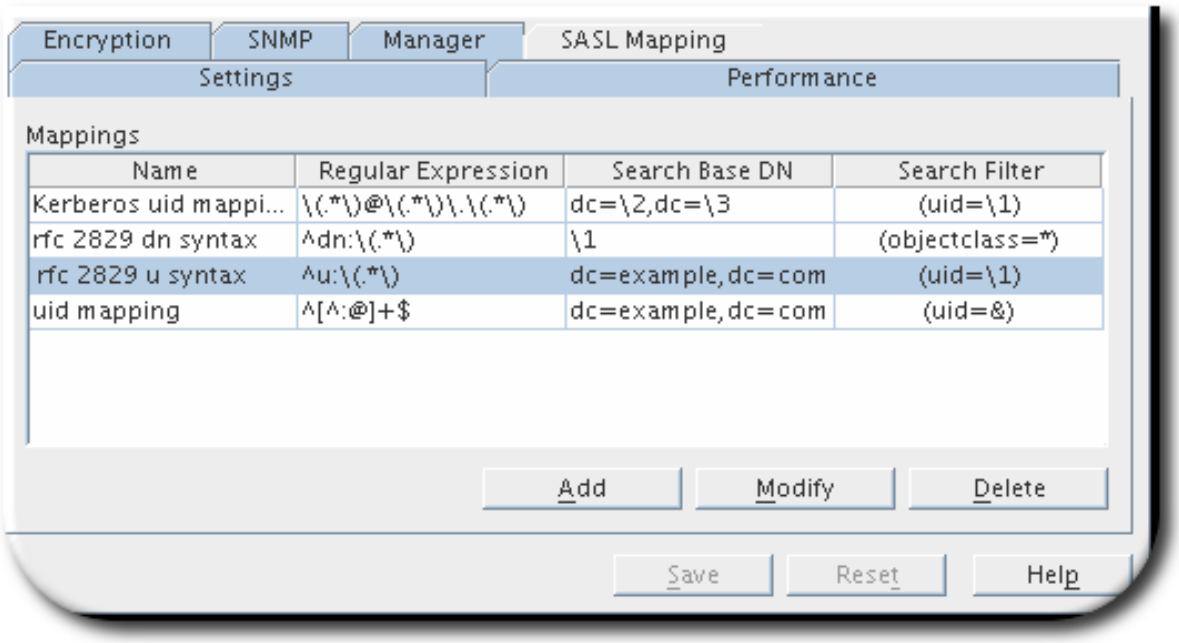
```
dn: cn=uid mapping,cn=mapping,cn=sasl,cn=config
objectClass: top
objectClass: nsSaslMapping
cn: uid mapping
nsSaslMapRegexString: ^[^:@]+$
nsSaslMapBaseDNTemplate: dc=example,dc=com
nsSaslMapFilterTemplate: (uid=&)
```

9.9.3. SASL Identity マッピングの設定

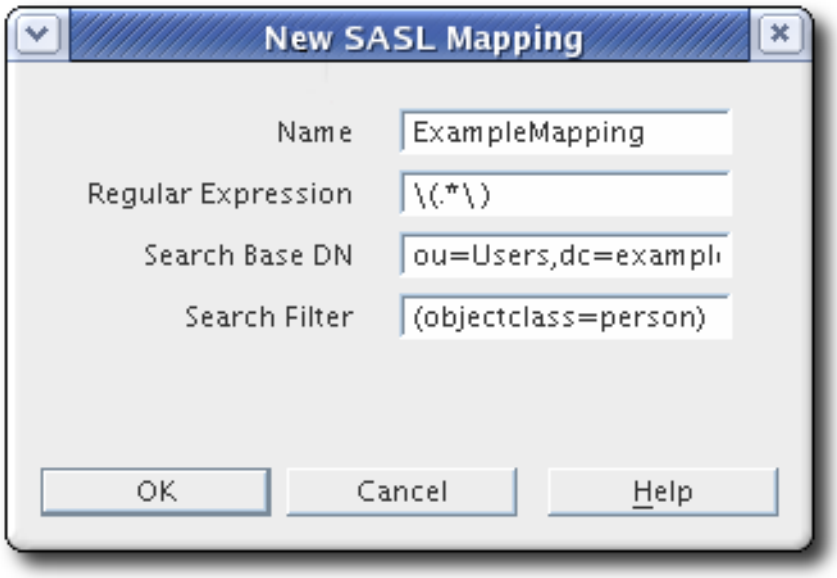
SASL ID マッピングは、Directory Server またはコマンドラインから設定できます。SASL 認証に対して SASL ID を機能させるには、マッピングが 1 つ返す必要があり、一致するエントリーと Kerberos をホストマシンに設定する必要があります。

9.9.3.1. コンソールからの SASL アイデンティティーマッピングの設定

1. Directory Server コンソールで、**Configuration** タブを開きます。
2. **SASL Mapping** タブを選択します。



3. 新しい SASL ID マッピングを追加するには、**Add** ボタンを選択し、必要な値を入力します。



- Nameこのフィールドは SASL マッピングの一意の名前を設定します。
- 正規表現。このフィールドは、\(.*\) などの DN コンポーネントに一致するために使用される正規表現を設定します。このフィールドは、SASL マッピング LDIF エントリーの **nsSaslMapRegexString** 値に対応します。
- 検索ベース DN。このフィールドは、**ou=People,dc=example,dc=com** などのエントリーをマップするために検索するベース **DN** を指定します。このフィールドは、SASL マッピング LDIF エントリーの **nsSaslMapBaseDNTemplate** 値に対応します。
- 検索フィルターこのフィールドには 、**(objectclass=*)** などの置き換えるコンポーネントの検索フィルターを指定します。このフィールドは、SASL マッピング LDIF エントリーの **nsSaslMapFilterTemplate** 値に対応します。

SASL アイデンティティーマッピングを編集するには、SASL **Mapping** タブでそのアイデンティティーを強調表示し、**Modify** をクリックします。値を変更して保存します。

SASL アイデンティティーマッピングを削除するには、それを強調表示し、**Delete** を押します。ダイアログボックスが、削除を確定します。

9.9.3.2. コマンドラインでの SASL Identity マッピングの設定

コマンドラインから SASL ID マッピングを設定するには、**ldapmodify** ユーティリティーを使用して ID マッピングスキームを追加します。以下に例を示します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=example map,cn=mapping,cn=sasl,cn=config
changetype: add
objectclass: top
objectclass: nsSaslMapping
cn: example map
nsSaslMapRegexString: \(.*\)
nsSaslMapBaseDNTemplate: ou=People,dc=example,dc=com
nsSaslMapFilterTemplate: (cn=\1)
```

これは、ユーザーの共通名に一致し、フィルター **cn=**userId に基づいて、ベース **ou=People,dc=example,dc=com** を用いたサブツリー検索の結果にマッピングされます。

注記

SASL マップが LDAP に追加されると、再起動するまでサーバーでは使用されません。ldapmodify で SASL マップを追加すると、ASCII 順序に関係なく、リストの最後にマッピングが追加されます。

9.9.4. SASL マッピングフォールバックの有効化

デフォルト設定を使用すると、Directory Server は最初に一致する SASL マッピングのみを検証します。最初に一致するマッピングが失敗すると、バインド操作に失敗し、さらに一致するマッピングは検証されません。

ただし、*nsslapd-sasl-mapping-fallback* パラメーターを有効にすると、Directory Server が一致するすべてのマッピングを検証するように設定できます。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=config
changetype: modify
replace: nsslapd-sasl-mapping-fallback
nsslapd-sasl-mapping-fallback: on
```

フォールバックが有効であり、1つのユーザー ID のみが返されると、バインドは成功します。ユーザーがない場合、または複数のユーザーが返されると、バインドは失敗します。

9.9.4.1. SASL マッピングの優先度の設定

nsslapd-sasl-mapping-fallback 属性を使用して SASL マッピングフォールバックを有効にすると、任意でマッピング設定の *nsSaslMapPriority* 属性を設定して優先順位を設定できます。*nsSaslMapPriority* 属性は、1 (最も高い優先度) から 100 (最も低い優先度) の値をサポートします。デフォルトは 100 です。

たとえば、cn=Kerberos uid mapping,cn=mapping,cn=sasl,cn=config マッピングの最も高い優先度を設定するには、以下のコマンドを実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=Kerberos uid mapping,cn=mapping,cn=sasl,cn=config
changetype: modify
replace: nsSaslMapPriority
nsSaslMapPriority: 1
```

9.10. SASL での KERBEROS GSS-API の使用

Directory Server が SASL 認証に GSS-API メカニズムを使用するには、Kerberos v5 をホストにデプロイする必要があります。Kerberos サービスを活用するには、GSS-API および Kerberos クライアントライブラリーを Directory Server ホストにインストールする必要があります。

9.10.1. Directory Server の SASL の認証メカニズム

Directory Server は、以下の SASL 暗号化メカニズムをサポートします。

- PLAIN。 PLAIN は、簡単なパスワードベースの認証用にクリアテキストのパスワードを送信します。

- EXTERNAL。 TLS を使用する EXTERNAL は、証明書ベースの認証を実行します。この方法では、強力な認証に公開鍵を使用します。
- CRAM-MD5。 **CRAM-MD5** は弱く、単純な challenge-response 認証メソッドです。セキュリティー層を確立しません。

警告

Red Hat では、セキュアでない CRAM-MD5 メカニズムを使用することは推奨されません。

- DIGEST-MD5。 **DIGEST-MD5** は LDAPv3 サーバーの弱い認証方法です。

警告

Red Hat では、セキュアでない DIGGEST-MD5 メカニズムを使用することは推奨されません。

- Generic Security Services (GSS-API).汎用セキュリティーサービス (GSS) は、UNIX ベースのオペレーティングシステムが Kerberos サービスにアクセスして認証するためのネイティブな方法であるセキュリティー API です。GSS-API は TLS と同様にセッション暗号化もサポートします。これにより、Kerberos バージョン 5 の認証情報 (チケット) を使用して LDAP クライアントがサーバーで認証でき、ネットワークセッションの暗号化を使用できます。

  Directory Server が GSS-API を使用するには、Kerberos をホストマシンに設定する必要があります。「SASL での Kerberos GSS-API の使用」を参照してください。

注記

GSS-API および Kerberos は GSS-API サポートのあるプラットフォームでのみサポートされます。GSS-API を使用するには、Kerberos クライアントライブラリーをインストールする必要があります。必要な Kerberos ライブラリーはすべてオペレーティングシステムベンダーから利用できます。

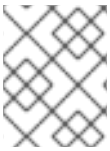
9.10.2. Directory Server の Kerberos の概要

Red Hat Enterprise Linux では、サポートされる Kerberos ライブラリーは MIT Kerberos バージョン 5 です。

Kerberos の概念、ならびに Kerberos の使用および設定については、MIT Kerberos の Web サイト http://web.mit.edu/Kerberos/ を参照してください。

9.10.2.1. プリンシパルおよびレルムについて

プリンシパル は、Kerberos 環境のユーザーまたはサービスです。レルム は、誰が何にアクセスできるかに関して、Kerberos が管理する内容を定義します。アクセスするクライアント、KDC、およびホストまたはサービスは同じレルムを使用する必要があります。



注記

Kerberos レルムは GSS-API 認証および暗号化でのみサポートされていますが、DIGEST-MD5 ではサポートされていません。

レルムは、LDAP DN のように、以下の形式でクライアントの DN を関連付けるためにサーバーによって使用されます。

uid=*user_name*/**[***server_instance***],cn=***realm***,cn=***mechanism***,cn=auth**

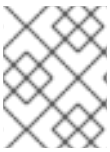
たとえば、**example.com** のヨーロッパの部門の **engineering** レルムの Mike Connors では、以下の関連を使用して、US レルムのサーバーにアクセスします。

uid=mconnors/cn=Europe.example.com,cn=engineering,cn=gssapi,cn=auth

Babara Jensen は、**US.example.com** の **accounting** レルムから、ローカルサーバーにアクセスする際にレルムを指定する必要はありません。

uid=bjensen,cn=accounting,cn=gssapi,cn=auth

レルムがメカニズムでサポートされ、デフォルトのレルムがサーバーに対する認証に使用されない場合、レルム は Kerberos プリンシパルで指定する必要があります。そうでない場合は、レルムを省略できます。



注記

Kerberos システムは、Kerberos レルムをデフォルトのレルムとして扱います。他のシステムはデフォルトでサーバーになります。

9.10.2.2. KDC サーバーおよびキータブの概要

キー配布センター (KDC) はユーザーを認証し、TGT (Ticket Granting Ticket) を発行します。これにより、ユーザーは GSS-API を使用して Directory Server に対して認証が可能になります。Kerberos 操作に応答するには、Directory Server でキータブファイルへのアクセスが必要になります。キータブには、Directory Server が他のサーバーへの認証に使用する暗号鍵が含まれます。

Directory Server は、Kerberos プリンシパルで **ldap** サービス名を使用します。以下に例を示します。

ldap/*server.example.com*/*EXAMPLE.COM*

キータブの作成に関する詳細は、Kerberos ドキュメントを参照してください。

注記

既存のエントリー識別名 (DN) にマッピングする Directory Server Kerberos プリンシパルの Simple Authentication and Security Layer (SASL) マッピングを作成する必要があります。

9.10.3. Directory Server 起動時の SASL 認証の設定

Kerberos チケットを認証に使用できるように、SASL GSS-API 認証は Directory Server でアクティベートする必要があります。これは、キータブファイルの場所を設定する変数を識別する init スクリプト用のシステム設定ファイルを指定することで行います。init スクリプトが Directory Server の起動時に実行すると、SASL 認証はすぐにアクティブになります。

デフォルトの SASL 設定は **/etc/sysconfig/dirsrv** ファイルに保存されます。

複数の Directory Server インスタンスがあり、これらすべてが SASL 認証を使用するわけではありません。その場合は、**dirsrv-**instance という名前の **/etc/sysconfig/** ディレクトリーにインスタンス固有の設定ファイルを作成できます (例: **dirsrv-example**)。ホストにインスタンスが 1 つある場合は、デフォルトの **dirsrv** ファイルを使用することができます。

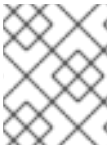
SASL 認証を有効にするには、**/etc/sysconfig/dirsrv** (またはインスタンス固有の) ファイルの **KRB5_KTNAME** 行のコメントを解除し、**KRB5_KTNAME** 変数のキータブの場所を設定します。以下に例を示します。

```
# In order to use SASL/GSSAPI the directory
# server needs to know where to find its keytab
# file - uncomment the following line and set
# the path and filename appropriately
KRB5_KTNAME=/etc/dirsrv/krb5.keytab
```

9.11. SASL メカニズムの設定

デフォルトでは、Directory Server は、簡単な認証とセキュリティーレイヤー (SASL) ライブラリーがサポートするすべてのメカニズムを有効にします。これらは、root dse ***supportedSASLMechanisms*** パラメーターに一覧表示されます。特定の SASL メカニズムを有効にするには、**cn=config** エントリーに ***nsslapd-allowed-sasl-mechanisms*** 属性を設定します。たとえば、**GSSAPI** および **DIGEST-MD5** メカニズムのみを有効にするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -x

dn: cn=config
changetype: modify
replace: nsslapd-allowed-sasl-mechanisms
nsslapd-allowed-sasl-mechanisms: GSSAPI, DIGEST-MD5
```



注記

EXTERNAL が ***nsslapd-allowed-sasl-mechanisms*** 属性に記載されていない場合でも、このメカニズムは常に有効になります。

詳細は、Red『Hat Directory Server の設定、コマンド、およびファイルリファレンス の該当するセクションを参照してください』。

9.12. LDAP クライアントでの SASL の使用

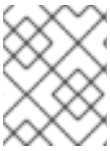
ldapsearch などの LDAP クライアントで SASL を使用するには、***-Y SASL_mechanism*** をコマンドに渡します。以下に例を示します。

- LDAP プロトコルで SASL メカニズム **GSSAPI** を使用するには、以下を行います。

```
# ldapsearch -Y GSSAPI -U "dn:uid=user_name,ou=people,dc=example,dc=com" -R
EXAMPLE.COM -H ldap://server.example.com -b "dc=example,dc=com"
```

- LDAPS プロトコルで SASL メカニズム **PLAIN** を使用するには、以下を行います。

```
# ldapsearch -Y PLAIN -D "uid=user_name,ou=people,dc=example,dc=com" -W -H
ldaps://server.example.com -b "dc=example,dc=com"
```



注記

Directory Server では、SASL プロキシーの認証はサポートされません。そのため、Directory Server は、クライアントによって提供される SASL **authzid** 値を無視します。

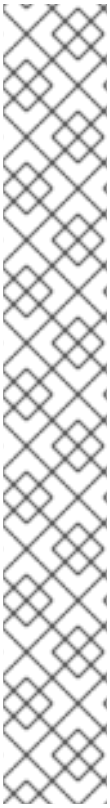
第10章 属性暗号化の設定

Directory Server は、権限のないユーザーがエントリー内の特定のエントリーや属性を読み取らないようにするアクセス制御ルールや、信頼できないネットワークでのデータの盗聴や改ざんからデータを保護する TLS など、機密データへのアクセスを保護するための多数のメカニズムを提供します。ただし、サーバーのデータベースファイルのコピーが権限のない人の手に渡った場合は、それらのファイルから機密情報を抽出する可能性があります。データベースの情報はプレーンテキストで保存されるため、政府の識別番号やパスワードなど、一部の機密情報が標準アクセス制御手段で保護されない可能性があります。

情報の機密性が高くなると、この情報損失の可能性は重大なセキュリティーリスクをもたらす可能性があります。このセキュリティーリスクを削除するには、Directory Server ではそのデータベースの一部を暗号化することができます。暗号化されると、攻撃者がサーバーのデータベースファイルのコピーがある場合にもデータを安全に実行できます。

データベース暗号化により、属性をデータベースで暗号化できます。暗号化と暗号化暗号の両方は、バックエンドごとの属性ごとに構成できます。これを設定すると、インデックスデータであっても、特定の属性内のすべてのインスタンスは、そのデータベースに保存されているすべてのエントリー用に暗号化されます。

属性の暗号化の利点として、暗号化された値は 1 を超える Security Strength Factor (SSF) を持つクライアントにのみ送信できます。



注記

暗号化されたデータには 1 つの例外があります。エントリーの RDN として使用される値はエントリー DN 内で暗号化されません。たとえば、*uid* 属性が暗号化されている場合、値はエントリーで暗号化されますが、DN に表示されます。

```
# entry-id: 16
dn: uid=jsmith1234,ou=People,dc=example,dc=com
nsUniqueId: ee91ea82-1dd111b2-9f36e9bc-39fb8550
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
givenName: John
sn: Smith
uid:: Sf04P9nJWGU1qiW9JJCGRg==
```

これにより、誰かが暗号化された値を検出できるようになります。

エントリー DN 内で使用される属性は常に DN に表示されるため、実質的には暗号化できません。DN を構築するのに使用される属性に注意し、それに応じて属性暗号化モデルを設計します。

インデックス化された属性は暗号化され、属性の暗号化は **eq** および **pres** のインデックスと完全に互換性があります。通常、属性値から派生するインデックスファイルの内容は、攻撃者がインデックスの分析から暗号化されたデータをすべて復旧することを防ぐためも暗号化されます。

サーバーは暗号化属性のインデックスを検索する前にすべてのインデックスキーを事前に暗号化するため、暗号化されたインデックスを使用する検索にはサーバーパフォーマンスにも影響が及ぶことはありますが、その影響はインデックスを使用する価値がなくなるほど深刻ではありません。

10.1. キーの暗号化

属性暗号化を使用するには、TLS に対してサーバーを設定し、TLS を有効にする必要があります。これは、属性暗号化がサーバーの TLS 暗号化キーと、TLS と同じ PIN 入力メソッドを使用するためです。サーバーの起動時に PIN を手動で入力するか、PIN ファイルを使用する必要があります。

無作為に生成される対称暗号キーは、属性データを暗号化および復号するために使用されます。設定された暗号には個別のキーが使用されます。これらの鍵は、サーバーの TLS 証明書から公開鍵を使用してラップ され、生成したラップ済みキーがサーバーの設定ファイル内に保存されます。属性暗号化の効果的な強度は、ラップに使用されるサーバーの TLS キーの強度よりも高くなります。サーバーの秘密鍵にアクセスできないと、ラップ済みのコピーから対称キーを復旧することができません。

警告

失われたキーを復元するメカニズムはありません。そのため、サーバーの証明書データベースを安全にバックアップすることが特に重要です。サーバーの証明書が失われた場合は、そのデータベースに保存されている暗号化データを復号することはできません。

警告

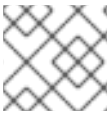
TLS 証明書の期限で更新が必要な場合は、更新前に暗号化されたバックエンドインスタンスをエクスポートします。証明書を更新して、エクスポートした LDIF ファイルを再インポートします。

10.2. 暗号化暗号

暗号化暗号は属性ごとに設定でき、属性に対する暗号化時に管理者が選択する必要があります。設定は、コンソールまたはコマンドラインから実行できます。

以下の暗号がサポートされます。

- AES (Advanced Encryption Standard)
- 3DES (Triple Data Encryption Standard)



注記

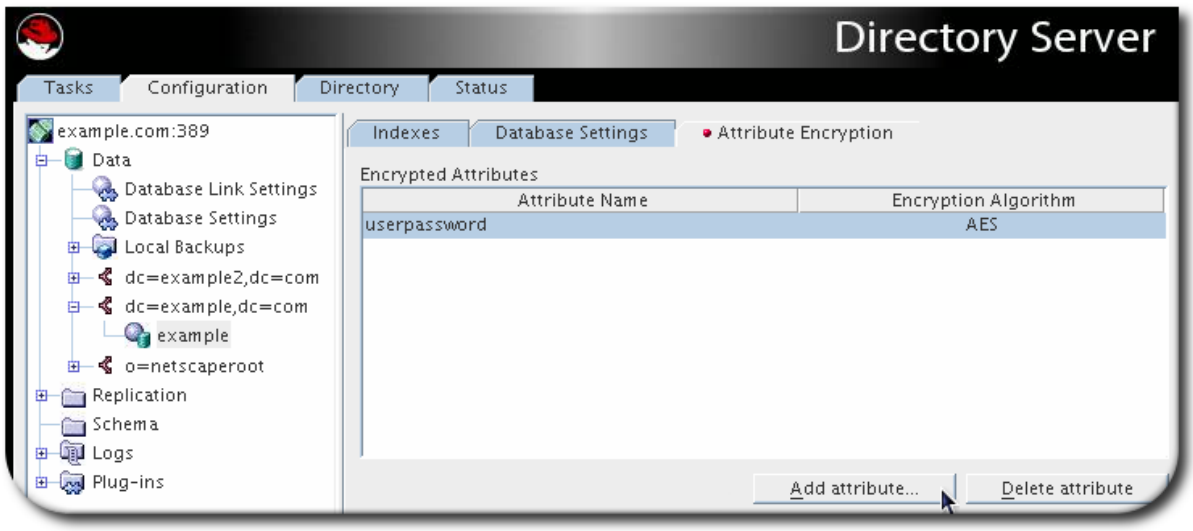
強力な暗号化の場合は、AES 暗号のみを使用することが推奨されます。

すべての暗号は Cipher Block Chaining モードで使用されます。

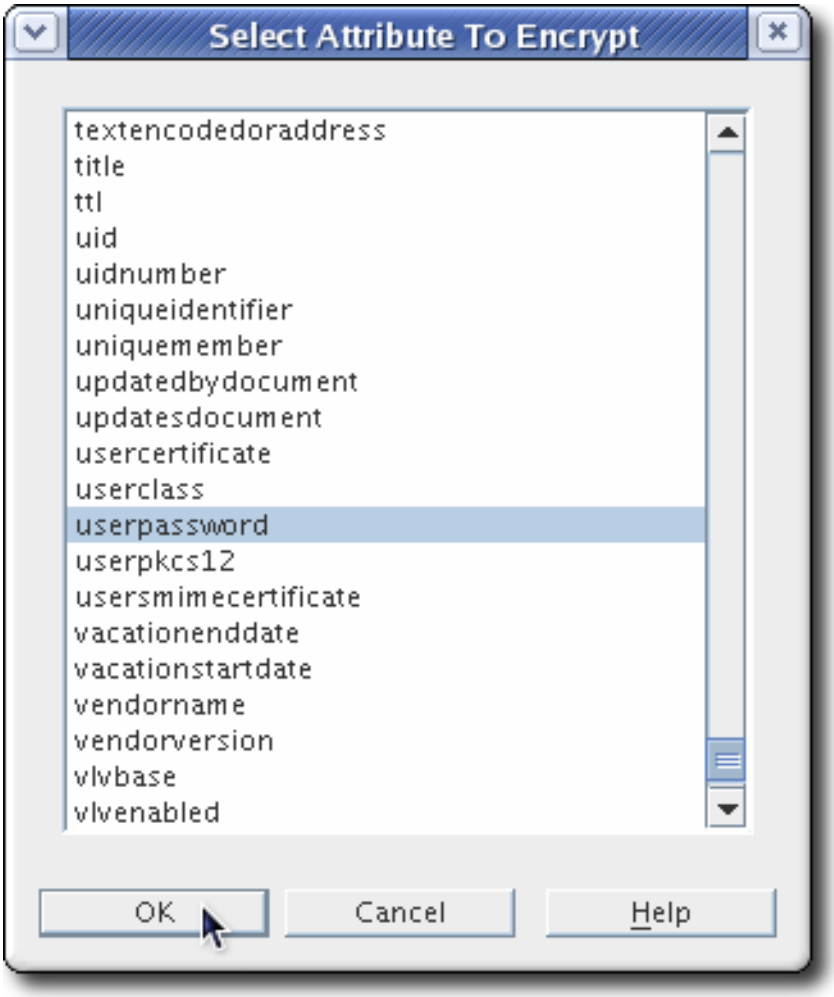
暗号化暗号が設定されたら、データをエクスポートおよび再インポートせずに変更しないでください。

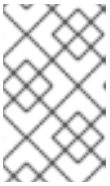
10.3. コンソールからの属性暗号化の設定

1. **Configuration タブで、Data node を選択します。**
2. **接尾辞を展開し、編集するデータベースを選択します。**
3. **Attribute Encryption タブを選択します。**



4. **Add Attribute ボタンをクリックして、属性の一覧を開きます。暗号化する属性を選択します。**

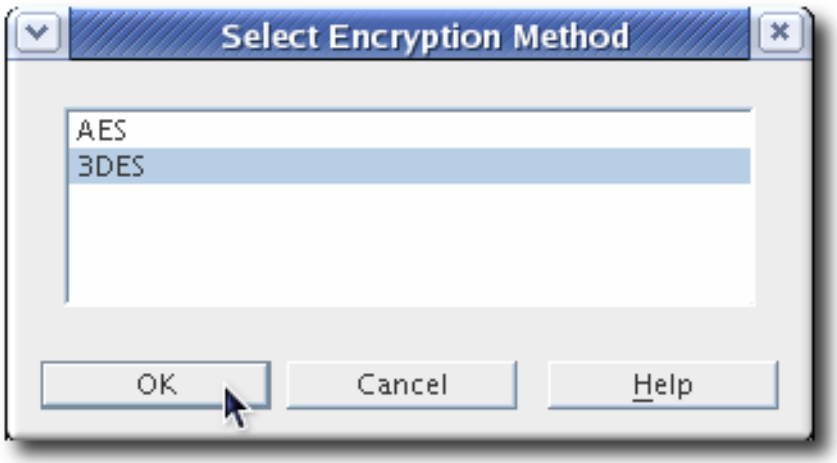


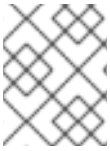


注記

既存の属性値を暗号化するには、データベースから情報をエクスポートしてから再インポートする必要があります。「暗号化したデータベースのエクスポートおよびインポート」 を参照してください。

5. 使用する暗号化暗号を選択します。





注記

使用する暗号化暗号は属性ごとに個別に設定されるため、属性の暗号化は一度に 1 つの属性に適用されます。

属性から暗号化を削除するには、属性暗号化 テーブルの 暗号化属性の 一覧から選択し、Delete ボタンをクリックし、Save をクリックして変更を適用します。削除された属性は、保存後に手動で再度追加する必要があります。

10.4. コマンドラインを使用した属性暗号化の設定

1. ldapmodify コマンドを実行します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
```

2. 暗号化する属性の暗号化エントリーを追加します。たとえば、このエントリーは、AES 暗号で *telephoneNumber* 属性を暗号化します。

```
dn: cn=telephoneNumber,cn=encrypted attributes,cn=Database1,cn=ldbm
database,cn=plugins,cn=config
changetype: add
objectclass: top
objectclass: nsAttributeEncryption
cn: telephoneNumber
nsEncryptionAlgorithm: AES
```

3. エントリーの既存の属性を暗号化するには、情報をエクスポートしてから再インポートする必要があります。「暗号化したデータベースのエクスポートおよびインポート」 を参照してください。

属性の暗号化設定スキーマの詳細は、Red 『Hat Directory Server の設定、コマンド、およびファイルリファレンス の cn=attributeName,cn=encrypted attributes,cn=database_name,cn=ldbm database,cn=plugins,cn=config" の「Database 属性」を参照してください』。

10.5. 既存の属性値の属性暗号化の有効化

既存の保存されたデータを持つ属性暗号化を有効にするには、最初にデータベースを LDIF にエクスポートしてから、データをデータベースに再インポートします。サーバーは暗号化設定と保存データの一貫性を強制しないため、暗号化の有効化または無効化の前に既存のデータがすべてエクスポートされることに注意してください。

10.6. 属性暗号化の有効化後の一般的な考慮事項

データベースにすでにあるデータの暗号化を有効にすると、以下を行います。

- 暗号化されていないデータは、サーバーのデータベースページプールのバッキングファイルで保持できます。このデータを削除するには、以下を実行します。

  1. インスタンスを停止します。

     ```
     # systemctl stop dirsrv@instance_name
     ```

  2. /var/lib/dirsrv/slapd-*instance_name*/db/guardian ファイルを削除します。

     ```
     # rm /var/lib/dirsrv/slapd-instance_name/db/guardian
     ```

  3. インスタンスを起動します。

     ```
     # systemctl start dirsrv@instance_name
     ```

- 暗号化を有効にし、データが正常にインポートされた後に、暗号化されていないデータで LDIF ファイルを削除します。
- 暗号化を有効にすると、データの再インポート時に Directory Server は新規データベースを削除し、作成します。
- レプリケーションログファイルは暗号化されません。このデータを保護するには、暗号化されたディスクに保存します。
- サーバーのメモリー (RAM) のデータは暗号化されず、swap パーティションに一時的に保存できます。このデータを保護するには、暗号化された swap 領域を設定します。

重要

暗号化されていないデータを含むファイルを削除すると、このデータは特定の状況で復元できます。

10.7. 暗号化したデータベースのエクスポートおよびインポート

暗号化されたデータベースのエクスポートおよびインポートは、通常のデータベースのエクスポートおよびインポートに似ています。ただし、暗号化された情報は、データをエクスポートする際に復号し、データベースに再インポートするときに再暗号化する必要があります。

10.7.1. 暗号化したデータベースのエクスポート

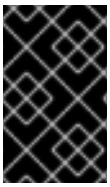
暗号化されたデータベースからデータをエクスポートするには、*-E* パラメーターを **db2ldif** スクリプトに渡します。このスクリプトは、Directory Sever 設定に保存されているパスワードを使用して、データベースを復号します。

完全なデータベースを暗号化するには、以下を実行します。

```
# db2ldif -Z instance_name -n database_name -E -a /tmp/data.ldif
```

または、特定のサブツリーのみをエクスポートできます。たとえば、**ou=People,dc=example,dc=com** エントリーから **/tmp/export.ldif** ファイルにすべてのデータをエクスポートするには、次のコマンドを実行します。

```
# db2ldif -Z instance_name -n database_name -E -s "ou=people,dc=example,dc=com" \
    -a /tmp/data.ldif
```



重要

db2ldif スクリプトは、Directory Server インスタンスのオペレーティングシステムアカウントを使用してコンテンツをエクスポートします。したがって、このアカウントは、**-a** オプションで設定したファイルに書き込みできる必要があります。

10.7.2. 暗号化されたデータベースへの LDIF ファイルのインポート

暗号化が有効な場合にデータをデータベースにインポートするには、以下を実行します。

1. Directory Server インスタンスを停止します。

   ```
   # systemctl stop dirsrv@instance_name
   ```

2. 前回のエクスポートと今回のインポートとの間で証明書データベースを置き換えた場合は、**/etc/dirsrv/slapd-*instance_name*/dse.ldif** ファイルを編集して、その属性を含む以下のエントリーを削除します。

   - **cn=AES,cn=encrypted attribute keys,cn=*database_name*,cn=ldbm database,cn=plugins,cn=config**
   - **cn=3DES,cn=encrypted attribute keys,cn=*database_name*,cn=ldbm database,cn=plugins,cn=config**

   重要

   全データベースのエントリーを削除します。***nsSymmetricKey*** 属性を含むエントリーが **/etc/dirsrv/slapd-*instance_name*/dse.ldif** ファイルに残されると、Directory Server は起動に失敗します。

3. LDIF ファイルをインポートします。以下に例を示します。

   ```
   # ldif2db -Z instance_name -n database_name -E -i /tmp/data.ldif
   ```

   -E パラメーターにより、スクリプトはインポート中に暗号化を設定する属性を暗号化できます。

4. インスタンスを起動します。

```
# systemctl start dirsrv@instance_name
```

10.8. 属性暗号化に使用される TLS 証明書の更新

属性の暗号化は TLS 証明書に基づいています。TLS 証明書の更新または置き換え後に属性の暗号化が失敗するのを防ぐには、以下を実行します。

1. 復号化された属性でデータベースをエクスポートします。「暗号化したデータベースのエクスポート」を参照してください。
2. Network Security Services (NSS) データベースから既存の秘密鍵と証明書を削除します。「秘密鍵の削除」を参照してください。
3. Certificate Signing Request (CSR) を新規作成します。「証明書署名要求の作成」を参照してください。
4. 新しい証明書をインストールします。「証明書のインストール」 を参照してください。
5. Directory Server インスタンスを停止します。

```
# systemctl stop dirsrv@instance_name
```

6. /etc/dirsrv/slapd-*instance_name*/dse.ldif ファイルを編集し、属性を含む以下のエントリーを削除します。
   - cn=AES,cn=encrypted attribute keys,cn=*database_name*,cn=ldbm database,cn=plugins,cn=config
   - cn=3DES,cn=encrypted attribute keys,cn=*database_name*,cn=ldbm database,cn=plugins,cn=config

重要

全データベースのエントリーを削除します。*nsSymmetricKey* 属性を含むエントリーが /etc/dirsrv/slapd-*instance_name*/dse.ldif ファイルに残されると、Directory Server は起動に失敗します。

7. データベースをインポートします。「暗号化されたデータベースへの LDIF ファイルのインポート」を参照してください。
8. インスタンスを起動します。

```
# systemctl start dirsrv@instance_name
```

第11章 FIPS モードサポートの管理

Red Hat Directory Server は、連邦情報処理標準 (FIPS) 140-2 を完全にサポートします。Directory Server が FIPS モードで実行すると、セキュリティー関連の設定が変更になります。たとえば、SSL は自動的に無効になり、TLS 1.1 および 1.2 暗号化のみが使用されます。

FIPS に関する一般的な情報は、『Red Hat Enterprise Linux セキュリティーガイド』 の 連邦情報処理標準 (FIPS) を参照してください。

FIPS モードサポートの有効化

Directory Server の FIPS モードのサポートを有効にするには、以下を実行します。

1. 必要に応じて、Red Hat Enterprise Linux で FIPS モードを有効にします。詳細は、『Red Hat Enterprise Linux セキュリティーガイド』 の該当するセクションを参照してください。
2. ネットワークセキュリティーサービス (NSS) データベースの FIPS モードを有効にします。

```
# modutil -dbdir /etc/dirsrv/slapd-instance_name/ -fips true
```

3. Directory Server インスタンスを再起動します。

```
# systemctl restart dirsrv@instance_name
```

FIPS モードサポートの無効化

Directory Server の FIPS モードのサポートを無効にするには、以下を実行します。

1. ネットワークセキュリティーサービス (NSS) データベースの FIPS モードを無効にします。

```
# modutil -dbdir /etc/dirsrv/slapd-instance_name/ -fips false
```

2. Directory Server インスタンスを再起動します。

```
# systemctl restart dirsrv@instance_name
```

3. 必要に応じて、Red Hat Enterprise Linux で FIPS モードを無効にします。詳細は、『Red Hat Enterprise Linux セキュリティーガイド』 の該当するセクションを参照してください。

第12章 ディレクトリースキーマの管理

Red Hat Directory Server には、数百のオブジェクトクラスおよび属性を含む標準スキーマが同梱されています。標準のオブジェクトクラスおよび属性はほとんどのデプロイメントの要件を満たす必要がありますが、特定のディレクトリーデータのスキーマを拡張しないといけない場合があります。スキーマの拡張は、新規オブジェクトクラスおよび属性を作成することで行われます。

『Red Hat Directory Server 10 Configuration, Command, and File Reference』 は、ほとんどの標準の Directory Server 属性およびオブジェクトクラスのリファレンスであり、許可された属性および必須属性、どのオブジェクトクラスがどの属性を取るか、および OID と値の情報を受け取ります。これは、ディレクトリーで有用なスキーマ要素を特定し、作成されるカスタムスキーマを決定するのに適したリソースです。

12.1. スキーマの概要

ディレクトリースキーマは、ディレクトリーへのデータの保存方法を定義する一連のルールです。ディレクトリー情報は個別のエントリーに保存され、各エントリーは属性のセットとその値で構成されます。エントリーで説明されるアイデンティティーの種類は、エントリーのオブジェクトクラスで定義されます。オブジェクトクラスは、オブジェクトクラスの定義された属性セットでエントリーが記述するオブジェクトの種類を指定します。

LDAP では、オブジェクトクラスはエントリーの定義に使用できる属性のセットを定義します。LDAP 標準仕様は、人、グループ、場所、組織、部門、機器など、多くの一般的なエントリーに対するオブジェクトクラスを提供します。ID は、属性とその値を含むディレクトリーエントリーで説明されています。ペアは、属性値表明 または AVA と呼ばれます。ディレクトリー内の情報には説明的な属性が関連付けられています。一致するルールや LDAP コントロールを含む Directory Server 設定のその他の側面は、スキーマにも定義されます。これらすべてが スキーマ要素 です。

すべての schema 要素は、一意のドット区切り番号で識別されます。これは オブジェクト ID または OID と呼ばれます。

12.1.1. デフォルトのスキーマファイル

Directory Server のスキーマは、複数のスキーマファイル (スキーマ要素を定義する LDIF ファイル) で定義されます。Directory Server スキーマファイルは、**/usr/share/dirsrv/schema/** ディレクトリーにあります。このディレクトリーのファイルは、新しい Directory Server インスタンスのテンプレートとして使用されます。このディレクトリーに新しいスキーマを追加すると、新しいインスタンスが利用可能になります。

Directory Server が操作を実行し、エントリーを管理するために使用する属性は、『Red Hat Directory Server 10 の設定、コマンド、およびファイルリファレンス の他の構成設定で説明されています。』

12.1.2. オブジェクトクラス

LDAP では、オブジェクトクラスはエントリーの定義に使用できる属性のセットを定義します。LDAP 標準仕様は、ユーザー (**person** および **inetOrgPerson**)、グループ (**groupOfNames**)、場所 (**locality**)、組織および部門 (**organization** および **organizationalUnit**)、および機器 (**device**) など、多くの一般的なエントリーに対するオブジェクトクラスを提供します。

スキーマファイルでは、オブジェクトクラスは ***objectclasses*** 行によって識別され、その後 OID、名前、説明、その直接の上位オブジェクトクラス (オブジェクトクラスと使用する必要のあるオブジェクトクラス、およびそのオブジェクトクラスと属性を共有するのに必要なオブジェクトクラス)、および必須属性の一覧 (**MUST**) および許可される属性の一覧 (**MAY**) が続きます。

これは、例12.1「個人のオブジェクトクラススキーマエントリー」 に示されています。

例12.1 個人のオブジェクトクラススキーマエントリー

```
objectClasses: ( 2.5.6.6 NAME 'person' DESC 'Standard LDAP objectclass' SUP top MUST ( sn $ cn ) MAY ( description $ seeAlso $ telephoneNumber $ userPassword ) X-ORIGIN 'RFC 4519' )
```

すべてのオブジェクトクラスは、必須属性 (そのスキーマの MUST キーワード) および許可された属性 (そのスキーマの MAY キーワード) を定義します。必須属性は、指定されたオブジェクトクラスを使用するエントリーに存在する必要がありますが、許可された属性は許可されており、エントリーで使用できますが、エントリーが有効である必要はありません。

例12.1「個人のオブジェクトクラススキーマエントリー」 のように、person オブジェクトクラスには、*cn* 属性、*sn* 属性、および *objectClass* 属性が必要で、*description* 属性、*seeAlso* 属性、*telephoneNumber* 属性、および *userPassword* 属性を許可します。

オブジェクトクラスは、独自の必須属性と許可される属性に加えて、別のクラスから属性を継承できます。2 つ目のオブジェクトクラスは、最初のオブジェクトクラスの superior または parent オブジェクトクラスです。

たとえば、ユーザーのエントリーに inetOrgPerson オブジェクトクラスが必要です。その場合、エントリーには、inetOrgPerson の上位オブジェクトクラスである organizationalPerson と、organizationalPerson の上位オブジェクトクラスである person も含める必要があります。

```
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
```

12.1.3. 属性

ディレクトリーエントリーは、属性とその値で構成されます。これらのペアは、属性値表明 または AVA と呼ばれます。ディレクトリー内の情報には説明的な属性が関連付けられています。たとえば、*cn* 属性は、cn: John Smith などのユーザーの氏名を保存するために使用されます。

追加の属性は、John Smith に関する補足情報を提供できます。

```
givenname: John
surname: Smith
mail: jsmith@example.com
```

スキーマファイルでは、属性が以下によって記述されます。

- OID
- name
- 構文マッチングルール (任意)
- 部分文字列マッチングルール (任意)
- 順序ルール (任意)

- 説明 (任意)
- 構文
- 単値または多値の属性
- 属性が定義されている場所の詳細

これは、例12.2「*uid*属性スキーマエントリー」 に示されています。

例12.2 *uid*属性スキーマエントリー

```
( 0.9.2342.19200300.100.1.1 NAME ( 'uid' 'userid' ) EQUALITY caseIgnoreMatch SUBSTR
caseIgnoreSubstringsMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN 'RFC 4519' )
```

12.1.4. スキーマの拡張

新規、カスタム属性、およびオブジェクトクラスを Directory Server インスタンスに追加してスキーマを拡張することができ、スキーマ要素を追加する方法は複数あります。Directory Server Console または LDAP ツールを使用すると、インスタンスのデフォルトのカスタムスキーマファイルにスキーマ要素が追加されます(99 user.ldif)。新しい個別のスキーマファイルを作成し、デフォルトのスキーマファイルで追加することもできます。

新規スキーマ要素を追加するには、3 点が必要になります。

1. 新規スキーマの OID の計画および定義。スキーマ要素はその OID によってサーバーが認識されるため、OID を一意で、整理することが重要です。Directory Server 自体は OID を管理しませんが、「オブジェクト識別子の管理」で説明するベストプラクティスがいくつかあります。

2. 新しい属性を作成します。属性定義には名前、構文 (許可される値の形式)、OID、および属性をエントリーごとに一度または複数回使用できるかどうかの説明が必要です。

3. 新規属性を含むオブジェクトクラスを作成します。オブジェクトクラスは、そのエントリータイプに必要な属性と許可される (許容) 属性を一覧表示します。デフォルトのスキーマは変更できないため、新規属性を作成している場合は、カスタムオブジェクトクラスに追加する必要があります。

スキーマ要素は事前に計画する必要があります。同じ情報に複数の属性を使用しないでください。可能な場合は、標準の Directory Server スキーマを使用します。Directory Server には、数百の属性があり、デフォルトのスキーマファイルで定義されたオブジェクトクラスが多数あります。『Red Hat Directory Server 10 の設定、コマンド、およびファイルリファレンス は』、標準の属性およびオブジェクトクラスを一覧表示して説明します。スキーマはすべて Directory Server コンソールに表示するか、/usr/share/dirsrv/schema/ のスキーマファイルで読み取ることができます。まずは、利用可能なスキーマを確認してください。次に、不足している情報属性と、不足している情報属性を補うためにカスタム属性を使用した最善の方法を計画します。スキーマのプランニングについては、『デプロイメントガイド』 で説明しています。

警告

Directory Server のデフォルトのオブジェクトクラスおよび属性は LDAP および X.500 標準仕様および RFC に基づいています。標準スキーマを使用すると、Directory Server が他のアプリケーションやサーバーとより簡単に統合され、LDAP クライアント、レガシー Directory Server インスタンス、および今後のリリースで相互運用性が可能になります。標準属性を編集したり、オブジェクトクラスを変更したりすることは推奨されません。

Directory Server スキーマをカスタマイズする場合は、以下のルールを念頭に置いてください。

- スキーマはできるだけシンプルに保ちます。
- 可能であれば、既存のスキーマ要素を再利用します。
- 各オブジェクトクラスに定義される必須属性の数を最小限に抑えます。
- 複数のオブジェクトクラスまたは属性を同じ目的で定義しないでください。
- 属性またはオブジェクトクラスの既存の定義は変更しないでください。

注記

標準スキーマを削除または置き換えることは ありません。これを行うと、他のディレクトリーやその他の LDAP クライアントアプリケーションとの互換性の問題が発生する可能性があります。

インスタンスが起動すると、スキーマが Directory Server インスタンスに読み込まれます。Directory Server が再起動するか、再読み込みタスクが開始されない限り、新しいスキーマファイルは読み込まれません。インスタンスのデフォルトのカスタムスキーマファイルは、*99user.ldif* が最後のスキーマファイルとして読み込まれます。標準スキーマファイルに定義がすでに含まれる場合、カスタム定義は標準スキーマファイルを上書きします。

12.1.5. スキーマレプリケーション

ディレクトリースキーマが **cn=schema** サブツリーで更新されると、Directory Server は変更状態番号 (CSN) を含むローカルの **/etc/dirsrv/slapd-*instance_name*/schema/99user.ldif** ファイルに変更を保存します。更新されたスキーマは他のレプリカに自動的に複製されません。スキーマレプリケーションは、ディレクトリーのコンテンツが複製されたツリーで更新されると開始します。たとえば、スキーマの変更後にユーザーエントリーまたはグループエントリーを更新すると、***nsSchemaCSN*** 属性に保存されている CSN と、コンシューマーにある CSN が比較されます。リモート CSN がサプライヤー上のものよりも小さい場合、スキーマはコンシューマーに複製されます。レプリケーションに成功すると、サプライヤーにあるすべてのオブジェクトクラスと属性タイプはコンシューマーの定義のスーパーセットである必要があります。

> **例12.3 スキーマのサブセットとスーパーセット**
>
> - **server1** では、**demo** オブジェクトクラスは ***a1*** 属性、***a2*** 属性、および ***a3*** 属性を許可します。
> - **server2** では、**demo** オブジェクトクラスは ***a1*** 属性および ***a3*** 属性を許可します。

例12.3「スキーマのサブセットとスーパーセット」では、server1 にある demo オブジェクトクラスのスキーマ定義は、server2 のオブジェクトクラスのスーパーセットです。検証フェーズで、スキーマが複製または許可されると、Directory Server はスーパーセット定義を取得します。たとえば、ローカルスキーマのオブジェクトクラスがサプライヤースキーマのオブジェクトクラスよりも少ない属性を許可していることをコンシューマーが検出すると、ローカルスキーマが更新されます。

スキーマ定義が正常に複製された場合、*nsSchemaCSN* 属性は両サーバーで同一であり、レプリケーションセッションの開始時に比較されなくなります。

以下のシナリオでは、スキーマは複製されません。

- あるホストのスキーマは、別のホストのスキーマのサブセットです。

  たとえば、例12.3「スキーマのサブセットとスーパーセット」では、server2 にある demo オブジェクトクラスのスキーマ定義は server1 のオブジェクトクラスのサブセットです。サブセットは、属性 (単一値属性は複数値属性のサブセット) および属性の構文 (IA5 は Octet_string のサブセット) に対しても発生する可能性があります。

- サプライヤースキーマとコンシューマースキーマの定義をマージする必要があります。

  Directory Server はマージスキーマをサポートしません。たとえば、1 台のサーバーのオブジェクトクラスが *a1* 属性、*a2* 属性、および *a3* 属性を許可し、別のサーバーのオブジェクトクラスが *a1* 属性、*a3* 属性、および *a4* 属性を許可する場合、スキーマはサブセットではなく、マージできません。

- /etc/dirsrv/slapd-*instance_name*/schema/99user.ldif 以外のスキーマファイルが使用されます。

  Directory Server を使用すると、/etc/dirsrv/slapd-*instance_name*/schema/ ディレクトリーにスキーマファイルを追加できます。ただし、99user.ldif ファイルの CSN のみが更新されます。このため、他のスキーマファイルはローカルでのみ使用され、レプリケーションパートナーに自動的に転送されません。更新されたスキーマファイルをコンシューマーに手動でコピーし、スキーマを再読み込みします。詳細は「スキーマの動的再読み込み」を参照してください。

  スキーマ定義の重複を回避し、自動レプリケーションを有効にするには、すべてのカスタムスキーマを /etc/dirsrv/slapd-*instance_name*/schema/99user.ldif ファイルに保存します。カスタムスキーマファイルの作成方法は、「カスタムスキーマファイルの作成」を参照してください。

12.2. オブジェクト識別子の管理

各 LDAP オブジェクトクラスまたは属性には、一意の名前と オブジェクト識別子 (OID) を割り当てる必要があります。OID は、サーバーへのスキーマ要素を識別するドットで区切られた番号です。OID は、異なるブランチに対応するために拡張できるベース OID を使用して階層化することができます。たとえば、ベース OID は 1 で、属性のブランチを 1.1 にし、オブジェクトクラスのブランチを 1.2 にすることもできます。

注記

カスタムスキーマを作成するために数字の OID を使用する必要はありませんが、Red Hat では前方互換性とパフォーマンスを向上させることを強く推奨します。

OID は、Internet Assigned Numbers Authority (IANA) を介して組織に割り当てられ、Directory Server は OID を取得するメカニズムを提供しません。OID の取得に関する情報を取得するには、IANA の Web サイト http://www.iana.org/cgi-bin/enterprise.pl にアクセスします。

IANA からベース OID を取得したら、OID をカスタムスキーマ要素に割り当てる方法を計画します。属性とオブジェクトクラスの両方にブランチを定義します。ルールと LDAP コントロールに一致するブランチを使用することもできます。

OID ブランチが定義されたら、OID 割り当てを追跡する OID レジストリーを作成します。OID レジストリーは、ディレクトリースキーマで使用される OID と説明を提供するリストです。これにより、OID が複数の目的に使用されないようにします。カスタムスキーマで OID レジストリーを公開します。

12.3. DIRECTORY SERVER 属性の構文

属性の構文は、属性が許可する値の形式を定義します。他のスキーマ要素と同様に、構文は、スキーマファイルエントリーで構文の OID を使用して属性に対して定義されます。Directory Server コンソールでは、構文はその分かりやすい名前で参照されます。

Directory Server は、属性の構文を使用してエントリーでのソートとパターン一致を実行します。

LDAP 属性の構文に関する詳細は、RFC 4517 を参照してください。

サポートされる LDAP 属性構文は、『Red Hat Directory Server 10 Configuration, Command, and File Reference』の「『Directory Server Attribute Syntaxes』」の項に記載されています。

12.4. コンソールでのカスタムスキーマの管理

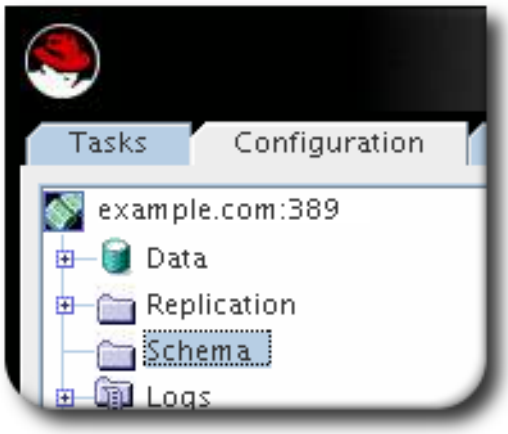
Directory Server コンソールは、スキーマのすべての属性を表示し、カスタム属性はスキーマから作成、編集、および削除できます。

- 「属性およびオブジェクトクラスの表示」
- 「属性の作成」
- 「オブジェクトクラスの作成」
- 「カスタムスキーマ要素の編集」
- 「スキーマの削除」

12.4.1. 属性およびオブジェクトクラスの表示

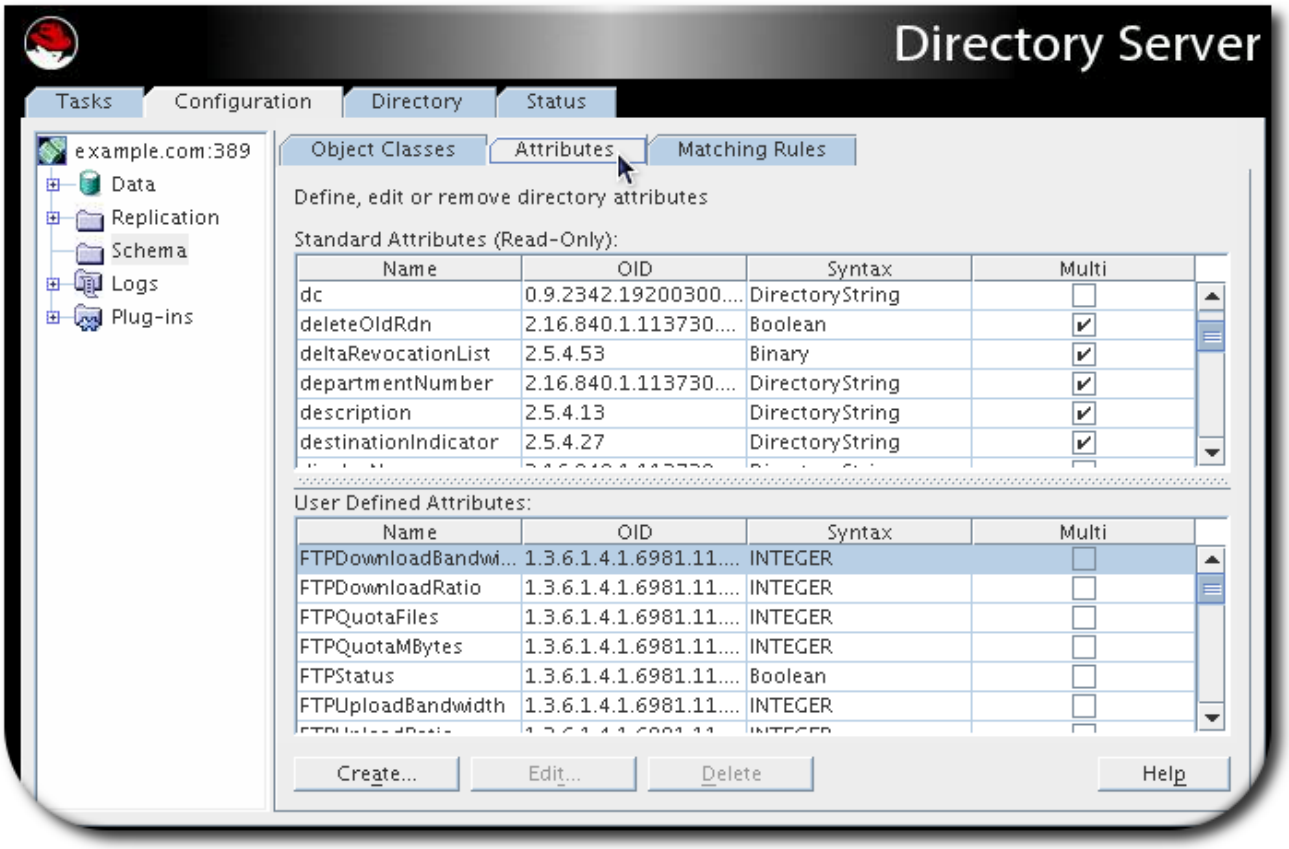
サーバーインスタンスに現在読み込まれている属性とオブジェクトクラスに関する情報はすべて、他のサーバー設定と表示されます。

1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のナビゲーションツリーで **Schema** フォルダーを選択します。

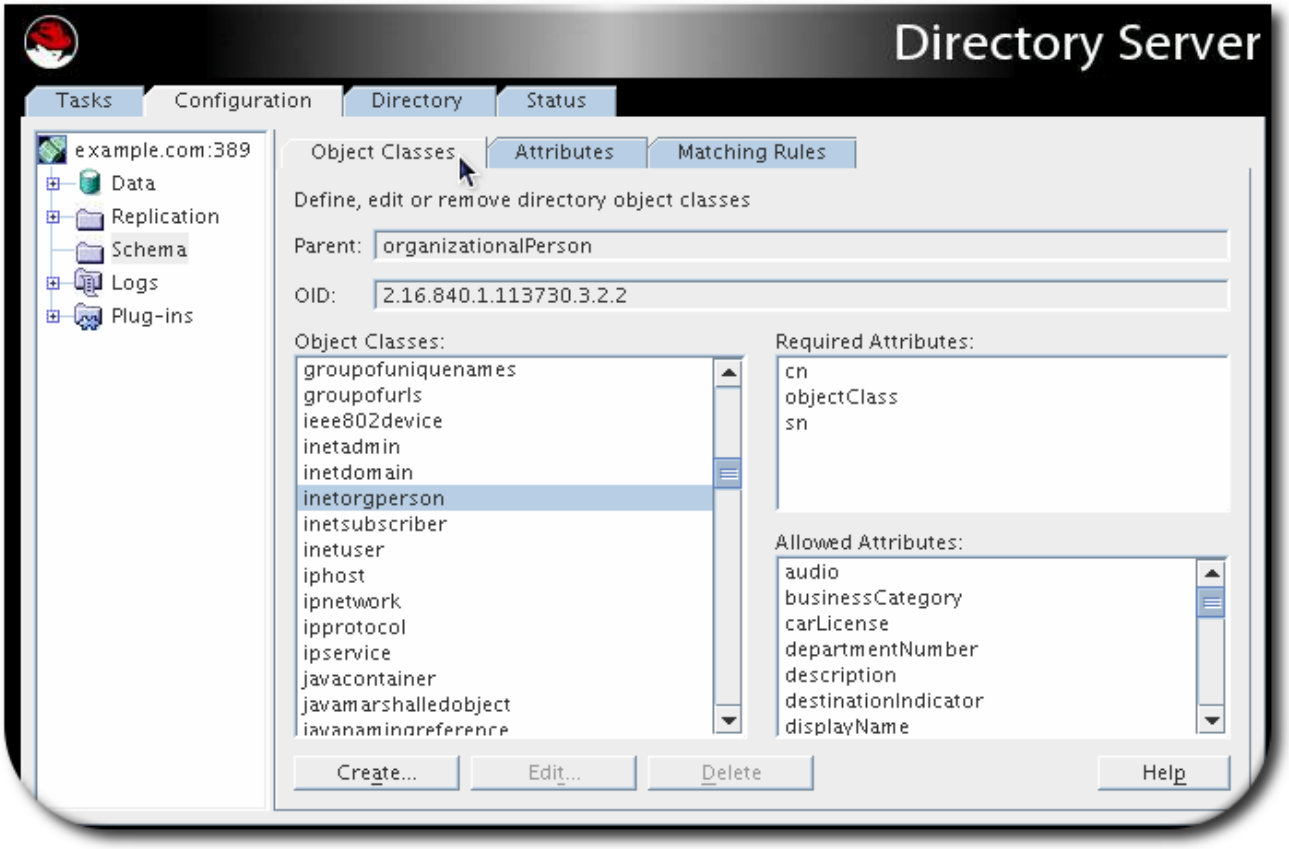


3. Directory Server に読み込まれているスキーマ要素を表示するタブには、オブジェクトクラス、属性、 および Matching Rules という 3 つのタブがあります。

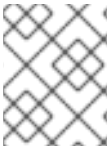
Attributes タブは、default 属性とカスタム属性の 2 つのセクションに分けられます。どちらのセクションも、属性名、OID、構文、および属性が多値であるかどうかを表示します。



Object Classes タブには、左側のオブジェクトクラスの一覧が表示されます。オブジェクトクラスが強調表示されると、その OID と superior オブジェクトクラスが top のフィールドと、必須属性および許可される属性と共に右側のボックスに一覧表示されます。



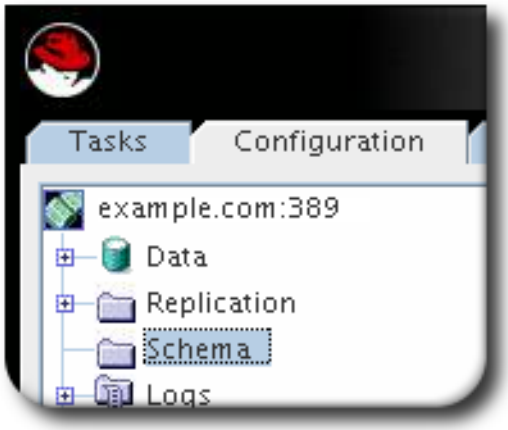
12.4.2. 属性の作成



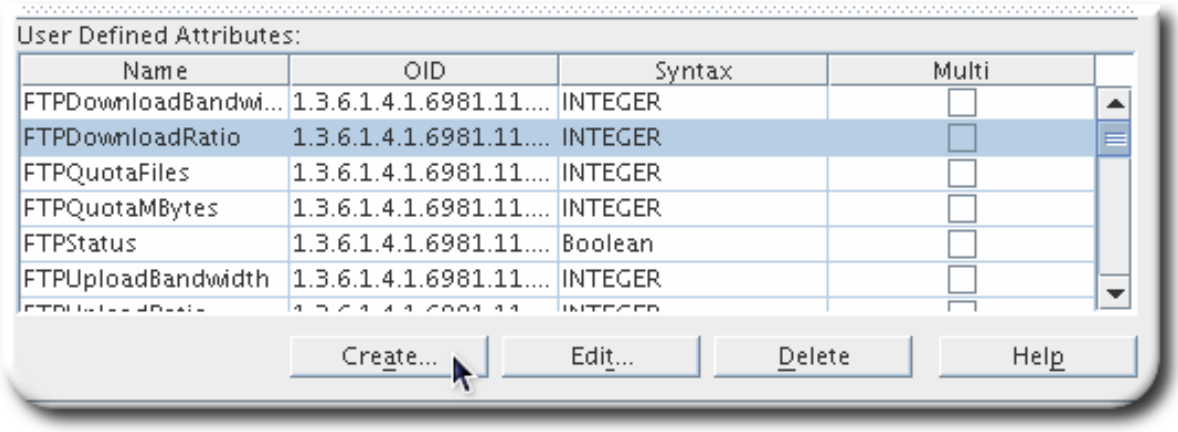
注記

スキーマに新しい属性を追加したら、「オブジェクトクラスの作成」 の説明に従って、それらのオブジェクトクラスを含む新しいオブジェクトクラスを作成します。

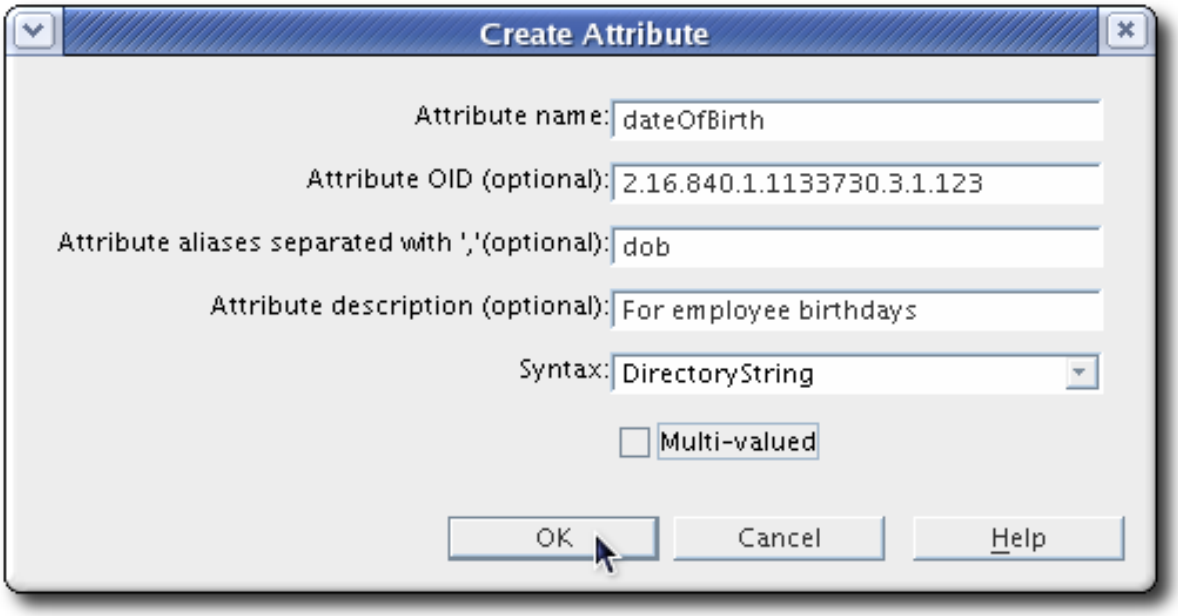
1. **Configuration タブを選択します。**
2. **左側のナビゲーションツリーで Schema フォルダーを選択し、右側のペインの Attributes タブを選択します。**



3. **作成 をクリックします。**



4. 新しい属性の情報を入力します。



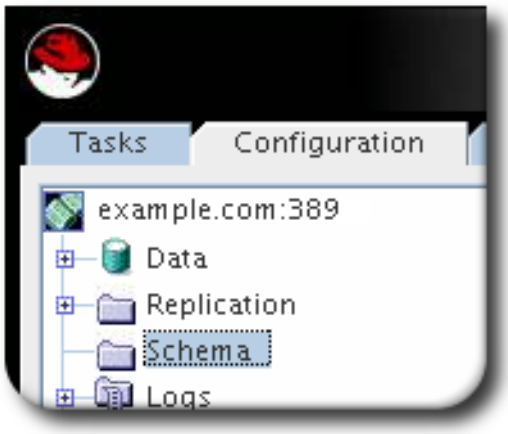
- 属性名。これは一意である必要があります。
- OID。これは必須ではありませんが、互換性およびサーバーパフォーマンスのために、一意の数値 OID を割り当てることが強く推奨されます。
- 構文。これは属性値で使用できる形式です。
- 属性が多値かどうか。デフォルトでは、すべての属性をエントリーで複数回使用できますが、チェックボックスの選択を解除すると、属性が一度だけ使用できることを意味します。

5. OK をクリックします。

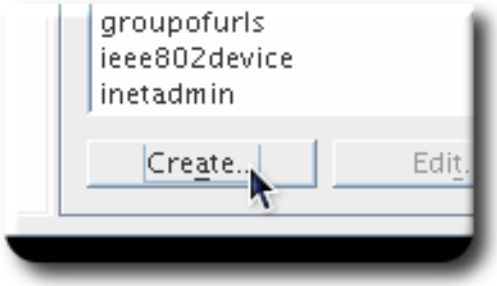
12.4.3. オブジェクトクラスの作成

新しいオブジェクトクラスは、一意の名前、親オブジェクト、および必須属性および任意の属性で作成する必要があります。オブジェクトクラスを作成するには、以下を実行します。

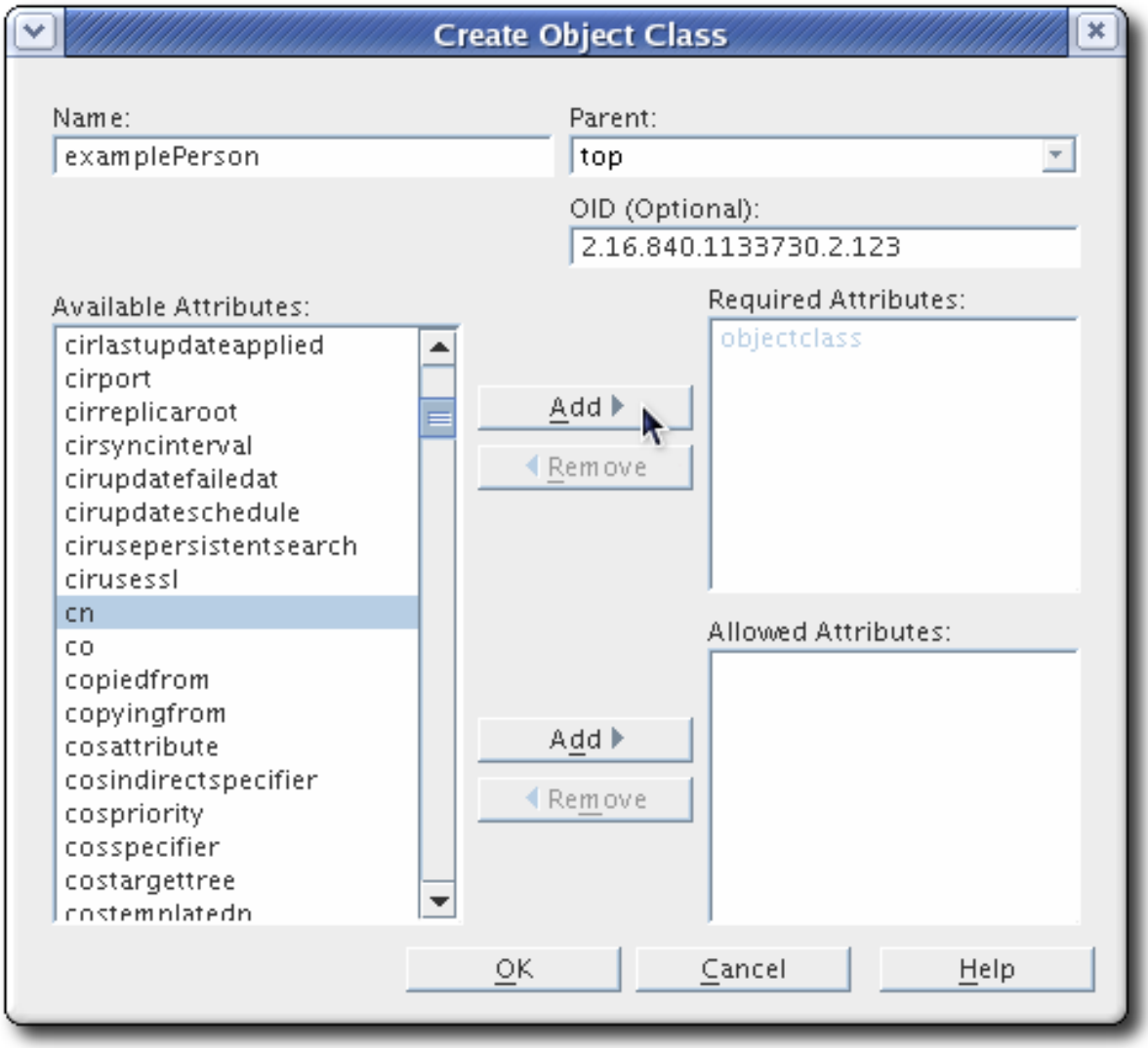
1. Directory Server コンソールで、**Configuration** タブを選択します。
2. ナビゲーションツリーで **Schema** フォルダーを選択し、右側のペインで **Object Classes** タブを選択します。



3. **Object Classes** タブの **Create** ボタンをクリックします。



4. 新規オブジェクトクラスの情報を入力します。



- 名前。これは一意である必要があります。
- OID。これは必須ではありませんが、互換性およびサーバーパフォーマンスのために、一意の数値 OID を割り当てることが強く推奨されます。
- エントリーの上位オブジェクトクラスデフォルトは top です。別のオブジェクトクラスを選択すると、新規オブジェクトクラスは、独自の定義された属性に加えて、親から必須属性および許可される属性をすべて継承します。
- 必須および許可される属性。左側の属性を選択し、Available Attributes および Required Attributes ボックス によって Add ボタンを使用して、属性を適宜追加します。

注記

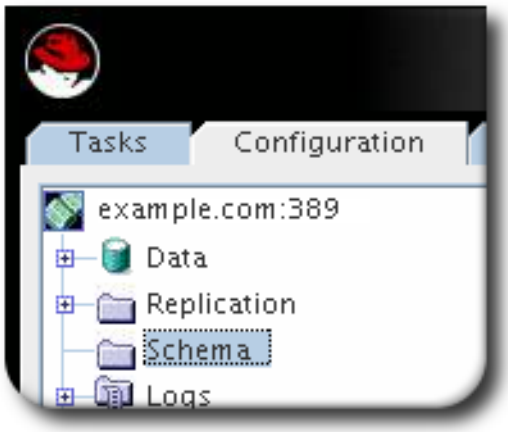
親オブジェクトクラスから継承される属性は、許可または拒否されるかどうかに関わらず、削除できません。

5. OK をクリックして、新規オブジェクトクラスを保存します。

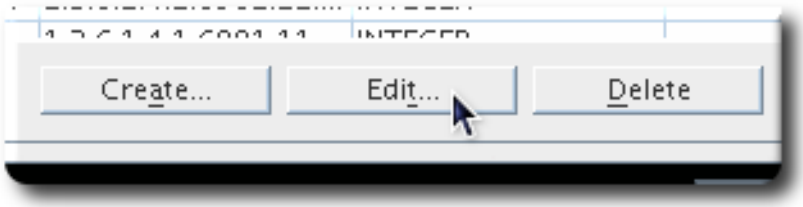
12.4.4. カスタムスキーマ要素の編集

ユーザーが作成した属性またはオブジェクトクラスのみを編集でき、標準のスキーマ要素を編集することはできません。

1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のナビゲーションツリーで **Schema** フォルダーを選択します。



3. **Object Classes** または **Attributes** タブを開きます。
4. リストから編集するスキーマ要素を選択します。カスタム（ユーザー定義の）スキーマのみが Directory Server コンソールで編集できます。
5. ウィンドウの下部にある **Edit** ボタンをクリックします。

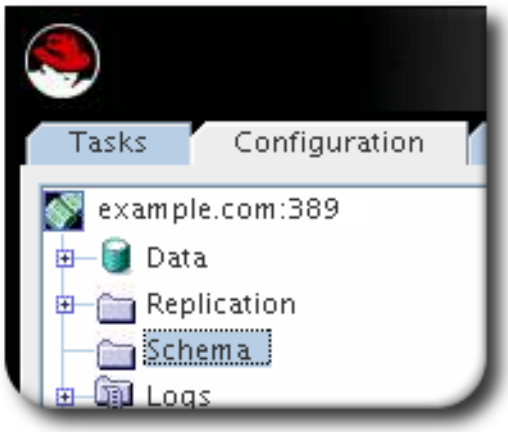


6. スキーマ情報を編集します。

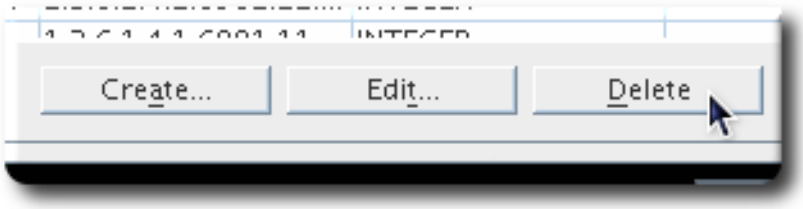
12.4.5. スキーマの削除

ユーザーが作成した属性またはオブジェクトクラスのみを削除できます。標準のスキーマ要素は削除できません。

1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のナビゲーションツリーで **Schema** フォルダーを選択します。



3. **Object Classes** または **Attributes** タブを開きます。
4. リストから削除するスキーマ要素を選択します。Directory Server コンソールでは、カスタム（ユーザー定義の）スキーマのみを削除できます。
5. ウィンドウの下部にある **Delete** ボタンをクリックします。



6. 削除を確認します。

警告

サーバーは即座に schema 要素を削除します。元に戻すことはできません。

12.5. LDAPMODIFY を使用したスキーマの管理

Directory Server コンソールと同様に、**ldapmodify** を使用してカスタムスキーマ要素を追加、編集、および削除できます。**ldapmodify** は、Directory Server インスタンス(99**user.ldif**)のデフォルトのカスタムスキーマファイルも変更します。

12.5.1. 属性の作成

カスタム属性エントリー自体は、**cn=schema** エントリーの ***attributetypes*** エントリーです。***attributetypes*** 属性の形式は以下のとおりです。

```
attributetypes: ( definition )
```

定義には 5 つのコンポーネントが含まれます。

- OID (通常はドット区切り番号)
- **NAME** 名前 形式の一意の名前
- **DESC** 説明 形式の説明
- 属性値の構文の OID。(**SYNTAX** OID 形式については「Directory Server 属性の構文」で説明)
- 任意で、属性が定義されているソース

LDAP コマンドを実行して **cn=schema** エントリーを変更することで、カスタムスキーマファイル **99user.ldif** に属性定義が追加されます。以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -x -v
```

```
dn: cn=schema
changetype: modify
add: attributetypes
attributetypes: ( 1.2.3.4.5.6.1 NAME 'dateofbirth' DESC 'For employee birthdays' SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUED X-ORIGIN 'Example defined')
```

12.5.2. オブジェクトクラスの作成

オブジェクトクラス定義は、cn=schema エントリーの *objectclasses* 属性です。*objectclasses* 属性の形式は以下のとおりです。

```
objectclasses: ( definition )
```

オブジェクトクラス定義には複数のコンポーネントが含まれます。

- **OID (通常はドット区切り番号)**
- **NAME 名前 形式の一意の名前**
- **DESC 説明 形式の説明**
- **SUP object_class の形式で、このオブジェクトクラスの上位または親のオブジェクトクラス。関連する親がない場合は、SUP top を使用してください。**
- **AUXILIARY という単語で、オブジェクトクラスを適用するエントリーのタイプを指定します。AUXILIARY は、任意のエントリーに適用できることを意味します。**
- **MUST の後に続く必要な属性のリスト。複数の属性を含めるには、グループを括弧で囲み、ドル記号 ($) で属性を区切ります。**
- **MAY の後に続く許可される属性のリスト。複数の属性を含めるには、グループを括弧で囲み、ドル記号 ($) で属性を区切ります。**

LDAP コマンドを実行して cn=schema エントリーを変更することで、オブジェクトクラスの定義がカスタムスキーマファイル 99user.ldif に追加されます。以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -v

dn: cn=schema
changetype: modify
add: objectclasses
objectclasses: ( 2.16.840.1133730.2.123 NAME 'examplePerson' DESC 'Example Person Object
Class' SUP inetOrgPerson AUXILIARY  MUST cn MAY (exampleDateOfBirth $
examplePreferredOS) )
```

12.5.3. スキーマの削除

警告

デフォルトのスキーマ要素 を削除しないでください。これらは、Directory Server での実行に必要です。

1. **不要な属性を使用するエントリーから、その属性を受け入れるスキーマファイルのオブジェクトクラスから削除します。同様に、オブジェクトクラスを削除するには、任意のエントリーから削除します。**

2. **ldapmodify を実行して属性を削除します。以下に例を示します。**

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=schema
changetype: modify
delete: objectclasses
objectclasses: ( 2.16.840.1133730.2.123 NAME 'examplePerson' DESC 'Example
Person Object Class' SUP inetOrgPerson AUXILIARY MUST cn MAY
(exampleDateOfBirth $ examplePreferredOS) )
```

警告

削除するオブジェクトクラスまたは属性を指定してください。値なしで *attributetypes* 属性または *objectclasses* 属性のみを使用すると、ファイル内のすべてのユーザー定義属性またはオブジェクトクラスが削除されます。

カスタム属性またはオブジェクトクラスが 99user.ldif 以外のカスタムスキーマファイルにある場合は、ファイルを直接編集します。Directory Server コンソールおよび LDAP ツールは 、99user.ldif 以外のスキーマファイルを編集できません。

12.6. カスタムスキーマファイルの作成

スキーマファイルは、cn=schema エントリーを定義する単純な LDIF ファイルです。各属性とオブジェクトクラスは、そのエントリーの属性として追加されます。スキーマファイルの作成要件を以下に示します。

- **最初の行は dn: cn=schema である必要があります。**
- **スキーマファイルには、属性とオブジェクトクラスの両方を含めることができますが、どちらか一方のみを含めることもできます。**
- **スタイルに属性とオブジェクトクラスの両方が定義されている場合は、最初にすべての属性がファイルに記載し、次にオブジェクトクラスを記載する必要があります。**

- オブジェクトクラスは、他のスキーマファイルで定義された属性を使用できます。
- このファイルは、[1-9][0-9]text.ldif の形式で指定する必要があります。

  このファイルは、常に 2 つの数字で開始する必要があります。数値的には、コア設定スキーマ (00 および 01) の前にスキーマファイルを読み込ませることができません。

  また、Directory Server は、常に、そのカスタムスキーマをスキーマディレクトリー内の数値およびアルファベット順で最も高い名前のスキーマファイルに書き込みます。このファイルは、99user.ldif であることを想定しています。このファイルが 99user.ldif ではない場合には、サーバーで問題が発生する可能性があります。そのため、常に、カスタムスキーマファイルが、少なくともアルファベット順で 99user.ldif よりも低くなることを確認します。名前 99alpha.ldif は問題ではありません。99zzz.ldif 名前は問題です。

スキーマファイル作成のプラクティスは、『デプロイメントガイド』 を参照してください。

属性は、スキーマへの *attributetypes* 属性として、5 つのコンポーネントがあるスキーマファイルで定義されます。

- OID (通常はドット区切り番号)
- NAME 名前 形式の一意の名前
- DESC 説明 形式の説明
- 属性値の構文の OID。(SYNTAX OID 形式については「Directory Server 属性の構文」で説明)
- 任意で、属性が定義されているソース

以下に例を示します。

```
attributetypes: ( 1.2.3.4.5.6.1 NAME 'dateofbirth' DESC 'For employee birthdays' SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUED X-ORIGIN 'Example defined')
```

同様に、オブジェクトクラスは *objectclasses* 属性の値として定義されますが、オブジェクトクラスの定義方法には若干柔軟性があります。必要な設定は、オブジェクトクラスの名前と OID のみになります。他のすべての設定は、オブジェクトクラスのニーズに依存します。

- OID (通常はドット区切り番号)
- NAME 名前 形式の一意の名前
- DESC 説明 形式の説明
- SUP object_class の形式で、このオブジェクトクラスの上位または親のオブジェクトクラス。関連する親がない場合は、SUP top を使用してください。
- AUXILIARY という単語で、オブジェクトクラスを適用するエントリーのタイプを指定します。AUXILIARY は、任意のエントリーに適用できることを意味します。
- MUST の後に続く必要な属性のリスト。複数の属性を含めるには、グループを括弧で囲み、ドル記号 ($) で属性を区切ります。
- MAY の後に続く許可される属性のリスト。複数の属性を含めるには、グループを括弧で囲み、ドル記号 ($) で属性を区切ります。

以下に例を示します。

```
objectclasses: ( 2.16.840.1133730.2.123 NAME 'examplePerson' DESC 'Example Person Object Class' SUP inetOrgPerson AUXILIARY  MUST cn MAY (exampleDateOfBirth $ examplePreferredOS) )
```

例12.4「スキーマファイルの例」は、簡潔なスキーマファイルを示しています。

例12.4 スキーマファイルの例

```
dn: cn=schema
attributetypes: ( 2.16.840.1133730.1.123 NAME 'dateofbirth' DESC 'For employee birthdays' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN 'Example defined')
objectclasses: ( 2.16.840.1133730.2.123 NAME 'examplePerson' DESC 'Example Person Object Class' SUP inetOrgPerson AUXILIARY MAY (dateofbirth) )
```

カスタムスキーマファイルは Directory Server インスタンスのスキーマディレクトリー /etc/dirsrv/slapd-*instance*/schema に追加する必要があります。サーバーが再起動するか、動的に再読み込みされたタスクが実行されない限り、これらのファイルのスキーマは読み込まれず、サーバーで使用できなくなります。

重要

/usr/share/data/ ディレクトリーから標準スキーマを使用する場合は、スキーマファイルを /usr/share/dirsrv/schema/ ディレクトリーにコピーします。標準スキーマが特定のインスタンスでのみ利用できるようにする必要がある場合は、スキーマファイルを /etc/dirsrv/slapd-*instance_name*/schema/ ディレクトリーにコピーしますが、宛先ディレクトリーで別のファイル名を使用します。それ以外の場合は、Directory Server はアップグレード中にファイルの名前を変更し、.bak 接尾辞を追加します。

12.7. スキーマの動的再読み込み

デフォルトでは、Directory Server インスタンスが使用するスキーマファイルが、起動時にディレクトリーに読み込まれます。つまり、サーバーが再起動しない限り、スキーマディレクトリーに追加された新しいスキーマファイルが使用できません。Directory Server には、サーバーの再起動を必要とせずに、カスタムファイルを含む Directory Server インスタンスの完全なスキーマを手動で再読み込みするタスクがあります。

スキーマ再読み込みタスクは、以下の 2 つの方法で開始できます。

- schema-reload.pl スクリプトの使用
- ldapmodifyを使用した cn=schema 再読み込みタスク エントリーの追加

12.7.1. schema-reload.pl を使用したスキーマの再読み込み

schema-reload.pl スクリプトは特殊なタスクを起動し、特定の Directory Server インスタンスが使用するスキーマファイルをすべて再読み込みします。これにより、スキーマ要素を 99user.ldif に追加しなくても、カスタムスキーマファイルを動的に読み込むことができます。

1. Directory Manager としてスクリプトを実行し、バインドします。

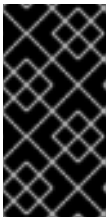
```
# schema-reload.pl -Z instance_name -D "cn=Directory Manager" -w secret
```

Directory Server は、新しい再読み込みタスクエントリーが追加されたことを伝えます。

```
adding new entry cn=schema_reload_2009_1_6_17_52_4,cn=schema reload
task,cn=tasks,cn=config
```

これにより、デフォルトのスキーマディレクトリー /etc/dirsrv/slapd-*instance*/schema からスキーマが再読み込みされます。-d オプションを使用して別のディレクトリーを指定することもできます。

```
# schema-reload.pl -Z instance_name-D "cn=Directory Manager" -w password -d
/export/custom-schema
```



重要

Directory Server スキーマ再読み込みタスクは、*schemadir* パラメーターで指定したディレクトリーからスキーマファイルを再読み込みします。さらに、サーバーは //usr/share/dirsrv/schema ディレクトリーからすべてのスキーマファイルを読み込みます。

schema-reload.pl は、設定、『コマンド、およびファイルリファレンス』 で詳細に説明されています。

12.7.2. ldapmodify を使用したスキーマの再読み込み

schema-reload.pl スクリプトは、スキーマファイルを再読み込みする Directory Server インスタンスに特別なタスクエントリーを作成します。また、タスクエントリーを直接作成することでスキーマをリロードすることもできます。タスクエントリーは、dse.ldif ファイルの cn=tasks 設定エントリーで発生するため、ldapmodify を使用してエントリーを追加してタスクを開始することもできます。タスクが完了するとすぐに、エントリーはディレクトリーから削除されます。

スキーマ再読み込みタスクを開始するには、cn=schema reload task,cn=tasks,cn=config エントリーの下にエントリーを追加します。必要な属性は、特定タスクの *cn* のみです。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=example schema reload,cn=schema reload task,cn=tasks,cn=config
changetype: add
objectclass: extensibleObject
cn:example schema reload
```

Directory Server インスタンスがスキーマを再読み込みするデフォルトのスキーマディレクトリーは、/usr/share/dirsrv/schema にあります。これは、*schemadir* 属性を使用して別のスキーマディレクトリーを指定できます。これは、schema -reload.pl の -d オプションに似ています。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=example schema reload,cn=schema reload task,cn=tasks,cn=config
changetype: add
objectclass: extensibleObject
cn:example schema reload
schemadir: /etc/dirsrv/slapd-instance_name/schema/
```

重要

Directory Server スキーマ再読み込みタスクは、*schemadir* パラメーターで指定したディレクトリーからスキーマファイルを再読み込みします。さらに、サーバーは //usr/share/dirsrv/schema ディレクトリーからすべてのスキーマファイルを読み込みます。

タスクが完了するとすぐに、エントリーは dse.ldif 設定から削除 されるため、同じタスクエントリーを継続的に再利用できます。

cn=schema 再読み込みタスク 設定は、『設定、コマンド、およびファイルリファレンス を参照してください』。

12.7.3. レプリケーションによるスキーマの再読み込み

スキーマの再読み込みタスクはローカル操作であるため、スキーマが 1 つのサプライヤーに追加され、他のサプライヤーに追加されない場合は、スキーマの変更がマルチマスター環境で複製されません。全サプライヤーサーバーに新しいスキーマファイルを読み込むには、次のコマンドを実行します。

1. レプリケーションを停止します。
2. 新しいスキーマファイルをコピーし、各サプライヤーおよびレプリカサーバーに対してスキーマ再読み込みタスクを実行します。
3. レプリケーションを再起動します。

12.7.4. スキーマの再読み込みエラー

スキーマ再読み込みタスクが実行されると、コマンドプロンプトにタスクが開始されることのみが表示されます。

```
adding new entry cn=schema reload task 1,cn=schema reload task,cn=tasks,cn=config
```

ただし、タスクは、正常に完了するかどうかを返しません。スキーマ再読み込み操作が正常に行われたことを確認するには、エラーログを確認します。スキーマの再読み込みには、最初にスキーマファイルを検証して読み込む 2 つのタスクがあります。

成功メッセージは、検証が渡され、タスクが完了したことを示しています。

```
[06/Jan/2009:17:52:04.001214874 -0500] schemareload - Schema reload task starts (schema
dir: default) ...
[06/Jan/2009:17:52:04.754335904 -0500] schemareload - Schema validation passed.
[06/Jan/2009:17:52:04.894255328 -0500] schemareload - Schema reload task finished.
```

失敗したステップがある場合は、失敗したステップおよびその理由がログに表示されます。

```
[..] schemareload - Schema reload task starts (schema dir: /bogus) ...
[..] schema - No schema files were found in the directory /bogus
[..] schema_reload - schema file validation failed
[..] schemareload - Schema validation failed.
```

12.8. スキーマチェックのオンとオフを切り替える

スキーマチェックがオンの場合、Directory Server では以下の 3 つの内容が確保されます。

- 使用するオブジェクトクラスおよび属性はディレクトリースキーマで定義されます。
- オブジェクトクラスに必要な属性はエントリーに含まれます。
- オブジェクトクラスで使用できる属性のみがエントリーに含まれます。

重要

Red Hat は、スキーマチェックを無効にしないことを推奨します。

Directory Server では、スキーマチェックはデフォルトで有効になっており、Directory Server は常にスキーマチェックが有効な状態で実行します。スキーマの確認をオフにしておくことは LDAP インポート操作を迅速化するのが唯一の状況です。ただし、スキーマに準拠していないエントリーをインポートするリスクがあります。したがって、このエントリーを更新することはできません。

12.8.1. コマンドラインでスキーマチェックのオンおよびオフを切り替え

LDAP コマンドを使用してスキーマチェックをオンおよびオフにするには、***nsslapd-schemacheck*** 属性の値を編集します。スキーマの確認を無効にするには、次のコマンドを実行します。

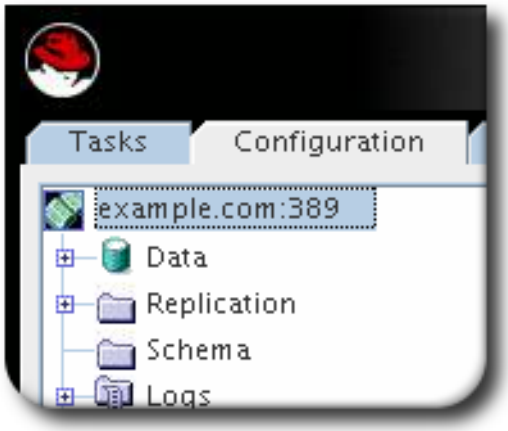
```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=config
changetype: modify
replace: nsslapd-schemacheck
nsslapd-schemacheck: off
```

nsslapd-schemacheck パラメーターの詳細は、Red『Hat Directory Server の設定、コマンド、およびファイルリファレンス で、パラメーターの説明を参照してください』。

12.8.2. コンソールを使用したスキーマチェックのオンおよびオフの切り替え

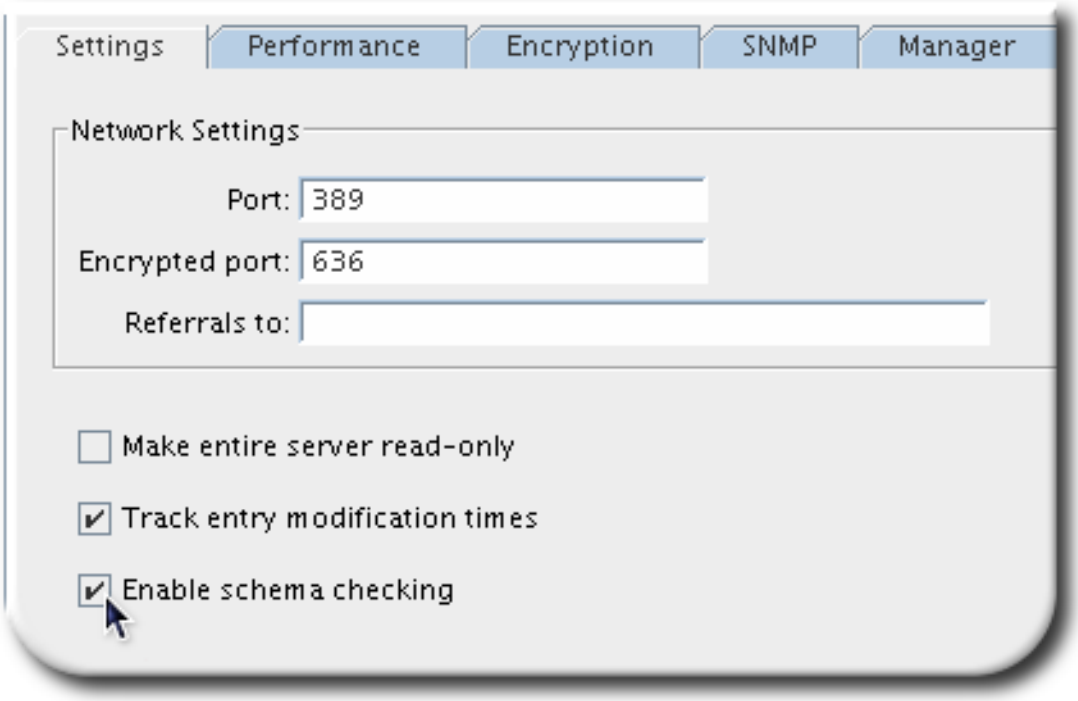
コンソールを使用してスキーマチェックを有効または無効にするには、以下を行います。

1. Directory Server コンソールで、**Configuration** タブを選択します。



2. ナビゲーションツリーの上部にあるサーバーアイコンを強調表示し、右側のペインの **Settings** タブを選択します。

3. スキーマチェックを有効にするには、**Enable Schema Checking** チェックボックスにチェックを入れます。スキーマチェックをオフにします。



4. **Save** をクリックします。

12.9. 構文の検証の使用

構文の検証では、Directory Server は、属性の値が、その属性の定義に指定された構文のルールに従うことを確認します。たとえば、構文の検証では、新しい *telephoneNumber* 属性に、実際にその値に有効な電話番号が指定されていることを確認します。

重要

Red Hat は、構文の検証を無効にしないことを推奨します。

12.9.1. 構文の検証の概要

スキーマチェックと同様に、検証によりディレクトリーの変更がレビューされ、構文に違反する変更を拒否します。オプションとして追加の設定を行い、構文検証により構文違反に関する警告メッセージをログに記録し、変更を拒否したり、変更プロセスを正常に実行できるようにしたりすることもできます。

この機能は、バイナリー構文 (検証できない) および標準以外の構文 (定義された必要な形式がない) を除き、すべての属性構文を検証します。構文は RFC 4514 に対して検証されます。

12.9.2. 構文の検証およびその他の Directory Server 操作

構文の検証は、エントリーの作成 (add) や属性の編集 (modify) などの標準の LDAP 操作に主に関係します。ただし、属性の構文の検証は他の Directory Server 操作に影響を及ぼす可能性があります。

データベース暗号化

通常の LDAP 操作では、値がデータベースに書き込まれる直前に属性は暗号化されます。これは、属性構文の検証 後に 暗号化が実行されることを意味します。

暗号化されたデータベース (「10章*属性暗号化の設定*」で説明) をエクスポートおよびインポートすることができます。通常、これらのエクスポート操作およびインポート操作は **db2ldif** および **ldif2db** と共に **-E** フラグを使用して行うことが強く推奨されます。これにより、インポート操作で構文の検証が問題になる可能性もあります。ただし、**-E** フラグを使用せずに暗号化されたデータベースをエクスポートする場合は (サポートされていない)、暗号化された値で LDIF が作成されます。この LDIF をインポートすると、暗号化された属性を検証できず、警告がログに記録され、インポートされたエントリーで属性検証はスキップされます。

同期

Windows Active Directory エントリーと Red Hat Directory Server エントリーでは、属性の許容構文または強制構文に違いがある場合があります。この場合、構文の検証により Directory Server エントリーの RFC 標準が強制されるため、Active Directory の値を適切に同期できませんでした。

レプリケーション

Directory Server 10.6 インスタンスがその変更をコンシューマーに複製するサプライヤーである場合は、構文検証を使用した問題はありません。ただし、レプリケーションのサプライヤーが古いバージョンの Directory Server であったり、構文の検証が無効になっていたりする場合は、Directory Server 10.6 コンシューマーはマスターが許可する属性値を拒否する可能性があるため、構文の検証をコンシューマーで使用しないでください。

12.9.3. 構文の検証の有効化または無効化

構文の検証は *nsslapd-syntaxcheck* 属性で設定されます。この属性の値は on または off（デフォルトではオン）です。構文の検証を変更するには、**ldapmodify** を使用するか、**dse.ldif** ファイルを直接編集してこの属性を変更します。

```
# ldapmodify -D "cn=Directory Manager" -W -x

dn: cn=config
changetype: modify
replace: nsslapd-syntaxcheck
nsslapd-syntaxcheck: off
```

注記

構文の検証が無効になっている場合は、構文の検証を再度有効にする前に、**syntax-validate.pl** スクリプトを実行して既存の属性値を監査します。「既存の属性値の構文の検証」 を参照してください。

12.9.4. DN の厳格な構文検証の有効化

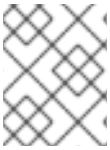
構文の検証が有効な場合、DN は他の属性構文と同様に RFC 4514 に対して検証されます。ただし、DN 構文の検証は、後の標準の厳格さが古いスタイルの DN やディレクトリーツリーを無効にする可能性があるため、個別に有効になります。

構文の検証では、RFC 4514 のセクション 3 に対して DN を確認します。

この属性の値は on または off（デフォルトでは off）です。構文の検証を変更するには、**ldapmodify** を使用するか、**dse.ldif** ファイルを直接編集してこの属性を変更します。

```
# ldapmodify -D "cn=Directory Manager" -W -x

dn: cn=config
```

```
changetype: modify
replace: nsslapd-dn-validate-strict
nsslapd-dn-validate-strict: on
```

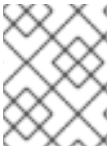


注記

厳密な DN 検証が有効になり、DN 値が必要な構文に準拠しない場合は、LDAP 結果コード 34、INVALID_DN_SYNTAX で操作に失敗します。

12.9.5. 構文検証警告の有効化(Logging)

デフォルトでは、構文の検証は、属性値が必要な構文に違反する追加または変更の操作を拒否します。ただし、違反自体は、デフォルトでは errors ログに記録されません。*nsslapd-syntaxlogging* 属性は、構文違反のエラーロギングを有効にします。



注記

構文検証スクリプトおよびタスクによって検出された構文違反は、Directory Server エラーログに記録されます。

nsslapd-syntaxlogging と *nsslapd-syntaxcheck* の両方を有効にすると、無効な属性の変更が拒否され、メッセージがログに書き込まれます。*nsslapd-syntaxlogging* が有効ですが *nsslapd-syntaxcheck* が無効の場合、操作は成功できますが、警告メッセージがエラーログに書き込まれます。

この属性の値は on または off（デフォルトでは off）です。構文検証ロギングを有効にするには、ldapmodify を使用して属性を編集するか、または dse.ldif ファイルを直接編集します。

```
# ldapmodify -D "cn=Directory Manager" -W -x

dn: cn=config
changetype: modify
replace: nsslapd-syntaxlogging
nsslapd-syntaxlogging: on
```

12.9.6. 既存の属性値の構文の検証

特定の状況では、既存の値の構文を手動で検証したい場合があります。以下に例を示します。

- *nsslapd-syntaxcheck* パラメーターで構文の検証が無効になっている場合。詳細は、「構文の検証の有効化または無効化」 を参照してください。

  重要

  Red Hat は、構文の検証を無効にしないことを推奨します。

- 構文検証なしまたは無効化されたサーバーからデータを移行する場合。

(objectclass=inetorgperson)フィルターに一致する ou=people,dc=example,dc=com サブツリーのすべての値の構文を検証するタスクを作成するには、以下を実行します。

```
# syntax-validate.pl -D "cn=Directory Manager" -w secret \
    -b "ou=people,dc=example,dc=com" -f "(objectclass=inetorgperson)"
```

```
ldap_initialize( ldap://server.example.com:389 )
Successfully added task entry "cn=syntax_validate_2017_7_3_10_52_47, cn=syntax validate,
cn=tasks, cn=config"
```

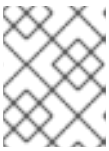
Directory Server は、結果を /var/log/dirsrv/slapd-*instance_name*/errors ファイルに記録します。以下に例を示します。

- 検証済みの値がすべて有効であれば、以下を実行します。

```
[28/Jun/2017:12:52:43.669867966 +0200] - ERR - syntax-plugin -
syntax_validate_task_thread - Starting (base: "dc=example,dc=com", filter: "
(objectclass=*)") ...
[28/Jun/2017:12:52:43.696850129 +0200] - ERR - syntax-plugin -
syntax_validate_task_thread - Complete.  Found 0 invalid entries.
```

- 無効なエントリーが見つかった場合は、以下を行います。

```
[28/Jun/2017:12:54:05.736087520 +0200] - ERR - syntax-plugin -
syntax_validate_task_thread - Starting (base: "dc=example,dc=com", filter: "
(objectclass=*)") ...
[28/Jun/2017:12:54:05.754195607 +0200] - ERR - syntax-plugin -
syntax_validate_task_callback - Entry "cn=user,ou=People,dc=example,dc=com"
violates syntax.
description: value #0 invalid per syntax
[28/Jun/2017:12:54:05.759905671 +0200] - ERR - syntax-plugin -
syntax_validate_task_thread - Complete.  Found 1 invalid entries.
```



注記

syntax-validate.pl スクリプトは、構文違反のみを識別します。誤った値を手動で修正する必要があります。

第13章 インデックスの管理

インデックス付けは、属性または値を分類および整理することにより、情報の検索と取得を容易にします。本章では、検索アルゴリズム自体、コンテキストにインデックスのメカニズムを配置し、インデックスを作成、削除、および管理する方法を説明します。

13.1. インデックスの概要

本セクションでは、Directory Server でのインデックスの概要を説明します。これには、以下のトピックが含まれます。

- 「インデックスタイプの概要」
- 「デフォルトインデックスおよびデータベースインデックスの概要」
- 「検索アルゴリズムの概要」
- 「インデックスのメリットとのバランス」

13.1.1. インデックスタイプの概要

インデックスはディレクトリーのデータベースにあるファイルに保存されます。ファイルの名前は、インデックス化された属性に基づいて、ファイルに含まれるインデックスの型は生成されません。各インデックスファイルには、特定の属性に対して複数のインデックスが保持されると、複数のインデックスが含まれる場合があります。たとえば、共通の name 属性用に保持されるすべてのインデックスは cn.db ファイルに含まれます。

Directory Server は、以下のタイプのインデックスをサポートします。

- Presence index (pres) には、特定の属性を含むエントリーの一覧が含まれており、検索には非常に便利です。たとえば、アクセス制御情報を含むエントリーを簡単に検証できます。プレゼンスインデックスを含む aci.db ファイルを生成すると、ACI=* の検索を効率的に実行して、サーバーのアクセス制御リストを生成します。

- Equality index (eq) により、特定の属性値を含むエントリーの検索が改善されます。たとえば、*cn* 属性の等価インデックスを使用すると、ユーザーは cn=Babs Jensen の検索をより効率的に実行できます。

- Approximate index (approx) は、効率的な近似検索や sounds-like 検索に使われます。たとえば、エントリーには属性値 cn=Robert E Lee を含めることができます。概算検索では、cn~=Robert Lee、cn~=Robert、または cn~= Lee に対する検索でこの値が返されます。同様に、l~=San Fransisco (スペルミスに注意) を検索すると、l=San Francisco を含むエントリーが返されます。

- Substring index (sub) は、維持するコストのかかるインデックスですが、エントリー内の部分文字列に対して効率的な検索が可能になります。部分文字列のインデックスは、各エントリーの最小 3 文字に制限されます。

  たとえば、cn=*derson の形式で検索すると、Bill Anderson、Jill Henderson、または Steve Sanderson といった文字列が含まれる共通名と一致します。同様に、telephoneNumber=*555* の検索は、555 が含まれる電話番号を持つディレクトリー内の全エントリーを返します。

- 国際インデックス は、国際ディレクトリー内の情報の検索を迅速化します。国際インデックスの作成プロセスは、通常のインデックスを作成するプロセスと似ています。ただし、オブジェクト識別子 (OID) をインデックス化する属性に関連付けることで一致するルール を適用する点

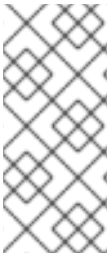
が異なります。

サポートされるロケールおよび関連付けられた OID が「付録D *国際化*」に一覧表示されています。追加のマッチングルールを受け入れるように Directory Server を設定する必要がある場合は、Red Hat コンサルティングにお問い合わせください。

- 参照インデックスまたは 仮想リストビュー(VLV)インデックスは、Directory Server コンソールのエントリーの表示を迅速化します。このインデックスは、ディレクトリーのブランチに数百のエントリーが含まれている場合に便利です（例： **ou=people** ブランチ）。ディレクトリーツリーの任意のブランチポイントに参照インデックスを作成して、Directory Server Console または **vlvindex** コマンドラインツールを使用して表示パフォーマンスを向上させることができます。これは、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス で説明されています。』

13.1.2. デフォルトインデックスおよびデータベースインデックスの概要

Directory Server には、一連のデフォルトインデックスが含まれます。新規データベースの作成時に、Directory Server はこれらのデフォルトインデックスを **cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config** から新規データベースにコピーします。次に、データベースはこれらのインデックスのコピーのみを使用します。このインデックスは **cn=index,cn=*database_name*,cn=ldbm database,cn=plugins,cn=config** に保存されます。



注記

Directory Server は cn=config エントリーの設定を複製しません。したがって、レプリケーショントポロジーの一部であるサーバーでは、インデックスを異なる方法で設定できます。たとえば、レプリケーションがカスケードする環境では、クライアントがハブからデータを読み取らない場合は、ハブにカスタムインデックスを作成する必要はありません。

Directory Server のデフォルトインデックスを表示するには、以下を実行します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com \
    -b "cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config" \
    '(objectClass=nsindex)'
```

注記

cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config に保存されているデフォルトのインデックス設定を更新しても、変更は **cn=index,cn=*database_name*,cn=ldbm database,cn=plugins,cn=config** の個々のデータベースには適用されません。

個別のデータベースのインデックスを表示するには、次のコマンドを実行します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com \
    -b "cn=index,cn=database_name,cn=ldbm database,cn=plugins,cn=config" \
    '(objectClass=nsindex)'
```

13.1.3. 検索アルゴリズムの概要

インデックスを使用して検索を迅速化します。ディレクトリーがインデックスをどのように使用するかを理解するには、検索アルゴリズムを理解するのに役立ちます。各インデックスには、属性の一覧（

cn、共通名、属性）、および各値に対応するエントリーへのポインターが含まれます。Directory Server は、以下のように検索要求を処理します。

1. LDAP クライアントアプリケーションは、検索要求をディレクトリーに送信します。
2. ディレクトリーは、受信要求を調べて、指定したベース DN が、1つ以上のデータベースまたはデータベースリンクに含まれるサフィックスと一致することを確認します。
   - 一致する場合には、ディレクトリーはリクエストを処理します。
   - 一致しない場合、ディレクトリーはサフィックスと一致しないことを示すエラーをクライアントに返します。**cn=config** 下の **nsslapd-referral** 属性で参照を指定している場合、ディレクトリーは、クライアントがリクエストを購入できる LDAP URL も返します。
   - Directory Server は、どのインデックスが適用されるかを確認するための検索フィルターを調べ、フィルターを満たす各インデックスからエントリー ID の一覧を読み込もうとします。ID リストは、フィルターで使用されている AND または OR に参加するかによって組み合わせられます。
   - エントリー ID の一覧が設定された ID リストのスキャン制限よりも大きい場合や、インデックスがない場合、Directory Server はデータベースのすべてのエントリーを検索します。これは インデックスのない 検索です。
3. Directory Server は、ID リストのすべてのエントリー ID について、**id2entry.db** データベースまたはエントリーキャッシュから (またはインデックスなしの検索の場合はデータベース全体から) すべてのエントリーを読み取ります。その後、サーバーはエントリーをチェックして、検索フィルターと一致するかどうかを確認します。それぞれの一致が見つかったら、それが返されます。

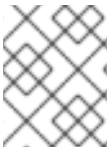
   サーバーは、すべての候補エントリーを検索するか、設定されたリソース制限に達するまで、ID のリストを検索し続けます。(リソース制限は「コマンドラインを使用したユーザーおよびグローバルリソース制限の設定」に一覧表示されます。)

注記

簡単なページ化された結果制御を使用して、検索に対して別のリソース制限を設定できます。たとえば、管理者は、高いサイズまたは無制限サイズを設定し、ページ化された検索で制限を検索しますが、ページのない検索には低いデフォルト制限を使用します。

13.1.4. おおよその検索

また、このディレクトリーでは、metaphone 表音アルゴリズムのバリエーションを用いて、近似的なインデックスで検索を行っています。各値は単語のシーケンスとして処理され、各単語について電話番号が生成されます。



注記

Directory Server の metaphone 表音アルゴリズムは US-ASCII 文字のみをサポートします。したがって、インデックスは、英語の値でのみ使用してください。

概算検索に入力した値は、表音コードシーケンスに変換されます。以下の両方が当てはまる場合、エントリーはクエリーに一致すると考えられます。

- すべてのクエリー文字列コードは、エントリー文字列に生成されたコードと一致します。

- クエリー文字列コードはすべて、エントリー文字列コードと同じ順序で実行されます。

ディレクトリーの名前 (フォネティックコード)	クエリー文字列 (フォネティックコード)	一致のコメント
Alice B Sarette (ALS B SRT)	Alice Sarette (ALS SRT)	一致。コードが正しい順序で指定されます。
	Alice Sarrette (ALS SRT)	一致。コードは、Sarette が間違っているにもかかわらず、正しい順序で指定されます。
	Surette (SRT)	一致。生成されたコードは、Sarette のスペルが間違っているにもかかわらず、元の名前で存在しています。
	Bertha Sarette (BR0 SRT)	一致するものはありません。コード BR0 は元の名前に存在しません。
	Sarette, Alice (SRT ALS)	一致するものはありません。コードが正しい順序で指定されていません。

13.1.5. インデックスのメリットとのバランス

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ことのメリットとコストのバランスを考えてください。

- 概算インデックスは、通常、数字を含む属性 (電話番号など) には効率的ではありません。
- 部分文字列のインデックスはバイナリー属性では機能しません。
- 等価インデックスは、値が大きい場合に使用しないようにしてください (例: 暗号化データを含む写真やパスワードを含む属性など)。
- 検索であまり使用されない属性のインデックスを維持することは、グローバル検索のパフォーマンスを向上させることなく、オーバーヘッドを増加させます。
- インデックス化されていない属性は、検索要求で依然として指定できますが、検索のタイプによっては検索パフォーマンスが大幅に低下する可能性があります。
- 保守するインデックスが多いほど、必要なディスク領域が多くなります。

インデックスは、非常に時間がかかります。以下に例を示します。

1. Directory Server は add 操作または modify 操作を受け取ります。
2. Directory Server は indexing 属性を調べ、属性値に対してインデックスが維持されているかどうかを判断します。

3. 作成した属性値がインデックス化されると、Directory Server は新しいインデックスエントリーを生成します。

4. サーバーがインデックス作成を完了すると、クライアント要求に応じて実際の属性値が作成されます。

たとえば、Directory Server はエントリーを追加します。

```
dn: cn=John Doe,ou=People,dc=example,dc=com
objectclass: top
objectClass: person
objectClass: orgperson
objectClass: inetorgperson
cn: John Doe
cn: John
sn: Doe
ou: Manufacturing
ou: people
telephoneNumber: 408 555 8834
description: Manufacturing lead for the Z238 line of widgets.
```

Directory Server は以下のインデックスを維持します。

- ***cn*** (一般名 (common name)) 属性および ***sn*** (姓 (surname)) 属性の等価、概算、および部分文字列インデックス。
- 電話番号属性の等価および部分文字列のインデックス。
- 説明属性の部分文字列インデックス。

そのエントリーをディレクトリーに追加する場合は、Directory Server で以下の手順を実行する必要があります。

1. John および John Doe の ***cn*** 等価インデックスエントリーを作成します。
2. John および John Doe の適切な ***cn*** の概算インデックスエントリーを作成します。
3. John および John Doe の適切な ***cn*** 部分文字列インデックスエントリーを作成します。
4. Doe の ***sn*** 等価インデックスエントリーを作成します。
5. Doe に対する ***sn*** 概算インデックスエントリーを作成します。
6. Doe に適切な ***sn*** 部分文字列インデックスエントリーを作成します。
7. 408 555 8834 に電話番号の等価インデックスエントリーを作成します。
8. 408 555 8834 に適切な電話番号の部分文字列インデックスエントリーを作成します。
9. Manufacturing lead for the Z238 line of widgets の適切な説明部分文字列インデックスエントリーを作成します。この文字列に対して多数の部分文字列エントリーが生成されます。

この例が示すように、大規模なディレクトリーのデータベースの作成および維持に必要なアクションの数は、リソースを必要とします。

13.1.6. インデックスの制限

nsrole や *cos_attribute* などの仮想属性をインデックス化できません。仮想属性には計算値が含まれます。これらの属性をインデックス化すると、Directory Server は無効なエントリーセットを返して直接的かつ内部検索を行うことができます。

13.2. 標準インデックスの作成

本セクションでは、Directory Server コンソールおよびコマンドラインを使用して、特定の属性について存在、等価、概算、部分文字列、および国際インデックスを作成する方法を説明します。

注記

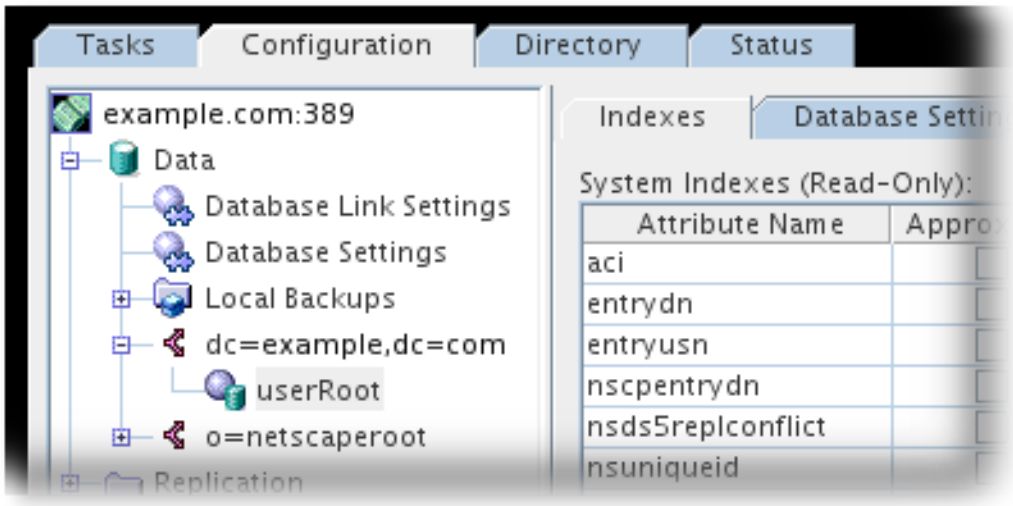
新しいインデックスタイプを作成すると、Directory Server はこのデフォルトインデックスを、今後作成する新規データベースのテンプレートとして使用します。デフォルトのインデックスを更新すると、更新された設定は既存のデータベースに適用されません。新しいインデックスを既存のデータベースに適用するには、「既存のデータベースへの新規インデックスの生成」 の説明に従って、db2index.pl スクリプトまたは cn=index,cn=tasks タスクを解除します。

- 「サーバーコンソールからのインデックスの作成」
- 「コマンドラインからのインデックスの作成」

13.2.1. サーバーコンソールからのインデックスの作成

存在、等価性、概算、部分文字列、または国際インデックスを作成するには、以下を実行します。

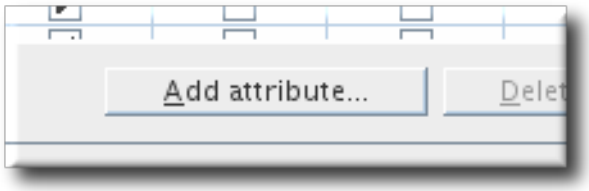
1. Configuration タブを選択します。
2. Data ノードを展開し、インデックスを作成するデータベースの接尾辞を展開して、データベースを選択します。
3. 右側のペインで Indexes タブを選択します。



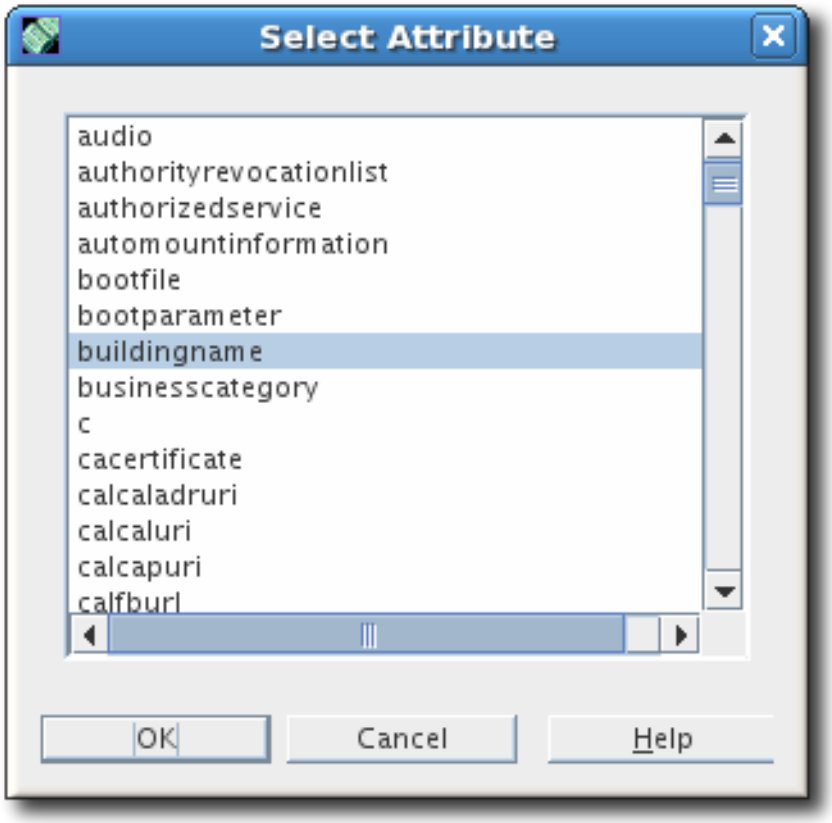
注記

Database Settings ノードをクリックしないでください。これにより Default Index Settings ウィンドウが開き、データベースごとにインデックスを設定するウィンドウは開かれません。

4. インデックス化する属性が Additional Indexes テーブルに一覧表示される場合は、ステップ 6 に進みます。それ以外の場合は、Add Attribute をクリックして、サーバースキーマで利用可能な属性の一覧を含むダイアログボックスを開きます。

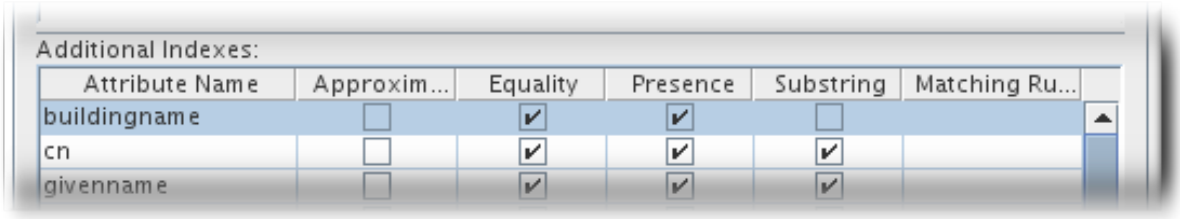


5. インデックスの属性を選択し、OK をクリックします。



サーバーは属性を Additional Indexes テーブルに追加します。

6. 各属性について保持するインデックスの各タイプの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には、Matching Rules フィールドで使用する照合順序 の OID を入力します。

複数の言語を使用して属性をインデックス化するには、複数の OID をコンマで区切って指定しますが、空白文字は一覧表示しません。言語のリスト、関連する OID、および照合順序の詳細は、付録D *国際化* を参照してください。

8. **Save** をクリックします。

新しいインデックスは、追加する新しいデータと、そのディレクトリーに既存のデータに対してすぐに有効になります。サーバーを再起動する必要はありません。

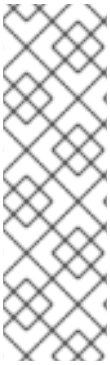
13.2.2. コマンドラインからのインデックスの作成

注記

システムインデックスは Directory Server にハードコーディングされるため、新しいシステムインデックスを作成できません。

ldapmodify を使用して、新しいインデックス属性をディレクトリーに追加します。

- デフォルトインデックスのいずれかになる新しいインデックスを作成するには、新しいインデックス属性を **cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config** エントリーに追加します。
- 特定のデータベースに新しいインデックスを作成するには、作成するインデックスを **cn=index,cn=**database_name**,cn=ldbm database,cn=plugins,cn=config** エントリーに追加します。ここで、**cn=**database_name はデータベースの名前に対応します。



注記

dse.ldif ファイルの **cn=config** の下にエントリーを作成しないでください。シンプルな flat **dse.ldif** 設定ファイルの **cn=config** エントリーは、通常のエントリーと同じ拡張性の高いデータベースには保存されません。その結果、多くのエントリー (特に頻繁に更新される可能性のあるエントリー) を **cn=config** に保存すると、パフォーマンスが低下する可能性があります。パフォーマンスの理由から、**cn=config** に単純なユーザーエントリーを保存することは推奨していませんが、設定情報が一元化されるため、**cn=config** に Directory Manager エントリーやレプリケーションマネージャー (サプライヤーのバインド DN) エントリーなどの特別なユーザーエントリーを保存すると便利です。

エントリーの追加に必要な LDIF 更新ステートメントの詳細は、「ディレクトリーエントリーの更新」を参照してください。

たとえば、**Example1** データベースに ***sn*** (姓 (surname)) 属性の有無、等価、および部分文字列インデックスを作成するには、以下を実行します。

1. **ldapmodify** を実行して、新しいインデックスの LDIF エントリーを追加します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=sn,cn=index,cn=Example1,cn=ldbm database,cn=plugins,cn=config
changetype: add
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、インデックスの属性の名前 (この例では *sn* 属性) が含まれます。エントリーは nsIndex オブジェクトクラスのメンバーです。nsSystemIndex 属性は false で、インデックスが Directory Server 操作に不可欠ではないことを示します。複数値の nsIndexType 属性は、存在 (pres)、等価 (eq)、および部分文字列 (sub) のインデックスを指定します。各キーワードは別々の行で入力する必要があります。この例の nsMatchingRule 属性は、ブルガリア語の照合順序の OID を指定しています。マッチングルールは、言語や、日付や整数などの他のフォーマットなど、値の一致の可能性を示すことができます。

nsIndexType 属性の none キーワードを使用して、インデックスが属性に対して維持されないように指定できます。この例では、nsIndexType を none に変更して、Example1 データベースの *sn* インデックスを一時的に無効にします。

```
dn: cn=sn,cn=index,cn=Example1,cn=ldbm database,cn=plugins,cn=config
objectClass:top
objectClass:nsIndex
cn:sn
nsSystemIndex:false
nsIndexType:none
```

マッチングルールとその OID の完全リストについては、「一致するルールの使用」を参照してください。インデックス設定属性については、『Red Hat Directory Server Configuration, Command, and File Reference』を参照してください。

注記

インデックスの作成時に、(属性のエイリアスではなく) 属性のプライマリー名を常に使用します。属性の主な名前は、スキーマの属性に対して最初に一覧表示される名前です。たとえば、ユーザー ID 属性の uid です。

13.3. 既存のデータベースへの新規インデックスの生成

新しいインデックスは、既存のデータベースに自動的に追加されません。これらは手動で追加し、Directory Server には、db 2index.pl スクリプトを実行するか、cn =index,cn=tasks タスクを実行するという 2 つの方法があります。

重要

インデックスを再生成する前に、検索は失敗します。

13.3.1. db2index.pl スクリプトの実行

インデックスエントリーの作成、または既存のインデックスエントリーにインデックスタイプを追加したら、db 2index.pl スクリプトを実行して、Directory Server が保持する新しいインデックスセットを生成します。スクリプトが実行されると、ディレクトリーに追加された新しいデータと、ディレクトリー内の既存のデータに対して、新しいインデックスのセットがアクティブになります。

db2index.pl Perl スクリプトを実行します。

```
# db2index.pl -Z instance_name -D "cn=Directory Manager" -w secret -n ExampleServer -t sn
```

この Perl スクリプトの使用方法は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス を参照してください』。

この例で使用されるパラメーターの詳細は、Red『Hat Directory Server の設定、コマンド、およびファイルリファレンス の db2index ユーティリティーの説明を参照してください』。

13.3.2. cn=tasks エントリーを使用したインデックスの作成

Directory Server 設定の cn=tasks,cn=config エントリーは、サーバーがタスクの管理に使用する一時的なエントリーのコンテナーエントリーです。複数の共通ディレクトリータスクには、cn=tasks,cn=config の下にコンテナーエントリーがあります。一時タスクエントリーは、cn=index,cn=tasks,cn=config の下に作成し、インデックス操作を開始できます。

このタスクエントリーには、一意の名前(*cn*)と、属性およびインデックスの定義が必要です。形式は attribute:index_type で *nsIndexAttribute* に設定します。

以下に例を示します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=example presence index,cn=index,cn=tasks,cn=config
changetype: add
objectclass: top
objectclass: extensibleObject
cn: example presence index
nsInstance: userRoot
nsIndexAttribute: "cn:pres"
```

index_typeは 3 つあります。

- 存在 インデックスの事前
- 等価インデックスの eq
- 部分文字列インデックスのサブ

タスクが完了するとすぐに、エントリーはディレクトリー設定から削除されます。

この例で使用される属性と、このエントリーに設定できるその他の属性の詳細は、Red『Hat Directory Server 設定、コマンド、およびファイルリファレンス の cn=*task_name*,cn=index,cn=tasks,cn=config エントリーの説明を参照してください』。

13.4. ロージング(VLV)インデックスの作成

仮想リストビュー(VLV)インデックスは、サーバーパフォーマンスの強化中に検索を迅速化するために切り捨てられるリストを作成する方法です。VLV インデックス自体はリソースを維持するのに役に立ちますが、大規模なディレクトリーでは（1000 エントリーより）有益です。

参照インデックスは、アルファベット順に一覧表示されているエントリーを整理する VLV インデックス一種で、エントリーの検索が容易になります。

VLV インデックスは、標準的なインデックスのように属性に適用されるのではなく、エントリーに設定された属性と、ディレクトリーツリー内のエントリーの位置に基づいて動的に生成されます。VLV インデックスは、標準のインデックスとは異なり、データベースの設定ではなく、データベース内の特別なエントリーになります。

注記

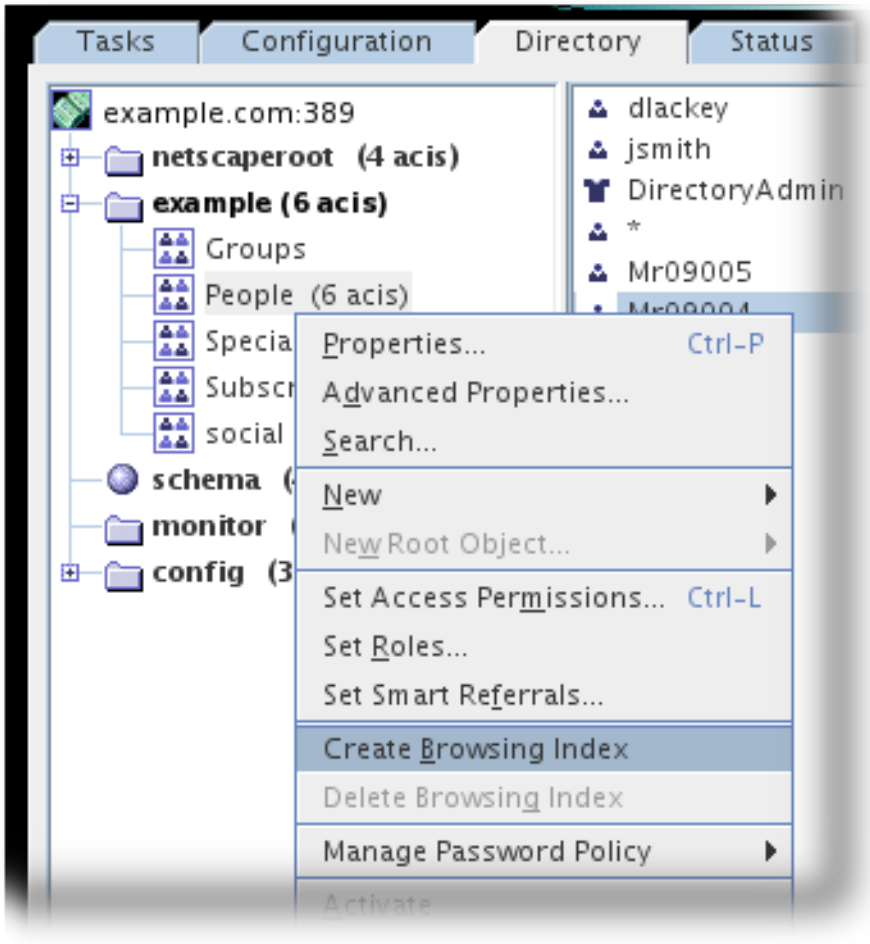
VLV インデックスは、一部の外部 LDAP クライアントで返される簡単なページ化された結果と似ています。単純なページ化された結果は検索ごとに計算されますが、VLV インデックスは永続的なリストであるため、VLV インデックスは検索を迅速化します。ただし、サーバーが維持するオーバーヘッドの一部が必要になります。

シンプルなページの結果と VLV インデックスは、同じ検索では使用 できません。

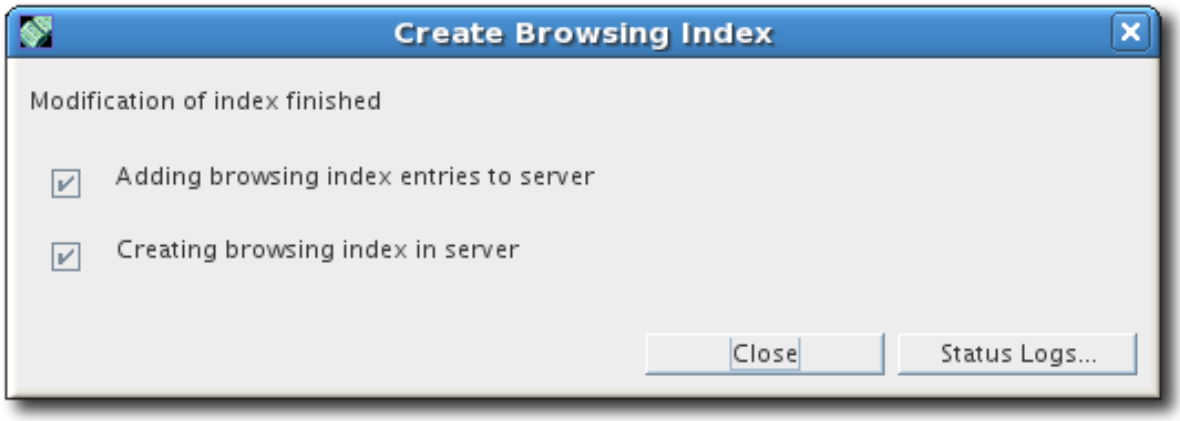
詳細は、「単純なページ結果の使用」を参照してください。

13.4.1. サーバーコンソールから参照インデックスの作成

1. **Directory** タブを選択します。
2. 左側のナビゲーションツリーで、インデックスを作成する **People** などのエントリーを選択します。
3. **Object** メニューから **Create Browsing Index** を選択します。



Create Browsing Index ダイアログボックスが表示され、インデックス作成のステータスを表示します。**Status Logs** ボックスをクリックし、作成されたインデックスのステータスを表示します。



4. **Close** をクリックします。

ディレクトリーに追加される新しいデータに対して、新しいインデックスがすぐにアクティブになります。サーバーを再起動する必要はありません。

VLV 検索情報、または VLV 検索用にデフォルトで設定されるアクセス制御ルールを変更する方法は、「参照インデックスエントリーの追加」 および「VLV 情報のアクセス制御の設定」 を参照してください。

13.4.2. コマンドラインから参照インデックスの作成

コマンドラインから参照インデックスまたは仮想リストビュー (VLV) インデックスを作成するには、以下の手順を行います。

1. **ldapmodify** を使用して新しい参照インデックスエントリーを追加するか、既存の参照インデックスエントリーを編集します。「参照インデックスエントリーの追加」を参照してください。
2. **vlvindex** スクリプトを実行して、サーバーが維持する参照インデックスの新しいセットを生成します。「vlvindex スクリプトの実行」 を参照してください。または、**cn=tasks,cn=config** (「cn=tasks エントリーを使用した参照インデックスの作成」)で適切なタスクを起動します。
3. VLV インデックス情報へのアクセス制御が適切に設定されていることを確認します。「VLV 情報のアクセス制御の設定」を参照してください。

13.4.2.1. 参照インデックスエントリーの追加

作成する参照インデックスエントリーの型は、加速化する **ldapsearch** 属性のソートタイプによって異なります。以下を考慮することが重要です。

- 検索の範囲 (base、one、sub)
- 検索のベース (検索の開始点として使用するエントリー)
- ソートする属性
- 検索のフィルター

  検索用のフィルターを指定する方法は、「*14章 ディレクトリーエントリーの検索*」を参照してください。
- 検索のベースとなるエントリーが属する LDBM データベース。LDBM データベースで参照先インデックスのみを作成できます。

たとえば、以下の属性で **Example1** データベースに保持されるエントリー **ou=People,dc=example,dc=com** で **ldapsearch** を加速する参照インデックスを作成します。

- **検索ベースは ou=People,dc=example,dc=com です。**
- **検索フィルターは （|(objectclass=*)(objectclass=ldapsubentry)）です。**
- **スコープは 1 です。**
- **返される属性のソート順序は、*cn*、*givenname*、*o*、*ou*、*sn* です。**

1. **ldapmodify を実行し、参照しているインデックスのベース、スコープ、およびフィルターを指定するエントリーを追加します。**

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=MCC ou=People dc=example dc=com,cn=userRoot,cn=ldbm
database,cn=plugins,cn=config
changetype: add
objectClass: top
objectClass: vlvSearch
cn: MCC ou=People dc=example dc=com
vlvBase: ou=People,dc=example,dc=com
vlvScope: 1
vlvFilter: (|(objectclass=*)(objectclass=ldapsubentry))
```

   - ***cn* には、参照インデックスを作成するエントリーを指定する参照インデックス識別子 (この例では ou=People,dc=example,dc=com エントリー) が含まれます。Red Hat は、Directory Server コンソールによって採用されるアプローチである参照インデックス識別子にエントリーの dn を使用して、同一の参照インデックスが作成されないようにすることを推奨します。エントリーは vlvSearch オブジェクトクラスのメンバーです。**
   - **vlvbase 属性の値は、参照インデックスを作成するエントリーを指定します。この例では、ou=People,dc=example,dc=com エントリー (参照インデックス識別子) を指定します。**
   - **vlvScope 属性は 1 で、加速する検索の範囲が 1 であることを示します。1 の検索範囲は、*cn* 属性に指定されたエントリーの即時の子のみで、エントリー自体ではなく検索が実行されます。**
   - **vlvFilter は、検索に使用するフィルターを指定します (例:(|(objectclass=*) (objectclass=ldapsubentry)))。**

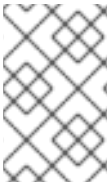
2. **2 番目のエントリーを追加して、返された属性のソート順序を指定します。**

```
dn: cn=by MCC ou=People dc=example dc=com,cn=MCC ou=People
  dc=example dc=com,cn=userRoot,cn=ldbm database,cn=plugins,
  cn= config
objectClass: top
objectClass: vlvIndex
cn: by MCC ou=People dc=example dc=com
vlvSort: cn givenName o ou sn
```

   - ***cn* には、参照先のインデックスソート識別子が含まれます。上記の *cn* は、デフォルトでコンソールによって作成されたタイプです。これには、参照インデックス ベースで設定されるソート順序が含まれます。エントリーは vlvIndex オブジェクトクラスのメンバーで**

す。

- *vlvSort* 属性の値は、属性のソート順序を指定します。この例では、*cn*、*givenName*、*o*、*ou*、さらに *sn* になります。



注記

最初に参照するインデックスエントリーを cn=database_name,cn=ldbm database,cn=plugins,cn=config ディレクトリーツリーノードに追加し、2 番目のエントリーは最初のエントリーの子である必要があります。

13.4.2.2. vlvindex スクリプトの実行

2 つの参照インデックスエントリーを作成したり、既存のインデックス参照エントリーに属性タイプを追加したりした後、vlv index スクリプトを実行して、Directory Server が保持する新しい参照インデックスのセットを生成します。スクリプトの実行後に、ディレクトリーに追加された新しいデータと、ディレクトリー内の既存のデータに対して、新しい参照インデックスのセットがアクティブになります。

vlvindex スクリプトを実行するには、以下を実行します。

1. サーバーを停止します。

```
# systemctl stop dirsrv@instance_name
```

2. vlvindex スクリプトを実行します。

```
# vlvindex -Z instance_name -n Example1 -T "by MCC ou=people dc=example dc=com"
```

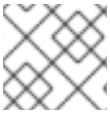
この例で使用されるパラメーターの詳細は、Red『Hat Directory Server の設定、コマンド、およびファイルリファレンス の vlvindex スクリプトの説明を参照してください』。

3. サービスを起動します。

```
# systemctl start dirsrv instance
```

13.4.2.3. cn=tasks エントリーを使用した参照インデックスの作成

vlvindex スクリプトを実行する代わりに、インデックスタスクを直接開始することもできます。



注記

インデックスタスクの実行は、vlv index スクリプトの実行 と同じです。

Directory Server 設定の cn=tasks,cn=config エントリーは、サーバーがタスクの管理に使用する一時的なエントリーのコンテナーエントリーです。複数の共通ディレクトリータスクには、cn=tasks,cn=config の下にコンテナーエントリーがあります。一時タスクエントリーは、cn=index,cn=tasks,cn=config の下に作成し、インデックス操作を開始できます。

このタスクエントリーには、一意の名前(*cn*)と他の属性 *nsIndexVLVAttribute* が必要です。これにより、VLV インデックスの生成に使用する参照インデックス定義エントリーの名前を指定します。

以下に例を示します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=example VLV index,cn=index,cn=tasks,cn=config
changetype: add
objectclass: extensibleObject
cn: example VLV index
nsIndexVLVAttribute: "by MCC ou=people,dc=example,dc=com"
```

タスクが完了するとすぐに、エントリーはディレクトリー設定から削除されます。

『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス』 には、**cn=tasks** エントリーで Directory Server タスクの実行に関する詳細情報が記載されています。

13.4.3. VLV 情報のアクセス制御の設定

デフォルトのアクセス制御命令 (ACI) は、認証されたユーザーのみが VLV インデックス情報を使用できます。認証されていないユーザーが VLV インデックス情報を使用するのを許可する必要がある場合は、***aci*** 属性を更新して ***userdn*** パラメーターを **ldap://anyone** に設定します。

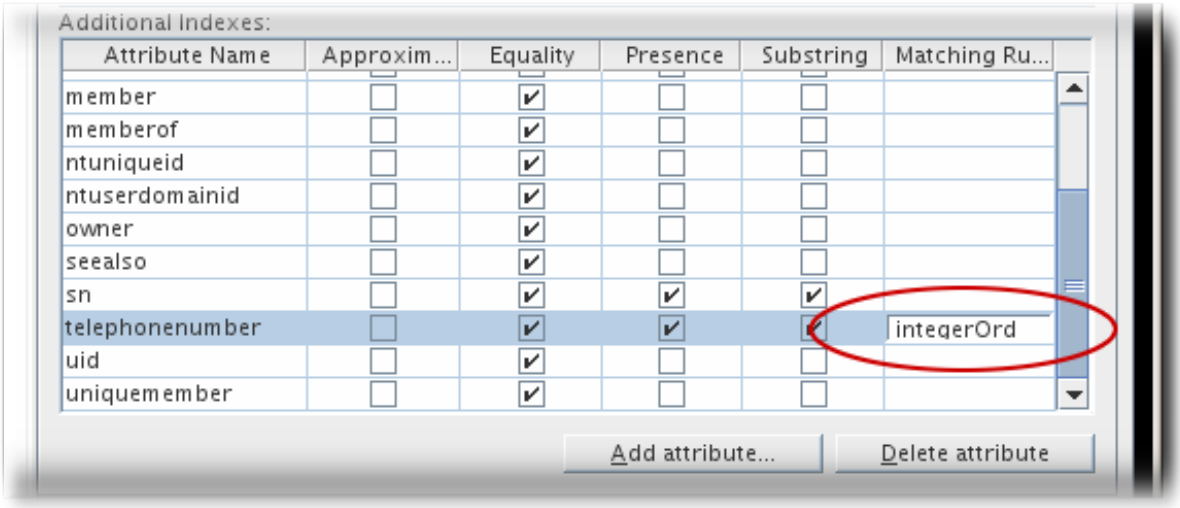
```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: oid=2.16.840.1.113730.3.4.9,cn=features,cn=config
changetype: modify
replace: aci
aci: (targetattr != "aci")(version 3.0; acl "VLV Request Control";
    allow( read, search, compare, proxy ) userdn = "ldap://anyone" ;)
```

13.5. インデックスのソート順序の変更

デフォルトでは、インデックスは ASCII 降順でアルファベット順にソートられます。これは、整数または電話番号などの数値属性値がある属性であっても、すべての属性に対して当てはまります。属性のマッチングルールセットを変更することで、sort メソッドを変更できます。

13.5.1. コンソールでのソート順序の変更

1. **Configuration** タブを選択します。
2. **Data** 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. 右側のペインで **Indexes** タブを選択します。
4. インデックスを選択し、Matching **Rules** フィールドで、使用する新しいソート順序を入力します。たとえば、アルファベットでなく、数字でソートするには、**integerOrderingMatch** を入力します。



5. **Save をクリックします。**

13.5.2. コマンドラインでのソート順序の変更

コマンドラインを使用してソート順序を変更するには、属性インデックスの *nsMatchingRule* を変更します。以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -x

dn: cn=sn,cn=index,cn=Example1,cn=ldbm database,cn=plugins,cn=config
changetype:modify
replace:nsMatchingRule
nsMatchingRule:integerOrderingMatch
```

13.6. INDEXED SUBSTRING SEARCH の WIDTH の変更

デフォルトでは、検索がインデックス化されるようにするには、検索文字列はワイルドカード文字をカウントせずに 3 文字以上である必要があります。たとえば、abc という文字列はインデックス検索になりますが、ab* はインデックス検索になりません。インデックス化された検索は、インデックスなし検索よりもはるかに高速であるため、検索キーの最小長を変更すると、インデックス化された検索の数を増やすと便利です。

検索パフォーマンスを改善するために、特に多くのワイルドカード検索を持つサイトの場合は、インデックス化された検索の検索文字列の長さを変更できます。Directory Server には、インデックス化された検索に必要な最小文字数を変更できる属性が 3 つあります。

- ***nsSubStrBegin* 属性は、ワイルドカードの前に検索文字列の最初にインデックス化された検索に必要な文字数を設定します。**

  ```
  abc*
  ```

- ***nsSubStrMiddle* 属性は、検索文字列の途中でワイルドカードが使用される、インデックス化された検索に必要な文字数を設定します。以下に例を示します。**

  ```
  ab*z
  ```

- ***nsSubStrEnd* 属性は、ワイルドカードの後に検索文字列の最後にインデックス化された検索に必要な文字数を設定します。以下に例を示します。**

```
*xyz
```

文字列トリプレット (before、middle、end) のデフォルトの部分文字列検索の長さは3、3、および 3 であり、すべての検索でワイルドカードの位置に最低 3 文字を必要とします。

別の文字列の長さを持つ属性インデックスは、**extensibleObject** オブジェクトクラスをエントリーに追加してから、部分文字列の検索の長さを設定します。

1. 特定の属性インデックスの新しいキーの長さを設定します。これには、**extensibleObject** オブジェクトクラスを追加してから、***nsSubStrBegin*** 属性、***nsSubStrEnd*** 属性、または ***nsSubStrMiddle*** 属性を適宜追加する必要があります。以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: attribute_name,cn=index,cn=database_name,cn=ldbm database,cn=plugins,cn=config
changetype: modify
add: objectclass
objectclass: extensibleObject
-
add: nsSubStrBegin
nsSubStrBegin: 2
-
add: nsSubStrMiddle
nsSubStrMiddle: 2
-
add: nsSubStrEnd
nsSubStrEnd: 2
```

2. サーバーを停止します。

```
# systemctl stop dirsrv.target
```

3. 属性インデックスを再作成します。部分文字列検索の幅オプションのいずれかを変更した場合は、インデックス全体を再作成する必要があります。

```
# db2index -t attribute_name
```

4. サーバーを再び起動します。

```
# systemctl start dirsrv.target
```

13.7. インデックスの削除

本セクションでは、インデックスから属性とインデックスタイプを削除する方法を説明します。

13.7.1. デフォルトインデックスエントリーからの属性の削除

Directory Server のデフォルト設定を使用する場合は、***sn*** などのデフォルトのインデックスエントリーに一覧表示される複数の属性がインデックス化されます。以下の属性はデフォルトインデックスの一部です。

表13.1 デフォルトのインデックス属性

aci	cn	entryusn
givenName	mail	mailAlternateAddress
mailHost	member	memberOf
nsUniqueId	ntUniqueId	ntUserDomainId
numsubordinates	objectclass	owner
parentid	seeAlso	sn
telephoneNumber	uid	uniquemember

警告

システムインデックスを削除すると、Directory Server のパフォーマンスが大幅に影響を受ける可能性があります。

たとえば、デフォルトのインデックスから *sn* 属性を削除するには、次のコマンドを実行します。

1. ***cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config* エントリーから属性を削除します。**

```
# ldapdelete -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
cn=sn,cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config
```

このエントリーから属性を削除しない場合は、サーバーの再起動後に *sn* 属性のインデックスが自動的に再作成され、破損します。

2. ***cn=attribute_name,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config* エントリーを削除します。詳細は、次を参照してください。**

   - **「サーバーコンソールを使用したインデックスからの属性の削除」**
   - **「コマンドラインを使用したインデックスから属性の削除」**

3. **db2index.pl Perl スクリプトを実行してインデックスを再作成します。**

```
# db2index.pl -Z instance_name -D "cn=Directory Manager" -w secret -n database_name
```

db2index.pl Perl スクリプトの使用に関する詳細は、db2index.pl(8) の man ページを参照してください。

13.7.2. サーバーコンソールを使用したインデックスからの属性の削除

Directory Server コンソールは、カスタムインデックス、メッセージングや Web サーバーなどの他のサーバーアプリケーションによって使用されるインデックス、およびデフォルトのインデックスを削除できます。

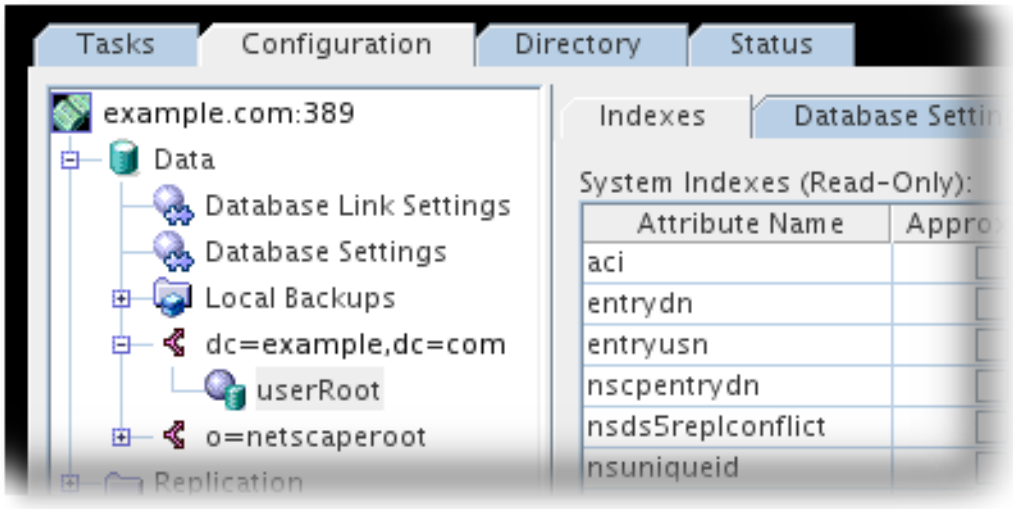
インデックスから属性を削除するには、以下を実行します。

1. 削除する属性が *cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config* デフォルトインデックスエントリーに一覧表示されている場合は、最初にこのエントリーから削除します。詳細は「デフォルトインデックスエントリーからの属性の削除」を参照してください。

2. **Configuration** タブを選択します。

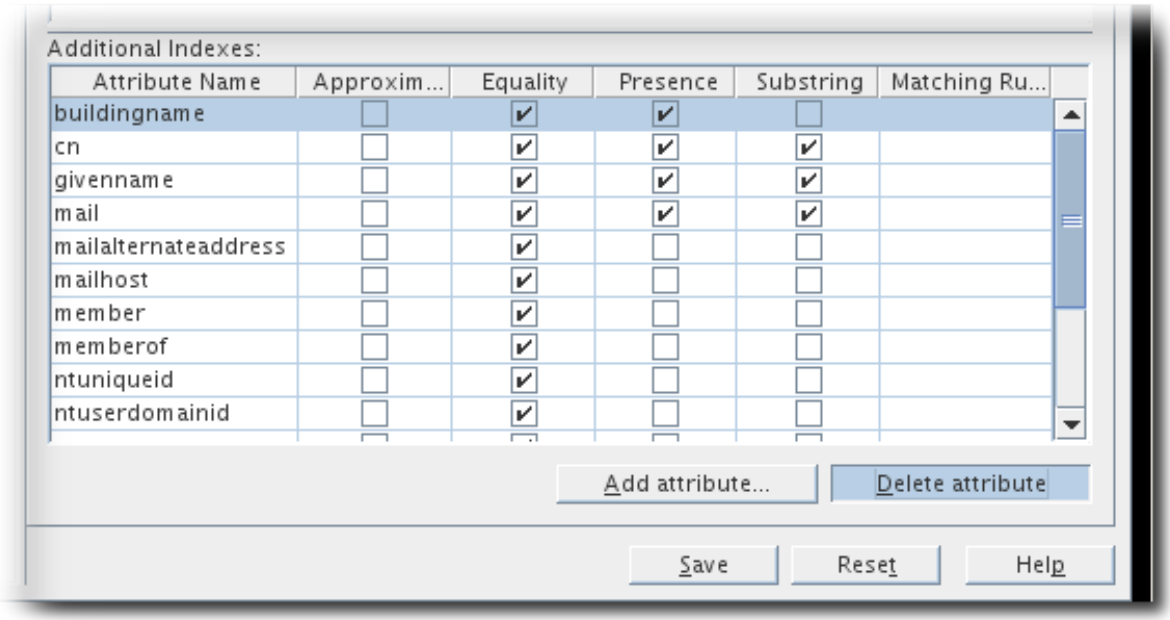
3. **Data** ノードを展開し、インデックスが含まれるデータベースに関連する接尾辞を展開します。

4. インデックスを削除するデータベースを選択します。



5. 削除するインデックスが含まれる属性を見つけます。インデックスの下のチェックボックスの選択を解除します。

   特定の属性に対して保持されるすべてのインデックスを削除するには、**Attribute Name** で属性のセルを選択し、**Delete Attribute** をクリックします。



6. **Save** をクリックします。

   Delete Index 警告ダイアログボックスが開き、インデックスの削除の確認が必要になります。

7. **Yes** をクリックしてインデックスを削除します。

13.7.3. コマンドラインを使用したインデックスから属性の削除

特定の状況では、インデックスから属性を削除します。たとえば、*sn* 属性を削除するには、以下のコマンドを実行します。

1. **削除する属性が *cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config* デフォルトインデックスエントリーに一覧表示されている場合は、最初にこのエントリーからこれを削除する必要があります。詳細は「デフォルトインデックスエントリーからの属性の削除」を参照してください。**

2. **インデックスから属性を削除します。**

   ```
   # ldapdelete -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
   cn=sn,cn=index,cn=database_name,cn=ldbm database,cn=plugins,cn=config
   ```

   エントリーを削除すると、*sn* 属性のインデックスは維持されなくなります。

3. **db2index.pl Perl スクリプトを実行してインデックスを再作成します。**

   ```
   # db2index.pl -Z instance_name -D "cn=Directory Manager" -w secret -n database_name
   ```

   db2index.pl Perl スクリプトの使用に関する詳細は、db2index.pl(8) の man ページを参照してください。

13.7.4. コマンドラインでのインデックスタイプの削除

たとえば、インデックスから *sn* 属性の sub インデックスタイプを削除するには、以下を実行します。

1. **インデックスタイプを削除します。**

   ```
   # ldapmodify -D "cn=Directory Manager" -W -x
   dn: cn=sn,cn=index,cn=database_name,cn=ldbm database,cn=plugins,cn=config

   changetype: modify
   delete: nsIndexType
   nsIndexType:sub
   ```

   インデックスエントリーを削除すると、*sn* 属性の部分文字列インデックスは維持されなくなります。

2. **db2index.pl Perl スクリプトを実行してインデックスを再作成します。以下に例を示します。**

   ```
   # db2index.pl -Z instance_name -D "cn=Directory Manager" -w secret -n database_name
   ```

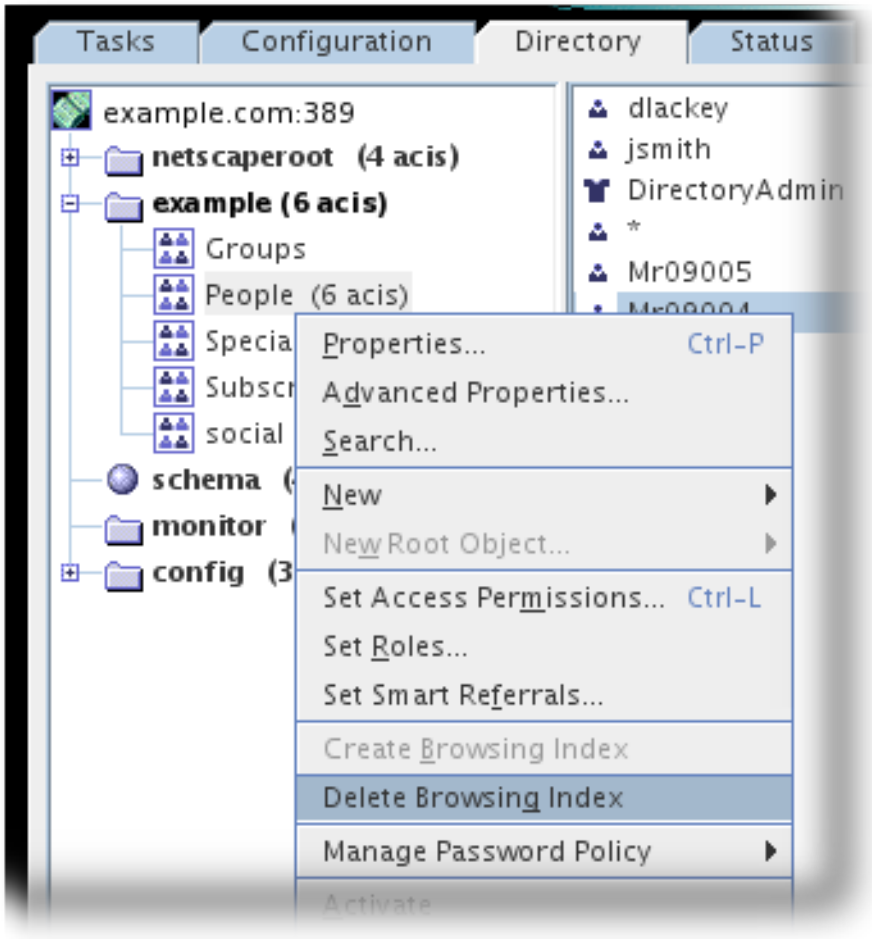
   db2index.pl Perl スクリプトの使用に関する詳細は、db2index.pl(8) の man ページを参照してください。

13.7.5. サーバーコンソールからの参照インデックスの削除

1. **Directory** 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を削除するエントリーを選択し、**Object** メニューから **Delete Browsing Index** を選択します。

   または、インデックスのエントリーを選択して、ナビゲーションツリーで削除する項目を選択し、ポップアップメニューから **Delete Browsing Index** を選択します。



3. **Delete Browsing Index** ダイアログボックスが表示され、インデックスの削除を確定するように求められます。**Yes** をクリックします。

4. **Delete Browsing Index** ダイアログボックスが表示され、インデックスの削除のステータスを表示します。

13.7.6. コマンドラインから参照インデックスの削除

コマンドラインから参照インデックスまたは仮想リストビュー(VLV)インデックスを削除するには、2つの手順を実行します。

1. **ldapdelete** を使用して、参照インデックスエントリーを削除するか、既存の参照インデックスエントリー(「参照インデックスエントリーの削除」)を編集します。

2. **vlvindex** スクリプトを実行して、サーバーが維持する参照インデックスの新しいセットを生成します(「vlvindex スクリプトの実行」)。または、**cn=tasks,cn=config** (「cn=tasks エントリーを使用した参照インデックスの作成」)で適切なタスクを起動します。

アルファベット順の参照インデックスと仮想リストビューの実際のエントリーは同じです。以下のセクションでは、参照インデックスを削除する手順を説明します。

13.7.6.1. 参照インデックスエントリーの削除

ldapdelete コマンドラインユーティリティーを使用して、インデックスエントリーの参照エントリーを削除するか、既存の参照インデックスエントリーを編集します。特定のデータベースの参照インデックスを削除するには、cn =index,cn= database_name,cn= ldbm database,cn=plugins,cn=config エントリーから参照しているインデックスエントリーを削除します。cn=database_name はデータベースの名前に対応します。

たとえば、ou=People,dc=example,dc=com エントリーで ldapsearch 操作の参照インデックスがあります。検索ベース が ou=People,dc=example,dc=com で、検索フィルターは （|(objectclass=*)(objectclass=ldapsubentry)） で、スコープは 1 で、返された属性のソート順序は *cn*、*givenname*、*o*、*ou*、*sn* です。

この参照インデックスを削除するには、対応する参照インデックスエントリーを 2 つ削除します。

```
dn: cn=MCC ou=People dc=example dc=com,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvSearch
cn: MCC ou=People dc=example dc=com
vlvBase: ou=People,dc=example,dc=com
vlvScope: 1 vlvFilter: (|(objectclass=*)(objectclass=ldapsubentry))

dn: cn=by MCC ou=People dc=example dc=com,cn=MCC ou=People
dc=example dc=com,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvIndex
cn: by MCC ou=People dc=example dc=com
vlvSort: cn givenname o ou sn
```

両方のエントリーを指定して ldapdelete を実行します。

```
# ldapdelete -D "cn=Directory Manager" -W -p 389 -h server.example.com -x "cn=MCC ou=People
dc=example dc=com,cn=userRoot,cn=ldbm database,cn=plugins,cn=config" "cn=by MCC ou=People
dc=example dc=com,cn=MCC ou=People dc=example dc=com,cn=userRoot,cn=ldbm
database,cn=plugins,cn=config"
```

2 つの参照インデックスエントリーを削除すると、参照インデックスは Example1 データベースで維持されなくなります。

13.7.6.2. vlvindex スクリプトの実行

既存の参照インデックスエントリーから参照インデックスエントリーまたは不要な属性タイプを削除したら、v lvindex スクリプトを実行して、Directory Server が保持する新しい参照インデックスを生成します。スクリプトが実行されると、ディレクトリーに追加された新しいデータと、ディレクトリー内の既存のデータに対して、新しい参照インデックスのセットがアクティブになります。

1. **サーバーを停止します。**

```
# systemctl stop dirsrv.target instance
```

2. **vlvindex スクリプトを実行します。**

```
# vlvindex -Z instance_name -n Example1 -T "by MCC ou=people dc=example dc=com"
```

この例で使用されるパラメーターの詳細は、Red『Hat Directory Server の設定、コマンド、およびファイルリファレンス の vlvindex スクリプトの説明を参照してください』。

3. サービスを再起動します。

```
# systemctl start dirsrv.target instance
```

または、cn =index,cn=tasks,cn=config の下に新しいタスクエントリーを作成して、インデックス操作を開始します。このタスクエントリーには、一意の名前(*cn*)と他の属性 *nsIndexVLVAttribute* が必要です。これにより、VLV インデックスの生成に使用する参照インデックス定義エントリーの名前を指定します。このタスクは vlvindex の実行と同じです。

以下に例を示します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=example VLV index,cn=index,cn=tasks,cn=config
changetype: add
objectclass: extensibleObject
cn: example VLV index
nsIndexVLVAttribute: "by MCC ou=people,dc=example,dc=com"
```

タスクが完了するとすぐに、エントリーはディレクトリー設定から削除されます。

この例で使用される属性と、このエントリーに設定できるその他の属性の詳細は、Red『Hat Directory Server 設定、コマンド、およびファイルリファレンス の cn=*task_name*,cn=index,cn=tasks,cn=config エントリーの説明を参照してください』。

第14章 ディレクトリーエントリーの検索

ディレクトリーのエントリーは検索でき、LDAP クライアントを使用して検索できます。ほとんどのクライアントは、ディレクトリーを簡単に検索でき、エントリー情報を簡単に取得できるように、何らかの形で検索インターフェースを提供しています。

14.1. リソース制限による検索パフォーマンスの改善

ディレクトリーが大きいと、データベース内のすべてのエントリーを検索すると、サーバーのパフォーマンスに影響を及ぼす可能性があります。効果的なインデックス化は、特定のシナリオでパフォーマンスを向上できます。ただし、大規模なデータベースでは、パフォーマンスを向上するのに十分な検索範囲が減っていない場合があります。

ユーザーおよびクライアントアカウントで妥当な制限を設定して、エントリーの合計数または個々の検索で費やした合計時間を節約できます。両方により、応答性が向上し、サーバー全体のパフォーマンスが向上します。

検索操作のサーバー制限は、ディレクトリーへのクライアントアプリケーションバインディングの特別な操作属性値を使用して制御されます。以下の検索操作制限を設定できます。

- ルックスルー制限。検索操作で確認できるエントリーの数を指定します。
- サイズ制限。検索操作にしたがって、サーバーがクライアントアプリケーションに返すエントリーの最大数を指定します。
- 時間制限。サーバーが検索操作の処理に費やす最大時間を指定します。
- アイドルタイムアウト。接続が切断される前に、サーバーへの接続がアイドル状態でいられる時間を指定します。
- 範囲のタイムアウト。範囲を使用した検索のために、別のルックスルー制限を指定します。

クライアントアプリケーションに設定されたリソース制限は、グローバルサーバー設定で設定されるデフォルトのリソース制限よりも優先されます。

注記

Directory Manager は、範囲検索を除き、デフォルトで無制限のリソースを受け取ります。

14.1.1. パフォーマンスおよびリソース制限の検索

詳細は、『Red『Hat Directory Server パフォーマンスチューニングガイド』の該当するセクションを参照してください』。

14.1.2. 粒度の細かい ID リストサイズ

詳細は、『Red『Hat Directory Server パフォーマンスチューニングガイド』の該当するセクションを参照してください』。

14.1.3. 単一ユーザーでのリソース制限の設定

1. **Directory** タブを選択します。

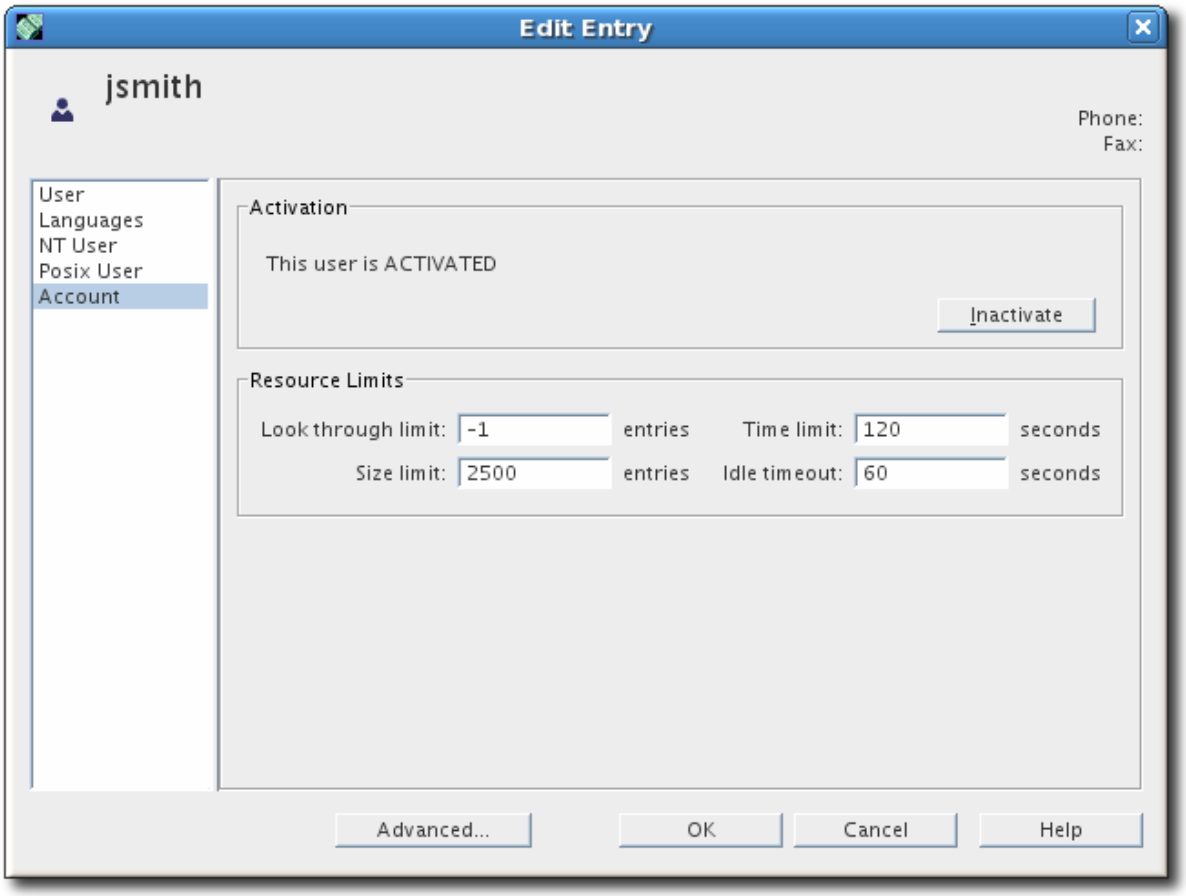
2. 左側のナビゲーションペインでナビゲーションツリーを参照し、リソース制限を設定するユーザーまたはロールをダブルクリックします。

   Edit Entry ダイアログボックスが表示されます。

3. 左側のペインで アカウント をクリックします。

4. リソース制限を設定します。設定可能な 4 つの制限があります。

   - ルックスルー制限。検索操作に対して、エントリーの最大数を調べます。
   - サイズ制限。検索操作に対応するために、サーバーがクライアントアプリケーションに返すエントリーの最大数。
   - 時間制限。サーバーが検索操作の処理に費やす最大時間。
   - アイドルタイムアウト。サーバーへの接続がアイドル状態でいられる期間。



   -1 を値として指定すると、制限がないことを意味します。

5. OK をクリックします。

14.1.4. コマンドラインを使用したユーザーおよびグローバルリソース制限の設定

Directory Server コンソールを使用する場合、コマンドラインでリソース制限を設定する場合に、さらにオプションを使用できます。Directory Server コンソールは、ユーザーレベルのリソース制限を設定します。コマンドラインで、管理者は、簡単なページや範囲検索など、ユーザーレベルのリソース制限、グローバルリソース制限、および特定の種類の検索を設定できます。「検索アルゴリズムの概要」は、これらのリソース制限が Directory Server 検索パフォーマンスにどのように影響するかについて詳しく説明しています。

「コマンドラインを使用したユーザーおよびグローバルリソース制限の設定」は、コマンドラインを使用して各エントリーに設定できる操作属性を一覧表示します。ldapmodify を使用して、エントリーに属性を追加します。

ユーザーレベルの属性は各エントリーに設定されますが、グローバル設定属性は適切なサーバー設定エリアに設定されます。

シークスルー制限

検索操作に対して検査するエントリーの数を指定します。この属性を -1 の値に指定すると、制限がないことを意味します。

- ユーザーレベルの属性: *nsLookThroughLimit*
- グローバル設定:
  - 属性: *nsslapd-lookthroughlimit*
  - エントリー: cn=config,cn=ldbm database,cn=plugins,cn=config

ページルックアップの制限

ルックスルー制限と同様に、検査するエントリーの数を指定しますが、単純なページング検索操作に特化しています。この属性を -1 の値に指定すると、制限がないことを意味します。

- ユーザーレベルの属性: *nsPagedLookThroughLimit*
- グローバル設定:
  - 属性： *nsSizeLimit*
  - エントリー: cn=config

サイズ制限

検索操作にしたがって、サーバーがクライアントアプリケーションに返すエントリーの最大数を指定します。この属性を -1 の値に指定すると、制限がないことを意味します。

- ユーザーレベルの属性: *nsSizeLimit*
- グローバル設定:
  - 属性: *nsslapd-sizelimit*
  - エントリー: cn=config

ページサイズ制限

サイズの制限と同様、サーバーがクライアントアプリケーションに戻る最大エントリー数を指定しますが、単純なページング検索操作の場合に限ります。この属性を -1 の値に指定すると、制限がないことを意味します。

- ユーザーレベルの属性: *nsPagedSizeLimit*
- グローバル設定:
  - 属性: *nsslapd-pagedsizelimit*
  - エントリー: cn=config

時間制限

サーバーが検索操作の処理に費やす最大時間を指定します。この属性を -1 の値に指定すると、制限がないことを意味します。

- ユーザーレベルの属性: *nsTimeLimit*
- グローバル設定:
  - 属性: *nsslapd-timelimit*
  - エントリー: cn=config

アイドルタイムアウト

接続が切断される前に、サーバーへの接続がアイドル状態でいられる時間を指定します。値は秒単位で指定されます。この属性を -1 の値に指定すると、制限がないことを意味します。

- ユーザーレベルの属性: *nsidletimeout*
- グローバル設定:
  - 属性: *nsslapd-idletimeout*
  - エントリー: cn=config

ID リストのスキャン制限

検索結果のインデックスファイルから読み込まれるエントリー ID の最大数を指定します。ID リストのサイズがこの値よりも大きい場合、検索はインデックスリストを使用せず、インデックスなしの検索として扱われ、データベース全体を検索します。

- ユーザーレベルの属性: *nsIDListScanLimit*
- グローバル設定:
  - 属性: *nsslapd-idlistscanlimit*
  - エントリー: cn=config,cn=ldbm database,cn=plugins,cn=config

ページ ID リストスキャンの制限

ID リストスキャンの制限と同様、検索結果のインデックスファイルから読み込まれるエントリー ID の最大数を指定しますが、ページングされた検索操作に対して指定します。

- ユーザーレベルの属性: *nsPagedIDListScanLimit*
- グローバル設定:
  - 属性: *nsslapd-pagedidlistscanlimit*
  - エントリー: cn=config,cn=ldbm database,cn=plugins,cn=config

範囲のルックアップの制限

範囲検索操作に関するエントリー数を指定します(greater-than、equal-to-or-greater-than、less-than、または equal-to-less-than 演算子を使用した検索)。この属性を -1 の値に指定すると、制限がないことを意味します。

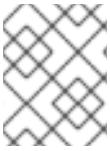
- ユーザーレベルの属性: 利用できません。
- グローバル設定:
  - 属性: *nsslapd-rangelookthroughlimit*
  - エントリー: cn=config,cn=ldbm database,cn=plugins,cn=config

上記のパラメーターの詳細は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス』 の説明を参照してください。

たとえば、これにより、ldapmodify を使用して her エントリーを変更することで、Barbara Jensen のサイズ制限を設定します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: uid=user_name,ou=People,dc=example,dc=com
changetype: modify
add: nsSizeLimit
nsSizeLimit: 500
```

ldapmodify ステートメントは、*nsSizeLimit* 属性を Babs Jensen のエントリーに追加し、検索結果のサイズ制限を 500 エントリーにします。



注記

ユーザーが設定を変更できないように、アクセス制御リスト (ACL) を設定します。ACL の詳細は、「18章 *アクセス制御の管理*」を参照してください。

14.1.5. 匿名バインドでのリソース制限の設定

リソース制限はユーザーエントリーに設定されます。匿名のバインディングは、当然ながら、ユーザーエントリーとは関係ありません。これは、通常グローバルリソース制限が匿名操作に適用されることを意味します。ただし、リソース制限のあるテンプレートユーザーエントリーを作成し、そのテンプレートを匿名バインドに適用することで、匿名バインド専用のリソース制限を設定することができます。

1. テンプレートエントリーを作成し、匿名バインドに適用するリソース制限を設定します。

注記

パフォーマンス上の理由から、テンプレートは、エントリーキャッシュは使用しない cn=config 接尾辞ではなく、通常のバックエンドになければなりません。

以下に例を示します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=anon template,ou=people,dc=example,dc=com
changetype: add
objectclass: nsContainer
objectclass: top
cn: anon template
```

```
nsSizeLimit: 250
nsLookThroughLimit: 1000
nsTimeLimit: 60
```

2. レプリケーショントポロジー内のすべてのマスターで、テンプレートエントリーの DN を参照するサーバー設定に *nsslapd-anonlimitsdn* を追加します。「コマンドラインを使用したユーザーおよびグローバルリソース制限の設定」に任意のリソース制限を設定できます。以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -x

dn: cn=config
changetype: modify
add: nsslapd-anonlimitsdn
nsslapd-anonlimitsdn: cn=anon template,ou=people,dc=example,dc=com
```

14.1.6. 範囲検索のパフォーマンス向上

範囲検索は演算子 (「検索フィルターでの演算子の使用」) を使用して括弧を設定して検索し、ディレクトリー内のエントリーのサブセット全体を返します。たとえば、これにより 1 月 1 日の午前 0 時以降に変更されたすべてのエントリーを検索します。

```
(modifyTimestamp>=20200101010101Z)
```

範囲検索の性質は、ディレクトリー内のすべてのエントリーを評価して、その範囲内にあるかどうかを確認する必要があることです。基本的に、範囲検索は常に ID 検索です。

ほとんどのユーザーの場合は、ルックスルーの制限が開始され、範囲の検索が全 ID 検索に変換するのを防ぎます。これにより、全体的なパフォーマンスが向上し、さまざまな検索結果を加速します。ただし、Directory Manager などの一部のクライアントまたは管理ユーザーには、ルックスルー制限が設定されていない場合があります。この場合は、範囲検索が完了するまで数分かかるか、無限に続行することがあります。

範囲に対するルックスルー制限を個別に設定することも可能です。これにより、クライアントや管理者ユーザーは、パフォーマンスが低下する可能性のある範囲検索に合理的な制限を設けながらも、高いルックスルー制限を設定することができます。

これは *nsslapd-rangelookthroughlimit* 属性で設定されます。デフォルト値は 5000 で、デフォルトの *nsslapd-lookthroughlimit* 属性値と同じです。

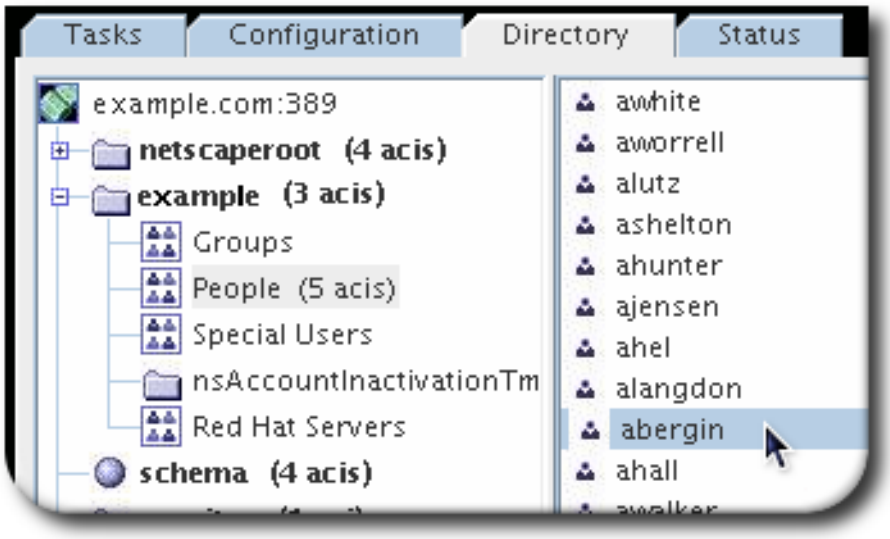
以下に例を示します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=config,cn=ldbm database,cn=plugins,cn=config
changetype: add
add: nsslapd-rangelookthroughlimit
nsslapd-rangelookthroughlimit: 7500
```

14.2. DIRECTORY SERVER コンソールを使用したエントリーの検索

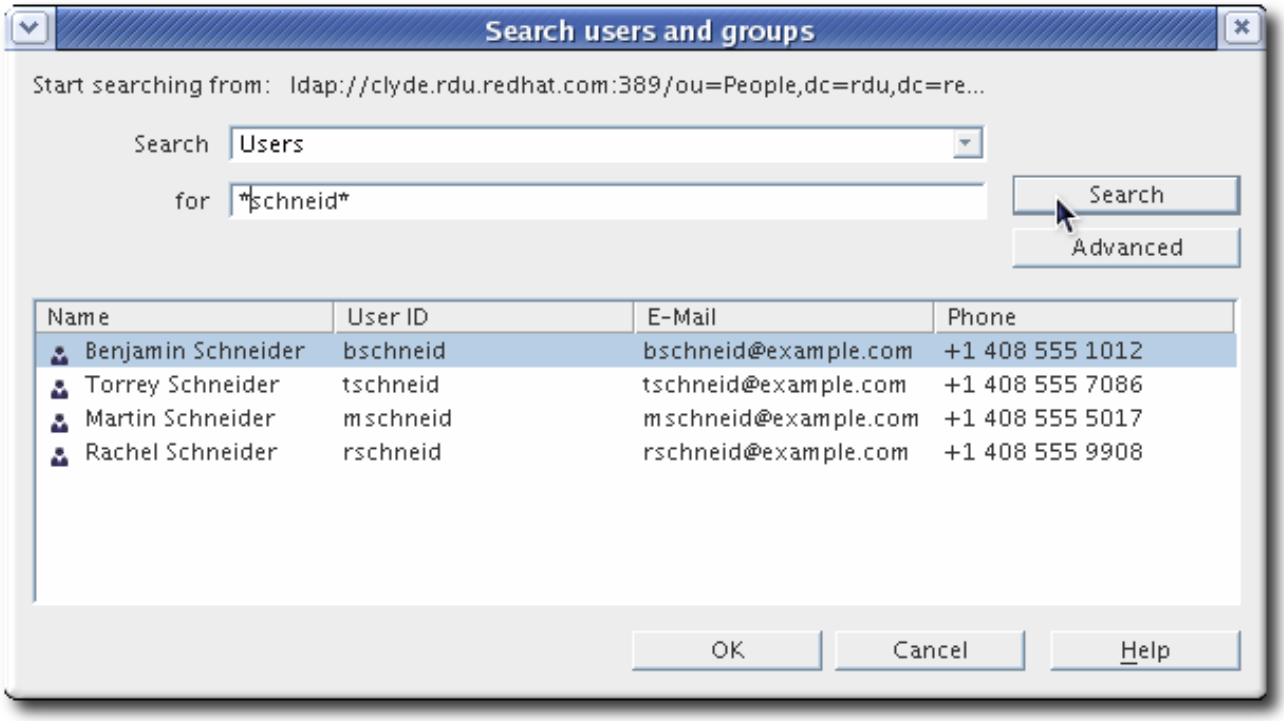
Directory Server コンソールの Directory タブを参照して、ディレクトリーツリーのコンテンツを表示し、ディレクトリー内の特定のエントリーを検索できます。

図14.1 ディレクトリータブでエントリーの閲覧



ディレクトリーへの認証に使用される DN に応じて、このタブには、ユーザーアカウントが表示できるディレクトリーの内容が表示されます。ツリーの内容を参照するか、エントリーを右クリックし、ポップアップメニューから **Search** を選択します。

図14.2 エントリーの検索

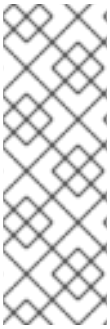


警告

Red Hat テクニカルサポートによる指示がない限り、Directory タブを使用して o=NetscapeRoot 接尾辞のコンテンツを変更しないでください。

14.3. LDAPSEARCH の使用

ldapsearch コマンドラインユーティリティーは、ディレクトリーエントリーの検索および取得が可能です。このユーティリティーは、指定した ID および認証情報を使用して指定のサーバーへの接続を開き、指定の検索フィルターに基づいてエントリーを見つけます。検索範囲には、単一のエントリー (-s base)、エントリーの即時サブエントリー (-s one)、またはツリー全体またはサブツリー (-s sub) を含めることができます。



注記

一般的な間違いは、識別名で使用される属性に基づいてディレクトリーを検索することを仮定することです。識別名はディレクトリーエントリーの一意の識別子であり、検索キーとして使用できません。代わりに、エントリー自体に保存されている属性とデータのペアに基づいてエントリーを検索します。したがって、エントリーの識別名が uid=bjensen,ou=People,dc=example,dc=com の場合、dc=example の検索は、そのエントリーの属性として dc:example が明示的に追加されない限り、そのエントリーと一致しません。

検索結果は LDIF 形式で返されます。LDIF は RFC 2849 に定義されており、「付録B *LDAP データ交換形式*」で詳細に説明されています。

このセクションには、以下のトピックに関する情報が含まれます。

- **「ldapsearch コマンドライン形式」**
- **「一般的に使用される ldapsearch オプション」**
- **「特殊文字の使用」**

14.3.1. ldapsearch コマンドライン形式

ldapsearch コマンドは以下の形式を使用する必要があります。

```
# ldapsearch [-x | -Y mechanism] [options] [search_filter] [list_of_attributes]
```

- **-x (簡単なバインドを使用するため) または -Y (SASL メカニズムを設定するため) は、接続の種類を設定するために使用されます。**
- **オプション は、一連のコマンドラインオプションです。これが存在する場合は、検索フィルターの前に指定する必要があります。**
- search_filter は、**「LDAP 検索フィルター」**で説明されている LDAP 検索フィルターです。**-f** オプションを使用して、検索フィルターがファイルで指定されている場合には、別の検索フィルターを指定しないでください。
- list_of_attributes は、スペースで区切られた属性の一覧です。属性の一覧を指定すると、検索結果で返される属性の数を減らすことができます。この属性のリストは、検索フィルターの後に表示されなければなりません。例は、**「属性のサブセットの表示」**を参照してください。属性の一覧が指定されていない場合、検索はディレクトリーで設定されているアクセスコントロールで許可されているすべての属性の値を返しますが、運用上の属性は例外となります。

  操作属性を検索結果として返すためには、検索コマンドの中で明示的に指定する必要があります。オブジェクトのすべての操作属性を返すには、**+** を指定します。明示的に指定した操作属性に加えて通常の属性を取得するには、**ldapsearch** コマンドの属性一覧でアスタリスク (*) を使用します。

一致する DN の一覧のみを取得するには、特別な属性 1.1 を使用します。以下に例を示します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com \
    -b "dc=example,dc=com" -x "(objectclass=inetorgperson)" 1.1
```

14.3.2. 一般的に使用される ldapsearch オプション

以下の表は、最も一般的に使用される **ldapsearch** コマンドラインオプションを示しています。指定した値に空白 ( )が含まれる場合、値は **-b "cn=My Special Group,ou=groups,dc=example,dc=com"** などの単一引用符または二重引用符で囲む必要があります。

重要

OpenLDAP の **ldapsearch** ユーティリティーは、デフォルトで SASL 接続を使用します。簡単なバインドを実行するか、TLS を使用するには、**-x** 引数を使用して SASL を無効にし、他の接続方法を許可します。

オプション	説明
-b	検索の開始点を指定します。ここで指定する値は、現在データベースに存在する識別名である必要があります。これは、**LDAP_BASEDN** 環境変数がベース DN に設定されている場合は任意です。このオプションで指定する値は、単一引用符または二重引用符で指定する必要があります。以下に例を示します。 **-b "cn=Barbara Jensen,ou=Product Development,dc=example,dc=com"** ルート DSE エントリーを検索するには、**-b ""** などの空の文字列を指定します。
-D	サーバーへの認証に使用する識別名を指定します。これは、サーバーによって匿名アクセスに対応している場合はオプションです。指定している場合、この値は Directory Server が認識する DN である必要があります。また、エントリーを検索する権限も必要です 例： **-D "uid=bjensen,dc=example,dc=com"**
-H	サーバーへの接続に使用する LDAP URL を指定します。従来の LDAP URL の場合は、以下の形式になります。 **ldap[s]://***hostname***[:***port***]** ポート は任意です。ポートを使用しないと、LDAP ポートの場合は 389、LDAPS ポートの場合は 636 がデフォルトで使用されます。 これは LDAPI URL を使用することもできます。各要素は、スラッシュ (/) ではなく、HTML の 16 進コード **%2F** で区切られています。 **ldapi://%2F***full***%2F***path***%2F***to***%2Fslapd-example.socket** LDAPI の場合は、サーバーがリッスンする LDAPI ソケットの完全パスおよびファイル名を指定します。この値は LDAP URL として解釈されるため、パスおよびファイル名のスラッシュ (/) をエスケープ処理して、URL エスケープ値 **%2F** としてエスケープする必要があります。 **-h** および **-p** の代わりに **-H** オプションが使用されます。

オプション	説明
-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 アドレスを指定します (例: **-h server.example.com**)。ホストが指定されない場合、**ldapsearch** は localhost を使用します。  **注記** Directory Server は、IPv4 アドレスと IPv6 アドレスの両方に対応します。
-l	検索要求が完了するまで待つ最大秒数を指定します (例: **-l 300**)。***nsslapd-timelimit*** 属性のデフォルト値は **3600** 秒です。指定された値に関係なく、**ldapsearch** はサーバーの ***nsslapd-timelimit*** 属性によって許可されるよりも長く待機します。
-p	Directory Server が使用する TCP ポート番号を指定します (例: **-p 1049**)。デフォルトは **389** です。 **-h** が指定されている場合は、デフォルト値が指定されていても **-p** も指定する必要があります。
-s *scope*	検索の範囲を指定します。範囲は以下のいずれかになります。 **base**: **-b** オプションで指定されたエントリー、または **LDAP_BASEDN** 環境変数により定義されたエントリーだけを検索します。 **one**: **-b** オプションで指定したエントリーの即時の子のみを検索します。子のみが検索されます。**-b** オプションに指定された実際のエントリーは検索されません。 **sub**: **-b** オプションで指定されたエントリーおよびその子すべてを検索します。つまり、**-b** オプションで識別された時点から、サブツリー検索を実行します。これがデフォルトになります。
-W	パスワードの入力を要求します。このオプションが設定されていない場合は、匿名アクセスが使用されます。 あるいは、**-w** オプションを使用して、パスワードをそのユーティリティーに渡します。他のユーザーのプロセス一覧にパスワードが表示され、シェルの履歴に保存されていることに注意してください。
-x	単純なバインドを許可するためにデフォルトの SASL 接続を無効にします。
-Y *SASL_mechanism*	認証に使用する SASL メカニズムを設定します。メカニズムが設定されていない場合、**ldapsearch** はサーバーがサポートする最適なメカニズムを選択します。 **-x** を使用しない場合は、**-Y** オプションを使用する必要があります。
-z *number*	検索リクエストへの応答で返すエントリーの最大数を設定します。この値は、ルート DN を使用してバインディングする際にサーバー側の ***nsslapd-sizelimit*** パラメーターを上書きします。wibrown>

14.3.3. 特殊文字の使用

ldapsearch コマンドラインユーティリティーを使用する場合は、スペース ( )、アスタリスク (*)、バックスラッシュ (\) などのコマンドラインインタープリターに対して特別な意味を持つ文字を含む値を指定する必要がある場合があります。引用符 ("") で特殊文字が含まれる値を囲みます。以下に例を示します。

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

コマンドラインインタープリターに応じて、一重引用符または二重引用符を使用します。通常、単一引用符 (') を使用して値を囲みます。シェル変数がある場合は、二重引用符 (") を使用して変数挿入を許可します。詳細は、オペレーティングシステムのドキュメントを参照してください。

14.4. LDAP 検索フィルター

検索フィルターでは、返されるエントリーを選択します。これらは ldapsearch コマンドラインユーティリティーで最もよく使用されます。ldapsearch を使用する場合は、ファイルに複数の検索フィルターがあり、各フィルターがファイルの別々の行にあるか、検索フィルターをコマンドラインに直接指定することができます。

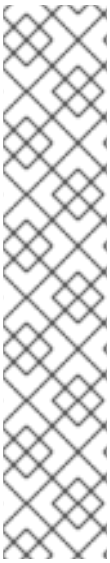
検索フィルターの基本的な構文は、以下のとおりです。

```
attribute operator value
```

以下に例を示します。

```
buildingname>=alpha
```

この例では、*buildingname* が属性、>= が演算子、および alpha が値となります。フィルターは、異なる属性をブール値演算子とともに使用するように定義することもできます。



注記

一致するルールフィルターを使用して部分文字列検索を実行する場合は、アスタリスク (*) 文字をワイルドカードとして使用し、ゼロ以上の文字を表します。

たとえば、文字 l で始まり文字 n で終わる属性値を検索するには、検索フィルターの値の部分に l*n を入力します。同様に、文字 u で始まるすべての属性値を検索するには、検索フィルターの値の部分に u* の値を入力します。

アスタリスク (*) 文字を含む値を検索するには、アスタリスクを指定のエスケープシーケンス \5c2a でエスケープする必要があります。たとえば、*businessCategory* 属性の値が Example*Net product line の社員をすべて検索するには、検索フィルターに以下の値を入力します。

```
Example\5c2a*Net product line
```

注記

一般的な間違いは、識別名で使用される属性に基づいてディレクトリーを検索することを仮定することです。識別名はディレクトリーエントリーの一意の識別子であり、検索キーとして使用できません。代わりに、エントリー自体に保存されている属性とデータのペアに基づいてエントリーを検索します。したがって、エントリーの識別名が **uid=bjensen,ou=People,dc=example,dc=com** の場合、**dc=example** の検索は、そのエントリーの属性として **dc:example** が明示的に追加されない限り、そのエントリーと一致しません。

14.4.1. 検索フィルターの属性の使用

検索の最も基本的なソートでは、エントリーに属性や特定の値があるかどうかを調べます。エントリーで属性を検索する 方法 は多数あります。属性が存在するかどうかを確認したり、完全値と一致するか、または部分的な値に対して一致を一覧表示したりすることができます。

存在 の検索では、値に関係なく、ワイルドカード (アスタリスク) を使用して、属性が設定されているすべてのエントリーを返します。たとえば、以下は、*manager* 属性を持つすべてのエントリーを返します。

```
"(manager=*)"
```

特定の値を持つ属性を検索することもできます。これは 等価 検索と呼ばれます。以下に例を示します。

```
"(cn=babs jensen)"
```

この検索フィルターは、Babs Jensen という共通名が含まれるすべてのエントリーを返します。多くの場合、等価検索では大文字と小文字が区別されません。

属性が言語タグに関連付けられた値を持つ場合は、すべての値が返されます。そのため、以下の 2 つの属性値は両方とも **"(cn=babs jensen)"** フィルターと一致します。

```
cn: babs jensen
cn;lang-fr: babs jensen
```

属性値 (部分文字列 インデックス) で部分的な一致を検索することもできます。以下に例を示します。

```
"(description=*X.500*)"
"(sn=*nderson)"
"(givenname=car*)"
```

部分文字列検索の長さは、「Indexed Substring Search の Width の変更」で説明されているように部分文字列のインデックス自体で設定されます。

14.4.2. 検索フィルターでの演算子の使用

検索フィルターの演算子は、属性と指定の検索値間の関係を設定します。人名検索では、演算子を使用して範囲を設定し、アルファベットのサブセット内の名字を返したり、ある数字以降の社員番号を返したりすることができます。

```
"(employeeNumber>=500)"
"(sn~=suret)"
"(salary<=150000)"
```

演算子は、音声検索や近似検索も可能で、不完全な情報でも効果的な検索ができ、特に国際化されたディレクトリーでは有効です。

検索フィルターで使用できる演算子が表14.1「検索フィルター演算子」に一覧表示されています。これらの検索フィルターに加えて、特別なフィルターを指定して、望ましい言語の照合順序で動作させることができます。国際文字セットを持つディレクトリーを検索する方法は、「国際化されたディレクトリーの検索」を参照してください。

表14.1 検索フィルター演算子

検索タイプ	演算子	説明
等号	=	指定された値と完全に一致する属性値が含まれるエントリーを返します （例： cn=Bob Johnson）
部分文字列	=string* string	指定の部分文字列が含まれる属性が含まれるエントリーを返します たとえば、cn=Bob* cn=*Johnson cn=*John* cn=B*John になります。アスタリスク (*) はゼロ (0) 以上の文字を示します。
以上	>=	指定された値以上の属性を含むエントリーを返します たとえば、build name >= alpha です。
より小か等しい	<=	指定された値以下の属性が含まれるエントリーを返します たとえば、builds name <= alpha のようになります。
存在	=*	指定属性の 1 つ以上の値が含まれるエントリーを返します （例： cn=* telephoneNumber=* manager=*）。
概算値	~=	指定した属性を含むエントリーを、検索フィルターで指定された値とほぼ等しい値で返します。たとえば、cn~=suret l~=san fransico は cn=sarette l=san francisco を返す可能性があります。

14.4.3. 複合検索フィルターの使用

複数の検索フィルターコンポーネントは、以下のように接頭辞表記で表現されるブール値演算子を使用して組み合わせることができます。

```
(Boolean-operator(filter)(filter)(filter)...)
```

boolean-operator は、表14.2「検索フィルターのブール値演算子」に一覧表示されているブール値演算子の 1 つになります。

たとえば、このフィルターは、指定された値を含まないすべてのエントリーを返します。

```
(!(cn=Ray Kultgen))
(!(objectClass=person))
```

当然のことながら、複合検索フィルターは、入れ子にして完成された表現にしたときに最も有効です。

```
(Boolean-operator(filter)((Boolean-operator(filter)(filter)))
```

これらの複合フィルタは、他のタイプの検索 (近似、部分文字列、その他の演算子) と組み合わせることで、非常に詳細な結果を得ることができます。たとえば、このフィルターは組織単位が Marketing で、説明フィールドに部分文字列 X.500 が含まれていないすべてのエントリーを返します。

```
(&(ou=Marketing)(!(description=*X.500*)))
```

このフィルターは、組織単位が Marketing で、部分文字列 X.500 を持たないエントリーや、 Julie Fulmer または Cindy Zwaska をマネージャーとして持つエントリーを返すように拡張できます。

```
(&(ou=Marketing)(!(description=*X.500*))(|(manager=cn=Julie
Fulmer,ou=Marketing,dc=example,dc=com)(manager=cn=Cindy
Zwaska,ou=Marketing,dc=example,dc=com)))
```

このフィルターは、人を表すことなく、共通名が printer3b と似たすべてのエントリーを返します。

```
(&(!(objectClass=person))(cn~=printer3b))
```

表14.2 検索フィルターのブール値演算子

演算子	記号	説明
AND	&	文が true になるには、指定したフィルターはすべて true である必要があります。例: (&(filter)(filter)(filter)...)
OR	|	文が true になるには、少なくとも 1 つのフィルターを true にする必要があります。例: (|(filter)(filter)(filter)...)
NOT	!	文が true になるには、指定の文が true にならないようにする必要があります。NOT 演算子の影響を受けるフィルターは 1 つだけです。例:(!(filter))

ブール値は、以下の順番で評価されます。

- 一番内側から外側に向かって、親表現が優先されます。
- 左から右へのすべての式。

14.4.4. 一致するルールの使用

マッチングルール は、Directory Server に対して、2 つの値 (属性に保存されている値および検索フィルターの値) を比較する方法を説明します。マッチングルールは、インデックスキーの生成方法も定義します。マッチングルールは、属性構文に関連するものです。構文は属性値の 形式 を定義します。マッチングルールは、形式が比較およびインデックス化される方法を定義します。

マッチングルールは 3 種類あります。

- EQUALITY は、同じ一致の 2 つの値を比較する方法を指定します。たとえば、「Fred」および「FRED」などの文字列の処理方法です。等価をテストする検索フィルター (attribute=value など) は EQUALITY ルールを使用します。等価 (eq) インデックスは EQUALITY ルールを使用

してインデックスキーを生成します。更新操作は EQUALITY ルールを使用して、値を比較して、エントリーにすでにある値と比較します。

- ORDERING は、2 つの値を比較して、ある値が別の値以上であるかを確認できます。範囲 (例: attribute<=value または attribute>=value) を設定する検索フィルターは、ORDERING ルールを使用します。ORDERING ルールを持つ属性のインデックスは等価値の順序です。
- SUBSTR は、部分文字列照合を行う方法を指定します。部分文字列検索フィルター (例: attribute=*partial_string* または attribute=*end_string) は SUBSTR ルールを使用します。部分文字列 (サブ) インデックスは SUBSTR ルールを使用してインデックスを生成します。

重要

対応する検索フィルターまたはインデックスタイプの検索またはインデックスをサポートするには、マッチングルールが必要です。たとえば、ある属性の等価検索フィルターや eq インデックスをサポートするには、その属性に EQUALITY マッチングルールが必要です。範囲検索フィルターとインデックス化の範囲検索に対応するために、属性に ORDERING マッチングルールと EQUALITY マッチングルールの両方が必要です。

マッチングルールのない属性の検索フィルターの使用を試みた場合は、PROTOCOL_ERROR または UNWILLING_TO_PERFORM で検索操作は拒否されます。

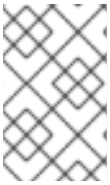
例14.1 マッチングルールおよびカスタム属性

Example Corp. 管理者は、IA5 文字列 (7 ビット ASCII) 構文で ***MyFirstName*** という名前のカスタム属性タイプと、caseExactIA5Match の EQUALITY マッチングルールを作成します。***MyFirstName*** の値が **Fred** のエントリーは、**(MyFirstName=Fred)** のフィルターを使用した検索で返されますが、**(MyFirstName=FRED)** および **(MyFirstName=fred)** のフィルターでは返されません。**Fred**、**FRED**、および **fred** はすべて有効な IA5 文字列値ですが、caseExactIA5Match ルールを使用した場合一致しません。

検索で返される Fred の 3 つのすべてのバリアントについては、caseIgnoreIA5Match マッチングルールを使用するように ***MyFirstName*** を定義する必要があります。

拡張されたマッチングルール検索フィルターを使用すると、属性に定義されたルールとは異なるマッチングルールを持つ属性値を検索できます。マッチングルールは、検索される属性の構文と互換性がある必要があります。たとえば、大文字と小文字を区別するマッチングルールが定義されている属性に対して大文字と小文字を区別しない検索を行うには、検索フィルターに大文字と小文字を区別しないマッチングルールを指定します。

(MyFirstName:caseIgnoreIA5Match:**=fred)**



注記

マッチングルールは、国際化されたディレクトリーの検索に使用され、結果に使用する言語タイプを指定します。詳細は、「国際化されたディレクトリーの検索」を参照してください。

注記

属性のインデックスは、その属性のスキーマ定義で定義されているマッチングルールを使用します。インデックスに使用する追加のマッチングルールは、「コマンドラインからのインデックスの作成」にあるように ***nsMatchingRule*** 属性を使用して設定できます。

マッチングルールフィルターの構文では、一致するルール名または OID が検索フィルターに挿入されます。

> *attr:matchingRule:=value*

- attr は、***cn*** や ***mail*** など、検索されるエントリーに属する属性です。
- matchingRule は、必要な構文に従って属性値と一致するために使用するルールの名前または OID を含む文字列です。
- value は、検索する属性値か、比較演算子および検索する属性値のいずれかです。フィルターの値の構文は、使用されるマッチングルール形式によって異なります。

マッチングルールは実際にはスキーマ要素であり、他のスキーマ要素と同様に、オブジェクト識別子 (OID) によって一意に識別されます。

Red Hat Directory Server 向けに定義されたマッチングルールの多くは言語コードに関連し、Directory Server によってサポートされる国際化された照合順序が設定されます。たとえば、OID **2.16.840.1.113730.3.3.2.17.1** はフィンランドの照合順序を識別します。

注記

その他のスキーマ要素とは異なり、Directory Server の設定には、追加のマッチングルールを追加できません。

以下の一覧におけるマッチングルール一覧のほとんどは、等価インデックスに使用されます。名前に 順序 を含むマッチングルールは順序インデックスに、名前に 部分文字列 を含むマッチングルールは部分文字列 (SUBSTR) インデックスに使用されます。(国際的な一致や照合順序に用いられるマッチングルールは、別の命名法を用いています。)

ビット単位の AND 一致

ビット単位の **AND** 一致を実行します。

OID: 1.2.840.113556.1.4.803

互換性のある構文: 通常、**Integer** および数値の文字列で使用されます。Directory Server は自動的に数値の文字列を整数に変換します。

ビット単位の OR 一致

ビット単位の **OR** 一致を実行します。

OID: 1.2.840.113556.1.4.804

互換性のある構文: 通常、**Integer** および数値の文字列で使用されます。Directory Server は自動的に数値の文字列を整数に変換します。

booleanMatch

照合する値が TRUE または FALSE かを評価します。

OID: 2.5.13.13

互換性のある構文: ブール値

caseExactIA5Match
: 値の大文字と小文字を区別する比較を行います。

OID: 1.3.6.1.4.1.1466.109.114.1

互換性のある構文: IA5 構文、URI

caseExactMatch
: 値の大文字と小文字を区別する比較を行います。

OID: 2.5.13.5

互換性のある構文: Directory String、Printable String、OID

caseExactOrderingMatch
: 大文字と小文字を区別する範囲検索が可能になります (「より小さい」および「より大きい」)。

OID: 2.5.13.6

互換性のある構文: Directory String、Printable String、OID

caseExactSubstringsMatch
: 大文字と小文字を区別した部分文字列とインデックスの検索を実行します。

OID: 2.5.13.7

互換性のある構文: Directory String、Printable String、OID

caseIgnoreIA5Match
: 値に対して大文字と小文字を区別しない比較を実行します。

OID: 1.3.6.1.4.1.1466.109.114.2

互換性のある構文: IA5 構文、URI

caseIgnoreIA5SubstringsMatch
: 部分文字列およびインデックスで大文字と小文字を区別しない検索を実行します。

OID: 1.3.6.1.4.1.1466.109.114.3

互換性のある構文: IA5 構文、URI

caseIgnoreListMatch
: 値に対して大文字と小文字を区別しない比較を実行します。

OID: 2.5.13.11

互換性のある構文: 住所

caseIgnoreListSubstringsMatch

部分文字列およびインデックスで大文字と小文字を区別しない検索を実行します。

OID: 2.5.13.12

互換性のある構文: 住所

caseIgnoreMatch

値に対して大文字と小文字を区別しない比較を実行します。

OID: 2.5.13.2

互換性のある構文: Directory String、 Printable String、 OID

caseIgnoreOrderingMatch

大文字と小文字を区別しない範囲検索が可能になります (「より小さい」および「より大きい」)。

OID: 2.5.13.3

互換性のある構文: Directory String、 Printable String、 OID

caseIgnoreSubstringsMatch

部分文字列およびインデックスで大文字と小文字を区別しない検索を実行します。

OID: 2.5.13.4

互換性のある構文: Directory String、 Printable String、 OID

distinguishedNameMatch

識別名の値を比較します。

OID: 2.5.13.1

互換性のある構文: 識別名(DN)

generalizedTimeMatch

一般化された時間形式の値を比較します。

OID: 2.5.13.27

互換性のある構文: 一般化時刻

generalizedTimeOrderingMatch

一般化された時間形式の値の範囲検索 (「より小さい」および「より大きい」) が可能になります。

OID: 2.5.13.28

互換性のある構文: 一般化時刻

integerMatch

整数値を評価します。

OID: 2.5.13.14

互換性のある構文: 整数

integerOrderingMatch
: 整数値に範囲化された検索が可能になります (「より小さい」および「より大きい」)。

OID: 2.5.13.15

互換性のある構文: 整数

keywordMatch
: 指定した検索値を、属性値の文字列と比較します。

OID: 2.5.13.33

互換性のある構文: ディレクトリー文字列

numericStringMatch
: より一般的な数値を比較します。

OID: 2.5.13.8

互換性のある構文: 数値文字列

numericStringOrderingMatch
: 複数の一般的な値に対する範囲検索 (「より小さい」および「より大きい」) が可能になります。

OID: 2.5.13.9

互換性のある構文: 数値文字列

numericStringSubstringMatch
: より一般的な数値を比較します。

OID: 2.5.13.10

互換性のある構文: 数値文字列

objectIdentifierMatch
: オブジェクト識別子 (OID) 値を比較します。

OID: 2.5.13.0

互換性のある構文: OID

octetStringMatch
: octet 文字列の値を評価します。

OID: 2.5.13.17

互換性のある構文: オクテット文字列

octetStringOrderingMatch
一連のオクテット文字列値で範囲検索 (「より小さい」および「より大きい」) をサポートします。

OID: 2.5.13.18

互換性のある構文: オクテット文字列

telephoneNumberMatch
電話番号の値を評価します。

OID: 2.5.13.20

互換性のある構文: 電話番号

telephoneNumberSubstringsMatch
電話番号の値に対して部分文字列とインデックス検索を行います。

OID: 2.5.13.21

互換性のある構文: 電話番号

uniqueMemberMatch
名前と UID の値を比較します。

OID: 2.5.13.23

互換性のある構文: 名前および任意の UID

wordMatch
指定した検索値を、属性値の文字列と比較します。このマッチングルールは大文字と小文字を区別しません。

OID: 2.5.13.32

互換性のある構文: ディレクトリー文字列

表14.3 言語順序のマッチングルール

マッチングルール	オブジェクト識別子 (OID)
英語 (大文字と小文字を区別する順序の一致)	2.16.840.1.113730.3.3.2.11.3
アルバニア語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.44.1
アラビア語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.1.1
ベラルーシ語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.2.1

マッチングルール	オブジェクト識別子 (OID)
ブルガリア語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.3.1
カタロニア語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.4.1
中国語: 簡体字 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.49.1
中国語: 繁体字 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.50.1
クロアチア語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.22.1
チェコ語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.5.1
デンマーク語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.6.1
オランダ語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.33.1
オランダ語: ベルギー (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.34.1
英語 - アメリカ (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.11.1
英語 - カナダ語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.12.1
英語: アイルランド (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.14.1
エストニア語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.16.1
フィンランド語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.17.1
フランス語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.18.1
フランス語: ベルギー (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.19.1

マッチングルール	オブジェクト識別子 (OID)
フランス語 - カナダ語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.20.1
フランス語 - スイス (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.21.1
ドイツ語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.7.1
ドイツ語 - オーストリア語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.8.1
ドイツ語: スイス (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.9.1
ギリシャ語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.10.1
ヘブライ語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.27.1
ハンガリー語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.23.1
アイスランド語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.24.1
イタリア語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.25.1
イタリア語: スイス (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.26.1
日本語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.28.1
韓国語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.29.1
ラトビア語、レット語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.31.1
リトアニア語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.30.1
マケドニア語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.32.1
ノルウェー語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.35.1

マッチングルール	オブジェクト識別子 (OID)
ノルウェー語 - ブークモール (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.36.1
ノルウェー語 - ニーノルスク (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.37.1
ポーランド語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.38.1
ルーマニア語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.39.1
ロシア語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.40.1
セルビア語 - キリル文字 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.45.1
セルビア語 - ラテン語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.41.1
スロバキア語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.42.1
スロベニア語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.43.1
スペイン語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.15.1
スウェーデン語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.46.1
トルコ語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.47.1
ウクライナ語 (大文字と小文字を区別しない順序一致)	2.16.840.1.113730.3.3.2.48.1

表14.4 言語部分文字列マッチングルール

マッチングルール	オブジェクト識別子 (OID)
英語 (大文字と小文字を区別する部分文字列の一致)	2.16.840.1.113730.3.3.2.11.3.6
アルバニア語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.44.1.6

マッチングルール	オブジェクト識別子 (OID)
アラビア語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.1.1.6
ベラルーシ語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.2.1.6
ブルガリア語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.3.1.6
カタロニア語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.4.1.6
中国語 - 簡体字 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.49.1.6
中国語 - 繁体字 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.50.1.6
クロアチア語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.22.1.6
チェコ語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.5.1.6
デンマーク語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.6.1.6
オランダ語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.33.1.6
オランダ語: ベルギー (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.34.1.6
英語 - アメリカ (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.11.1.6
英語: カナダ (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.12.1.6
英語: アイルランド (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.14.1.6
エストニア語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.16.1.6
フィンランド語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.17.1.6

マッチングルール	オブジェクト識別子 (OID)
フランス語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.18.1.6
フランス語: ベルギー (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.19.1.6
フランス語: カナダ (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.20.1.6
フランス語: スイス (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.21.1.6
ドイツ語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.7.1.6
ドイツ語 - オーストリア (大文字小文字を区別しない文字列一致)	2.16.840.1.113730.3.3.2.8.1.6
ドイツ語: スイス (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.9.1.6
ギリシャ語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.10.1.6
ヘブライ語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.27.1.6
ハンガリー語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.23.1.6
アイスランド語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.24.1.6
イタリア語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.25.1.6
イタリア語: スイス (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.26.1.6
日本語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.28.1.6
韓国語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.29.1.6
ラトビア語、レット語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.31.1.6

マッチングルール	オブジェクト識別子 (OID)
リトアニア語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.30.1.6
マケドニア語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.32.1.6
ノルウェー語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.35.1.6
ノルウェー語 - ブークモール (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.36.1.6
ノルウェー語 - ニーノルスク (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.37.1.6
ポーランド語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.38.1.6
ルーマニア語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.39.1.6
ロシア語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.40.1.6
セルビア語 - キリル文字 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.45.1.6
セルビア語 - ラテン語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.41.1.6
スロバキア語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.42.1.6
ストベニア語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.43.1.6
スペイン語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.15.1.6
スウェーデン語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.46.1.6
トルコ語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.47.1.6
ウクライナ語 (大文字と小文字を区別しない部分文字列一致)	2.16.840.1.113730.3.3.2.48.1.6

14.5. 一般的な LDAPSEARCH の例

次の例セットは、以下を想定しています。

- 検索は、ディレクトリー内のすべてのエントリーに対するものです。
- このディレクトリーは、検索および読み取りの匿名アクセスをサポートするように設定されます。つまり、検索を実行するためにバインド情報を提供する必要はありません。匿名アクセスの詳細は、「匿名アクセスの付与」を参照してください。
- サーバーは、server.example.com という名前のホストにあります。
- サーバーはポート番号 389 を使用します。これはデフォルトのポートであるため、ポート番号は検索リクエストで送信する必要がありません。
- TLS は、ポート 636 のサーバー (デフォルトの LDAPS ポート番号) に対して有効になります。
- すべてのデータが格納される接尾辞は dc=example,dc=com です。

14.5.1. すべてのエントリーの返信

以前の情報を指定すると、以下の呼び出しはディレクトリー内のすべてのエントリーを返します (設定されるサイズおよび時間制限に従います)。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b
"dc=example,dc=com" -s sub -x "(objectclass=*)"
```

"objectclass=*" は、このディレクトリー内のエントリーに一致する検索フィルターです。すべてのエントリーにオブジェクトクラスが含まれる必要があり、*objectclass* 属性は常にインデックス化されるため、すべてのエントリーを返すための便利な検索フィルターになります。

14.5.2. コマンドラインでの検索フィルターの指定

フィルターが引用符 ("filter") で囲まれている場合、検索フィルターはコマンドラインで直接指定することができます。フィルターがコマンドで提供されている場合は、-f オプションを指定しないでください。以下に例を示します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b
"dc=example,dc=com" -s sub -x "cn=babs jensen"
```

14.5.3. ルート DSE エントリーの検索

ルート DSE は、ローカルの Directory Server でサポートされるすべての接尾辞を含む、ディレクトリーサーバーのインスタンスに関する情報が含まれる特別なエントリーです。このエントリーは、"" の検索ベース、base の検索範囲、および "objectclass=*" のフィルターを指定して検索できます。以下に例を示します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -b "" -s base
"objectclass=*"
```

14.5.4. スキーマエントリーの検索

cn=schema エントリーは、オブジェクトクラスや属性タイプなどのディレクトリースキーマに関する情報が含まれる特別なエントリーです。

以下のコマンドは、cn=schema エントリーの内容を一覧表示します。

```
# ldapsearch -o ldif-wrap=no -D "cn=Directory Manager" -W -b "cn=schema" \
    '(objectClass=subSchema)' -s sub objectClasses attributeTypes matchingRules \
    matchingRuleUse  dITStructureRules nameForms ITContentRules ldapSyntaxes
```

14.5.5. LDAP_BASEDN の使用

検索を容易にするには、LDAP_BASEDN 環境変数を使用して検索ベースを設定できます。これを行うと、検索ベースは -b オプションで設定する必要があります。環境変数の設定方法については、オペレーティングシステムのドキュメントを参照してください。

通常、LDAP_BASEDN をディレクトリーのサフィックスの値に設定します。ディレクトリーの接尾辞は、ディレクトリーのルート (最上位の) エントリーと等しいため、これにより、すべての検索はディレクトリーのルートエントリーから始まることになります。

たとえば、LDAP_BASEDN を dc=example,dc=com に設定し、ディレクトリー内の cn=babs jensen を検索するには、以下のコマンドライン呼び出しを使用します。

```
# export LDAP_BASEDN="dc=example,dc=com"

# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x "cn=babs
jensen"
```

この例では、-s オプションがスコープの指定に使用されないため、sub のデフォルトスコープが使用されます。

14.5.6. 属性のサブセットの表示

ldapsearch コマンドは、すべての検索結果を LDIF 形式で返します。デフォルトでは、ldapsearch はエントリーの識別名と、ユーザーが読み取りできるすべての属性を返します。ディレクトリーアクセス制御は、指定されたディレクトリーエントリーの属性のサブセットのみをユーザーが読み取りできるように設定できます。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場合は、search コマンドで明示的に指定するか、*+* を使用してすべての操作属性を返します。

検索結果で返されるエントリーのすべての属性を作成する必要はない場合があります。返された属性は、検索フィルターの直後にコマンドラインに必要な属性を指定することにより、いくつかの特定の属性のみに限定できます。たとえば、ディレクトリー内のすべてのエントリーの *cn* 属性および *sn* 属性を表示するには、コマンドライン呼び出しを使用します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b
"dc=example,dc=com" -s sub -x "(objectclass=*)" sn cn
```

14.5.7. 操作属性の検索

操作属性は、アクセス制御命令を処理するなどのメンテナンスタスクを実行するためにサーバーによって使用される Directory Server 自体によって設定される特別な属性です。また、最初に作成された時間や、作成したユーザーの名前など、エントリーに関する特定の情報も表示します。操作属性は、属性がエントリーのオブジェクトクラスに対して属性が特別に定義されているかどうかにかかわらず、ディレクトリー内のすべてのエントリーで使用することができます。

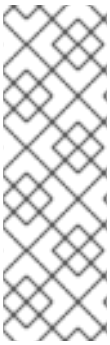
運用上の属性は、通常の ldapsearch で返されません。RFC3673 に従って、+ を使用して検索要求の操作属性をすべて返します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b
"dc=example,dc=com" -s sub -x "(objectclass=*)" '+'
```

定義された操作属性のみを返すには、ldapsearch リクエストに明示的に指定します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b
"dc=example,dc=com" -s sub -x "(objectclass=*)" creatorsName createTimestamp
modifiersName modifyTimestamp
```

操作属性の完全なリストは、『Red Hat Directory Server 10 の設定、コマンド、およびファイルリファレンス の「操作属性およびオブジェクトクラス」の章にあります。』



注記

指定した操作属性とともにすべての通常のエントリー属性を返すには、記載されている操作属性に加えて、特別な search 属性 "*" を使用します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b
"dc=example,dc=com" -s sub -x "(objectclass=*)" "*" aci
```

シェルで解釈されないように、アスタリスクを引用符で囲む必要があります。

14.5.8. ファイルを使用した検索フィルターの指定

検索フィルターは、コマンドラインで入力するのではなく、ファイルに入力できます。この場合は、ファイル内の個別の行に各検索フィルターを指定します。ldapsearch コマンドは、ファイルに表示される順序で各検索を実行します。

以下に例を示します。

```
sn=Francis
givenname=Richard
```

ldapsearch は、最初に、姓 Francis を持つすべての エントリーを検索し、次に givennameRichard を持つすべてのエントリーを検索します。両方の検索条件に一致するエントリーが見つかると、エントリーは 2 回返されます。

たとえば、この検索では、searchdb という名前のファイルにフィルターを指定します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -f searchdb
```

ここで返される属性のセットは、検索行の最後に属性名を指定すると制限されます。たとえば、以下の ldapsearch コマンドは両方の検索を実行しますが、各エントリーの DN、*givenname* 属性、および *sn* 属性のみが返されます。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -f searchdb sn
givenname
```

14.5.9. 検索フィルターでコンマを含む DN の指定

検索フィルター内の DN に値の一部としてコンマが含まれている場合は、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、example.com Bolivia, S.A. サブツリーですべてのユーザーが検索するには、以下のコマンドを使用します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -s base -b
"l=Bolivia\,S.A.,dc=example,dc=com" "objectclass=*"
```

14.5.10. クライアント証明書の Directory Server へのバインド

「証明書を使用した認証」を参照してください。

14.5.11. 言語マッチングルールでの検索

検索フィルターでマッチングルールを明示的に送信するには、属性の後にマッチングルールを挿入します。

```
attr:matchingRule:=value
```

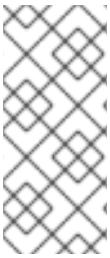
マッチングルールは、国際化されたディレクトリーの検索に頻繁に使用されます。たとえば、これにより、スウェーデン語(2.16.840.1.113730.3.3.2.46.1)のマッチングルールの N4709 以降の部署番号のアーキテクチャなどがあります。

```
departmentNumber:2.16.840.1.113730.3.3.2.46.1:=>= N4709
```

「国際化されたディレクトリーの検索」に、国際化された検索の実行例がいくつか含まれています。

14.5.12. Bit Field の値での属性の検索

ビット単位検索は、ビットフィールドの値を持つ属性に対してビット単位の検索作業を行うビット単位の AND またはビット単位の OR マッチングルールを使用します。



注記

ビットフィールドの値が含まれる属性は LDAP で一般的ではありません。(デフォルトの Directory Server スキーマは、ビットフィールドを属性構文として使用しません。) ただし、複数の LDAP 構文は整数形式の値をサポートします。カスタム属性はビットフィールド値を使用して定義でき、アプリケーションはこれらのカスタム属性を使用してビットフィールドの値に対してビット単位の操作を実行できます。

ビット単位 AND マッチングルール (1.2.840.113556.1.4.803) は、アサーション値に指定されたビットがビットフィールド属性値に設定されていることを確認します。(これは等価検索に類似しています) この例では、userAccountControl の値は、2 を表すビットに設定する必要があります。

```
"(UserAccountControl:1.2.840.113556.1.4.803:=2)"
```

この例では、userAccountControl の値には、値 6 (ビット 2 および 4) に設定されるすべてのビットセットが必要になります。

```
"(UserAccountControl:1.2.840.113556.1.4.803:=6)”
```

ビット単位 OR マッチングルール (1.2.840.113556.1.4.804) は、アサーション文字列のビットの いずれか が属性値で表されるかどうかを確認します。(これは部分文字列検索に似ています) この例では、

userAccountControl の値には、6 のビットフィールドに設定されるビットのいずれかが必要です。つまり、属性値は 2、4、または 6 のいずれかになります。

```
"(UserAccountControl:1.2.840.113556.1.4.804:=6)"
```

ビット単位検索は、Samba ファイルサーバーの使用など、Windows と Red Hat Enterprise Linux の統合で使用することができます。

注記

Microsoft 社は、ビット単位の http://msdn.microsoft.com/en-us/library/aa746475 に関する適切なドキュメントです。

14.6. 永続検索の使用

永続的な検索は **ldapsearch** で、最初の検索結果が返されても開かれたままになります。

重要

Red Hat Enterprise Linux の OpenLDAP クライアントツールは、永続的な検索に対応していません。ただし、サーバー自体は機能します。その他の LDAP クライアントは、永続的な検索を実行するために使用する必要があります。

永続検索の目的は、ディレクトリーエントリーへの変更の継続的なリストと、ハイブリッド検索や changelog などの完全なエントリー自体を提供することです。そのため、検索コマンドでは、どのエントリーを返すか (検索パラメーター)、どのような変更によってエントリーが返されるか (エントリー変更パラメーター) を指定する必要があります。

永続的な検索は、Directory Server にアクセスするアプリケーションまたはクライアントで特に役立ち、2 つの重要な利点を提供します。

- 一貫性のあるローカルキャッシュと現在のローカルキャッシュを保持します。

  クライアントは、ディレクトリーに接続してクエリーを試行する前にローカルキャッシュをクエリーします。永続検索は、これらのクライアントのパフォーマンスを改善するのに必要なローカルキャッシュを提供します。

- ディレクトリーアクションを自動的に開始します。

  永続キャッシュはエントリーの変更時に自動的に更新され、永続検索結果ではエントリーに対して実行された変更の種類を表示できます。別のアプリケーションでは、その出力を利用してエントリーを自動的に更新することができます。たとえば、新しいユーザーのためにメールサーバーにメールアカウントを自動的に作成したり、固有のユーザー ID 番号を生成したりすることができます。

永続検索を実行する場合のパフォーマンスに関する考慮事項がいくつかあります。

- クライアントの接続解除時に **ldapsearch** は通知を送信せず、検索が切断されている間に変更についての通知が送信されません。これは、クライアントのキャッシュが切断されても更新されないことを意味し、切断中に変更した新しいエントリー、変更されたエントリー、または削除されたエントリーでキャッシュを更新する適切な方法はありません。

- 攻撃者は、サービス拒否攻撃を開始するために多数の永続検索を開くことができます。

- 永続的な検索では、Directory Server とクライアント間で TCP 接続を開放する必要があります。これは、サーバーが多くのクライアント接続を許可し、アイドル状態の接続を閉じる方法を持つ場合にのみ行う必要があります。

アクセスログでは、永続検索はタグ options=persistent で識別されます。

```
[12/Jan/2009:12:51:54.899423510 -0500] conn=19636710736396323 op=0 SRCH
base="dc=example,dc=com" scope=2 filter="(objectClass=person)" attrs=ALL
options=persistent
```

14.7. 指定したコントロールでの検索

Directory Server は、DSE の ***supportedControls*** 属性に制御を定義しています。これらの中には、レプリケーションのようなサーバーの操作を定義するものもあれば、Get Effective Rights や、クライアントが LDAP 操作でサーバーに渡すことができる制御の逆参照などの拡張操作を許可するものもあります。

これらの制御は、**-E** オプションを使用して指定できます。制御 OID、ldapsearch の重大度、およびその制御操作に必要な情報を指定します。

```
-E '[!]control_OID:control_information'
```

一部の制御 (サーバー側のソートやシンプルなページングされた結果など、) には、検索操作にコントロールを渡すために使用できるエイリアスがあります。制御エイリアスを使用すると、制御がクライアントによって認識されるため、結果がフォーマットされます。

14.7.1. 効果のあるユーザー権限の取得

有効な権利を取得する検索コントロールは、コントロール OID を使用して渡されます。以下に例を示します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b
"dc=example,dc=com" -s sub -x -E
'!1.3.6.1.4.1.42.2.27.9.5.2=:dn:uid=jsmith,ou=people,dc=example,dc=com' "(objectclass=*)"
```

重要

コントロールの OID が渡されると、検索結果の形式化がされません。

Get Effective Rights の検索については、「アクセス制御」の章「エントリーのアクセス権利の確認 (Get Effective Rights)」で詳細に説明されています。

14.7.2. サーバー側のソートの使用

サーバー側のソートは、**-E** フラグと **sss** 制御エイリアスを使用して、他の制御操作として実行されます。操作の構造は、結果をソートし、任意でソート順序および順序ルールである属性を設定します。

```
-E sss=[-]attribute_name:[ordering_rule_OID]
```

ダッシュ (-) は、ソート順序を元に戻す任意のフラグで、降順の降順を逆に実行します。「一致するルールの使用」 のマッチングルールテーブルには、Directory Server でサポートされる順序ルールが含まれています。

以下に例を示します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b
"dc=example,dc=com" -s sub -x -E sss=-uidNumber:2.5.13.15 "(objectclass=*)"
```

14.7.3. 逆参照検索の実行

逆参照 検索は、エントリーの相互参照を追跡し、参照されたエントリーに関する情報を返す簡単な方法です。たとえば、グループエントリーには、そのメンバーのユーザーエントリーへの参照が含まれます。通常検索は、最初にグループを検索し、そのメンバーを一覧表示し、各メンバーに個別の検索が必要になります。グループエントリーの逆参照検索は、メンバーに関する情報 (場所、メールアドレス、マネージャーなど) を 1つの検索要求でグループの情報と ともに 返します。

逆参照は多くのクライアント操作を簡素化し、実行した検索操作の数を減らします。クロスリンクは、エントリー間の関係を表示します。一部の操作では、1つのエントリーから複数のリンクの一覧を取得し、後続の検索を実行してリスト上の各エントリーから情報を取得しないといけない場合があります。逆参照により、検索のシーケンスを 1つの検索に統合することができます。

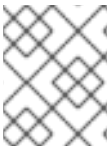
重要

逆参照操作は、OpenLDAP コマンドラインツールのバージョン 2.4.18 以降、または逆参照検索をサポートするその他のクライアントを使用して行う必要があります。

逆参照引数の形式は次のとおりです。

```
-E 'deref=deref_attribute:list_of_attributes'
```

deref_attribute は、参照が含まれる検索ターゲットの属性です。これには、member または manager などの値に DN を持つ属性を使用できます。

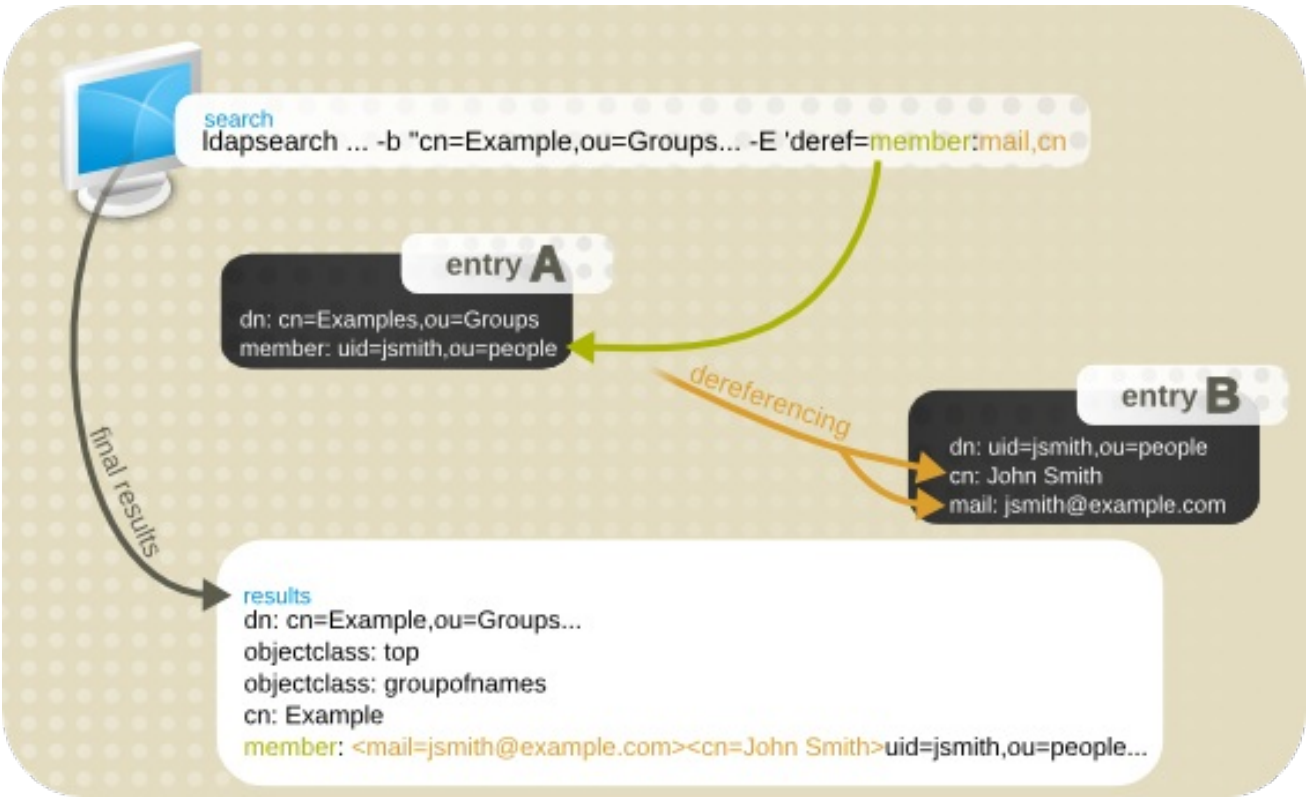


注記

deref_attribute の値は DN にする必要がありますが、この属性に対する実際の定義された構文は DN 構文 (1.3.6.1.4.1.1466.115.121.1.12) である必要があります。

list_of_attributes は、参照されたエントリーの 1つ以上の属性で、プライマリーの検索結果とともに返されます。l,mail,cn のように、複数の属性をコンマで区切ることができます。

図14.3 簡単な参照検索コマンド



検索引数で要求された逆参照された情報は、残りの検索結果とともに返されます。たとえば、この逆参照検索は、検索ターゲットエントリー (エンジニアグループ) の ***member*** 属性を deref_attribute として使用するように指示します。その後、各メンバーの locality 属性を返します。

```
# ldapsearch -x -D "cn=Directory Manager" -W -b
"cn=Example,ou=Groups,dc=example,dc=com" -E 'deref=member:mail,cn' "(objectclass=*)"

# Engineers, Groups, example.com
dn: cn=Engineers,ou=Groups,dc=example,dc=com
control: 1.3.6.1.4.1.4203.666.5.16 false MIQAAADNMIQAAAA1BAZtZW1iZXIEK2NuPURId
 mVsb3BlcnMsIG91PUdyb3VwcywgZGM9ZXhhbXBsZSxkYz1jb20whAAAADIEBm1lbWJlcgQoY
249VG
 VzdGVycywgb3U9R3JvdXBzLCBkYz1leGFtcGxlLGRjPWNvbTCEAAAAVAQGbWVtYmVyBCp1
aWQ9ZW5
 nLCBvdT1lbmdpbmVlcmluZywgZGM9ZXhhbXBsZSxkYz1jb22ghAAAABowhAAAABQEAWwx
hAAAAAsE
 CUNhbWJyaWRnZQ==
# member: <mail=jsmith@example.com><cn=John
Smith>;uid=jsmith,ou=people,dc=example,dc=com
 objectClass: top
objectClass: inetuser
objectClass: groupofnames
cn: Engineers
member: uid=jsmith,ou=people,dc=example,dc=com
```

14.7.4. 単純なページ結果の使用

検索結果は非常に大きくなる可能性があり、結果処理の一部が結果を整理することです。その方法の一つとして、ページングされた単純な結果 を使用することがあります。これは、結果を一定の長さのページに分割するコントロールです。

シンプルなページの結果で、一度に表示するエントリーの数を設定できます。結果は一度に 1ページ経由でスクロールされるため、ダイジェストで結果が容易になります。制御の完全な動作は RFC 2696 で説明されています。

ページングされた単純な結果は、Directory Server の LDAP コントロール拡張機能として実装されます。OID は 1.2.840.113556.1.4.319 です。

簡単なページ結果の仕組み

ページ結果の簡単な検索を開始すると、以下のようになります。

1. クライアントは検索をサーバーに送り、ページ付けの結果が制御し、最初のページで返すレコードの数を制御します。
2. Directory Server がデータの返信を開始する前に、サーバーは合計で何件のレコードを返信できるかの見積もりを生成します。

   レコードの推定値は、正確な数ではありません。返されるレコードの合計数は推定値よりも小さくなります。このようなシナリオの理由は以下の通りです。

   - 検索フィルターで使用される属性はインデックスに存在しません。最適な結果を得るには、クエリーされた属性をすべてインデックス化する必要があります。
   - エントリーがクライアントに送信される前に、アクセス制御リスト (ACL) が検証されます。パーミッションが十分にないため、エントリーが返されなくなります。

   推定値を生成した後、サーバーは最初の結果セット、Cookie、および推定レコード数を送信します。
3. 返されたレコードがクライアントに表示されます。ユーザーは、次のリクエストで返されるレコードの数を入力できるようになりました。要求された番号は、Cookie と一緒にサーバーに送信されます。
4. サーバーは要求された数のレコードをデータベースから取得し、それらを Cookie と共にクライアントに送信します。
5. 前述の 2 つのステップは、すべてのレコードが送信されるか、検索がキャンセルされるまで繰り返されます。

シンプルなページ結および OpenLDAP ツール

ldapsearch の簡単なページの結果検索オプションの形式は以下のとおりです。

```
-E pg=size
```

size の値はページサイズまたはページごとに含むエントリー数です。以下に例を示します。

```
ldapsearch -x -D "cn=Directory Manager" -W -b
"ou=Engineers,ou=People,dc=example,dc=com" -E pg=3 "(objectclass=*)" cn

dn: uid=jsmith,ou=Engineers,ou=People,dc=example,dc=com
  cn: John Smith

dn: uid=bjensen,ou=Engineers,ou=People,dc=example,dc=com
  cn: Barbara Jensen

dn: uid=hmartin,ou=Engineers,ou=People,dc=example,dc=com
```

```
    cn: Henry Martin

Results are sorted.
next page size (3): 5
```

末尾のタグには、検索で設定されたページサイズ (カッコ内の数字) が表示されています。コロンの後に、次のページのページサイズが表示されるため、以下に示すように 5 と入力すると、次のページが 5 つのエントリーで開きます。

重要

簡素なページ結果操作は、OpenLDAP コマンドラインツールのバージョン 2.4.18 以降、または Perl Net::LDAP などの簡素なページング結果をサポートするその他のクライアントを使用して実行する必要があります。

ページングされた簡素な結果およびサーバー一致

ページングされた簡素な結果は、サーバー側のソートと併用できます。サーバー側のソートは、クライアントではなくサーバーでソートプロセスを実行する制御です。これは通常、特定のマッチングルールを使用する検索を行います。(この動作は RFC 2891 で定義されています。) OpenLDAP クライアントツールは、簡素なページングされた結果制御でサーバー側のソートをサポートしませんが、Perl Net::LDAP などのその他の LDAP ユーティリティーは両方をサポートします。

1 つの接続に複数の簡素なページングされた結果要求

一部のクライアントは Directory Server への接続を 1 つ開きますが、簡素なページングされた結果拡張を使用する複数の検索要求など、複数の操作要求を送信します。

Directory Server は、複数の簡素なページングされた検索を管理および解釈できます。各検索は、アレイ内のエントリーとして追加されます。ページングされた検索要求が最初に送信されると、Cookie が作成され、検索結果に関連付けられます。結果の各ページはその Cookie で返されます。Cookie は結果の次のページを要求するために使用されます。最後のページでは、Cookie が空になり、結果が終了したことを示します。これにより、各検索結果のセットが個別に保持されます。

1 つの接続に複数の簡素なページングされた結果がある場合、タイムアウト制限は引き続き監視されますが、すべて のオープン検索要求は、いずれか のページ検索が切断される前に、設定された時間制限に到達します。

VLV インデックスと対照的な、簡素なページングされた結果

VLV インデックスは簡素なページングされた結果に類似しているため、それらのインデックスは、結果の閲覧可能な一覧も返すことになります。主な違いは、リストの生成方法です。簡素なページングされた結果は検索ごとに計算されますが、VLV インデックスは永続的なリストになります。全体的に、VLV インデックスは検索速度が速いですが、サーバー側の設定が必要で、サーバーが維持するためのオーバーヘッドがあります。

注記

シンプルなページの結果と VLV インデックスは、同じ検索では使用 できません。簡素なページングされた結果は、すでに参照しているインデックスである VLV インデックスの操作を試みます。VLV インデックスを使用した検索に制御が渡された場合、サーバーは **UNWILLING_TO_PERFORM** エラーを返します。

VLV インデックスの詳細は、「ロージング(VLV)インデックスの作成」を参照してください。

14.7.5. 読み取り前および後のエントリーレスポンス制御

Red Hat Directory Server は、RFC 4527 に準拠した、読み取り前および読み取り後のエントリー応答制御をサポートします。クライアントが応答制御を要求すると、LDAP 検索エントリーが返されます。これには、更新前および更新後に属性の値が含まれます。

事前読み取りの制御が使用されると、LDAP 検索クエリーには変更前に指定した属性の値が含まれます。読み取り後の制御が使用される場合、クエリーには変更後の属性の値が含まれます。両方の制御を同時に使用できます。たとえば、***description*** 属性を更新し、変更前および変更後に値を表示するには、次のコマンドを実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -x \
    -e \!preread=description -e \!postread=description
dn: uid=user,ou=People,dc=example,dc=com
changetype: modify
replace: description
description: new description

modifying entry "uid=user,ou=People,dc=example,dc=com"
control: 1.3.6.1.1.13.1 false ZCkEJXVpZD1qdXNlcixvdT1QZW9wbGUsZGM9ZXhhbXBsZSxk
 Yz1jb20wAA==
# ==> preread
dn: uid=user,ou=People,dc=example,dc=com
description: old description
# <== preread
control: 1.3.6.1.1.13.2 false ZEsEJXVpZD1qdXNlcixvdT1QZW9wbGUsZGM9ZXhhbXBsZSxk
Yz1jb20wIjAgBAtkZXNjcmlwdGlvbjERBA9uZXcgZGVzY3JpcHRpb24=
# ==> postread
dn: uid=user,ou=People,dc=example,dc=com
description: new description
# <== postread
```

第15章 レプリケーションの管理

レプリケーション は、ディレクトリーデータが 1 つの Red Hat Directory Server インスタンスから相互に自動的に同期されるメカニズムで、単一のサーバー設定以外にディレクトリーサービスを拡張する上で重要なメカニズムです。本章では、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、およびカスケードレプリケーションを設定するマスターサーバーおよびコンシューマーサーバーで実行するタスクについて説明します。

15.1. レプリケーションの概要

レプリケーションとは、Directory Server 間でディレクトリデータを自動的に同期させる仕組みです。あらゆる種類（エントリーの追加、変更、または削除など）の更新は、レプリケーションを使用して他の Directory Server に自動的にミラーリングされます。

- 「複製されるディレクトリーユニット」
- 「読み取り/書き込みレプリカおよび読み取り専用レプリカ」
- 「サプライヤーとコンシューマー」
- 「Changelog」
- 「レプリケーション ID」
- 「レプリカ合意」

15.1.1. 複製されるディレクトリーユニット

複製できるディレクトリー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複製できますが、データベース内のサブツリーは複製できません。したがって、ディレクトリーツリーを作成する際に、情報を配信する方法を決定する際に、レプリケーションプランを検討してください。

レプリケーションでは、1 つのデータベースが 1 つの接尾辞に対応する必要もあります。つまり、カスタム分散ロジックを使用して 2 つ以上のデータベースに分散される接尾辞 (または名前空間) は複製できません。このトピックの詳細については、「データベースの作成および維持」を参照してください。

15.1.2. 読み取り/書き込みレプリカおよび読み取り専用レプリカ

レプリケーションに参加するデータベースは レプリカ と呼ばれます。レプリカには、読み書き可能なものと、読み取り専用の 2 種類があります。読み取り/書き込みレプリカ には、ディレクトリー情報のマスターコピーが含まれ、更新できます。読み取り専用のレプリカ サービスは読み取り、検索、および比較要求ですが、読み取り/書き込みレプリカに対する更新操作をすべて参照します。サーバーは、任意の数の読み取り専用または読み書きレプリカを保持できます。

15.1.3. サプライヤーとコンシューマー

別のサーバーでレプリカにコピーされるレプリカを保持するサーバーは、そのレプリカの サプライヤー と呼ばれます。別のサーバーからコピーしたレプリカを保持するサーバーは、そのレプリカの コンシューマー と呼ばれます。通常、サプライヤーサーバーのレプリカは読み取り/書き込みレプリカで、コンシューマーサーバーのレプリカは 2 つの例外を持つ読み取り専用レプリカになります。

- レプリケーションをカスケードするとき、ハブサーバーは、コンシューマーに提供する読み取り専用レプリカを保持します。「カスケードレプリケーション」 に詳細情報があります。

- マルチマスターレプリケーションの場合、マスター は同じ情報についてサプライヤーとコンシューマーの両方を設定します。詳細は、「マルチマスターレプリケーション」 を参照してください。

レプリケーションは、常にサプライヤー・サーバーによって開始され、コンシューマーによって開始されることはありません (サプライヤーが開始するレプリケーション)。サプライヤーが開始するレプリケーションでは、サプライヤーサーバーを設定して、データを複数のコンシューマーサーバーにプッシュできます。

15.1.4. Changelog

すべてのサプライヤーサーバーは、changelog と、サプライヤーまたはハブがそのコンシューマーに送信する必要がある変更の記録です。changelog は、レプリカで発生した変更を記述する特別な種類のデータベースです。次に、サプライヤーサーバーは、マルチマスターレプリケーションの場合には、コンシューマーサーバーに保存されているレプリカまたは他のサプライヤーにこの変更を再生します。

エントリーが変更すると、実行した LDAP 操作を記述する変更レコードが changelog に記録されます。

changelog は、メインデータベースと同じデータベース環境を使用します。メインのデータベースの一部として changelog を実装すると、データベースおよび changelog が常に同期され、必要なデータベースキャッシュサイズが減少し、バックアップと復元操作が簡素化されます。

重要

changelog は、サーバーのシャットダウン時に RUV エントリーをデータベースにのみ書き込み、それ以外は RUV がメモリー内で管理されます。マスターのデータベースをバックアップする場合は、db **2bak.pl** ユーティリティーまたは Directory Server Console を使用します。いずれの方法でも、バックアップを開始する前に RUV がデータベースに書き込まれます。

Directory Server では、changelog はサーバーによる内部使用のみを目的としています。

15.1.5. レプリケーション ID

2 つのサーバー間でレプリケーションが発生すると、レプリケーションプロセスは、レプリケーションマネージャー エントリーと呼ばれる特別なエントリーを使用して、レプリケーションプロトコルの交換を特定し、ディレクトリーデータへのアクセスを制御します。レプリケーションマネージャーエントリーまたはレプリケーション中に使用されるエントリーは、以下の基準を満たしている必要があります。

- これは、サプライヤーサーバー ではなく、コンシューマーサーバーに作成されます。
- 別のサーバーから更新を受け取る すべて のサーバー (つまり、すべてのハブまたは専用のコンシューマー) でこのエントリーを作成します。
- レプリカがコンシューマーまたはハブとして設定されている場合は、このエントリーを、レプリケーションの更新を実行する権限のあるものとして指定する必要があります。
- レプリカ合意はサプライヤーサーバーで作成され、このエントリーの DN をレプリカ合意に指定する必要があります。
- このエントリーは、特別なユーザープロファイルで、そのレプリカ合意に関係するデータベースのコンシューマーサーバーで定義されるアクセス制御ルールをすべて回避します。

注記

Directory Server コンソールでは、このレプリケーションマネージャーエントリーは サプライヤーバインド DN と呼ばれます。これは、エントリーが実際にサプライヤーサーバーに存在しないため、誤解を招く可能性があります。これは、サプライヤーがコンシューマーにバインドするために使用するエントリーであるため、サプライヤー DN と呼ばれます。このエントリーは実際には、コンシューマーに存在します。

レプリケーションマネージャーエントリーの作成方法は、「サプライヤーバインド DN エントリーの作成」を参照してください。

15.1.6. レプリカ合意

Directory Server はレプリカ合意を使用してレプリケーション設定を定義します。レプリカ合意は、1 つ のサプライヤーと1つ のコンシューマーとの間のレプリケーションのみを説明します。この合意はサプライヤーサーバーに設定し、必要なレプリケーション情報をすべて指定する必要があります。

- 複製されるデータベース。
- データがプッシュされるコンシューマーサーバー
- レプリケーションが実行する曜日および時間帯
- サプライヤーサーバーがバインドに使用する必要のある DN および認証情報 (レプリケーションマネージャーエントリーまたはサプライヤーバインド DN)
- 接続をセキュアにする方法 (TLS、クライアント認証)。
- 複製されない属性 (一部レプリケーション)

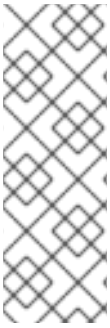
15.1.7. 一部レプリケーションを使用した属性のサブセットの複製

一部レプリケーションは、サプライヤーからコンシューマー (または別のサプライヤー) に送信されない特定の属性のサブセットを設定します。したがって、管理者は、含まれるすべての情報を複製したり、全エントリーのすべての情報を複製せずに、データベースを複製できます。

一部レプリケーションは、エントリーごとではなく、レプリカ合意ごとに有効になり、設定されます。レプリケーションから属性を除外すると、レプリカ合意の範囲内のすべてのエントリーと同等に適用されます。

コンシューマーサーバーに関する限り、除外された属性には常に値がありません。そのため、コンシューマーサーバーに対する検索を実行するクライアントは、除外された属性は表示されません。同様に、フィルターでそれらの属性を指定する検索を実行しても、一致するエントリーはありません。

スキーマで任意 (MAY キーワード) として定義された属性については、増分更新と全体更新で複製する属性を別々に設定することができます。増分更新リスト (*nsDS5ReplicatedAttributeList*) は、一部レプリケーションを有効にするために常に設定される必要があります。唯一の属性が設定されている場合は、増分更新と合計更新の両方に適用されます。任意の *nsDS5ReplicatedAttributeListTotal* 属性は、更新の合計に追加の一部レプリケーション一覧を設定します。これは、「合計更新および増分更新での異なる一部レプリケーション属性の設定」で説明されています。



注記

除外された属性への更新が依然として変更イベントをトリガーし、空のレプリケーション更新を生成します。*nsds5ReplicaStripAttrs*属性は、空のレプリケーションイベントでは送信できず、更新シーケンスから削除される属性の一覧を追加します。これには、*modifiersName*のような運用上の利便性が含まれます。

レプリケーションイベントが 空でない 場合は、ストライピングされた属性 が 複製されます。これらの属性は、イベントが空である場合にのみ更新から削除されます。

15.1.7.1. レプリケーションのキープアライブエントリー

マスターの属性を更新すると、マスターに対する変更シーケンス番号(CSN)が増えます。レプリケーショントポロジーでは、このサーバーは最初のコンシューマーに接続し、ローカル CSN をコンシューマーの CSN と比較できるようになりました。ローカル CSN の方が低い場合は、ローカルの changelog から更新内容を取得し、コンシューマーに複製します。一部レプリケーションを有効にしたレプリケーショントポロジーでは、これが問題になることがあります。たとえば、レプリケーションから除外された属性のみがマスター上で更新された場合、複製するための更新が見つからないため、コンシューマー上で CSN が更新されません。特定のシナリオでは、レプリケーションから除外されたマスターで属性のみが更新されると、サプライヤーの更新に不要な検索を行うと、他のサーバーが、必要に応じて後にデータを受け取る可能性があります。この問題を回避するために、Directory Server ではキープアライブエントリーを使用します。

マスター上の更新された属性がすべてレプリケーションから除外され、スキップされた更新の数が 100 を超えると、サプライヤーで *keepalivetimestamp* 属性が更新され、コンシューマーに複製されます。*keepalivetimestamp*属性はレプリケーションから除外されないため、キープアライブエントリーの更新は複製され、コンシューマー上の CSN が更新され、サプライヤー上のものと等しくなります。次回サプライヤーをコンシューマーに接続する際には、コンシューマーの CSN より新しい更新のみが検索されます。これにより、サプライヤーが送信する新規更新の検索に費やされた時間が短縮されます。

レプリケーションキープアライブエントリーはマスター上でオンデマンドで作成され、識別名(DN)にマスターのレプリカ ID が含まれます。キープアライブエントリーはそれぞれ特定のマスターに固有のものです。以下に例を示します。

```
dn: cn=repl keep alive 14,dc=example,dc=com
objectclass: top
objectclass: ldapsubentry
objectclass: extensibleObject
cn: repl keep alive 14
keepalivetimestamp: 20170227190346Z
```

キープアライブエントリーは、以下の状況で更新されます (更新前に存在しない場合には最初に作成されます)。

- 一部レプリカ合意が 100 を超える更新を省略し、レプリケーションセッションの終了前に更新を送信しません。
- マスターがコンシューマーを初期化すると、最初に独自のキープアライブエントリーを作成します。マスターでもあるコンシューマーは、別のコンシューマーも初期化しない限り、独自のキープアライブエントリーを作成しません。

15.2. コマンドラインでのレプリケーションの設定

レプリケーションは、サーバーに適切なレプリカおよび合意エントリーを作成して、コマンドラインで設定できます。このプロセスは、Directory Server コンソールからレプリケーションを設定するのと同じ順序に従います。

1. すべてのコンシューマー、ハブ、および複数マスターサプライヤー(「サプライヤーバインド DN エントリーの作成」)にサプライヤーバインド DN を作成します。
2. 対応するデータベースおよび接尾辞がレプリカのいずれかに存在しない場合は、これを作成します(「接尾辞の作成」)。
3. サプライヤーレプリカ(「コマンドラインでのサプライヤーの設定」)を設定します。
4. コンシューマーを設定します(「コマンドラインを使用したコンシューマーの設定」)。
5. レプリケーションをカスケードするためのハブ(「コマンドラインでのハブの設定」)を設定します。
6. レプリカ合意(「コマンドラインからのレプリカ合意の設定」)を作成します。カスケードレプリケーションの場合は、サプライヤーとハブ間の合意を作成し、ハブとコンシューマー間の合意を作成します。マルチマスターの場合は、すべてのサプライヤーとコンシューマー間の合意を作成します。
7. 最後に、レプリカ合意の作成時にコンシューマーが初期化されていない場合は、すべてのコンシューマー(「コマンドラインからのコンシューマーオンラインの初期化」)を初期化します。

15.2.1. コマンドラインでのサプライヤーの設定

サプライヤーレプリカのセットアップには、2 つの手順があります。まず、changelog を有効にする必要があります。これにより、サプライヤーは Directory Server への変更を追跡できます。次に、サプライヤーレプリカが作成されます。

1. サプライヤーサーバーで **ldapmodify** を使用して changelog エントリーを作成します。

例15.1 Changelog エントリーの例

```
# ldapmodify -D "cn=Directory Manager" -W -x -h supplier1.example.com -v -a

dn: cn=changelog5,cn=config
changetype: add
objectclass: top
objectclass: extensibleObject
cn: changelog5
nsslapd-changelogdir: /var/lib/dirsrv/slapd-instance/changelogdb
nsslapd-changelogmaxage: 10d
```

changelog には 2 つの重要な属性があります。

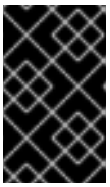
- ***nsslapd-changelogdir*** changelog が保持されるディレクトリーを設定します。
- ***nsslapd-changelogmaxage*** changelog が保持する期間を設定します。changelog は非常に大きくなる可能性があるため、これは、サーバーのパフォーマンスに影響を与え、ディスク領域を占有しないように changelog をトリミングするのに役立ちます。このパラメーターが設定されていない場合、デフォルトは持続するように changelog になります。

changelog エントリー属性は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス』 に記載されています。

2. サプライヤーレプリカを作成します。

例15.2 サプライヤーレプリカエントリーの例

```
# ldapmodify -D "cn=Directory Manager" -W -x -h supplier1.example.com -v -a

dn: cn=replica,cn=dc\=example\,dc\=com,cn=mapping tree,cn=config
changetype: add
objectclass: top
objectclass: nsds5replica
objectclass: extensibleObject
cn: replica
nsds5replicaroot: dc=example,dc=com
nsds5replicaid: 7
nsds5replicatype: 3
nsds5flags: 1
nsds5ReplicaPurgeDelay: 604800
nsds5ReplicaBindDN: cn=replication manager,cn=config
```



重要

以下の例に示すように、レプリカエントリーの *cn* パラメーターを replica に設定する必要があります。Directory Server は、パラメーターを異なる値に設定するとエントリーを無視します。

changelog エントリー属性は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス』 に記載されています。

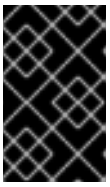
レプリケーション設定に含まれるすべてのサプライヤーを作成したら、レプリカ合意の作成を開始します。

15.2.2. コマンドラインを使用したコンシューマーの設定

コマンドラインを使用してコンシューマーを設定するには、コンシューマーホストで以下の設定が必要です。

1. レプリカエントリーを作成します。

```
# ldapadd -D "cn=Directory Manager" -W -p 389 -h consumer.example.com -x

dn: cn=replica,cn=dc\=example\,dc\=com,cn=mapping tree,cn=config
objectclass: top
objectclass: nsds5replica
objectclass: extensibleObject
cn: replica
nsds5replicaroot: dc=example,dc=com
nsds5replicaid: 65535
nsds5replicatype: 2
nsds5ReplicaBindDN: cn=replication manager,cn=config
nsds5flags: 0
```



重要

以下の例に示すように、レプリカエントリーの *cn* パラメーターを replica に設定する必要があります。Directory Server は、パラメーターを異なる値に設定するとエントリーを無視します。

このエントリーは、レプリケーションに参加しているようにデータベースと接尾辞を特定し、データベースがレプリカの種類を設定します。

2. ***nsslapd-referral*** パラメーターをサプライヤーサーバーの LDAP URLに、***nsslapd-state*** を更新時に参照 するように設定します。以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h consumer.example.com -x
dn: cn=dc\=example\,dc\=com,cn=mapping tree,cn=config
changetype: modify
replace: nsslapd-referral
nsslapd-referral: ldap://supplier.example.com:389/dc\=example\,dc\=com
-
replace: nsslapd-state
nsslapd-state: referral on update
```

この例で使用される属性の詳細は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス の該当するセクションを参照してください』。

15.2.3. コマンドラインでのハブの設定

ハブは、サプライヤーから更新を受け取り、他のコンシューマーに渡す中間読み取り専用レプリカです。これらは、「カスケードレプリケーション」 で説明されているカスケードレプリケーションのシナリオの一部です。ハブの作成には、最初に changelog データベースを作成します。これは、ハブがサプライヤーによって送信された変更の記録を保持し、2 番目にハブレプリカを設定します。

1. hub1.example.com などのハブサーバー で、ldapmodify を使用して changelog エントリーを作成します。

```
# ldapmodify -D "cn=Directory Manager" -W -x -h hub1.example.com -v -a

dn: cn=changelog5,cn=config
changetype: add
objectclass: top
objectclass: extensibleObject
cn: changelog5
nsslapd-changelogdir: /var/lib/dirsrv/slapd-instance/changelogdb
```

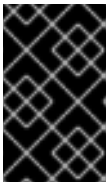
changelog には重要な属性が 1 つあります。***nsslapd-changelogdir*** は、changelog が保持されるディレクトリーを設定します。

changelog エントリー属性は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス』 に記載されています。

2. ハブホストで、レプリカエントリーを作成します。この ldapmodify コマンドは、dc=example ,dc= com サブツリーの hub1.example.com ホストに新しいハブレプリカを作成します。

```
# ldapmodify -D "cn=Directory Manager" -W -x -h hub1.example.com -v -a
```

```
dn: cn=replica,cn=dc\=example\,dc\=com,cn=mapping tree,cn=config
changetype: add
objectclass: top
objectclass: nsds5replica
objectclass: extensibleObject
cn: replica
nsds5replicaid: 65535
nsds5replicaroot: dc=example,dc=com
nsds5replicatype: 2
nsds5ReplicaPurgeDelay: 604800
nsds5ReplicaBindDN: cn=replication manager,cn=config
nsds5flags: 1
```



重要

以下の例に示すように、レプリカエントリーの *cn* パラメーターを replica に設定する必要があります。Directory Server は、パラメーターを異なる値に設定するとエントリーを無視します。

このエントリーは、レプリケーションに参加しているようにデータベースと接尾辞を特定し、データベースがレプリカの種類を設定します。

changelog エントリー属性は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス』 に記載されています。

15.2.4. コマンドラインからのレプリカ合意の設定

レプリカ合意を設定する場合は、まずすべてのサプライヤー間で設定してから、サプライヤーとハブの間で、最後にハブとコンシューマーの間を設定します。

レプリカ合意は、8 つの個別の属性を定義する必要があります。

- コンシューマーホスト(*nsds5replicahost*)およびポート(*nsds5replicaport*)
- コンシューマーにバインドするために使用するサプライヤーの DN(*nsds5ReplicaBindDN*)。
- サプライヤーがバインドする方法(*nsds5replicabindmethod*)。
- バインドメソッドおよび指定された DN に必要な認証情報(*nsDS5ReplicaCredentials*)。
- 複製されるサブツリー(*nsds5replicaroot*)。
- レプリケーションスケジュール(*nsds5replicaupdateschedule*)
- 複製されない属性 （nsds5replicatedattributelist および *nsDS5ReplicatedAttributeListTotal*）。

ldapmodify を使用して、更新するすべてのコンシューマーのすべてのサプライヤーにレプリカ合意を追加します。以下に例を示します。

例15.3 レプリカ合意エントリーの例

```
dn: cn=ExampleAgreement,cn=replica,cn=dc\=example\,dc\=com,cn=mapping
tree,cn=config
objectclass: top
```

```
objectclass: nsds5ReplicationAgreement
cn: ExampleAgreement
nsds5replicahost: consumer1
nsds5replicaport: 389
nsds5ReplicaBindDN: cn=replication manager,cn=config
nsds5replicabindmethod: SIMPLE
nsds5replicaroot: dc=example,dc=com
description: agreement between supplier1 and consumer1
nsds5replicaupdateschedule: 0000-0500 1
nsds5replicatedattributelist: (objectclass=*) $ EXCLUDE authorityRevocationList
accountUnlockTime memberof
nsDS5ReplicatedAttributeListTotal: (objectclass=*) $ EXCLUDE accountUnlockTime
nsds5replicacredentials: secret
```

例で使用するパラメーターの説明と、cn=*agreement_name,cn=replica,cn=suffix_ DN*,cn=mapping tree,cn=config エントリーで設定できるパラメーターの説明は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス を参照してください。』

すべてのレプリカ合意を作成したら、コンシューマーの初期化を開始します。

15.2.4.1. 証明書ベースの認証を使用するようにレプリケーションパートナーの設定

レプリケーションパートナーに対するバインド DN およびパスワードを使用する代わりに、証明書ベースの認証を使用できます。

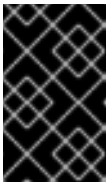
以下の手順では、レプリケーショントポロジーに server2.example.com という名前の新規サーバーを追加する方法を説明します。また、証明書ベースの認証を使用して、新規ホストと既存の server1.example.com との間でレプリカ合意を設定する方法を説明します。

1. 両方のホストで、証明書ベースの認証を設定します。詳細は「証明書ベースの認証の設定」を参照してください。
2. server1.example.com ホスト上で:
   a. 両方のサーバー (cn=server1,example,dc=com や cn=server2,dc=example,dc=com など) にアカウントを作成し、クライアント証明書を対応するアカウントに追加します。詳細は、次を参照してください。
      - 「ldapadd を使用したエントリーの追加」
      - 「ユーザーへの証明書の追加」

      両方のサーバーは、後でこれらのアカウントと証明書を使用して、相互にレプリケーション接続を確立するときに認証を行います。
   b. cn=repl_server,ou=Groups,dc=example,dc=com などのグループを作成し、両方のサーバーアカウントを追加します。「コマンドラインでのグループの作成」 を参照してください。
   c. レプリカエントリーを作成し、*nsds5ReplicaBindDNGroup* 属性を直前の手順で作成されたグループの DN に設定します。

      ```
      # ldapmodify -D "cn=Directory Manager" -W -p 636 -h server1.example.com -x

      dn: cn=replica,cn=dc\=example\,dc\=com,cn=mapping tree,cn=config
      ```

```
changetype: add
objectclass: top
objectclass: nsds5replica
objectclass: extensibleObject
cn: replica
nsds5replicaroot: dc=example,dc=com
nsds5replicaid: 7
nsds5replicatype: 3
nsds5flags: 1
nsds5ReplicaPurgeDelay: 604800
nsds5replicabinddngroup: cn=repl_server,ou=Groups,dc=example,dc=com
nsDS5ReplicaBindDNGroupCheckInterval: 0
```



重要

以下の例に示すように、レプリカエントリーの *cn* パラメーターを **replica** に設定する必要があります。Directory Server は、パラメーターを異なる値に設定するとエントリーを無視します。

3. 新しいサーバーを初期化します。

   a. **server2.example.com** で、**cn=Replication Manager,cn=config** などの一時的なレプリケーションマネージャーアカウントを作成します。「サプライヤーバインド DN エントリーの作成」 を参照してください。

   b. **server1.example.com** で、認証用に直前の手順でアカウントを使用する一時的なレプリカ合意を作成します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 636 -h server1.example.com -x

dn: cn=temporary_agreement,cn=replica,cn=dc\=example\,dc\=com,cn=mapping
tree,cn=config
objectclass: top
objectclass: nsds5ReplicationAgreement
cn: temporary_agreement
nsds5replicahost: server2.example.com
nsds5replicaport: 636
nsds5replicabindmethod: SIMPLE
nsds5ReplicaBindDN: cn=Replication Manager,cn=config
nsds5replicacredentials: password_of_replication_manager_account
nsds5replicaroot: dc=example,dc=com
description: Temporary agreement between server1 and server2
nsds5replicaupdateschedule: 0000-0500 1
nsds5replicatedattributelist: (objectclass=*) $ EXCLUDE authorityRevocationList
accountUnlockTime memberof
nsDS5ReplicatedAttributeListTotal: (objectclass=*) $ EXCLUDE
accountUnlockTime
nsds5BeginReplicaRefresh: start
```

   この合意は、以前に作成したレプリケーションマネージャーアカウントを使用してデータベースを初期化します。この初期化前に、**server2.example.com** のデータベースが空で、関連する証明書を持つアカウントは存在しません。したがって、データベースの初期化前に、証明書を使用してレプリケーションすることはできません。

4. 新しいサーバーが初期化された後

a. **server1.example.com から一時的なレプリカ合意を削除します。**

```
# ldapdelete -D "cn=Directory Manager" -W -p 636 -h server1.example.com -x
"cn=temporary_agreement,cn=replica,cn=dc\=example\,dc\=com,cn=mapping
tree,cn=config"
```

b. **server2.example.com から一時的なレプリケーションマネージャーアカウントを削除します。**

```
# ldapdelete -D "cn=Directory Manager" -W -p 636 -h server2.example.com -x
"cn=Replication Manager,cn=config"
```

5. **証明書ベースの認証を使用する両サーバーでレプリカ合意を作成します。**

a. **server1.example.com 上で:**

```
# ldapmodify -D "cn=Directory Manager" -W -p 636 -h server1.example.com -x

dn: cn=example_agreement,cn=replica,cn=dc\=example\,dc\=com,cn=mapping
tree,cn=config
objectclass: top
objectclass: nsds5ReplicationAgreement
cn: example_agreement
nsds5replicahost: server2.example.com
nsds5replicaport: 636
nsds5replicabindmethod: SSLCLIENTAUTH
nsds5replicaroot: dc=example,dc=com
description: Agreement between server1 and server2
nsds5replicaupdateschedule: 0000-0500 1
nsds5replicatedattributelist: (objectclass=*) $ EXCLUDE authorityRevocationList
accountUnlockTime memberof
nsDS5ReplicatedAttributeListTotal: (objectclass=*) $ EXCLUDE
accountUnlockTime
nsDS5ReplicaTransportInfo: SSL
```

b. **server2.example.com 上で:**

```
# ldapmodify -D "cn=Directory Manager" -W -p 636 -h server2.example.com -x

dn: cn=example_agreement,cn=replica,cn=dc\=example\,dc\=com,cn=mapping
tree,cn=config
objectclass: top
objectclass: nsds5ReplicationAgreement
cn: example_agreement
nsds5replicahost: server1.example.com
nsds5replicaport: 636
nsds5replicabindmethod: SSLCLIENTAUTH
nsds5replicaroot: dc=example,dc=com
description: Agreement between server2 and server1
nsds5replicaupdateschedule: 0000-0500 1
nsds5replicatedattributelist: (objectclass=*) $ EXCLUDE authorityRevocationList
accountUnlockTime memberof
```

```
nsDS5ReplicatedAttributeListTotal: (objectclass=*) $ EXCLUDE
accountUnlockTime
nsDS5ReplicaTransportInfo: SSL
```

6. レプリケーションが正しく機能していることを確認するには、レプリカ合意に *nsds5replicaLastUpdateStatus* 属性を表示します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 636 -h server1.example.com -b
"cn=example_agreement,cn=replica,cn=dc\=example\,dc\=com,cn=mapping
tree,cn=config" nsds5replicaLastUpdateStatus
```

可能なステータスの詳細は、Red『Hat Directory Server の設定、コマンド、およびファイルリファレンス の付録 『レプリカ合意の状況』 を参照してください』。

15.2.5. コマンドラインからのコンシューマーオンラインの初期化

nsds5replicarefresh 属性をレプリカ合意エントリーに追加することで、コマンドラインからオンライン初期化を開始できます。レプリカ合意の作成時に属性が含まれる場合、初期化が開始されます。後で追加して、コンシューマーをいつでも初期化できます。この属性はデフォルトでは存在しないため、コンシューマーの初期化が完了すると自動的に削除されます。

1. コンシューマーを初期化するサプライヤーサーバーでレプリカ合意の DN を検索します。以下に例を示します。

```
# ldapsearch -x -h supplier1.example.com -p 389 -D "cn=Directory Manager" -W -s sub
-b cn=config "(objectclass=nsds5ReplicationAgreement)"
```

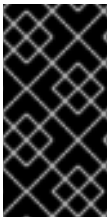
このコマンドは、サプライヤーに設定されたすべてのレプリカ合意を LDIF 形式で返します。初期化されるコンシューマーとのレプリカ合意の DN を取得します。これは、編集されるレプリカ合意です。

2. レプリカ合意を編集し、*nsds5BeginReplicaRefresh* 属性を追加します。

```
# ldapmodify -D "cn=Directory Manager" -W -x -h supplier1.example.com

dn: cn=ExampleAgreement,cn=replica,cn=dc\=example\,dc\=com,cn=mapping
tree,cn=config
changetype: modify
replace: nsds5BeginReplicaRefresh
nsds5BeginReplicaRefresh: start
```

ldapmodify は入力を要求しません。LDIF ステートメントに入力するだけで、LDIF ステートメントが完了すると 2 回到達されます。Ctrl+C を押して ldapmodify ユーティリティーを閉じます。

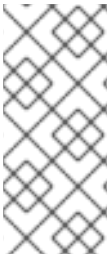
初期化が完了すると、*nsds5BeginReplicaRefresh* 属性はレプリカ合意エントリーから自動的に削除されます。



重要

マルチマスターレプリケーションの場合は、コンシューマーが 1 つのサプライヤーによって 1 度だけ 初期化されていることを確認します。レプリケーションのステータスを確認する際には、コンシューマーの初期化に使用された適切なサプライヤーでレプリカ合意のエントリーを確認してください。

コマンドラインからのコンシューマーの初期化についても、「コマンドラインを使用したコンシューマーオンラインの初期化」で説明されています。コンシューマーを手動で初期化する場合は、「コマンドラインを使用した手動コンシューマーの初期化」で説明されています。レプリケーションの監視属性は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス を参照してください』。



注記

大規模なデータベースでは、*nsslapd-idletimeout* 設定を十分な期間（または無制限の時間）に設定する必要があります。これにより、操作がタイムアウトする前にデータベース全体を初期化できるようにする必要があります。または、サプライヤーバインド DN エントリーの *nsIdleTimeout* 設定は、グローバル設定を変更しなくても、オンライン初期化操作を完了できるように設定することも可能です。

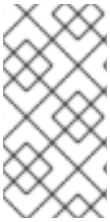
データの整合性を保持するには、適切なサプライヤーからコンシューマーデータベースを初期化します。正しいサプライヤーを決定することは、レプリケーション環境の組み合わせにおいてより困難になる場合がありますが、コンシューマーを手動で初期化した場合でも、以下の 4 つの点を考慮してください。

- コンシューマーを初期化するソースとして、1つのサプライヤー（ データマスター ）を使用します。
- レプリカ合意の作成時にデータマスター を再初期化 しないでください。たとえば、server2 が server1 からすでに初期化されている場合は、server2 から server1 を初期化しないでください。
- マルチマスターシナリオでは、1つのマスターからの設定内の他のすべてのマスターサーバーを初期化します。
- カスケードレプリケーションの場合は、サプライヤーからすべてのハブを初期化し、ハブからコンシューマーを初期化します。

15.3. レプリケーションシナリオ

- 「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」
- 「カスケードレプリケーション」

これらの基本ストラテジーをさまざまな方法で組み合わせ、最適なレプリケーション環境を作成できます。



注記

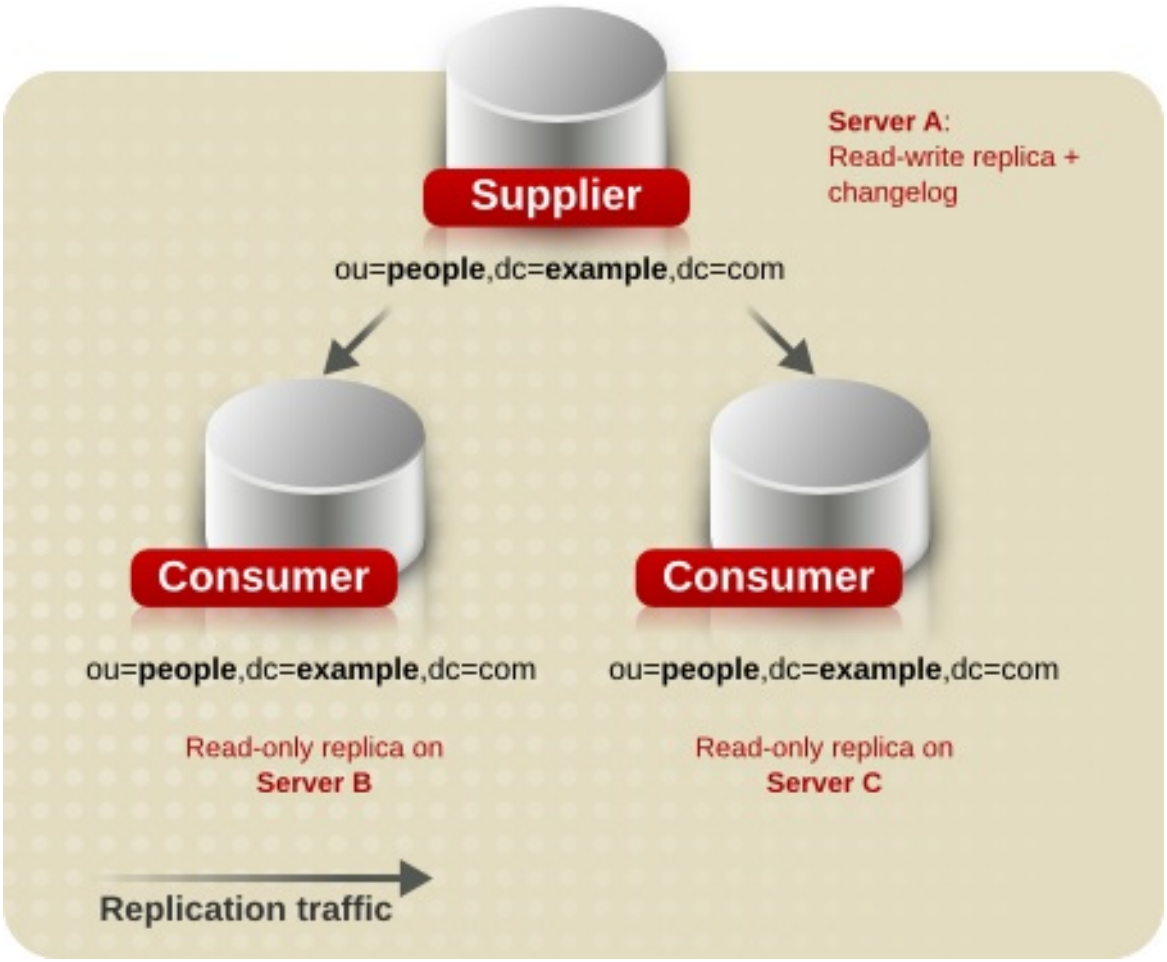
レプリケーションのシナリオが実装される場合でも、スキーマレプリケーションを考慮してください。競合解決ループを避けるために、Referential Integrity プラグインは、マルチマスターレプリケーション環境の 1 つのサプライヤーレプリカでのみ有効にする必要があります。プラグインはデフォルトで off です。

15.3.1. 単一マスターレプリケーション

最も単純なレプリケーションシナリオでは、ディレクトリーデータのマスターコピーが、サプライヤーサーバー と呼ばれる 1 台のサーバーの 1 つの読み取り/書き込みレプリカに保持されます。サプライヤーサーバーは、このレプリカの changelog も維持します。コンシューマーサーバー と呼ばれる別のサー

バーでは、複数の読み取り専用レプリカが存在する可能性があります。このようなシナリオは、単一マスター設定 と呼ばれます。図15.1「単一マスターレプリケーション」 は、単一マスターレプリケーションの例を示しています。

図15.1 単一マスターレプリケーション



この構成では、ou=people,dc=example,dc=com 接尾辞は多数の検索要求を受け取ります。そのため、負荷を分散するには、サーバー A にマスターされているツリーが、サーバー B とサーバー C にある 2 つの読み取り専用レプリカに複製されます。

シングルマスターレプリケーション環境の設定に関する詳細は、「単一マスターレプリケーションの設定」 を参照してください。

15.3.2. マルチマスターレプリケーション

Directory Server は、多数のサーバーで同じ接尾辞（データベース）をマスターできる複雑なレプリケーションシナリオもサポートします。この接尾辞は、各サーバーの読み取り/書き込みレプリカに保持されます。つまり、各サーバーは読み取り/書き込みレプリカの changelog を維持することを意味します。

Directory Server のマルチマスターレプリケーションは、20 マスター、無制限のハブサプライヤー、および無制限のコンシューマーサーバーをサポートします。各コンシューマーサーバーは読み取り専用レプリカを保持します。コンシューマーはすべてのサプライヤーから更新を受け取ることができます。コンシューマーには、コンシューマーが受信する更新リクエストを転送するために、すべてのサプライヤーに参照が定義されます。このようなシナリオは、マルチマスターレプリケーション と呼ばれます。

図15.2「マルチマスターレプリケーション(Two Masters)」 は、2 つのサプライヤーサーバーと 2 つのコンシューマーサーバーで構成されるマルチマスターレプリケーションのシナリオの例を示しています。

図15.2 マルチマスターレプリケーション(Two Masters)

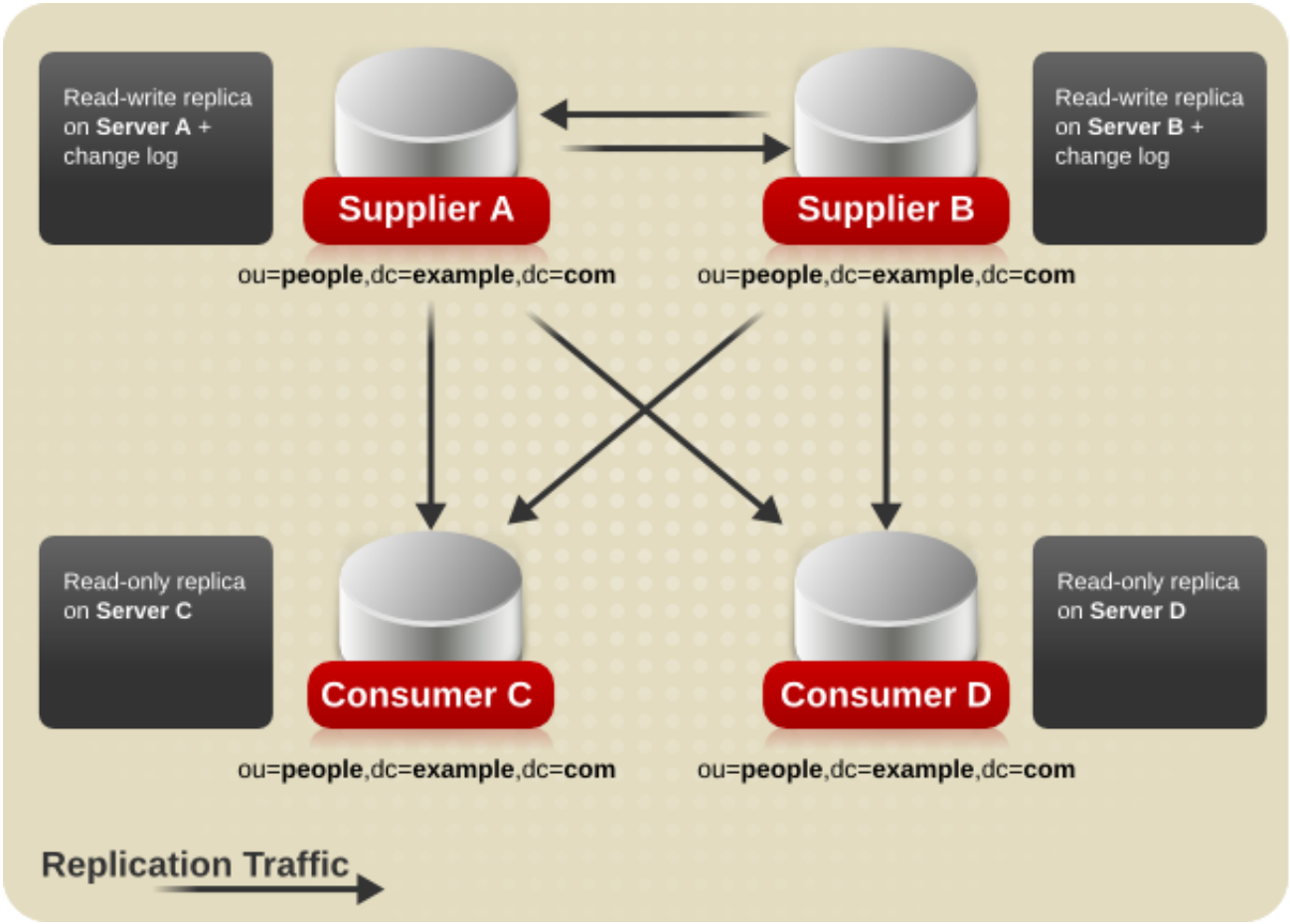
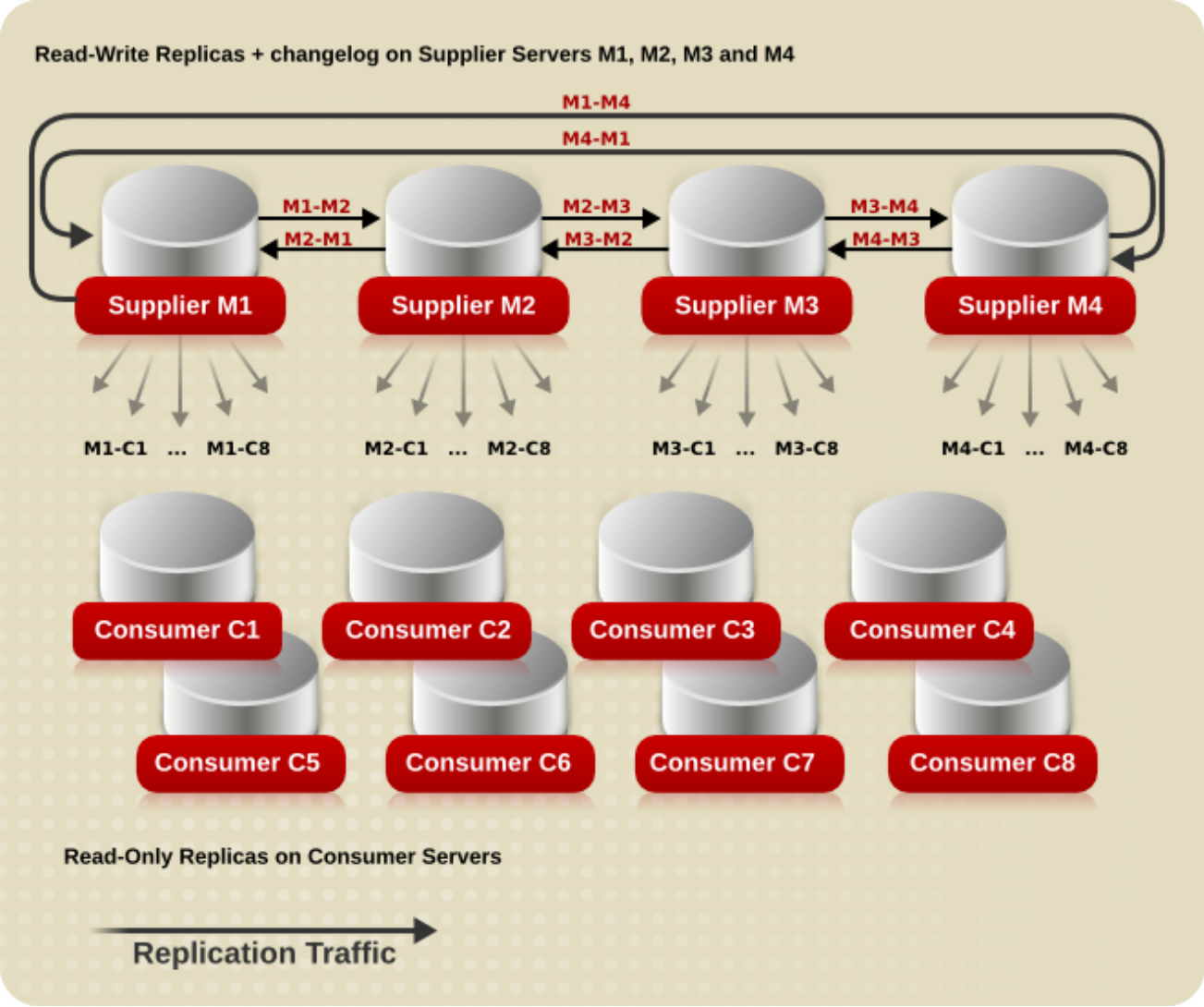


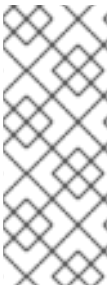
図15.3「マルチマスターレプリケーション（4 つのマスター）」 は、4 つのサプライヤーサーバーと 8 つのコンシューマーサーバーを使用するマルチマスターレプリケーションのシナリオの例を示しています。このサンプル設定では、各サプライヤーサーバーは、他の 2 つのサプライヤーサーバーとすべての 8 つのコンシューマーサーバーにデータをフィードするために 10 個のレプリカ合意で設定されます。（Directory Server には、マルチマスターレプリケーションに 20 マスターをいくつでも指定できます。）

図15.3 マルチマスターレプリケーション（4 つのマスター）



マルチマスター設定の利点は以下のとおりです。

- 1つのサプライヤーにアクセスできない場合に自動書き込みフェイルオーバー。
- 更新は、地理的に分散した環境のローカルサプライヤーで実行されます。



注記

レプリケーションが続行する速度は、以下によって異なります。

- ネットワークの速度。
- 送信および受信のレプリカ合意の数。最適なパフォーマンスを得るために、最大 8 アウトバウンドと 4 つの受信レプリカ合意を設定します。

マルチマスターレプリケーションを設定する手順は、「マルチマスターレプリケーションの設定」 を参照してください。

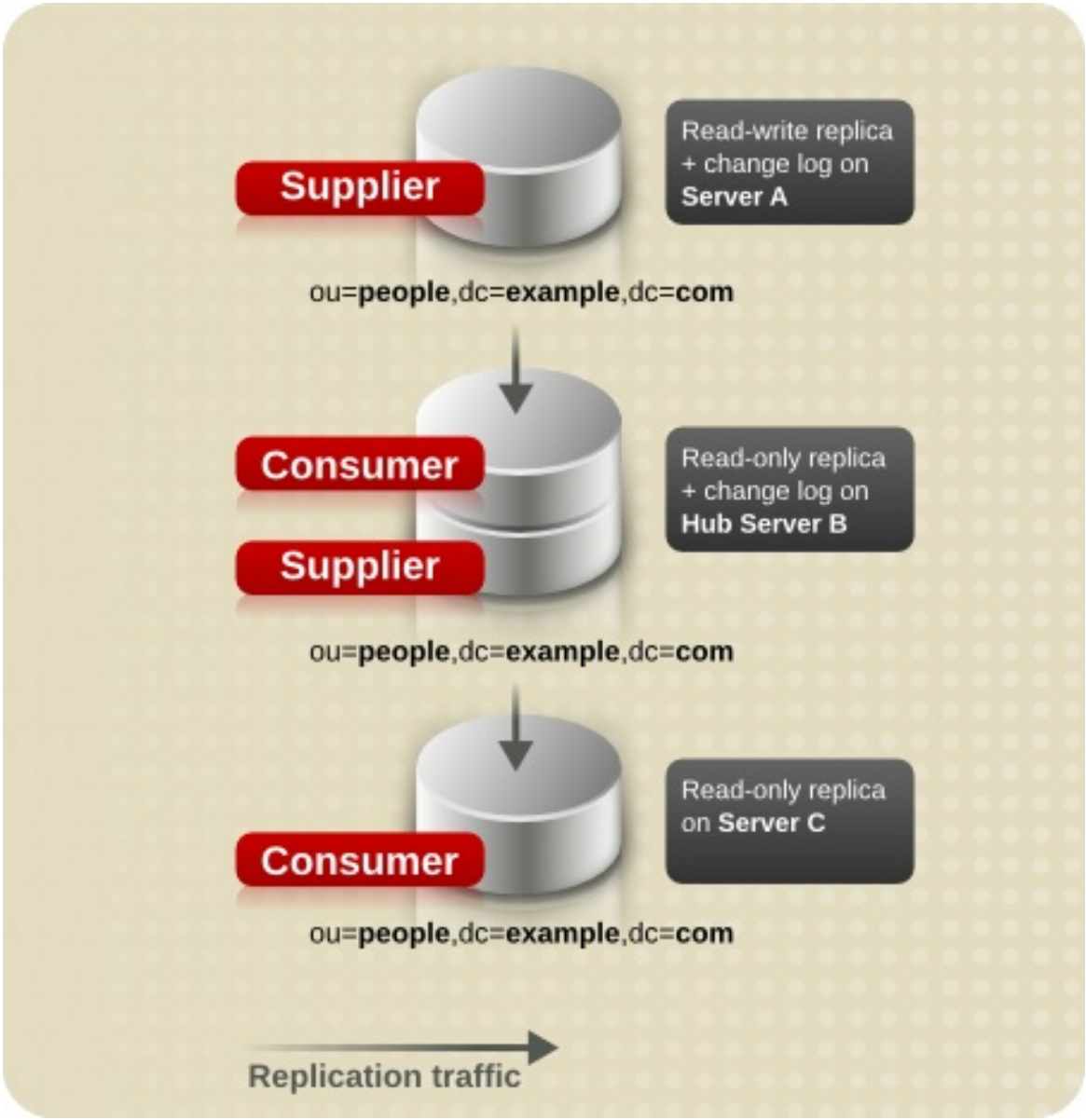
15.3.3. カスケードレプリケーション

カスケードレプリケーションのシナリオでは、ハブ となる 1台のサーバーがコンシューマーとサプライヤーの両方の役割を果たします。読み取り専用のレプリカを保持し、changelog を維持するため、デー

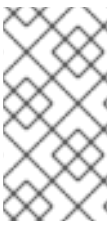
タのマスターコピーを保持するサプライヤーサーバーから更新を受け取って、その更新をコンシューマーに提供します。カスケードレプリケーションは、負荷の高いトラフィックのバランスを分散したり、地理的に分散した環境でマスターサーバーをローカルで維持したりする際に非常に便利です。

図15.4「カスケードレプリケーション」 は、複数のハブサーバーを使用してより複雑なシナリオを作成することはできますが、単純なカスケードレプリケーションシナリオの例を示しています。

図15.4 カスケードレプリケーション



カスケードレプリケーションの設定に関する詳細は、「カスケードレプリケーションの設定」 を参照してください。



注記

マルチマスターとカスケードレプリケーションを組み合わせることができます。たとえば、図15.2「マルチマスターレプリケーション(Two Masters)」 に記載されているマルチマスターレプリケーションでは、サーバー C およびサーバー D は、任意の数のコンシューマーサーバーに複製されるハブサーバーになります。

15.4. サプライヤーバインド DN エントリーの作成

レプリケーションのセットアップには、レプリケーションマネージャーまたはサプライヤーバインド DN エントリーと呼ばれるエントリーを作成することは、サプライヤーがコンシューマーサーバーにバインドしてレプリケーションの更新を実行するのに使用するエントリーを作成することです。

サプライヤーバインド DN は以下の基準を満たしている必要があります。

- 一意である必要があります。
- サプライヤーサーバー ではなく、コンシューマーサーバー（またはハブ）に作成する必要があります。
- コンシューマーサーバーの実際のエントリーに対応する必要があります。
- 別のサーバーから 更新を受け取る すべてのサーバーで作成する必要があります。
- セキュリティー上の理由から、複製されたデータベースの一部にすることはできません。
- これは、サプライヤーサーバーのレプリカ合意で定義する必要があります。
- 大規模なデータベースの初期化プロセスを完了させるには、アイドルタイムアウトの期間を十分な制限に設定する必要があります。*nsIdleTimeOut* 操作属性を使用すると、レプリケーションマネージャーエントリーがグローバル *nsslapd-idletimeout* 設定をオーバーライドできます。

たとえば、エントリー cn=Replication Manager,cn=config は、コンシューマーサーバーの cn=config ツリーの下に作成できます。これは、すべてのサプライヤーサーバーがコンシューマーにバインドし、レプリケーション操作を実行するために使用するサプライヤーバインド DN です。

注記

dse.ldif ファイルの cn=config エントリーの下に単純なエントリーを作成しないでください。シンプルな flat dse.ldif 設定ファイルの cn=cn=config エントリーは、通常のエントリーと同じ拡張性の高いデータベースに保存されません。その結果、多くのエントリーと、頻繁に更新される可能性のあるエントリーが cn=config に保存されると、パフォーマンスが低下します。ただし、Red Hat はパフォーマンス上の理由から、cn=config に単純なユーザーエントリーを保存しないことを推奨しますが、設定情報を一元化するため、cn=config に Directory Manager エントリーやレプリケーションマネージャー（サプライヤーのバインド DN）エントリーなどの特別なユーザーエントリーを保存すると便利です。

レプリカ合意でコンシューマーとして機能する各サーバーで、サプライヤーがコンシューマーにバインドするために使用する特別なエントリーを作成します。レプリカ合意で指定された認証方法に必要な属性でエントリーを作成してください。

1. Directory Server を停止します。サーバーが停止していない場合は、dse.ldif ファイルへの変更は保存されません。サーバーの停止に関する詳細は、「Directory Server インスタンスの起動および停止」 を参照してください。
2. dse.ldif ファイルに、cn=replication manager,cn=config などの新規エントリーを作成します。
3. *userPassword* 属性と値のペアを指定します。
4. 大規模なデータベースのレプリケーション初期化を完了できるように、レプリケーションユーザーに十分な時間制限を提供する *nsIdleTimeout* 期間を設定します。

5. パスワードの有効であるか、または有効になる場合は、レプリケーションマネージャーのエントリーでこれを無効にし、パスワードが期限切れになってレプリケーションが失敗するのを防ぎます。*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、20380119031407Z の値で *passwordExpirationTime* 属性を追加します。つまり、パスワードは期限切れになりません。

6. Directory Server を再起動します。サーバーの起動の詳細は、「Directory Server インスタンスの起動および停止」 を参照してください。

最後のエントリーは 例15.4「サプライヤーバインド DN エントリーの例」 のようになります。

例15.4 サプライヤーバインド DN エントリーの例

```
dn: cn=replication manager,cn=config
objectClass: top
objectClass: device
objectClass: simpleSecurityObject
cn: replication manager
userPassword: strong_password
nsIdleTimeout: 0
```

レプリカをコンシューマーとして設定する場合は、このエントリーの DN を使用してサプライヤーバインド DN を定義します。

15.5. 単一マスターレプリケーションの設定

図15.1「単一マスターレプリケーション」 に示されている設定などの単一マスターレプリケーションを設定するには、読み取り/書き込みレプリカを保持するサプライヤーサーバー A と 2 つのコンシューマーサーバー B とサーバー C の間で、それぞれ読み取り専用レプリカを保持するには、以下の 3 つの主要な手順があります。

- 「Supplier サーバーでの読み書きレプリカの設定」
- 「コンシューマーでの読み取り専用レプリカの設定」
- 「レプリカ合意の作成」

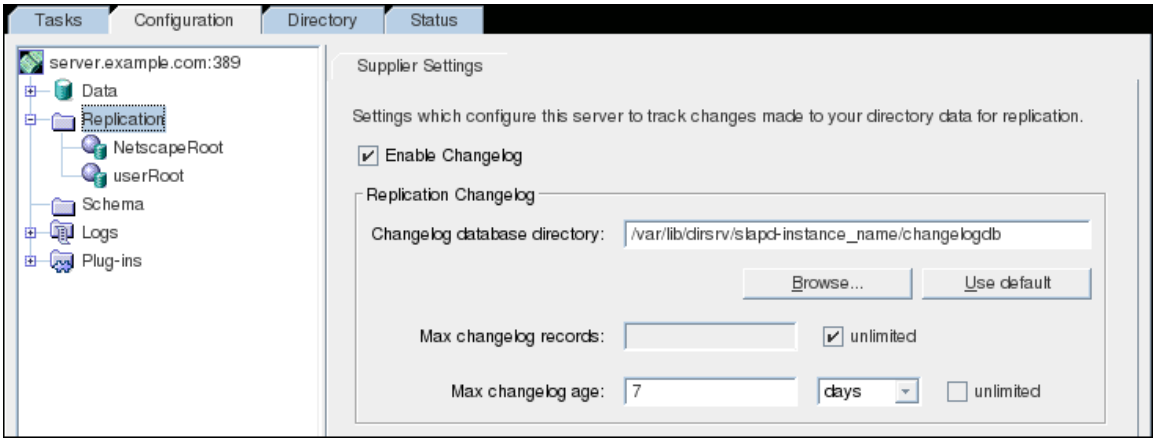
15.5.1. Supplier サーバーでの読み書きレプリカの設定

1. サーバーのサプライヤー設定を指定します。

   a. Directory Server コンソールで、**Configuration** タブを選択します。

   b. ナビゲーションツリーで、**Replication** フォルダーを選択します。

   c. ウィンドウの右側で、**Supplier Settings** タブを選択します。



d. **Enable Changelog** チェックボックスを選択します。

これにより、以前にグレーアウトされていたペインのフィールドがすべて有効になります。

e. **Use default** ボタンをクリックして changelog を指定するか、**Browse** ボタンをクリックしてファイルセレクターを表示します。

f. ログファイルの数と期間の changelog パラメーターを設定します。

異なる値を指定するには、無制限のチェックボックスをクリアします。

注記

Red Hat は、最大変更ログ 期間を 7 日に 設定することを推奨します。

g. **Save** をクリックします。

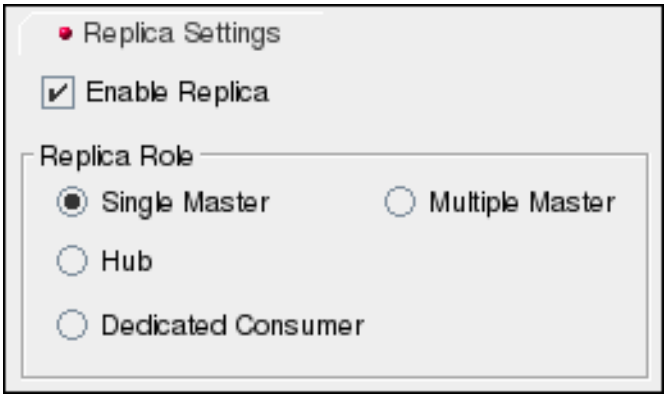
2. 読み取り/書き込みレプリカに必要なレプリケーション設定を指定します。

a. **Configuration** タブのナビゲーションツリーで、**Replication** ノードを展開し、複製するデータベースを強調表示します。

Replica Settings タブがウィンドウの右側で開きます。

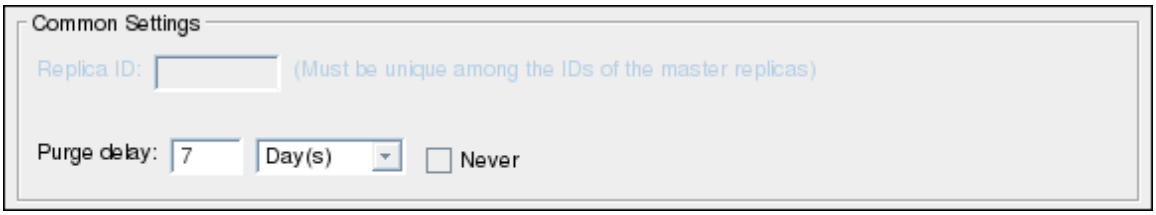
b. **Enable Replica** チェックボックスを選択します。

c. **Replica Role** セクションで、**Single Master** ラジオボタンを選択します。



d. **Common Settings** セクションで、**Replica ID** を指定します。レプリカ ID は1 から 65534 までの整数です。

指定の接尾辞のレプリカには、レプリカ ID を一意にする必要があります。このサーバーおよび他のサーバーで読み取り/書き込みレプリカに使用される他の ID とは異なります。



e. **Common Settings** セクションで、**Purge delay** フィールドでパージ遅延を指定します。

パージの遅延は、複製されたエントリーに対する状態情報を削除する頻度です。

f. **Save** をクリックします。

15.5.2. コンシューマーでの読み取り専用レプリカの設定

1. 読み取り専用レプリカのデータベースがない場合は、これを作成します。接尾辞の作成方法については、「接尾辞の作成」 を参照してください。

2. コンシューマーサーバーにサプライヤーバインド DN のエントリーを作成します（存在しない場合）。サプライヤーバインド DN は、サプライヤーがコンシューマーにバインドするために使用する特別なエントリーです。これは「サプライヤーバインド DN エントリーの作成」 で説明されています。

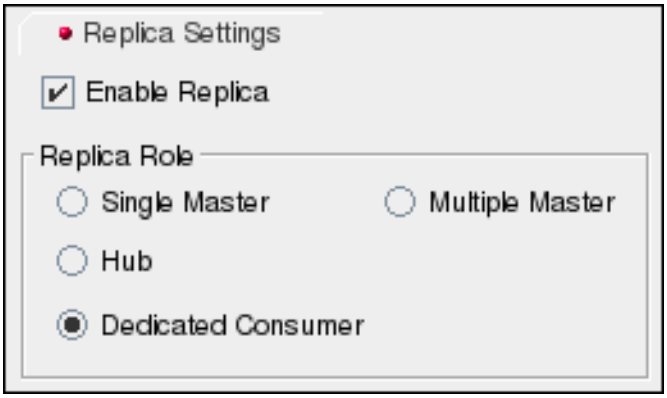
3. 読み取り専用レプリカに必要なレプリケーション設定を指定します。

a. Directory Server コンソールで、**Configuration** タブを選択します。

b. ナビゲーションツリーで **Replication** フォルダーを展開し、レプリカデータベースを選択します。

o=NetscapeRoot データベースを複製 する場合は、「管理サーバーのフェイルオーバー用の o=NetscapeRoot の複製」 を参照してください。

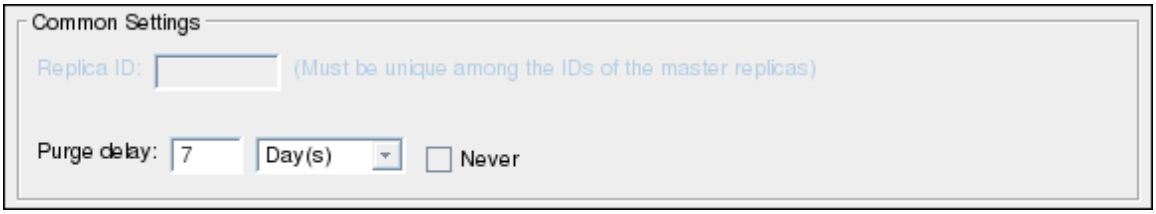
c. 選択したデータベースの **Replica Settings** タブで、**Enable Replica** チェックボックスを選択します。



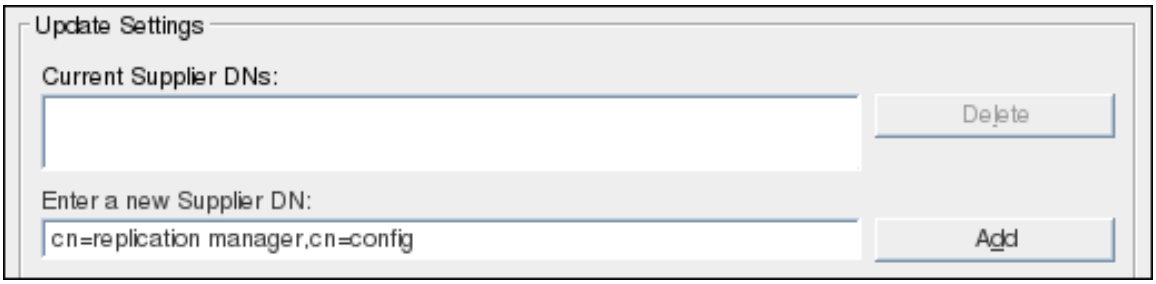
d. **Replica Role** セクションで、**Dedicated Consumer** ラジオボタンを選択します。

e. **Common Settings** セクションで、**Purge delay** フィールドでパージ遅延を指定します。

このオプションは、複製されたエントリーのステータス情報がパージされる頻度を示します。



f. **Update Settings** セクションで、サプライヤーがレプリカにバインドするために使用するバインド DN を指定します。**Enter a new Supplier DN** フィールドにサプライヤーバインド **DN** を入力し、**Add** をクリックします。サプライヤーバインド DN が現在のサプライヤー **DN** 一覧に表示されます。



サプライヤーバインド DN は、手順 2 で作成されたエントリーである必要があります。サプライヤーバインド DN はアクセス制御の対象ではないため、特権ユーザーです。

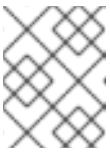
注記

コンシューマーごとに複数のサプライヤーバインド DN を指定できますが、レプリカ合意ごとに 1 つのサプライヤー DN のみがあります。

g. 更新を参照するサプライヤーサーバーの URL を指定します。

デフォルトでは、すべての更新は、ここで指定されたサプライヤーサーバーが最初に参照されます。ここでサプライヤーが設定されていない場合、更新は現在のレプリカを含むレプリカ合意を持つサプライヤーサーバーと呼ばれます。

自動参照は、クライアントが通常の接続上でバインドすることを前提としています。これには **ldap://**hostname:port 形式の URL があります。クライアントが TLS を使用してサプライヤーにバインドするには、このフィールドを使用して **ldaps://**hostname:port 形式の参照を指定します。**ldaps** の **s** はセキュアな接続を示します。



注記

ホスト名の代わりに IPv4 アドレスまたは IPv6 アドレスを使用することもできます。

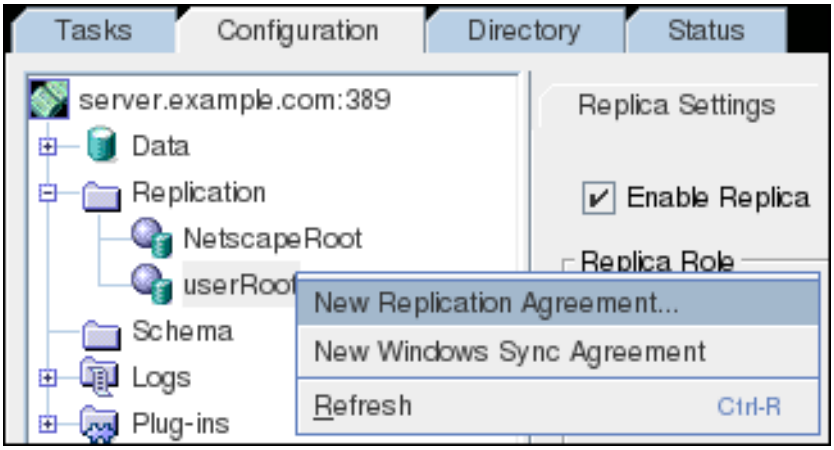
4. **Save** をクリックします。

レプリケーション設定のすべてのコンシューマーサーバーでこの手順を繰り返します。

15.5.3. レプリカ合意の作成

サプライヤーで、読み取り専用レプリカごとに 1 つのレプリカ合意を作成します。たとえば、図15.1「単一マスターレプリケーション」 で説明されているシナリオでは、サーバー A には 2 つのレプリカ合意があり、サーバー B の場合は 1 つ、サーバー C 用のレプリカ合意があります。

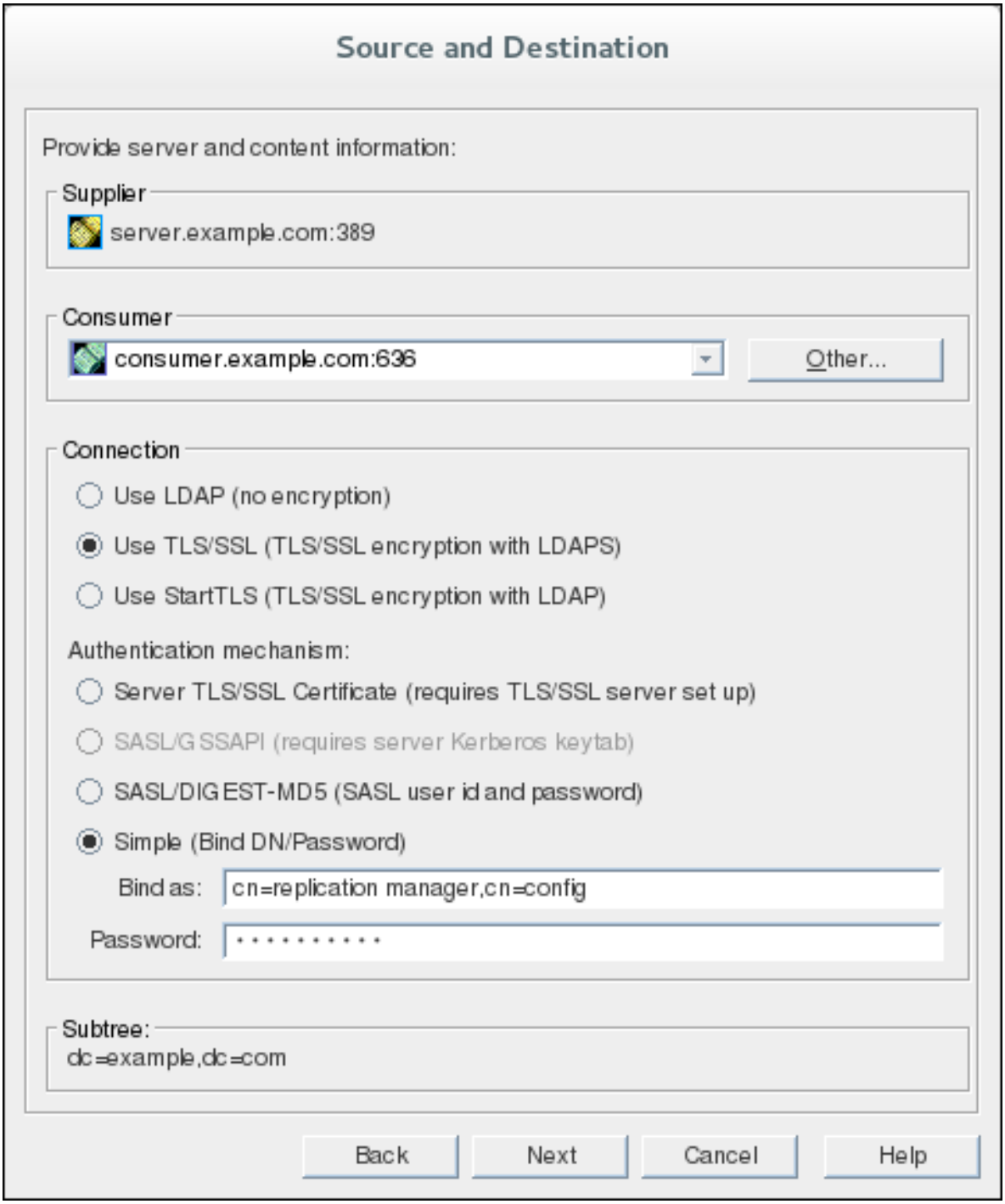
1. **Configuration** タブのナビゲーションツリーで、データベースを右クリックし、**New Replication Agreement** を選択します。



または、データベースを強調表示し、**Object** メニューから **New Replication Agreement** を選択して Replication Agreement Wizard を起動します。

2. 最初の画面で、レプリカ合意の名前および説明を入力し、**Next** を押します。

3. **Source and Destination** 画面で、コンシューマーの URL（hostname:port または IP_address:port ）と、コンシューマー上のサプライヤーバインド DN とパスワードを入力します。ターゲットサーバーが利用できない場合は、他のサーバーにアクセスして情報を手動で入力します。



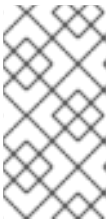
- 複数の Directory Server インスタンスが設定されていない場合、デフォルトでは、ドロップダウンメニューにはコンシューマーがありません。
- Directory Server インスタンスが TLS で実行されるように設定されている場合でも、一覧表示されるポートは TLS 以外のポートになります。このポート番号は、コンソールで Directory Server インスタンスを識別するためにのみ使用されます。これは、レプリケーションに使用される実際のポート番号またはプロトコルを指定しません。
- サーバーで TLS が有効になっている場合は、TLS クライアント認証に **Using encrypted SSL connection** ラジオボタンを選択できます。それ以外の場合は、サプライヤーバインド DN およびパスワードを入力します。

注記

属性の暗号化が有効な場合は、暗号化された属性を複製するセキュアな接続を使用する必要があります。

4. 接続タイプを選択します。以下の 3 つのオプションがあります。

    - LDAP を使用します。これにより、標準の暗号化されていない接続が設定されます。
    - TLS/SSL を使用します。これは、636 などのサーバーのセキュアな LDAPS ポートを介したセキュアな接続を使用します。この設定は TLS を使用するために必要です。
    - Start TLS を使用します。Start TLS を使用して、サーバーの標準ポートでセキュアな接続を確立します。

    注記

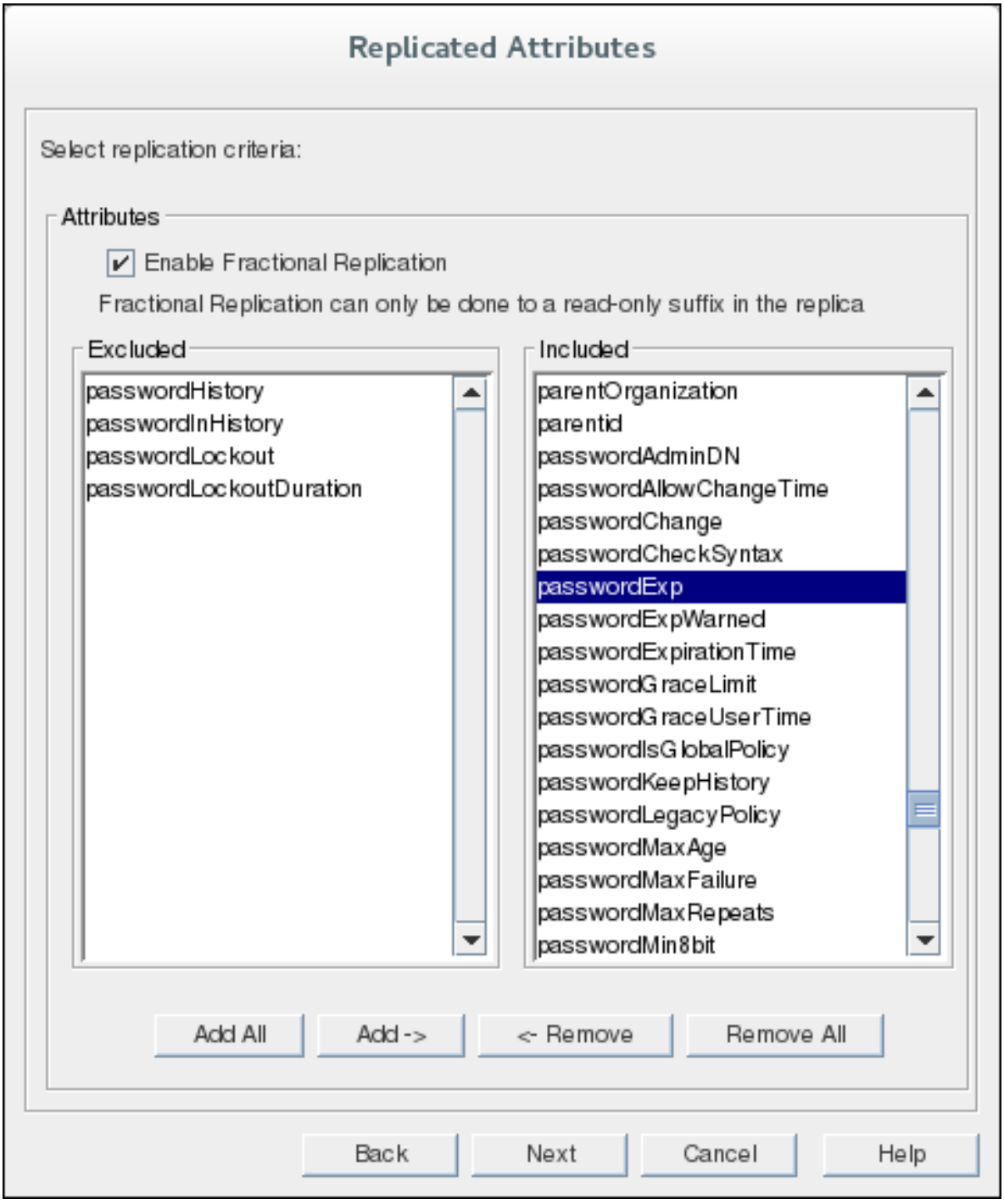
    シンプルなパスワード認証(「セキュアなバインドの要求」)にセキュアなバインドが必要な場合は、セキュアな接続で行われる場合を除き、レプリケーション操作は失敗します。セキュアな接続（TLS および Start TLS 接続または SASL 認証）の使用が推奨されます。

5. 適切な認証方法を選択し、必要な情報を提供します。これにより、サプライヤーがコンシューマーサーバーにバインドして更新を送信するために使用する情報を提供します。

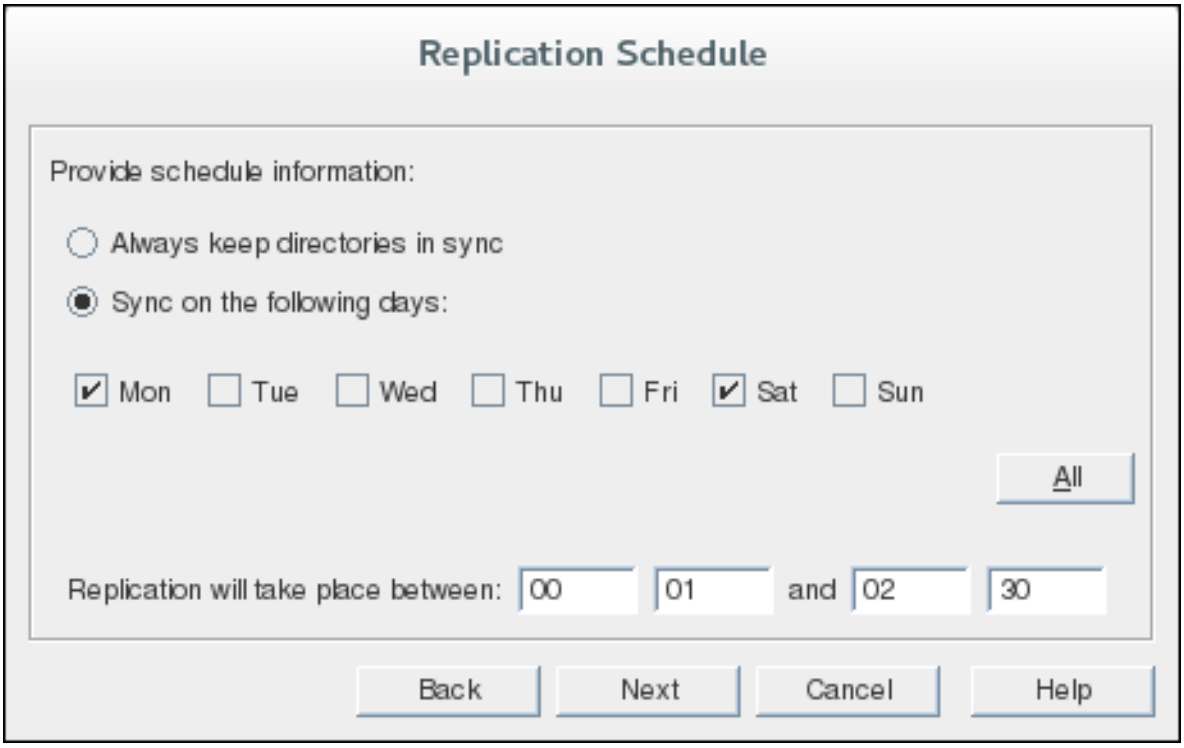
    - simple は、サーバーが、暗号化なしで標準ポートで接続することを意味します。必要な情報は、Replication Manager のバインド DN およびパスワード（コンシューマーサーバーに存在する必要がある）です。
    - サーバー TLS/SSL 証明書は、サプライヤーの TLS 証明書を使用して、コンシューマーサーバーに対して認証します。証明書は、証明書ベースの認証のサプライヤーにインストールされ、コンシューマーサーバーには、サプライヤーの証明書内のサブジェクト DN をその Replication Manager エントリーにマッピングできるように、証明書マッピングを設定する必要があります。

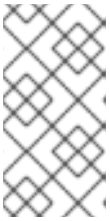
      TLS および証明書マッピングの設定については、「TLS の有効化」 を参照してください。
    - 簡易認証などの SASL/DIGEST-MD5。このセキュアでない方法で認証するには、バインド DN およびパスワードのみが必要になります。これは、標準または TLS 接続上で実行できます。
    - SASL/GSSAPI では、サプライヤーサーバーに Kerberos キータブ（「KDC サーバーおよびキータブの概要」にあるように）があり、コンシューマーサーバーでサプライヤーのプリンシパルを実際のレプリケーションマネージャーエントリーにマップするために SASL マッピングが必要です（「コンソールからの SASL アイデンティティーマッピングの設定」のように）。

6. 一部レプリケーションは、サーバー間でエントリーがレプリケートされるエントリー属性を制御します。デフォルトでは、すべての属性がレプリケートされます。コンシューマーに複製されない属性 を選択するには、Enable Fractional Replication チェックボックスを選択します。次に、右側の Included コラムの属性（または属性）を強調表示し、Remove をクリックします。レプリケートされない属性はすべて左側の Excluded 列に一覧表示されます。また、レプリカ合意が完了する概要にも表示されます。



7. **レプリケーションの実行時にスケジュールを設定します。デフォルトでは、レプリケーションは継続的に実行されます。**



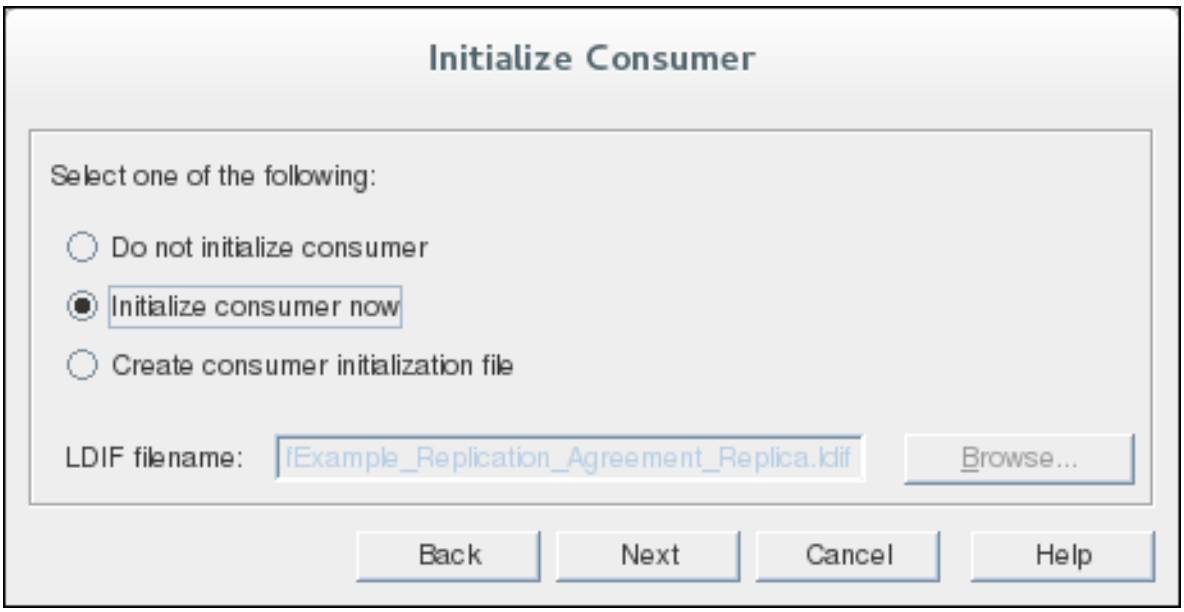


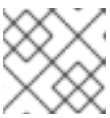
注記

レプリケーションスケジュールは、真夜中(0000)を越えません。そのため、0001 を開始し、同じ日に 2359 で終わるスケジュールを設定できますが、1 日の 2359 から開始したスケジュールを設定し、次の 部分で終了することはできません。

Next を押します。

8. **Initialize consumer now を選択して、レプリカ合意の完了後に初期化を開始し、Next をクリックします。**



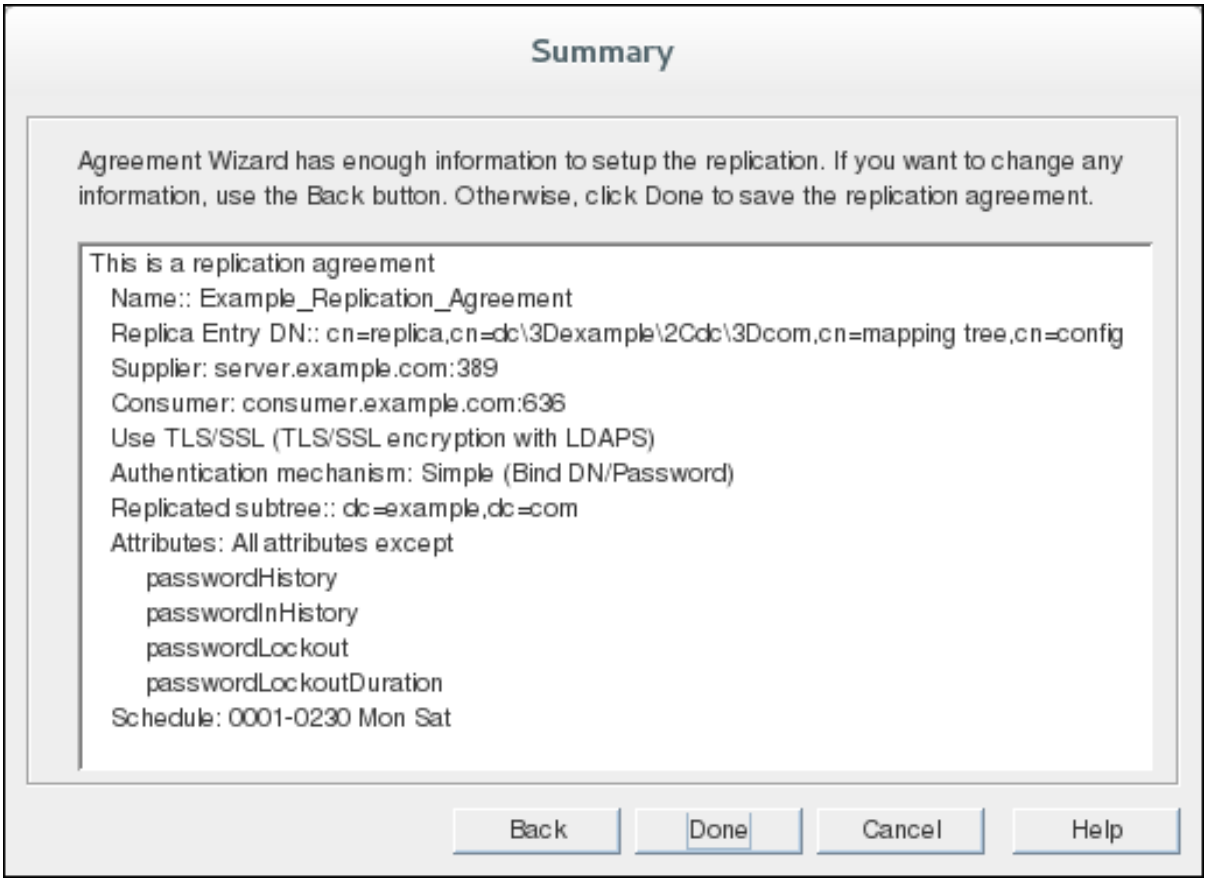


注記

レプリケーションは、コンシューマーが初期化されるまで 開始されません。

コンシューマーの初期化の詳細は、「コンシューマーの初期化」 を参照してください。

9. 最後の画面には dse.ldif ファイルに含まれるため、レプリカ合意の設定が表示されます。完了 を押して合意を保存します。



レプリカ合意が設定されている。

注記

レプリカ合意の作成後、LDAP および LDAPS 接続が異なるポートを使用するため、接続タイプ（TLS または非 TLS）を変更することはできません。接続タイプを変更するには、レプリカ合意を再作成します。

15.6. マルチマスターレプリケーションの設定

マルチマスター設定では、多くのサプライヤーは更新を受け入れ、相互に同期し、すべてのコンシューマーを更新できます。コンシューマーは、すべてのマスターに更新の参照を送信することができます。

Directory Server は 20 方向のマルチマスターレプリケーションをサポートします。つまり、単一のレプリケーションシナリオにおいて、最大 20 マスター（および無制限のハブサプライヤー）が存在する可能性があります。Directory Server は、無制限のコンシューマーを許可します。

マルチマスターレプリケーションを設定するには、最初にすべてのコンシューマーを設定し、その後にサプライヤーを設定し、最後にすべてのデータベースを初期化します。

- 「サプライヤーサーバーでの読み書きレプリカの設定」
- 「コンシューマーサーバーでの読み取り専用レプリカの設定」
- 「レプリカ合意の設定」

- 「マルチマスターレプリケーションにおけるコンシューマーの独占を防ぐ」

注記

レプリケーションまたはサプライヤーで実行中の 10 を超えるデータベースは、パフォーマンスが低下する可能性があります。この数のコンシューマーをサポートするには、サプライヤーとコンシューマー間のハブレプリカを導入します。「カスケードレプリケーションの設定」 を参照してください。

15.6.1. サプライヤーサーバーでの読み書きレプリカの設定

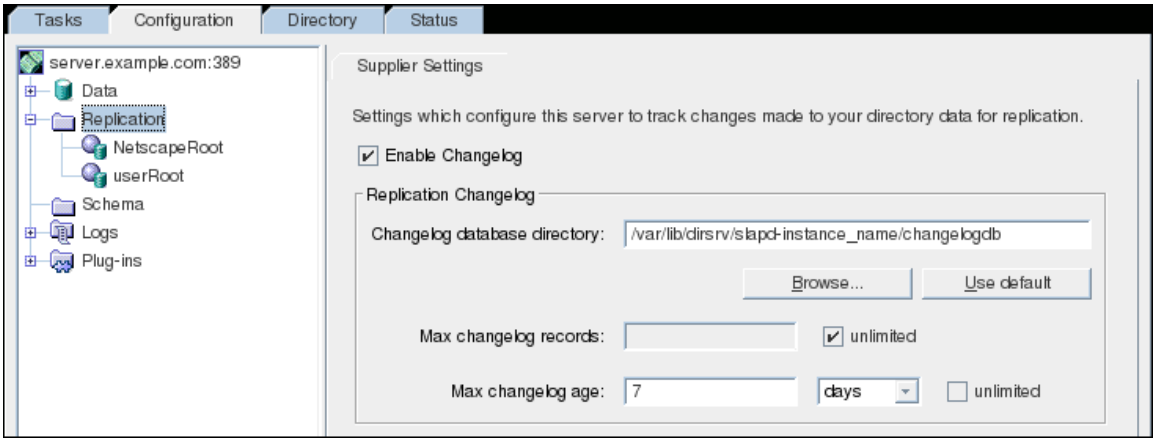
各サプライヤーサーバーを設定します。マルチマスターレプリケーション環境で他のサプライヤーを初期化するには、設定された最初のサプライヤーを使用する必要があります。

1. サーバーのサプライヤー設定を指定します。

   a. Directory Server コンソールで、**Configuration** タブを選択します。

   b. ナビゲーションツリーで、**Replication** フォルダーを選択します。

   c. ウィンドウの右側で、**Supplier Settings** タブを選択します。



   d. **Enable Changelog** チェックボックスを選択します。

      これにより、以前にグレーアウトされていたペインのフィールドがすべて有効になります。

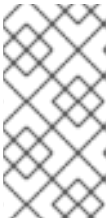
   e. **Use default** ボタンをクリックして changelog を指定するか、**Browse** ボタンをクリックしてファイルセレクターを表示します。

   f. ログファイルの数と期間の changelog パラメーターを設定します。

      異なる値を指定するには、無制限のチェックボックスをクリアします。

   g. **Save** をクリックします。

2. コンシューマーサーバーにサプライヤーバインド DN のエントリーを作成します（存在しない場合）。これは、他のサプライヤーとコンシューマーの関係と同様に、他のサプライヤーがこのサプライヤーにバインドするために使用する特別なエントリーです。これは「サプライヤーバインド DN エントリーの作成」 で説明されています。



注記

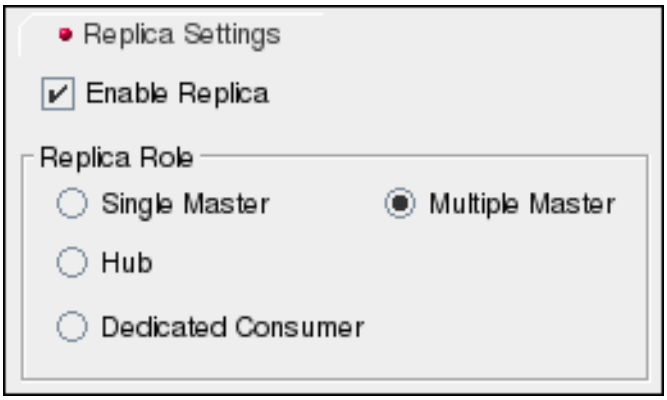
マルチマスターレプリケーションでは、サプライヤーがコンシューマーとサプライヤーの両方を他のサプライヤーサーバーに対して機能するため、サプライヤーサーバーとコンシューマーにこのサプライヤーバインド DN を作成する必要があります。

3. マルチマスターの読み取り/書き込みレプリカのレプリケーション設定を指定します。

   a. Directory Server コンソールで、**Configuration** タブを選択します。

   b. ナビゲーションツリーで、**Replication** フォルダーを展開し、レプリカデータベースを強調表示します。

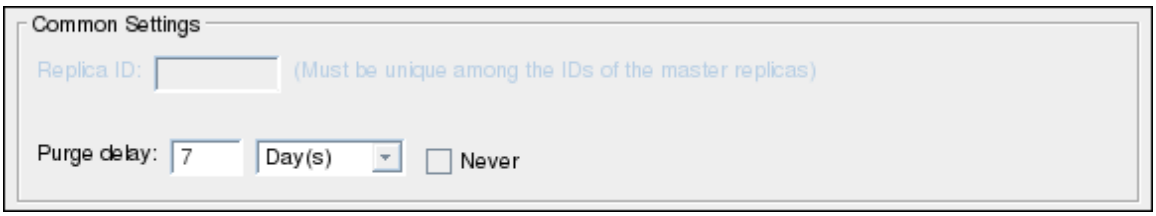
      ウィンドウの右側で、そのデータベースの **Replica Settings** タブが開きます。



   c. **Enable Replica** チェックボックスを選択します。

   d. **Replica Role** セクションで、**Multiple Master** ラジオボタンを選択します。

   e. **Common Settings** セクションで、**Replica ID** を指定します。レプリカ ID は1 から 65534 までの整数です。

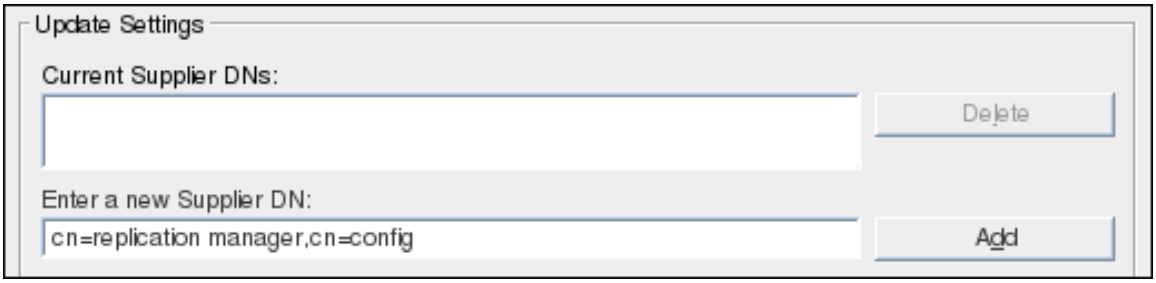


      指定の接尾辞のレプリカには、レプリカ ID を一意にする必要があります。このサーバーおよび他のサーバーで読み取り/書き込みレプリカに使用される他の ID とは異なります。

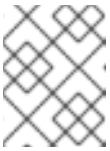
   f. **Common Settings** セクションで、**Purge delay** フィールドでパージ遅延を指定します。

      パージの遅延は、複製されたエントリーに対する状態情報を削除する頻度です。

   g. **Update Settings** セクションで、サプライヤーがレプリカにバインドするために使用するバインド DN を指定します。**Enter a new Supplier DN** フィールドにサプライヤーバインド DN を入力し、**Add** をクリックします。サプライヤーバインド DN が現在のサプライヤー DN 一覧に表示されます。



サプライヤーバインド DN は、手順 2 で作成されたエントリーである必要があります。サプライヤーバインド DN は、複製されたデータベースのアクセス制御の対象ではないため、特権ユーザーです。



注記

コンシューマーごとに複数のサプライヤーバインド DN を指定できますが、レプリカ合意ごとに 1 つのサプライヤー DN のみがあります。

h. マルチマスターレプリケーションセットの他のサプライヤーなど、更新を参照するサプライヤーサーバーの LDAP URL（ldap://hostname:port または ldap://IP_address:port） を指定します。サプライヤーサーバーの URL のみを指定します。

   クライアントが TLS を使用してバインドするには、**ldaps://** で始まる URL を指定します。

i. **Save** をクリックします。

15.6.2. コンシューマーサーバーでの読み取り専用レプリカの設定

まず、レプリカ合意 を作成する前に、すべてのコンシューマーを設定します。

1. 読み取り専用レプリカのデータベースがない場合は、これを作成します。接尾辞の作成方法については、「接尾辞の作成」 を参照してください。

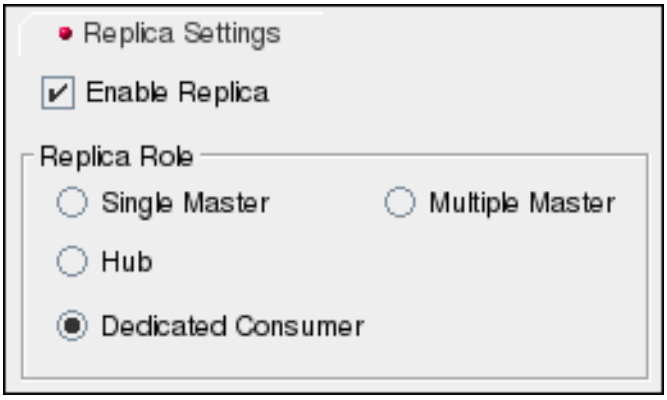
2. コンシューマーサーバーにサプライヤーバインド DN のエントリーを作成します（存在しない場合）。サプライヤーバインド DN は、サプライヤーがコンシューマーにバインドするために使用する特別なエントリーです。これは 「サプライヤーバインド DN エントリーの作成」 で説明されています。

3. 読み取り専用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 コンソールで、**Configuration** タブを選択します。

   b. ナビゲーションツリーで、**Replication** フォルダーを展開し、レプリカデータベースを強調表示します。

      ウィンドウの右側で、そのデータベースの **Replica Settings** タブが開きます。

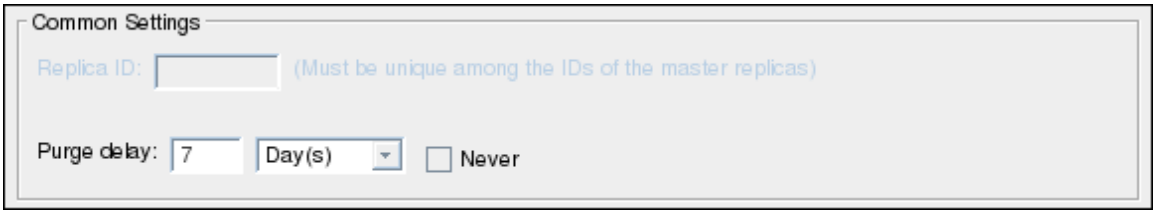


c. **Enable Replica** チェックボックスを選択します。

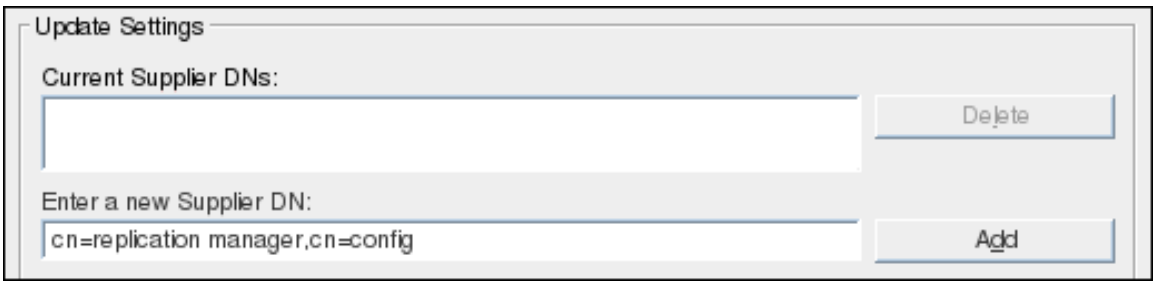
d. **Replica Role** セクションで、**Dedicated Consumer** ラジオボタンを選択します。

e. **Common Settings** セクションで、**Purge delay** フィールドでパージ遅延を指定します。

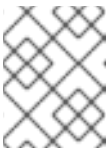
このオプションは、複製されたエントリーのステータス情報がパージされる頻度を示します。



f. **Update Settings** セクションで、サプライヤーがレプリカにバインドするために使用するバインド DN を指定します。**Enter a new Supplier DN** フィールドにサプライヤーバインド DN を入力し、**Add** をクリックします。サプライヤーバインド DN が現在のサプライヤー DN 一覧に表示されます。



サプライヤーバインド DN は、手順 2 で作成されたエントリーである必要があります。サプライヤーバインド DN は、複製されたデータベースのアクセス制御の対象ではないため、特権ユーザーです。



注記

コンシューマーごとに複数のサプライヤーバインド DN を指定できますが、レプリカ合意ごとに 1 つのサプライヤー DN のみがあります。

g. 更新を参照するサプライヤーサーバーの URL を指定します。

デフォルトでは、すべての更新は、ここで指定されたサプライヤーサーバーが最初に参照されます。ここでサプライヤーが設定されていない場合、更新は現在のレプリカを含むレプリカ合意を持つサプライヤーサーバーと呼ばれます。

自動参照は、クライアントが通常の接続上でバインドすることを前提としています。これには **ldap://hostname:port** 形式の URL があります。クライアントが TLS を使用してサプライヤーにバインドするには、このフィールドを使用して **ldaps://hostname:port** 形式の参照を指定します。**ldaps** の **s** はセキュアな接続を示します。

注記

ホスト名の代わりに IPv4 アドレスまたは IPv6 アドレスを使用することもできます。

4. **Save** をクリックします。

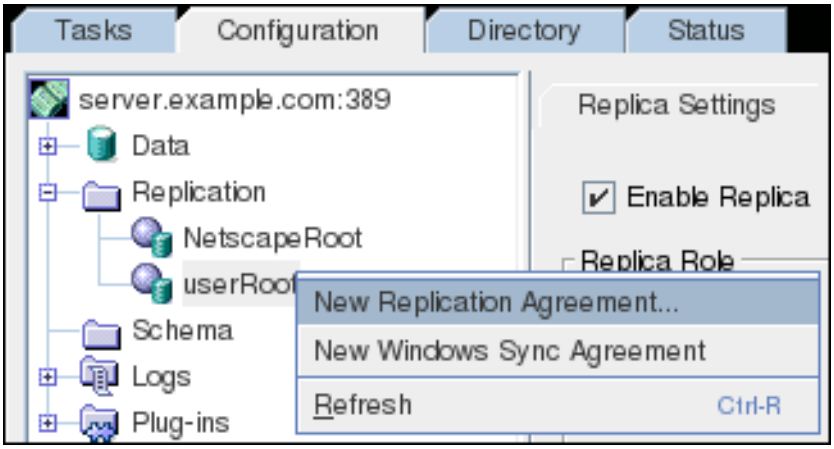
レプリケーション設定のすべてのコンシューマーサーバーでこの手順を繰り返します。

15.6.3. レプリカ合意の設定

注記

1. まず、1つのサプライヤー、他のマルチマスターレプリケーション間のデータマスターでレプリカ合意を設定し、他のすべてのサプライヤーを初期化します。
2. 次に、マルチマスターレプリケーションセットで他のすべてのサプライヤーに対してレプリカ合意を作成しますが、サプライヤーは再初期化 しません。
3. 次に、1つのデータマスターからすべてのコンシューマーに対してレプリカ合意を作成し、コンシューマーを初期化します。
4. 次に、他のすべてのサプライヤーに対してすべてのコンシューマーに対してレプリカ合意を作成しますが、コンシューマーも再初期化 しません。

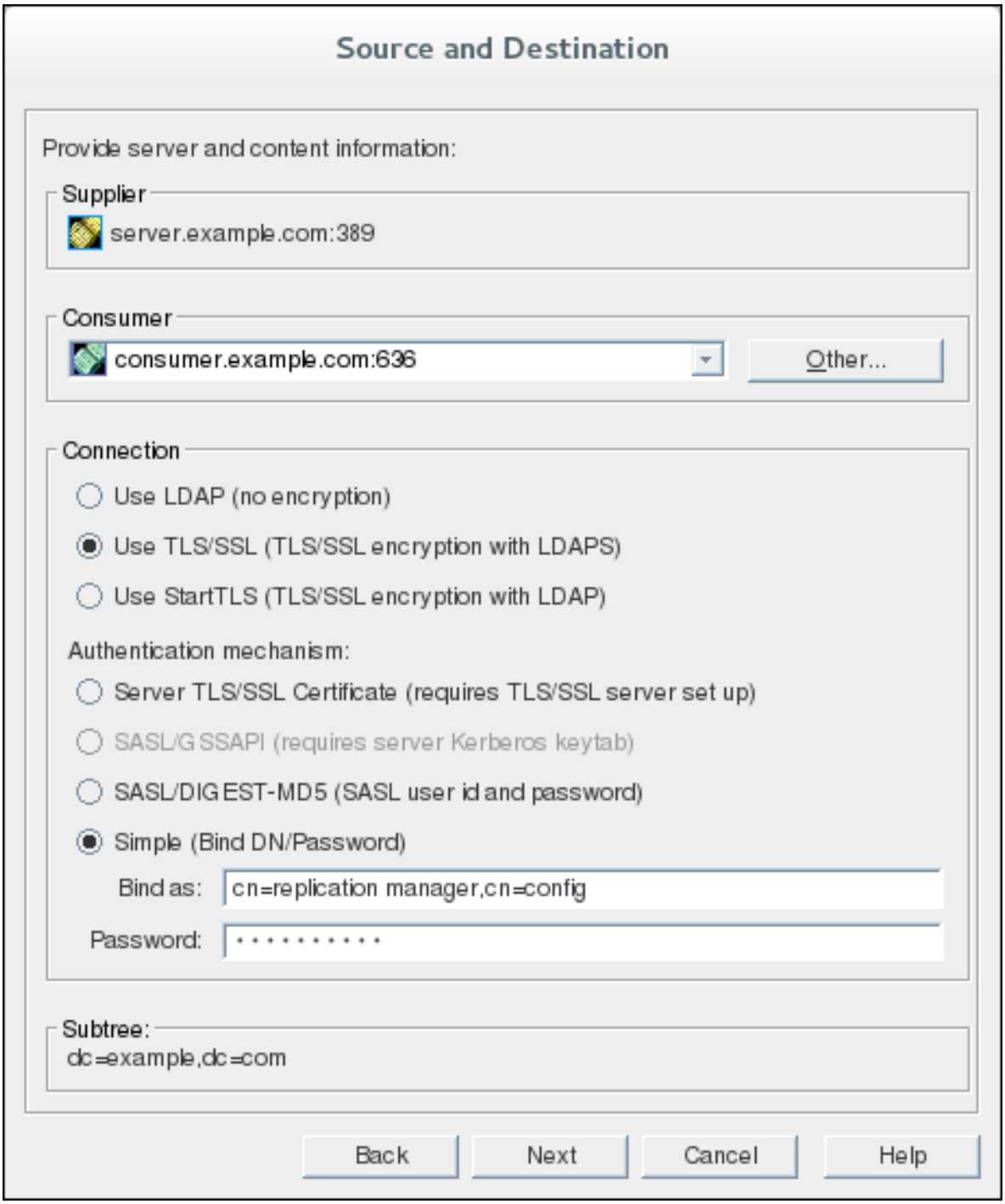
1. **Configuration** タブのナビゲーションツリーで、データベースを右クリックし、**New Replication Agreement** を選択します。



または、データベースを強調表示し、**Object** メニューから **New Replication Agreement** を選択して Replication Agreement Wizard を起動します。

2. 最初の画面で、レプリカ合意の名前および説明を入力し、**Next** を押します。
3. **Source and Destination** 画面で、コンシューマーの URL（hostname:port または IP_address:port ）と、コンシューマー上のサプライヤーバインド DN とパスワードを入力します。ターゲットサーバーが利用できない場合は、他のサーバーにアクセスして情報を手動で入

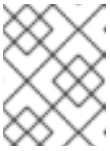
力します。



- 複数の Directory Server インスタンスが設定されていない場合、デフォルトでは、ドロップダウンメニューにはコンシューマーがありません。サーバー URL は、IPv4 アドレスまたは IPv6 アドレスで、hostname:port または IP_address:port の形式で手動で入力できます。

- Directory Server インスタンスが TLS で実行されるように設定されている場合でも、一覧表示されるポートは TLS 以外のポートになります。このポート番号は、コンソールで Directory Server インスタンスを識別するためにのみ使用されます。これは、レプリケーションに使用される実際のポート番号またはプロトコルを指定しません。

- サーバーで TLS が有効になっている場合は、TLS クライアント認証に **Using encrypted SSL connection** ラジオボタンを選択できます。それ以外の場合は、サプライヤーバインド DN およびパスワードを入力します。



注記

属性の暗号化が有効な場合は、暗号化された属性を複製するのにセキュアな接続が必要になります。

4. 接続タイプを選択します。以下の 3 つのオプションがあります。

   - LDAP を使用します。これにより、標準の暗号化されていない接続が設定されます。
   - TLS/SSL を使用します。これは、636 などのサーバーのセキュアな LDAPS ポートを介したセキュアな接続を使用します。この設定は TLS を使用するために必要です。
   - Start TLS を使用します。Start TLS を使用して、サーバーの標準ポートでセキュアな接続を確立します。

注記

シンプルなパスワード認証(「セキュアなバインドの要求」)にセキュアなバインドが必要な場合は、セキュアな接続で行われる場合を除き、レプリケーション操作は失敗します。セキュアな接続（TLS および Start TLS 接続または SASL 認証）の使用が推奨されます。

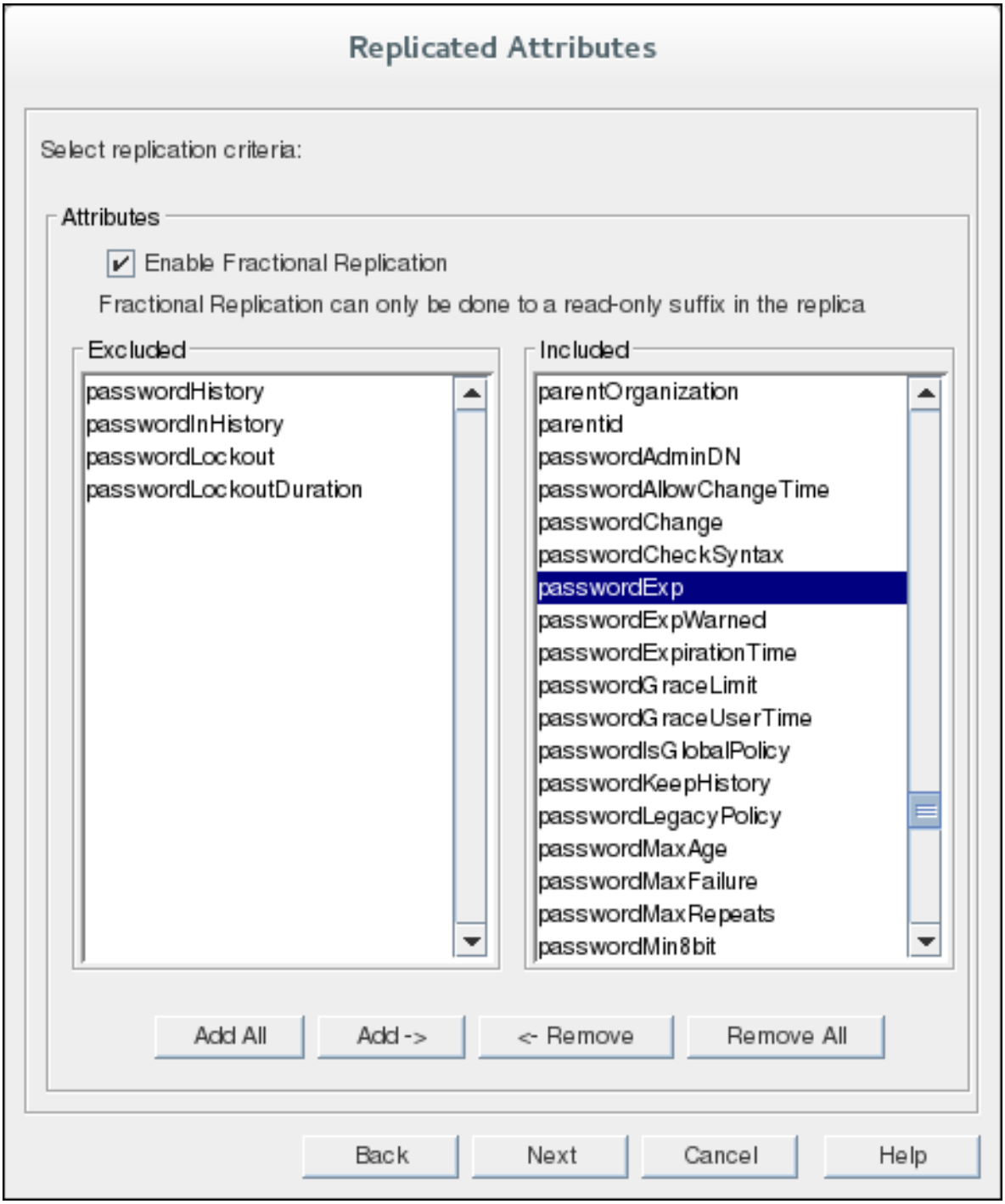
5. 適切な認証方法を選択し、必要な情報を提供します。これにより、サプライヤーがコンシューマーサーバーにバインドして更新を送信するために使用する情報を提供します。

   - simple は、サーバーが、暗号化なしで標準ポートで接続することを意味します。必要な情報は、Replication Manager のバインド DN およびパスワード（コンシューマーサーバーに存在する必要がある）です。
   - サーバー TLS/SSL 証明書は、サプライヤーの TLS 証明書を使用して、コンシューマーサーバーに対して認証します。証明書は、証明書ベースの認証のサプライヤーにインストールされ、コンシューマーサーバーには、サプライヤーの証明書内のサブジェクト DN をその Replication Manager エントリーにマッピングできるように、証明書マッピングを設定する必要があります。

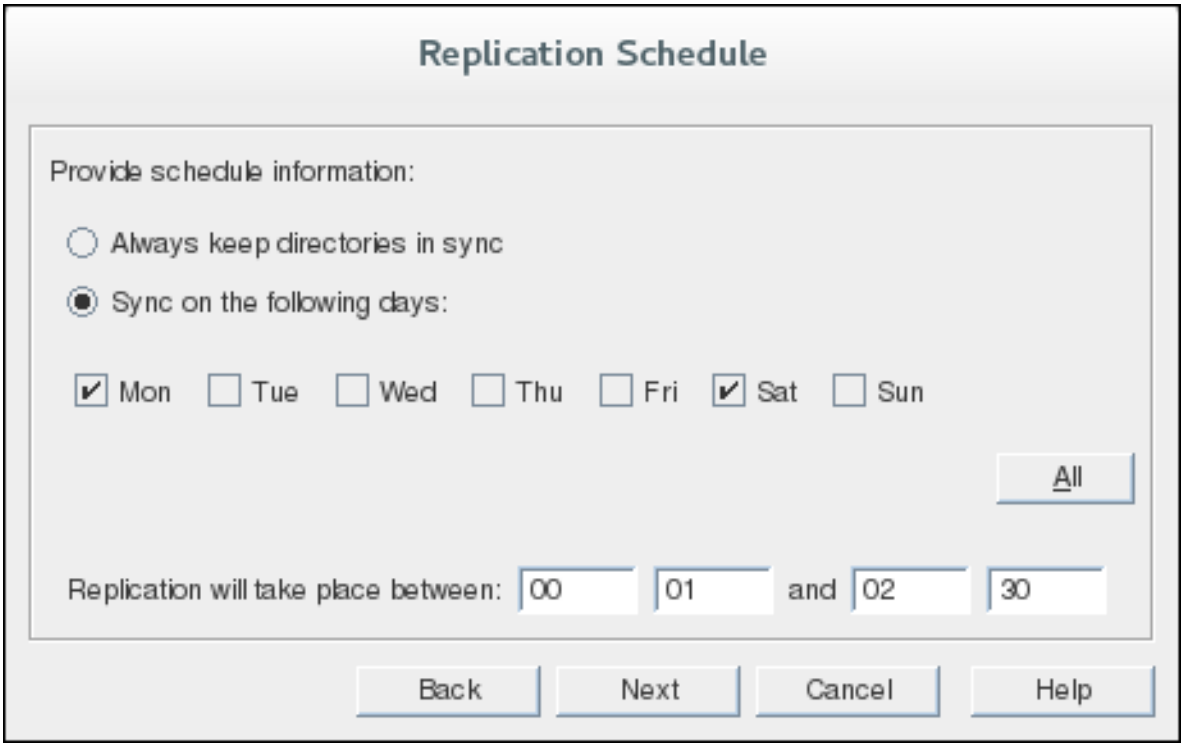
     TLS および証明書マッピングの設定については、「TLS の有効化」 を参照してください。
   - 簡易認証などの SASL/DIGEST-MD5 では、認証に使用するバインド DN とパスワードのみが必要になります。これは、標準または TLS 接続上で実行できます。
   - SASL/GSSAPI では、サプライヤーサーバーに Kerberos キータブ（「KDC サーバーおよびキータブの概要」にあるように）があり、コンシューマーサーバーでサプライヤーのプリンシパルを実際のレプリケーションマネージャーエントリーにマップするために SASL マッピングが必要です（「コンソールからの SASL アイデンティティーマッピングの設定」のように）。

6. Next を押します。

7. 一部レプリケーションは、サーバー間でエントリーがレプリケートされるエントリー属性を制御します。デフォルトでは、すべての属性がレプリケートされます。コンシューマーに複製されない属性 を選択するには、Enable Fractional Replication チェックボックスを選択します。次に、右側の Included コラムの属性（または属性）を強調表示し、Remove をクリックします。レプリケートされない属性はすべて左側の Excluded 列に一覧表示されます。また、レプリカ合意が完了する概要にも表示されます。



8. **レプリケーションの実行時にスケジュールを設定します。デフォルトでは、レプリケーションは継続的に実行されます。**



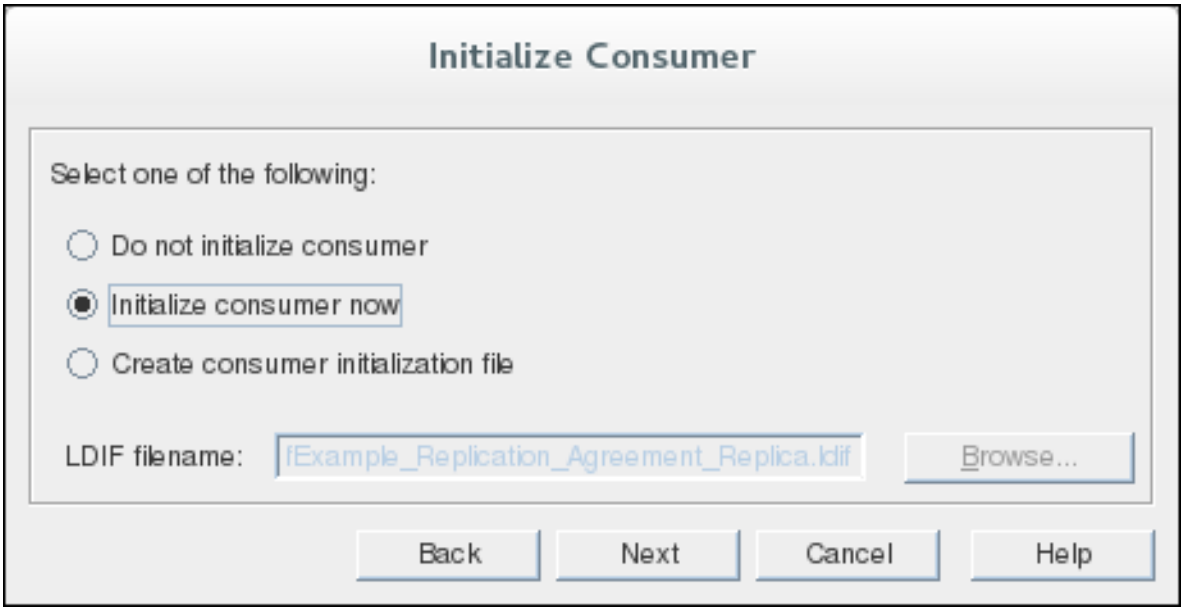
注記

レプリケーションスケジュールは、真夜中(0000)を越えません。そのため、0001 を開始し、同じ日に 2359 で終わるスケジュールを設定できますが、1 日の 2359 から開始したスケジュールを設定し、次の 部分で終了することはできません。

Next を押します。

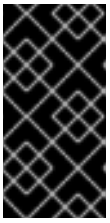
9. **コンシューマーが初期化されると設定されます。コンシューマー を初期化すると、サプライヤーからコンシューマーにすべてのデータを手動でコピーします。デフォルトでは、コンシューマーを後で初期化できるように初期化ファイル（すべてのサプライヤーデータの LDIF）を作成します。レプリカ合意が完了した後、または全くない時に、コンシューマーを初期化することもできます。コンシューマーの初期化の詳細は、「コンシューマーの初期化」 を参照してください。マルチマスターレプリケーションについては、以下を考慮してください。**

   - **1つのサプライヤーに、他のサプライヤーに複製するための完全なデータセットがあることを確認します。この 1つのサプライヤーを使用して、マルチマスターレプリケーションセットの他のすべてのサプライヤーでレプリカを初期化します。**
   - **いずれかのマルチマスターサプライヤーからコンシューマーサーバーのレプリカを初期化します。**
   - **レプリカ合意が設定されると、サーバー の再初期化 を試行しないでください。たとえば、server2 が server1 からすでに初期化されている場合は、server2 から server1 を初期化しないでください。この場合は、Do not initialize consumer を選択します。**



注記

レプリケーションは、コンシューマーが初期化されるまで 開始されません。

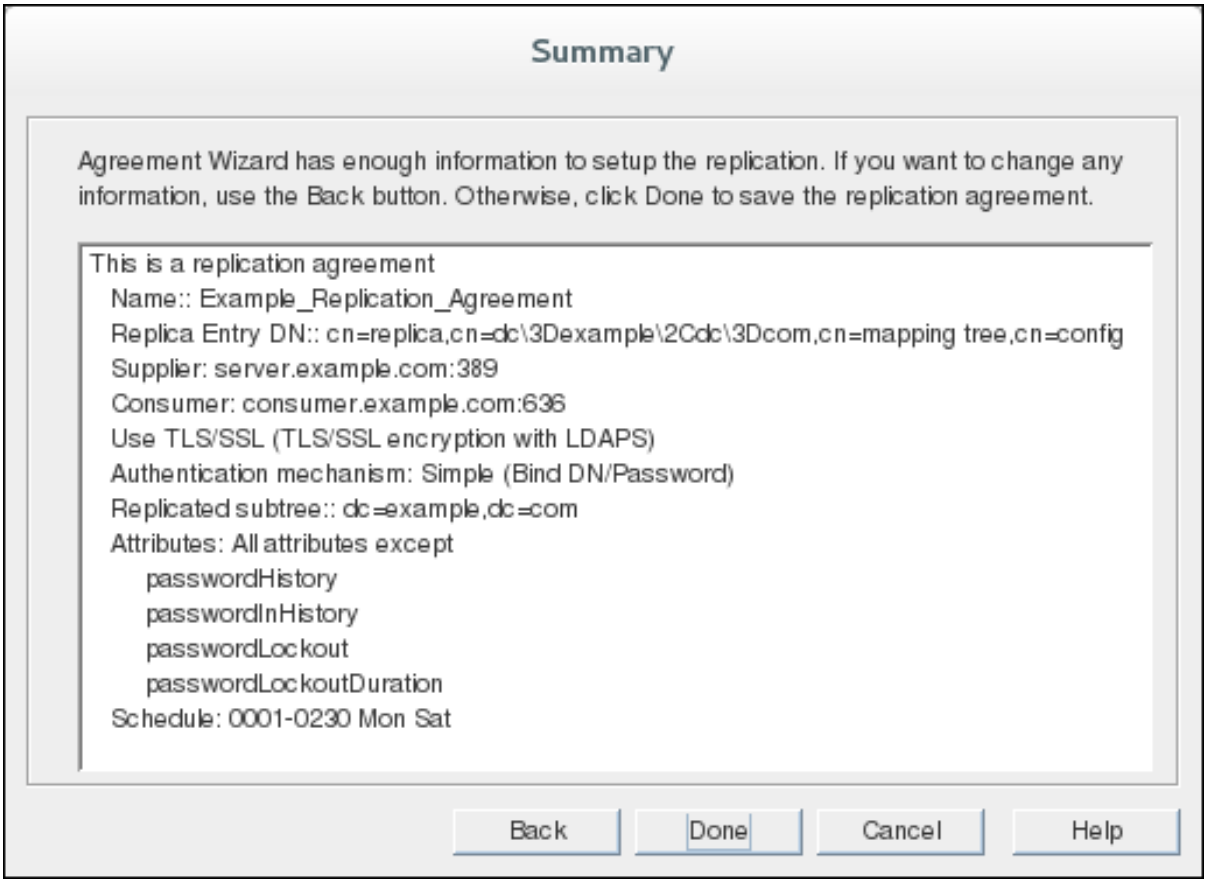


重要

マルチマスターレプリケーションの場合は、コンシューマーが 1 つのサプライヤーによって 1 度だけ 初期化されていることを確認します。レプリケーションのステータスを確認する際には、コンシューマーの初期化に使用された適切なサプライヤーでレプリカ合意のエントリーを確認してください。

Next を押します。

10. **最後の画面には dse.ldif ファイルに含まれるため、レプリカ合意の設定が表示されます。完了を押して合意を保存します。**



レプリカ合意が設定されている。

注記

この手順の最後で、すべてのサプライヤーサーバーには相互レプリカ合意があります。つまり、各サプライヤーサーバーは相互に更新を受け入れることができることを意味します。

注記

レプリカ合意の作成後、LDAP および LDAPS 接続が異なるポートを使用するため、接続タイプ（TLS または非 TLS）を変更することはできません。接続タイプを変更するには、レプリカ合意を再作成します。

15.6.4. マルチマスターレプリケーションにおけるコンシューマーの独占を防ぐ

マルチマスターレプリケーションの機能の 1 つは、サプライヤーが複製されたエリアのコンシューマーへの排他的アクセスを取得することです。この間、他のサプライヤーは、コンシューマーによる直接通信がロックされます。ロックアウトされた状態でサプライヤーがアクセス権を取得しようとすると、コンシューマーはビジー応答を返し、サプライヤーは数秒間スリープしてから再度アクセスを試みます。更新負荷が低い間に、最初のコンシューマーがロックされたときにサプライヤーが別のコンシューマーに更新を送信し、最初のコンシューマーが再び解放されると更新を送信します。

ロックサプライヤーの更新負荷が高かったり、changelog に多くの保留中の更新があったりすると、問題が発生することがあります。ロックサプライヤーが更新の送信を終了し、送信の保留中の変更が多くなると、他のサプライヤーは通常スリープ状態であるため、すぐにコンシューマーの再取得を試み、成功する可能性が高くなります。これにより、単一のサプライヤーが数時間またはそれ以上にわたってコンシューマーを独占することになります。

以下の属性は、この問題に対応します。

nsds5ReplicaBusyWaitTime

別のアクセスの取得を試みる前に、コンシューマーがビジー応答を返した後のサプライヤーが待機する時間を秒単位で設定します。

たとえば、別の取得を試みる前に、サプライヤーが 5 秒待機するように設定するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=Replication_Agreement_Name,cn=replica,cn=suffix_Name,cn=mapping tree,cn=config
changetype: modify
replace: nsds5ReplicaBusyWaitTime
nsds5ReplicaBusyWaitTime: 5
```

nsds5ReplicaSessionPauseTime

2 つの更新セッションの間にサプライヤーが待機する時間を秒単位で設定します。***nsds5ReplicaBusyWaitTime*** で指定した値またはそれよりも小さい値を設定すると、Directory Server は ***nsds5ReplicaSessionPauseTime*** パラメーターの値を自動的に使用します。これは、***nsds5ReplicaBusyWaitTime*** に設定した値よりも大きな値になります。

たとえば、サプライヤーは 2 つの更新セッション間で 10 秒待機するように設定するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=Replication_Agreement_Name,cn=replica,cn=suffix_Name,cn=mapping tree,cn=config
changetype: modify
replace: nsds5ReplicaSessionPauseTime
nsds5ReplicaSessionPauseTime: 10
```

nsds5ReplicaReleaseTimeout

更新の送信を終了したかどうかにかかわらず、マスターがレプリカを解放するタイムアウトを設定します。これにより、単一マスターがレプリカを独占しなくなります。

たとえば、マスターが 90 秒後にレプリカを解放する大規模なレプリケーション環境に設定するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=replica,cn=suffix_Name,cn=mapping tree,cn=config
changetype: modify
replace: nsds5ReplicaReleaseTimeout
nsds5ReplicaReleaseTimeout: 90
```

詳細は、Red『Hat Directory Server の設定、コマンド、およびファイルリファレンス のパラメーターの説明を参照してください』。

レプリカのビジーエラーをログに記録するには、Replication エラーログ (ログレベル 8192) を有効にします。「ログレベルの設定」 を参照してください。

15.7. カスケードレプリケーションの設定

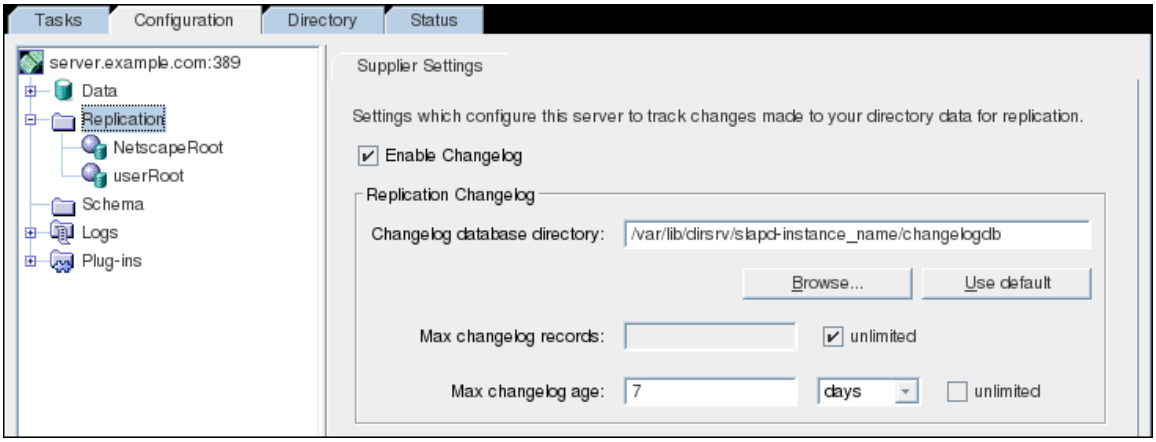
図15.4「カスケードレプリケーション」 にあるように、カスケードレプリケーションの設定は、シナリオの各サーバーごとに、読み取り/書き込みレプリカを保持するサーバー A のサプライヤー、読み取り専用レプリカを保持するハブサーバー B のコンシューマー、およびサーバー C のコンシューマー（読み取り専用レプリカを保持する）の 3 つの主要なステップがあります。

- 「Supplier サーバーでの読み書きレプリカの設定」
- 「コンシューマーサーバーでの読み取り専用レプリカの設定」
- 「ハブでの読み取り専用レプリカの設定」
- 「レプリカ合意の設定」

15.7.1. Supplier サーバーでの読み書きレプリカの設定

次に、データベースの元のコピーを保持するサプライヤーサーバーを設定します。

1. サーバーのサプライヤー設定を指定します。
   a. Directory Server コンソールで、**Configuration** タブを選択します。
   b. ナビゲーションツリーで、**Replication** フォルダーを選択します。
   c. ウィンドウの右側で、**Supplier Settings** タブを選択します。



   d. **Enable Changelog** チェックボックスを選択します。

      これにより、以前にグレーアウトされていたペインのフィールドがすべて有効になります。

   e. **Use default** ボタンをクリックして changelog を指定するか、**Browse** ボタンをクリックしてファイルセレクターを表示します。
   f. ログファイルの数と期間の changelog パラメーターを設定します。

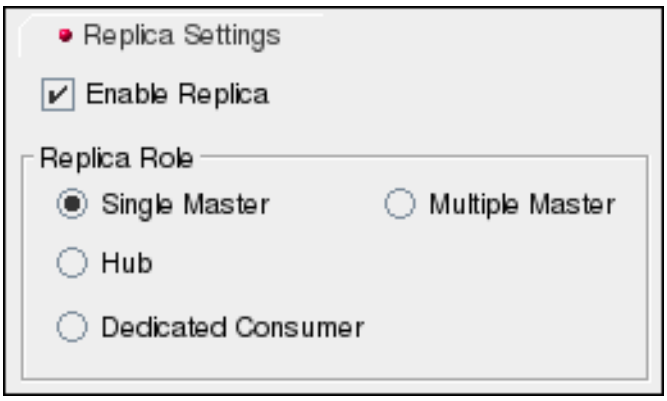
      異なる値を指定するには、無制限のチェックボックスをクリアします。

   g. **Save** をクリックします。
2. 読み取り/書き込み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. **Configuration** タブのナビゲーションツリーで、**Replication** ノードを展開し、複製するデータベースを強調表示します。

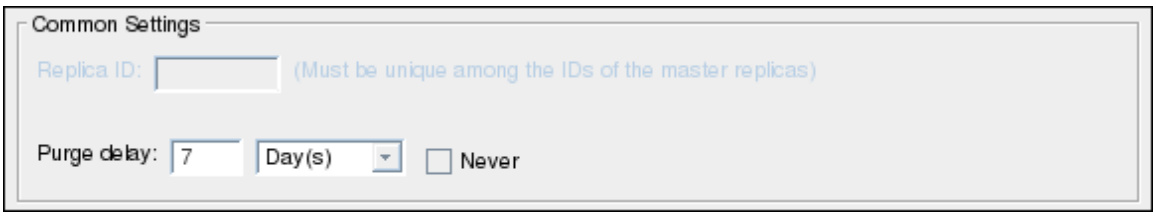
Replica Settings タブがウィンドウの右側で開きます。

b. Enable Replica チェックボックスを選択します。

c. Replica Role セクションで、Single Master ラジオボタンを選択します。



d. Common Settings セクションで、Replica ID を指定します。レプリカ ID は1 から 65534 までの整数です。



指定の接尾辞のレプリカには、レプリカ ID を一意にする必要があります。このサーバーおよび他のサーバーで読み取り/書き込みレプリカに使用される他の ID とは異なります。

e. Common Settings セクションで、Purge delay フィールドでパージ遅延を指定します。

パージの遅延は、複製されたエントリーに対する状態情報を削除する頻度です。

f. Save をクリックします。

サプライヤーレプリカの設定後に、レプリカ合意の設定を開始します。

15.7.2. コンシューマーサーバーでの読み取り専用レプリカの設定

1. 読み取り専用レプリカのデータベースがない場合は、これを作成します。接尾辞の作成方法については、「接尾辞の作成」 を参照してください。

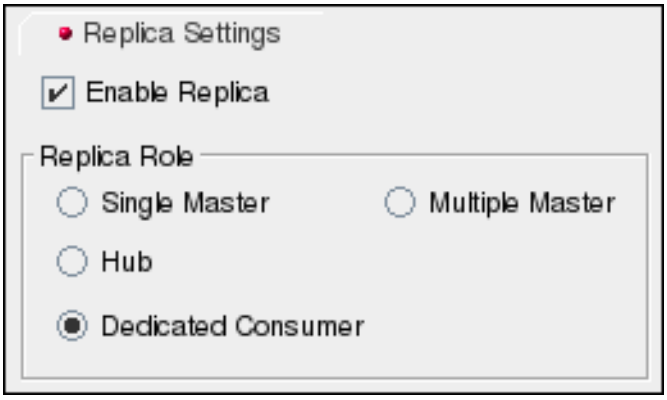
2. コンシューマーサーバーにサプライヤーバインド DN のエントリーを作成します（存在しない場合）。サプライヤーバインド DN は、サプライヤーがコンシューマーにバインドするために使用する特別なエントリーです。これは「サプライヤーバインド DN エントリーの作成」 で説明されています。

3. 読み取り専用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 コンソールで、Configuration タブを選択します。

   b. ナビゲーションツリーで、Replication フォルダーを展開し、レプリカデータベースを強調表示します。

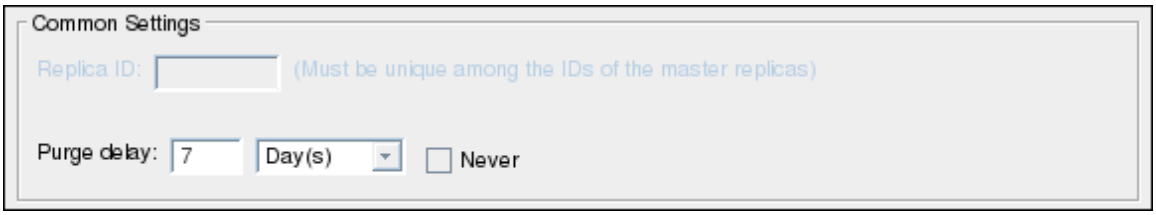
   ウィンドウの右側で、そのデータベースの Replica Settings タブが開きます。



c. **Enable Replica** チェックボックスを選択します。

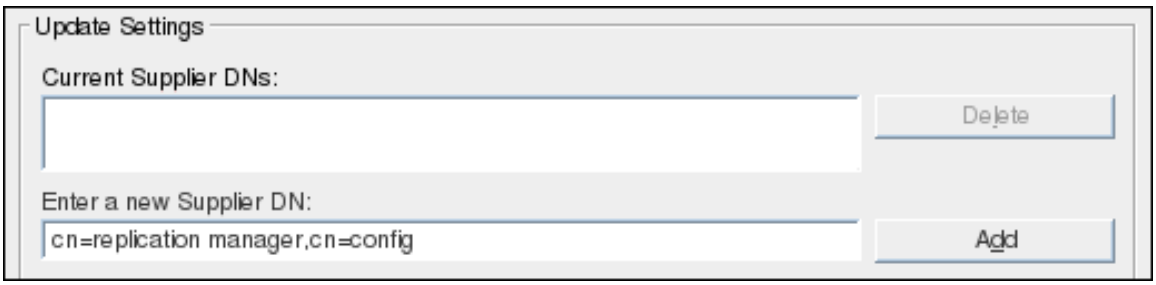
d. **Replica Role** セクションで、**Dedicated Consumer** ラジオボタンを選択します。

e. **Common Settings** セクションで、**Purge delay** フィールドでパージ遅延を指定します。



このオプションは、複製されたエントリーのステータス情報がパージされる頻度を示します。

f. **Update Settings** セクションで、サプライヤーがレプリカにバインドするために使用するバインド DN を指定します。**Enter a new Supplier DN** フィールドにサプライヤーバインド DN を入力し、**Add** をクリックします。サプライヤーバインド DN が現在のサプライヤー DN 一覧に表示されます。



サプライヤーバインド DN は、手順 2 で作成されたエントリーである必要があります。サプライヤーバインド DN は、複製されたデータベースのアクセス制御の対象ではないため、特権ユーザーです。

注記

コンシューマーごとに複数のサプライヤーバインド DN を指定できますが、レプリカ合意ごとに 1 つのサプライヤー DN のみがあります。

g. 更新を参照するサプライヤーサーバーの URL（hostname :port または IP_address:port、IPv4 アドレスまたは IPv6 アドレスを使用）を指定します。

デフォルトでは、すべての更新は、ここで指定されたサプライヤーサーバーが最初に参照されます。ここでサプライヤーが設定されていない場合、更新は現在のレプリカを含むレプリカ合意を持つサプライヤーサーバーと呼ばれます。

カスケードレプリケーションでは、参照は自動的にハブサーバーに送信され、元のサプライヤーへの要求を参照します。そのため、自動的に生成された参照を交換するように、元のサプライヤーに参照を設定します。

4. **Save** をクリックします。

レプリケーション設定のすべてのコンシューマーサーバーでこの手順を繰り返し、ハブレプリカを設定します。

15.7.3. ハブでの読み取り専用レプリカの設定

ハブを設定して、サプライヤ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ーに伝播します。

1. 読み取り専用レプリカのデータベースがない場合は、これを作成します。接尾辞の作成方法については、「接尾辞の作成」 を参照してください。

2. コンシューマーサーバーにサプライヤーバインド DN のエントリーを作成します（存在しない場合）。サプライヤーバインド DN は、サプライヤーがコンシューマーにバインドするために使用する特別なエントリーです。これは「サプライヤーバインド DN エントリーの作成」 で説明されています。

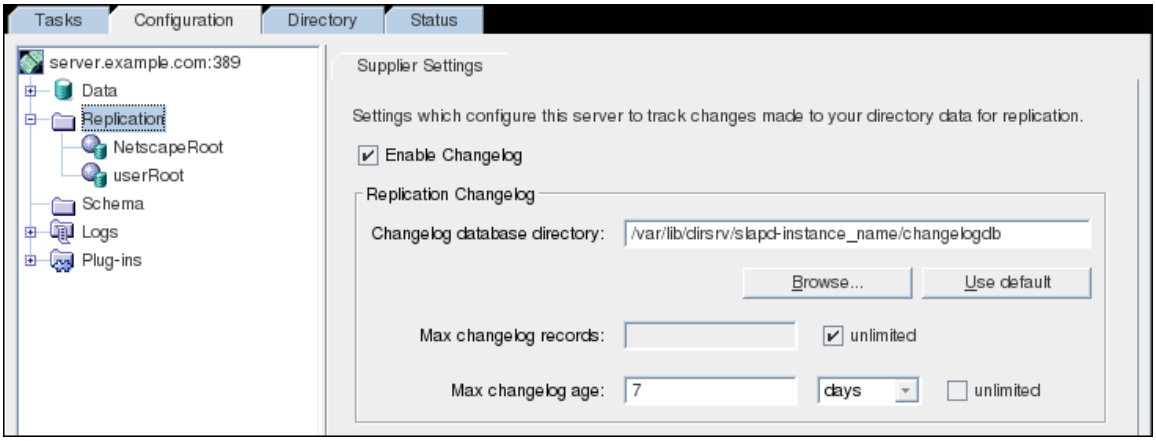
3. ハブサーバーの changelog を作成します。

   ハブは、サプライヤーサーバーから送信された変更を記録するので、更新操作を受け入れなくても changelog を維持する必要があります。

   a. Directory Server コンソールで、**Configuration** タブを選択します。

   b. ナビゲーションツリーで、**Replication** フォルダーを選択します。

   c. ウィンドウの右側で、**Supplier Settings** タブを選択します。



   d. **Enable Changelog** チェックボックスを選択します。

   これにより、以前にグレーアウトされていたペインのフィールドがすべて有効になります。

   e. **Use default** ボタンをクリックして changelog を指定するか、**Browse** ボタンをクリックしてファイルセレクターを表示します。

f. ログファイルの数と期間の changelog パラメーターを設定します。

異なる値を指定するには、無制限のチェックボックスをクリアします。

g. **Save** をクリックします。

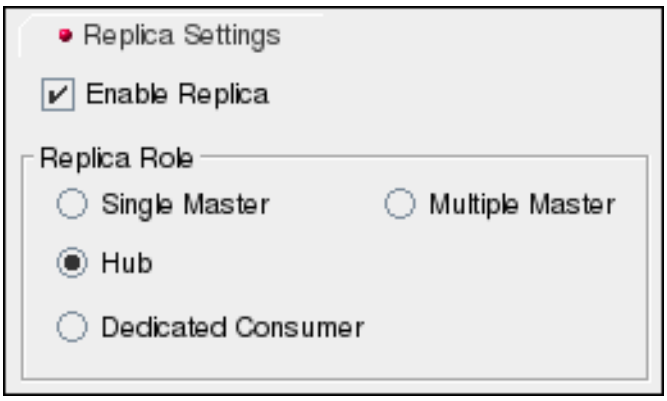
4. 必要なハブレプリカ設定を指定します。

a. Directory Server コンソールで、**Configuration** タブを選択します。

b. ナビゲーションツリーで、**Replication** フォルダーを展開し、レプリカデータベースを強調表示します。

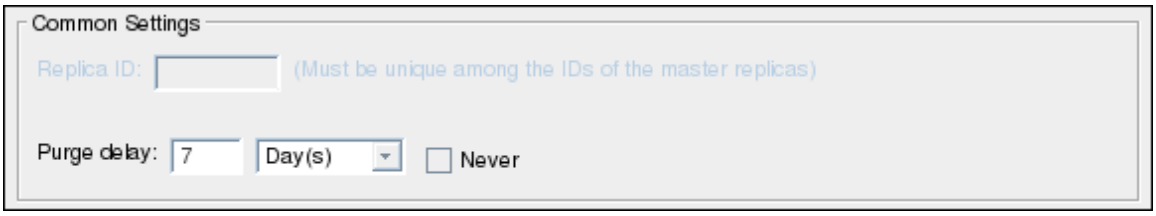
ウィンドウの右側で、そのデータベースの **Replica Settings** タブが開きます。

c. **Enable Replica** チェックボックスを選択します。



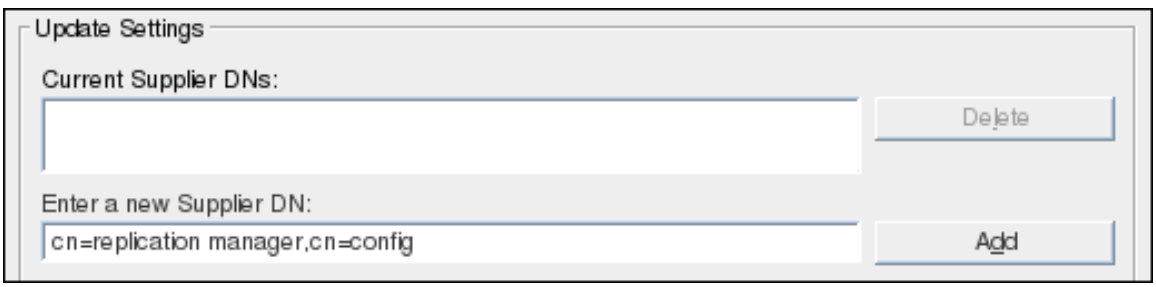
d. **Replica Role** セクションで、**Hub** ラジオボタンを選択します。

e. **Common Settings** セクションで、**Purge delay** フィールドでパージ遅延を指定します。

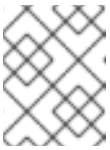


このオプションは、複製されたエントリーのステータス情報がパージされる頻度を設定します。

f. **Update Settings** セクションで、サプライヤーがレプリカにバインドするために使用するバインド DN を指定します。**Enter a new Supplier DN** フィールドにサプライヤーバインド **DN** を入力し、**Add** をクリックします。サプライヤーバインド DN が現在のサプライヤー **DN** 一覧に表示されます。



サプライヤーバインド DN は、手順 2 で作成されたエントリーである必要があります。サプライヤーバインド DN は、複製されたデータベースのアクセス制御の対象ではないため、特権ユーザーです。



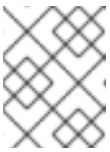
注記

コンシューマーごとに複数のサプライヤーバインド DN を指定できますが、レプリカ合意ごとに 1 つのサプライヤー DN のみがあります。

g. 更新を参照するサプライヤーサーバーの URL を指定します。

デフォルトでは、すべての更新は、ここで指定されたサプライヤーサーバーが最初に参照されます。ここでサプライヤーが設定されていない場合、更新は現在のレプリカを含むレプリカ合意を持つサプライヤーサーバーと呼ばれます。

自動参照は、クライアントが通常の接続上でバインドすることを前提としています。これには **ldap**://hostname:port 形式の URL があります。クライアントが TLS を使用してサプライヤーにバインドするには、このフィールドを使用して **ldaps**://hostname:port 形式の参照を指定します。**ldaps** の **s** はセキュアな接続を示します。



注記

ホスト名の代わりに IPv4 アドレスまたは IPv6 アドレスを使用することもできます。

5. **Save** をクリックします。

すべてのハブが設定されたら、サプライヤーレプリカを設定します。

15.7.4. レプリカ合意の設定

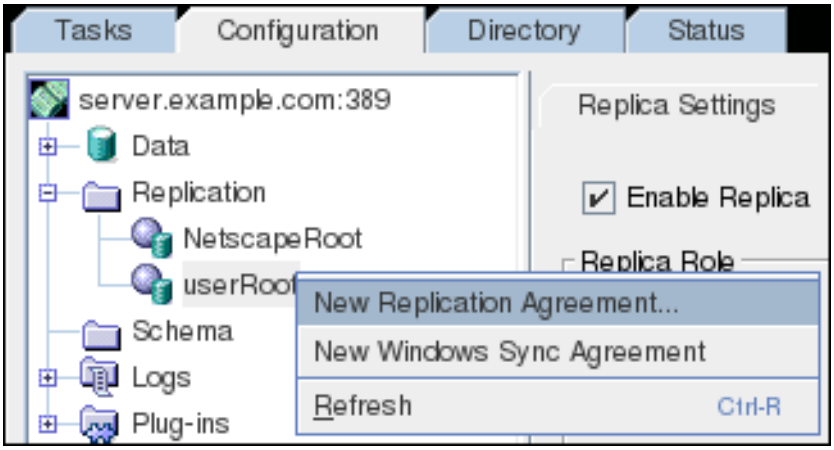
カスケードレプリケーションには、2 セットのレプリカ合意（サプライヤーとハブの間の 1 つとハブとコンシューマーの間の 2 セット）が必要です。レプリカ合意を設定するには、以下を実行します。

1. ハブのサプライヤーにレプリカ合意を作成してから、サプライヤーサーバーを使用してハブサーバーでレプリカを初期化します。

2. 次に、各コンシューマーのハブでレプリカ合意を作成し、ハブからコンシューマーレプリカを初期化します。

レプリカ合意を設定するには、以下を実行します。

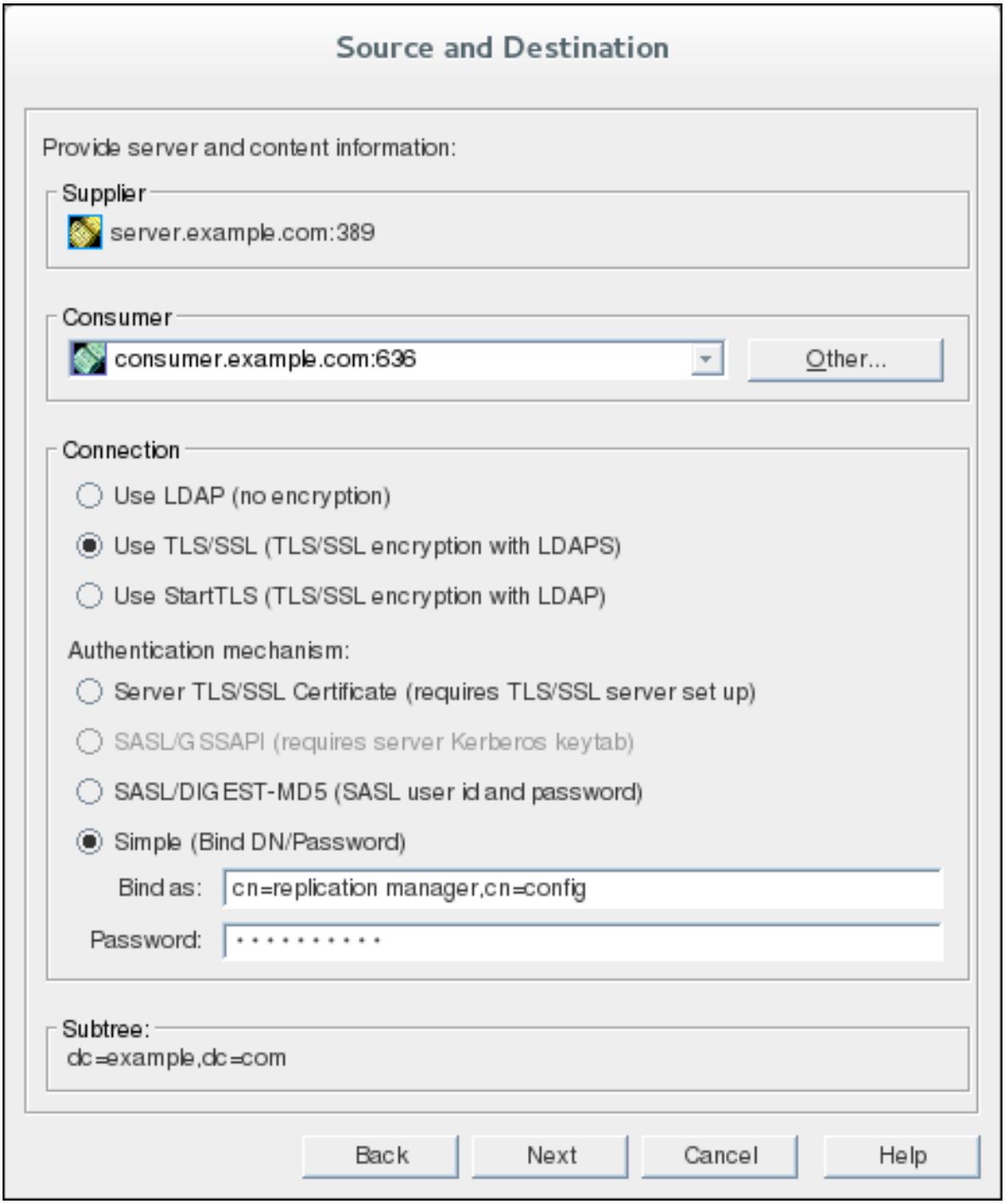
1. **Configuration** タブのナビゲーションツリーで、データベースを右クリックし、**New Replication Agreement** を選択します。



または、データベースを強調表示し、**Object** メニューから **New Replication Agreement** を選択して Replication Agreement Wizard を起動します。

2. 最初の画面で、レプリカ合意の名前および説明を入力し、**Next** を押します。

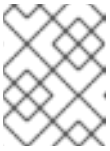
3. **Source and Destination** 画面で、コンシューマーの URL（hostname:port または IP_address:port ）と、コンシューマー上のサプライヤーバインド DN とパスワードを入力します。ターゲットサーバーが利用できない場合は、他のサーバーにアクセスして情報を手動で入力します。



- 複数の Directory Server インスタンスが設定されていない場合、デフォルトでは、ドロップダウンメニューにはコンシューマーがありません。サーバーの URL は、IPv4 アドレスまたは IPv6 アドレスで .hostname :port または IP_address:port として手動で入力できます。

- Directory Server インスタンスが TLS で実行されるように設定されている場合でも、一覧表示されるポートは TLS 以外のポートになります。このポート番号は、コンソールで Directory Server インスタンスを識別するためにのみ使用されます。これは、レプリケーションに使用される実際のポート番号またはプロトコルを指定しません。

- サーバーで TLS が有効になっている場合は、TLS クライアント認証に **Using encrypted SSL connection** ラジオボタンを選択できます。それ以外の場合は、サプライヤーバインド DN およびパスワードを入力します。

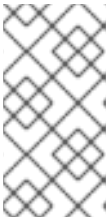


注記

属性の暗号化が有効な場合は、暗号化された属性を複製するセキュアな接続を使用する必要があります。

4. 接続タイプを選択します。以下の 3 つのオプションがあります。

   - LDAP を使用します。これにより、標準の暗号化されていない接続が設定されます。
   - TLS/SSL を使用します。これは、636 などのサーバーのセキュアな LDAPS ポートを介したセキュアな接続を使用します。この設定は TLS を使用するために必要です。
   - Start TLS を使用します。Start TLS を使用して、サーバーの標準ポートでセキュアな接続を確立します。



注記

シンプルなパスワード認証(「セキュアなバインドの要求」)にセキュアなバインドが必要な場合は、セキュアな接続で行われる場合を除き、レプリケーション操作は失敗します。セキュアな接続（TLS および Start TLS 接続または SASL 認証）の使用が推奨されます。

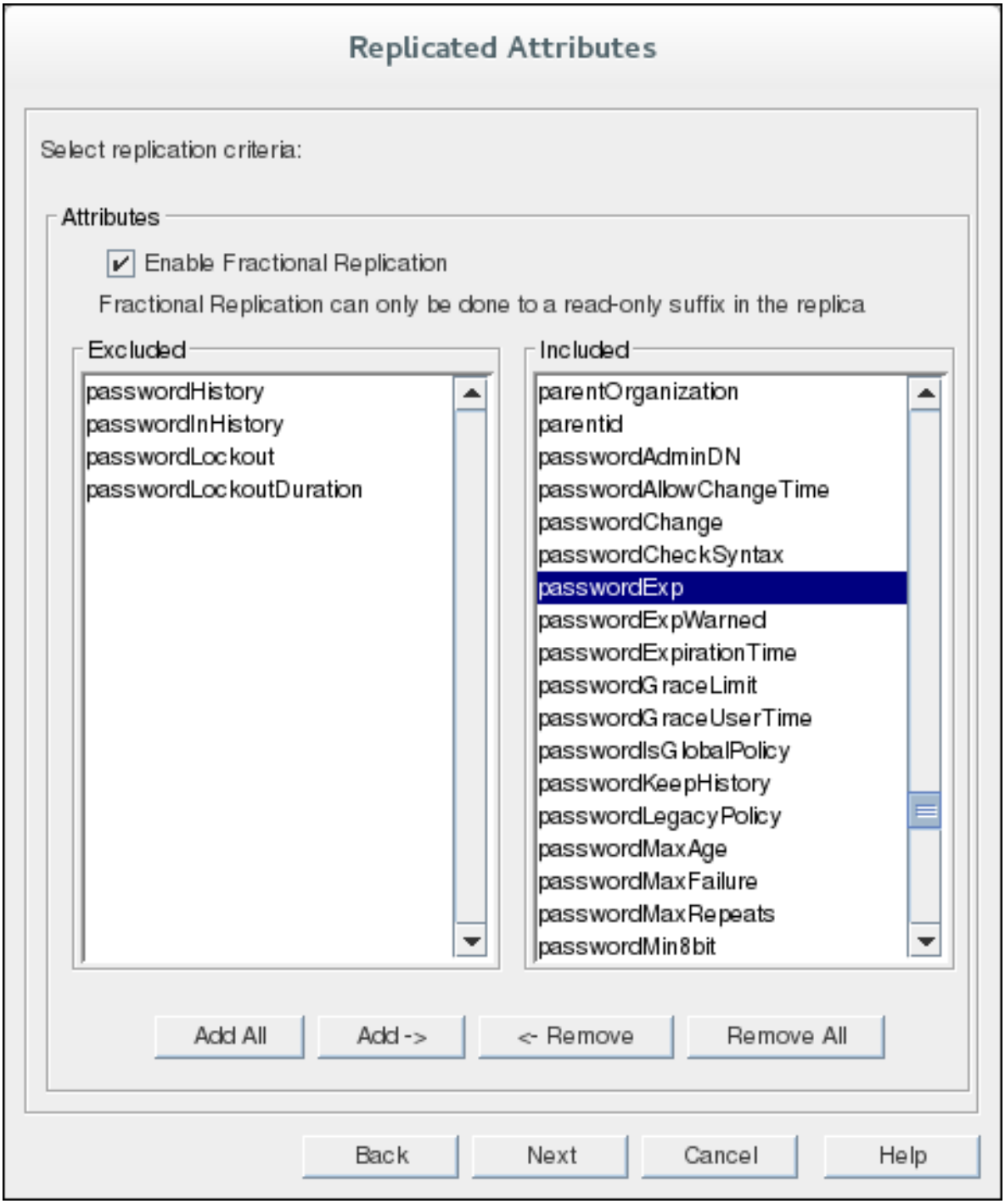
5. 適切な認証方法を選択し、必要な情報を提供します。これにより、サプライヤーがコンシューマーサーバーにバインドして更新を送信するために使用する情報を提供します。

   - simple は、サーバーが、暗号化なしで標準ポートで接続することを意味します。必要な情報は、Replication Manager のバインド DN およびパスワード（コンシューマーサーバーに存在する必要がある）です。
   - サーバー TLS/SSL 証明書は、サプライヤーの TLS 証明書を使用して、コンシューマーサーバーに対して認証します。証明書は、証明書ベースの認証のサプライヤーにインストールされ、コンシューマーサーバーには、サプライヤーの証明書内のサブジェクト DN をその Replication Manager エントリーにマッピングできるように、証明書マッピングを設定する必要があります。

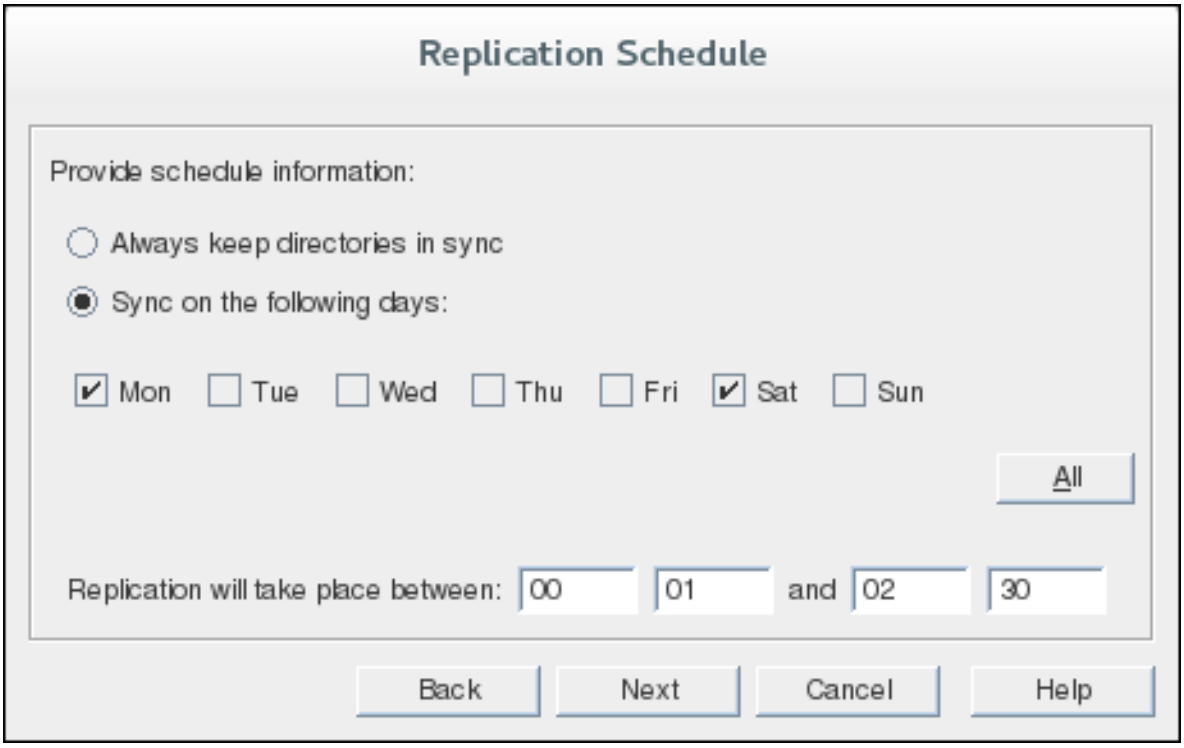
     TSL および証明書マッピングの設定については、「TLS の有効化」 を参照してください。
   - 簡易認証などの SASL/DIGEST-MD5 では、認証に使用するバインド DN とパスワードのみが必要になります。これは、標準または TLS 接続上で実行できます。
   - SASL/GSSAPI では、サプライヤーサーバーに Kerberos キータブ（「KDC サーバーおよびキータブの概要」にあるように）があり、コンシューマーサーバーでサプライヤーのプリンシパルを実際のレプリケーションマネージャーエントリーにマップするために SASL マッピングが必要です（「コンソールからの SASL アイデンティティーマッピングの設定」のように）。

6. **Next** を押します。

7. 一部レプリケーションは、サーバー間でエントリーがレプリケートされるエントリー属性を制御します。デフォルトでは、すべての属性がレプリケートされます。コンシューマーに複製されない属性 を選択するには、**Enable Fractional Replication** チェックボックスを選択します。次に、右側の **Included** コラムの属性（または属性）を強調表示し、**Remove** をクリックします。レプリケートされない属性はすべて左側の **Excluded** 列に一覧表示されます。また、レプリカ合意が完了する概要にも表示されます。



8. **レプリケーションの実行時にスケジュールを設定します。デフォルトでは、レプリケーションは継続的に実行されます。**



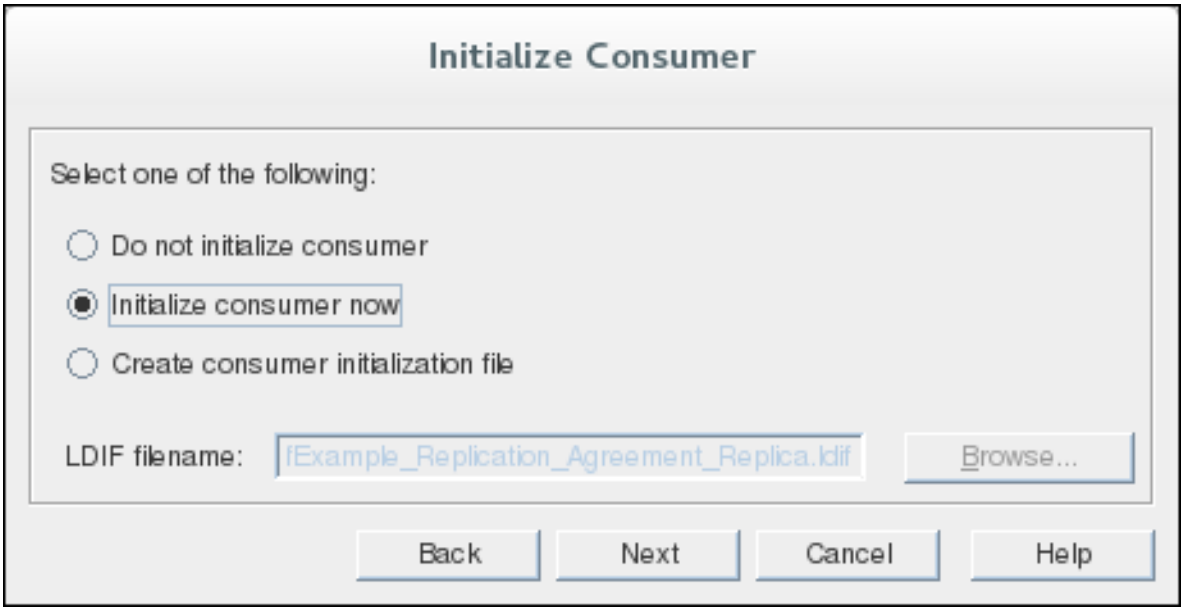
注記

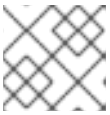
レプリケーションスケジュールは、真夜中(0000)を越えません。そのため、0001 を開始し、同じ日に 2359 で終わるスケジュールを設定できますが、1 日の 2359 から開始したスケジュールを設定し、次の 部分で終了することはできません。

Next を押します。

9. **コンシューマーが初期化されると設定されます。コンシューマー を初期化すると、サプライヤーからコンシューマーにすべてのデータを手動でコピーします。デフォルトでは、コンシューマーを後で初期化できるように初期化ファイル（すべてのサプライヤーデータの LDIF）を作成します。レプリカ合意が完了した後、または全くない時に、コンシューマーを初期化することもできます。コンシューマーの初期化の詳細は、「コンシューマーの初期化」 を参照してください。カスケードレプリケーションについては、以下を考慮してください。**

   - **最初にサプライヤーで supplier-hub レプリカ合意を作成し、サプライヤーからハブを初期化します。**
   - **ハブで hub-consumer レプリカ合意を作成し、ハブからコンシューマーを初期化します。**





注記

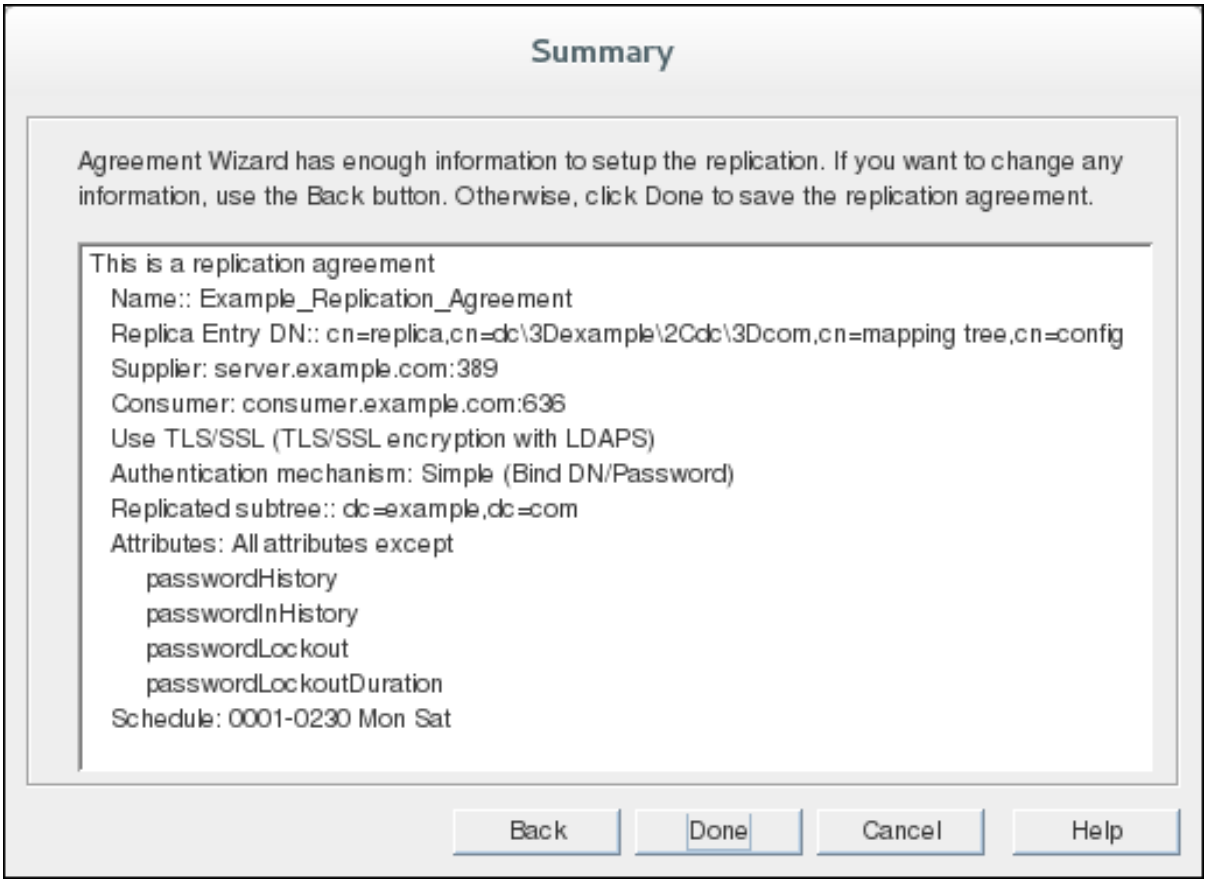
レプリケーションは、コンシューマーが初期化されるまで 開始されません。

重要

マルチマスターレプリケーションの場合は、コンシューマーが 1 つのサプライヤーによって 1 度だけ 初期化されていることを確認します。レプリケーションのステータスを確認する際には、コンシューマーの初期化に使用された適切なサプライヤーでレプリカ合意のエントリーを確認してください。

Next を押します。

10. **最後の画面には dse.ldif ファイルに含まれるため、レプリカ合意の設定が表示されます。完了を押して合意を保存します。**



注記

レプリカ合意の作成後、LDAP および LDAPS 接続が異なるポートを使用するため、接続タイプ（TLS または non-TLS）は変更できません。接続タイプを変更するには、レプリカ合意を再作成します。

15.8. 一時的にレプリケーションを一時停止

レプリケーションを一時的に中断するには、レプリカ合意を無効にします。合意を再度有効にしたら、レプリケーションを続行します。詳細は、「レプリカ合意の無効化および再有効化」 を参照してください。

15.9. レプリカ合意の無効化および再有効化

レプリカ合意を一時的に無効にするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=replication_agreement_name,cn=replica,cn=suffix_DN,cn=mapping tree,cn=config
changetype: modify
replace: nsds5ReplicaEnabled
nsds5ReplicaEnabled: off
```

レプリカ合意を再度有効にするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=replication_agreement_name,cn=replica,cn=suffix_DN,cn=mapping tree,cn=config
changetype: modify
```

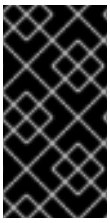
```
replace: nsds5ReplicaEnabled
nsds5ReplicaEnabled: on
```

15.10. 一部レプリケーションによる属性の管理

「一部レプリケーションを使用した属性のサブセットの複製」で説明するように、一部レプリケーションでは、管理者がレプリケーションの更新から 除外 する属性を設定することができます。管理者は、さまざまなパフォーマンスの理由により、ネットワーク上で送信される大きな属性の数を制限したり、修正タスク (*memberOf* 計算など) が実行される回数を減らすために実行できます。

レプリケーションから除外する属性のリストは、*nsDS5ReplicatedAttributeList* 属性で定義されます。この属性はレプリカ合意の一部で、レプリカ合意の作成時に（またはコマンドラインから）Directory Server Console のレプリカ合意ウィザードで設定できます。

```
nsDS5ReplicatedAttributeList: (objectclass=*) $ EXCLUDE memberof authorityRevocationList
accountUnlockTime
```

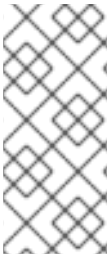


重要

Directory Server には、*nsDS5ReplicatedAttributeList* 属性の値に (objectclass=*) $ EXCLUDE の部分が必要です。ldapmodify ユーティリティーなどを使用して属性を直接編集する場合は、上記の例で示されている属性の一覧とともにこの部分を指定する必要があります。

15.10.1. 合計更新および増分更新での異なる一部レプリケーション属性の設定

一部レプリケーションが最初に設定されている場合、除外された属性の一覧は更新操作ごとに適用されます。つまり、属性の一覧は、完全更新と通常の増分更新に対して除外されます。しかし、パフォーマンスを向上させるために増分更新から属性を除外しても、ディレクトリデータセットを完全にするために全体更新に含めるべき場合があります。この場合は、全体更新 (*nsDS5ReplicatedAttributeListTotal*) から除外する属性の別のリストを定義する 2 番目の属性を追加できます。



注記

nsDS5ReplicatedAttributeList プライマリーの一部レプリケーション属性です。*nsDS5ReplicatedAttributeList* のみが設定されている場合、増分更新と合計更新の両方に適用されます。*nsDS5ReplicatedAttributeList* と *nsDS5ReplicatedAttributeListTotal* の両方が設定されている場合、*nsDS5ReplicatedAttributeList* は増分更新にのみ適用されます。

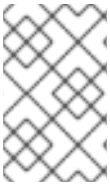
たとえば、*memberOf* 属性がエントリーに追加されるたびに、memberOf 修正タスクが実行してグループメンバーシップを解決します。これにより、レプリケーションが発生するたびにそのタスクが実行する場合に、サーバーでオーバーヘッドが発生する可能性があります。合計の更新は、レプリケーションに新たに追加されたり、長期間オフラインになったデータベースでのみ実行されるため、合計更新後の memberOf 修正タスクを実行すると、合計値が許容オプションになります。この場合、*nsDS5ReplicatedAttributeList* 属性には *memberOf* というリストが記載されるため、増分更新から除外されるようにしますが、*nsDS5ReplicatedAttributeListTotal* は *memberOf* を一覧表示しないため、すべての更新に含まれるようにします。

増分更新の除外リストは、レプリカ合意の *nsDS5ReplicatedAttributeList* 属性に設定されます。

```
nsds5replicatedattributelist: (objectclass=*) $ EXCLUDE authorityRevocationList
accountUnlockTime memberof
```

nsDS5ReplicatedAttributeList が唯一の属性セットである場合、そのリストは増分更新と合計更新の両方に適用されます。更新の合計に別のリストを設定するには、*nsDS5ReplicatedAttributeListTotal* 属性をレプリカ合意に追加します。

```
# ldapmodify -D "cn=Directory Manager" -W -x -D "cn=directory manager" -W -p 389 -h
server.example.com -x

dn: cn=ExampleAgreement,cn=replica,cn=dc\=example\,dc\=com,cn=mapping tree,cn=config
changetype: modify
add: nsDS5ReplicatedAttributeListTotal
nsDS5ReplicatedAttributeListTotal: (objectclass=*) $ EXCLUDE accountUnlockTime
```



注記

nsDS5ReplicatedAttributeList 属性は、すべての更新に対して *nsDS5ReplicatedAttributeListTotal* を設定する前に増分更新のために設定される必要があります。

15.10.2. 一部レプリケーションによる「空」のアップデートの防止

一部レプリケーションでは、レプリケーション更新 (*nsDS5ReplicatedAttributeList*) から削除される属性の一覧が許可されます。しかし、除外された属性への変更があっても、修正イベントが発生し、空のレプリケーション更新が生成されます。

nsds5ReplicaStripAttrs 属性は、空のレプリケーションイベントでは送信できず、更新シーケンスから削除される属性の一覧を追加します。これには、*modifiersName* のような運用上の利便性が含まれます。

たとえば、*accountUnlockTime* 属性が除外されたとします。John Smith のユーザーアカウントがロックされ、期間が期限切れになり、自動的にロック解除されます。*accountUnlockTime* 属性のみが変更し、その属性はレプリケーションから除外されます。ただし、運用する属性 *internalmodifytimestamp* も 変更しています。John Smith のユーザーアカウントが変更しているため、レプリケーションイベントがトリガーされます。ただし、送信する唯一のデータは新しい変更タイムスタンプであり、更新は空になります。(たとえば) ログイン時間やパスワードの有効期限などに関連する多くの属性がある場合は、空のレプリケーション更新が作成され、サーバーのパフォーマンスに悪影響を与えるか、関連するアプリケーションを妨げる可能性があります。

これを防ぐには、一部レプリケーションの動作を調整するのに役立つように、*nsds5ReplicaStripAttrs* 属性をレプリカ合意に追加します。

```
# ldapmodify -D "cn=Directory Manager" -W -x -D "cn=directory manager" -W -p 389 -h
server.example.com -x

dn: cn=ExampleAgreement,cn=replica,cn=dc\=example\,dc\=com,cn=mapping tree,cn=config
changetype: modify
add: nsds5ReplicaStripAttrs
nsds5ReplicaStripAttrs: modifiersname modifytimestamp internalmodifiersname
internalmodifytimestamp
```

レプリケーションイベントが 空でない 場合は、削除済みの属性 は 他の変更で複製されます。これらの属性は、イベントが空である場合にのみ更新から削除されます。

15.11. 読み取り専用レプリカの設定

読み取り専用のサーバーを書き込み可能にすると、レプリカを専用のコンシューマーまたはハブからサプライヤーに変更できます。

1. 実行中の更新がないことを確認します。
2. サプライヤーサーバーを停止します。
3. 変更ログを有効にします。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=changelog5,cn=config

changetype: add
objectClass: top
objectClass: extensibleObject
cn: changelog5
nsslapd-changelogdir: /var/lib/dirsrv/slapd-instance_name/changelogdb/
```

4. レプリカロールを変更します。
   - 単一マスターの場合：

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
changetype: modify
replace: nsDS5ReplicaType
nsDS5ReplicaType: 3
-
replace: nsDS5Flags
nsDS5Flags: 1
-
replace: nsDS5ReplicaId
nsDS5ReplicaId: unique_replica_id
-
delete: nsDS5ReplicaBindDN
```

   - マルチマスターの場合：

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
changetype: modify
replace: nsDS5ReplicaType
nsDS5ReplicaType: 3
-
replace: nsDS5Flags
nsDS5Flags: 1
-
replace: nsDS5ReplicaId
nsDS5ReplicaId: unique_replica_id
-
replace: nsDS5ReplicaBindDN
nsDS5ReplicaBindDN: cn=Replication Manager,cn=config
```

5. Directory Server インスタンスを停止します。
   -

```
# systemctl stop dirsrv
```

6. /etc/dirsrv/slapd-*instance*/dse.ldif ファイルをバックアップします。

```
# cp /etc/dirsrv/slapd-instance_name/dse.ldif \
    /etc/dirsrv/slapd-instance_name/dse.ldif-1
```

バックアップファイル dse.ldif.bak には名前を付けないでください。Directory Server は、このファイル名を使用して dse.ldif ファイルの既知の作業コピーを保持します。

7. /etc/dirsrv/slapd-*instance_name*/dse.ldif ファイルを編集します。

a. レプリカ合意を検索します。以下に例を示します。

```
dn: cn=replica,cn=dc\5c3Dexample\5c2Cdc\5c3Dcom,cn=mapping tree,cn=config
```

b. すべてのレプリカ合意から *nsState* 属性を含む行を削除します。

8. Directory Server インスタンスを停止します。

```
# systemctl start dirsrv.target
```

9. エラーメッセージがないか、エラーログファイルを監視します。詳細は、「ログの表示」 を参照してください。

レプリケーションが失敗した場合は、以下を行います。

a. すべてのレプリカ合意を削除します(「レプリケーショントポロジーからのサプライヤーの削除」)。

b. レプリケーションを無効にします：「レプリカ合意の無効化および再有効化」

c. changelog 設定を削除します(「Changelog の削除」)。

d. Directory Server および管理コンソールを再起動します。

```
# systemctl restart dirsrv.target
# systemctl restart dirsrv-admin.service
```

e. レプリケーションを有効にします(「レプリカ合意の無効化および再有効化」)。

f. レプリカ合意の作成：「レプリケーションシナリオ」

15.12. レプリケーショントポロジーからのサプライヤーの削除

レプリケーショントポロジーからサプライヤーを完全に削除することは、単にサプライヤーエントリーを削除するよりも複雑です。これは、トポロジー内のすべてのサプライヤーが他のサプライヤーに関する情報を保存し、サプライヤーが利用できない状態になった場合でも、その情報を保持するためです。

レプリケーショントポロジーに関する情報、つまり、相互に、および同じレプリケーショングループ内の他のレプリカに更新を提供するすべてのサプライヤーは、レプリカ更新ベクトル (RUV) と呼ばれるメタデータのセットに含まれています。RUV には、ID と URL、ローカルサーバー上の最新の変更状態

番号 (CSN)、最初の変更の CSN などのサプライヤーに関する情報が含まれています。サプライヤーとコンシューマーはいずれも RUV 情報を保存し、これを使用してレプリケーションの更新を制御します。

サプライヤーを正常に削除するには、そのメタデータを設定エントリーとともに削除する必要があります。

1. 削除するレプリカで、 更新を防ぐためにデータベースを読み取り専用モードにします。

```
# ldapmodify -D "cn=Directory Manager" -W -x -p 389 -h dead-replica.example.com

dn: cn=userRoot,cn=ldbm database,cn=plugins,cn=config
changetype: modify
replace: nsslapd-readonly
nsslapd-readonly: on
```

   レプリカが保留中の変更をすべてフラッシュするまで数分待機します。

2. トポロジー内の他のすべてのサプライヤーで、削除するレプリカとのレプリカ合意を削除します。

```
# ldapmodify -D "cn=Directory Manager" -W -x -p 389 -h replica1.example.com

dn: cn=Agmt_with_dead-
replica,cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
changetype: delete
```

3. 削除するレプリカで、削除する レプリカのレプリカ ID を取得します。これは設定エントリーの *nsds5replicaid* 属性にあります。

```
# ldapsearch -xLLL -D "cn=Directory Manager" -W -s sub -b cn=config
objectclass=nsds5replica nsds5replicaid

dn: cn=dead-replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
nsds5replicaid: 55
...
```

4. 削除するレプリカで、 すべてのレプリカ合意エントリーとその独自の設定エントリーを削除します。

```
# ldapmodify -D "cn=Directory Manager" -W -x -p 389 -h dead-replica.example.com

dn: cn=to_replica1,cn=dead-replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping
tree,cn=config
changetype: delete
...

dn: cn=to_replica2,cn=dead-replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping
tree,cn=config
changetype: delete

dn: cn=dead-replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
changetype: delete
```

5. トポロジー内の他のマスターサーバーの 1 つで、 レプリカ ID でクリーン コマンドを実行します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=clean 55, cn=cleanallruv, cn=tasks, cn=config
objectclass: extensibleObject
replica-base-dn: dc=example,dc=com
replica-id: 55
cn: clean 55
```

各レプリカの tombstone エントリーを検索して、他のレプリカでタスクの進捗を監視できます。

```
# ldapsearch -xLLL -D "cn=Directory Manager" -W -h remaining-replica.example.com -
b "dc=example,dc=com" '(&(nsuniqueid=ffffffff-ffffffff-ffffffff-ffffffff)
(objectclass=nstombstone))' nsds50ruv
```

15.13. レプリケーションを使用した削除されたエントリーの管理

エントリーが削除されると、すぐにデータベースから削除されません。代わりに、tombstone エントリーに変換されます。これは、レプリケーションのサーバーで使用されるバックアップエントリーを使用して競合を解決します。tombstone エントリーは、元のエントリーの状態情報を保持します。

レプリケーションの競合がある場合は、サプライヤーはレプリカ ID（変更が開始されるサーバー）と変更シーケンス番号の変更のタイムスタンプを使用して競合を解決します。最も古い変更が優先されます。削除されたエントリーと同様に、削除された属性も tombstone エントリーに保存されます。

廃棄は無限に保持されません。パージジョブは、指定した間隔で定期的に実行します (***nsDS5ReplicaTombstonePurgeInterval*** 属性の設定)。パージは古い tombstone エントリーを削除します。tombstone エントリーは、指定の期間 (***nsDS5ReplicaPurgeDelay*** 属性で設定) に保存されます。tombstone エントリーが遅延期間よりも古い場合が、次のパージジョブで復元します。

パージ遅延とパージの間隔は、cn =replica,cn=replicated suffix,cn= mapping tree,cn=config 設定エントリーのサプライヤーサーバーのレプリカエントリーに設定されます。レプリケーションのパージ設定を定義する場合、2 つの考慮事項があります。

- パージ操作は、特にサーバーが多くの削除操作を処理する場合に時間がかかりません。パージの間隔が低すぎるか、サーバーのリソースを過剰に消費してパフォーマンスに影響を及ぼす可能性があります。

- サプライヤーは、tombstone エントリーなどの変更情報を使用して、初期化後に Prime レプリケーションを実行します。コンシューマーを効果的に再初期化し、レプリケーションの競合を解決するには、変更のバックログが十分にある必要があります。パージ遅延 (tombstone エントリーの経過時間) を低く設定しすぎないでください。または、レプリケーションの競合の解決に必要な情報が失われる可能性があります。

  レプリケーショントポロジーの長いレプリケーションスケジュールよりもわずかに長くパージ遅延を設定します。たとえば、最長のレプリケーションの間隔が 24 時間の場合は、tombstone エントリーを 25 時間保持します。これにより、コンシューマーを初期化するのに十分な変更履歴が確保され、異なるサプライヤーに保存されているデータが乖離するのを防ぐことができます。

パージ設定を変更するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -x  -h supplier1.example.com

dn: cn=replica,cn=dc\=example\,dc\=com,cn=mapping tree,cn=config
changetype: modify
replace: nsDS5ReplicaTombstonePurgeInterval
nsDS5ReplicaTombstonePurgeInterval: 43200 # in seconds, 12 hours
-
changetype: modify
replace: nsDS5ReplicaPurgeDelay
nsDS5ReplicaPurgeDelay: 90000 # in seconds, 25 hours
```

注記

tombstone エントリーおよび状態情報をすぐにクリーンアップするには、非常に小さい値を *nsDS5ReplicaTombstonePurgeInterval* 属性および *nsDS5ReplicaPurgeDelay* 属性に設定します。どちらの属性も値が秒単位で設定されているため、パージ操作をほぼ即座に開始することができます。

警告

常にパージ期間を使用して、changelog から tombstone エントリーを消去します。tombstone エントリーを手動で削除しないでください。

15.14. CHANGELOG 暗号化の設定

セキュリティーを強化するために、Directory Server は changelog の暗号化をサポートします。本セクションでは、この機能を有効にする方法を説明します。

前提条件

サーバーに、ネットワークセキュリティーサービス (NSS) データベースに証明書およびキーを保存する必要があります。したがって、「Directory Server での TLS の有効化」の説明に従ってサーバーで TLS 暗号化を有効にします。

Procedure

changelog 暗号化を有効にするには、以下を実行します。

1. changelog 暗号化を有効にするサーバーを除き、以下のコマンドを入力してレプリケーショントポロジー内のすべてのインスタンスを停止します。

```
# systemctl stop dirsrv@instance_name
```

2. changelog 暗号化を有効にするサーバーで、以下を実行します。

   a. changelog をエクスポートするタスクを作成します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=replica,cn=suffix,cn=mapping tree,cn=config
changetype: modify
```

```
add: nsds5Task
nsds5Task: CL2LDIF
```

Directory Server は、エクスポートを /var/lib/dirsrv/slapd-*instance_name*/changelogdb/ ディレクトリーに保存します。

b. インスタンスを停止します。

```
# systemctl stop dirsrv@instance_name
```

c. /etc/dirsrv/slapd-*instance_name*/dse.ldif ファイルの dn: cn=changelog5,cn=config エントリーに、以下の設定を追加します。

```
nsslapd-encryptionalgorithm: AES
```

d. インスタンスを起動します。

```
# systemctl start dirsrv@instance_name
```

e. changelog をインポートするタスクを作成します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=replica,cn=suffix,cn=mapping tree,cn=config
changetype: modify
add: nsds5Task
nsds5Task: LDIF2CL
```

3. 以下のコマンドを実行して、レプリケーショントポロジー内の他のサーバー上のインスタンスをすべて起動します。

```
# systemctl start dirsrv@instance_name
```

検証

changelo が暗号化されていることを確認するには、暗号化された changelo を使用して、サーバー上で以下の手順を実行します。

1. エントリーの更新など、LDAP ディレクトリーに変更を加えます。
2. インスタンスを停止します。

```
# systemctl stop dirsrv@instance_name
```

3. 以下のコマンドを実行して、changelog の一部を表示します。

```
# dbscan -f /var/lib/dirsrv/slapd-instance_name/changelogdb/replica_name_replGen.db |
tail -50
```

changelog が暗号化されている場合は、暗号化されたデータのみが表示されます。

4. インスタンスを起動します。

```
# systemctl start dirsrv@instance_name
```

15.15. CHANGELOG の削除

changelog は、サプライヤーがコンシューマーサーバー (マルチマスターレプリケーションの場合はサプライヤー) のレプリカにこれらの変更を再生するために使用する特定のレプリカに対するすべての変更の記録です。

サプライヤーサーバーがオフラインになると、すべての変更の true レコードを保持しなくなったため、変更ログを削除することが重要です。そのため、レプリケーションのベースとして使用することはできません。ログファイルを削除して、changelog を効果的に削除できます。

15.15.1. コマンドラインを使用した Changelog の削除

サプライヤーサーバーから変更ログを削除するには、次のコマンドを実行します。

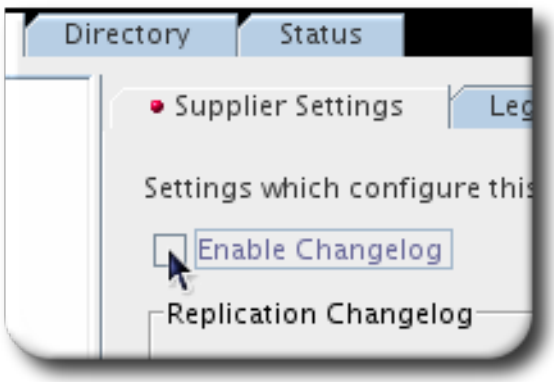
```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=changelog5,cn=config
changetype: delete
```

Directory Server は、cn=changelog 5,cn=config エントリーを削除した後に、changelog ディレクトリーのコンテンツを自動的に削除します。

15.15.2. コンソールを使用した changelog の削除

サプライヤーサーバーから変更ログを削除するには、次のコマンドを実行します。

1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のナビゲーションツリーで **Replication** フォルダーを選択し、右側のペインで **Supplier Server Settings** タブを選択します。
3. **Enable Changelog** チェックボックスの選択を解除します。



4. **Save** をクリックします。
5. Directory Server を再起動します。「コンソールを使用した Directory Server インスタンスの起動および停止」 を参照してください。
6. コンシューマーを再初期化します。「コンシューマーの初期化」 を参照してください。

15.16. レプリケーション CHANGELOG ディレクトリーの移動

特定の状況では、Directory Server レプリケーション changelog ディレクトリーを変更したい場合があります。たとえば、ディレクトリーを /var/lib/dirsrv/slapd-*instance_name*/new_changelogdb/ に変更するには、以下のコマンドを実行します。

1. 現在のディレクトリーを表示します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x \
    -b "cn=changelog5,cn=config" nsslapd-changelogdir
...
nsslapd-changelogdir: /var/lib/dirsrv/slapd-instance_name/changelogdb/
```

   ディレクトリーを移動するには、後のステップで表示されたパスが必要です。

2. 新しいパスを設定します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=changelog5,cn=config
changetype: modify
replace: nsslapd-changelogdir
nsslapd-changelogdir: /var/lib/dirsrv/slapd-instance_name/new_changelogdb/
```

3. Directory Server インスタンスを停止します。

```
# systemctl stop dirsrv@instance_name
```

4. 前のディレクトリーのコンテンツを /var/lib/dirsrv/slapd-*instance_name*/new_changelogdb/ に移動します。

```
# mv /var/lib/dirsrv/slapd-instance_name/changelogdb/ \
    /var/lib/dirsrv/slapd-instance_name/new_changelogdb/
```

5. 以前のディレクトリーを削除します。

```
# rm /var/lib/dirsrv/slapd-instance_name/changelogdb/
```

6. Directory Server インスタンスを起動します。

```
# systemctl start dirsrv@instance_name
```

15.17. レプリケーション CHANGELOG のトリム

Directory Server の changelog は、受け取ったおよび処理された変更の一覧を管理します。これには、サーバーで実行され、他のレプリケーションパートナーから、その他のディレクトリーが変更するクライアントの変更が含まれます。デフォルト設定を使用すると、Directory Server ではエントリーが自動的に削除されず、changelog は無限に増大します。削除するエントリーを制御するには、以下のパラメーターを使用します。

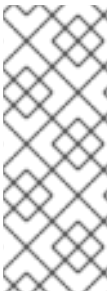
- ***nsslapd-changelogmaxage*** (推奨): このパラメーターで設定された時間を超えた場合はエントリーを削除します。
- ***nsslapd-changelogmaxentries***: レコードの合計数がこのパラメーターに設定された値を超えると、最も古いエントリーを削除します。

どのレコードも、その後に作成されたすべてのレコードも、トポロジー内のすべてのサーバーに正常にレプリケートされるまで、changelog に残ります。たとえば、Directory Server マスターがトポロジーから削除されましたが、レプリカ更新ベクター (RUV) が削除されていなかった場合などに発生します。

15.17.1. レプリケーション changelog のトリムの有効化

レプリケーション changelog のトリムを有効にするには、7 日 (7d) より古いエントリーを自動的に削除します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=changelog5,cn=config
changetype: modify
replace: nsslapd-changelogmaxage
nsslapd-changelogmaxage: 7d
```



注記

Red Hat は、*nsslapd-changelogmaxentries* の最大エントリー数ではなく、*nsslapd-changelogmaxage* で最大期間を設定することを推奨します。*nsslapd-changelogmaxage* に設定された時間は、*nsDS5ReplicaPurgeDelay* で設定したレプリケーションパージ遅延と一致する必要があります。*nsDS5ReplicaPurgeDelay* の詳細は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス のパラメーターの説明を参照してください』。

15.17.2. 大きな changelog のサイズを手動で縮小

レプリケーション changelog をトリムして、データベースのサイズが大きい場合に、短い期間で changelog サイズを手動で縮小します。

1. changelog サイズを縮小した後にパラメーターをリセットできるようにするには、対応するパラメーターの現在の値を表示します。以下に例を示します。

```
# ldapsearch -x -D 'cn=Directory Manager' -W -b "cn=changelog5,cn=config" \
    nsslapd-changelogmaxage nsslapd-changelogcompactdb-interval \
    nsslapd-changelogtrim-interval nsslapd-changelogmaxage

dn: cn=changelog5,cn=config
nsslapd-changelogmaxage: 7d
nsslapd-changelogcompactdb-interval: 2592000
nsslapd-changelogtrim-interval: 300
```

出力に表示されないパラメーターには設定されず、Directory Server はデフォルト値を使用します。デフォルト値は、Red『Hat Directory Server の設定、コマンド、およびファイルリファレンス のパラメーターの説明を参照してください』。

2. 以下のパラメーターの値を一時的に減らします。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=changelog5,cn=config
changetype: modify
replace: nsslapd-changelogmaxage
```

```
nsslapd-changelogmaxage: 3d
-
replace: nsslapd-changelogtrim-interval
nsslapd-changelogtrim-interval: 30
-
replace: nsslapd-changelogcompactdb-interval
nsslapd-changelogcompactdb-interval: 300
```

この設定により、Directory Server は、次の 30 秒で (*nsslapd-changelogtrim-interval*)、3 日 (*nsslapd-changelogmaxage*) よりも古い changelog エントリーを削除します。

3. Directory Server インスタンスを再起動して、*nsslapd-changelogcompactdb-interval* パラメーターの新しい値が有効になります。

```
# systemctl restart dirsrv@instance
```

次のデータベースの更新後、*nsslapd-changelogcompactdb-interval* パラメーターで設定した時間間隔内でデータベースが自動的に圧縮されます。

4. 更新されたパラメーターを以前の値にリセットします。以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=changelog5,cn=config
changetype: modify
replace: nsslapd-changelogmaxage
nsslapd-changelogmaxage: 7d
-
replace: nsslapd-changelogtrim-interval
nsslapd-changelogtrim-interval: 300
-
replace: nsslapd-changelogcompactdb-interval
nsslapd-changelogcompactdb-interval: 2592000
```

重要

パフォーマンス上の理由から、短い間隔設定を永続的に使用しないでください。

5. Directory Server インスタンスを再起動します。

```
# systemctl restart dirsrv@instance
```

15.18. コンシューマーの初期化

レプリカ合意が作成されたら、コンシューマーを初期化する必要があります。つまり、データはサプライヤーサーバーからコンシューマーサーバーに物理的にコピーされる必要があります。

注記

レプリケーションは、コンシューマーが初期化されるまで 開始されません。

- 「コンシューマーの初期化のタイミング」

- 「コンソールを使用したオンラインコンシューマーの初期化」
- 「コマンドラインを使用したコンシューマーオンラインの初期化」
- 「コマンドラインを使用した手動コンシューマーの初期化」

注記

大規模なデータベースでは、*nsslapd-idletimeout* 設定を十分な期間（または無制限の時間）に設定する必要があります。これにより、操作がタイムアウトする前にデータベース全体を初期化できるようにする必要があります。または、サプライヤーバインド DN エントリーの *nsIdleTimeout* 設定は、グローバル設定を変更しなくても、オンライン初期化操作を完了できるように設定することも可能です。

15.18.1. コンシューマーの初期化のタイミング

コンシューマーの初期化には、サプライヤーサーバーからコンシューマーサーバーにデータをコピーする必要があります。サブツリーがコンシューマーに物理的に配置されたら、サプライヤーサーバーはコンシューマーサーバーへの更新操作のリプレイを開始できます。

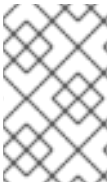
通常の操作では、コンシューマーは再初期化する必要はありません。ただし、サプライヤーのデータとコンシューマーのデータの間に大きな不一致がある可能性があり、コンシューマーを再初期化します。たとえば、サプライヤーサーバーのデータがバックアップから復元されると、そのサーバーによって提供されたすべてのコンシューマーを再初期化する必要があります。別の例として、サプライヤーがコンシューマーと長期間 (例: 1週間) にわたって通信できない場合、サプライヤーはコンシューマーが更新できないほど古くなっている可能性があると判断し、再初期化する必要があります。

コンシューマーは、コンソールを使用してオンラインで、またはコマンドラインを使用して手動で初期化できます。コンソールを使用したオンラインコンシューマーの初期化は、少数のコンシューマーを初期化する効果的な方法です。ただし、各レプリカは順番に初期化されるため、この方法は大量のレプリカを初期化するのに適していません。オンラインコンシューマーの初期化は、コンシューマーがサプライヤーサーバーでのレプリカ合意の設定の一部として初期化される時に使用する方法です。

コマンドラインを使用した手動コンシューマーの初期化は、1つの LDIF ファイルから多数のコンシューマーを初期化するより効果的な方法です。

15.18.2. コンソールを使用したオンラインコンシューマーの初期化

コンソールを使用したオンラインコンシューマーの初期化は、コンシューマーを初期化または再初期化する最も簡単な方法です。ただし、リンクが遅い場合は、このプロセスに非常に時間がかかる場合があります。また、コマンドラインを使用して手動によるコンシューマーの初期化をより効率化することがあります。詳細は、「コマンドラインを使用した手動コンシューマーの初期化」 を参照してください。



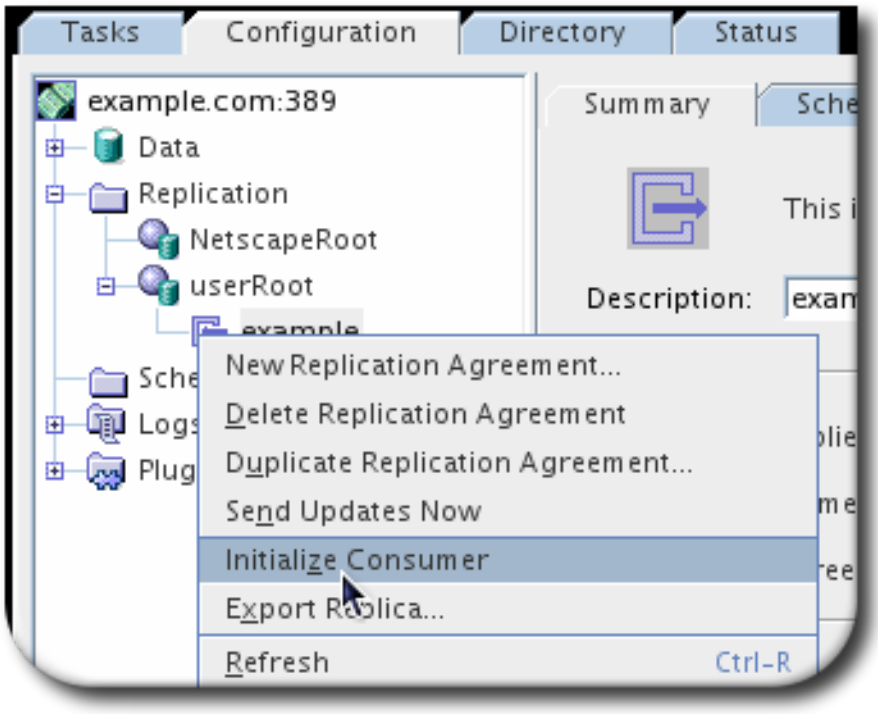
注記

コンシューマーサーバーがオンラインコンシューマー作成方法を使用して初期化されている場合、レプリカ上のすべての操作（検索を含む）は、初期化プロセスが完了するまでサプライヤーサーバーと呼ばれます。

オンラインでコンシューマーを初期化するか、または再初期化するには、以下を実行します。

1. レプリカ合意を作成します。
2. サプライヤーサーバーの Directory Server コンソールで、**Configuration** タブを選択します。

3. **Replication** フォルダーを展開し、複製されたデータベースを展開します。レプリカ合意を右クリックし、ポップアップメニューから **Initialize Consumer** を選択します。



コンシューマーのレプリカにすでに保存された情報を削除するというメッセージが表示されます。

4. 確認ボックスの **Yes** をクリックします。

オンラインコンシューマーの初期化がすぐに開始します。オンラインコンシューマーの初期化のステータスを確認するには、**Status** ボックスの **Summary** タブを開きます。オンラインコンシューマーの初期化が進行中であれば、レプリカが初期化されていることが表示されます。

重要

マルチマスターレプリケーションの場合は、コンシューマーが 1 つのサプライヤーによって 1 度だけ 初期化されていることを確認します。レプリケーションのステータスを確認する際には、コンシューマーの初期化に使用された適切なサプライヤーでレプリカ合意のエントリーを確認してください。

このウィンドウを更新するには、ナビゲーションツリーで複製されたデータベースアイコンを右クリックし、レプリカ合意 の更新 を選択します。オンラインコンシューマーの初期化が完了すると、ステータスが変更されてこれが反映されます。

レプリケーションおよび初期化ステータスの監視に関する詳細は、「レプリケーションステータスの監視」 を参照してください。

15.18.3. コマンドラインを使用したコンシューマーオンラインの初期化

オンラインコンシューマーの初期化は、コマンドラインで ***nsds5BeginReplicaRefresh*** 属性をレプリカ合意エントリーに追加することで実行できます。この属性はデフォルトでは存在しないため、コンシューマーの初期化が完了すると自動的に削除されます。

1. コンシューマーを初期化するサプライヤーサーバーでレプリカ合意の DN を検索します。以下に例を示します。

```
# ldapsearch -h supplier1.example.com -p 389 -D "cn=Directory Manager" -W -s sub
    -b cn=config "(objectclass=nsds5ReplicationAgreement)"
```

このコマンドは、サプライヤーに設定されたすべてのレプリカ合意を LDIF 形式で返します。初期化されるコンシューマーとのレプリカ合意の DN を取得します。これは、編集されるレプリカ合意です。

2. レプリカ合意を編集し、*nsds5BeginReplicaRefresh* 属性を追加します。

```
# ldapmodify -D "cn=Directory Manager" -W -x -h supplier1.example.com

dn: cn=ExampleAgreement,cn=replica,cn=dc\=example\,dc\=com,cn=mapping
tree,cn=config
changetype: modify
replace: nsds5BeginReplicaRefresh
nsds5BeginReplicaRefresh: start
```

ldapmodify は入力を要求しません。LDIF ステートメントに入力するだけで、LDIF ステートメントが完了すると 2 回到達されます。Ctrl+C を押して ldapmodify ユーティリティーを閉じます。

初期化ステータスを確認するには、レプリカ合意エントリーに対して ldapsearch を実行します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -s base -b
'cn=ExampleAgreement,cn=dc\=example\,dc\=com,cn=mapping tree,cn=config'
'(objectclass=*)'
```

nsds5BeginReplicaRefresh 属性が存在する場合は、初期化が進行中です。初期化が完了すると、属性 *nsds5ReplicaLastInitStatus* にステータスが表示されます。初期化に成功すると、*nsds5ReplicaLastInitStatus* の値は Total update succeeded になります。初期化に成功しなかった場合、この属性はエラーに関する情報を表示します。追加の情報については、サプライヤーとコンシューマーの両方にエラーログを確認します。

重要

マルチマスターレプリケーションの場合は、コンシューマーが 1 つのサプライヤーによって 1 度だけ 初期化されていることを確認します。レプリケーションのステータスを確認する際には、コンシューマーの初期化に使用された適切なサプライヤーでレプリカ合意のエントリーを確認してください。

レプリケーションの監視属性は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス を参照してください』。

15.18.4. コマンドラインを使用した手動コンシューマーの初期化

コマンドラインを使用した手動コンシューマーの初期化は、非常に多くのエントリーを複製するサイト向けのコンシューマーの初期化の最速な方法です。ただし、手動によるコンシューマーの初期化プロセスは、オンラインコンシューマーの初期化プロセスよりも複雑です。Red Hat は、パフォーマンスに関する懸念事項によりオンラインプロセスが不適切な場合に常に手動プロセスの使用を推奨します。

サーバーを手動で初期化または再初期化するには、以下の 3 つの手順を実行します。

1. レプリカ合意を作成します。

2. サプライヤーサーバーのレプリカを LDIF ファイルにエクスポートします。

   「レプリカから LDIF へのエクスポート」を参照してください。

3. サプライヤーレプリカの内容を持つ LDIF ファイルをコンシューマーサーバーにインポートします。

   「LDIF ファイルのコンシューマーサーバーへのインポート」を参照してください。

15.18.4.1. レプリカから LDIF へのエクスポート

レプリカデータベースを LDIF に変換するには、以下の 3 つの方法があります。

- レプリカ合意の作成時に、Replication Agreement Wizard の**Initialize Consumer** ダイアログボックスで **Create consumer initialization file** を選択します。
- Directory Server Console から、**Replication** フォルダーでレプリカ合意を右クリックし、ポップアップメニューから **Create LDIF File** を選択します。
- 「コマンドラインを使用した LDIF へのデータベースのエクスポート」で説明されているように、export コマンドを使用してコマンドラインから実行します。コマンドラインツールで LDIF へのエクスポートは、データベースをレプリカとしてエクスポートするオプションを使用する必要があります。つまり、エクスポートされた LDIF には、LDIF のインポート時にコンシューマーを初期化する適切なエントリーが含まれていることを意味します。

  db2ldif スクリプトおよび **db2ldif.pl** スクリプトの場合、これは **-r** オプションになります。以下に例を示します。

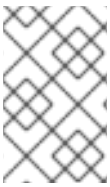
  ```
  # db2ldif -r -n database1 -a /export/output.ldif
  ```

  cn=export,cn=tasks,cn=config エントリーの場合、これは ***nsExportReplica*** 属性です。

  ```
  #ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

  dn: cn=example export,cn=export,cn=tasks,cn=config
  changetype: add
  objectclass: extensibleObject
  cn: example export
  nsInstance: userRoot
  nsFilename: /home/files/example.ldif
  nsExportReplica: true
  ```

15.18.4.2. LDIF ファイルのコンシューマーサーバーへのインポート

Directory Server コンソールのインポート機能を使用するか、ldif2db スクリプトまたは **ldif2db.pl** スクリプトを使用して、サプライヤーレプリカの内容が含まれる LDIF ファイルをコンシューマーサーバーにインポートします。両方のインポート方法は「コマンドラインからのインポート」で説明されています。



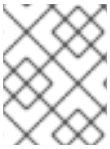
注記

ldif2db.pl スクリプトでは、LDIF ファイルインポート操作にはサーバーの再起動は必要ありません。コマンドラインスクリプトの詳細は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス を参照してください』。

15.19. レプリケーション更新の強制

通常のメンテナンスでレプリケーションに関連する Directory Server が停止している場合は、オンラインに戻るとすぐに更新する必要があります。マルチマスター環境でのサプライヤーでは、複数マスターセットの他のサプライヤーがディレクトリー情報を更新する必要があります。他の場合は、ハブまたは専用コンシューマーがメンテナンス用にオフラインになると、オンラインに戻った時に、サプライヤーサーバーで更新する必要があります。

レプリカ合意が、サプライヤーサーバーおよびコンシューマーサーバーを常に同期するように構成されている場合でも、5 分間、オフラインで稼働していたサーバーを最新のバックアップにするだけでは不十分です。**Sync** オプションの **Always Keep** は、サーバーが処理する更新操作ごとにレプリケーション操作を生成することを意味します。ただし、コンシューマーがオフラインであるため、このレプリケーション操作を実行できないと、操作は 10 分後にタイムアウトします。



注記

本セクションで説明されている手順は、レプリケーションがすでに設定されており、コンシューマーが初期化されている場合にのみ使用できます。

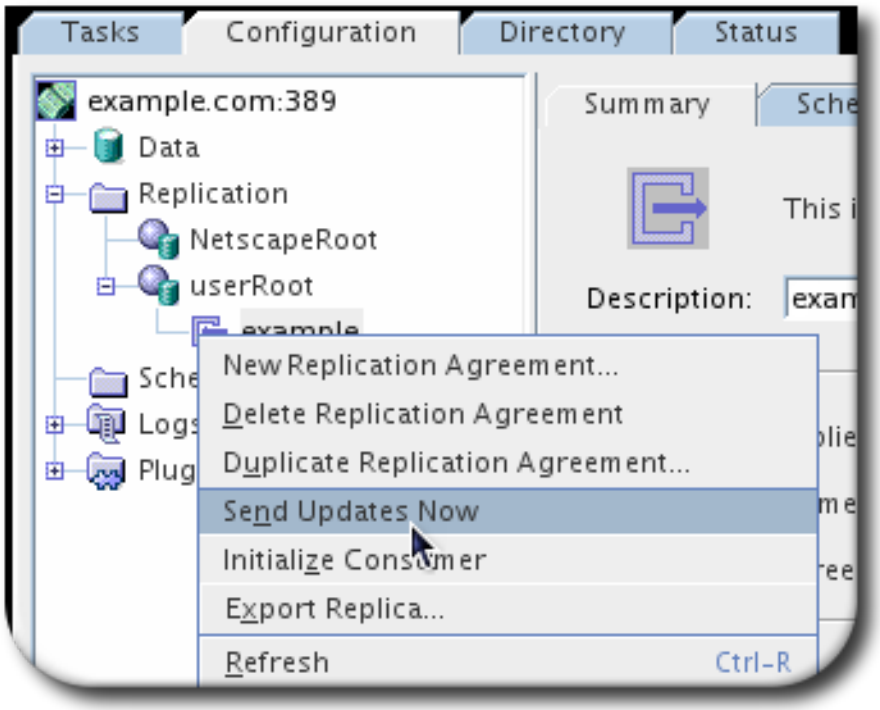
サーバーがオンラインに戻った直後にディレクトリー情報を同期するようにするには、ディレクトリー情報の参照コピーまたはカスタマイズ可能なスクリプトを保持するサプライヤーサーバーの Directory Server Console を使用します。

15.19.1. コンソールからのレプリケーション更新の強制

一定期間後に、マルチマスターレプリケーション設定のコンシューマーまたはサプライヤーがオンラインに戻ると、レプリケーションの更新がすぐに送信されるようにするには、ディレクトリー情報の最新バージョンを保持するサプライヤーサーバーで以下を実行します。

1. Directory Server コンソールで、**Configuration** タブをクリックし、**Replication** フォルダーおよびデータベースノードを展開し、更新するレプリカに対応するレプリカ合意を選択します。

2. レプリカ合意を右クリックし、ドロップダウンリストから **Send Updates Now** を選択します。



これにより、更新が必要な情報を保持するサーバーにレプリケーションが開始されます。

15.19.2. コマンドラインでのレプリケーション更新の強制

レプリケーションの更新を強制するには、レプリカ合意を無効にして再度有効にします。詳細は、「レプリカ合意の無効化および再有効化」 を参照してください。

15.20. TLS 上のレプリケーション

セキュリティー上の理由から、レプリケーションに関連する Directory Server は、すべてのレプリケーション操作が TLS 接続で実行されるように設定する必要があります。TLS でレプリケーションを使用するには、以下を実行します。

- サプライヤーサーバーとコンシューマーサーバーの両方が TLS を使用するように設定します。
- コンシューマーサーバーが、サプライヤーサーバーの証明書を サプライヤー DN として認識するように設定します。これは、簡易認証ではなく TLS クライアント認証のみを使用します。

これらの手順は、「TLS の有効化」 で説明しています。

属性の暗号化が有効な場合は、レプリケーションにセキュアな接続が必要になります。

注記

証明書ベースの認証で TLS 上で構成されたレプリケーションは、サプライヤーの証明書がサーバー証明書としてのみ動作し、TLS ハンドシェイク中にクライアントに対応していないと失敗します。証明書ベースの認証でのレプリケーションでは、リモートサーバーへの認証に Directory Server のサーバー証明書を使用します。

certutil を使用して証明書署名要求 (CSR) を生成する場合は、*--nsCertType=sslClient,sslServer* オプションをコマンドに渡し、必要な証明書を設定します。

サーバーが TLS を使用するように設定されている場合、Replication Agreement Wizardでレプリケーションの TLS 接続を設定します。**Source** および **Destination** は、サプライヤーとコンシューマーの間でバインドする方法を設定します。これは TLS が設定される場所です。

レプリケーションに TLS を使用する方法は 2 つあります。

- **SSL クライアント認証** を選択します。

  TLS クライアント認証では、サプライヤーサーバーおよびコンシューマーサーバーは証明書を使用して相互に対して認証します。

- **Simple Authentication** を選択します。

  簡易認証では、サプライヤーサーバーおよびコンシューマーサーバーはバインド DN およびパスワードを使用して相互に対して認証を行います。これは、提供される Replication Agreement Wizard テキストフィールドに指定されます。簡易認証はセキュアなチャンネルで実行されますが、証明書はありません。

注記

シンプルなパスワード認証(「セキュアなバインドの要求」)にセキュアなバインドが必要な場合は、セキュアな接続で行われる場合を除き、レプリケーション操作は失敗します。セキュアな接続（TLS および Start TLS 接続または SASL 認証）の使用が推奨されます。

レプリカ合意が作成されると、LDAP および LDAPS 接続が異なるポートを使用するため、接続タイプ（TLS または非 TLS）は変更できません。接続タイプを変更するには、レプリカ合意を再作成します。

また、Directory Server インスタンスが TLS 上で実行されるように設定されている場合でも、コンシューマーに一覧表示されるポートは TLS 以外のポートになります。このポート番号は、コンソールで Directory Server インスタンスを識別するためにのみ使用されます。これは、レプリケーションに使用される実際のポート番号またはプロトコルを指定しません。

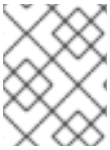
15.21. レプリケーションのタイムアウト期間の設定

ディレクトリーに更新を送信するには、サプライヤーには、コンシューマーへの排他的な接続が必要です。「マルチマスターレプリケーションにおけるコンシューマーの独占を防ぐ」で説明したように、コンシューマーへの接続を試みるサプライヤーに待機時間を設定し、コンシューマーが別のサプライヤーと関連付けられている間にサプライヤーがハングしないようにすることができます。

また、サプライヤーにタイムアウト期間を設定し、低速な接続や破損した接続で更新の送信をやり直してもコンシューマーに接続し続けないようにすることもできます。

タイムアウト期間を設定する属性は 2 つあります。

- ***nsDS5ReplicaTimeout*** は、レプリケーション操作がコンシューマーからの応答を待ってから、タイムアウトして失敗するまでの秒数を設定します。最適な数値を設定するには、アクセスログでレプリケーション処理にかかる平均時間を確認し、それに合わせてタイムアウト期間を設定します。
- ***nsDS5DebugReplicaTimeout*** は、デバッグロギングが有効な場合にレプリケーション操作のタイムアウト期間を設定します。デバッグロギングではディレクトリー操作が遅くなる可能性があるため、この設定は ***nsDS5ReplicaTimeout*** 設定よりも大幅に高くなる可能性があります。この属性は任意で、このパラメーターが適用されるエラーログレベルを設定できます。デフォルトはレプリケーションのデバッグ (8192) です。



注記

タイムアウトの期間は、最大 32 ビットの整数 (秒単位) に制限され、24.8 日に変換されます。

これらの属性はいずれも複製された接尾辞の設定で設定されます。たとえば、以下は、ou=People 接尾辞のタイムアウト期間を設定します。

```
# ldapmodify -D "cn=Directory Manager" -W -x

dn: cn=replica,cn="ou=People,dc=example,dc=com",cn=mapping tree,cn=config
changetype: modify
add: nsDS5ReplicaTimeout
nsDS5ReplicaTimeout: 600
add: nsDS5DebugReplicaTimeout
nsDS5DebugReplicaTimeout: 6000
```

15.22. 管理サーバーのフェイルオーバー用の O=NETSCAPEROOT の複製

レプリケーションは通常、ディレクトリーデータを分散するために Directory Server ユーザーデータベース間で行われますが、レプリケーションを使用して Administration Server データベース(o=NetscapeRoot)のフェイルオーバーサポートを提供することもできます。

1. **最初の Directory Server インスタンスをインストールして設定します。**

   setup-ds-admin.pl スクリプトには、inf を参照する -f オプションがあります。inf を使用して、ConfigFile パラメーターを使用して LDIF ファイルをインポートでき、LDIF ファイルはデータベース、接尾辞、およびレプリケーションエントリーを作成できます。（そのファイルの詳細は『『Red Hat Directory Server インストールガイド』を参照してください』。）

   ```
   # setup-ds-admin.pl -f /tmp/server1.inf
   ```

   server1 の o=NetscapeRoot データベースをマルチマスターサプライヤーレプリカとして設定するには、inf ファイルで以下のステートメント を使用します。

   ```
   [slapd]
   ...
   ConfigFile = repluser.ldif 例15.4「サプライヤーバインド DN エントリーの例」
   ConfigFile = changelog.ldif 例15.1「Changelog エントリーの例」
   ConfigFile = replica.ldif 例15.2「サプライヤーレプリカエントリーの例」
   ConfigFile = replagreement.ldif 例15.3「レプリカ合意エントリーの例」
   ...
   ```

2. **2 番目の Directory Server インスタンスをインストールして設定します。2 番目のサーバー server2.example.com の場合は、setup-ds.pl コマンドを使用して、ローカルの管理サーバーをインストールせずに Directory Server インスタンスをインストールします。**

   ```
   # setup-ds.pl -f /tmp/server2.inf
   ```

   server2 では、inf ファイルを使用して、server2 で o=NetscapeRoot データベースをマルチマスターサプライヤーレプリカとして作成および設定します。

   ```
   [slapd]
   ...
   ConfigFile = netscaperootdb.ldif 「コマンドラインでのルート接尾辞およびサブ接尾辞の作成」
   ConfigFile = repluser.ldif 例15.4「サプライヤーバインド DN エントリーの例」
   ConfigFile = changelog.ldif 例15.1「Changelog エントリーの例」
   ConfigFile = replica.ldif 例15.2「サプライヤーレプリカエントリーの例」
   ConfigFile = replagreement.ldif 例15.3「レプリカ合意エントリーの例」
   ...
   ```

3. **server 1 から server 2 で o=NetscapeRoot データベースを初期化します。*nsds5replicarefresh* 属性を server1 のレプリカ合意に追加します。**

   ```
   # ldapmodify -D "cn=Directory Manager" -W -x -h supplier1.example.com

   dn: cn=ExampleAgreement1,cn=replica,cn=o=NetscapeRoot,cn=mapping
   tree,cn=config
   ```

```
changetype: modify
replace: nsds5beginreplicarefresh
nsds5beginreplicarefresh: start
```

4. register-ds-admin.pl を実行して server2 にローカル管理サーバーを作成し、server2 の設定ディレクトリーを server1 から独自の o=NetscapeRoot データベースに切り替えます。

```
# register-ds-admin.pl
```

5. 以下のアクセス制御命令(ACI)を server2 に追加し、Configuration Administrators Group、サーバーインスタンスエントリー SIE グループ、および admin ユーザーのメンバーが server2 に属するサフィックスで実行できます。たとえば、dc =example,dc=com 接尾辞で実行する場合は、以下を入力します。

```
# ldapmodify -D "cn=Directory Manager" -W -x -h server2.example.com
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="*")(version 3.0; acl "Configuration Administrators Group";
 allow (all) groupdn="ldap:///cn=Configuration Administrators,ou=Groups,
 ou=TopologyManagement,o=NetscapeRoot";)
-
add: aci
aci: (targetattr="*")(version 3.0; acl "Configuration Administrator";
 allow (all) userdn="ldap:///uid=admin,
 ou=Administrators,ou=TopologyManagement,o=NetscapeRoot";)
-
add: aci
aci: (targetattr = "*")(version 3.0; acl "SIE Group";  allow (all) groupdn =
 "ldap:///cn=slapd-instance,cn=Red Hat Directory Server,cn=Server Group,
 cn=machine_name,ou=example.com,o=NetscapeRoot";)
```

6. server2 で PTA プラグインを無効にし、o=NetscapeRoot の管理者ユーザーのバインド操作を server1 に渡さないようにします。

「Directory Server コンソールでプラグインの有効化」を参照してください。

15.23. RETRO CHANGELOG プラグインの使用

Retro Changelog プラグインは、Directory Server 4.x に実装された changelog との互換性を維持するように Directory Server を設定します。

注記

Directory Server 4.x 形式の changelog に依存するディレクトリークライアントの changelog を維持する必要がある場合は、Retrolog プラグインのみを有効にします。

Retro Changelog プラグインを使用するには、Directory Server インスタンスをシングルマスターレプリカとして設定する必要があります。

Directory Server が Retro Changelog を維持するように設定されていると、この changelog は特別な接尾辞 cn=changelog の下に別のデータベースに保存されます。

Retro Changelog は単一レベルのエントリーで構成されます。changelog の各エントリーには、オブ

ジェクトクラス *changeLogEntry* があります。changelog エントリーで可能な属性の一覧は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス』 の『changelog 属性』 セクションを参照してください。

15.23.1. Retro Changelog プラグインの有効化

Retro Changelog プラグインの設定情報は、**dse.ldif** の **cn=Retro Changelog Plugin,cn=plugins,cn=config** エントリーに保存されます。コマンドラインから Retro Changelog プラグインを有効にするには、以下を実行します。

1. 以下の LDIF 更新ステートメントが含まれる 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ーにインポートします。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -f
retro.ldif
```

3. サービスを再起動します。

   サーバーの再起動に関する情報は、「Directory Server インスタンスの起動および停止」 を参照してください。

Retro Changelog は、特別な接尾辞 **cn=changelog** 下のディレクトリーツリーに作成されます。

Directory Server コンソールから Retro Changelog プラグインを有効にする手順は、すべての Directory Server プラグインと同じです。詳細は、「Directory Server コンソールでプラグインの有効化」 を参照してください。

15.23.2. Retro Changelog のトリム

nsslapd-changelogmaxage パラメーターで設定したレコードの最大年齢を下げ、***nsslapd-changelog-trim-interval*** で設定した次のトリミング間隔を実行すると、Retro Changelog のサイズが自動的に縮小されます。

たとえば、Retro Changelog のレコードの最大期間を 2 日に設定するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=Retro Changelog Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-changelogmaxage
nsslapd-changelogmaxage: 2d
```

15.23.3. Retro Changelog の検索および変更

Changelog は検索操作に対応し、**(&(changeNumber>=X)(changeNumber<=Y))** 形式のフィルターが含まれる検索に対して最適化されます。

一般的なルールとして、changelog のエントリーの追加操作や変更操作は実行しないでください。ただし、エントリーを削除して changelog のサイズをトリミングすることができます。デフォルトのアクセス制御ポリシーを変更するには、Retro Changelog エントリーのみを変更します。

15.23.4. Retro Changelog およびアクセス制御ポリシーの見直し

Retro Changelog が作成されると、デフォルトで以下のアクセス制御ルールが適用されます。

- すべての認証されたユーザー (**userdn=anyone**。ただし。**userdn=all** である匿名アクセスと混同しないようにしてください) には、Retro Changelog のトップエントリー **cn=changelog** に対して、読み取り、検索、および比較の権限が与えられます。
- Directory Manager に暗黙的ではなく、書き込みアクセスと削除のアクセスは付与されません。

changelog エントリーにはパスワードなどの機密情報の変更が含まれる可能性があるため、匿名ユーザーに読み取りアクセスを付与しないでください。この情報にアクセスできるのは認証アプリケーションやユーザーのみです。

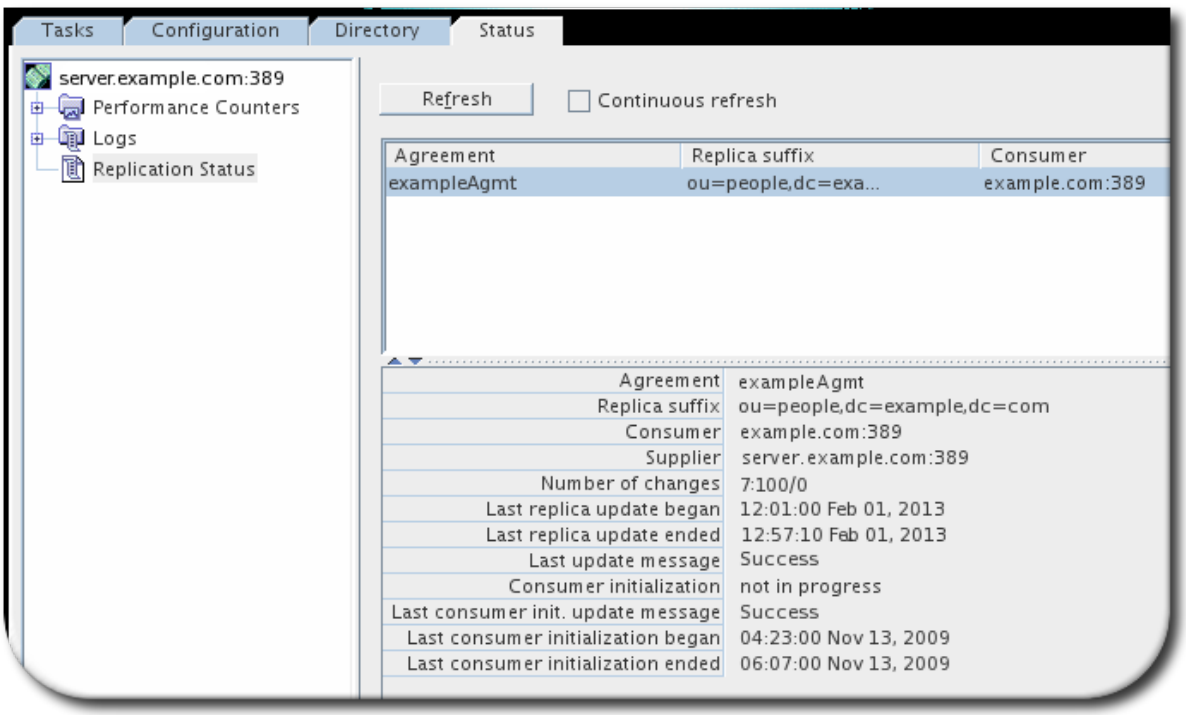
Retro Changelog に適用されるデフォルトのアクセス制御ポリシーを変更するには、**cn=changelog** エントリーの *aci* 属性を変更します。

15.24. レプリケーションステータスの監視

レプリケーションのステータスは、Directory Server Console、Red Hat Administration Express(「Admin Express からのレプリケーションの監視」)、またはコマンドラインから表示できます。

15.24.1. コンソールからのレプリケーションのステータスの監視

1. **Status** タブを選択し、左側のナビゲーションツリーで **Replication Status** を選択します。



右側のペインに、このサーバーに設定した各レプリカ合意に関する情報が含まれるテーブルが表示されます。

2. **Refresh** をクリックして、タブの内容を更新します。

   表示されるステータス情報は、表15.1「Directory Server コンソールのレプリケーションステータス」 で説明されています。

表15.1 Directory Server コンソールのレプリケーションステータス

テーブルヘッダー	詳細
合意	レプリカ合意の名前。
レプリカ接尾辞	複製される接尾辞。
supplier	合意内のサプライヤーサーバー。
コンシューマー	この合意のコンシューマーサーバー
変更数	サーバーの開始以降、このレプリカに送信された変更を示す比率。この値の形式は replica_id:changes_sent/changes_skipped になります。そのため、レプリカ ID が 7 で、100 個の変更が送信され、変更が省略されない場合、変更の数の値は 7:100/0 になります。
Last replica update began	最新のレプリケーション更新が開始された時間。
Last replica update ended	最新のレプリケーション更新が完了した時間。
Last update message	最新のレプリケーション更新のステータス。
コンシューマーの初期化	コンシューマーの初期化の現在のステータス（進捗中または否定）。
最後のコンシューマーの初期化更新メッセージ	コンシューマーの最後の初期化のステータス。
最後のコンシューマーの初期化が開始	コンシューマーレプリカの初期化が開始された時間。
最後のコンシューマーの初期化が終了しました	コンシューマーレプリカの初期化が終了したとき。

15.24.2. Admin Express からのレプリケーションの監視

Admin Express には、リアルタイムでレプリケーションステータスを監視するオプションがあります。つまり、更新の数、最新の更新が送信された時刻、エラー、成功メッセージ、レプリケーションスケジュール、および複製されたディレクトリーサフィックスなどを示すオプションがあります。レプリケーションステータスをチェックする他の方法とは異なり、Admin Express **Replication Status** ページには、レプリケーションのリアルタイムステータス（進行中更新、現在の変更シーケンス番号、サプライヤーで変更が行われ、その変更がコンシューマーに送信される時点間の遅延を含む）が表示されます。

監視レプリケーションは、監視するサーバーと、ステータスページに追加するサプライヤーおよびコンシューマーレプリカを指定する単純な設定ファイルを使用して設定されます。

Admin Express を使用してレプリケーションステータスをモニターしようとする際には、以下の 2 つの点を覚えておいてください。

- **Replication Status** ページはサプライヤーサーバーでのみ利用できます。（他のタイプのレプリカ用に開くこともできます。利用可能な情報がなく、このメッセージ にはマスターではないか、レプリカ合意がないという メッセージがあります。）

- 設定ファイルは、管理サーバーがアクセスできるディレクトリーにあり、このファイルは Administration Server ユーザーが読み取り可能でなければなりません。デフォルトでは、ユーザーは **dirsrv** です。

  ユーザーは **console.conf** ファイルに設定されます。ユーザーを確認するには、**grep** を使用して値を返します。

  ```
  # grep \^User /etc/dirsrv/admin-serv/console.conf
  ```

  設定ファイルは、管理サーバーユーザーが読み取り可能で、他のユーザーも行わないようにする必要があります。したがって、ファイルのパーミッションをリセットすることを検討してください。

  ```
  # chmod 0400 filename
  ```

Admin Express でレプリケーションの in-progress ステータスを表示するには、以下を実行します。

1. 設定ファイルを作成します。設定ファイルでは、レプリケーションを監視するすべてのサーバーを一覧表示し、ホスト名または IPv4 アドレスまたは IPv6 アドレス、ポート、使用するバインド認証情報、次にエイリアスおよび時間遅延の色の任意設定を提供します。

   ```
   #Configuration File for Monitoring Replication Using Admin Express
    [connection] Required. Gives the server host (or IPv4 or IPv6 address), port, supplier bind
   DN, and password.
   host1.example.com:389:cn=replication manager:mypassword
   host2.example.com:3891:cn=replication manager:altpassword

   [alias] Optional. Gives a friendly-name alias to the servers and consumers.
   M1 = host1.example.com:389
   M2 = host2.example.com:3891
   C1 = host3.example.com:3892
   C2 = host4.example.com:3890

   [color] Optional. Sets the color for the time lag boxes.
   0 = #ccffcc
   5 = #FFFFCC
   60 = #FFCCCC
   ```

   設定ファイルは、管理サーバーがアクセスできるディレクトリーにあり、このファイルは Administration Server ユーザーが読み取り可能でなければなりません。デフォルトでは、ユーザーは **dirsrv** です。

   ユーザーは **console.conf** ファイルに設定されます。ユーザーを確認するには、**grep** を使用して値を返します。

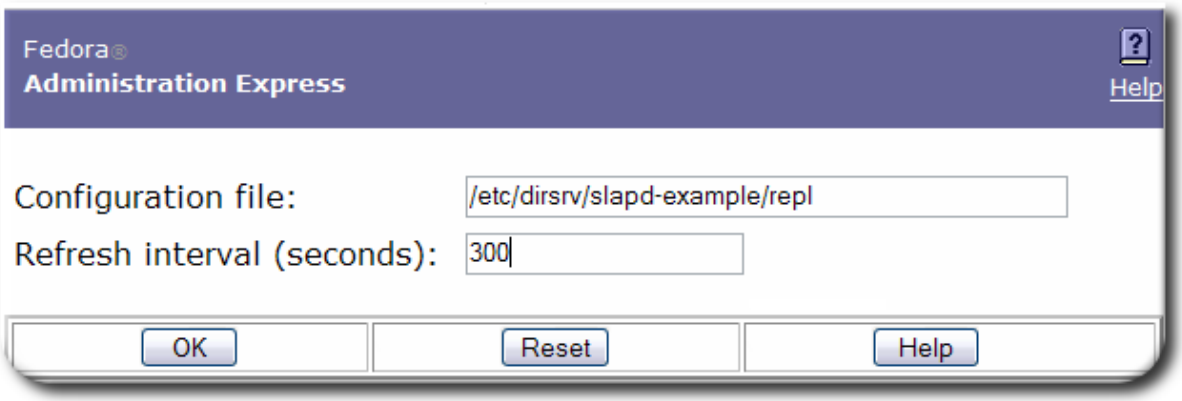
```
# grep \^User /etc/dirsrv/admin-serv/console.conf
```

設定ファイルは、管理サーバーユーザーが読み取り可能で、他のユーザーも行わないようにする必要があります。したがって、ファイルのパーミッションをリセットすることを検討してください。

```
# chmod 0400 filename
```

2. Administration Server Web ページで **Admin Express** リンクをクリックし、ログインします。
3. サプライヤーサーバー 名の横にある **Replication Status** リンクをクリックします。
4. **Configuration file** フィールドに設定ファイルへのパスを入力します。また、更新レートを設定します。これは、レプリケーションステータスページの更新頻度です。デフォルトは 300 秒です。

   図15.5 レプリケーションステータスの表示



5. **OK** をクリックします。

Replication Status ページには、設定ファイルに記載されているすべてのコンシューマーに更新を送信するステータスが表示されます。

図15.6 レプリケーションステータスの表示

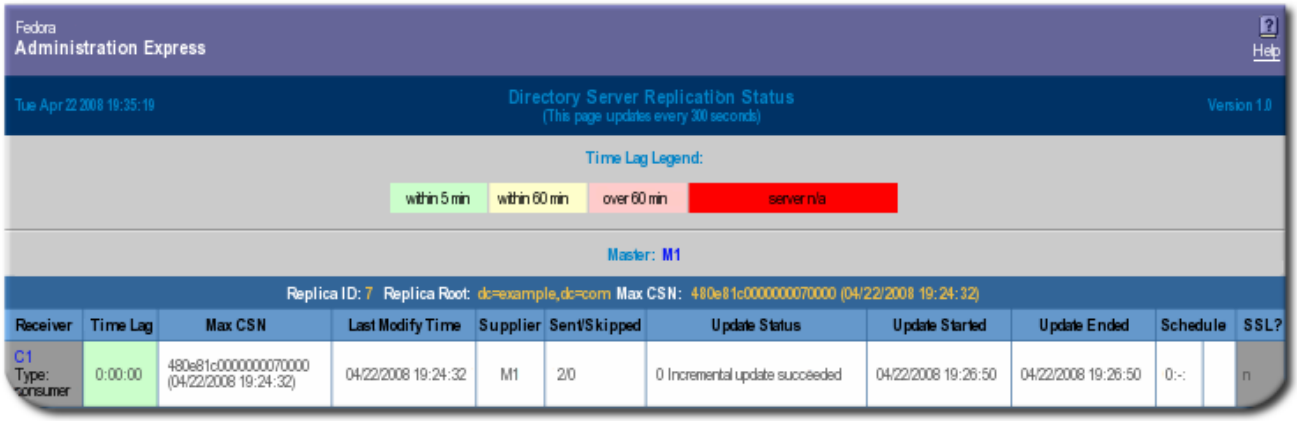


表	詳細

表	詳細
テーブルヘッダー	テーブルヘッダーには、サプライヤーレプリカのレプリカ ID、複製された接尾辞 root（dc **=example,dc=com**など）、およびサプライヤー上の最大変更状態番号(CSN)が表示されます。（CSN はサプライヤーの最新変更の ID で、サプライヤーの最大 CSN には、最後に受信した更新が表示されます。）
最大 CSN	コンシューマーがサプライヤーから送信された最新の CSN の ID 番号。
時間ラグ	コンシューマーがサプライヤーから更新を受け取るのにかかる時間。これは、サプライヤーとコンシューマーの最大 CSN 間の時間差です。コンシューマーがサプライヤーと同期している場合、時間ラグは **0** になります。
Last Modify Time	コンシューマーの最後の更新時間を指定します（最後の CSN エントリーが送信された時間）。
supplier	そのコンシューマーに更新を送信するサプライヤーの名前を指定します。これは、コンシューマーが複数のサプライヤーから更新を受信する場合や、**Replication Status** ページに複数のサプライヤーを監視する場合に便利です。
送信/スキップ	レプリケーション更新でサプライヤーから送信された変更の数およびスキップされた数。数字はサプライヤーのメモリーにのみ保持され、サプライヤーが再起動された場合は消去されます。
更新のステータス	最後の更新のステータスコード（および意味）。すべての サプライヤーがビジーなレプリカを取得できないことを伝えると、この列はデッドロックを示している可能性があります。サプライヤーの 1 つが更新を行う場合、ビジーなメッセージが発生するのは普通です。
Update Start and End	最新の更新プロセスの開始および終了時のタイムスタンプ。
スケジュール	設定されたレプリケーションスケジュール。**0:-** は、コンシューマーがサプライヤーによって継続的に更新されることを意味します。
SSL?	サプライヤーが TLS 経由でコンシューマーに接続するかどうかを示します。

15.24.3. コマンドラインからのレプリケーションの監視

コマンドラインからレプリケーションステータスを表示するには、**-s** オプションを追加して **/usr/bin/repl-monitor.pl** スクリプトを実行します。スクリプトはレポートをプレーンテキスト形式で出力し、たとえばユーザーがレプリケーションステータスを迅速に判別したい場合に有用ですが、ブラウザーは利用できません。「Admin Express からのレプリケーションの監視」で説明されている Admin Express と同様に、**repl-monitor.pl** は、リアルタイムでレプリケーションの状態を表示します。

repl-monitor.pl スクリプトは、多くのコマンドラインオプションを受け入れます。これの使用方法は、repl-monitor(1) の man ページまたはDirectory Server Configuration, Command, and File Referenceを参照してください。

注記

-s オプションを指定せずに実行すると、repl**-monitor.pl** により、レポートが HTML ファイルとして生成されます。

15.25. 2 つの DIRECTORY SERVER インスタンスの比較

特定の状況では、管理者が 2 つの Directory Server を同期すれば比較する必要があります。**ds-replcheck** ユーティリティーを使用すると、2 つのオンラインサーバーを比較できます。また、**ds-replcheck** は、オフラインモードで 2 つの LDIF 形式のファイルを比較できます。

注記

2 つのデータベースをオフラインで比較するには、**db2ldif -r** コマンドを使用して複製の状態情報を追加します。

2 つのオンラインサーバーを比較すると、負荷が大きい場合、そのデータベースの内容が異なります。この問題を回避するには、***-l time_in_seconds*** パラメーターを **ds-replcheck** に渡すことで、スクリプトでラグ時間値を使用します。デフォルトでは、この値は 300 秒 (5 分) に設定されます。ユーティリティーがラグ時間内にある不整合を検出すると、報告されません。これにより、誤検出が軽減されます。

デフォルトでは、レプリカ合意の特定の属性を複製から除外した場合、**ds-replcheck** はこれらの属性を異なると報告します。これらの属性を無視するには、***-i attribute_list*** パラメーターをユーティリティーに渡します。

たとえば、2 つの Directory Server の ***dc=example,dc=com*** 接尾辞を比較するには、以下のコマンドを実行します。

```
# ds-replcheck -D "cn=Directory Manager" -W \
    -m ldap://server1.example.com:389 \
    -r ldap://server2.example.com:389 \
    -b "dc=example,dc=com"
```

このユーティリティーの出力には、以下のセクションが含まれます。

データベース RUV

データベースの Replication Update Vectors (RUV) と、最小および最大の Change Sequence Numbers (CSN) を一覧表示します。以下に例を示します。

```
Master RUV:
 {replica 1 ldap://server1.example.com:389} 58e53b92000200010000
58e6ab46000000010000
 {replica 2 ldap://server2.example.com:389} 58e53baa000000020000
58e69d7e000000020000
 {replicageneration} 58e53b7a000000010000

Replica RUV:
 {replica 1 ldap://server1.example.com:389} 58e53ba1000000010000
58e6ab46000000010000
 {replica 2 ldap://server2.example.com:389} 58e53baa000000020000
58e7e8a3000000020000
 {replicageneration} 58e53b7a000000010000
```

エントリー数

tombstone エントリーを含む、両方のサーバー上のエントリーの合計数を表示します。以下に例を示します。

```
Master:  12
Replica: 10
```

Tombstones

各レプリカの tombstone エントリーの数を表示します。これらのエントリーは、合計エントリー数に追加されます。以下に例を示します。

```
Master:  4
Replica: 2
```

競合エントリー

各競合エントリーの識別名 (DN)、競合タイプ、および作成された日付を一覧表示します。以下に例を示します。

```
Master Conflict Entries: 1

 - nsuniqueid=48177227-2ab611e7-afcb801a-ecef6d49+uid=user1,dc=example,dc=com
   - Conflict:   namingConflict (add) uid=user1,dc=example,dc=com
   - Glue entry: no
   - Created:    Wed Apr 26 20:27:40 2017


Replica Conflict Entries: 1

 - nsuniqueid=48177227-2ab611e7-afcb801a-ecef6d49+uid=user1,dc=example,dc=com
   - Conflict:   namingConflict (add) uid=user1,dc=example,dc=com
   - Glue entry: no
   - Created:    Wed Apr 26 20:27:40 2017
```

エントリーがありません

足りない各エントリーの DN と、エントリーが存在する他のサーバーからの作成日を一覧表示します。以下に例を示します。

```
Entries missing on Master:
 - uid=user2,dc=example,dc=com (Created on Replica at: Wed Apr 12 14:43:24 2017)
 - uid=user3,dc=example,dc=com (Created on Replica at: Wed Apr 12 14:43:24 2017)

Entries missing on Replica:
 - uid=user4,dc=example,dc=com (Created on Master at: Wed Apr 12 14:43:24 2017)
```

エントリーの不整合

相手のサーバーとは異なる属性を持つエントリーの DN を一覧表示します。状態情報が利用可能な場合は、これも表示されます。属性の状態情報が利用できない場合には、元の値として一覧表示されます。レプリケーションが初めて初期化されてから、値が更新されていないことを意味します。以下に例を示します。

```
cn=group1,dc=example,dc=com
```

```
-------------------------
Replica missing attribute "objectclass":

 - Master's State Info: objectClass;vucsn-58e53baa000000020000: top
 - Date: Wed Apr  5 14:47:06 2017

 - Master's State Info: objectClass;vucsn-58e53baa000000020000: groupofuniquenames
 - Date: Wed Apr  5 14:47:06 2017
```

ds-replcheck ユーティリティーの詳細は、Red Hat 設定、コマンド、およびファイルリファレンス の説明を参照してください。

15.26. 一般的な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は、最終的に調整されたレプリケーションモデルを使用します。つまり、同じエントリーを別のサーバーで変更できることを意味します。この 2 つのサーバー間でレプリケーションが発生した場合は、競合する変更を解決する必要があります。多くの場合は、各サーバーでの変更に関連するタイムスタンプに基づいて解決が自動的に行われます。最新の変更が優先されます。

ただし、解像度に到達するには、競合を手動で介入する必要があるケースがあります。レプリケーションプロセスで自動的に解決できない変更競合を持つエントリーには、*nsds5ReplConflict* 競合マーカー属性および *ldapSubEntry* オブジェクトクラスが含まれます。*nsds5ReplConflict* 属性は、存在および等価性のためにインデックス化される操作属性です。

競合エントリーを一覧表示するには、次のコマンドを実行します。

```
# ldapsearch -D "cn=Directory Manager" -W -b "dc=example,dc=com" \
    "(&(objectClass=ldapSubEntry)(nsds5ReplConflict=*))" \* nsds5ReplConflict
```

15.26.1. ネーミングの競合の解決

異なるサーバーの同じ DN で 2 つのエントリーを作成すると、レプリケーション中に自動的に競合解決が行われ、DN のエントリーの一意識別子を含む、最後に作成されたエントリーの名前が変更します。すべてのディレクトリーエントリーには、*nsuniqueid* の操作属性に格納されている一意の識別子が含まれます。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、2 つの異なるサーバーに uid=*user_name*,ou=People,dc=example,dc=com エントリーが作成された場合、レプリケーションは一意の ID を、作成した最後のエントリーの DN に追加します。つまり、以下のエントリーが存在します。

- uid=*user_name*,dc=example,dc=com
- nsuniqueid=*66446001-1dd211b2*+uid=*user_name*,dc=example,dc=com

レプリケーションの競合を解決するには、以下の手順を手動で決定する必要があります。

- 競合エントリーを削除して、有効なエントリー(uid=*user_name*,dc=example,dc=com)のみを保持するには、次のコマンドを実行します。

  ```
  # ldapdelete -D "cn=Directory Manager" -W -p 389 -h server.example.com -x \
      uid=nsuniqueid=66446001-1dd211b2+user_name,dc=example,dc=com
  ```

- 競合エントリー(nsuniqueid=*66446001-1dd211b2*+uid=*user_name*,dc=example,dc=com)のみを保持します。

  1. 有効なエントリーを削除します。

     ```
     # ldapdelete -D "cn=Directory Manager" -W -p 389 -h server.example.com -x \
       uid=user_name,dc=example,dc=com
     ```

  2. 競合エントリーの名前を変更します。「多値命名属性を使用したエントリーの名前変更」を参照してください。

- 両方のエントリーを保持するには、競合エントリーの名前を変更します。「多値命名属性を使用したエントリーの名前変更」 を参照してください。

15.26.1.1. 多値命名属性を使用したエントリーの名前変更

多値命名属性を持つエントリーの名前を変更するには、次のコマンドを実行します。

1. naming 属性の新しい値を使用してエントリーの名前を変更し、古い RDN を保持します。以下に例を示します。

   ```
   # ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
   dn: nsuniqueid=66446001-1dd211b2+uid=adamss,dc=example,dc=com
   changetype: modrdn
   newrdn: uid=NewValue
   deleteoldrdn: 0
   ```

   エントリーの名前変更時に RDN を維持する方法は、「*deleteOldRDN* パラメーター」 を参照してください。

2. naming 属性と conflict マーカー属性の古い RDN 値を削除します。以下に例を示します。

   ```
   # ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
   dn: uid=NewValue,dc=example,dc=com
   changetype: modify
   delete: uid
   uid: adamss
   -
   delete: nsds5ReplConflict
   -
   ```

注記

一意の識別子属性 *nsuniqueid* は削除できません。

コンソールは、多値 RDN の編集をサポートしません。たとえば、マルチマスターレプリケーションモードで 2 つのサーバーがある場合、同じユーザー ID を持つ各サーバーにエントリーを作成し、新しいエントリーの RDN が *nsuniqueid uid* 値に変更されます。コンソールからこのエントリーを変更しようとすると、多値 RDN を持つエントリーについてエラーの変更を保存できません。

高度なモードでエントリーを開くと、命名属性が *nsuniqueid uid* に設定されていることが分かります。ただし、ユーザー ID と RDN の値を異なる値に変更したり、修正したりすることはできません。たとえば、ユーザー ID がユーザー ID であり、jdoe1 に 変更する必要がある場合はコンソールから実行できません。代わりに ldapmodify コマンドを使用します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=John Doe
changetype: modify
replace: uid
uid: jdoe

dn: cn=John Doe
changetype: modrdn
newrdn: uid=jdoe1
deleteoldrdn: 1
```

15.26.1.2. 単一の値命名属性でエントリーの名前変更

1 値の naming 属性を持つエントリーの名前を変更するには、次のコマンドを実行します。

1. 別の naming 属性を使用してエントリーの名前を変更し、古い RDN を保持します。以下に例を示します。

   ```
   # ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
   dn: nsuniqueid=66446001-1dd211b2+dc=pubs,dc=example,dc=com
   changetype: modrdn
   newrdn: cn=TempValue
   deleteoldrdn: 0
   ```

   エントリーの名前変更時に RDN を維持する方法は、「*deleteOldRDN* パラメーター」 を参照してください。

2. naming 属性と conflict マーカー属性の古い RDN 値を削除します。以下に例を示します。

   ```
   # ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
   dn: cn=TempValue,dc=example,dc=com
   changetype: modify
   delete: dc
   dc: pubs
   -
   delete: nsds5ReplConflict
   -
   ```

   注記

   一意の識別子属性 *nsuniqueid* は削除できません。

3. 目的の属性と値のペアでエントリーの名前を変更します。以下に例を示します。

   ```
   # ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
   dn: cn=TempValue,dc=example,dc=com
   changetype: modrdn
   newrdn: dc=NewValue
   deleteoldrdn: 1
   ```

   deleteoldrdn 属性の値を 1 に設定して、一時属性と値のペア cn=TempValue を削除します。この属性を保持するには、*deleteoldrdn* 属性の値を 0 に設定します。

15.26.2. 孤立エントリーの競合の解決

削除操作が複製され、コンシューマーサーバーが、削除されるエントリーに子エントリーがあることを検出すると、競合解決の手順により、ディレクトリーに孤立したエントリーが存在しないように、glue エントリーが作成されます。

同様に、追加操作が複製され、コンシューマーサーバーが親エントリーを検出できない場合は、競合解決の手順により、新しいエントリーが孤立エントリーではないように、親を表す glue エントリーが作成されます。

glue エントリーは、オブジェクトクラス glue および extensibleObject を含む一時エントリーです。チャンネルエントリーは、複数の方法で作成できます。

- 競合解決手続で、一致する一意の識別子を持つ削除されたエントリーが見つかった場合、glue エントリーは、そのエントリーの再生であり、glue のオブジェクトクラスと *nsds5ReplConflict* の属性が追加されます。

  このような場合は、glue エントリーを変更して glue オブジェクトクラスと *nsds5ReplConflict* 属性を削除して、エントリーを通常のエントリーとして維持するか、その子エントリーを削除します。

- サーバーは glue および extensibleObject オブジェクトクラスを使用して最小のエントリーを作成します。

このような場合は、エントリーを変更して意味のあるエントリーに変換するか、またはそのすべての子エントリーを削除します。

15.26.3. 廃止または不明なエラーの解決

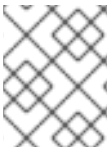
レプリケーショントポロジーに関する情報、つまり、相互に、および同じレプリケーショングループ内の他のレプリカに更新を提供するすべてのサプライヤーは、レプリカ更新ベクトル (RUV) と呼ばれるメタデータのセットに含まれています。RUV には、ID と URL、ローカルサーバー上の最新の変更状態番号 (CSN)、最初の変更の CSN などのサプライヤーに関する情報が含まれています。サプライヤーとコンシューマーはいずれも RUV 情報を保存し、これを使用してレプリケーションの更新を制御します。

あるサプライヤーがレプリケーショントポロジーから削除されると、別のレプリカの RUV に残っている場合があります。他のレプリカが再起動すると、ログにエラーを記録し、レプリケーションプラグインが削除されたサプライヤーを認識しないことを警告します。エラーは以下の例のようになります。

```
[22/Jan/2020:17:16:01 -0500] NSMMReplicationPlugin - ruv_compare_ruv: RUV [changelog
max RUV] does not contain element [{replica 8 ldap://m2.example.com:389}
4aac3e59000000080000 4c6f2a02000000080000] which is present in RUV [database RUV]

<...several more samples...>

[22/Jan/2020:17:16:01 -0500] NSMMReplicationPlugin - replica_check_for_data_reload:
Warning: for replica dc=example,dc=com there were some differences between the changelog
max RUV and the database RUV. If there are obsolete elements in the database RUV, you
should remove them using the CLEANALLRUV task. If they are not obsolete, you should
check their status to see why there are no changes from those servers in the changelog.
```

レプリカとその ID (この場合はレプリカ 8 を書き留めます。

サブライヤーがトポロジーから永久に削除されると、そのサブライヤーに関する残存するメタデータは、他のすべてのサプライヤーの RUV エントリーから消去されるはずです。**cleanallruv** ディレクトリータスクを使用して、トポロジー内のすべてのサプライヤーから RUV エントリーを削除します。



注記

cleanallruv タスクが複製されます。そのため、1つのマスターでのみ実行する必要があります。

手順15.1 cleanallruv タスク操作を使用した、廃止または欠落したサプライヤーの削除

1. 削除されたマスターが他のマスターにメタデータを残す可能性があるため、有効無効を問わず、すべての RUV レコードとレプリカ ID を一覧表示します。

```
# ldapsearch -o ldif-wrap=no -xLLL -H m1.example.com -D "cn=Directory Manager" -W -b
dc=example,dc=com '(&(nsuniqueid=ffffffff-ffffffff-ffffffff-ffffffff)(objectclass=nstombstone))'
nsDS5ReplicaId nsDS5ReplicaType nsds50ruv
dn: cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
nsDS5ReplicaId: 1
nsDS5ReplicaType: 3
nsds50ruv: {replicageneration} 55d5093a000000010000
nsds50ruv: {replica 1 ldap://m1.example.com:389} 55d57026000000010000
55d57275000000010000
nsds50ruv: {replica 20 ldap://m2.example.com:389} 55e74b8c000000140000
55e74bf7000000140000
nsds50ruv: {replica 9 ldap://m2.example.com:389}
nsds50ruv: {replica 8 ldap://m2.example.com:389} 506f921f000000080000
50774211000500080000
```

返されたレプリカ ID **1**、**20**、**9**、および **8** を書き留めます。

2. cn= **config** 接尾辞のレプリカ設定エントリー DN **cn=replica** を検索して、データベースを複製するすべてのマスターで現在定義され、有効なレプリカ ID を一覧表示します。

注記

コンシューマーおよび読み取り専用ノードには、レプリカ ID が **65535** に常に設定され、**nsDS5ReplicaType: 3** はマスターを署名します。

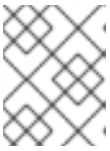
```
# ldapsearch -o ldif-wrap=no -xLLL -H m1.example.com m2.example.com -D "cn=Directory
Manager" -W -b cn=config cn=replica nsDS5ReplicaId nsDS5ReplicaType
dn: cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
nsDS5ReplicaId: 1
nsDS5ReplicaType: 3

dn: cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
nsDS5ReplicaId: 20
nsDS5ReplicaType: 3
```

最初のステップで返されるすべての URI を検索すると（この手順では **m1.example.com** および **m2.example.com**）、返されたマスターのリスト(**nsDS5ReplicaType: 3**)を直前の手順の RUV の一覧と比較します。上記の例では、この検索で ID **1** と **20** のみが返されますが、以前の検索では URI **m2.example.com** で **9** および **8** も返されます。これは、後者の 2 つが削除済みで、その RUV を消去する必要があることを意味します。

3. クリーニングが必要な RUV を判別した後に、新しい **cn=cleanallruv,cn=tasks,cn=config** エントリーを作成し、レプリケーション設定に関する以下の情報を提供します。

   - 複製されたデータベースのベース DN (**replica-base-dn**)
   - レプリカ ID (**replica-id**)
   - 欠落しているサプライヤーからの最大変更状態番号 (CSN) に追いつくか、あるいはすべての RUV エントリーを削除して更新を見逃すか (**replica-force-cleaning**)。この属性を **no** に設定すると、タスクは設定されているすべてのレプリカが、まず削除されたレプリカからのすべての変更に追いつくのを待ってから、RUV を削除することになります。

```
# ldapmodify -a -D "cn=Directory Manager" -W -H m2.example.com -x
dn: cn=clean 8,cn=cleanallruv,cn=tasks,cn=config
objectclass: extensibleObject
replica-base-dn: dc=example,dc=com
replica-id: 8
replica-force-cleaning: no
cn: clean 8
```



注記

cleanallruv タスクが複製されます。そのため、1つのマスターでのみ実行する必要があります。

整理したい各 RUV に同じことを繰り返す (この手順では ID 9)。

4. 先に確認したすべてのレプリカの RUV をクリーンアップした後、最初の手順からの検索を再度使用して、追加の RUV がすべて削除されていることを確認します。

```
# ldapsearch -o ldif-wrap=no -xLLL -H m1.example.com -D "cn=Directory Manager" -W -b
dc=example,dc=com '(&(nsuniqueid=ffffffff-ffffffff-ffffffff-ffffffff)(objectclass=nstombstone))'
nsDS5ReplicaId nsDS5ReplicaType nsds50ruv
dn: cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
nsDS5ReplicaId: 1
nsDS5ReplicaType: 3
nsds50ruv: {replicageneration} 55d5093a000000010000
nsds50ruv: {replica 1 ldap://m1.example.com:389} 55d57026000000010000
55d57275000000010000
nsds50ruv: {replica 20 ldap://m2.example.com:389} 55e74b8c000000140000
55e74bf7000000140000
```

上記の出力で分かるように、レプリカ ID 8 および 9 は存在しなくなり、RUV が正常に消去されていることを示しています。

15.27. レプリケーション関連の問題のトラブルシューティング

ここでは、いくつかのエラーメッセージを挙げ、その原因と対処法を説明します。

エラーログレベルを 8192 (レプリケーションのデバッグ) に設定すると、レプリケーションに関する詳細なデバッグ情報を取得できます。「ログレベルの設定」 を参照してください。

ldapmodify でエラーログレベルを 8192 に変更するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=config
changetype: modify
replace: nsslapd-errorlog-level
nsslapd-errorlog-level: 8192
```

ログレベルは加算されるため、上記のコマンドを実行すると、エラーログに過剰なメッセージが表示されます。だから、判断して使用してください。

レプリケーションのデバッグログを無効にするには、同じ属性を 0 に設定します。

cl-dump.pl スクリプト は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス で詳細に説明され』、レプリケーション関連の問題のトラブルシューティングにも役立ちます。使用方法に応じて、スクリプトは特定のレプリカを選択的にダンプできます。

- **replication-change-log** ファイルとインメモリー変数の内容をダンプします。また、メモリー変数は **RUV** および **maxRUV** をパージ します。
- Changelog で grep 状態番号(CSN)を解釈します。
- Directory Server から base-64 でエンコードされた changelog を取得し、changelog をデコードします。

以下のセクションでは、数多くの一般的なレプリケーションの問題を説明します。

agmt=%s (%s:%d) Replica has a different generation ID than the local data

- 理由: このメッセージの最初に指定されたコンシューマーがまだ (正常に)初期化されていないか、または異なるルートサプライヤーから初期化されました。
- 影響: ローカルのサプライヤーは、コンシューマーにデータを複製することはありません。
- 対策: コンシューマーを初期化する前に発生する場合は、このメッセージを無視してください。または、メッセージが永続的であれば、コンシューマーを再初期化します。マルチマスター環境では、すべてのサーバーは、ルートサプライヤーから直接または間接的に 1回のみ初期化する必要があります。たとえば、M1 は M2 および M4 を初期化し、M2 は M3 の初期化を行います。重要なことは、M2 自身の初期化が完了するまで M3 の初期化を開始しないでください（M1 のコンソール、M1 または M2 のエラーログから合計更新ステータスを確認してください）。また、M2 は M1 を再び初期化しないでください。

Warning: data for replica's was reloaded, and it no longer matches the data in the changelog.Recreating the changelog file.This could affect replication with replica's consumers, in which case the consumers should be reinitialized.

- 理由: このメッセージは、サプライヤーが再開した場合にのみ表示されます。これは、サプライヤーが changelog を書き込みできないか、または最後のシャットダウン時に RUV を消去できなかったことを示しています。前者はディスク領域の問題、後者はサーバーのクラッシュや不適切なシャットダウンなどが原因です。
- 影響: コンシューマの ***maxcsn*** がサーバーの changelog に存在しない場合、サーバーはコンシューマーに変更を送信できません。
- 対策: ディスク領域と考えられるコアファイル (サーバーの logs ディレクトリー配下) を確認します。これが単一マスターレプリケーションの場合は、コンシューマーを再初期化します。そうでなければ、後にサーバーがコンシューマーの CSN を見つけられないと訴えた場合に、コン

シューマーが他のサプライヤーから CSN を入手できるかどうかを確認します。そうでない場合には、コンシューマーを再初期化します。

agmt=%s(%s:%d): Can't locate CSN %s in the changelog (DB rc=%d).The consumer may need to be reinitialized.

- 理由: ディスクが満杯になったり、またはサーバーが適切にシャットダウンしたりしたたあります。あります。
- 影響: ローカルサーバーは、コンシューマーが再初期化されたり、他のサプライヤーから CSN を取得するまで、そのコンシューマーへの追加の変更を送信できません。
- 対策: これが単一マスターのレプリケーションである場合は、コンシューマーを再初期化します。そうでない場合は、コンシューマーが他のサプライヤーから CSN を取得できるかどうかを確認します。そうでない場合には、コンシューマーを再初期化します。

Too much time skew

- 理由: ホストマシンのシステムクロックは同期が非常に低下します。
- 影響: システムクロックは、CSN の一部を生成するのに使用されます。複数のサプライヤー間で変更シーケンスを反映させるために、サプライヤーは、他のサプライヤーのリモートクロックに基づいて、ローカルクロックを転送します。調整は一定量に制限されているため、許可される制限を超過すると、レプリケーションセッションが中止します。
- 対策: Directory Server ホストマシンでシステムクロックを同期します。該当する場合は、それらのホストでネットワークタイムプロトコル (ntp) デーモンを実行します。

agmt=%s(%s:%d): Warning: Unable to send endReplication extended operation (%s)

- 理由: コンシューマーが応答しない。
- 影響: コンシューマーが再起動せずに回復すると、サプライヤーからリリースロックメッセージを受け取らないと、コンシューマーのレプリカが永久にロックされる可能性が高くなります。
- 対策: コンシューマーがあらゆるサプライヤーから新しい変更を受け取ったり、レプリケーションモニターを開始できる場合に、このコンシューマーの全サプライヤーが、レプリカがビジーであることを警告しているかどうかを確認します。レプリカが永久にロックされ、サプライヤーを取得できない場合は、コンシューマーを再起動します。

Changelog is getting too big.

- 理由: changelog のパージがオフになっていて、それがデフォルトの設定になっているか、changelog のパージがオンになっていても一部のコンシューマーがサプライヤーよりもずっと遅れているかのどちらかです。
- 対策： デフォルトでは、changelog のパージはオフになっています。コマンドラインからこれを有効にするには、以下のように **ldapmodify** を実行します。

```
ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=changelog5,cn=config
changetype: modify
add: nsslapd-changelogmaxage
nsslapd-changelogmaxage: 1d
```

1d は 1 日を意味します。その他の有効な時間単位は、秒 (s)、分 (m)、時 (h)、および週 (w) になります。0 の場合は、パージをオフにします。

changelog パージがオンになっている場合、5 分ごとに起動するパージスレッドは、そのエイジが nsslapd-changelogmaxage の値よりも大きく、そのサプライヤー (サプライヤーまたはハブ) のすべての直接コンシューマーに再生されている場合にその変更を削除します。

パージのしきい値に達したときに changelog がパージされていないと思われる場合は、すべてのコンシューマー間でレプリケーションモニターからの最大遅延時間を確認します。パージのしきい値に関わらず、すべての消費者によって再生される前に変更がパージされることはありません。

The Replication Monitor is not responding.

- 理由: LDAPS ポートは一部のレプリカ合意で指定されますが、証明書データベースは Replication Monitor によって指定されず、アクセスできません。LDAPS ポートに問題がある場合は、レプリケーショントポロジー内のいずれかのサーバーがハングする可能性があります。
- 対策: Replication Monitor の設定ファイルで TLS ポートを TLS 以外のポートにマッピングします。たとえば、636 が TLS ポートであり、389 が TLS 以外のポートである場合は、[connection] セクションに以下の行を追加します。

```
*:636=389:*:password
```

In the Replication Monitor, some consumers show just the header of the table.

- 理由: 対応するサプライヤーから作成された変更はありません。この場合、ヘッダー部分の *MaxCSN*: は "None" である必要があります。
- 対策: サプライヤーからの変更がない場合は、何も間違ってありません。

第16章 RED HAT DIRECTORY SERVER と MICROSOFT ACTIVE DIRECTORY の同期

Windows 同期は、ディレクトリーの変更 (Red Hat Directory Server と Microsoft Active Directory の間のグループ、ユーザー、およびパスワードの追加、削除、変更) を行います。これにより、ディレクトリー全体で一貫した情報を維持する方がはるかに効率的で効果的になります。

16.1. WINDOWS 同期の概要

同期により、Active Directory のユーザーエントリーおよびグループエントリーが Red Hat Directory Server のエントリーと一致するようになります。エントリーは作成、変更、または削除されるため、対応する変更が同期ピアサーバーに加えられ、ユーザー、パスワード、およびグループの双方向の同期が可能になります。

同期プロセスはレプリケーションプロセスに似ています。同期はプラグインによって有効にされ、同期合意を介して開始して、ディレクトリーの変更の記録はその changelog に従って送信されます。これにより、Directory Server と Windows サーバーの間のユーザーおよびグループが同期されます。

Windows Synchronization には、ユーザーエントリーおよびグループエントリー用の部分が 2 つあり、もう 1 つはパスワード用です。

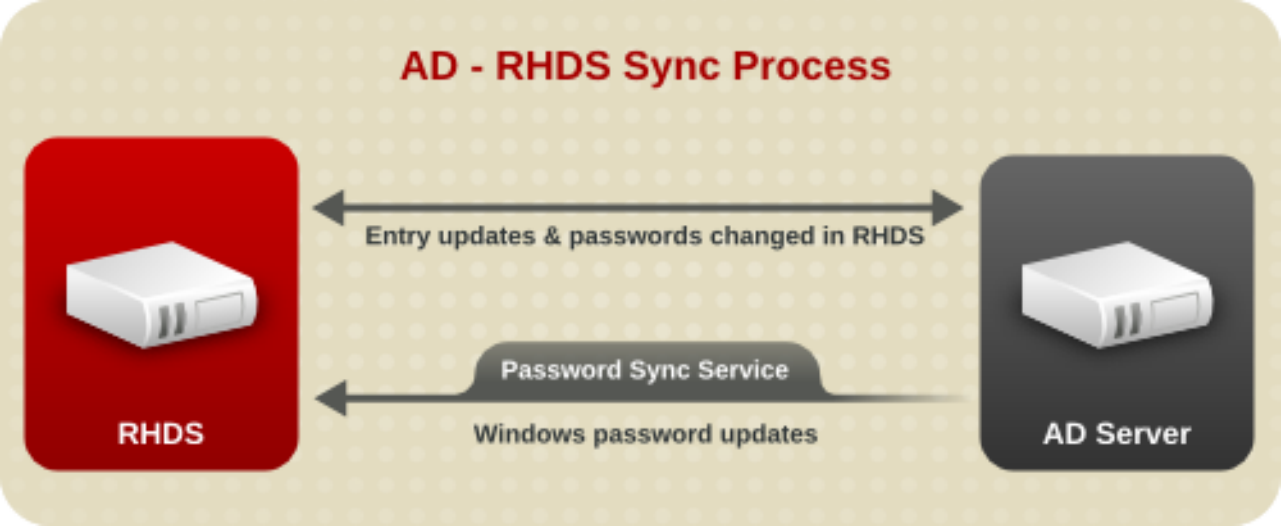
- Directory Server Windows 同期ユーザーエントリーおよびグループエントリーの同期は同期合意で設定されます。同様に、レプリケーションがレプリカ合意で設定されます。同期合意は、同期するエントリーの種類 (ユーザー、グループ、またはその両方) と、同期する方向の変更 (Directory Server から Active Directory へ、Active Directory から Directory Server へ、またはその両方) を定義します。

  Directory Server は、マルチマスターレプリケーションプラグインを使用して、ユーザーエントリーおよびグループエントリーを同期します。マルチマスターレプリケーションに使用されるものと同じ changelog は、LDAP 操作として Directory Server から Active Directory に更新を送信するために使用されます。サーバーは、Windows サーバーに対して LDAP 検索操作を実行し、Windows エントリーに加えた変更を対応する Directory Server エントリーと同期します。

- パスワード同期サービス。Directory Server で行われたパスワードの変更は Active Directory に自動的に同期されますが、Active Directory でパスワード変更を認識して Directory Server に送るための特別なフックが必要です。これは、パスワード同期サービスにより実行されます。このアプリケーションは、Windows マシンのパスワード変更をキャプチャーして、LDAPS 経由で Directory Server に送信します。

  パスワード同期サービスは、すべての Active Directory ドメインコントローラーにインストールする必要があります。

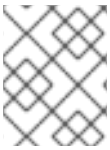
図16.1 Active Directory - Directory Server の同期プロセス



同期は、1つ以上の 同期合意 により設定され、制御され、同期ピア と同期されるディレクトリーサーバー間の同期を確立します。これらはレプリカ合意と似ており、ホスト名 (IPv4 または IPv6 アドレス) や Active Directory のポート番号などに同様の情報が含まれています。Directory Server は、LDAP/LDAPS を使用してピア Windows サーバーに接続して、更新の送受信を行います。

LDAP (標準接続) は、ユーザーエントリーおよびグループエントリーのみの同期に使用することができますが、パスワードを同期するためには、セキュアな接続の一部が必要になります。セキュアな接続を使用しない場合、Windows ドメインは Directory Server からのパスワードの変更を受け入れず、Password Synchronization サービスは Active Directory ドメインから Directory Server にパスワードを送信しません。Windows Synchronization では、TLS と Start TLS を使用して LDAPS の両方を許可します。

複数のサブツリーのペアを設定して、相互に同期することができます。データベース に接続するレプリケーションとは異なり、同期はディレクトリーツリー構造の 接尾辞 間で行われます。同期された Active Directory と Directory Server の接尾辞はいずれも、同期合意で指定します。各サブツリー内のすべてのエントリーは、指定の接尾辞 DN の子ではないエントリーを含む、同期用の候補となります。



注記

管理者によって Active Directory とは別に子コンテナーエントリーを作成する必要があります。Windows Synchronization はコンテナーエントリーを作成しません。

Directory Server は、発生した変更を記録するデータベースである changelog を維持します。changelog は、Windows Synchronization により Active Directory ピアに追加された変更を調整および送信するために使用されます。Active Directory のエントリーへの変更は、Active Directory の Dirsync 検索機能を使用して確認できます。Directory Server は、デフォルトで 5 分ごとに定期的に Dirsync 検索を実行し、Active Directory サーバーの変更を確認します。**cn=*syncAgreement_Name*,cn=WindowsReplica,cn=*suffix_Name*,cn=mapping tree,cn=config** エントリーの ***winSyncInterval*** パラメーターを設定して、このデフォルトを変更することができます。Dirsync を使用すると、以前の検索以降に変更されたエントリーのみが取得されます。

同期が設定されている場合や、ディレクトリーデータに大きな変更があった場合など、状況によっては全体の更新 (再同期) を実行することができます。これにより、同期ピアのすべてのエントリーを調べ、変更または不足しているエントリーを送信します。更新全体が実行するたびに、完全な Dirsync 検索が開始します。詳細は、「同期更新の送信」を参照してください。

Windows 同期機能では、同期するエントリーを管理者が細かく制御できるように、また、さまざまな導入シナリオをサポートするための十分な柔軟性を持たせるために、同期するエントリーをある程度制

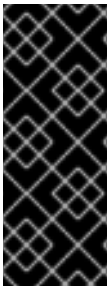
御することができます。このコントロールは、Directory Server に設定された異なる設定属性を使用して設定されます。

- 同期合意の作成時に、作成時に新しい Windows エントリー (*nsDS7NewWinUserSyncEnabled* および *nsDS7NewWinGroupSyncEnabled*) を同期するオプションがあります。これらの属性が **on** に設定されている場合は、既存の Windows ユーザー/グループが Directory Server に同期され、作成されるユーザー/グループが Directory Server と同期されます。

  Windows サブツリー内では、ユーザーまたはグループのオブジェクトクラスを持つエントリーのみを Directory Server に同期できます。

- Directory Server で、**ntUser** または **ntGroup** オブジェクトクラス、および属性を持つエントリーのみを同期できます。

同期合意の配置は、同期される接尾辞によって異なります。単一の接尾辞の場合、同期合意はその接尾辞に対してのみ行われます。複数の接尾辞の場合、同期合意はディレクトリーツリーの上位ブランチで行われます。Directory Server のデプロイメント全体で Windows エントリーおよび更新を伝播するには、図16.2「マルチマスターディレクトリーサーバー - Windows ドメインの同期」にあるように、マルチマスターレプリケーション環境でマスター間で合意を作成し、そのマスターを使用して変更を Directory Server デプロイメント全体に複製します。



重要

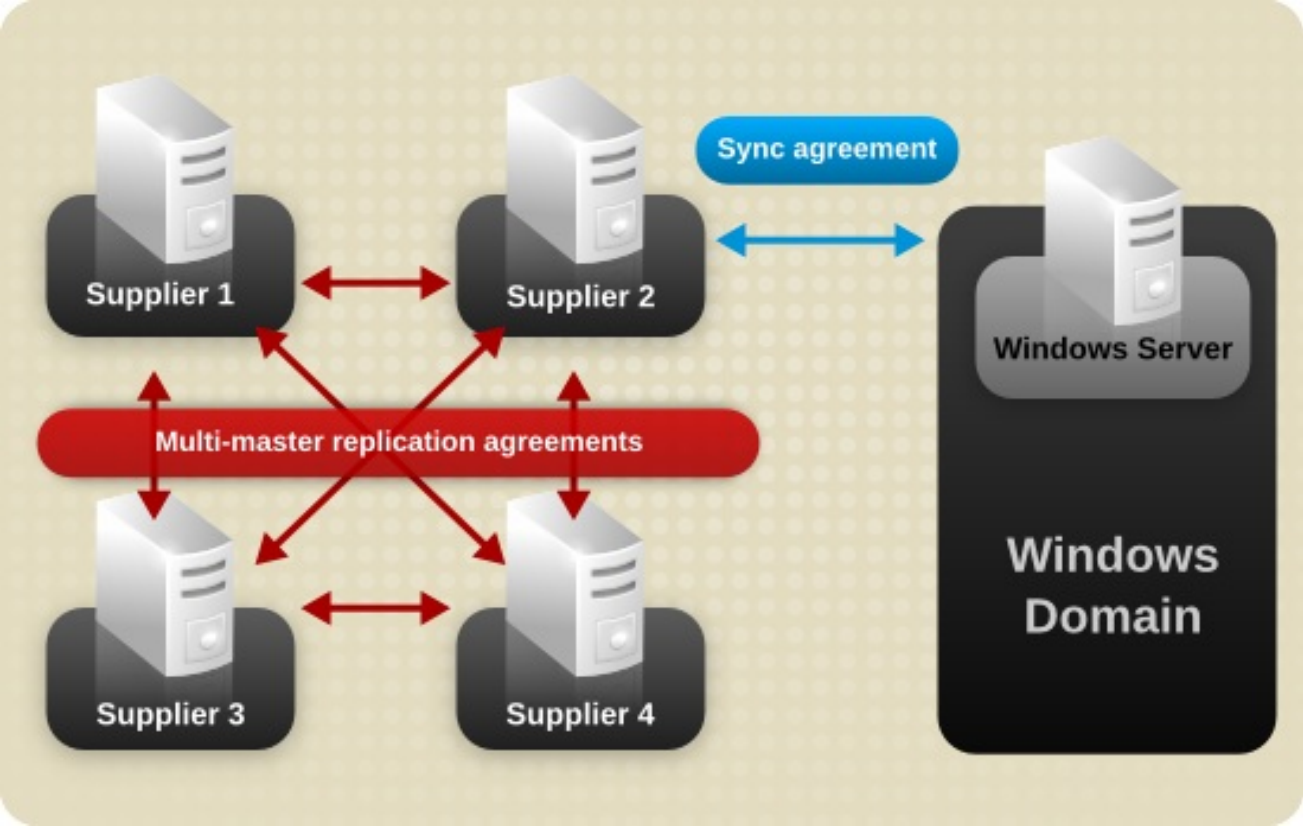
ハブサーバーで同期合意を設定することは可能ですが、Red Hat Directory Server から Active Directory への一方向の同期のみが許可されます。Active Directory サーバーは、変更内容をハブに同期することができません。

同期合意を設定するために、マルチマスターレプリケーションのマスターのみを使用することが強く推奨されます。

警告

Directory Server 環境と Active Directory 環境との間の同期合意は 1 つのみです。同じ Active Directory ドメインに同期合意が複数存在すると、エントリーの競合を作成できます。

図16.2 マルチマスターディレクトリーサーバー - Windows ドメインの同期



Directory Server の changelog で平文のパスワードが保持されるため、Directory Server のパスワードは他のエントリー属性と同期されます。Active Directory で行われたパスワード変更を取得するには、パスワード同期サービスが必要です。パスワード同期サービスがないと、パスワードが Active Directory でハッシュ化され、Windows ハッシュ機能が Directory Server が使用するものと同じであるため、Windows パスワードが同期できません。

16.2. サポート対象の ACTIVE DIRECTORY のバージョン

『『Red Hat Directory Server リリースノート』を参照してください』。

16.3. パスワードの同期

Directory Server エントリーのパスワード変更は、Password Sync ユーティリティーを使用して Active Directory エントリーのパスワード属性に同期できます。

パスワードの同期時に、パスワードポリシーは各同期ピアに対してローカルに強制されます。Directory Server でパスワードが変更すると、Directory Server の構文または最小長の要件が適用されます。変更したパスワードが Windows サーバーと同期すると、Windows パスワードポリシーが適用されます。パスワードポリシー自体は同期されません。

パスワード変更履歴やアカウントロックアウトカウンターなどの設定情報はローカルに保持され、同期できません。

同期用のパスワードポリシーを設定する場合は、以下の点を考慮してください。

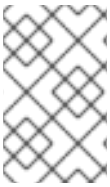
- Password Sync ユーティリティーは、Directory Server と同期する Windows マシンにローカルにインストールする必要があります。

- Password Sync は、Windows マシンを 1 つの Directory Server にのみリンクできます。複数の Directory Server インスタンスと変更を同期するには、マルチマスターレプリケーション用に Directory Server を設定します。
- パスワードの有効期限の警告および時間、バインド試行の失敗、その他のパスワード関連の情報はサーバーごとにローカルで適用され、同期ピアサーバー間で同期されません。
- バインド動作は、すべてのサーバーで発生します。Directory Server サーバーおよび Active Directory サーバーの両方で、同じパスワードポリシーまたは同様のパスワードポリシーを作成してください。
- 同期用に作成されるエントリー (例： サーバーアイデンティティー) には有効期限のないパスワードが必要です。これらの特別なユーザーが期限切れにならないパスワードを持っていることを確認するために、Directory Server のエントリーに ***passwordExpirationTime*** 属性を追加し、それに **20380119031407Z** の値 (有効範囲の一番上) を指定します。

Directory Server および Windows ユーザーとパスワードの同期の詳細は、16章*Red Hat Directory Server と Microsoft Active Directory の同期* を参照してください。

16.4. WINDOWS 同期の設定手順

同期の設定は、レプリケーションの設定と非常に似ています。このデータベースを changelog を使用してマスターとして設定し、同期を定義する合意を作成する必要があります。同期ユーザーである一般的なユーザーアイデンティティーは、Windows 同期ピアに接続して Directory Server から更新を送信し、更新が Directory Server に同期するために更新を確認します。



注記

（Directory Server と Active Directory の両方でユーザーがアクティブになる唯一の方法）パスワードを同期するには、TLS 経由で同期を設定する必要があります。そのため、この設定セクションでは TLS も設定する必要があります。

TLS での同期を設定することも、TLS でのレプリケーションの設定と同様です。両方の同期ピアは、暗号化セッションに対して相互を信頼するように設定する必要があります（すべてのパスワード操作は TLS 上で実行されます）。

ユーザーとグループのエントリーの同期はすべて、Active Directory(AD)側からパッシブになります。これは、サイドで更新を送信し、AD ドメインの更新をポーリングする Directory Server です。パスワードには、AD サーバーに別のパスワードサービスが必要です。このサービスは、AD ドメインから Directory Server にパスワード変更をアクティブに送信します。

16.4.1. ステップ 1: Directory Server での TLS の設定

TLS で実行される Directory Server の設定方法は 「Directory Server での TLS の有効化」にあります。基本的に、Directory Server には適切な TLS 証明書がインストールされ、LDAPS ポート経由で実行するように設定し、他のサーバーからのクライアント認証を許可するように設定する必要があります。

Directory Server と AD 同期ピアの両方で、2 つの証明書を発行し、インストールする必要があります。

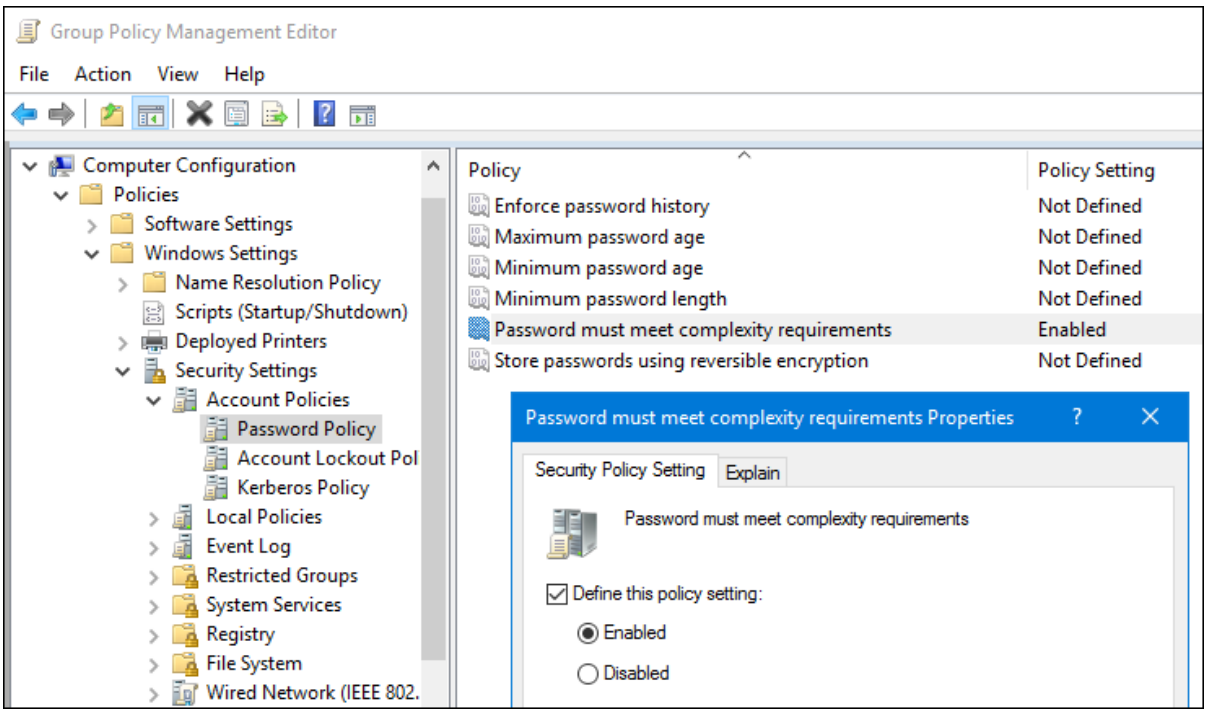
- Directory Server と AD 間で共有される CA 証明書
- 同期サービスがアクセスできる Directory Server および AD 同期ピアのサーバー証明書

16.4.2. ステップ 2: Active Directory ドメインの設定

同期は AD ドメインコントローラーでのみ設定できます。さらに、パスワードの複雑さを AD で有効にする必要があります。

パスワードの複雑性を有効にするには、以下を実行します。

1. Group Policy Management コンソールを開き、新しい Group Policy Object(GPO)を作成します。詳細は、Windows のドキュメントを参照してください。
2. GPO を右クリックし、**Edit** を選択して Group Policy Management Editor を開きます。
3. Computer Configuration → Windows Settings → Security Settings → Account Policies → Password Policy に移動し、**Password must meet complexity requirements** という名前のポリシーをダブルクリックします。
4. ポリシーを有効にし、**OK** をクリックします。



5. Group Policy Management Editor および Group Policy Management コンソールを閉じます。

「Microsoft ナレッジベース」で説明されているように、TLS を設定し、AD サーバーにルート CA を設定します http://technet.microsoft.com/en-us/library/cc772393%28v=ws.10%29.aspx#BKMK_AS1。

1. 認証局をインストールします。
   a. **Administrative Tools** エリアで **Server Manager** を開き、ロールを追加します。
   b. **Active Directory Certificate Services** チェックボックスを選択します。
   c. **Select Role Services** ページをクリックし、**Certification Authority** チェックボックスを選択します。
   d. CA の設定時に、適切な画面で以下のオプションを選択します。

- Enterprise（設定タイプの場合）
- オプション設定の認証局の Web 登録

e. AD サーバーを再起動します。

2. TLS サーバー証明書を使用するように AD サーバーを設定します。

a. AD の完全修飾ドメイン名を証明書サブジェクトとして使用し、証明書要求 .inf を作成します。以下に例を示します。

```
;----------------- request.inf -----------------

[Version]

Signature="$Windows NT$

[NewRequest]

Subject = "CN=ad.server.example.com, O=Engineering, L=Raleigh, S=North
Carolina, C=US"
KeySpec = 1
KeyLength = 2048
Exportable = TRUE
MachineKeySet = TRUE
SMIME = False
PrivateKeyArchive = FALSE
UserProtected = FALSE
UseExistingKeySet = FALSE
ProviderName = "Microsoft RSA SChannel Cryptographic Provider"
ProviderType = 12
RequestType = PKCS10
KeyUsage = 0xa0

[EnhancedKeyUsageExtension]

OID=1.3.6.1.5.5.7.3.1

;-------------------------------------------
```

b. 証明書要求を生成します。

```
# certreq -new request.inf request.req
```

c. AD CA に要求を送信します。以下に例を示します。

```
# certreq -submit request.req certnew.cer
```

注記

コマンドラインツールがエラーメッセージを返す場合は、Web ブラウザーを使用して CA にアクセスし、証明書要求を送信します。IIS が実行されている場合、CA URL は **http:**//servername/**certsrv** になります。

d. 証明書要求を受け入れます。以下に例を示します。

```
# certreq -accept certnew.cer
```

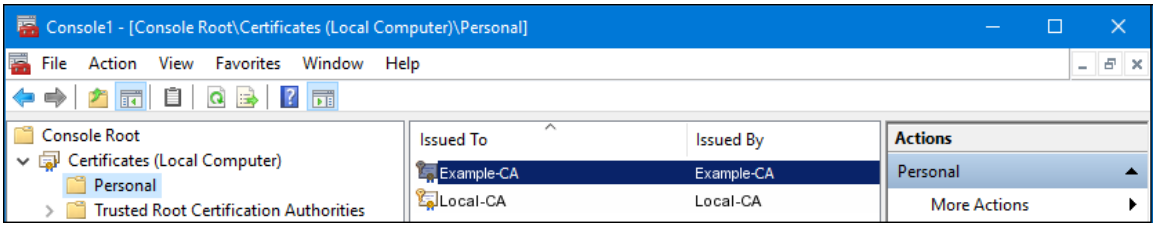
3. サーバー証明書が AD サーバーに存在する。

a. **Run** メニューで MMC コンソールを開きます。

b. File メニューで、Add/Remove Snap-in.. をクリックします。

c. Certificates snap-in を選択し、**Add** をクリックしてこれを追加し、**Next** をクリックします。

d. 左側の 証明書（ローカル） メニューを展開します。Personal 項目を展開し、**Certificates** をクリックします。



e. 新しい証明書は他の証明書と共に一覧表示される必要があります。

4. Directory Server で、CA 証明書をエクスポートします。

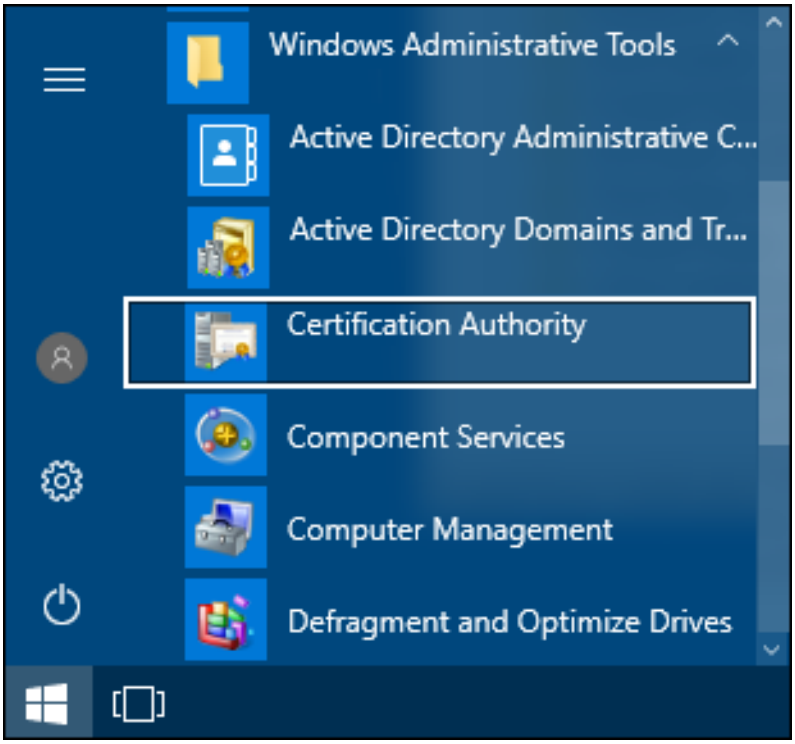
```
# cd /etc/dirsrv/slapd-instance_name/
# certutil -d . -L -n "CA certificate" -a > dsca.crt
```

5. エクスポートされた証明書を Directory Server から Windows マシンにコピーします。

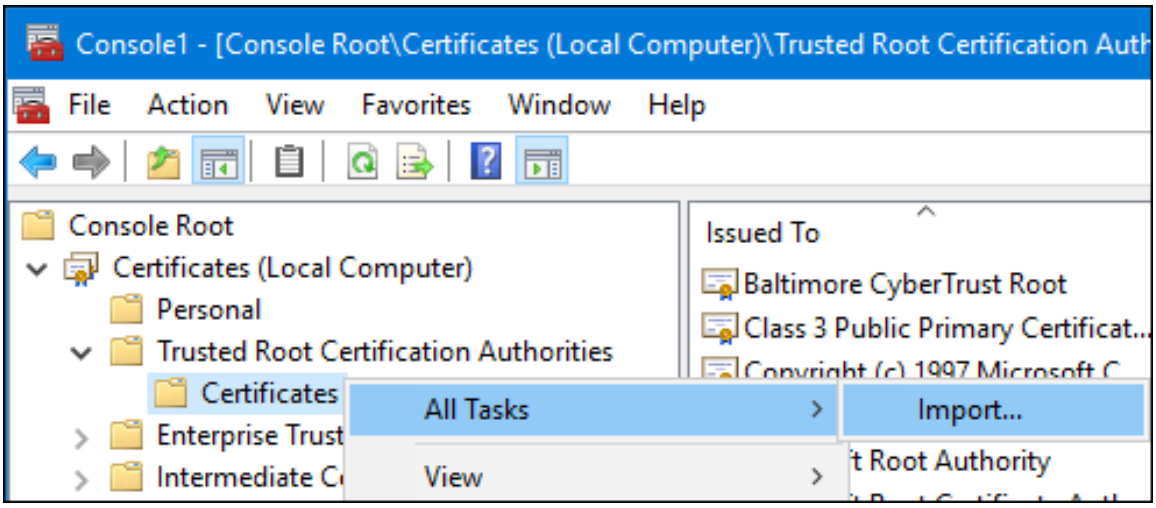
6. Directory Server から AD に CA 証明書をインポートします。

a. **Administrative Tools** を開き、認証局 項目を選択します。

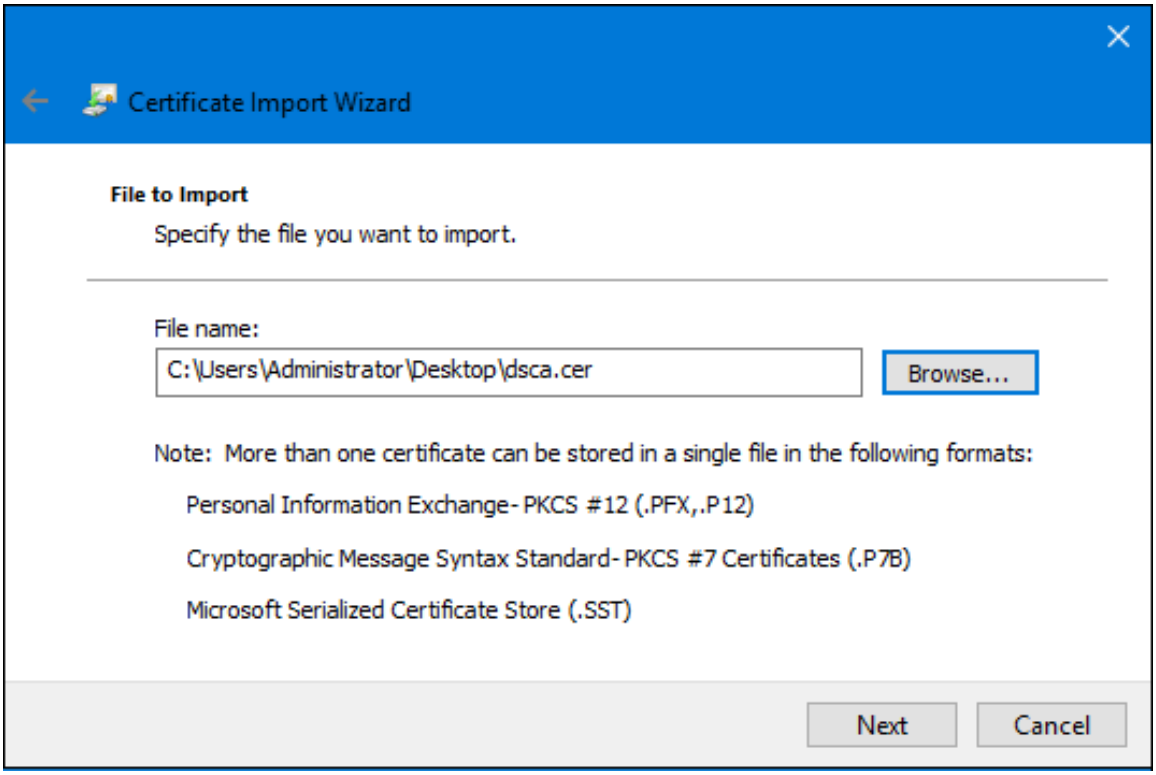
b. **Trusted Root Certification Authorities** を展開します。



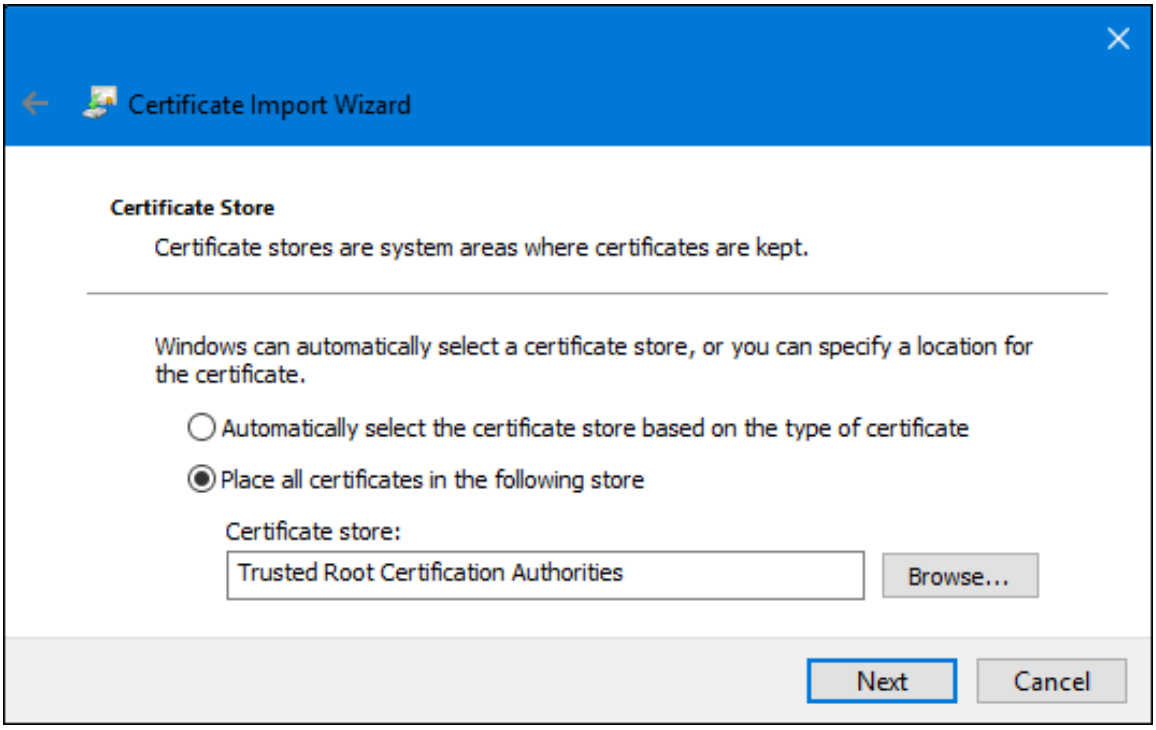
c. **Certificates** 項目を右クリックし、**Import** を選択します。



d. ダウンロードした Directory Server CA 証明書を参照し、**Next** をクリックします。



e. CA 証明書を **Trusted Root** 認証局 ストアに保存します。



7. ドメインコントローラーを再起動します。

サーバーが TLS で正しく実行されていることをテストするには、AD を LDAPS で検索してみてください。

16.4.3. ステップ 3: 同期 ID を選択または作成

Windows 同期の設定に使用するユーザーは 2 つあります。

- 同期合意で指定された AD ユーザー。

同期合意で指定されたユーザーは、Directory Server が AD にバインドして更新の送受信を行うエンティティーです。AD ユーザーは Domain Admins グループのメンバーであるか、同等の権限を持っている必要があります。また、ディレクトリーの変更を複製する権限が必要になります。

AD でユーザーを追加し、権限を設定する方法は、Microsoft のドキュメントを参照してください。

- Password Sync サービスで指定される Directory Server ユーザー。

  Password Sync サービスで参照されるユーザーは、同期サブツリー内のすべてのエントリーに対する読み取りおよび書き込みパーミッションを持っている必要があります。また、Password Sync がパスワード変更を更新できるように、Directory Server のパスワード属性への書き込みアクセス権が必要です。

注記

同期合意（サプライヤー DN）にユーザーが AD サーバーに存在する。Password Sync 設定にユーザーが Directory Server に存在する。

Directory Server で同期ユーザーを作成するには、以下を実行します。

1. パスワードで **cn=sync user,cn=config** などの新規エントリーを作成します。以下に例を示します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=sync user,cn=config
changetype: add
objectClass: inetorgperson
objectClass: person
objectClass: top
cn: sync user
sn: SU
userPassword: secret
passwordExpirationTime: 20380119031407Z
```

2. ユーザーパスワードの比較および書き込みのために同期ユーザーにアクセスを付与する ACI を設定します。

   ACI は、同期するサブツリーの上部に設定する必要があります。以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: ou=people,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="userPassword")(version 3.0;acl "password sync";allow
(write,compare) userdn="ldap:///cn=sync user,cn=config";)
```

   セキュリティー上の理由から、Password Sync ユーザーは Directory Manager ではなく、同期されたサブツリーに含めることはできません。

16.4.4. ステップ 4: パスワード同期サービスの インストール

Password Sync サービスのインストール手順は、『Red Hat Directory Server インストールガイド』の「パスワード同期サービス」のインストール」を参照してください。

Red Hat が Password Sync サービスの実行をサポートするオペレーティングシステムの一覧は、『『Red Hat Directory Server リリースノート』を参照してください』。

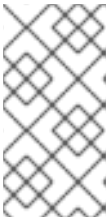
注記

パスワード同期アプリケーションのインストール時に発生するパスワードの同期を初めて試みます。これは、Password Sync の証明書データベースに CA 証明書が存在しないため、常に失敗します。CA 証明書の追加は、アプリケーションの設定ステップの一部です。

16.4.5. ステップ 5: パスワード同期サービス の設定

次に、Password Sync が TLS 経由で Directory Server にアクセスするために使用する証明書を設定します。

1. Directory Server で TLS を有効にします。詳細は、「Directory Server での TLS の有効化」を参照してください。



注記

Password Sync が Directory Server にパスワードを送信するには、TLS が必要です。サービスは、TLS 以外にパスワードを送信せず、Active Directory マシンから Directory Server マシンに送られるクリアテキストパスワードを保護します。つまり、Password Sync は TLS が設定されるまで機能しません。

2. Directory Server で、サーバー証明書をエクスポートします。

```
# certutil -d /usr/lib64/dirsrv/slapd-instance -L -n "CA certificate" -a > dsca.crt
```

3. エクスポートされた証明書を Directory Server から Windows マシンにコピーします。

4. Windows マシンでコマンドプロンプトを開き、**Password Sync** インストールディレクトリーを開きます。

```
> cd "C:\Program Files\Red Hat Directory Password Synchronization""
```

5. Windows マシンに **cert8.db** データベースおよび **key.db** データベースを作成します。

```
> certutil.exe -d . -N
```

6. Directory Server から新規証明書データベースにサーバー証明書をインポートします。

```
> certutil.exe -d . -A -n "DS CA cert" -t CT,, -a -i \path\to\dsca.crt
```

7. CA 証明書が正しくインポートされていることを確認します。

```
> certutil.exe -d . -L -n "DS CA cert"
```

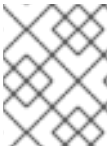
8. Windows マシンを再起動します。システムを再起動するまで、Password Sync サービスは利用できません。

注記

Password Sync が最初にインストール時に Active Directory ユーザーアカウントが存在する場合は、Password Sync が Active Directory でハッシュ化された後にパスワードを復号できないため、これらのユーザーアカウントのパスワードは変更されるまで同期できません。

16.4.6. ステップ 6: 同期用の Directory Server データベースの設定

レプリケーションと同様に、ディレクトリーの変更を追跡し、送信するには changelog が利用可能でなければなりません。また、同期する Directory Server データベースをレプリカとして設定する必要があります。



注記

Directory Server データベースがレプリケーション用に設定されている場合は、この手順は必要ありません。

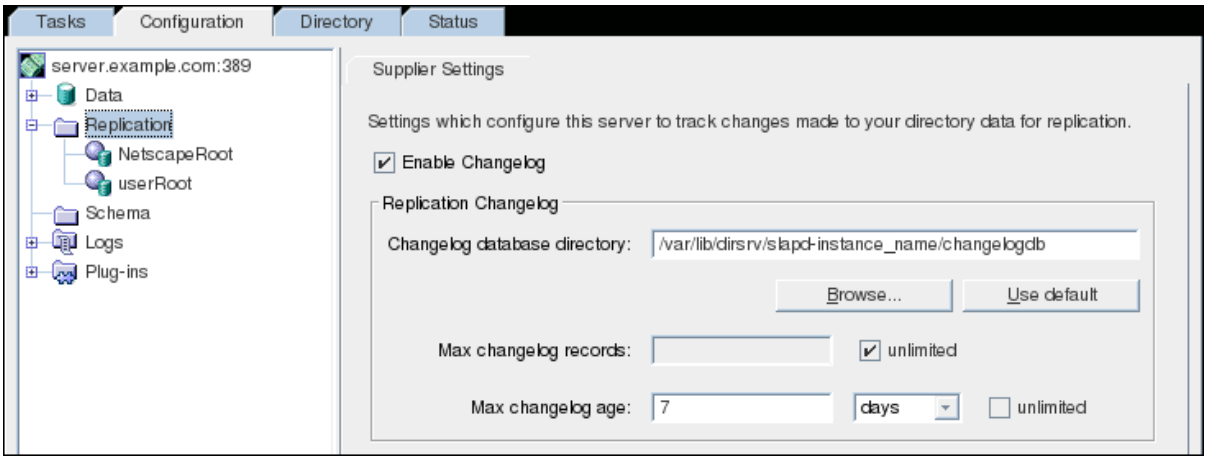
レプリケーション用のデータベースの設定は、「サプライヤーサーバーでの読み書きレプリカの設定」に記載されています。

16.4.6.1. コンソールからの同期用の Directory Server の設定

まず、changelog を有効にします。

1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のナビゲーションツリーで、**Replication** フォルダーをクリックします。
3. メインウィンドウで、**Supplier Settings** タブをクリックします。
4. **Enable Changelog** データベースを確認します。

図16.3 Configuration タブ



5. changelog データベースディレクトリーを設定します。**Use default** ボタンをクリックして、デフォルトまたは **Browse...** を使用してカスタムディレクトリーを選択します。
6. changelog 設定を保存します。

changelog の設定後に、レプリカとして同期するデータベースを設定します。レプリカロールは、単一マスターまたはマルチマスターレプリケーションのいずれかである必要があります。

重要

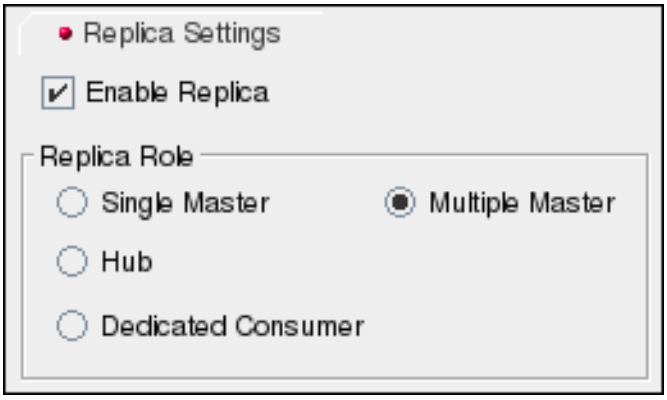
ハブサーバーで同期合意を設定することは可能ですが、Red Hat Directory Server から AD への一方向の同期のみが許可されます。AD サーバーは、変更内容をハブに同期できません。

同期合意を設定するために、マルチマスターレプリケーションのマスターのみを使用することが強く推奨されます。

1. Directory Server コンソールで、**Configuration** タブを選択します。
2. 左側のナビゲーションツリーで **Replication** フォルダーをクリックし、同期するデータベースの名前をクリックします。

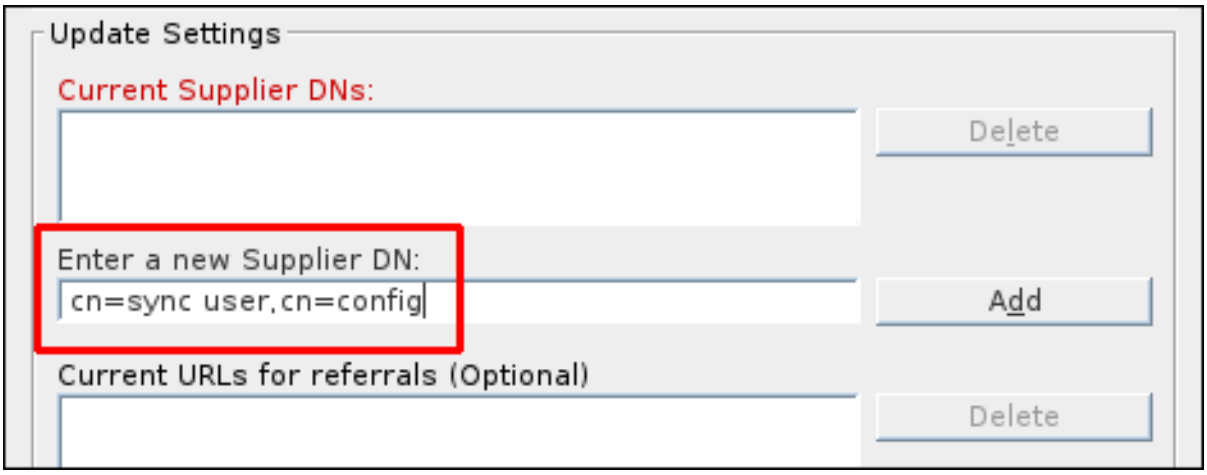
   デフォルトでは、ディレクトリー設定用の **NetscapeRoot** とディレクトリーエントリー用の **userRoot** の 2 つのデータベースがあります。Directory Server に追加されているその他のデータベースは、一覧表示できます。
3. **Enable Replica** チェックボックスを選択し、データベースが存在するレプリカのタイプでラジオボタンを選択します。

   図16.4 **Enable Replica** チェックボックス



4. **Update Settings** セクションで、サプライヤー DN を選択または追加します。これは、同期プロセスが実行されるユーザーアカウントです。「ステップ 3: 同期 ID を選択または作成」で説明されているように、このユーザーは Directory Server 上に配置し、同期される全ユーザーの **userPassword** 属性に対するアクセス権が必要です。

図16.5 Update Settings セクション



5. データベースのレプリケーション設定を保存します。

注記

レプリケーション設定の詳細は、15章*レプリケーションの管理*を参照してください。

16.4.6.2. コマンドラインからの同期用の Directory Server の設定

まず、changelog を有効にします。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=changelog5,cn=config
changetype: add
objectclass: top
objectclass: extensibleObject
cn: changelog5
nsslapd-changelogdir: /var/lib/dirsrv/slapd-instance_name/changelogdb
nsslapd-changelogmaxage: 7d
```

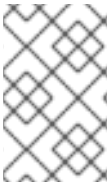
次に、サプライヤーレプリカエントリーを作成します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=sync replica,cn="dc=example,dc=com",cn=mapping tree,cn=config
changetype: add
objectclass: top
objectclass: nsds5replica
objectclass: extensibleObject
cn: sync replica
nsds5replicaroot: dc=example,dc=com
nsds5replicaid: 7
nsds5replicatype: 3
nsds5flags: 1
nsds5ReplicaPurgeDelay: 604800
nsds5ReplicaBindDN: cn=sync user,cn=config
```

これらのパラメーターは、設定、『コマンド、およびファイルリファレンス および』「コマンドラインでのサプライヤーの設定」で詳細に説明されています。

16.4.7. ステップ 7: 同期合意の作成

同期合意を作成します。



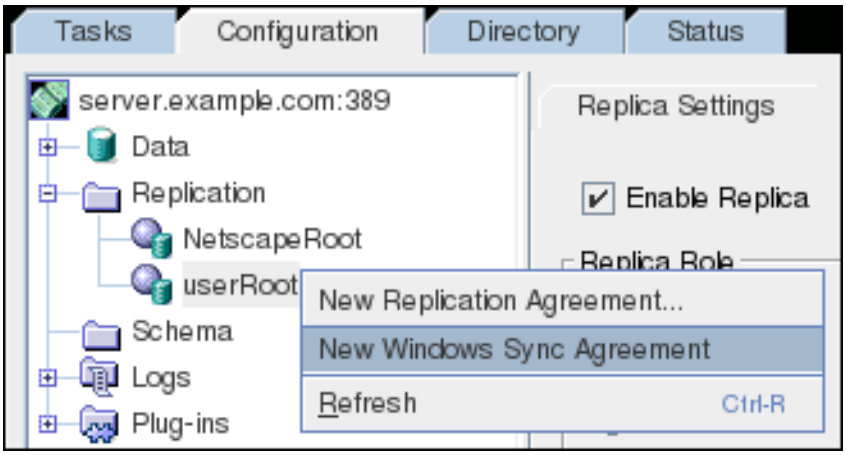
注記

シンプルなパスワード認証(「セキュアなバインドの要求」)にセキュアなバインドが必要な場合は、セキュアな接続で行われる場合を除き、レプリケーション操作は失敗します。セキュアな接続（LDAPS または StartTLS）の使用が推奨されます。

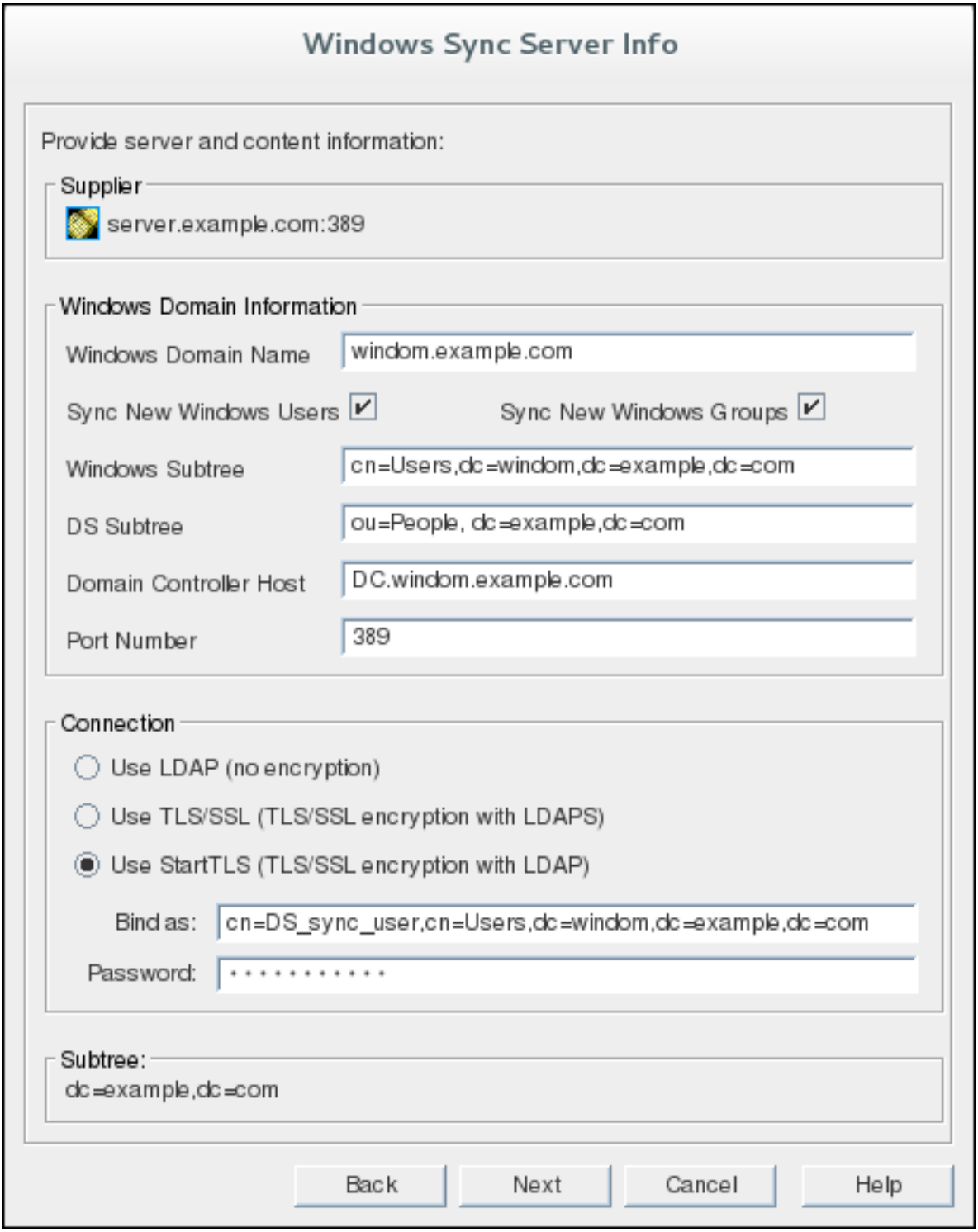
16.4.7.1. コンソールからの同期合意の作成

1. **Directory Server コンソールで、Configuration タブを選択します。**
2. **左側のナビゲーションツリーで Replication をクリックし、同期するデータベースを右クリックします。デフォルトのユーザーデータベースは userRoot ですが、Directory Server に新しい接尾辞が追加されるため、追加のデータベースが追加されます。**

   または、データベースを強調表示し、トップツールバーで Object をクリックします。
3. **メニューから New Windows Synchronization Agreement を選択します。**



4. **2 つのフィールドに、同期合意の名前と説明を指定します。Next を押します。**
5. **Windows Sync Server Info ウィンドウで、Windows Domain Information エリアに AD 情報を入力します。**



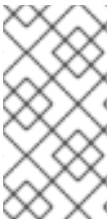
- Windows ドメインの名前。
- 同期するエントリーの種類。ユーザーおよびグループは個別に同期されます。エントリーのタイプを選択すると、Windows サブツリーにあるそのタイプのエントリーがすべて Directory Server に作成されます。
- Windows および Directory Server のサブツリー情報。これは自動的に入力されます。
- ドメインコントローラーのホスト名、IPv4 アドレス、または IPv6 アドレス
- Windows サーバーのポート番号

6. 接続タイプを設定します。以下の 3 つのオプションがあります。

- LDAP を使用します。これにより、標準の暗号化されていない接続が設定されます。
- TLS/SSL を使用します。これは、636 などのサーバーのセキュアな LDAPS ポートを介したセキュアな接続を使用します。Directory Server と Windows サーバーの両方が、この接続に対して TLS で実行されるよう適切に設定し、サーバー証明書を信頼するために相互の CA 証明書をインストールする必要があります。
- Start TLS を使用します。Start TLS を使用して、サーバーの標準ポートでセキュアな接続を確立します。通常の TLS と同様に、これらのピアサーバーは相互の証明書を信頼できる必要があります。

セキュリティー上の理由から、TLS または Start TLS のいずれかを使用することが推奨されます。AD は、接続が TLS で保護されない限り、パスワードの同期に TLS または Start TLS が必要です。

7. **Bind as...** および **Password** フィールドに同期 ID 情報を入力します。このユーザーは AD に存在している必要があります。
8. 同期合意を保存します。



注記

デフォルトでは、Windows Synchronization は AD ピアを 5 分ごとにポーリングして変更の有無をチェックします。同期合意の概要では、これは **Update Interval** として表示されます。更新間隔は、***winSyncInterval*** 属性を手動で編集することで変更できます。「コマンドラインでの同期合意の追加および編集」 を参照してください。

この合意が完了すると、新しい同期合意が接尾辞の下に一覧表示されます。

16.4.7.2. コマンドラインからの同期契約の作成

コマンドラインで同期合意を追加することもできます。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=replication_agreement_name,cn=replica,cn="dc=example,dc=com",cn=mapping
tree,cn=config
changetype: add
objectclass: top
objectclass: nsDSWindowsReplicationAgreement
cn: replication_agreement_name
nsds7WindowsReplicaSubtree: cn=Users,dc=ad1
nsds7DirectoryReplicaSubtree: ou=People,dc=example,dc=com
nsds7NewWinUserSyncEnabled: on
nsds7NewWinGroupSyncEnabled: on
nsds7WindowsDomain: ad1
nsDS5ReplicaRoot: dc=example,dc=com
nsDS5ReplicaHost: ad1.windows-server.com
nsDS5ReplicaPort: 389
nsDS5ReplicaBindDN: cn=sync user,cn=config
nsDS5ReplicaCredentials: {DES}ffGad646dT0nnsT8nJOaMA==
nsDS5ReplicaTransportInfo: TLS
winSyncInterval: 1200
```

例で使用するパラメーターと設定可能なその他の属性の説明は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス を参照してください』。

16.4.8. ステップ 8: 同期用の Directory Server ユーザーとグループエントリーの設定

ntUser オブジェクトクラスと ntGroup オブジェクトクラスを、それぞれ必要な属性とともに同期されるユーザーエントリーおよびグループエントリーに追加します。これらのオブジェクトクラスを持つ Directory Server エントリーのみが同期されます。Directory Server に同期される AD エントリーには、これらのオブジェクトクラスが自動的に含まれます。

新しいエントリーと既存のエントリーの両方に対して、適切なオブジェクトクラスがエントリーに追加されるたびに、エントリーは次の増分更新で同期されます。

同期用の Directory Server ユーザーエントリーの設定については「Directory Server ユーザーのユーザー同期の設定」に、同期用の Directory Server グループエントリーの設定については「Directory Server グループのグループ同期の設定」で説明しています。

16.4.9. ステップ 9: 同期の開始

同期合意を作成したら、同期を開始します。

コマンドラインでの同期の開始

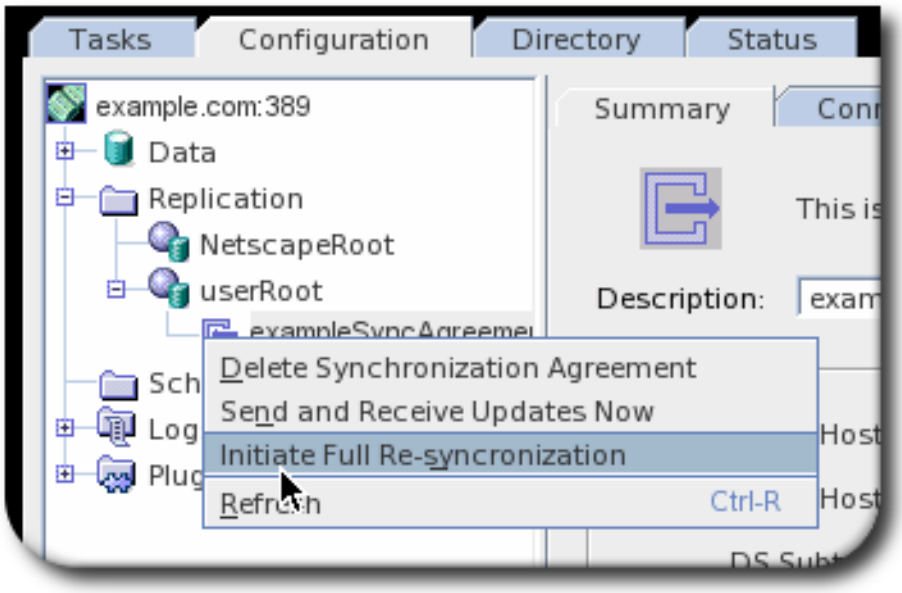
コマンドラインで同期を開始するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=replication_agreement_name,cn=replica,cn="dc=example,dc=com",cn=mapping
tree,cn=config
changetype: modify
replace: nsds5beginreplicarefresh
nsds5beginreplicarefresh: start
```

初期化が完了すると、Directory Server はレプリカ合意エントリーから *nsds5BeginReplicaRefresh* 属性を自動的に削除します。

コンソールを使用した同期の開始

1. コンソールの Configuration タブに移動します。
2. Replication フォルダーを開き、適切なデータベースを展開します。
3. 同期合意を選択します。
4. 合意を右クリックしたり、Object メニューを開きます。
5. Initiate Full Re-synchronization を選択します。



何らかの理由で同期が停止した場合には、同期合意メニューからこれを選択して、別の合計更新（再同期）を開始します。合計更新（再同期）を開始すると、データベースの削除や上書きは行われません。

16.5. ユーザーの同期

ユーザーは、Directory Server と Active Directory 間で自動的に同期されません。両方向の同期を設定する必要があります。

- Active Directory ドメインのユーザーは、**Sync New Windows Users** オプションを選択し、同期合意に設定すると同期されます。同期が開始すると、すべての Windows ユーザーが Directory Server にコピーされ、その後、新しいユーザーが作成されると、そのユーザーが同期されます。

- Directory Server のユーザーアカウントは、Directory Server エントリーにある特定の属性を使用して Active Directory と同期されます。Directory Server エントリーには、**ntUser** オブジェクトクラスと ***ntUserCreateNewAccount*** 属性が必要です。***ntUserCreateNewAccount*** 属性 (既存のエントリーでも) は、Active Directory サーバーにエントリーを書き込むように Directory Server Windows Synchronization プラグインに通知します。

  ntUser オブジェクトクラスが追加された新規または変更されたユーザーエントリーが作成され、エントリーの標準的なポーリングである次の定期更新時に Windows マシンに同期されます。

注記

Active Directory ドメインでは、パスワードが使用されるまでユーザーがアクティブではありません。既存のユーザーが必要な Windows 属性を持つように変更されると、そのユーザーエントリーは Active Directory ドメインに同期されますが、Directory Server 側でパスワードが変更されるか、管理者が Active Directory にパスワードを設定するまでログインできません。これは、Directory Server に保存されているパスワードが暗号化され、Password Sync はすでに暗号化されたパスワードを同期できないためです。

Active Directory ドメインでユーザーを有効にするには、ユーザーのパスワードをリセットします。

Directory Server で同期されたエントリーは、それが Directory Server で発生したものであるか、Active Directory で発生したものであるかに関わらず、すべて特別な同期属性を持っています。

- ntUserDomainId.これは、Active Directory エントリーの ***sAMAccountName*** 属性に対応します。
- ntUniqueId.これには、対応する Windows エントリーの ***objectGUID*** 属性の値が含まれます。この属性は同期プロセスで設定され、手動で設定または変更しないでください。
- ntUserDeleteAccount.この属性は、Windows エントリーが同期され、Directory Server エントリーに対して手動で設定する必要がある場合に自動的に設定されます。***ntUserDeleteAccount*** の値が true であれば、Directory Server エントリーが削除された場合に対応する Windows エントリーが削除されます。それ以外の場合、エントリーは Active Directory のままになりますが、Directory Server で削除されている場合は Directory Server データベースから削除されます。

Directory Server エントリーで ***ntUserCreateNewAccount*** および ***ntUserDeleteAccount*** を設定すると、Directory Manager では、同期されたサブツリー内のどのユーザーが Active Directory で同期されるかを正確に制御できます。

16.5.1. Directory Server と Active Directory との間で同期されるユーザー属性

Directory Server 属性および Active Directory 属性のサブセットのみが同期されます。これらの属性はハードコーディングされ、エントリーの同期方法に関わらず定義されます。Directory Server または Active Directory のいずれかにあるエントリーにあるその他の属性は、同期の影響を受けないままになります。

Directory Server および Active Directory で使用される属性の一部は同一です。これは通常、すべての LDAP サービスに共通する LDAP 標準で定義された属性です。これらの属性は、相互に正確に同期されます。表16.2「Directory Server および Windows サーバーで同一のユーザースキーマ」は、Directory Server と Windows サーバーとの間で同じ属性を示しています。

同じ情報を定義する属性もありますが、属性やスキーマ定義の名前が異なります。これらの属性は Active Directory と Directory Server の間でマッピングされるため、1 つのサーバーの属性 A がもう 1 つのサーバーの属性 B として扱われます。同期の場合、これらの属性の多くは Windows 固有の情報に関連します。表16.1「Directory Server と Active Directory との間でマッピングされるユーザースキーマ」は、Directory Server と Windows サーバーとの間で同じ属性を示しています。

Directory Server および Active Directory が一部のスキーマ要素を処理する方法の違いについての詳細は、「Red Hat Directory Server と Active Directory との間のユーザースキーマの相違点」を参照してください。

表16.1 Directory Server と Active Directory との間でマッピングされるユーザースキーマ

Directory Server	Active Directory
cn[a]	name
ntUserDomainId	sAMAccountName
ntUserHomeDir	homeDirectory
ntUserScriptPath	scriptPath
ntUserLastLogon	lastLogon

Directory Server	Active Directory
ntUserLastLogoff	lastLogoff
ntUserAcctExpires	accountExpires
ntUserCodePage	codePage
ntUserLogonHours	logonHours
ntUserMaxStorage	maxStorage
ntUserProfile	profilePath
ntUserParms	userParameters
ntUserWorkstations	userWorkstations

[a] ***cn*** は、他の同期属性とは異なる方法で処理されます。Directory Server から Active Directory に同期する際に、直接 (***cn*** から ***cn***へ) マッピングされます。ただし、Active Directory から Directory Server に同期する場合、 ***cn*** は Windows の ***name*** 属性から Directory Server の ***cn*** 属性にマッピングされます。

表16.2 Directory Server および Windows サーバーで同一のユーザースキーマ

cn[a]	physicalDeliveryOfficeName
description	postOfficeBox
destinationIndicator	postalAddress
facsimileTelephoneNumber	postalCode
givenname	registeredAddress
homePhone	sn
homePostalAddress	st
initials	street
l	telephoneNumber
mail	teletexTerminalIdentifier
mobile	telexNumber

o	title
ou	usercertificate
pager	x121Address

[a] ***cn*** は、他の同期属性とは異なる方法で処理されます。Directory Server から Active Directory に同期する際に、直接 (***cn*** から ***cn***へ) マッピングされます。ただし、Active Directory から Directory Server に同期する場合、 ***cn*** は Windows の ***name*** 属性から Directory Server の ***cn*** 属性にマッピングされます。

16.5.2. Red Hat Directory Server と Active Directory との間のユーザースキーマの相違点

Active Directory は Directory Server と同じ基本的な X.500 オブジェクトクラスをサポートしますが、管理者が認識すべき非互換性がいくつかあります。

16.5.2.1. cn 属性の値

Directory Server では、 ***cn*** 属性に複数の値を指定できますが、Active Directory ではこの属性に単一の値しか持たせません。Directory Server の ***cn*** 属性が同期されると、単一の値のみが Active Directory ピアに送信されます。

これは、同期の意味としては、 ***cn*** の値が Active Directory エントリーに追加され、その値が Directory Server の ***cn*** の値のいずれでもない場合、Directory Server の ***cn*** 値がすべて単一の Active Directory 値で上書きされます。

もう 1 つの重要な相違点として、Active Directory は ***cn*** 属性を命名属性として使用するのに対し、Directory Server では ***uid*** を使用する点があります。つまり、 ***cn*** 属性が Directory Server で編集されると、エントリーの名前が完全に (誤って) 変更される可能性があります。この ***cn*** の変更が Active Directory エントリーに書き込まれると、エントリーの名前が変更になり、新しい名前付きエントリーが Directory Server に書き込まれます。

16.5.2.2. パスワードポリシー

Active Directory と Directory Server の両方は、パスワードの最小長や最大期間などのパスワードポリシーを強制できます。Windows 同期を使用すると、ポリシーの一貫性、強制、同期がなくなります。Directory Server と Active Directory の両方においてパスワードポリシーの一貫性がないため、他のシステムと同期すると、システムに加えられたパスワードの変更が失敗する可能性があります。Directory Server におけるデフォルトのパスワード構文設定は、Active Directory が実施するデフォルトのパスワードの複雑さルールに準拠します。

16.5.2.3. street および streetAddress の値

Active Directory は、ユーザーまたはグループの住所に ***streetAddress*** 属性を使用します。これは、Directory Server が ***street*** 属性を使用する方法です。Active Directory および Directory Server が ***streetAddress*** 属性および ***street*** 属性を使用する方法には 2 つの重要な相違点があります。

- Directory Server では、 ***streetAddress*** は ***street*** のエイリアスです。Active Directory にも ***street*** 属性がありますが、 ***streetAddress*** のエイリアスではなく、独立した値を保持することができる個別の属性です。

- Active Directory は ***streetAddress*** と ***street*** を単一値の属性として定義しますが、Directory Server は RFC 4519 で指定されるように ***street*** を多値属性として定義します。

Directory Server および Active Directory が ***streetAddress*** および ***street*** 属性を処理する方法が異なるため、Active Directory および Directory Server で address 属性を設定する際に従う 2 つのルールがあります。

- Windows 同期は、Windows エントリーの ***streetAddress*** を Directory Server の ***street*** にマッピングします。競合を回避するために、***street*** 属性は Active Directory では使用しないようにしてください。

- 1 つの Directory Server ***street*** 属性値のみが Active Directory に同期されます。***streetAddress*** 属性が Active Directory で変更され、新しい値が Directory Server に存在しない場合は、Directory Server のすべての ***street*** 属性値が新しい Active Directory 値に置き換えられます。

16.5.2.4. initials 属性の制約

initials 属性では、Active Directory は最大長 6 文字の制限を課しますが、Directory Server には長さ制限がありません。6 文字を超える ***initials*** 属性が Directory Server に追加されると、その値は Active Directory エントリーと同期したときにトリミングされます。

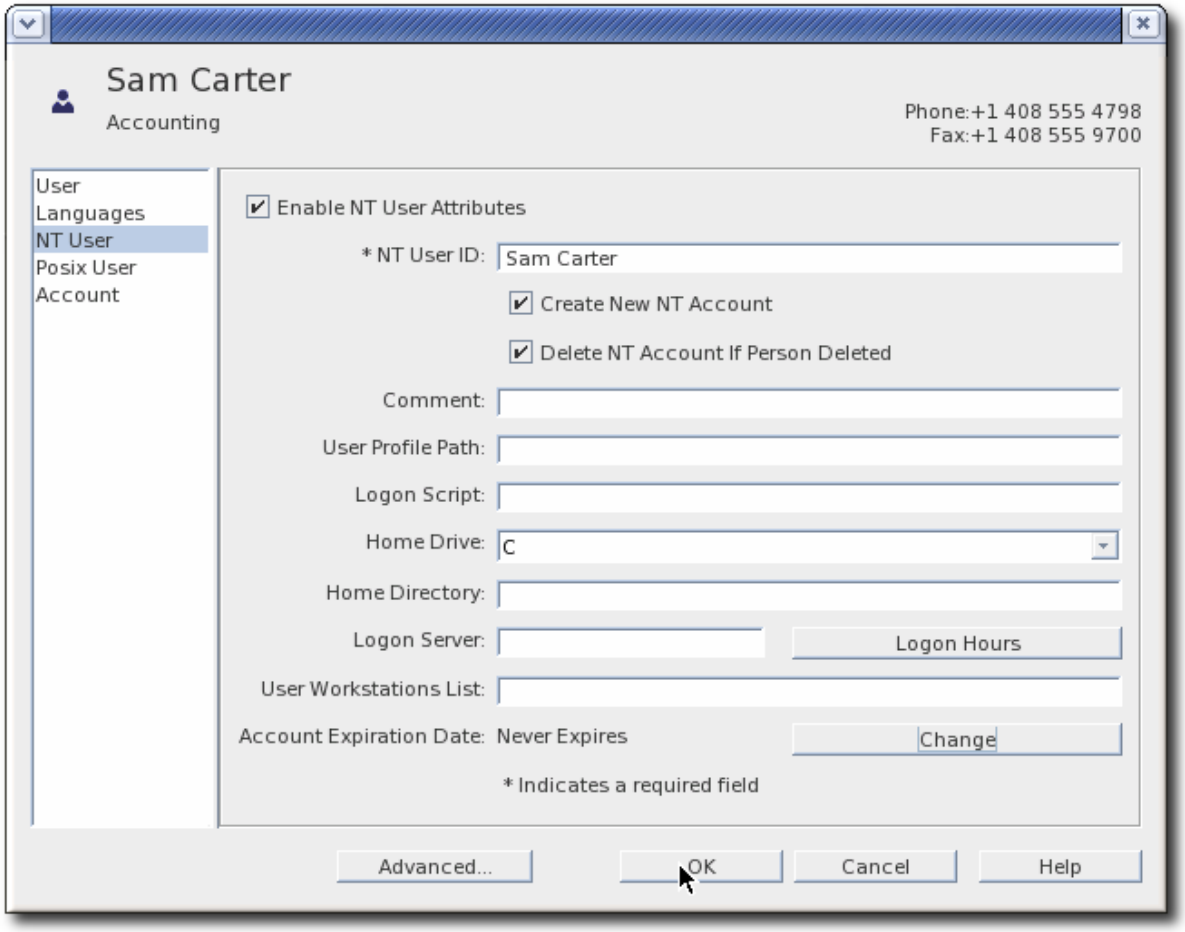
16.5.3. Directory Server ユーザーのユーザー同期の設定

Directory Server ユーザーが Active Directory に同期するには、ユーザーエントリーに適切な同期属性を設定する必要があります。

16.5.3.1. コンソールでのユーザー同期の設定

1. Directory Server コンソールで、Directory タブを選択します。
2. 既存のエントリーでエントリーを右クリックし、**Properties** をクリックしてエントリーのプロパティーエディターを開きます。

   新しいエントリーについては、左側のウィンドウのメインエントリーを右クリックして、新しいエントリーを追加し、**User** を選択して必要なエントリー属性を入力します。
3. **Property Editor** の左側で、**NT User** リンクをクリックします。
4. **NT User** タブで、**NT 属性の有効化** チェックボックスを選択します。



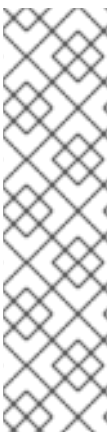
5. 同期を有効にするには、2つのフィールドが必要です。

   - NT ユーザー IDの設定
   - 「Create New NT Account 」チェックボックスの選択

6. Delete NT Account チェックボックスを選択すると、Directory Server エントリーが削除された場合に対応する Windows ユーザーが削除されることを意味します。

7. 他の Windows 属性を設定します。これらの属性は、関連する Windows 属性にマッピングされます。

   追加の ntUser 属性は Advanced ボタンを使用して作成できます。「ディレクトリーエントリーの更新」 を参照してください。



注記

ユーザーのパスワードをリセットします。

Active Directory ドメインでは、パスワードが使用されるまでユーザーがアクティブではありません。既存のユーザーが必要な Windows 属性を持つように変更されると、そのユーザーエントリーは Active Directory ドメインに同期されますが、Directory Server 側でパスワードが変更されるか、管理者が Active Directory にパスワードを設定するまでログインできません。Password Sync は、暗号化したパスワードを同期できません。

したがって、Active Directory ドメインでユーザーをアクティブにするには、ユーザーのパスワードをリセットします。

16.5.3.2. コマンドラインでのユーザー同期の設定

コマンドラインで同期を有効にするには、必要な同期属性をエントリーに追加するか、これらの属性でエントリーを作成します。

同期には、以下の 3 つのスキーマ要素が必要です。

- ntUser オブジェクトクラス。
- Windows ID を指定する *ntUserDomainId* 属性
- 同期プラグインに Active Directory 経由で Directory Server エントリーを同期するように通知する *ntUserCreateNewAccount* 属性

たとえば、以下のように ldapmodify ユーティリティーを使用します。

```
dn: uid=scarter,ou=People,dc=example,dc=com
changetype: modify
add: objectClass
objectClass:ntUser
-
add: ntUserDomainId
ntUserDomainId: Sam Carter
-
add: ntUserCreateNewAccount
ntUserCreateNewAccount: true
-
add: ntUserDeleteAccount
ntUserDeleteAccount: true
```

エントリーに、さらに多くの Windows およびユーザー属性を追加できます。同期されたスキーマは、「Directory Server と Active Directory との間で同期されるユーザー属性」にすべてリストされます。ntUser オブジェクトクラスに属する Windows 固有の属性は、『Red Hat Directory Server 10 Configuration, Command, and File Reference』 で説明されています。

注記

ユーザーのパスワードをリセットします。

Active Directory ドメインでは、パスワードが使用されるまでユーザーがアクティブではありません。既存のユーザーが必要な Windows 属性を持つように変更されると、そのユーザーエントリーは Active Directory ドメインに同期されますが、Directory Server 側でパスワードが変更されるか、管理者が Active Directory にパスワードを設定するまでログインできません。Password Sync は、暗号化したパスワードを同期できません。

したがって、Active Directory ドメインでユーザーをアクティブにするには、ユーザーのパスワードをリセットします。

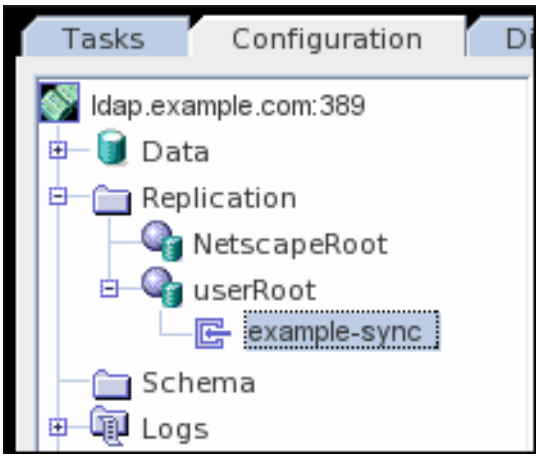
16.5.4. Active Directory ユーザーのユーザー同期の設定

Windows ユーザー (Active Directory ドメインにあるユーザー) の同期は、同期合意で設定されます。

16.5.4.1. コンソールでのユーザー同期の設定

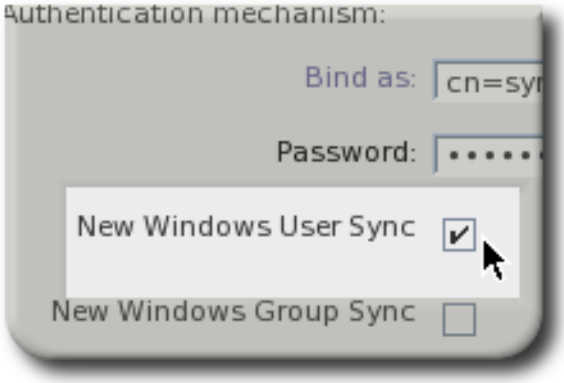
1. Configuration タブを開き、Replication フォルダーを展開します。

2. 適切なデータベースを開き、同期合意を選択します。



3. **Connection** タブを開きます。

4. **New Windows User Sync** チェックボックスにチェックを入れて、ユーザーの同期を有効にします。同期を無効にするには、チェックボックスの選択を解除します。



新しい同期合意については、同期合意の作成ウィザードで、対応するユーザー同期のチェックボックスを選択します。

16.5.4.2. コマンドラインでのユーザー同期の設定

Active Directory ユーザー同期を設定する属性は ***nsds7NewWinUserSyncEnabled*** で、同期合意に設定されます。ユーザーの同期を有効にするには、この属性を同期合意に追加するか、**ldapmodify** を使用して、この属性を **on** に設定して同期合意を作成します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=replication_agreement_name,cn=replica,cn="dc=example,dc=com",cn=mapping
tree,cn=config
changetype: modify
replace: nsds7NewWinUserSyncEnabled
nsds7NewWinUserSyncEnabled: on
```

ユーザーの同期を無効にするには、***nsds7NewWinUserSyncEnabled: off*** を設定します。

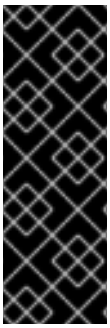
16.6. グループの同期

ユーザーエントリーと同様に、グループは Directory Server と Active Directory の間で自動的に同期されません。両方向の同期を設定する必要があります。

- 同期合意で設定されている場合、Active Directory ドメインのグループは、新規 **Windows** グループの同期 オプションを選択すると同期されます。同期を開始すると、すべての Windows グループが Directory Server にコピーされ、新規グループは作成時に同期されます。

- Directory Server のグループアカウントは、Directory Server エントリーにある特定の属性を使用して Active Directory と同期します。Directory Server エントリーには、**ntGroup** オブジェクトクラスと ***ntGroupCreateNewGroup*** 属性が必要です。***ntGroupCreateNewGroup*** 属性 (既存のエントリーでも) は、Active Directory サーバーにエントリーを書き込むように Directory Server Windows Synchronization に通知します。

  ntGroup オブジェクトクラスを持つ新規または変更されたグループが作成され、次の通常の更新時に Windows マシンと同期されます。



重要

グループを同期すると、そのメンバーの一覧も同期されます。ただし、ユーザー同期が有効で、これらのエントリーに適用する限り、メンバーエントリー自体は同期されません。

これにより、アプリケーションやサービスが Active Directory サーバー上のグループのすべてのメンバーに対して修正操作を行おうとしたときに、それらのユーザーの一部が存在しない場合に問題が発生する可能性があります。

また、グループには、その他の一般的な属性がいくつかあります。

- Active Directory では、Directory Server グループが作成/削除されるかどうかを制御する 2 つの属性 (***ntGroupCreateNewGroup*** および ***ntGroupDeleteGroup***) を制御します。

  ntGroupCreateNewGroup は、Active Directory に Directory Server グループを同期するために必要です。

- ntUserDomainId には、Active Directory ドメインのエントリーの一意の ID が含まれます。これは、**ntGroup** オブジェクトクラスの唯一の必須属性です。

- ntGroupType は Windows グループのタイプです。Windows のグループタイプには、global/security、domain local/security、builtin、universal/security、global/distribution、domain local/distribution、universal/distribution があります。この属性は、同期をとる Windows グループには自動的に設定されますが、Directory Server エントリーには、同期をとる前にこの属性を手動で設定する必要があります。

16.6.1. Windows グループタイプの概要

Active Directory には、セキュリティーとディストリビューションの 2 つの主要なグループタイプがあります。セキュリティーグループは、アクセス制御、リソースの制限、およびその他のパーミッションに対してポリシーを設定することができるため、Directory Server のグループには最も似ています。配信グループは、メール配信のためのグループです。これはさらに、グローバルグループおよびローカルグループに分けられます。Directory Server ntGroupType は、以下の 4 つのグループタイプをすべてサポートします。

- グローバル/セキュリティーの場合 (デフォルト) は **-2147483646**

- ドメインローカル/セキュリティーの場合は **-2147483644**

- 組み込みの場合は -2147483643
- 汎用/セキュリティーの場合は -2147483640
- グローバル/ディストリビューションの場合は 2
- ドメインローカル/ディストリビューションの場合は 4
- ユニバーサル/ディストリビューションの場合は 8

16.6.2. Directory Server と Active Directory との間で同期されるグループ属性

Directory Server 属性および Active Directory 属性のサブセットのみが同期されます。これらの属性はハードコーディングされ、エントリーの同期方法に関わらず定義されます。Directory Server または Active Directory のいずれかにあるエントリーにあるその他の属性は、同期の影響を受けないままになります。

Directory Server エントリーおよび Active Directory グループエントリーで使用される属性の一部は同一です。これは通常、すべての LDAP サービスに共通する LDAP 標準で定義された属性です。これらの属性は、相互に正確に同期されます。表16.4「Directory Server と Active Directory との間のグループエントリー属性」は、Directory Server と Windows サーバーとの間で同じ属性を示しています。

同じ情報を定義する属性もありますが、属性やスキーマ定義の名前が異なります。これらの属性は Active Directory と Directory Server の間でマッピングされるため、1 つのサーバーの属性 A がもう 1 つのサーバーの属性 B として扱われます。同期の場合、これらの属性の多くは Windows 固有の情報に関連します。表16.3「Directory Server と Active Directory との間のグループエントリー属性のマッピング」は、Directory Server と Windows サーバーとの間で同じ属性を示しています。

Directory Server および Active Directory が一部のスキーマ要素を処理する方法の違いについての詳細は、「Red Hat Directory Server と Active Directory のグループスキーマの相違点」を参照してください。

表16.3 Directory Server と Active Directory との間のグループエントリー属性のマッピング

Directory Server	Active Directory
cn	name
ntUserDomainID	name
ntGroupType	groupType
uniqueMember member	メンバー[a]

[a] Active Directory の ***Member*** 属性は、Directory Server の ***uniqueMember*** 属性に同期されます。

表16.4 Directory Server と Active Directory との間のグループエントリー属性

cn	o
description	ou
l	seeAlso
mail	

16.6.3. Red Hat Directory Server と Active Directory のグループスキーマの相違点

Active Directory は Directory Server と同じ基本的な X.500 オブジェクトクラスをサポートしますが、管理者が認識すべき非互換性がいくつかあります。

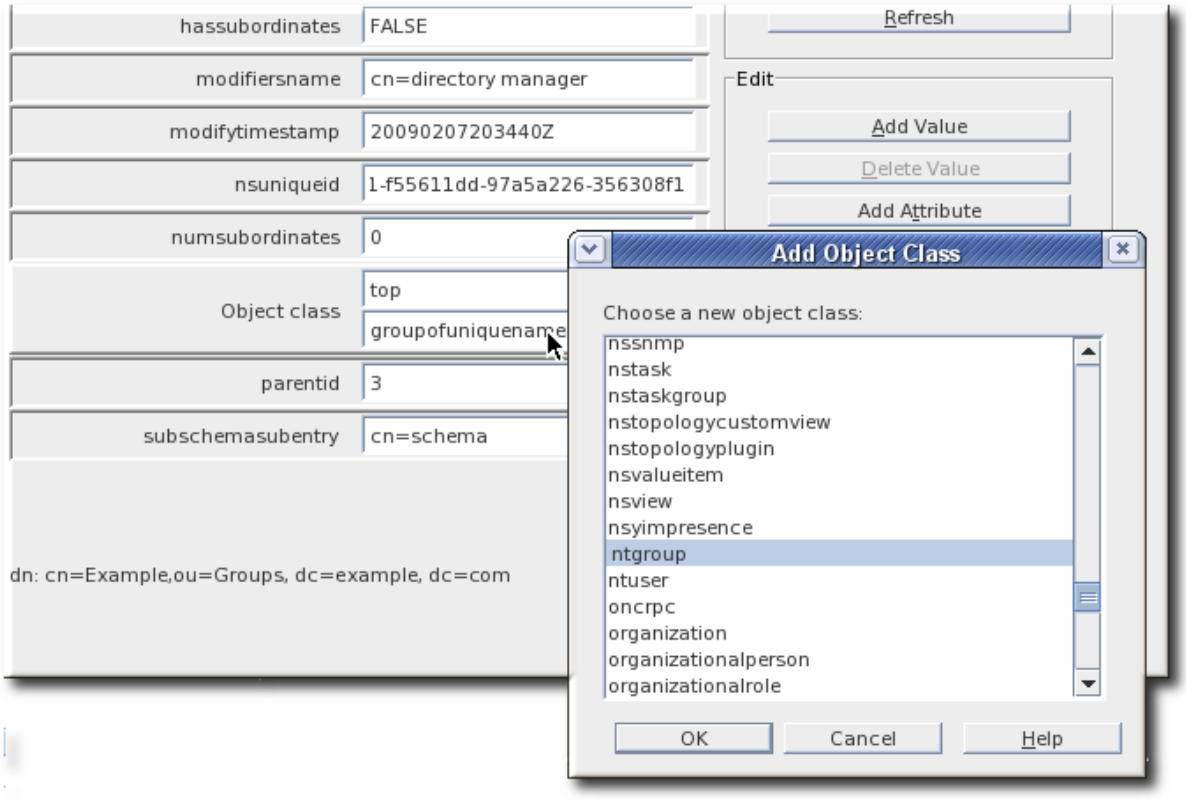
ネスト化されたグループ (グループに別のグループをメンバーとして追加) がサポートされ、Windows Synchronization では同期します。ただし、Active Directory では、ネストされたグループの構成として特定の制約が適用されます。たとえば、グローバルグループには、ドメインローカルグループをメンバーとして追加することはできません。Directory Server にはローカルグループとグローバルグループの概念がないため、同期時に Active Directory の制約に違反する Directory Server 側でエントリーを作成できます。

16.6.4. Directory Server グループのグループ同期の設定

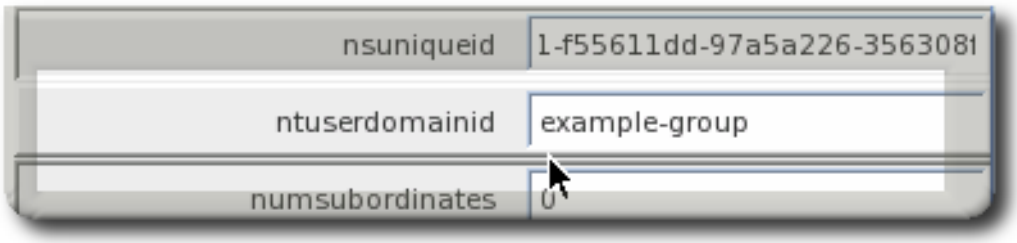
Directory Server グループが Active Directory に同期するには、グループエントリーに適切な同期属性を設定する必要があります。

16.6.4.1. コンソールでのグループ同期の設定

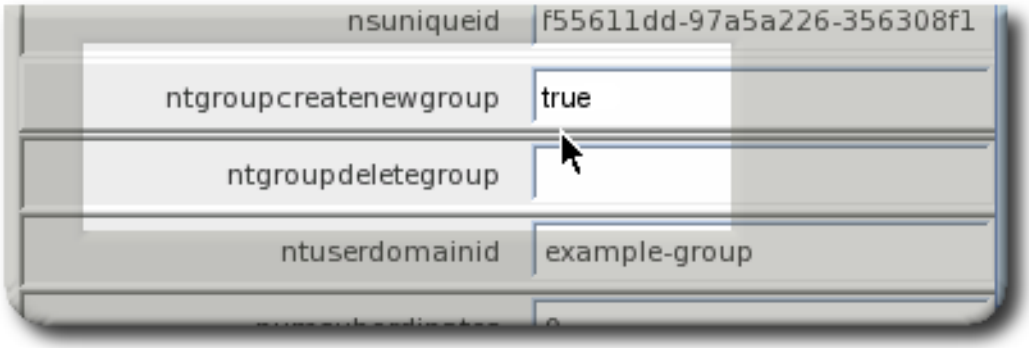
1. Directory Server コンソールで、Directory タブを選択します。
2. **グループエントリーを右クリックし、Advanced をクリックして、エントリーの高度なプロパティーエディターを開きます。同期関連の属性はすべて手動で追加する必要があるため、高度なプロパティーエディターのみが属性を設定できます。**
3. objectclasses フィールド をクリックしてから、**Add Value** ボタンをクリックします。
4. **ntGroup オブジェクトクラス** を選択します。



5. ntGroup オブジェクトクラス を設定すると、*ntUserDomainId* 属性が自動的に追加されます。この属性は必須となるため、値を追加します。



6. 同期を有効にするには、**Add Attribute** ボタンをクリックし、一覧から *ntGroupCreateNewGroup* 属性を選択します。次に、その値を **true** に設定します。これは、エントリーを Active Directory ディレクトリーに追加する必要がある同期プラグインに信号を送ります。



Directory Server データベースからグループエントリーが削除された場合に Active Directory ドメインからグループエントリーを削除するには、*ntGroupDeleteGroup* 属性を設定して **true** に設定します。

7. Directory Server エントリーの他の Windows 属性を追加します。利用可能な属性は、「Directory Server と Active Directory との間で同期されるグループ属性」に記載されています。

   ntGroupType が追加されない場合は、グループはグローバルセキュリティーグループ (ntGroupType:-2147483646)として自動的に追加されます。

16.6.4.2. コマンドラインでのグループ同期の設定

コマンドラインで同期を有効にするには、必要な同期属性をエントリーに追加するか、これらの属性でエントリーを作成します。

同期には、以下の 3 つのスキーマ要素が必要です。

- ntGroup オブジェクトクラス。
- エントリーの Windows ID を与える *ntUserDomainId* 属性。
- *ntGroupCreateNewGroup* 属性は、同期プラグインに Active Directory 経由で Directory Server エントリーを同期するように通知します。

  ntGroupDeleteGroup 属性は任意ですが、Directory Server で削除される場合に、自動的に Active Directory ドメインからエントリーを削除するかどうかを設定します。

また、*ntGroupType* 属性を追加することも推奨されます。この属性が指定されていない場合、グループはグローバルセキュリティーグループ (ntGroupType:-2147483646) として自動的に追加されます。

たとえば、ldapmodify を使用するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=Example Group,ou=Groups,dc=example,dc=com
changetype: modify
add: objectClass
objectClass:ntGroup
-
add: ntUserDomainId
ntUserDomainId: example-group
-
add: ntGroupCreateNewGroup
ntGroupCreateNewGroup: true
-
add: ntGroupDeleteGroup
ntGroupDeleteGroup: true
-
add: ntGroupType
ntGroupType: 2
```

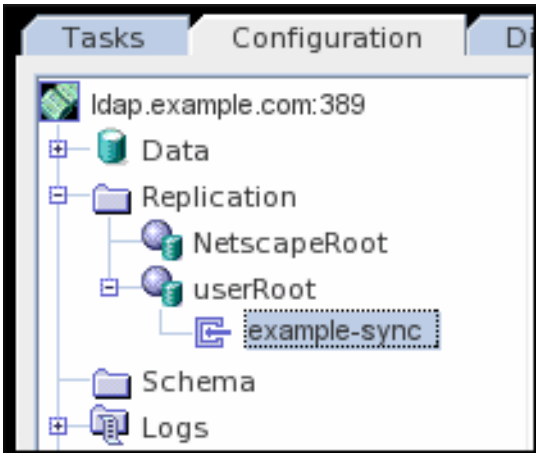
エントリーには、多くの Windows やグループの属性を追加することができます。同期されたスキーマは、「Directory Server と Active Directory との間で同期されるグループ属性」にすべてリストされます。ntGroup オブジェクトクラス に属する Windows 固有の属性は、『Red Hat Directory Server 10 Configuration, Command, and File Reference』 で説明されています。

16.6.5. Active Directory グループのグループ同期の設定

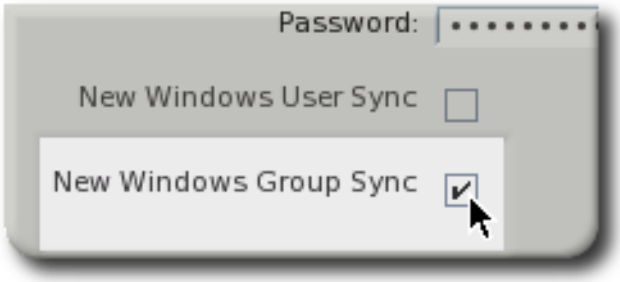
Windows ユーザー (Active Directory ドメインにあるユーザー) の同期は、同期合意で設定されます。

16.6.5.1. コンソールでのグループ同期の設定

1. **Configuration タブを開き、Replication フォルダーを展開します。**
2. **適切なデータベースを開き、同期合意を選択します。**



3. **Connection タブを開きます。**
4. **新規 Windows Group Sync チェックボックスを選択して、グループ同期を有効にします。同期を無効にするには、チェックボックスの選択を解除します。**



新しい同期合意については、同期合意の作成ウィザードで対応するグループ同期 チェックボックスを選択します。

16.6.5.2. コマンドラインでのグループ同期の設定

Active Directory グループ同期を設定する属性は *nsds7NewWinGroupSyncEnabled* で、同期合意に設定されます。グループ同期を有効にするには、この属性を同期合意に追加するか、この属性が on に設定された同期合意を作成します。ldapmodify の使用：

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=replication_agreement_name,cn=replica,cn="dc=example,dc=com",cn=mapping
tree,cn=config
changetype: modify
replace: nsds7NewWinGroupSyncEnabled
nsds7NewWinGroupSyncEnabled: on
```

グループ同期を無効にするには、nsds7NewWinGroupSyncEnabled: off を設定します。

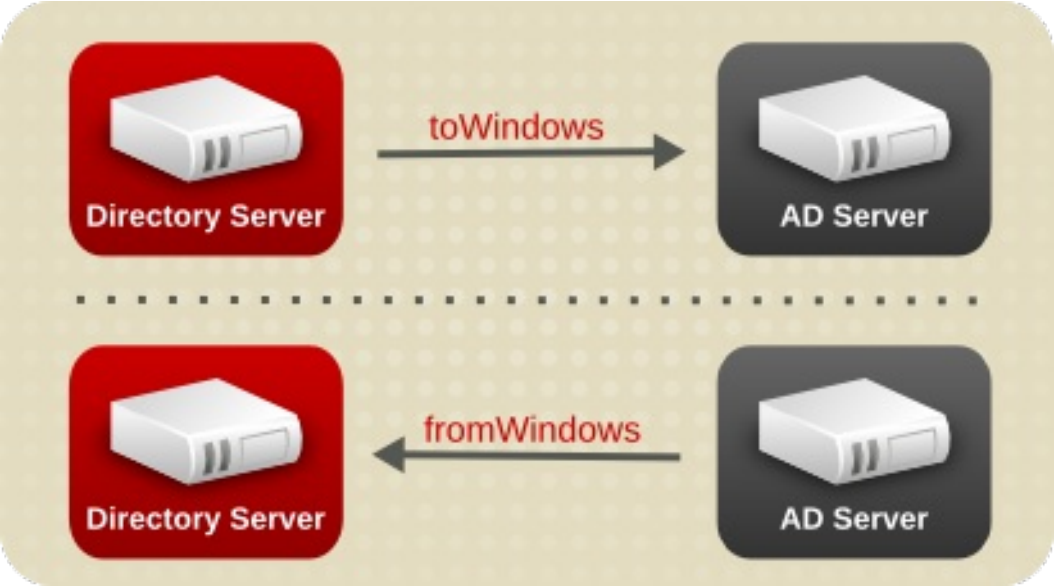
16.7. 一方向の同期の設定

図16.1「Active Directory - Directory Server の同期プロセス」に示すように、同期はデフォルトで 双方向 となります。つまり、Active Directory の変更が Directory Server に送信され、Directory Server の変更が Active Directory に送信されることを意味します。

変更が一方向のみに送信される場合は、一方向同期 を作成できます。これはマスターとコンシューマーの関係と似ています。[2] マルチマスターとは対照的です。

同期合意の追加属性 ***oneWaySync*** は、一方向の同期を有効にし、変更を送信する方向を指定します。使用できる値は ***fromWindows*** (Active Directory から Directory Server への同期の場合) および ***toWindows*** (Directory Server から Active Directory の同期の場合) です。この属性がない場合、同期は双方向になります。

図16.6 一方向の同期



同期プロセス自体は、双方向と一方向の同期にほぼ同じです。これは、同じ同期間隔と設定を使用します。唯一の違いは、同期情報の要求方法にあります。

Windows Active Directory から Directory Server への同期では、定期的な同期の更新間隔で、Directory Server が Active Directory Server に接続し、DirSync コントロールを送信して更新を要求します。ただし、Directory Server は、その側から変更やエントリーは送信されません。つまり、同期更新は、Active Directory の変更内容が Directory Server のエントリーに送信され、更新されることで構成されています。

Directory Server から Active Directory 同期では、Directory Server は通常の更新で Active Directory サーバーにエントリー変更を送信しますが、Active Directory 側から更新を要求しないように DirSync 制御は含まれません。

一方向の同期を有効にするには、以下を実行します。

1. 「ステップ 7: 同期合意の作成」 にあるように、同期合意を作成します。
2. Directory Server コンソールには、合意の初回作成時に一方向同期を設定するオプションはありません。同期合意を編集して、***oneWaySync*** 属性を追加します。ldapmodify の使用：

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=replication_agreement_name,cn=replica,cn="dc=example,dc=com",cn=mapping
tree,cn=config
```

```
changetype: modify
add: oneWaySync
oneWaySync: fromWindows
```

注記

一方向同期を有効にすると、同期されていないサーバーで自動的に変更ができなくなるわけではないため、同期更新間の同期ピア間で不整合が生じる可能性があります。たとえば、一方向性同期は、Active Directory から Directory Server に行くように設定されているため、Active Directory が (実質的に) データマスターとなります。Directory Server でエントリーを変更または削除すると、Directory Server 情報はその情報とは異なるため、これらの変更は Active Directory に引き継がれません。次の同期更新時に、編集内容は Directory Server で上書きされ、削除済みのエントリーが再追加されます。

データの不整合が発生するのを防ぐには、アクセス制御ルールを使用して、同期 されていない サーバーの同期サブツリー内のエントリーを編集または削除しないようにします。Directory Server のアクセス制御については、「18章 *アクセス制御の管理*」で説明しています。Active Directory の場合は、適切な Windows ドキュメントを参照してください。

一方向の同期はパスワードの同期には影響しません。*oneWaySync* が *toWindows* に設定されている場合でも、Active Directory サーバーでパスワードを更新した後に、パスワードは Directory Server に送信されます。

16.8. WINDOWS 同期での複数のサブツリーおよびフィルターの設定

Windows Synchronization は、Directory Server (DS) と Active Directory (AD) のサブツリーの複数のペアとの間で同期するように作られています。フィルターを使用すると、サブツリーの配下にあるエントリーのみが同期されます。

Windows 同期における複数のサブツリー

複数のサブツリーのペア間で同期するには、Windows 同期合意の **winSyncSubtreePair** パラメーターに Directory Server サブツリーと Active Directory サブツリーを設定します。**ldapmodify** を使用して、以下のように複数のサブツリーを設定します。

```
changetype: modify
add: winSyncSubtreePair
winSyncSubtreePair: ou=OU1,dc=DSexample,dc=com:ou=OU1,DC=ADexample,DC=com
```

winSyncSubtreePair が設定されていない場合は、代わりに **nsds7WindowsReplicaSubtree** AD サブツリーパラメーターと **nsds7DirectoryReplicaSubtree** DS サブツリーパラメーターが同期ターゲットチェックに使用されます。それ以外の場合は、この 2 つのパラメーターは無視されます。

Windows 同期のフィルター

以下のパラメーターで同期されるデータを選択するフィルターを設定できます。

- **winSyncWindowsFilter** は、Active Directory サーバーに追加のフィルターを設定します。
- **winSyncDirectoryFilter** パラメーターは、Directory Server に追加のフィルターを設定します。

以下の例では、**ldapmodify** は CN にユーザーまたはグループ が含まれるエントリーを同期するために使用されます。

```
changetype: modify
add: winSyncWindowsFilter
winSyncWindowsFilter: (|(cn=*user*)(cn=*group*))
-
add: winSyncDirectoryFilter
winSyncDirectoryFilter: (|(uid=*user*)(cn=*group*))
```

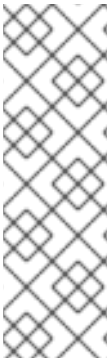
16.9. ユーザーとグループの POSIX 属性の同期

すべての可能なユーザーと属性のサブセットが、Active Directory と Red Hat Directory Server の間で同期されます。一部の属性はマッピングされ、Active Directory と Directory Server スキーマには違いがあり、一部の属性が直接照合されます。同期される属性 (一致およびマップされた)「Directory Server と Active Directory との間で同期されるユーザー属性」および「Directory Server と Active Directory との間で同期されるグループ属性」に一覧表示されます。

デフォルトでは、これらの属性のみが同期されます。

その同期一覧にない属性のタイプの 1 つは、POSIX 関連の属性です。Linux システムでは、システムユーザーおよびグループは POSIX エントリーとして識別され、LDAP POSIX 属性に必要な情報が含まれています。しかし、Windows ユーザーが同期すると、Windows アカウントであることを示す *ntUser* 属性および *ntGroup* 属性が自動的に追加されますが、POSIX 属性は同期されず (Active Directory エントリーに存在していても)、Directory Server 側でも POSIX 属性は追加されません。

POSIX Winsync API プラグインは、Active Directory エントリーと Directory Server エントリーとの間で POSIX 属性を同期します。



注記

すべての POSIX 属性 (*uidNumber*、*gidNumber*、および *homeDirectory*) は、Active Directory エントリーと Directory Server エントリー間で同期されます。ただし、新しい POSIX エントリーまたは POSIX 属性が Directory Server の既存のエントリーに追加されると、POSIX 属性のみが Active Directory に対応するエントリーと同期します。POSIX オブジェクトクラス (ユーザーの場合は posixAccount、グループの場合は posixGroup) は Active Directory エントリーに追加されません。

16.9.1. POSIX 属性同期の有効化

Posix Winsync API プラグインはデフォルトで無効になっており、Active Directory ユーザーおよびグループのエントリーから対応する Directory Server エントリーに同期するように POSIX 属性に対して有効にする必要があります。

1. *nsslapd-pluginEnabled* 属性を on に設定します。

```
ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=Posix Winsync API,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

注記

Posix 同期プラグインが最初に読み込まれるように、優先順位は 50 未満である必要があります。デフォルトの設定では、優先順位は 25 であり、この値はほとんどのデプロイメントで同じままになる可能性があります。

2. Directory Server を再起動して、新しい構成を読み込みます。

16.9.2. Posix グループ属性の同期設定の変更

POSIX グループの属性とグループメンバーを Active Directory エントリーから対応する Directory Server のグループエントリーおよびユーザーエントリに同期する方法を制御するために、複数のプラグイン属性を設定することができます。詳細は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス の該当するセクションを参照してください』。

デフォルト設定はほとんどのデプロイメントに使用できますが、Active Directory 環境に応じて設定を変更できます。たとえば、ネスト化されたグループマッピングを有効にするには、次のコマンドを実行します。

1. ldapmodify を使用して、属性を適切な設定に変更します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=Posix Winsync API,cn=plugins,cn=config
changetype: modify
replace: posixWinsyncMapNestedGrouping
posixWinsyncMapNestedGrouping: true
```

2. Directory Server を再起動して、新しい構成を読み込みます。

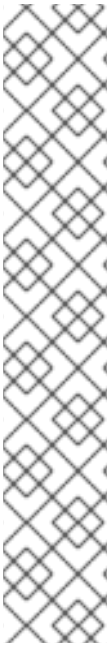
16.10. エントリーの削除および復元

ここでは、同期を有効にすることで、同期先の削除されたエントリーにどのような影響があるか、また復活したエントリーがどのように処理されるかについて説明します。

16.10.1. エントリーの削除

Active Directory ピアのすべての変更は、常に Directory Server と同期されます。つまり、Active Directory ドメイン上で Active Directory グループやユーザーアカウントが削除されると、その削除内容が Directory Server の同期ピアサーバーに自動的に同期して戻ってくるということです。

一方、Directory Server では、Directory Server アカウントが削除されると、Directory Server エントリーに *ntUserDeleteAccount* 属性または *ntGroupDeleteGroup* 属性が true に設定されている場合のみ、Active Directory の対応するエントリーが削除されます。



注記

Directory Server エントリーが Active Directory に初めて同期すると、Active Directory は自動的に一意の ID を割り当てます。次の同期間隔で、一意の ID が Directory Server エントリーに同期され、 *ntUniqueId* 属性として保存されます。一意の ID が Directory Server に同期する 前 に Active Directory で Directory Server エントリーを削除すると、このエントリーは Directory Server で削除 されません。Directory Server は *ntUniqueId* 属性を使用して、Active Directory に追加された変更を対応する Directory Server エントリーに識別し、同期します。その属性がないと、Directory Server は削除を認識しません。

Active Directory のエントリーを削除し、Directory Server で削除を同期するには、*ntUniqueId* 属性が削除される前に、エントリーの作成後に *winSyncInterval* の長さ (デフォルトでは 5 分) 待ちます。

16.10.2. エントリーのレスキュー

削除済みのエントリーを Directory Server に戻すことができます。削除されたエントリーは tombstone エントリーと呼ばれます。Directory Server と Active Directory の間で同期されていた削除エントリーが Directory Server に再び追加されると、再開する Directory Server エントリーには元の属性と値がすべて含まれます。これは tombstone reanimation と呼ばれます。再取得されたエントリーには、エントリーの同期に使用された元の *ntUniqueId* 属性が含まれます。これは、この新規エントリーが tombstone エントリーである Active Directory サーバーに通知されます。

Active Directory は古いエントリーを再取得し、エントリーの元の一意の ID を保持します。

Active Directory エントリーの場合、tombstone エントリーが Directory Server 上で復活すると、元の Directory Server の属性がすべて保持され、復活した Active Directory エントリーにも含まれます。

16.11. 同期更新の送信

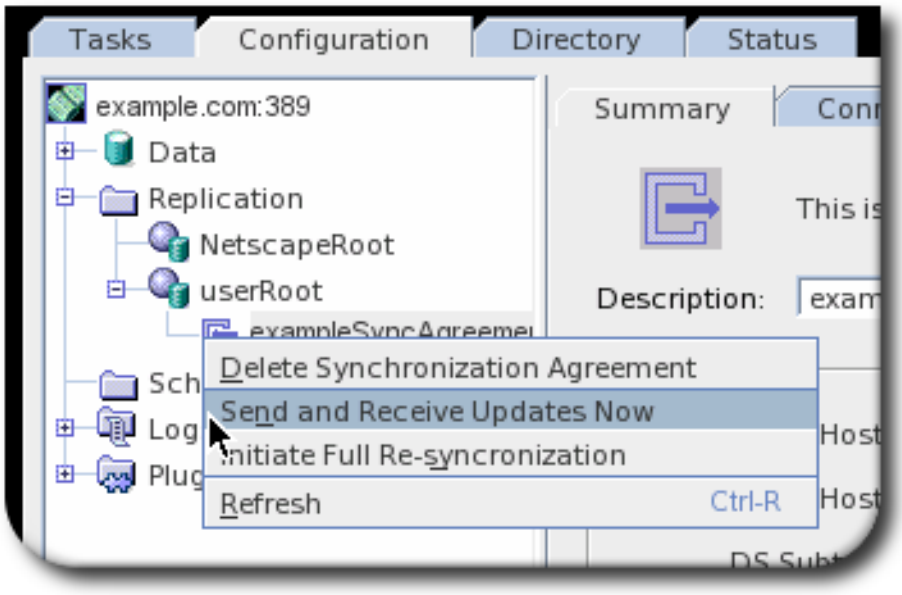
同期は、*winSyncInterval* (Active Directory ドメインから変更内容を取得する場合) または*nsds5replicaupdateschedule* 設定 (Directory Server から変更内容をプッシュする場合) で設定された頻度で行われます。デフォルトでは、変更は 5 分ごとに Active Directory から取得され、Directory Server からの変更がすぐに送信されます。

同期の更新は手動でトリガーできます。また、完全な再同期を行うことも可能で、Directory Server と Active Directory のすべてのエントリーを、あたかも新しいもののように送信したり、引き出したりすることができます。完全な再同期には、以前同期されていない既存の Directory Server エントリーが含まれます。

16.11.1. 手動増分同期の実行

通常の操作時に、Active Directory に送信する必要のある Directory Server のエントリーに追加された更新はすべて changelog を収集し、増分更新時に再生されます。

1. コンソールの Configuration タブに移動します。
2. Replication フォルダーを開き、適切なデータベースを展開します。
3. 同期合意を選択します。
4. 合意を右クリックしたり、Object メニューを開きます。
5. ドロップダウンメニューから Send and Receive Updates を選択します。



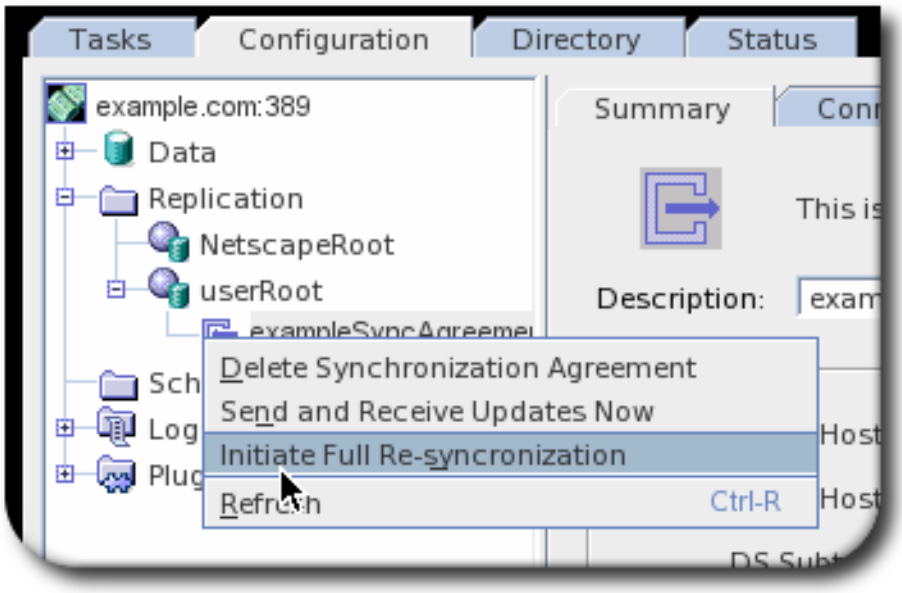
16.11.2. 完全同期の実行

データに大きな変更があった場合や、既存の Directory Server エントリーに同期属性を追加する場合は、再同期 を開始する必要があります。再同期は合計更新であり、同期されたサブツリーのコンテンツ全体が検証され、必要に応じて更新されます。再同期は changelog を使用せずに行われます。これは、レプリケーションのコンシューマーの初期化または再初期化に似ています。

16.11.2.1. コンソールを使用した完全同期の実行

完全同期を実行するには、以下を実行します。

1. コンソールの Configuration タブに移動します。
2. Replication フォルダーを開き、適切なデータベースを展開します。
3. 同期合意を選択します。
4. 合意を右クリックしたり、Object メニューを開きます。
5. ドロップダウンメニューから Initialize Full Re-synchronization を選択します。



Resynchronizing は、同期ピアのデータを削除しません。すべての更新を送受信して、新規または変更した Directory Server エントリーを追加します。たとえば、ntUser オブジェクトクラスが追加された既存の Directory Server ユーザーを追加します。

16.11.2.2. コマンドラインを使用した完全同期の実行

コマンドラインを使用して完全な同期を開始するには、**start** 値を持つ *nsDS5BeginReplicaRefresh* 属性を同期合意に追加します。

たとえば、サンプル 合意で完全同期を開始するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=replication_agreement_name,cn=replica,cn="dc=example,dc=com",cn=mapping
tree,cn=config
changetype: modify
add: nsDS5BeginReplicaRefresh
nsDS5BeginReplicaRefresh: start
```

同期後、Directory Server は合意エントリーから *nsDS5BeginReplicaRefresh* 属性を自動的に削除します。

16.11.3. 同期ステータスの確認

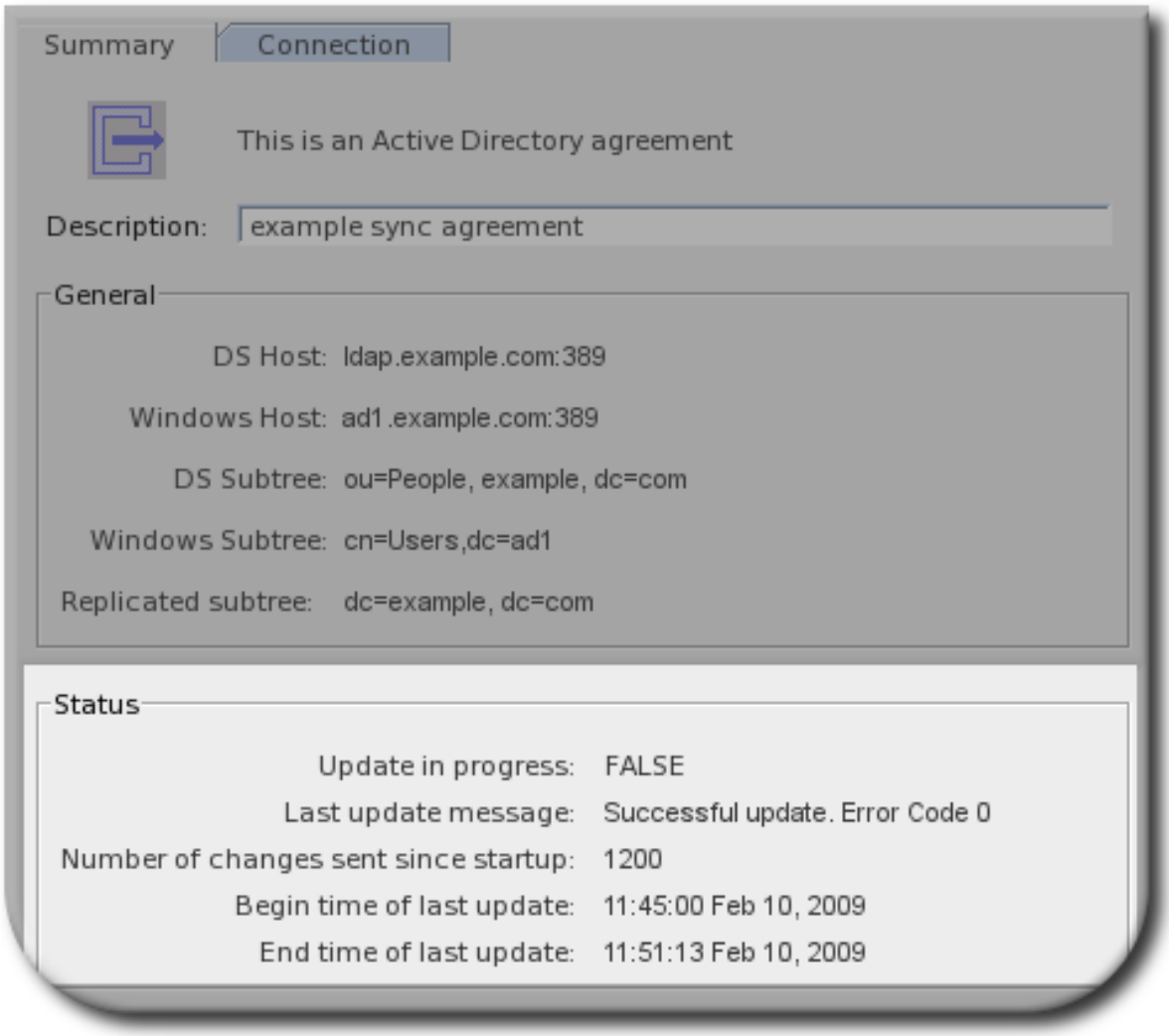
コンソールの **Status** の **Replication** タブで同期のステータスを確認します。モニターする同期合意を強調表示し、適切な情報が右側のペインに表示されるはずです。**Status** エリアは、最後の増分および合計の更新が成功したかどうか、および発生したタイミングを表示します。

1. コンソールの **Configuration** タブに移動します。

2. **Replication** フォルダーを開き、適切なデータベースを展開します。

3. 同期合意を選択します。

4. **Summary** タブで、最新の同期プロセスのステータスが下部に表示されます。



16.12. 同期合意の変更

接続情報を含む、同期合意の特定属性を変更できます。コマンドラインで使用すると、同期間隔の変更や同期スケジュールの設定など、多くの追加パラメーターを同期合意に作成できます。

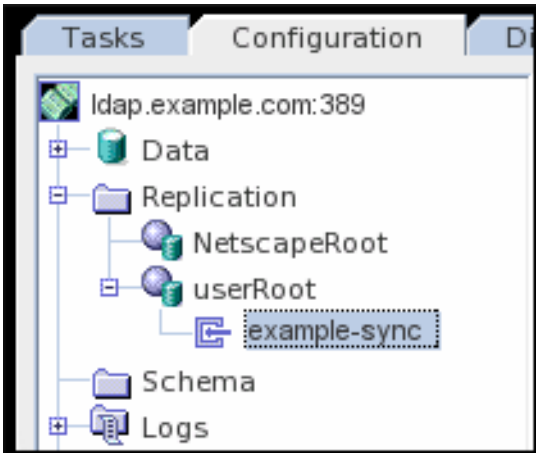
16.12.1. コンソールでの同期合意の編集

コンソールで編集できる情報のほとんどは、使用するプロトコルやバインド認証情報など、接続情報に限定されます。同期合意の説明を編集することもできます。

1. Configuration タブで、Replication フォルダーを展開します。

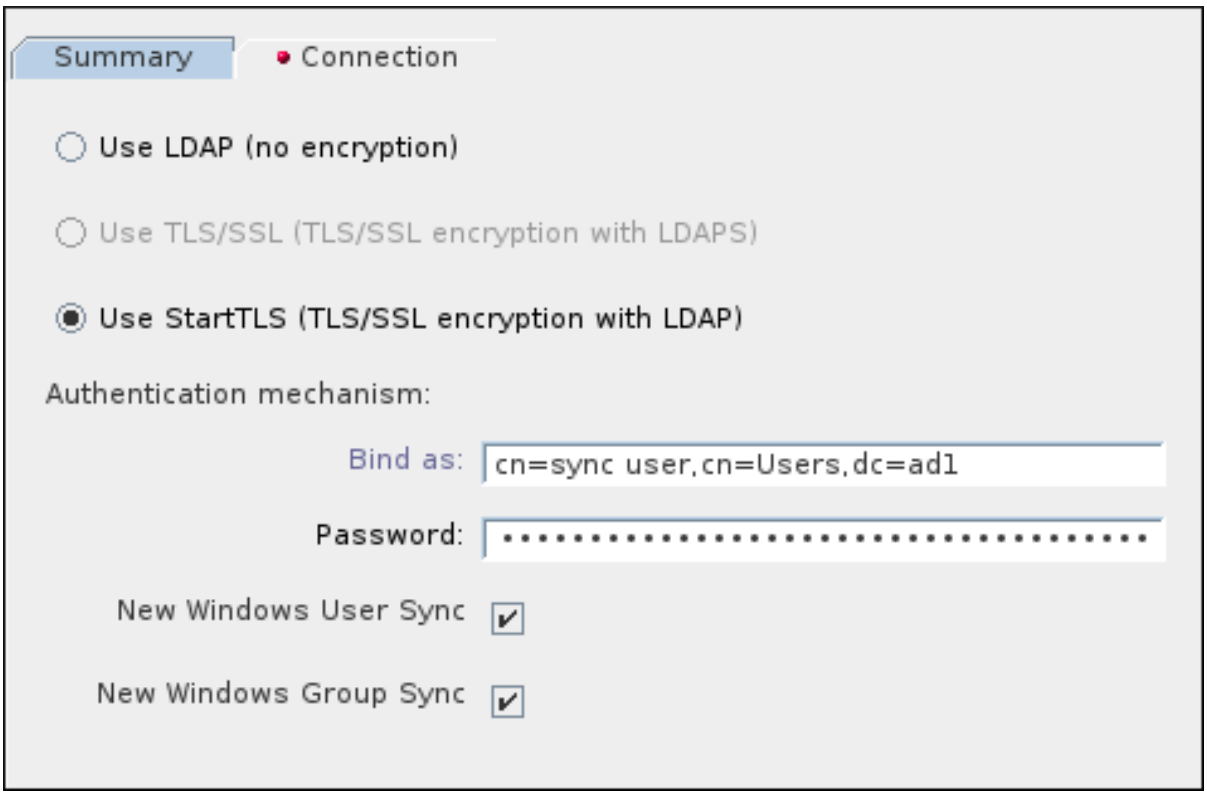
2. 同期されるデータベースを展開します。同期合意はすべて、データベースの下に一覧表示されます。同期合意をダブルクリックして、メインウィンドウで開きます。

図16.7 同期合意の選択



3. Connection タブをクリックします。

図16.8 Connection タブ



編集できる情報の 3 つの項目があります。

- 接続タイプ（標準、TLS、および Start TLS）
- バインドユーザー（DN およびパスワードの両方）。
- 新しい Directory Server ユーザーおよび新しい Directory Server グループを自動的に同期するかどうか。

接続タイプには、standard、TLS、および Start TLS の 3 つのオプションがありますが、実際には LDAP および LDAPS の接続プロトコルが 2 つしかありません。標準接続と Start TLS 接続はいずれも LDAP を使用します（Start TLS は、非セキュアなポートでセキュアな接続を作成します）。

接続プロトコルは 、 Windows 同期ピアへの接続に使用するポート番号を変更することができないため、接続プロトコルを変更できません。

標準接続と Start TLS の間で接続タイプを変更できますが、TLS から標準または Start TLS 接続のいずれかに変更することはできません。同様に、標準または Start TLS から TLS に移動することはできません。接続プロトコルまたはポート番号を変更する必要がある場合は、同期合意を削除し、新しいアカウントを作成します。

16.12.2. コマンドラインでの同期合意の追加および編集

コマンドラインで同期合意を作成または編集すると、より柔軟であり、Directory Server コンソールを使用する場合よりも多くのオプションが提供されます。同期合意属性の完全なリストは、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンスの該当するセクション』 に記載されています。

16.12.2.1. Basic 同期合意の作成

最も基本的な同期合意は、Directory Server データベースと Active Directory 同期ピアを定義します。

- Directory Server データベースの場合：

- ディレクトリーの同期されたサブツリー(*nsds7DirectoryReplicaSubtree*)
- Directory Server のルート DN(*nsDS5ReplicaRoot*)

- Active Directory ドメインの場合：
  - Active Directory ドメインで同期されたサブツリー (*nsds7WindowsReplicaSubtree*)
  - Active Directory ドメイン名(*nsds7WindowsDomain*)

また、Active Directory ドメインにバインドするために Directory Server が使用する接続情報も定義します。

- Active Directory ホスト名、IPv4 アドレス、または IPv6 アドレス(*nsDS5ReplicaHost*)
- Active Directory ポート(*nsDS5ReplicaPort*)
- 標準(LDAP)、TLS(SSL)、またはStartTLS(TLS)の接続の種類（標準ポートを介したセキュアな接続）です。*nsDS5ReplicaTransportInfo*
- Active Directory サーバーにバインドする Directory Server のユーザー名 (*nsDS5ReplicaBindDN*)およびパスワード(*nsDS5ReplicaCredentials*)。

たとえば、ldapmodify を使用するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=replication_agreement_name,cn=replica,cn="dc=example,dc=com",cn=mapping
tree,cn=config
changetype: add
objectclass: top
objectclass: nsDSWindowsReplicationAgreement
cn: replication_agreement_name
nsds7WindowsReplicaSubtree: cn=Users,dc=ad1
nsds7DirectoryReplicaSubtree: ou=People,dc=example,dc=com
```

```
nsds7WindowsDomain: ad1
nsDS5ReplicaRoot: dc=example,dc=com
nsDS5ReplicaHost: ad1.windows-server.com
nsDS5ReplicaPort: 389
nsDS5ReplicaBindDN: cn=sync user,cn=Users,dc=ad1
nsDS5ReplicaCredentials: {DES}ffGad646dT0nnsT8nJOaMA==
nsDS5ReplicaTransportInfo: TLS
nsds7NewWinUserSyncEnabled: on
nsds7NewWinGroupSyncEnabled: on
```

複数のサブツリーのペア間で同期するには、「Windows 同期での複数のサブツリーおよびフィルターの設定」 を参照してください。

16.12.2.2. 同期スケジュールの設定

同期は 2 つの方法で機能します。Directory Server は、*nsds5replicaupdateschedule* 属性を使用して、レプリケーションと同様に設定可能なスケジュールの Active Directory に更新を適用します。Directory Server は Active Directory をポーリングして変更の有無 (Active Directory サーバーが *winSyncInterval* 属性に設定されている頻度) を確認します。

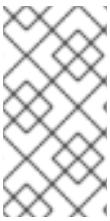
デフォルトでは、Directory Server の更新スケジュールは常に同期されます。Active Directory の間隔は、5 分ごとに Active Directory をポーリングする間隔です。

Directory Server が更新を Active Directory に送信するために使用するスケジュールを変更するには、*nsds5replicaupdateschedule* 属性を編集します。スケジュールは、24 時間クロックを使用して HHMM 形式の開始 (SSSS) および終了 (EEEE) 時間で設定されます。同期の更新スケジュールを設定する日は、0 (日曜日) から 6 (土曜日) までです。

```
nsds5replicaupdateschedule: SSSS EEEE DDDDDDD
```

たとえば、日曜日、火曜日、木曜日、土曜日の正午から午後 2 時まで同期を実行するようにスケジュールを設定します。

```
nsds5replicaupdateschedule: 1200 1400 0246
```



注記

同期時間は真夜中を含むことはできないため、設定 2300 0100 は有効ではありません。

Directory Server が Active Directory エントリーへの変更をチェックする頻度を変更するに

は、*winSyncInterval* 属性をリセットします。この属性は秒単位で設定されるため、デフォルトの 300 は、Directory Server が Active Directory サーバーを 300 秒または 5 分間隔でポーリングすることを意味します。これを高い値に設定すると、ディレクトリーの検索に時間がかかり、パフォーマンスに影響する場合に便利です。

```
winSyncInterval: 1000
```

16.12.2.3. 同期接続の変更

同期合意について接続の 2 つの側面を変更することができます。

- バインドユーザー名およびパスワード (*nsDS5ReplicaBindDN* および *nsDS5ReplicaCredentials*)
- 接続メソッド (*nsDS5ReplicaTransportInfo*)

nsDS5ReplicaTransportInfo を LDAP から TLS に変更すること しかできません。SSL への変更、または SSL からの変更はポート番号を変更することができないため、LDAP と LDAPS の切り替えにはポート番号の変更が必要となります。

以下に例を示します。

```
nsDS5ReplicaBindDN: cn=sync user,cn=Users,dc=ad1
nsDS5ReplicaCredentials: {DES}ffGad646dT0nnsT8nJOaMA==
nsDS5ReplicaTransportInfo: TLS
```

警告

Active Directory 同期ピアのポート番号を変更することはできません。そのため、標準ポートとセキュアでないポートの間で変更する必要があるため、標準/Start TLS 接続と TLS 接続を切り替えることはできません。

TLS に変更するには、同期合意を削除し、更新されたポート番号と新しいトランスポート情報を再度追加します。

16.12.2.4. 同期しているサブツリーから移動するエントリーの処理

同期合意は、Active Directory と Directory Server の両方において、またその 2 つの間で同期するサブツリーを定義します。スコープ 内のエントリー (サブツリー) が同期され、他のエントリーは無視されます。

ただし、同期プロセスは実際にルート DN で開始し、同期のエントリーの評価を開始します。エントリーは、Active Directory の *samAccount* と Directory Server の *uid* 属性に基づいて相関します。同期プラグインは、(*samAccount*/*uid* 関係に基づいて) エントリーが削除または移動されたために、同期されたサブツリーから削除された場合、その旨を通知します。これは、同期プラグインに対して、そのエントリーがもう同期されないことを示す信号です。

この問題は、同期プロセスで、移動したエントリーの処理方法を決定するための設定が必要となることです。対応するエントリーの削除、エントリー (デフォルト) の無視、またはエントリーの同期解除を 3 つのオプションがあります。

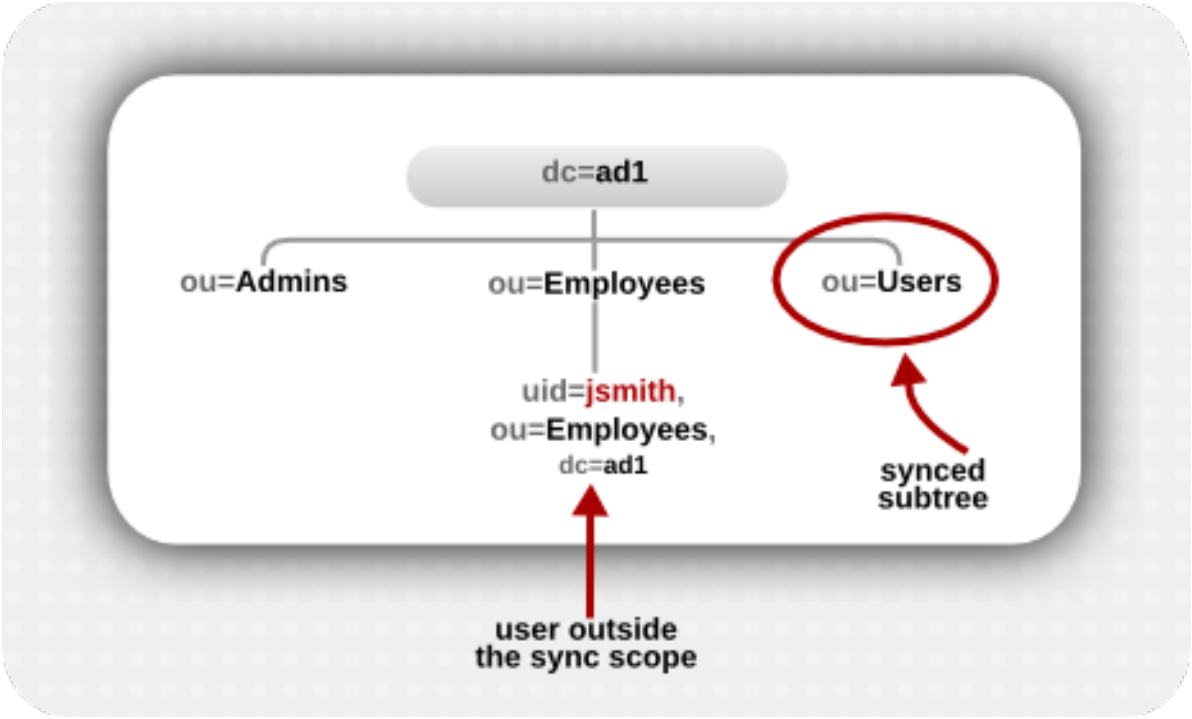
注記

これらの同期アクションは、Active Directory 側でエントリーが範囲外に移動した場合の Directory Server側での 処理方法にのみ関連しています。エントリーが Directory Server 側で同期されたサブツリーからエントリーを移動しても、Active Directory エントリーには影響はありません。

Directory Server 9.0 のデフォルトの動作では、対応する Directory Server エントリーが削除されるようになりました。これは、Active Directory 側のエントリーが Directory Server 側に同期されていなかった場合でも該当します。Directory Server 9.1 以降、デフォルトの動作ではエントリーを無視して、何も実行しません。

たとえば、*samAccount* ID が jsmith のユーザーは、Active Directory の ou=Employees サブツリーに作成されています。同期されたサブツリーは ou=Users であるため、jsmith ユーザーは Directory Server に同期されませんでした。

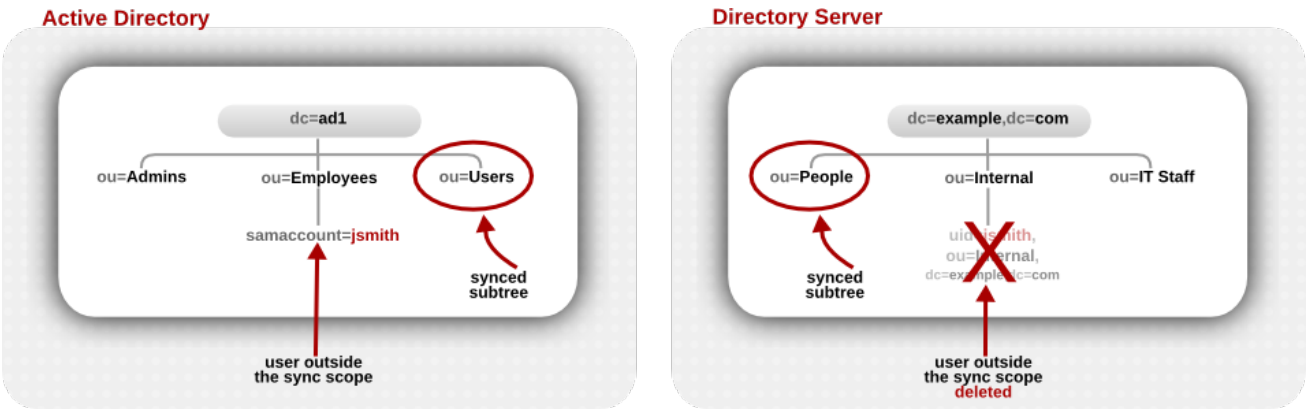
図16.9 Active Directory ツリー



バージョン 7.x および 8.x の Directory Server では、そのユーザーは同期されたサブツリーの外にあったため、同期は単にそのユーザーを無視していました。

Directory Server 9.0 以降、Directory Server はサブツリーの名前変更をサポートします。つまり、既存のエントリーはディレクトリーツリーのブランチ間で移動できるようになりました。同期プラグインは、Active Directory ツリーの中で、Directory Server のユーザー (*samAccount/uid* 関係) に対応し、同期サブツリーの外にあるエントリーを、意図的に 同期サブツリーの外に移動させることを前提としています (基本的には名前の変更操作)。「対応する」Directory Server エントリーを削除する必要があることを前提とします。

図16.10 Active Directory と Directory Server ツリーの比較

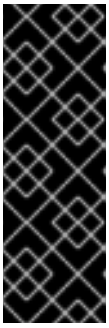


この仮定は必ずしも正確なものではなく、特に同期サブツリーの外側に常に存在するユーザーエントリーの場合は注意が必要です。

同期合意の *winSyncMoveAction* 属性は、これらの移動したエントリーの処理方法を設定します。

- none は何もしないため、同期した Directory Server エントリーが存在する場合は、同期するか、スコープ 内 に Active Directory エントリーを作成したりできます。同期された Directory Server エントリーが存在しない場合は、何も起こりません (Directory Serverバージョン 9.1 以降では、これがデフォルトの動作です)。

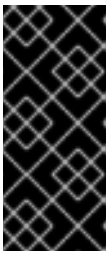
- unsync は、Directory Server エントリーから同期関連の属性 (*ntUser* または *ntGroup*) を削除しますが、Directory Server エントリーはそのまま残されます。



重要

エントリーの同期を解除すると、Active Directory のエントリーが後から削除され、Directory Server のエントリーがそのまま残ってしまう危険性があります。これにより、特に Active Directory 側でエントリーを再作成するのに Directory Server エントリーを使用する場合などに、データが不整合になる可能性があります。

- delete は、Active Directory と同期していたかどうかに関わらず、Directory Server で該当するエントリーを削除します (これは 9.0 のデフォルト動作です)。



重要

対応する Active Directory エントリーを削除せずに Directory Server エントリーを削除することはありません。このオプションは、Directory Server 9.0 システムとの互換性でのみ利用できます。

デフォルト動作を none から変更する必要がある場合は、同期合意を編集して *winSyncMoveAction* 属性を追加します。ldapmodify の使用：

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=replication_agreement_name,cn=replica,cn="dc=example,dc=com",cn=mapping
tree,cn=config
changetype: modify
add: winSyncMoveAction
winSyncMoveAction: unsync
```

16.13. パスワード同期サービスの管理

重要

Windows パスワードを同期するには、Active Directory ドメインのすべてのドメインコントローラーにパスワード同期をインストールする必要があります。

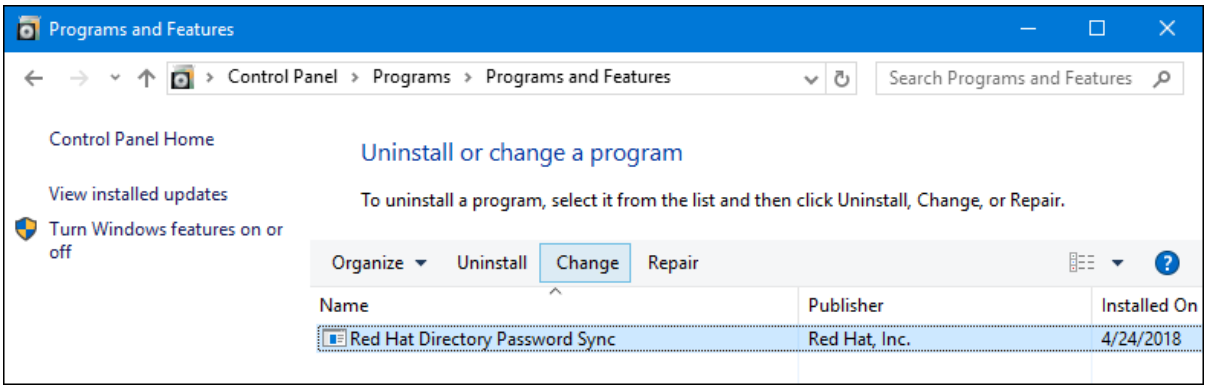
このサービスは、Active Directory で行ったパスワード変更を、Directory Server の対応するエントリーのパスワードと同期します。Windows サービスと同様に、Directory Server と Active Directory の同期方法に応じて、修正、起動、停止、およびアンインストールが可能です。

16.13.1. パスワード同期の変更

パスワード同期を再設定するには、以下を実行します。

1. コントロールパネル を開き、プログラムと機能 を選択します。

2. Red Hat Directory Password Sync エントリーを選択し、Change ボタンをクリックしてインストーラーを再起動して設定を変更します。



3. 設定画面に戻り、設定を変更します。

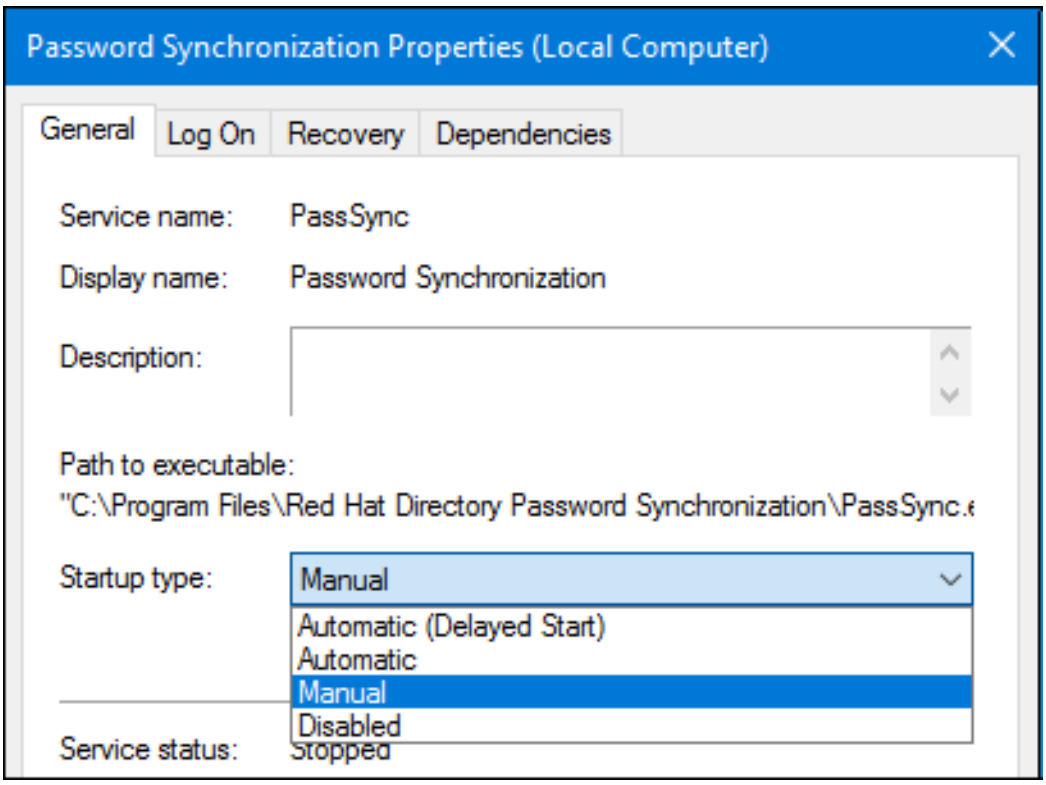
16.13.2. パスワード同期サービスの起動と停止

Password Sync サービスは、Active Directory ホストを起動するたびに起動するように設定されています。サービスを再設定し、Windows の再起動時に起動しないようにするには、以下を実行します。

1. Services アプリケーションを開きます。

2. **Password Synchronization サービスをダブルクリックします。**

3. **手動 ラジオボタンを選択し、OK をクリックします。**

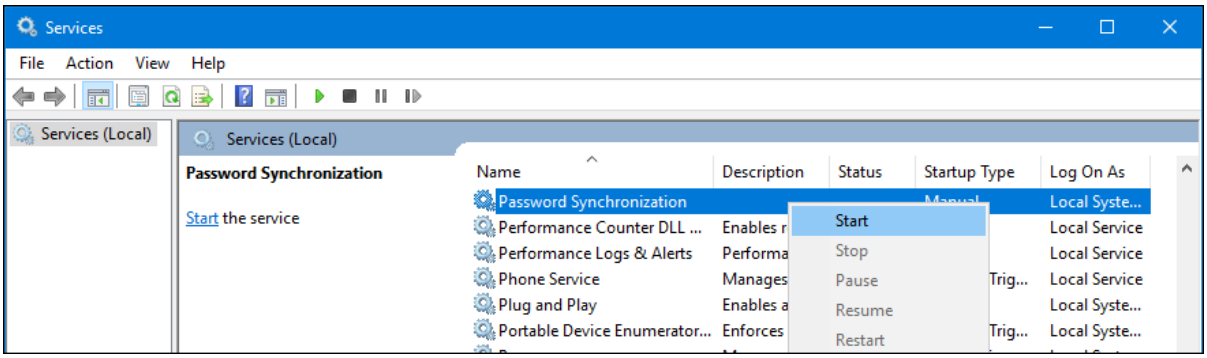


パスワード同期を開始および停止するには、以下を実行します。

1. **Services アプリケーションを開きます。**

2. **Password Synchronization サービスを右クリックします。**

3. **Stop、Start、または Restart を選択し、OKをクリックします。**

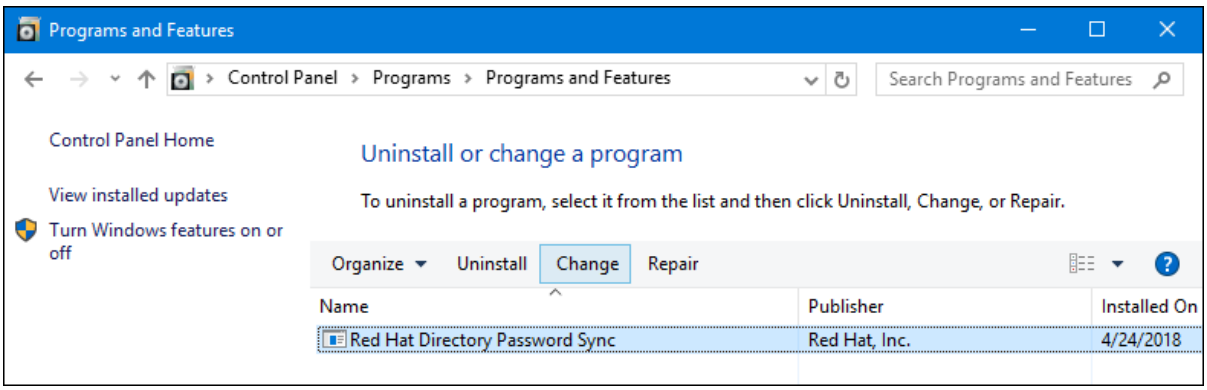


変更したパスワードは、Password Synchronization サービスが実行されていない場合でもキャプチャーされます。パスワード同期 を再起動すると、パスワードの変更は次の同期で Directory Server に送信されます。

16.13.3. パスワード同期サービスの アンインストール

1. コントロールパネル を開き、プログラムと機能 を選択します。

2. Red Hat Directory Password Sync エントリーを選択し、Uninstall ボタンをクリックします。



3. Password Sync に対して TLS が設定されている場合、パスワード同期 のアンインストール時に作成された cert8.db データベースおよび key3.db データベースは削除されません でした。これらのファイルは手動で削除します。

16.13.4. パスワード同期のアップグレード

詳細は、『Red 『Hat Directory Server インストールガイド』の該当するセクションを参照してください』。

16.14. トラブルシューティング

同期が正常に行われない場合は、Windows のイベントログや Directory Server のエラーログで、問題がないか確認してください。

レプリケーションロギングを有効にして同期エラーを記録する

レプリケーションロギングを有効にすると、同期に関するより詳細な情報がエラーログに記録されます。レプリケーションログレベルは、同期コードからより詳細なログを生成します。同期トラフィックに関連するメッセージ (レプリケーショントラフィックと同じ) は、問題を診断するのに役立ちます。

1. コンソールで、Configuration タブをクリックします。

2. 右側のナビゲーションメニューから Logs を選択し、エラーログを開きます。

3. エラーログレベルまでスクロールダウンし、メニューから Replication を選択します。

4. 保存します。

Error #1: 同期合意の作成時にメッセージのボックスは、Active Directory に接続できないことを示しています。

ディレクトリーのサフィックス、Windows ドメインおよびドメインホスト、および管理者 DN およびパスワードが正しいことを確認します。LDAPS に使用されるポート番号が正しいことを確認します。すべての接続情報が正しい場合は、Active Directory マシンが実行中であることを確認してください。

エラー #2: 同期後、ステータスは error 81 を返します。

同期ピアサーバーの 1 つは TLS 通信に対して適切に設定されていません。Directory Server のアクセスログファイルを調べ、Directory Server が接続試行を受け取っているかどうかを確認します。Directory Server のエラーログファイルには有用なメッセージがあります。

設定ミスの原因を突き止めるために、Directory Server への LDAPS 接続を試みます。この接続に失敗した場合は、すべての値 (ポート番号、ホスト名、IPv4/IPv6 アドレス、検索ベース、およびユーザー認証情報など) を確認して、それらのいずれかが問題かどうかを確認します。すべてが失敗した場合は、新しい証明書で Directory Server を再設定します。

Directory Server への LDAPS 接続に成功すると、設定が間違っていることが Active Directory にある可能性があります。Windows イベントログファイルでエラーメッセージを確認します。

注記

典型的な問題は、Windows 同期サービス証明書データベースが設定されたときに認証局が信頼できるものとして設定されていないことです。

エラー #3: エントリーは Active Directory のサブツリーから別のサブツリーに移動していますが、ユーザーは Directory Server の対応するサブツリーに移動していません。

これは、Active Directory の modrdn 操作と Directory Server のエントリーとの同期に関する既知の問題です。これを回避するには、Active Directory のエントリーを削除して、新しいサブツリーに追加します。削除と追加は、Directory Server ピアに対して適切に同期されます。

---

[2]
コンシューマーとは異なり、同期されていないサーバーで変更は引き続き可能です。ACL を使用して、同期されていないサーバーでエントリーを編集または削除し、データの整合性を維持します。

第17章 コンテンツの同期の設定

Content Synchronization プラグインを使用すると、Directory Server は RFC 4533 に従って SyncRepl プロトコルをサポートします。このプロトコルにより、LDAP サーバーとクライアントは Red Hat Directory Server をソースとして使用し、ローカルデータベースを Directory Server の変更するコンテンツと同期させることができます。

SyncRepl プロトコルを使用するには、以下を実行します。

- Directory Server で Content Synchronization プラグインを有効にし、必要に応じてクライアントが Directory Server にバインドするために使用する新規ユーザーを作成します。アカウントには、ディレクトリー内のコンテンツを読み取るパーミッションが必要です。
- クライアントを設定します。たとえば、同期するサブツリーの検索ベースを設定します。詳細は、クライアントのドキュメントを参照してください。

クライアントが Directory Server に接続できるようにするには、Content Synchronization プラグインを設定します。

1. Content Synchronization プラグインでは、*nsuniqueid* 属性をログに記録するのに Retro Changelog プラグインが必要です。

   a. Retro Changelog が有効になっているかどうかを確認するには、次のコマンドを実行します。

   ```
   # ldapsearch -D "cn=Directory Manager" -W -x -b \
       'cn=Retro Changelog Plugin,cn=plugins,cn=config' nsslapd-pluginEnabled
   ...
   dn: cn=Retro Changelog Plugin,cn=plugins,cn=config
   nsslapd-pluginEnabled: off
   ```

   nsslapd-pluginEnabled 属性が off に設定されている場合、Retro Changelog は無効になります。有効にする場合は、「Retro Changelog プラグインの有効化」 を参照してください。

   b. *nsuniqueid* 属性を、Retro Changelog プラグインの設定に追加します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=Retro Changelog Plugin,cn=plugins,cn=config
changetype: add
add: nsslapd-attribute
nsslapd-attribute: nsuniqueid:targetUniqueId
```

c. 必要に応じて、パフォーマンスを向上させるために、以下の推奨事項を適用します。

i. Retro Changelog のエントリーの最大有効期間を設定します。たとえば、2 日 (2d) を設定するには、以下のコマンドを実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=changelog5,cn=config
changetype: modify
replace: nsslapd-changelogmaxage
nsslapd-changelogmaxage: 2d
```

ii. データを同期するバックエンドまたはサブツリーのクライアントアクセスを把握している場合は、Retro Changelog プラグインのスコープを制限します。たとえば、*cn=demo,dc=example,dc=com* サブツリーを除外するには、以下を入力します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=Retro Changelog Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-exclude-suffix
nsslapd-exclude-suffix: cn=demo,dc=example,dc=com
```

2. Content Synchronization プラグインを有効にします。

- コマンドラインの使用するには、以下を行います。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=Retro Changelog Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

- Directory Server コンソールの使用： 「Directory Server コンソールでプラグインの有効化」 を参照してください。

3.
デフォルトの Directory Server は、*oid=1.3.6.1.4.1.4203.1.9.1.1,cn=features,cn=config* エントリーにアクセス制御命令 (ACI) を作成し、すべてのユーザーが SyncRepl プロトコルを使用できるようにします。

```
aci: (targetattr != "aci")(version 3.0; acl "Sync Request Control";
  allow( read, search ) userdn = "ldap:///all";)
```

必要に応じて、SyncRepl コントロールを使用して ACI を制限するように更新します。ACI の詳細は、「バインドルールの定義」を参照してください。

4.
Directory Server を再起動します。

```
# systemctl restart dirsrv@instance_name
```

クライアントは、SyncRepl プロトコルを使用して、Directory Server とデータを同期できるようになりました。

第18章 アクセス制御の管理

本章では、Red Hat Directory Server の Access Control Instructions (ACI) を使用してエントリーへのアクセスを管理する方法を説明します。

18.1. アクセス制御要件

Directory Server が要求を受信すると、bind 操作でユーザーによって提供される認証情報、およびディレクトリーに定義されている ACI を使用し、要求されたエントリーまたは属性へのアクセスを許可または拒否します。サーバーは、read、write、search、および compare などのアクションのパーミッションを許可または拒否できます。ユーザーに付与されたパーミッションレベルは、指定される認証情報によって異なります。

Directory Server のアクセス制御により、ACI が適用される場合に正確なルールを設定できます。

- ディレクトリー全体、サブツリー、または特定のエントリーの場合
- 特定のユーザー、特定のユーザーまたはグループに属するすべてのユーザー、またはディレクトリー内のすべてのユーザーの場合
- IP アドレス、IP 範囲、または DNS 名などの特定の場所。

ロードバランサーは場所固有のルールに影響を及ぼす可能性があることに注意してください。

重要

複雑な ACI の読み取りと理解は難しくなります。ほとんどの場合で複雑な ACI を 1 つではなく、同じ効果を達成するために複数の単純なルールを作成することが推奨されます。

18.2. ACI 配置

Directory Server は、ディレクトリーエントリーの複数値の *aci* 操作属性に ACI を保存します。ACI を設定するには、*aci* を対応するディレクトリーエントリーに追加します。Directory Server は ACI を適用します。

- ACI を含むエントリー (子エントリーがない場合) にのみ適用されます。

  たとえば、クライアントが uid=*user_name*,ou=People,dc=example,dc=com オブジェクトへのアクセスを必要とし、ACI が dc=example,dc=com にのみ設定されており、子エントリーには設定されていない場合は、この ACI のみが適用されます。

- ACI を含むエントリーと、(子エントリーがある場合は) その下のすべてのエントリーへ。これにより、サーバーが指定のエントリーに対するアクセスパーミッションを評価すると、リクエストされたディレクトリー接尾辞と、エントリー自体の ACI との間のすべてのエントリーについて ACI を検証します。

  たとえば、ACI は dc=example,dc=com および ou=People,dc=example,dc=com エントリーに設定されます。ACI が設定されていない uid=*user_name*,ou=People,dc=example,dc=com オブジェクトにクライアントがアクセスする場合、Directory Server はまず dc=example,dc=com エントリー上の ACI を検証します。この ACI がアクセスを許可する場合、Directory Server は ou=People,dc=example,dc=com 上の ACI を検証します。この ACI がクライアントを正常に承認すると、オブジェクトにアクセスできます。

注記

rootDSE エントリーに設定された ACI はこのエントリーにのみ適用されます。

エントリーで作成された ACI は、そのエントリーに直接適用するのではなく、以下のサブツリーの一部のエントリーまたはすべてのエントリーに適用できます。この方法の利点は、一般的な ACI をディレクトリーツリーの上位において、下位にあるエントリーに影響を与えることができることです。たとえば、inetOrgPerson オブジェクトクラスを含むエントリーをターゲットにする ACI は、*organizationalUnit* エントリーまたは *locality* エントリーのレベルで作成できます。

注記

一般的なルールを高レベルのブランチポイントに配置し、ディレクトリーツリー内の ACI の数を最小限にします。より具体的なルールの範囲を制限するには、できるだけ早くリーフエントリーに配置します。

18.3. ACI 構造

aci 属性は以下の構文を使用します。

(*target_rule*) (version 3.0; acl "*ACL_name*"; *permission_rule bind_rules*;)

- ***target_rule*** は、アクセスを制御するためのエントリー、属性、またはエントリーと属性のセットを指定します。詳細は、「ターゲットの定義」を参照してください。

- ***version 3.0*** は、ACI バージョンを識別する必須の文字列です。

- ***permission_rule*** は、read または write など、どの権限が許可または拒否されるかを設定します。詳細は、「パーミッションの定義」を参照してください。

- ***bind_rules*** は、アクセスを許可または拒否するために、バインド時に一致するルールを指定します。詳細は、「バインドルールの定義」を参照してください。

注記

パーミッションとバインドルールのペアはアクセス制御ルールと呼ばれます。

特定のターゲットに複数のアクセス制御を効率的に設定するには、ターゲットごとに複数のアクセス制御ルールを設定します。

(*target_rule*)(version 3.0; acl "*ACL_name*"; *permission_rule bind_rules*; *permission_rule bind_rules*; ... ;)

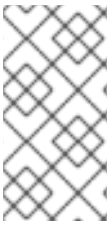
18.4. ACI 評価

特定のエントリーに対するアクセス権を評価するには、サーバーによりエントリー自体に存在する ACI の一覧と、親エントリーにある ACI の一覧が Directory Server に保存されている最上位のエントリーに再び作成されます。ACI は、特定のインスタンス用のデータベース全体で評価されますが、異なるインスタンスもすべて評価されます。

Directory Server は、ディレクトリーツリー内の配置ではなく、ACI のセマンティクスに基づいてこの一覧を評価します。これは、ディレクトリーツリーのルートに近い ACI が、ディレクトリーツリーのリーフに近い ACI よりも優先されないことを意味しています。

Directory Server では、ACI の deny パーミッションは allow パーミッションよりも優先されます。たとえば、ディレクトリーのルートレベルで書き込みパーミッションを拒否する場合は、他の ACI がこのパーミッションを付与していても、ユーザーはディレクトリーに書き込むことができません。特定の

ユーザーにディレクトリーへの書き込みパーミッションを付与するには、ユーザーがそのディレクトリーに書き込むことができるように、元の拒否ルールに例外を追加する必要があります。



注記

ACI を改善するには、deny ルールの代わりに、粒度の細かい allow ルールを使用します。

18.5. ACI の制限

ACI を設定すると、以下の制限が適用されます。

- ディレクトリーデータベースが複数のサーバーに分散されている場合は、ACI で使用できるキーワードに以下の制限が適用されます。
  - groupdn キーワードを使用したグループエントリーに依存する ACI は、グループエントリーと同じサーバーに置く必要があります。

    グループが動的の場合、グループのすべてのメンバーに、サーバーのエントリーが必要です。静的グループのメンバーエントリーは、リモートサーバーに配置できます。
  - roledn キーワードを使用したロール定義に依存する ACI は、ロール定義エントリーと同じサーバーにある必要があります。ロールを持つすべてのエントリーは、同じサーバーに配置する必要があります。

  ただし、ターゲットエントリーに保存されている値を、たとえば userattr キーワードを使用して、bind ユーザーのエントリーに保存されている値と一致させることができます。この場合、通常、バインドユーザーに ACI を格納するサーバーにエントリーがない場合でも、アクセスが評価されます。

  詳細は、「データベースリンクおよびアクセス制御評価」を参照してください。
- 以下の ACI キーワードでは、Class of Service (CoS) 属性などの仮想属性を使用することはできません。

- targetfilter
- targattrfilters
- userattr

詳細は、「8章エントリーの編成とグループ化」を参照してください。

- アクセス制御ルールは、ローカルサーバーでのみ評価されます。たとえば、ACI キーワードの LDAP URL にサーバーのホスト名を指定すると、URL は無視されます。

18.6. DIRECTORY SERVER がレプリケーショントポロジーで ACI を処理する方法

ACI はエントリーの *aci* 属性に保存します。したがって、ACI を含むエントリーが複製されたデータベースの一部である場合、ACI は複製されます。

ACI は、受信 LDAP 要求を解決するサーバーで常に評価されます。コンシューマーサーバーが更新要求を受け取ると、サプライヤーサーバーに参照を返してから、その要求がサプライヤーでサービスを提供できるかどうかを評価します。

18.7. ACI の表示

本セクションでは、ACI を表示する方法を説明します。

18.7.1. コマンドラインを使用した ACI の表示

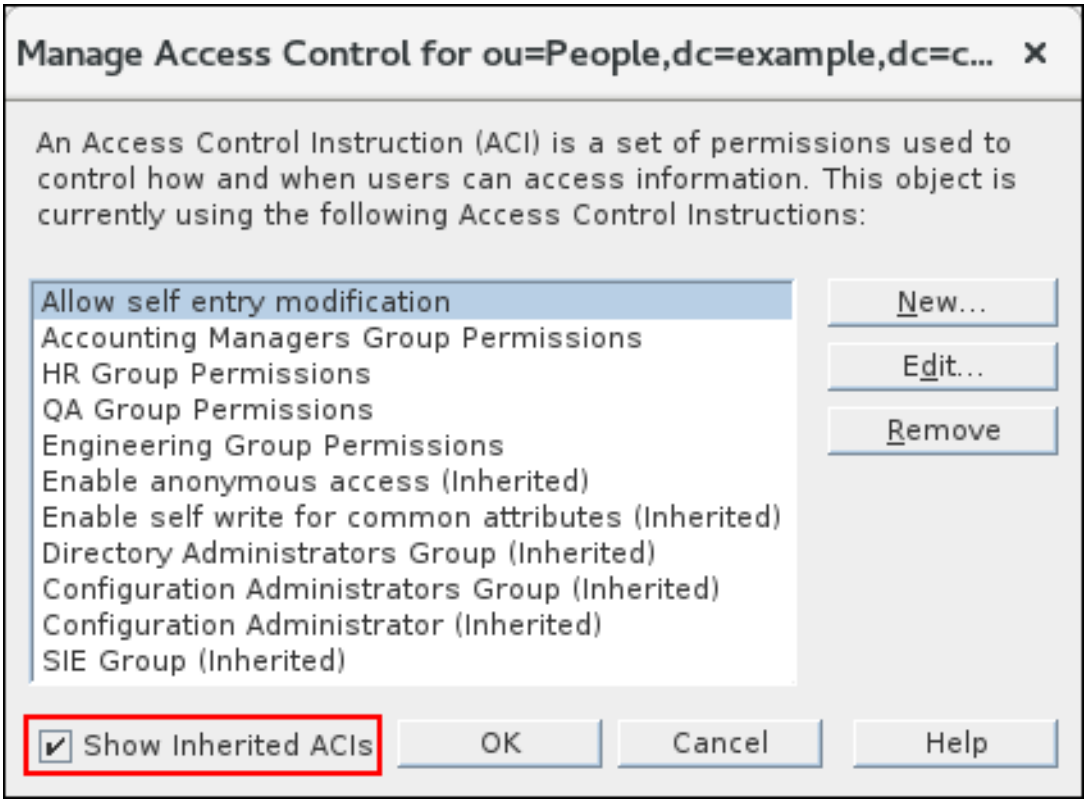
ldapsearch ユーティリティーを使用して、コマンドラインを使用して ACI を表示します。たとえば、dc=example,dc=com およびサブエントリーに設定された ACI を表示するには、以下のコマンドを実行します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x \
    -b "dc=example,dc=com" -s sub '(aci=*)' aci
```

18.7.2. コンソールを使用した ACI の表示

コンソールを使用して ACI を表示するには、以下を実行します。

1. Directory Server コンソールを開きます。
2. Directory タブで、エントリーを右クリックし、Set Access Permissionsを選択します。
3. オプションで、Show Inherited ACI を選択して、ディレクトリーの上位レベルのエントリーをさらに表示します。



18.8. ACI の追加

本セクションでは、ACI を追加する方法を説明します。

18.8.1. コマンドラインを使用した ACI の追加

ldapmodify ユーティリティーを使用して ACI を追加します。以下に例を示します。

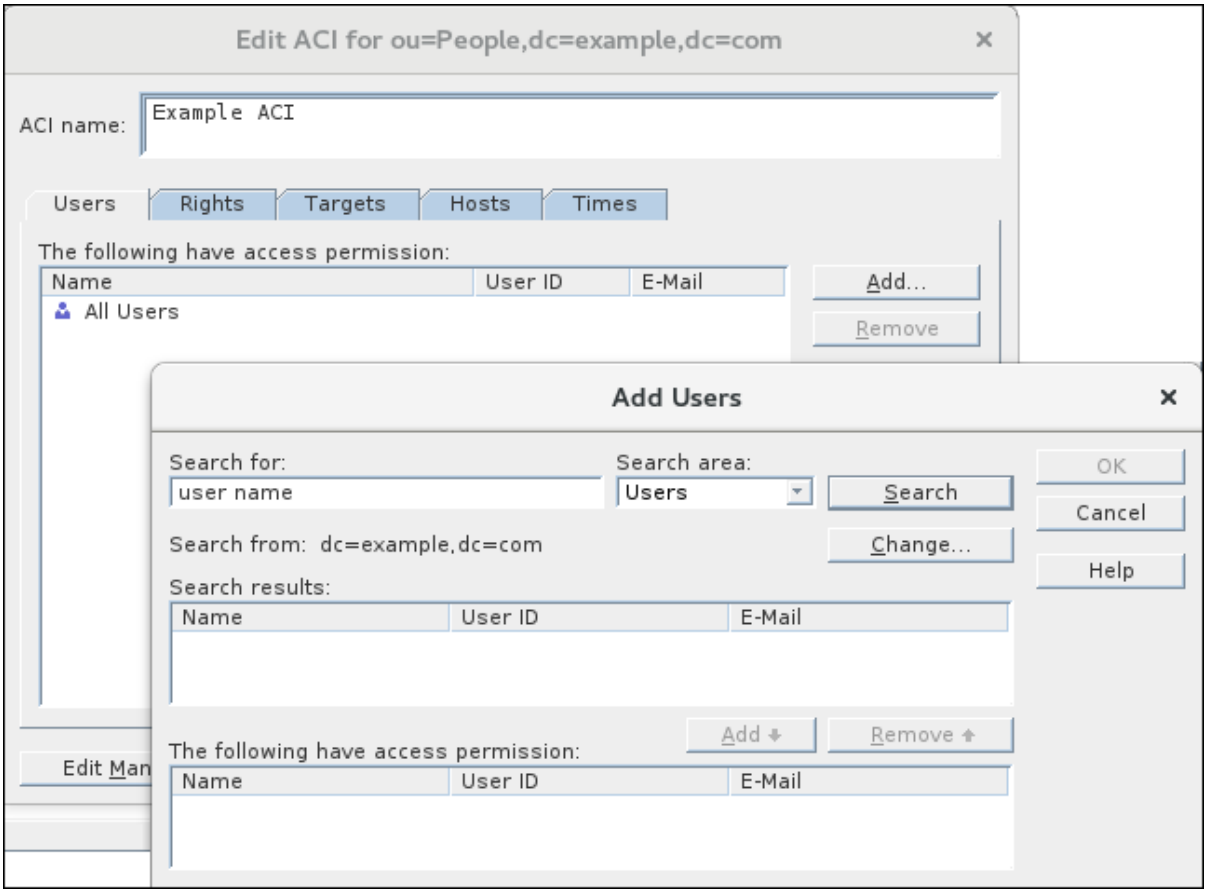
```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=People,dc=example,dc=com
changetype: modify
```

```
add: aci
aci: (targetattr="userPassword") (version 3.0; acl "Allow users updating their password";
 allow (write) userdn= "ldap:///self";)
```

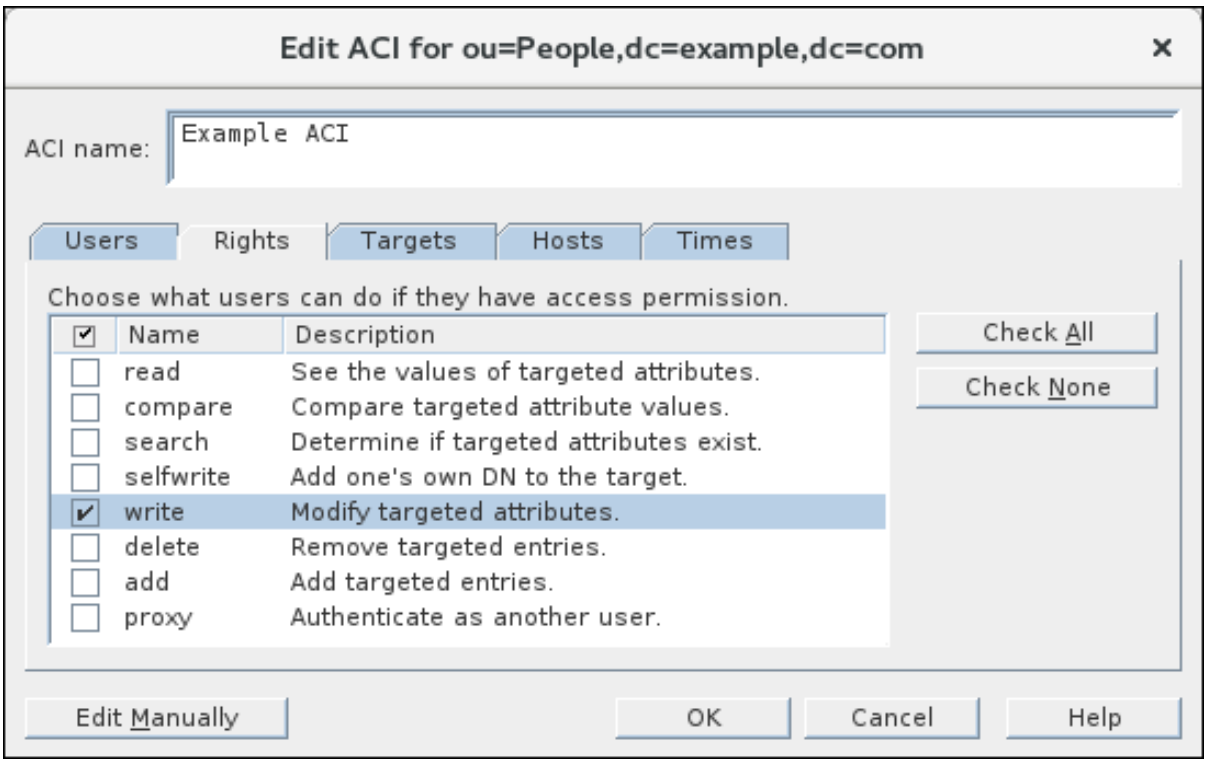
18.8.2. コンソールを使用した ACI の追加

コンソールを使用して ACI を追加するには、以下を行います。

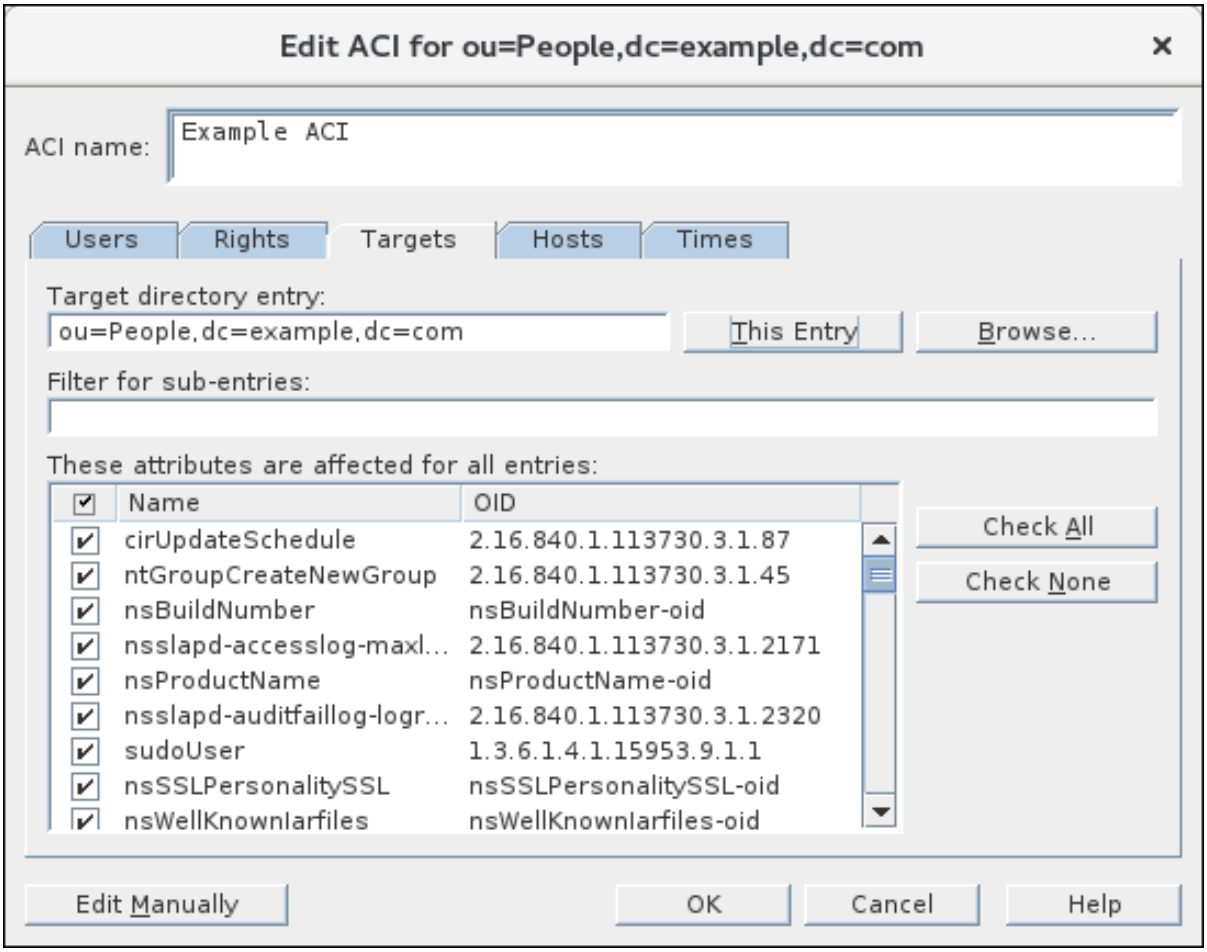
1. Directory Server コンソールを開きます。
2. Directory タブで、エントリーを右クリックし、Set Access Permissionsを選択します。
3. ACI Name フィールドに ACI の名前を入力します。
4. Users タブで、Add ボタンをクリックしてユーザー、グループ、ロール、管理者、または特別な権限を一覧に追加します。
   a. Search for フィールドに文字列を入力し、検索エリアを選択して Search をクリックします。
   b. 検索結果からエントリーを選択し、Add をクリックします。
   c. OK をクリックします。

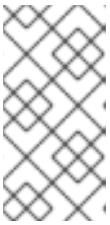


5. **Rights タブで、この ACI に設定するパーミッションを選択します。**



6. **Targets タブで、ターゲットディレクトリーエントリーを選択します。**





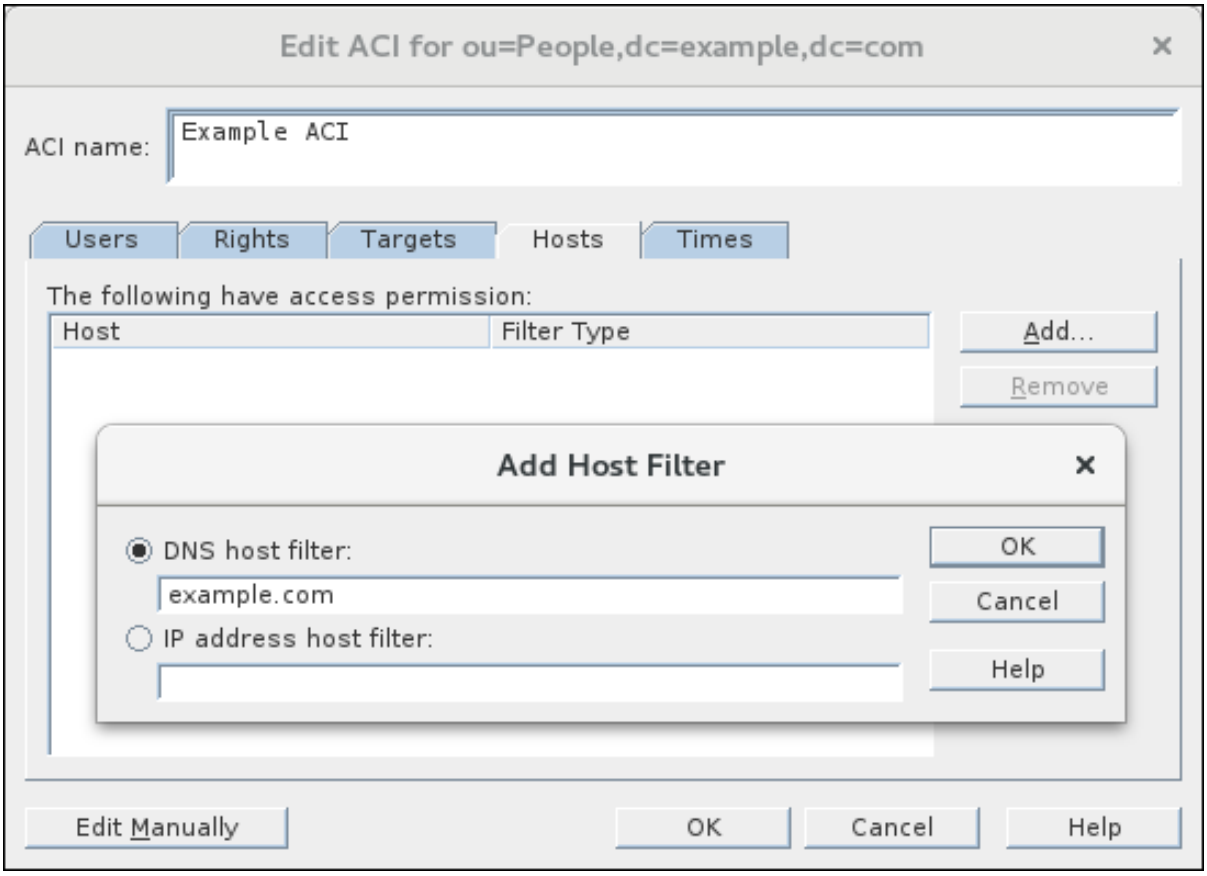
注記

ターゲット DN の値を変更できますが、新しい DN は選択したエントリーの直接または間接子である必要があります。

ACI がこのノード下のサブツリーのすべてのエントリーをターゲットにするには、Filter for Sub-entries フィールドにフィルターを入力します。フィルターは、ターゲットエントリー下のすべてのエントリーに適用されます。たとえば、フィルターを ou=Sales に設定すると、DN に ou=Sales を持つエントリーのみが返されます。

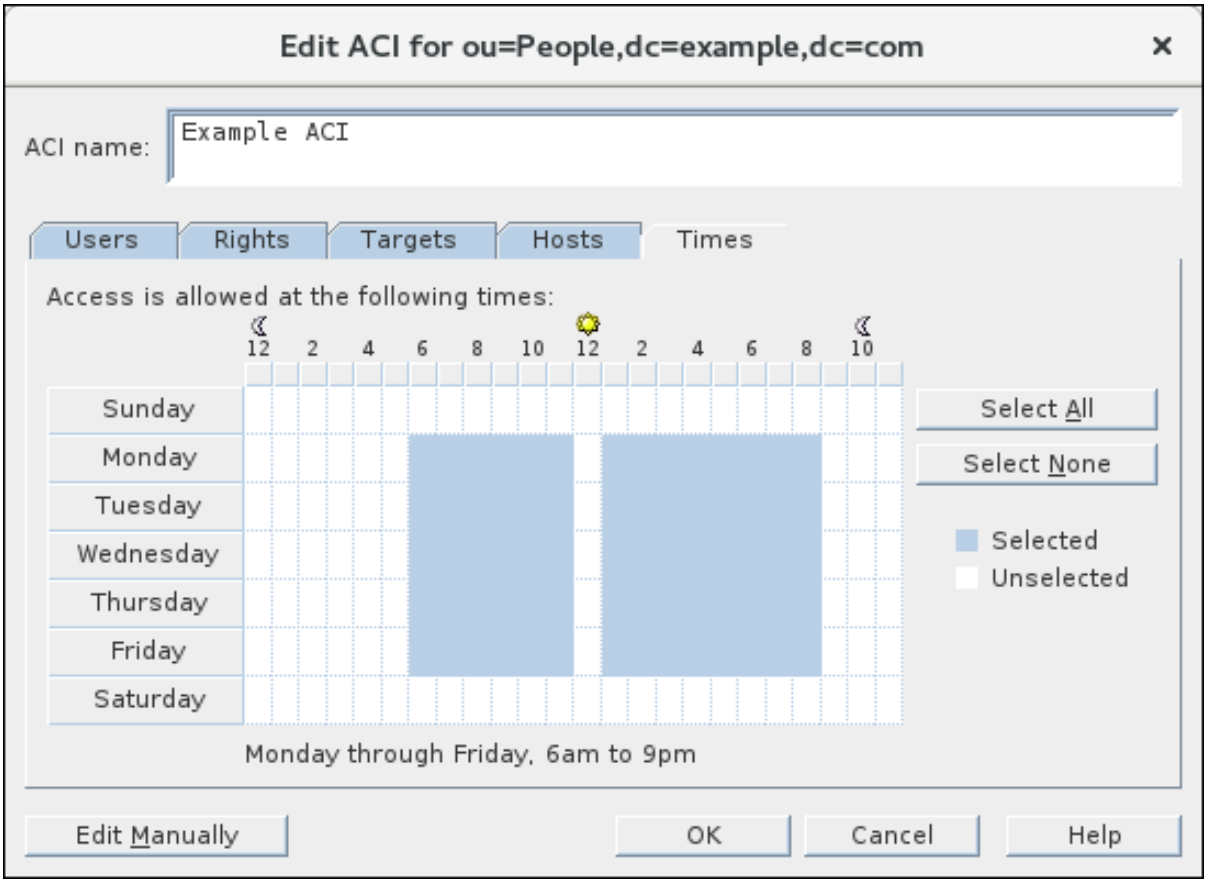
さらに、リスト内の属性を選択して、ACI の範囲を特定の属性に制限することもできます。

7. **ホスト タブで、オプションで DNS 名または IP アドレスを追加します。**



DNS 名または IP アドレスを設定する場合、ACI はこれらのホストからの LDAP 操作にのみ適用されます。

8. **Times タブで、ACI を適用する時間をオプションから選択します。**



デフォルトでは、アクセスはいつでも許可されます。カーソルをテーブル上でクリックしてドラッグして、アクセス時間を変更します。継続的な時間範囲のみを選択できることに注意してください。

9. **OK をクリックします。**

注記

ACI を作成する時点で、Edit Manually ボタンをクリックして、ウィザード入力に対応する LDIF ステートメントを表示します。このウィンドウでこのステートメントを編集できますが、変更がグラフィカルインターフェースに表示されない場合があります。

18.9. ACI の削除

本セクションでは、エントリーから ACI を削除する方法を説明します。

18.9.1. コマンドラインを使用した ACI の削除

コマンドラインで ACI を削除するには、次を実行します。

1. エントリーに設定された ACI を表示します。「コマンドラインを使用した ACI の表示」を参照してください。

2. ACI を削除します。

   - 1 つの *aci* 属性がエントリーに設定されているか、またはエントリーからすべての ACI を削除する場合は、以下を実行します。

     ```
     # ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
     dn: ou=People,dc=example,dc=com
     changetype: delete
     delete: aci
     ```

   - 複数の ACI がエントリーに存在し、特定の ACI を削除する場合は、実際の ACI を指定します。

     ```
     # ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
     dn: ou=People,dc=example,dc=com
     changetype: modify
     delete: aci
     aci: (targetattr="userPassword") (version 3.0; acl "Allow users
      updating their password"; allow (write) userdn= "ldap:///self";)
     ```

属性の削除に関する詳細は「エントリーからの属性の削除」を参照してください。

18.9.2. コンソールを使用した ACI の削除

コンソールを使用して ACI を削除するには、以下を実行します。

1. Directory Server コンソールを開きます。

2. Directory タブで、エントリーを右クリックし、Set Access Permissionsを選択します。

3. 一覧から ACI を選択し、Remove をクリックします。

4. OK をクリックします。

18.10. ACI の更新

本セクションでは、既存の ACI を更新する方法を説明します。

18.10.1. コマンドラインを使用した ACI の更新

コマンドラインを使用して ACI を更新するには、以下を実行します。

1. 既存の ACI を削除します。「コマンドラインを使用した ACI の削除」 を参照してください。

2. 更新された設定で新しい ACI を追加します。「コマンドラインを使用した ACI の追加」を参照してください。

18.10.2. コンソールを使用した ACI の更新

コンソールを使用して ACI を更新するには、以下を実行します。

1. Directory Server コンソールを開きます。

2. Directory タブで、エントリーを右クリックし、Set Access Permissionsを選択します。

3. 一覧から ACI を選択し、Edit をクリックします。

4. ACI を更新します。個別の画面については、「コンソールを使用した ACI の追加」 セクションで説明されています。

5. OK をクリックします。

18.11. ターゲットの定義

ACI のターゲットルールは、Directory Server が ACI を適用するエントリーを定義します。ターゲットを設定しない場合、ACI は *aci* 属性が含まれるエントリーと以下のエントリーに適用されます。

ACI では、以下の強調表示された部分がターゲットルールになります。

(*target_rule*)(version 3.0; acl "*ACL_name*"; *permission_rule bind_rules*;)

複雑な ACI の場合、Directory Server は異なるキーワードを持つ複数のターゲットルールをサポートします。

(*target_rule_1*)(*target_rule_2*)(...)(version 3.0; acl "*ACL_name*"; *permission_rule bind_rules*;)

複数のターゲットルールを指定した場合に、その順番は関係ありません。以下のキーワードはそれぞれ、ACI で一度だけ使用できることに注意してください。

- **target**
- **targetattr**
- **targetattrfilters**
- **targetfilter**
- **target_from**
- **target_to**

構文

ターゲットルールの一般的な構文は、以下のとおりです。

(*keyword comparison_operator* "*expression*")

-

keyword: ターゲットの種類を設定します。「よく使用されるターゲットキーワード」を参照してください。

- *comparisonoperator*: 有効な値は = および != で、ターゲットが式で指定されたオブジェクトであるかを示します。

警告

セキュリティー上の理由から、Red Hat は、他のすべてのエントリーまたは属性で指定の操作を許可するため、!= 演算子を使用しないことを推奨します。以下に例を示します。

```
(targetattr != "userPassword");(version 3.0; acl "example"); allow (write)
... );
```

前の例では、ACI を設定する識別名 (DN) の下にある *userPassword* 属性以外の属性の設定、更新、または削除を行うことができます。ただし、これにより、ユーザーはこの属性への書き込みアクセスを許可する *aci* 属性を追加することもできます。

- *expression*: ターゲットを設定し、引用符で囲む必要があります。式自体は使用するキーワードによって異なります。

18.11.1. よく使用されるターゲットキーワード

管理者は、以下のターゲットキーワードを頻繁に使用します。

- target: 「ディレクトリーエントリーのターゲット」を参照してください。
- targetattr: 「ターゲット属性」を参照してください。
- targetfilter: 「LDAP フィルターを使用したエントリーと属性の対象」を参照してください。

- **targattrfilters:** 「LDAP フィルターを使用した属性値のターゲット」を参照してください。

18.11.1.1. ディレクトリーエントリーのターゲット

DN およびその下のエントリーに基づいてアクセスを制御するには、ACI の target キーワードを使用します。target キーワードを使用するターゲットルールは、DN を式として取ります。

```
(target comparison_operator "ldap:///distinguished_name")
```

注記

対象となる DN またはその上位 DN に、target キーワードで ACI を設定する必要があります。たとえば、ou=People,dc=example,dc=com をターゲットにする場合、ACI を ou=People,dc=example,dc=com または dc=example,dc=com のいずれかに設定する必要があります。

例18.1 target キーワードの使用

ou=People,dc=example,dc=com エントリーに保存されているユーザーを有効にして、独自のエントリー内の全属性を検索および表示するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: dc=People,dc=example,dc=com
changetype: modify
add: aci
aci: (target = "ldap:///ou=People,dc=example,dc=com") (version 3.0;
 acl "Allow users to read and search attributes of own entry"; allow (search, read)
 (userdn = "ldap:///self");)
```

target キーワードでのワイルドカードの使用

* ワイルドカード文字ターゲットに複数のエントリーを使用できます。

以下のターゲットルールの例は、*uid* 属性が文字 a で始まる値に設定される ou=People,dc=example,dc=com のすべてのエントリーと一致します。

```
(target = "ldap:///uid=a*,ou=People,dc=example,dc=com")
```

ワイルドカードの位置に応じて、ルールは属性値だけでなく、完全な DN にも適用されます。その

ため、ワイルドカードを DN の一部の代わりに使用できます。

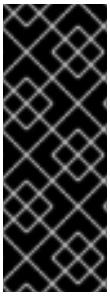
例18.2 ワイルドカードを使用したディレクトリーエントリーのターゲット

次のルールは、dc=example,dc=com ツリー内のすべてのエントリーを対象とし、*uid* 属性が一致するもので、dc=example,dc=com エントリー自体に格納されているエントリーだけではありません。

```
(target = "ldap:///uid=user_name*,dc=example,dc=com")
```

以前のターゲットルールは、以下のような複数のエントリーと一致します。

- uid=*user_name*,dc=example,dc=com
- uid=*user_name*,ou=People,dc=example,dc=com
- uid=*user_name2*,dc=example,dc=com



重要

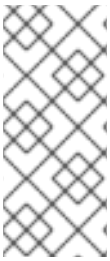
Directory Server は、DN の接尾辞部分でのワイルドカードをサポートしません。たとえば、ディレクトリーの接尾辞が dc=example,dc=com の場合は、(target = "ldap:///dc=*.com") などのように、この接尾辞でワイルドカード付きのターゲットは使用できません。

18.11.1.2. ターゲット属性

ACI のアクセスを特定の属性に制限するには、targetattr キーワードを使用します。たとえば、このキーワードは以下を定義します。

- 読み取り操作では、どの属性がクライアントに返されるか
- 検索操作では、どのような属性が検索されるのか

- 書き込み操作では、どの属性がオブジェクトに書き込むことができるか
- add 操作では、新規オブジェクトの作成時に追加できる属性



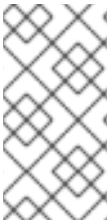
注記

特定の状況では、targetattr キーワードを使用して、他のターゲットキーワードを targetattr と組み合わせることで、ACI をセキュアにすることができます。サンプルについては、「ターゲットルールの高度な使用方法」を参照してください。

targetattr キーワードを使用するターゲットルールで複数の属性を分離するには、|| コマンドを使用します。

```
(targetattr comparison_operator "attribute_1 || attribute_2 || ...")
```

式に設定された属性はスキーマに定義する必要があります。



注記

式に指定される属性は、ACI の作成先となるエントリーと、その下のすべてのエントリーに適用されます。

例18.3 targetattr キーワードの使用

dc=example,dc=com に保存されているユーザーとすべてのサブエントリーで、独自のエントリー内の *userPassword* 属性を更新するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "userPassword") (version 3.0;
 acl "Allow users updating own userPassword";
 allow (write) (userdn = "ldap:///self");)
```

targetattr キーワードでのワイルドカードの使用

* ワイルドカード文字を使用すると、たとえば全属性をターゲットにすることができます。

```
(targetattr = "*")
```

警告

セキュリティー上の理由から、操作属性を含むすべての属性へのアクセスが許可されているため、targetattr ではワイルドカードを使用しないでください。たとえば、ユーザーがすべての属性を追加または変更できると、ユーザーは追加の ACI を作成し、独自の権限を増やす可能性があります。

18.11.1.3. LDAP フィルターを使用したエントリーと属性の対象

特定の基準に一致するエントリーのグループを対象にするには、LDAP フィルターで targetfilter キーワードを使用します。

```
(targetfilter comparison_operator "LDAP_filter")
```

フィルター式は、「14章ディレクトリーエントリーの検索」で説明される標準の LDAP 検索フィルターです。

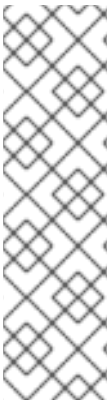
例18.4 targetfilter キーワードの使用

department 属性が Engineering または Sales に設定されているすべてのエントリーを変更するために、cn=Human Resources,dc=example,dc.com グループのメンバーにパーミッションを付与するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetfilter = "(|(department=Engineering)(department=Sales)")
 (version 3.0; acl "Allow HR updating engineering and sales entries";
 allow (write) (groupdn = "ldap:///cn=Human Resources,dc=example,dc.com");)
```

targetfilter キーワードはエントリー全体を対象にします。これを targetattr キーワードと組み合わせると、ACI はターゲットエントリーの属性のサブセットにのみ適用されます。「フィルターに一致す

るエントリーの個別属性のターゲット設定」を参照してください。



注記

LDAP フィルターは、ディレクトリーに分散されるエントリーおよび属性をターゲットにする場合に便利です。ただし、フィルターにはアクセスを管理するオブジェクトの名前を直接付けないため、結果が予測できないことがあります。フィルターが設定された ACI がターゲットとするエントリーのセットは、属性が追加または削除される際に変更する可能性が高くなります。したがって、ACI で LDAP フィルターを使用する場合は、ldapsearch 操作などで同じフィルターを使用して、正しいエントリーおよび属性を対象としていることを確認してください。

targetfilter キーワードでのワイルドカードの使用

targetfilter キーワードは、標準の LDAP フィルターと同様にワイルドカードをサポートします。たとえば、値が adm で始まるすべての *uid* 属性をターゲットにするには、次のコマンドを実行します。

```
(targetattr = "(uid=adm*) ...)
```

18.11.1.4. LDAP フィルターを使用した属性値のターゲット

アクセス制御を使用すると、属性の特定値を対象にできます。つまり、ある属性の値が ACI で定義されている基準を満たしていれば、その属性に対してパーミッションを付与したり、拒否したりすることができるのです。属性の値に基づいてアクセスを許可または拒否する ACI は、値ベースの ACI と呼ばれます。

値ベースの ACI を作成するには、以下の構文で targattrfilters キーワードを使用します。

- 1 つの属性とフィルターの組み合わせが含まれる操作の場合:

  ```
  (targattrfilters="operation=attribute:filter")
  ```

- 複数の属性とフィルターの組み合わせのある操作の場合:

  ```
  (targattrfilters="operation=attribute_1:filter_1 && attribute_2:filter_2 ... &&
  attribute_m:filter_m")
  ```

- 複数の属性とフィルターを組み合わせた 2 つの操作の場合。

```
(targattrfilters="operation_1=attribute_1_1:filter_1_1 && attribute_1_2:filter_1_2 ... &&
attribute_1_m:filter_1_m , operation_2=attribute_2_1:filter_2_1 && attribute_2_2:filter_2_2 ...
& attribute_2_n:filter_2_n ")
```

上記の構文の例では、オペレーションを **add** または **del** のいずれかに設定できます。**attribute:filter** の組み合わせは、フィルターと、フィルターが適用される属性を設定します。

注記

値ベースの **ACI** はコマンドラインを使用している場合に限りサポートされます。

以下では、フィルターを一致させる方法を説明します。

- エントリーを作成する際に、新しいエントリーの属性にフィルターが適用されると、その属性の各インスタンスがフィルターに一致する必要があります。
- エントリーとフィルターを削除するとエントリーの属性に適用される場合、その属性の各インスタンスはフィルターと一致する必要があります。
- エントリーを変更し、操作が属性を追加する場合は、その属性に適用される **add** フィルターが一致している必要があります。
- 操作が属性を削除すると、その属性に適用される **del** フィルターが一致している必要があります。エントリーに属性の個別の値が置き換えられる場合は、**add** および **del** フィルターの両方が一致する必要があります。

例18.5 targattrfilters キーワードの使用

Admin ロールを除く独自のエントリーにロールを追加できるようにする **ACI** を作成するには、値が **123** プレフィックスで始まる限り、***telephone*** 属性を追加するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targattrfilters="add=nsroledn:(!(nsroledn=cn=Admin)) &&
 telephoneNumber:(telephoneNumber=123*)") (version 3.0;
 acl "Allow adding roles and telephone";
 allow (add) (userdn = "ldap:///self");)
```

18.11.2. 詳細なターゲットキーキーワード

このセクションでは、頻繁に使用されないターゲットキーワードを説明します。

18.11.2.1. ソースおよび宛先 DN のターゲット

特定の状況では、管理者がディレクトリーエントリーを移動できるようにします。ACI で target_from および target_to キーワードを使用すると、ユーザーを有効にしなくても、操作の送信元および宛先を指定できます。

- ACI に設定される別のソースからエントリーを移動します。
- エントリーを ACI のセットとして別の宛先に移動するには、以下のコマンドを実行します。
- ソース DN から既存のエントリーを削除するには、以下を実行します。
- 宛先 DN に新規エントリーを追加するには、以下を行います。

例18.6 target_from および target_to キーワードの使用

たとえば、uid=user,dc=example,dc=com アカウントがユーザーアカウントを cn=staging,dc=example,dc=com エントリーから cn=people,dc=example,dc=com に移動するようにするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (target_from="ldap:///uid=*,cn=staging,dc=example,dc=com")
 (target_to="ldap:///cn=People,dc=example,dc=com")
 (version 3.0; acl "MODDN from"; allow (moddn))
 userdn="ldap:///uid=user,dc=example,dc=com";)
```

注記

ACI は、それらが定義されているサブツリーにのみ適用されます。この例では、ACI は dc=example,dc=com サブツリーにのみ適用されます。

target_from または target_to キーワードが設定されていない場合は、ACI がソースまたは宛先と一致します。

18.11.3. ターゲットルールの高度な使用方法

複数のキーワードを組み合わせることで、複雑なターゲットルールを作成できます。本セクションでは、ターゲットルールの高度な使用例を紹介します。

18.11.3.1. グループの作成およびメンテナンスへのパーミッションの委譲

特定の状況では、管理者はパーミッションを他のアカウントまたはグループに委譲する必要があることがあります。ターゲットキーワードを組み合わせることで、この要求を解決するセキュアな ACI を作成できます。

例18.7 グループの作成およびメンテナンスへのパーミッションの委譲

uid=*user*,ou=People,dc=example,dc=com" アカウントが ou=groups,dc=example,dc=com エントリーでグループを作成および更新できるようにするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (target = "ldap:///cn=*,ou=Groups,dc=example,dc=com")
 targetfilter="(&(objectClass=top)(objectClass=groupOfUniqueNames))")
 (targetattr="cn || uniqueMember || objectClass")
 (version 3.0; acl "example"; allow (read, search, write, add)
 (userdn = "ldap:///uid=test,ou=People,dc=example,dc=com");)
```

前述の例は、セキュリティー上の理由から、特定の制限を追加します。uid=test,ou=People,dc=example,dc=com ユーザー:

- top オブジェクトクラスおよび groupOfUniqueNames オブジェクトクラスが含まれる必要があるオブジェクトを作成できます。
-

account などの追加のオブジェクトクラスを追加できません。たとえば、ローカル認証に Directory Server アカウントを使用して、無効なユーザー ID (例: root ユーザーの 0) を持つ新規ユーザーを作成できなくなります。

18.11.3.2. エントリーと属性の両方をターゲットに設定

target は、DN に基づいてアクセスを制御します。ただし、ワイルドカードと targetattr キーワードと組み合わせて使用する場合は、エントリーと属性の両方をターゲットにすることができます。

例18.8 エントリーと属性の両方をターゲットに設定

uid=user,ou=People,dc=example,dc=com ユーザーが、dc=example,dc=com サブツリー内のすべての組織単位でグループのメンバーを読み取り、検索できるようにするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (target="ldap:///cn=*,dc=example,dc=com")(targetattr="member" || "cn") (version 3.0;
 acl "Allow uid=user to search and read members of groups";
 allow (read, search) (userdn = "ldap:///uid=user,ou=People,dc=example,dc=com");)
```

18.11.3.3. フィルターに一致するエントリーの個別属性のターゲット設定

2 つのターゲットルールで targetattr および targetfilter キーワードを組み合わせる場合は、フィルターに一致するエントリーの特定の属性をターゲットにすることができます。

例18.9 フィルターに一致するエントリーの個別属性のターゲット設定

department 属性が Engineering に設定されている全エントリーの *jpegPhoto* 属性および *manager* 属性を cn=Engineering Admins,dc=example,dc=com グループのメンバーが変更できるようにするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "jpegPhoto|| manager")
 (targetfilter = "(department=Engineering)") (version 3.0;
 acl "Allow engineering admins updating jpegPhoto and manager of department members";
 allow (write) (groupdn = "ldap:///cn=Engineering Admins,dc=example,dc=com");)
```

18.11.3.4. 単一ディレクトリーエントリーのターゲット設定

単一ディレクトリーエントリーを対象にするには、targetattr および targetfilter キーワードを組み合わせます。

例18.10 単一ディレクトリーエントリーのターゲット設定

uid=user,ou=People,dc=example,dc=com ユーザーが ou=Engineering,dc=example,dc=com エントリーで *ou* および *cn* 属性を読み取り、検索できるようにするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=Engineering,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "ou || cn")
 (targetfilter = "(ou=Engineering)") (version 3.0;
 acl "Allow uid=user to search and read engineering attributes";
 allow (read, search) (userdn = "ldap:///uid=user,ou=People,dc=example,dc.com");)
```

以前の例が ou=Engineering,dc=example,dc=com エントリーのみを対象にできるようにするには、ou=Engineering,dc=example,dc=com のサブエントリーは、*ou* 属性を Engineering に設定しないでください。

重要

ディレクトリーの構造が変更すると、これらの種類の ACI が失敗する可能性があります。

または、ターゲットエントリーに保存される属性値を使用して、バインド要求のユーザー入力に一致するバインドルールを作成できます。「値の一致に基づくアクセスの定義」を参照してください。

18.12. パーミッションの定義

パーミッションルールは、ACI に関連付けられた権限と、アクセスを許可または拒否されるかどうかを定義します。

ACI では、以下の強調表示された部分はパーミッションルールになります。

(*target_rule*) (version 3.0; acl "*ACL_name*"; *permission_rule bind_rules*;)

構文

パーミッションルールの一般的な構文は、以下のとおりです。

permission (*rights*)

- ***permission*: ACI がパーミッションを許可するか、拒否するかを設定します。**
- ***rights*: ACI が許可または拒否する権限を設定します。「ユーザーの権利」を参照してください。**

例18.11 パーミッションの定義

ou=People,dc=example,dc=com エントリーに保存されているユーザーが、独自のエントリー内の全属性を検索し、表示するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: dc=People,dc=example,dc=com
changetype: modify
add: aci
aci: (target = "ldap:///ou=People,dc=example,dc=com") (version 3.0;
 acl "Allow users to read and search attributes of own entry"; allow (search, read)
 (userdn = "ldap:///self");)
```

18.12.1. ユーザーの権利

パーミッションルールの権限は、付与または拒否される操作を定義します。ACI では、以下の権限の 1 つまたは複数を設定できます。

表18.1 ユーザーの権利

権利	説明
read	ユーザーがディレクトリーデータを読み込めるかどうかを設定します。このパーミッションは、LDAP の検索操作にのみ適用されます。
write	属性を追加、変更、または削除してユーザーがエントリーを変更できるかどうかを設定します。このパーミッションは、LDAP の **modify** および **modrdn** 操作に適用されます。

権利	説明
add	ユーザーがエントリーを作成できるかどうかを設定します。このパーミッションは、LDAP の **add** 操作にのみ適用されます。
削除	ユーザーがエントリーを削除できるかどうかを設定します。このパーミッションは、LDAP の **delete** 操作にのみ適用されます。
search	ユーザーがディレクトリーデータを検索できるかどうかを設定します。検索結果の一部として返されたデータを表示するには、**search** および **read** 権限を付与します。このパーミッションは、LDAP の検索操作にのみ適用されます。
compare	ユーザーが提供したデータとディレクトリーに保存されているデータを比較できるかどうかを設定します。**compare** 権限では、ディレクトリーは問い合わせに対して成功または失敗のメッセージを返しますが、ユーザーはエントリーや属性の値を見ることはできません。このパーミッションは、LDAP の比較操作にのみ適用されます。
selfwrite	ユーザーがグループから独自の DN を追加または削除できるかどうかを設定します。この権限は、グループ管理にのみ使用されます。
proxy	指定した DN が他のエントリーの権限でターゲットにアクセスできるかどうかを設定します。**proxy** 権限は ACL の範囲内で付与され、その権限が付与されたユーザーやグループは、Directory Server のユーザーとしてコマンドを実行することができます。プロキシー権限を特定のユーザーに制限することはできません。 セキュリティー上の理由から、**proxy** 権限を使用する ACI は、ディレクトリーの最も対象となるレベルに設定してください。
all	**proxy** 以外のすべての権限を設定します。

18.12.2. LDAP 操作に必要な権限

このセクションでは、実行を承認する LDAP 操作のタイプに応じて、ユーザーに付与する必要のある権限を説明します。

- **エントリーの追加:**
  - **追加するエントリーの add パーミッションを付与します。**
  - **エントリーの各属性の値に write パーミッションを付与します。この権限はデフォルトで付与されますが、targattrfilters キーワードを使用して制限できます。**

- エントリーの削除:
  - 削除するエントリーの delete パーミッションを付与します。
  - エントリーの各属性の値に write パーミッションを付与します。この権限はデフォルトで付与されますが、targattrfilters キーワードを使用して制限できます。
- エントリーの属性の変更:
  - 属性タイプで write パーミッションを付与します。
  - 各属性種別の値の write 権限を付与します。この権限はデフォルトで付与されますが、targattrfilters キーワードを使用して制限できます。
- エントリーの RDN の変更:
  - エントリーで write パーミッションを付与します。
  - 新しい RDN で使用される属性タイプの write パーミッションを付与します。
  - 古い RDN の削除に適した権限を付与する場合は、古い RDN で使用される属性タイプの write パーミッションを付与します。
  - 新しい RDN で使用される属性型の値に対して write 権限を付与します。この権限はデフォルトで付与されますが、targattrfilters キーワードを使用して制限できます。
- 属性の値を比較します。
  - 属性タイプで compare パーミッションを付与します。

- エントリーの検索:
  - 検索フィルターで使用される各属性タイプの search パーミッションを付与します。
  - エントリーで使用される属性タイプの read パーミッションを付与します。

18.12.3. アクセス制御と modrdn 操作

ACI を使用して modrdn 操作を明示的に拒否するには、関連するエントリーをターゲットにしますが、targetattr キーワードは省略します。たとえば、cn=example,ou=Groups,dc=example,dc=com グループを定義する ACI を追加するには、*cn* 属性が含まれる ou=people,dc=example,dc=com のエントリーの名前を変更できません。

```
ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (target="ldap:///cn=*,ou=people,dc=example,dc=com")
 (version 3.0; acl "Deny modrdn rights to the example group";
 deny(write) groupdn="ldap:///cn=example,ou=groups,dc=example,dc=com";)
```

18.13. バインドルールの定義

ACI のバインドルールは、Directory Server が ACI を適用するのに必要なバインドパラメーターを定義します。たとえば、以下に基づいてバインドルールを設定できます。

- DNS
- グループメンバーシップまたは割り当てられたロール
- エントリーがバインドする場所
- バインド時に使用する必要のある認証の種類
- バインドが実行される回数または日数

ACI では、以下の強調表示された部分はバインドルールになります。

```
(target_rule) (version 3.0; acl "ACL_name"; permission_rule bind_rules;)
```

構文

バインドルールの一般的な構文は以下のとおりです。

```
keyword comparison_operator "expression"
```

- ***keyword***: bind 操作のタイプを設定します。「頻繁に使用されるバインドルール」を参照してください。
- ***comparison_operator***: 有効な値は = および != で、ターゲットが式で指定されたオブジェクトであるかを示します。キーワードが追加の比較演算子に対応している場合は、該当するセクションで説明されます。
- ***expression***: 式を設定し、引用符で囲む必要があります。式自体は使用するキーワードによって異なります。

18.13.1. 頻繁に使用されるバインドルール

管理者は、以下のバインドキーワードを使用します。

- **userDN:** 「ユーザーベースのアクセスの定義」を参照してください。
- **groupdn:** 「グループベースのアクセスの定義」を参照してください。

さらに、バインドルールはブール値演算子を使用して頻繁に組み合わせられます。詳細は、「ブール演算子を使用したバインドルールの組み合わせ」を参照してください。

18.13.1.1. ユーザーベースのアクセスの定義

userdn キーワードを使用すると、1 つまたは複数の DN に基づいてアクセスを許可または拒否でき、以下の構文を使用します。

-

```
userdn comparison_operator "ldap:///distinguished_name || ldap:///distinguished_name || ..."
```

式の DN を以下のように設定します。

- DN: 「userdn キーワードでの DN の使用」を参照してください。
- LDAP フィルター: 「LDAP フィルターで userdn キーワードの使用」を参照してください。
- anyone エイリアス: 「匿名アクセスの付与」を参照してください。
- all エイリアス: 「認証済みユーザーへのアクセスの付与」を参照してください。
- self エイリアス: 「ユーザーが空のエントリーにアクセスできるようにする」を参照してください。
- parent エイリアス: 「ユーザーの子エントリーへのアクセス設定」を参照してください。

注記

LDAP URL 内でホスト名またはポート番号を指定しないでください。URL は常にローカルサーバーに適用されます。

18.13.1.1.1. userdn キーワードでの DN の使用

userdn キーワードを DN に設定して、ACI を一致するエントリーのみに適用します。複数のエントリーを照合するには、DN で * ワイルドカードを使用します。

userdn キーワードを DN とともに使用するには、以下の構文を使用します。

```
userdn comparison_operator ldap:///distinguished_name
```

例18.12 userdn キーワードでの DN の使用

uid=admin,ou=People,dc=example,dc=com ユーザーが ou=People,dc=example,dc=com エントリーで他のすべてのユーザーの *manager* 属性を読み取るようにするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="manager") (version 3.0; acl "Allow uid=admin reading manager attribute";
 allow (search, read) userdn = "ldap:///uid=admin,ou=People,dc=example,dc=com";)
```

18.13.1.1.2. LDAP フィルターで userdn キーワードの使用

ユーザーへのパーミッションを動的に許可または拒否するには、LDAP フィルターで userdn キーワードを使用します。

```
userdn comparison_operator "ldap:///distinguished_name??scope?(filter)"
```

注記

LDAP フィルターは * ワイルドカードをサポートします。

例18.13 LDAP フィルターで userdn キーワードの使用

***department* 属性が Human Resources に設定されたユーザーを有効にするには、ou=People,dc=example,dc=com エントリーでユーザーの *homePostalAddress* 属性を更新します。**

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="homePostalAddress") (version 3.0;
 acl "Allow HR setting homePostalAddress"; allow (write)
 userdn = "ldap:///ou=People,dc=example,dc=com??sub?(department=Human Resources)";)
```

18.13.1.1.3. 匿名アクセスの付与

特定の状況では、管理者はディレクトリー内のデータへの匿名アクセスを設定します。匿名アクセスは、以下を指定してディレクトリーにバインドできることを意味します。

- バインド DN およびパスワードなし
- 有効なバインド DN およびパスワード

匿名アクセスを設定するには、bind ルールの userdn キーワードで ldap:///anyone 式を使用します。

```
userdn comparison_operator "ldap:///anyone"
```

例18.14 匿名アクセスの付与

認証のないすべてのユーザーが、ou=People,dc=example,dc=com エントリーで *sn* 属性、*givenName* 属性、および *telephoneNumber* 属性を読み取りおよび検索できるようにするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="sn" || targetattr="givenName" || targetattr = "telephoneNumber")
 (version 3.0; acl "Anonymous read, search for names and phone numbers";
 allow (read, search) userdn = "ldap:///anyone")
```

18.13.1.1.4. 認証済みユーザーへのアクセスの付与

特定の状況では、管理者は匿名バインドを除き、Directory Server に正常にバインドできるユーザーにパーミッションを付与します。この機能を設定するには、bind ルールの userdn キーワードで ldap:///all 式を使用します。

```
userdn comparison_operator "ldap:///all"
```

例18.15 認証済みユーザーへのアクセスの付与

認証されたユーザーが ou=example,ou=groups,dc=example,dc=com グループからのメンバーとして追加および削除できるようにするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=example,ou=Groups,dc=example,dc=com
changetype: modify
add: aci
```

```
aci: (targetattr="member") (version 3.0;
 acl "Allow users to add/remove themselves from example group";
 allow (selfwrite) userdn = "ldap:///all")
```

18.13.1.1.5. ユーザーが空のエントリーにアクセスできるようにする

ユーザーの独自のエントリーへのアクセスを許可または拒否する **ACI** を設定するには、**bind** ルールの **userdn** キーワードで **ldap:///self** 式を使用します。

```
userdn comparison_operator "ldap:///self"
```

例18.16 ユーザーが空のエントリーにアクセスできるようにする

ou=People,dc=example,dc=com エントリーのユーザーが独自の ***userPassword*** 属性を更新できるようにするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="userPassword") (version 3.0;
 acl "Allow users updating their password";
 allow (write) userdn = "ldap:///self")
```

18.13.1.1.6. ユーザーの子エントリーへのアクセス設定

バインド **DN** がターゲットエントリーの親である場合にのみエントリーへのアクセスを許可または拒否されるように設定するには、**bind** ルールの **userdn** キーワードで **self:///parent** 式を使用します。

```
userdn comparison_operator "ldap:///parent"
```

例18.17 ユーザーの子エントリーへのアクセス設定

cn=*user*,ou=People,dc=example,dc=com ユーザーが独自のサブエントリー (**cn=example,cn=*user*,ou=People,dc=example,dc=com** など) の ***manager*** 属性を更新できるようにするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=user,ou=People,dc=example,dc=com
changetype: modify
add: aci
```

```
aci: (targetattr="manager") (version 3.0;
 acl "Allow cn=user to update manager attributes";
 allow (write) userdn = "ldap:///parent")
```

18.13.1.2. グループベースのアクセスの定義

グループベースの ACI を使用すると、グループへのユーザーの追加、またはグループからのユーザーの削除により、アクセスを管理できます。グループメンバーシップに基づく ACI を設定するには、groupdn キーワードを使用します。ユーザーが指定された 1 つまたは複数のグループのメンバーである場合は、ACI が一致します。

groupdn キーワードを使用すると、Directory Server は以下の属性に基づいてグループメンバーシップを検証します。

- *member*
- *uniqueMember*
- *memberURL*
- *memberCertificateDescription*

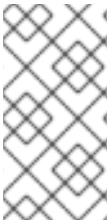
groupdn キーワードでルールをバインドするには、以下の構文を使用します。

```
groupdn comparison_operator "ldap:///distinguished_name || ldap:///distinguished_name || ..."
```

式の DN を以下のように設定します。

- DN。「groupdn キーワードでの DN の使用」を参照してください。
- LDAP フィルター。「LDAP フィルターで groupdn キーワードの使用」を参照してください。

1 つのバインドルールに複数の DN を設定する場合は、認証されたユーザーがこれらのグループのいずれかのメンバーの場合、Directory Server は ACI を適用します。ユーザーを複数のグループのメンバーとして設定するには、複数の groupdn キーワードを使用して、ブール値 and 演算子を使用して組み合わせます。詳細は、「ブール演算子を使用したバインドルールの組み合わせ」を参照してください。



注記

LDAP URL 内でホスト名またはポート番号を指定しないでください。URL は常にローカルサーバーに適用されます。

18.13.1.2.1. groupdn キーワードでの DN の使用

ACI をグループのメンバーに適用するには、groupdn キーワードをグループの DN に設定します。

DN に設定された groupdn キーワードは、以下の構文を使用します。

```
groupdn comparison_operator ldap:///distinguished_name
```

例18.18 groupdn キーワードでの DN の使用

cn=example,ou=Groups,dc=example,dc=com グループのメンバーが ou=People,dc=example,dc=com のエントリーの *manager* 属性を検索および読み取るようにするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="manager") (version 3.0;
 acl "Allow example group to read manager attribute";
 allow (search, read) groupdn = "ldap:///cn=example,ou=Groups,dc=example,dc=com";)
```

18.13.1.2.2. LDAP フィルターで groupdn キーワードの使用

groupdn キーワードを使用した LDAP フィルターを使用すると、ACI に一致させるために、認証されたユーザーがフィルター検索で返されるグループの少なくとも 1 つのメンバーでなければならないことを定義できます。

LDAP フィルターが含まれる groupdn キーワードは以下の構文を使用します。

```
groupdn comparison_operator "ldap:///distinguished_name??scope?(filter)"
```

注記

LDAP フィルターは * ワイルドカードをサポートします。

例18.19 LDAP フィルターで groupdn キーワードの使用

dc=example,dc=com のグループのメンバーや、*manager* 属性が example に設定されているサブツリーを有効にするには、ou=People,dc=example,dc=com のエントリーの*homePostalAddress* を更新します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="homePostalAddress") (version 3.0;
 acl "Allow manager=example setting homePostalAddress"; allow (write)
 userdn = "ldap:///dc=example,dc=com??sub?(manager=example)";)
```

18.13.2. さらなるバインドルール

このセクションでは、頻繁に使用されないバインドルールを説明します。

18.13.2.1. 値の一致に基づくアクセスの定義

バインドルールの userattr キーワードを使用して、ディレクトリーとターゲットエントリーにバインドするのに使用されるエントリー間でどの属性が一致するかを指定します。

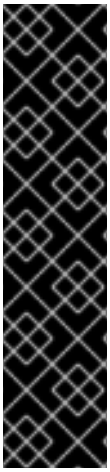
userattr キーワードは、以下の構文を使用します。

```
userattr comparison_operator "attribute_name#bind_type_or_attribute_value
```

詳細は、以下を参照してください。

- **「USERDN バインドタイプの使用」**

- 「GROUPDN バインドタイプの使用」
- 「ROLEDN バインドタイプの使用」
- 「SELFDN バインドタイプの使用」
- 「LDAPURL バインドタイプの使用」
- 「バインド DN とターゲット DN の属性値の一致」



重要

デフォルトでは、Directory Server は、作成したエントリーに対するアクセス権限を評価します。ただし、同じレベルのユーザーオブジェクトを防ぐために、Directory Server は、userattr キーワードを使用した場合に、ACI を設定したエントリーに add パーミッションを付与しません。この動作を設定するには、parent キーワードとともに userattr キーワードを使用して、レベル 0 にもパーミッションを付与します。

継承の詳細は、「継承による userattr キーワードの使用」を参照してください。

18.13.2.1.1. USERDN バインドタイプの使用

バインディングユーザー DN が属性に保存されている DN と一致する場合に ACI を適用するには、USERDN バインドタイプを使用します。

USERDN バインドタイプの userattr キーワードには、以下の構文を使用します。

```
userattr comparison_operator "attribute_name#USERDN"
```

例18.20 USERDN バインドタイプの使用

マネージャーに対し、すべての権限を独自の関連付けの *telephoneNumber* 属性に付与するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
```

```
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "telephoneNumber")
 (version 3.0; acl "Manager: telephoneNumber";
 allow (all) userattr = "manager#USERDN";)
```

前述の ACI は、ou=People,dc=example,dc=com のエントリーに対して操作を行ったユーザの DN が、このエントリーの*manager* 属性に格納されている DN と一致すれば、真と評価されます。

18.13.2.1.2. GROUPDN バインドタイプの使用

バインディングユーザー DN が属性に設定されたグループのメンバーである場合に ACI を適用するには、GROUPDN バインドタイプを使用します。

GROUPDN バインドタイプの userattr キーワードには、以下の構文を使用します。

```
userattr comparison_operator "attribute_name#GROUPDN"
```

例18.21 GROUPDN バインドタイプの使用

ユーザーに、ou=Social Committee,ou=Groups,dc=example,dc=com エントリーを所有するグループエントリーを削除する権限を付与するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=Social Committee,ou=Groups,dc=example,dc=com
changetype: modify
add: aci
aci: (target="ou=Social Committee,ou=Groups,dc=example,dc=com)
 (targattrfilters="del=objectClass:(objectClass=groupOfNames)")
 (version 3.0; acl "Delete Group";
 allow (delete) userattr = "owner#GROUPDN";)
```

操作を実行するユーザーの DN が *owner* 属性で指定されたグループのメンバーである場合に、以前の ACI が true になります。

指定のグループは動的グループで、グループの DN はデータベースの任意の接尾辞にすることができます。しかし、このタイプの ACI をサーバーが評価するには、リソースを大量に必要とします。

ターゲットエントリーと同じ接尾辞の下にある静的グループを使用している場合は、パフォーマンスを改善するために以下の式を使用します。

```
userattr comparison_operator "ldap:///distinguished_name?attribute_name#GROUPDN"
```

18.13.2.1.3. ROLEDN バインドタイプの使用

バインディングユーザーが属性で指定されたロールに属する場合に ACI を適用するには、ROLEDN バインドタイプを使用します。

ROLEDN バインドタイプの userattr キーワードには、以下の構文を使用します。

```
userattr comparison_operator "attribute_name#ROLEDN"
```

例18.22 ROLEDN バインドタイプの使用

cn=Administrators,dc=example,dc=com ロールを持つユーザーが ou=People,dc=example,dc=com のエントリーの *manager* 属性を検索および読み取るようにするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (version 3.0; acl "Allow example role owners to read manager attribute";
 allow (search, read) roledn="ldap:///cn=Administrators,dc=example,dc=com";)
```

指定のロールはデータベースの任意の接尾辞の下に置くことができます。フィルターされたロールも使用している場合、このタイプの ACI の評価は、サーバー上の多くのリソースを使用します。

静的ロール定義を使用し、ロールエントリーがターゲットエントリーと同じ接尾辞下にある場合は、パフォーマンスを向上させるために以下の式を使用します。

```
userattr comparison_operator "ldap:///distinguished_name?attribute_name#ROLEDN"
```

18.13.2.1.4. SELFDN バインドタイプの使用

SELFDN バインドタイプを使用すると、バインドされたユーザーの DN がエントリーの単一値属性に設定されている場合にパーミッションを付与できます。

SELFDN バインドタイプの userattr キーワードには、以下の構文を使用します。

```
userattr comparison_operator "attribute_name#SELFDN"
```

例18.23 SELFDN バインドタイプの使用

ユーザーが *ipatokenOwner* 属性にバインドユーザーの DN が設定された ipatokenuniqueid=*,cn=otp,dc=example,dc=com エントリーを追加できるようにするには、次のコマンドを実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=otp,dc=example,dc=com
changetype: modify
add: aci
aci: (target = "ldap:///ipatokenuniqueid=*,cn=otp,dc=example,dc=com")
 (targetfilter = "(objectClass=ipaToken)")(version 3.0;
 acl "token-add-delete"; allow (add) userattr = "ipatokenOwner#SELFDN";)
```

18.13.2.1.5. LDAPURL バインドタイプの使用

バインド DN がターゲットエントリーの属性で指定されたフィルターと一致する場合に ACL を適用するには、LDAPURL バインドタイプを使用します。

LDAPURL バインドタイプの userattr キーワードには、以下の構文を使用します。

```
userattr comparison_operator "attribute_name#LDAPURL"
```

例18.24 LDAPURL バインドタイプの使用

ldap:///ou=People,dc=example,dc=com??one?(uid=*user) に設定した *aciurl* 属性が含まれるユーザーオブジェクトに読み取りパーミッションおよび検索パーミッションを付与するには、以下を実行します。**

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "*")
 (version 3.0; acl "Allow read,search "; allow (read,search)
 (userattr = "aciurl#LDAPURL);)
```

18.13.2.1.6. バインド DN とターゲット DN の属性値の一致

バインド DN エントリーとターゲットエントリーの両方に同じ値に設定された属性が含まれる場合に ACL を適用するには、以下の構文を使用します。

```
userattr comparison_operator "attribute_name#value"
```

例18.25 バインド DN とターゲット DN の属性値の一致

l 属性が office_1 に設定されたツリー内の操作およびユーザーの両方に読み取りパーミッションおよび検索パーミッションを付与するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr != "userPassword")
 (version 3.0; acl "Users in the same location";
 allow (read,search) userattr = "l#office_1";)
```

18.13.2.1.7. 継承による userattr キーワードの使用

userattr キーワードを使用してターゲットエントリーにバインドするために使用されるエントリーを関連付ける場合、ACI は指定されたターゲットにのみ適用され、その下のエントリーには適用されません。特定の状況下では、管理者は ACI の適用範囲を、対象となるエントリーよりも数レベル広げたいと考えます。これは、parent キーワードを使用して、ACI を継承するターゲットよりも低いレベルの数を指定できます。

parent キーワードで userattr キーワードを使用する場合、構文は以下のようになります。

```
userattr comparison_operator
"parent[inheritance_level].attribute_name#bind_type_or_attribute_value
```

- *inheritance_level*: ターゲットが ACI を継承するレベルの数を指定します。ターゲットエントリーの下に、5 つのレベル (0、1、2、3、4) を追加できます。ゼロ (0) はターゲットエントリーを示します。
- *attribute_name*: userattr または groupattr のキーワードでターゲットとする属性。

bind_type_or_attribute_value: USERDN などの属性値またはバインドタイプを設定します。

以下は例になります。

```
userattr = "parent[0,1].manager#USERDN"
```

このバインドルールは、バインド DN がターゲットエントリーのマネージャー属性と一致する場合に true になります。バインドルールが true であるときに付与されるパーミッションは、ターゲットエントリーと、その下のすべてのエントリーに適用されます。

例18.26 継承による userattr キーワードの使用

ユーザーの DN が *owner* 属性に設定されている cn=Profiles,dc=example,dc=com エントリー、および cn=mail,cn=Profiles,dc=example,dc=com および cn=news,cn=Profiles,dc=example,dc=com を含む第 1 レベルの子エントリーの読み取りと検索を可能にするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=Profiles,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="*") (version 3.0; acl "Profile access",
 allow (read,search) userattr="parent[0,1].owner#USERDN" ;)
```

18.13.2.2. 特定の IP アドレスまたは範囲からのアクセスの定義

バインドルールの ip キーワードを使用すると、特定の IP アドレスまたは IP アドレスの範囲からのアクセスを許可または拒否できます。

ip キーワードでルールをバインドするには、以下の構文を使用します。

```
ip comparison_operator "IP_address_or_range"
```

例18.27 バインドルールでの IPv4 アドレス範囲の使用

192.0.2.2/24 ネットワークから dc=example,dc=com エントリーへのアクセスを拒否するには、以下のコマンドを実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
```

```
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "*") (version 3.0;acl "Deny 192.0.2.2/24"; deny (all)
 (userdn = "ldap:///anyone") and (ip != "192.0.2.");)
```

例18.28 バインドルールでの IPv6 アドレス範囲の使用

2001:db8::/64 ネットワークから dc=example,dc=com エントリーへのアクセスを拒否するには、以下のコマンドを実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "*") (version 3.0;acl "Deny 2001:db8::/64"; deny (all)
 (userdn = "ldap:///anyone") and (ip != "2001:db8::");)
```

18.13.2.3. 特定のホストまたはドメインからアクセスの定義

バインドルールの dns キーワードを使用すると、特定のホストまたはドメインからのアクセスを許可または拒否できます。

警告

DNS を使用して Directory Server が完全修飾ドメイン名 (FQDN) への接続 IP アドレスを解決できない場合、サーバーはこのクライアントの dns バインディングルールを持つ ACI を適用しません。

クライアント IP アドレスが DNS を使用して解決できない場合は、代わりに ip キーワードおよび IP アドレスを使用してください。「特定の IP アドレスまたは範囲からのアクセスの定義」を参照してください。

dns キーワードでルールをバインドするには、以下の構文を使用します。

```
dns comparison_operator "host_name_or_domain_name"
```

例18.29 特定のホストからのアクセスの定義

client.example.com ホストから dc=example,dc=com エントリーへのアクセスを拒否するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "*") (version 3.0;acl "Deny client.example.com"; deny (all)
 (userdn = "ldap:///anyone") and (dns != "client.example.com");)
```

例18.30 特定のドメインからアクセスの定義

example.com ドメイン内のすべてのホストから dc=example,dc=com エントリーへのアクセスを拒否するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "*") (version 3.0;acl "Deny example.com"; deny (all)
 (userdn = "ldap:///anyone") and (dns != "*.example.com");)
```

18.13.2.4. 接続に一定レベルのセキュリティーの要求

接続のセキュリティーは、操作を処理するために必要な最低限の鍵の強度を設定する SSF (Security Strength Factor) によって決定されます。バインドルールで ssf キーワードを使用すると、接続が一定レベルのセキュリティーを使用する必要があります。これにより、パスワード変更などの操作を強制的に、暗号化された接続上で実行できます。

すべての操作の SSF 値は、TLS 接続と SASL バインドの間の値が高くなります。これは、サーバーが TLS で実行されるように設定され、レプリカ合意が SASL/GSSAPI に対して設定されている場合は、操作の SSF が利用可能な暗号化タイプがよりセキュアであることを意味します。

ssf キーワードでルールをバインドするには、以下の構文を使用します。

```
ssf comparison_operator key_strength
```

以下の比較演算子を使用できます。

- = (等しい)
- ! (等しくない)
- < (より小さい)
- > (より大きい)
- <= (より小さいか等しい)
- >= (より大きいか等しい)

key_strength パラメーターが 0 に設定されている場合、LDAP 操作にセキュアな操作は必要ありません。

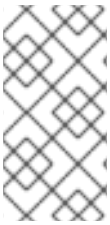
例18.31 接続に一定レベルのセキュリティーの要求

dc=example,dc=com エントリーのユーザーが、SSF が 128 以上の場合にのみ、*userPassword* 属性を更新できるように設定する場合は、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "userPassword") (version 3.0;
 acl "Allow users updating own userPassword";
 allow (write) (userdn = "ldap:///self") (ssf >= "128");)
```

18.13.2.5. 曜日の特定の日におけるアクセスの定義

バインドルールの dayofweek キーワードを使用すると、曜日に基づいてアクセスを許可または拒否できます。



注記

Directory Server はサーバー上で時間を使用して ACI を評価しますが、クライアントの時間ではありません。

dayofweek キーワードでルールをバインドするには、以下の構文を使用します。

```
dayofweek comparison_operator "comma-separated_list_of_days"
```

例18.32 特定の曜日にアクセスの付与

毎週日曜日と日曜日のサーバーにバインドするために uid=*user*,ou=People,dc=example,dc=com ユーザーエントリーのアクセスを拒否するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (version 3.0; acl "Deny access on Saturdays and Sundays";
 deny (all)
 (userdn = "ldap:///uid=user,ou=People,dc=example,dc=com") and
 (dayofweek = "Sun,Sat");)
```

18.13.2.6. 特定の時刻におけるアクセスの定義

バインドルールで timeofday キーワードを使用すると、時間帯に基づいてアクセスを許可または拒否することができます。

注記

Directory Server はサーバー上で時間を使用して ACI を評価しますが、クライアントの時間ではありません。

timeofday キーワードでルールをバインドするには、以下の構文を使用します。

```
timeofday comparison_operator "time"
```

以下の比較演算子を使用できます。

- = (等しい)
- ! (等しくない)
- < (より小さい)
- > (より大きい)
- <= (より小さいか等しい)
- >= (より大きいか等しい)

重要

timeofday キーワードには、24 時間形式で時間を指定する必要があります。

例18.33 特定の時刻におけるアクセスの定義

uid=*user*,ou=People,dc=example,dc=com ユーザーエントリーへのアクセスを拒否するには、6pm から 0am までのサーバーにバインドします。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
 aci: (version 3.0; acl "Deny access between 6pm and 0am";
 deny (all)
 (userdn = "ldap:///uid=user,ou=People,dc=example,dc=com") and
 (timeofday >= "1800" and timeofday < "2400");)
```

18.13.2.7. 認証方法に基づいたアクセスの定義

bind ルールの authmethod キーワードは、サーバーに接続する際にクライアントが使用する認証方

法を設定し、ACI を適用します。

auth キーワードでルールをバインドするには、以下の構文を使用します。

```
authmethod comparison_operator "authentication_method"
```

以下の認証方法を設定できます。

- none: 認証は不要で、匿名のアクセスを表します。これがデフォルトになります。
- simple: クライアントは、ディレクトリーにバインドするユーザー名とパスワードを提供する必要があります。
- SSL: クライアントは、データベース、スマートカード、または他のデバイスのいずれかで TLS 証明書を使用してディレクトリーにバインドする必要があります。証明書ベースの認証の詳細は、「証明書ベースのクライアント認証の使用」を参照してください。
- SASL: クライアントは、Simple Authentication and Security Layer (SASL) 接続を介してディレクトリーにバインドする必要があります。bind ルールでこの認証方法を使用する場合は、EXTERNAL などの SASL メカニズムも指定します。

例18.34 EXTERNAL SASL 認証方法を使用した接続でのみアクセスのみの有効化

接続が証明書ベースの認証メソッドまたは SASL を使用していない場合にサーバーへのアクセスを拒否するには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (version 3.0; acl "Deny all access without certificate"; deny (all)
 (authmethod = "none" or authmethod = "simple");)
```

18.13.2.8. ロールに基づくアクセスの定義

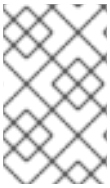
bind ルールの roledn キーワードを使用すると、1 つまたは複数のロールが設定されたユーザーへのアクセスを許可または拒否できます。

注記

Red Hat は、ロールの代わりにグループを使用することを推奨します。ロールおよび制限の詳細は、「ロールの概要」を参照してください。

roledn キーワードでルールをバインドするには、以下の構文を使用します。

```
userdn comparison_operator "ldap:///distinguished_name || ldap:///distinguished_name || ..."
```



注記

DN にコンマが含まれている場合は、バックスラッシュでエスケープしてください。

例18.35 ロールに基づくアクセスの定義

***nsRole* 属性で cn=Human Resources,ou=People,dc=example,dc=com ロールを設定したユーザーが ou=People,dc=example,dc=com のエントリーの *manager* 属性を検索および読み取るようにするには、以下を実行します。**

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="manager") (version 3.0;
 acl "Allow manager role to update manager attribute";
 allow (search, read) roledn = "ldap:///cn=Human Resources,ou=People,dc=example,dc=com";)
```

18.13.3. ブール演算子を使用したバインドルールの組み合わせ

複雑なバインドルールを作成する場合は、AND、OR、および NOT のブール値演算子を使用すると、複数のキーワードを組み合わせることができます。

バインドルールとブール演算子を組み合わせた構文は以下の通りです。

```
bind_rule_1 boolean_operator bind_rule_2...
```

例18.36 ブール演算子を使用したバインドルールの組み合わせ

cn=Administrators,ou=Groups,dc=example,com および cn=Operators,ou=Groups,dc=example,com の両方のグループのメンバーであるユーザーが、ou=People,dc=example,dc=com のエントリーを読み取り、検索、追加、更新、および削除できるように設定するには、次のコマンドを実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (target="ldap:///ou=People,dc=example,dc=com") (version 3.0;
 acl "Allow members of administrators and operators group to manage users";
 allow (read, search, add, write, delete)
 groupdn = "ldap:///cn=Administrators,ou=Groups,dc=example,com" AND
 groupdn = "ldap:///cn=Operators,ou=Groups,dc=example,com";)
```

Directory Server によるブール値演算子の評価方法

Directory Server は以下のルールを使用してブール値演算子を評価します。

- **左から右へのすべての式。**

  以下の例では、*bind_rule_1* が最初に評価されます。

  ```
  (bind_rule_1) OR (bind_rule_2)
  ```

- **一番内側から外側に向かって、親表現が優先されます。**

  以下の例では、*bind_rule_2* を最初に評価し、次に *bind_rule_3* を評価します。

  ```
  (bind_rule_1) OR ((bind_rule_2) AND (bind_rule_3))
  ```

- **AND または OR 演算子の前に NOT。**

  以下の例では、*bind_rule_2* が最初に評価されます。

  ```
  (bind_rule_1) AND NOT (bind_rule_2)
  ```

AND および OR 演算子には優先順位がありません。

18.14. エントリーのアクセス権利の確認 (GET EFFECTIVE RIGHTS)

ユーザーが特定のエントリー内の属性に対して持っているアクセス権を見つけることは、管理者がアクセス権限を見つけて制御するのための便利な方法を提供します。

Get effective rights は、ディレクトリー検索を拡張して、ユーザーが特定のエントリーに対してどのようなアクセス権 (読み取り、検索、書き込みと自己書き込み、追加、削除など) を持っているかを表示する方法です。

Directory Server では、通常のユーザーは、表示できるエントリーに対する権限を確認して、他の人による個人エントリーへのアクセスを確認することができます。Directory Manager は、あるユーザーが別のユーザーに属する権限を確認できます。

エントリーの実効権限を確認することが便利な状況は 2 つあります。

- 管理者は、ディレクトリーに対するアクセス制御手順をより適切に整理するために、get effective rights コマンドを使用できます。あるグループのユーザーが閲覧または編集できる内容を、別のグループと比較して制限する必要があることがよくあります。たとえば、QA Managers グループのメンバーには、*manager* や *salary* などの属性を検索および読み取る権利がありますが、HR Group メンバーのみが変更または削除する権限を持ちます。ユーザーまたはグループの実効権限を確認する方法は、適切なアクセス制御が有効であることを確認する方法です。

- ユーザーは、get effective rights コマンドを実行して、個人エントリーで表示または変更できる属性を確認することができます。たとえば、ユーザーは *homePostalAddress* や *cn* などの属性にアクセスできますが、*manager* 属性および *salary* 属性への読み取りアクセスしかできません。

get effective rights 検索には、以下の 3 人があります。1 つ目は、search コマンド（ 要求側 ）を実行しているユーザーです。権限がチェックされます（さまざまな対象者は、A がエントリー B に対して持っている権限を確認）します。この権限がチェックされる人は GER サブジェクトで、その権限は GER サブジェクト であり、それらの権限は検索の対象になります。ユーザーの権利(Entry B)を持つエントリーまたはエントリーは、検索ベース または 検索ベースになります。

18.14.1. Get Effective Rights 検索表示される権限

Directory Server コンソールのエントリーを表示し、コマンドラインで検索するときの両方の get

effective rights 検索では、ユーザー A がユーザー B のエントリーに対して必要とする権限を表示します。

任意のエントリーに指定できるアクセス権には、2 種類があります。1 つ目は上位の権利で、エントリー自体に対する権利 です。つまり、ユーザー A がユーザー B のエントリー全体に対して実行できる操作の種類を意味します。第 2 レベルのアクセス権はより詳細で、ユーザー A が ある属性に対してどのような権利 を有しているかを示しています。この場合、ユーザー A は同じエントリーで異なる属性のアクセスパーミッションがある可能性があります。ユーザーに許可されるアクセス制御は、そのエントリーに対する 有効な 権限です。

以下に例を示します。

```
entryLevelRights: vadn
attributeLevelRights: givenName:rscWO, sn:rscW, objectClass:rsc, uid:rsc, cn:rscW
```

表18.2「エントリーの権限」および 表18.3「属性権」は、エントリーおよび属性へのアクセス権限をそれぞれ表示し、get effective rights 検索で返されます。

表18.2 エントリーの権限

パーミッション	説明
a	エントリーを追加します。
d	このエントリーを削除します。
n	DN の名前を変更します。
v	エントリーを表示します。

表18.3 属性権

パーミッション	説明
r	読み取り。
s	検索。
w	書き込み (mod-add)。
o	抹消 (mod-del)。削除に類似しています。

パーミッション	説明
c	比較。
W	自己書き込み。
O	自己削除。

18.14.2. Get Effective Rights 検索の形式

Get effective rights (GER と呼ばれることもあります) は、拡張ディレクトリー検索です。GER パラメーターは、-E オプションを定義して、ldapsearch コマンドで LDAP コントロールを渡します。(-E オプションなしで ldapsearch を実行すると、get effective rights 情報なしに、エントリーが通常通りに返されます。)

```
# ldapsearch -x -D bind_dn -W -p server_port -h server_hostname -E
[!]1.3.6.1.4.1.42.2.27.9.5.2=:GER_subject (searchFilter) attributeList
```

- -b は、GER サブジェクトの検索に使用されるサブツリーまたはエントリーのベース DN です。

  検索ベースが特定のエントリー DN である場合や、1 つのエントリーのみが返される場合、結果には要求元がその特定のエントリーに対して所有している権利が表示されます。検索ベースの下にある複数のエントリーがフィルターと一致する場合、検索は一致するすべてのエントリーを返し、各エントリーに対する要求側の権限で返します。

- 1.3.6.1.4.1.42.2.27.9.5.2 は、get effective rights control の OID です。

- 感嘆符 (!) は、サーバーがこの制御 (!) をサポートしていない場合に、検索操作でエラーを返すか、無視して通常通りの検索を行うか (nothing) を指定します。

- GER_subject は、権限を確認するユーザーです。GER_subject が空白 (dn:) のままの場合、匿名ユーザーの権限が返されます。

- 任意の attributeList は、Get Effective Rights 結果を指定された属性またはオブジェクトクラスに制限します。通常の ldapsearch の場合のように、*mail* などの特定の属性を指定できます。属性が表示されない場合は、エントリーの present 属性がすべて返されます。アスタリスク (*) を使用すると、エントリーに対して可能なすべての属性の権限が返されます (既存の属性と存在しない属性の両方)。プラス記号 (+) を使用すると、エントリーの操作属性が返されま

す。「Non-Existent 属性の Get Effective Rights 検索の例」および「特定の属性またはオブジェクトクラスの Get Effective Rights 検索の例」で特定属性の権限を確認する例。

get effective rights 検索の要点は、GER 対象者 (-E) が検索対象者 (-b) に対してどのような権利を持っているかを確認できることです。get effective rights 検索は通常の ldapsearch で、検索パラメーターに一致するエントリーを検索し、その情報を返します。get effective rights オプションは、これらの検索結果に追加の情報を加え、特定のユーザーがそれらの検索結果に対してどのような権利を持っているかを示します。その GER サブジェクトユーザーは、リクエスター (-D が -E と同じ) でも、別のユーザーでも構いません。

要求元が (Directory Manager ではなく) 一般ユーザーである場合、要求元は GER サブジェクトが要求元独自のエントリーに存在することのみを確認できます。つまり、John Smith が Babs Jensen の持つ有効な権利を確認するために要求を実行した場合、John Smith は Babs Jensen が自分のエントリーで持つ有効な権利しか得ることができません。他のエントリーはすべて、有効権限が不十分なアクセスエラーを返します。

get effective rights 検索を実行する際の通常ユーザーには、一般的な 3 つのシナリオがあります。

- ユーザー A は、他のディレクトリーエントリーに対する権利を確認します。
- ユーザー A は、自身のエントリーに必要な権限をチェックします。
- ユーザー A は、ユーザー B がユーザー A のエントリーに対して持っている権利をチェックします。

get effective rights 検索には、属性の権利を確認するための柔軟な方法がいくつかあります。

18.14.3. GER 検索の例

GER 検索を実行する方法は複数あります。返される情報のタイプや、検索されるエントリーおよび属性のタイプによって異なります。

18.14.3.1. アクセス権限の確認に関する一般的な例

効果的な権利検索のための一般的なシナリオとして、一般ユーザーが自身の個人的なエントリーにどのような変更を加えることができるかを判断することがあります。

たとえば、Ted Morris は、エントリーに対する権限を確認します。-D と -E オプションの両方により、そのエントリーは要求元として付与されます。-b オプションには、個人エントリーを確認するため、その DN も含まれます。

例18.37 個人の権利の確認 (ユーザー A からユーザー A)

```
# ldapsearch -x -p 389 -h server.example.com -D
"uid=tmorris,ou=people,dc=example,dc=com" -W -b
"uid=tmorris,ou=people,dc=example,dc=com" -E
'!1.3.6.1.4.1.42.2.27.9.5.2=:dn:uid=tmorris,ou=people,dc=example,dc=com' "
(objectClass=*)"

 dn: uid=tmorris,ou=People,dc=example,dc=com
 givenName: Ted
 sn: Morris
 ou: IT
 ou: People
 l: Santa Clara
 manager: uid=jsmith,ou=People,dc=example,dc=com
 roomNumber: 4117
 mail: tmorris@example.com
 facsimileTelephoneNumber: +1 408 555 5409
 objectClass: top
 objectClass: person
 objectClass: organizationalPerson
 objectClass: inetOrgPerson
 uid: tmorris
 cn: Ted Morris
 userPassword: {SSHA}bz0uCmHZM5b357zwrCUCJs1IOHtMD6yqPyhxBA==
 entryLevelRights: v
 attributeLevelRights: givenName:rsc, sn:rsc, ou:rsc, l:rsc, manager:rsc,
roomNumber:rscwo, mail:rscwo, facsimileTelephoneNumber:rscwo, objectClass:rsc,
uid:rsc, cn:rsc, userPassword:wo
```

Ted Morris は、たとえば、管理職であったり、IT や人事など、他のユーザーのエントリーを編集しなければならない部署で働いていたりします。この場合、例18.38「別のユーザーの権限を個人的に確認 (ユーザー A からユーザー B へ)」のように、Ted (-D) が Dave Miller のエントリー (-b) に対する自分の権利 (-E) を確認しているように、他のユーザーのエントリーに対して自分がどのような権利を持っているかを確認したい場合があります。

例18.38 別のユーザーの権限を個人的に確認 (ユーザー A からユーザー B へ)

```
# ldapsearch -p 389 -h server.example.com -D
"uid=tmorris,ou=people,dc=example,dc=com" -W -b
"uid=dmiller,ou=people,dc=example,dc=com" -E
'!1.3.6.1.4.1.42.2.27.9.5.2=:dn:uid=tmorris,ou=people,dc=example,dc=com' "
(objectClass=*)"

dn: uid=dmiller,ou=People,dc=example,dc=com
... snip ...
```

```
entryLevelRights: vad
attributeLevelRights: givenName:rscwo, sn:rscwo, ou:rscwo, l:rscwo, manager:rsc,
roomNumber:rscwo, mail:rscwo, facsimileTelephoneNumber:rscwo, objectClass:rscwo,
uid:rscwo, cn:rscwo, userPassword:rswo
```

すべての属性について、Ted Morris は、Dave Miller のエントリーへのパーミッションを読み取り、検索、比較、修正、および削除するパーミッションがあります。これらの結果は、Ted Morris が自分のエントリーへのアクセスをチェックしたときに返されたものとは異なります。Ted Morris は個人的に、これらの属性のほとんどに対して読み取り、検索、比較の権利しか持っていなかったからです。

Directory Manager には、あるユーザーが別のユーザーのエントリーに対する権限をチェックする機能があります。例18.39「Directory Manager での別のユーザーに対する (User A からユーザー B に対する) 権利の確認」では、Directory Manager が、マネージャーである Jane Smith (-E) が部下である Ted Morris (-b) に対して持っている権限をチェックしています。

例18.39 Directory Manager での別のユーザーに対する (User A からユーザー B に対する) 権利の確認

```
# ldapsearch -p 389 -h server.example.com -D "cn=Directory Manager" -W -b
"uid=tmorris,ou=people,dc=example,dc=com" -E
'!1.3.6.1.4.1.42.2.27.9.5.2=:dn:uid=jsmith,ou=people,dc=example,dc=com' "(objectClass=*)"

dn: uid=tmorris,ou=People,dc=example,dc=com
... snip ...
entryLevelRights: vadn
attributeLevelRights: givenName:rscwo, sn:rscwo, ou:rscwo, l:rscwo, manager:rscwo,
roomNumber:rscwo, mail:rscwo, facsimileTelephoneNumber:rscwo, objectClass:rscwo,
uid:rscwo, cn:rscwo, userPassword:rscwo
```

管理者のみが、異なるユーザーがエントリーにある実効権限を取得することができます。Ted Morris が Dave Miller のエントリーを判定しようとすると、アクセスエラーが不十分になります。

```
# ldapsearch -p 389 -h server.example.com -D "uid=dmiller,ou=people,dc=example,dc=com" -
W -b "uid=tmorris,ou=people,dc=example,dc=com" -E
'!1.3.6.1.4.1.42.2.27.9.5.2=:dn:uid=tmorris,ou=people,dc=example,dc=com' "(objectClass=*)"

ldap_search: Insufficient access
ldap_search: additional info: get-effective-rights: requester has no g permission on the entry
```

ただし、一般ユーザーは get effective rights 検索を実行して、別のユーザーが個人エントリーに対して持っている権利を確認することができます。例18.40「個人のエントリーに対する他ユーザーの権利の確認」では、Ted Morris は、Dave Miller が Ted Morris のエントリーに対してどのような権利を持っているかを確認します。

例18.40 個人のエントリーに対する他ユーザーの権利の確認

```
# ldapsearch -p 389 -h server.example.com -D
"uid=tmorris,ou=people,dc=example,dc=com" -W -b
"uid=tmorris,ou=people,dc=example,dc=com" -E
'!1.3.6.1.4.1.42.2.27.9.5.2=:dn:uid=dmiller,ou=people,dc=example,dc=com' "(objectClass=*)"

dn: uid=tmorris,ou=people,dc=example,dc=com
... snip ...
entryLevelRights: v
attributeLevelRights: givenName:rsc, sn:rsc, ou:rsc, l:rsc,manager:rsc, roomNumber:rsc,
mail:rsc, facsimileTelephoneNumber:rsc, objectClass:rsc, uid:rsc, cn:rsc,
userPassword:none
```

この場合、Dave Miller は、エントリーの DN を閲覧する権利と、*ou*、*givenName*、*l*、およびその他の属性を読み取り、検索し、比較する権利を有しており、*userPassword* 属性については権利を有していません。

18.14.3.2. Non-Existent 属性の Get Effective Rights 検索の例

デフォルトでは、値を持たないエントリーの属性には情報は提供されません。たとえば、*userPassword* の値が削除された場合、上記のエントリーで将来的に有効な権利を検索しても、自己書き込みおよび自己削除の権利が許可されていても、*userPassword* に対する有効な権利は返されません。

get effective rights 検索でアスタリスク (*) を使用すると、エントリーに設定されていない属性も含めて、そのエントリーで利用可能なすべての属性が返されます。

例18.41 Non-Existent 属性の有効な権限を返す

```
# ldapsearch -D "cn=Directory Manager" -W -b
"uid=scarter,ou=people,dc=example,dc=com" -E
'!1.3.6.1.4.1.42.2.27.9.5.2=:dn:uid=scarter,ou=people,dc=example,dc=com' "(objectclass=*)"
"*"

dn: uid=scarter,ou=People,dc=example,dc=com
givenName: Sam
telephoneNumber: +1 408 555 4798
sn: Carter
ou: Accounting
ou: People
l: Sunnyvale
manager: uid=dmiller,ou=People,dc=example,dc=com
roomNumber: 4612
mail: scarter@example.com
facsimileTelephoneNumber: +1 408 555 9700
objectClass: top
objectClass: person
```

```
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: scarter
cn: Sam Carter
userPassword: {SSHA}Xd9Jt8g1UsHC8enNDrEmxj3iJPKQLItlDYdD9A==
entryLevelRights: vadn
attributeLevelRights: objectClass:rscwo, aci:rscwo, sn:rscwo, cn:rscwo,
description:rscwo, seeAlso:rscwo, telephoneNumber:rscwo, userPassword:rscwo,
destinationIndicator:rscwo, facsimileTelephoneNumber:rscwo,
internationaliSDNNumber:rscwo, l:rscwo, ou:rscwo, physicalDeliveryOfficeName:rscwo,
postOfficeBox:rscwo, postalAddress:rscwo, postalCode:rscwo,
preferredDeliveryMethod:rscwo, registeredAddress:rscwo, st:rscwo, street:rscwo,
teletexTerminalIdentifier:rscwo, telexNumber:rscwo, title:rscwo, x121Address:rscwo,
audio:rscwo, businessCategory:rscwo, carLicense:rscwo, departmentNumber:rscwo,
displayName:rscwo, employeeType:rscwo, employeeNumber:rscwo, givenName:rscwo,
homePhone:rscwo, homePostalAddress:rscwo, initials:rscwo, jpegPhoto:rscwo,
labeledUri:rscwo, manager:rscwo, mobile:rscwo, pager:rscwo, photo:rscwo,
preferredLanguage:rscwo, mail:rscwo, o:rscwo, roomNumber:rscwo, secretary:rscwo,
uid:rscwo,x500UniqueIdentifier:rscwo, userCertificate:rscwo, userSMIMECertificate:rscwo,
userPKCS12:rscwo
```

secretary など、エントリーに使用できる属性はすべて、その属性が存在しない場合でも表示されます。

18.14.3.3. 特定の属性またはオブジェクトクラスの Get Effective Rights 検索の例

属性関連の GER 検索を追加で取り込むと、特定の属性への権限を検索し、属性セットを検索し、エントリーに設定したオブジェクトクラスのいずれかで使用できる属性をすべて表示することができます。

「Get Effective Rights 検索の形式」のフォーマット例に記載されているオプションの 1 つは attributeList です。特定属性のみの実効権限を返すには、search コマンドの最後に、属性をスペースで区切って指定します。

例18.42 特定の属性に対する get effective rights の結果

```
# ldapsearch -D "cn=Directory Manager" -W -b
"uid=scarter,ou=people,dc=example,dc=com" -E
'!1.3.6.1.4.1.42.2.27.9.5.2=:dn:uid=scarter,ou=people,dc=example,dc=com' "(objectclass=*)"
cn mail initials

dn: uid=scarter,ou=People,dc=example,dc=com
cn: Sam Carter
mail: scarter@example.com
entryLevelRights: vadn
attributeLevelRights: cn:rscwo, mail:rscwo, initials:rscwo
```

例18.42「特定の属性に対する get effective rights の結果」の *initials* 属性のように、attributeList に存在しない属性を指定することで、アスタリスクを使用してすべての属性をリストアップするのと同様に、利用可能な権利を確認することができます。

Directory Manager は、特定のオブジェクトクラスで利用可能なすべての属性の権限を一覧表示することもできます。このオプションの形式は attribute@objectClass です。これにより、2 つのエントリーが返されます。1 つ目は指定された GER サブジェクトのエントリー、2 つ目はオブジェクトクラスのテンプレートエントリーです。

例18.43 オブジェクトクラス内の属性に対する get effective rights 結果

```
# ldapsearch -D "cn=Directory Manager" -W -b
"uid=scarter,ou=people,dc=example,dc=com" -E
'!1.3.6.1.4.1.42.2.27.9.5.2=:dn:uid=scarter,ou=people,dc=example,dc=com' "(objectclass=*)"
uidNumber@posixAccount

... snip ...

dn: cn=template_posixaccount_objectclass,uid=scarter,ou=people,dc=example,dc=com
uidnumber: (template_attribute)
entryLevelRights: v
attributeLevelRights: uidNumber:rsc
```

注記

検索形式 attribute@objectClass は、要求元 (-D) が Directory Manager の場合のみ利用できます。

特定の属性の代わりにアスタリスク (*) を使用すると、指定された GER サブジェクトのすべての属性 (present と non-existent) と、オブジェクトクラステンプレートの属性の全リストが返されます。

例18.44 オブジェクトクラスのすべての属性に対する get effective rights 結果

```
# ldapsearch -D "cn=Directory Manager" -W -b
"uid=scarter,ou=people,dc=example,dc=com" -E
'!1.3.6.1.4.1.42.2.27.9.5.2=:dn:uid=scarter,ou=people,dc=example,dc=com' "(objectclass=*)"
*@posixaccount

... snip ...

dn: cn=template_posixaccount_objectclass,uid=scarter,ou=people,dc=example,dc=com
objectClass: posixaccount
objectClass: top
homeDirectory: (template_attribute)
gidNumber: (template_attribute)
```

```
uidNumber: (template_attribute)
uid: (template_attribute)
cn: (template_attribute)
entryLevelRights: v
attributeLevelRights: cn:rsc, uid:rsc, uidNumber:rsc, gidNumber:rsc, homeDirectory:rsc,
objectClass:rsc, userPassword:none, loginShell:rsc, gecos:rsc, description:rsc, aci:rsc
```

18.14.3.4. 存在しないエントリーの get effective rights 検索の例

管理者は、既存のアクセス制御ルールに基づいて、特定のユーザー (jsmith) が存在しないユーザーにどのような権限を確認したい場合があります。存在しないエントリーをチェックする場合、サーバーはそのサブツリー内にダミーエントリーを生成します。たとえば、ダミーエントリー cn=joe new user,cn=accounts,ou=people,dc=example,dc=com を確認するには、サーバーは cn=template,cn=accounts,ou=people,dc=example,dc=com を作成します。

存在しないエントリーをチェックする場合、get effective rights 検索は、指定したオブジェクトクラスを使用して、(存在しない) エントリーのすべての属性を持つテンプレートエントリーを生成することができます。person のオブジェクトクラス (@person) を持つ cn=joe new user,cn=accounts,ou=people,dc=example,dc=com の場合、サーバーは cn=template_person_objectclass,cn=accounts,ou=people,dc=example,dc=com を生成します。

サーバーがテンプレートエントリーを作成すると、オブジェクトクラス定義の最初の MUST 属性を使用して RDN 属性を作成します (または、MUST 属性がない場合は MAY を使用します)。しかし、これは誤った RDN 値になる可能性があり、その結果、与えられたサブツリーに対して確立された ACI に違反または回避することになります。この場合は、使用する RDN 値をオブジェクトクラスを渡すことで指定できます。これには @objectclass:rdn_attribute の形式があります。

たとえば、RDN として、*uidNumber* の存在しない Posix エントリーについて scarter の権限を確認するには、次のコマンドを実行します。

```
# ldapsearch -D "cn=Directory Manager" -W -b "ou=people,dc=example,dc=com" -E
'!1.3.6.1.4.1.42.2.27.9.5.2=:dn:uid=scarter,ou=people,dc=example,dc=com' "(objectclass=*)"
@posixaccount:uidnumber

dn: uidNumber=template_posixaccount_objectclass,ou=people,dc=example,dc=com
entryLevelRights: v
attributeLevelRights: description:rsc, gecos:rsc, loginShell:rsc, userPassword
 :rsc, objectClass:rsc, homeDirectory:rsc, gidNumber:rsc, uidNumber:rsc, uid:
 rsc, cn:rsc
```

18.14.3.5. 操作属性の get effective rights 検索の例

操作属性は、get effective rights 検索など、通常の ldapsearch で返されません。操作属性の情報を返すには、プラス記号 (+) を使用します。これにより、エントリーで使用できる操作属性のみが返さ

れます。

例18.45 操作属性に対する get effective rights の結果

```
# ldapsearch -D "cn=Directory Manager" -W -x -b
"uid=scarter,ou=people,dc=example,dc=com" -E
'!1.3.6.1.4.1.42.2.27.9.5.2=:dn:uid=scarter,ou=people,dc=example,dc=com' "(objectclass=*)"
"+"

dn: uid=scarter,ou=People,dc=example,dc=com
entryLevelRights: vadn
attributeLevelRights: nsICQStatusText:rscwo, passwordGraceUserTime:rscwo,
pwdGraceUserTime:rscwo, nsYIMStatusText:rscwo, modifyTimestamp:rscwo,
passwordExpWarned:rscwo, pwdExpirationWarned:rscwo, entrydn:rscwo, aci:rscwo,
nsSizeLimit:rscwo, nsAccountLock:rscwo, passwordExpirationTime:rscwo, entryid:rscwo,
nsSchemaCSN:rscwo, nsRole:rscwo, retryCountResetTime:rscwo, ldapSchemas:rscwo,
nsAIMStatusText:rscwo, copiedFrom:rscwo, nsICQStatusGraphic:rscwo,
nsUniqueId:rscwo, creatorsName:rscwo, passwordRetryCount:rscwo, dncomp:rscwo,
nsTimeLimit:rscwo, passwordHistory:rscwo, pwdHistory:rscwo, nscpEntryDN:rscwo,
subschemaSubentry:rscwo, nsYIMStatusGraphic:rscwo, hasSubordinates:rscwo,
pwdpolicysubentry:rscwo, nsAIMStatusGraphic:rscwo, nsRoleDN:rscwo,
createTimestamp:rscwo, accountUnlockTime:rscwo, copyingFrom:rscwo,
nsLookThroughLimit:rscwo, nsds5ReplConflict:rscwo, modifiersName:rscwo,
parentid:rscwo, passwordAllowChangeTime:rscwo, nsBackendSuffix:rscwo,
nsIdleTimeout:rscwo, ldapSyntaxes:rscwo, numSubordinates:rscwo
```

18.14.3.6. get effective rights 結果とアクセスコントロールルールの例

get effective 権限は、get effective rights サブジェクトエントリーに対して有効な ACL に応じて返されます。

たとえば、この ACL が設定され、この例の目的でのみ ACL が設定されます。

```
dn: dc=example,dc=com
objectClass: top
objectClass: domain
dc: example
aci: (target=ldap:///ou=Accounting,dc=example,dc=com)(targetattr="*")(version
 3.0; acl "test acl"; allow (read,search,compare) (userdn = "ldap:///anyone") ;)

dn: ou=Accounting,dc=example,dc=com
objectClass: top
objectClass: organizationalUnit
ou: Accounting
```

ACL には dc=example,dc=com サブツリーが含まれていないため、get effective rights 検索では、ユーザーに dc=example,dc=com エントリーに対する権限がないことが表示されます。

例18.46 ACL を設定しない (Directory Manager) Get Effective Rights の結果

```
# ldapsearch -D "cn=Directory Manager" -W -b "dc=example,dc=com" -E
'!1.3.6.1.4.1.42.2.27.9.5.2=:dn:uid=scarter,ou=people,dc=example,dc=com' "(objectclass=*)"
"*@person"

dn: cn=template_person_objectclass,uid=scarter,ou=people,dc=example,dc=com
objectClass: person
objectClass: top
cn: (template_attribute)
sn: (template_attribute)
description: (template_attribute)
seeAlso: (template_attribute)
telephoneNumber: (template_attribute)
userPassword: (template_attribute)
entryLevelRights: none
attributeLevelRights: sn:none, cn:none, objectClass:none, description:none,
seeAlso:none, telephoneNumber:none, userPassword:none, aci:none
```

Directory Manager ではなく通常のユーザーが同じコマンドを実行しようとすると、結果は単に空白になります。

例18.47 ACL を設定しない (通常ユーザー) Get Effective Rights の結果

```
# ldapsearch -D "uid=scarter,ou=people,dc=example,dc=com" -W -b "dc=example,dc=com"
-E '!1.3.6.1.4.1.42.2.27.9.5.2=:dn:uid=scarter,ou=people,dc=example,dc=com' "
(objectclass=*)" "*@person"
```

18.14.4. コンソールからの Get Effective Rights の使用

1. Directory タブを開き、権限を確認するエントリーを右クリックします。

2. ドロップダウンメニューから Advanced Properties を選択します。

3. Show effective rights チェックボックスにチェックを入れます。

4.

属性ごとに、属性レベルの get effective rights が表示されます。エントリーレベルの権限は、エントリーの DN の下に表示されます。

属性レベルの有効な権限（r、s、c、w、o）が属性の横に表示されます。エントリーレベルの権限（v、 a、d、n）は、Property Editor の左下にあるエントリーの完全な DN に表示されます。

Show all allowed attributes チェックボックスにチェックマークを入れると、値がなくてもこれらの属性の有効な権限が追加の属性の横に表示されます。

18.14.5. Get Effective Rights 戻りコード

Get Effective Rights 検索とエラーに重大度が設定されていない場合は、通常のエントリー情報が返されますが、entryLevelRights および attributeLevelRights の権限の代わりに、エラーコードが返されます。このコードにより、クエリーされたエントリーの設定に関する情報を取得できます。表 18.4「返された結果コード」は、エラーコードと、リレーが可能な潜在的な設定情報を要約します。

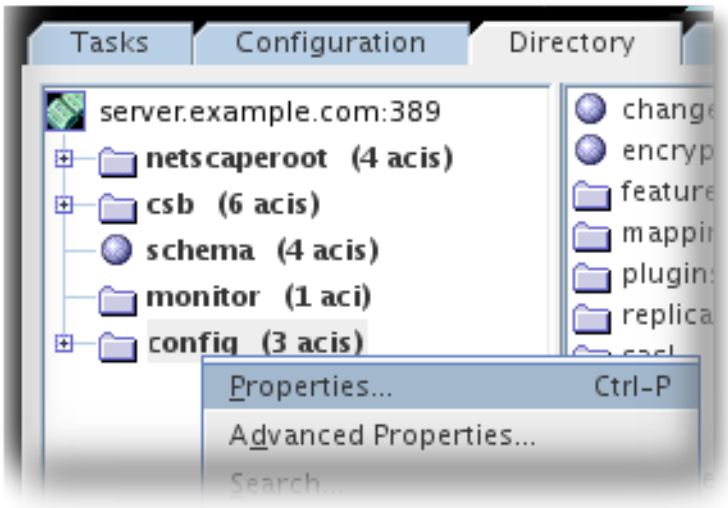
表18.4 返された結果コード

コード	説明
0	正常に完了しました。
1	操作エラー。
12	重要な拡張機能は利用できません。重大度式が true に設定され、クエリー対象のエントリーに有効な権限がない場合は、このエラーが返されます。
16	そのような属性はありません。アクセス権のために特定の属性をクエリーしましたが、その属性がスキーマに存在しない場合は、このエラーが返されます。
17	未定義の属性タイプ。
21	無効な属性構文。
50	権限が不十分。
52	利用できません。
53	不本意なパフォーマンス。
80	その他。

18.15. アクセス制御情報のロギング

エラーログでアクセス制御に関する情報を取得するには、適切なログレベルを設定する必要があります。コンソールからエラーログレベルを設定するには、以下を実行します。

1. Console で Directory タブをクリックし、設定ノードを右クリックし、ポップアップメニューから Properties を選択します。



これにより、cn=config エントリーの Property Editor が表示されます。

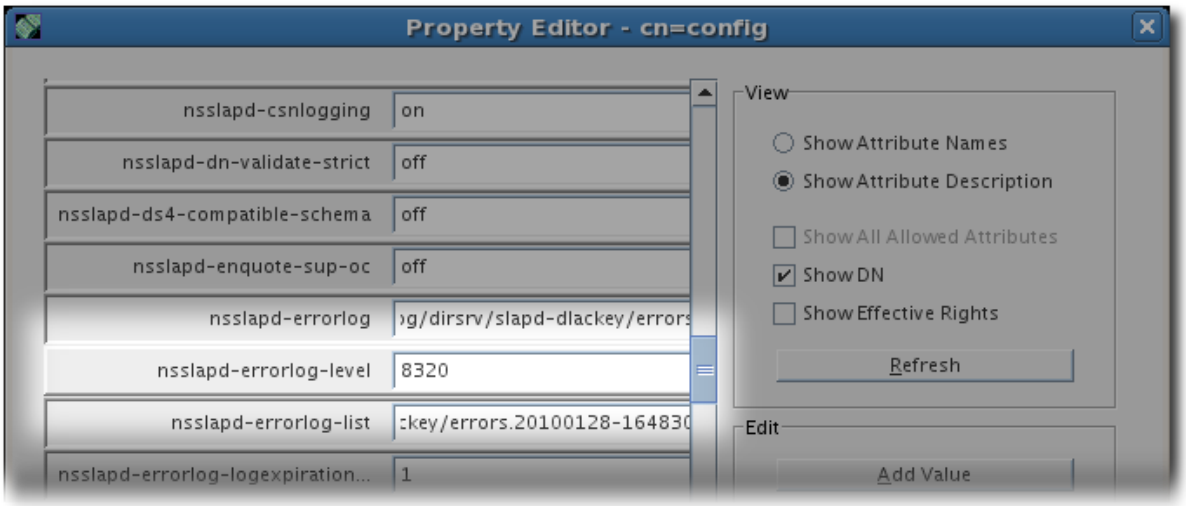
2.

属性値のペアのリストを下方向にスクロールして、nsslapd-errorlog-level 属性を見つけます。

3.

nsslapd-errorlog-level value フィールドに、現在表示されている値に 128 を追加します。

たとえば、すでに表示される値が 8192（レプリケーションのデバッグ）の場合は、値を 8320 に変更します。エラーログレベルの詳細は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス を参照してください』。



4.

OK をクリックして、Property Editor を外します。

18.16. 高度なアクセス制御: マクロ ACI の使用

ディレクトリーツリー構造の繰り返しを使用する組織では、マクロを使用してディレクトリーで使用される ACI の数を最適化できます。ディレクトリーツリーの ACI の数を減らすと、アクセス制御ポリシーの管理が容易になり、ACI メモリー使用量の効率が向上します。

マクロは、ACI の DN または DN の一部を表すために使用されるプレースホルダーです。マクロを使って、ACI のターゲット部分、バインドルール部分、またはその両方で DN を表現することができます。実際には、Directory Server が受信 LDAP 操作を取得すると、ACI マクロは LDAP 操作によってターゲットとなっているリソースと照合されます。一致する場合、マクロはターゲットリソースの DN の値に置き換えられます。次に、Directory Server は ACI を正常に評価します。

18.16.1. マクロ ACI の例

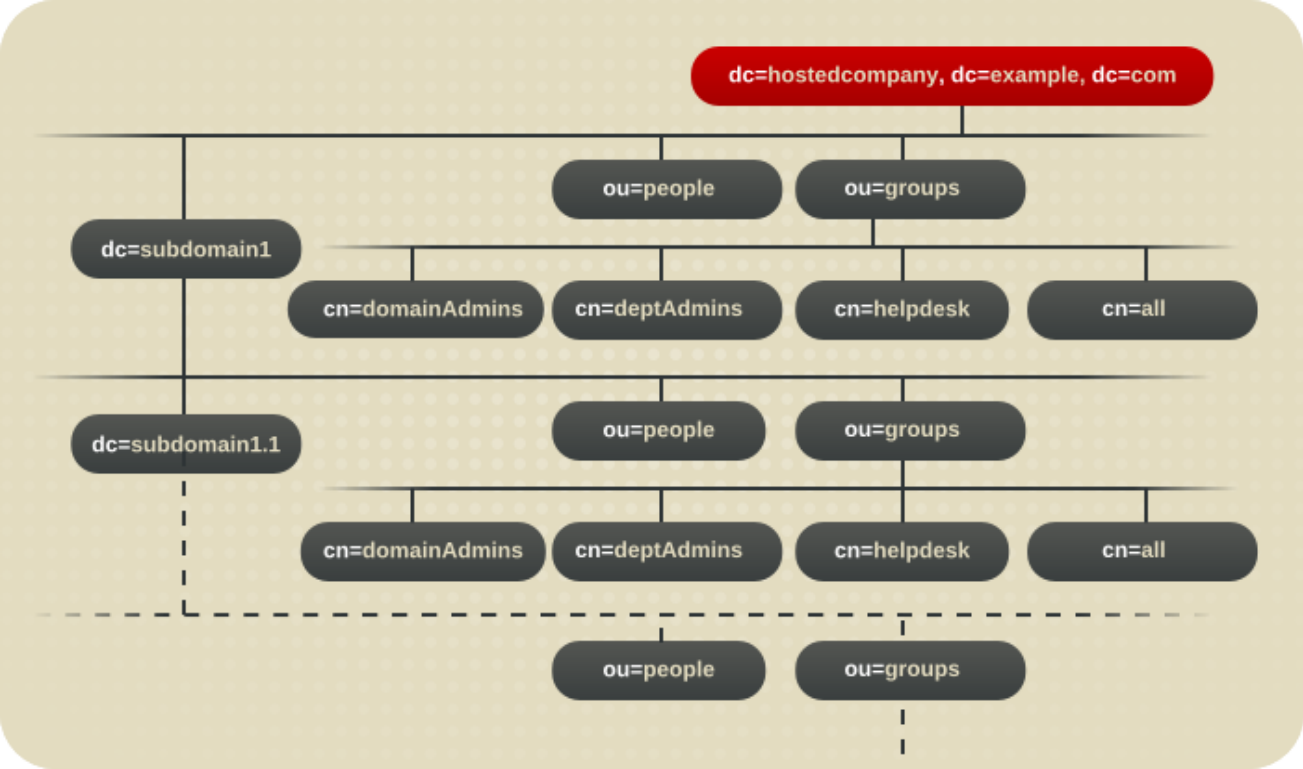
図18.1「マクロ ACI のディレクトリーツリーの例」は、マクロ ACI を使用して ACI の全体的な数を効果的に減らすディレクトリーツリーを表示します。この図は、同じツリー構造 (ou=groups、ou=people) を持つサブドメインの繰り返しパターンを使用します。Example Corp. ディレクトリーツリーが接尾辞 dc=hostedCompany2,dc=example,dc=com および dc=hostedCompany3,dc=example,dc=com を格納するため、このパターンはツリー全体で繰り返されます。

ディレクトリーツリーに適用される ACI には、繰り返しパターンがあります。たとえば、以下の ACI は dc=hostedCompany1,dc=example,dc=com ノードにあります。

```
aci: (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
    (version 3.0; acl "Domain access"; allow (read,search)

groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany1,dc=example,dc=com";)
```

この ACI は、dc=hostedCompany1,dc=example,dc=com ツリー内の任意のエントリーに DomainAdmins グループに対する読み取り権限および検索権限を付与します。

図18.1 マクロ ACI のディレクトリーツリーの例



以下の ACI は dc=hostedCompany1,dc=example,dc=com ノードにあります。

```
aci: (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
   (version 3.0; acl "Domain access"; allow (read,search)

groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany1,dc=example,dc=com";)
```

以下の ACI は dc=subdomain1,dc=hostedCompany1,dc=example,dc=com ノードにあります。

```
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=ex
ample,dc=com";)
```

以下の ACI は dc=hostedCompany2,dc=example,dc=com ノードにあります。

```
aci: (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
    (version 3.0; acl "Domain access"; allow (read,search)

groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany2,dc=example,dc=com";)
```

以下の ACI は dc=subdomain1,dc=hostedCompany2,dc=example,dc=com ノードにあります。

```
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=ex
ample,dc=com";)
```

上記の 4 つの ACI での唯一の違いは groupdn キーワードで指定される DN です。DN にマクロを使用して、これらの ACI を dc=example,dc=com ノード上のツリーのルートにある単一の ACI に置き換えることができます。この ACI は以下のように読み取ります。

```
aci: (target="ldap:///ou=Groups,($dn),dc=example,dc=com")
    (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
    (version 3.0; acl "Domain access"; allow (read,search)
    groupdn="ldap:///cn=DomainAdmins,ou=Groups,[$dn],dc=example,dc=com";)
```

以前使用されていなかった target キーワードが新しい ACI で使用されます。

この例では、ACI の数が 4 から 1 に減ります。実際の効果は、ディレクトリツリーにどれだけ多くの繰り返しパターンがあるかにかかっています。

18.16.2. マクロ ACI 構文

マクロ ACI には、DN、または DN の一部を置き換える以下のタイプの式が含まれます。

- ($dn)
- [$dn]
- ($attr.attrName) ここで、attrName はターゲットエントリーに含まれる属性を表します。

このセクションでは、userdn、roledn、groupdn、および userattr などのバインド認証情報を提供するために使用される ACI キーワードは、ACI の ターゲット とは異なり、サブジェクト をまとめて呼びます。マクロ ACI は、ターゲット部分または ACI のサブジェクト部分で使用できます。

表18.5「ACI キーワードのマクロ」は、DN マクロを使用できる ACI の概要を示します。

表18.5 ACI キーワードのマクロ

マクロ	ACI キーキーワード
($dn)	target, targetfilter, userdn, roledn, groupdn, userattr
[$dn]	targetfilter, userdn, roledn, groupdn, userattr
($attr.attrName)	userdn, roledn, groupdn, userattr

以下の制限が適用されます。

- targetfilter、userdn、roledn、groupdn、userattr で ($dn) を使用する場合は、($dn) を含むターゲットを定義する 必要があります。
- targetfilter、userdn、roledn、groupdn、userattr で [$dn] を使用する場合は、($dn) を含むターゲットを定義する 必要があります。

注記

マクロを使用する場合は、($dn) マクロが含まれるターゲット定義を 常に 必要とします。

($dn) マクロと ($attr.attrName) マクロを組み合わせることができます。

18.16.2.1. ($dn) のマクロ一致

($dn) マクロは、LDAP 要求でターゲットとするリソースの一致する部分に置き換えられます。たとえば、cn=all,ou=groups,dc=subdomain1,dc=hostedCompany1,dc=example,dc=com エントリーで LDAP 要求と、以下のようにターゲットを定義する ACI があります。

```
(target="ldap:///ou=Groups,($dn),dc=example,dc=com")
```

($dn) マクロは dc=subdomain1,dc=hostedCompany1 と一致します。

ACI のサブジェクトも ($dn) を使用する場合は、ターゲットと一致する部分文字列を使用してサブジェクトを展開します。以下に例を示します。

```
aci: (target="ldap:///ou=*,($dn),dc=example,dc=com")
    (targetattr = "*") (version 3.0; acl "Domain access"; allow (read,search)
    groupdn="ldap:///cn=DomainAdmins,ou=Groups,($dn),dc=example,dc=com";)
```

この場合、ターゲットの ($dn) に一致する文字列が dc=subdomain1,dc=hostedCompany1 の場合、サブジェクトで同じ文字列が使用されます。その後、ACI は以下のように拡張されます。

```
aci: (target="ldap:///ou=Groups,dc=subdomain1,dc=hostedCompany1,
    dc=example,dc=com") (targetattr = "*") (version 3.0; acl "Domain
    access"; allow (read,search) groupdn="ldap:///cn=DomainAdmins,ou=Groups,
    dc=subdomain1,dc=hostedCompany1,dc=example,dc=com";)
```

マクロの拡張後、Directory Server は通常のプロセスに従って ACI を評価し、アクセスが付与されているかどうかを確認します。

18.16.2.2. [$dn] のマクロ一致

[$dn] と一致するメカニズムは、($dn) とは若干異なります。対象となるリソースの DN は、一致するものが見つかるまで、左端の RDN コンポーネントを削除するたびに複数回調べられます。

たとえば、cn=all,ou=groups,dc=subdomain1,dc=hostedCompany1,dc=example,dc=com サブツリーをターゲットとする LDAP 要求と、以下の ACI があります。

```
aci: (target="ldap:///ou=Groups,($dn),dc=example,dc=com")
    (targetattr = "*") (version 3.0; acl "Domain access"; allow (read,search)
    groupdn="ldap:///cn=DomainAdmins,ou=Groups,[$dn],dc=example,dc=com";)
```

この ACI を拡張する手順は以下のとおりです。

1. ターゲットの ($dn) は dc=subdomain1,dc=hostedCompany1 と一致します。
2. サブジェクトの [$dn] は、dc=subdomain1,dc=hostedCompany1 に置き換えられます。

   その結果は、groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=subdomain1,dc=hostedCompany1,dc=example,dc=com" になります。バインド DN が対象のグループのメンバーである場合は、一致するプロセスが停止し、ACI が評価されます。一致しない場合は、プロセスが続行します。

3.
サブジェクトの [$dn] は、dc=hostedCompany1 に置き換えられます。

その結果は、groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany1,dc=example,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はない場合、ACI は評価されません。これがメンバーの場合には、ACI が評価されます。

[$dn] は、マクロの利点は、ドメインレベルの管理者にディレクトリーツリー内の すべての サブドメインにアクセスを付与する柔軟な方法を提供することです。したがって、ドメイン間の階層関係を表現するのに便利です。

たとえば、以下の ACI について考えてみましょう。

```
aci: (target="ldap:///ou=*, ($dn),dc=example,dc=com")
    (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
    (version 3.0; acl "Domain access"; allow (read,search)
    groupdn="ldap:///cn=DomainAdmins,ou=Groups,[$dn],dc=example,dc=com";)
```

cn=DomainAdmins,ou=Groups,dc=hostedCompany1,dc=example,dc=com のメンバーに、dc=hostedCompany1 の下にあるすべてのサブドメインへのアクセス権限を付与します。そのため、該当グループに属する管理者は ou=people,dc=subdomain1.1,dc=subdomain1 などのサブツリーにアクセスできます。

ただし、同時に cn=DomainAdmins,ou=Groups,dc=subdomain1.1 のメンバーは、ou=people,dc=hostedCompany1 および ou=people,dc=hostedCompany1 ノードへのアクセスが拒否されます。

18.16.2.3. Macro Matching for ($attr.attrName)

($attr.attrName) マクロは、DN のサブジェクト部分に常に使用されます。たとえば、以下の roledn を定義します。

```
roledn = "ldap:///cn=DomainAdmins,($attr.ou)"
```

ここで、サーバーが、以下のエントリーでターゲットに設定された LDAP 操作を受け取ることを想定しています。

```
dn: cn=Jane Doe,ou=People,dc=HostedCompany1,dc=example,dc=com
cn: Jane Doe
```

```
sn: Doe
ou: Engineering,dc=HostedCompany1,dc=example,dc=com
...
```

ACI の roledn 部分を評価するため、サーバーはターゲットエントリーに保存されている *ou* 属性を確認し、この属性の値をマクロを展開します。そのため、この例では roledn は以下のように展開されます。

```
roledn =
"ldap:///cn=DomainAdmins,ou=Engineering,dc=HostedCompany1,dc=example,dc=com"
```

次に、Directory Server は通常の ACI 評価アルゴリズムに従って ACI を評価します。

属性が多値を持つ場合、それぞれの値を使ってマクロを展開し、一致した最初の値を使用します。以下に例を示します。

```
dn: cn=Jane Doe,ou=People,dc=HostedCompany1,dc=example,dc=com
cn: Jane Doe
sn: Doe
ou: Engineering,dc=HostedCompany1,dc=example,dc=com
ou: People,dc=HostedCompany1,dc=example,dc=com...
```

この場合、Directory Server が ACI を評価すると、以下の拡張された式で論理 OR が実行します。

```
roledn =
"ldap:///cn=DomainAdmins,ou=Engineering,dc=HostedCompany1,dc=example,dc=com"

roledn = "ldap:///cn=DomainAdmins,ou=People,dc=HostedCompany1,dc=example,dc=com"
```

18.17. DIRECTORY MANAGER でのアクセス制御の設定

未制約の管理ユーザーがあると、メンテナンスパースペクティブからは妥当になります。Directory Manager では、メンテナンスタスクを実行し、インシデントへの対応に高いレベルのアクセスが必要です。

ただし、Directory Manager ユーザーの権限により、ある程度のアクセス制御は、root ユーザーとして、承認されていないアクセスや攻撃を阻止することをお勧めします。

通常のアクセス制御ルールはディレクトリーツリーに適用されます。Directory Manager は通常のユーザーエントリーではないため、(通常の) ACI を Directory Manager ユーザーに適用できません。

ACI は、特別なプラグイン設定エントリーを介して適用されます。

18.17.1. Directory Manager アカウントのアクセス制御

通常のアクセス制御ルールは、Directory Manager ユーザーには適用されません。Directory Manager は、通常のユーザーデータベースではなく、dse.ldif ファイルで定義されます。したがって、サブツリー内のエントリーに基づく ACI ターゲット(「ターゲットの定義」)には Directory Manager は含まれません。

Directory Manager のアクセス制御は、RootDN アクセス制御プラグイン を介して実装されます。このプラグインは Directory Server 設定に適用されるため、Directory Manager エントリーにはアクセス制御ルールを適用できます。

プラグインは標準の ACL を定義しません。これには、ターゲット (Directory Manager エントリー) および許可される権限 (すべて) などの一部の情報が暗に示されています。RootDN アクセス制御プラグインの目的は、Directory Manager が実行可能な もの を制限しません。そのため、場所または時間をもとに、(有効な認証情報でも) Directory Manager としてログインできるユーザーを制限することで、セキュリティーレベルを提供することが目的です。

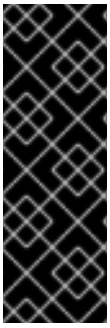
このため、Directory Manager の ACI はバインドルールのみを設定します。

- 時間ベースのアクセス制御 (例: 8a.m. から 5p.m)(0800 から 1700)、および曜日のアクセス制御により、アクセスは明示的に定義された日数でのみ許可されます。これは「曜日の特定の日におけるアクセスの定義」および「特定の時刻におけるアクセスの定義」に類似しています。

- 指定した IP アドレス、ドメイン、またはサブネットのみが明示的に許可または拒否される IP アドレスルール。これは、「特定の IP アドレスまたは範囲からのアクセスの定義」に類似しています。

- ホストアクセスルール。指定されたホスト名、ドメイン名、またはサブドメインのみが明示的に許可または拒否されます。これは、「特定のホストまたはドメインからアクセスの定義」に類似しています。

他のアクセス制御ルールと同様、deny ルールは allow ルールよりも優先されます。



重要

Directory Manager が常に適切なレベルのアクセスを許可されていることを確認してください。Directory Manager は、オフタイム (ユーザーの負荷が少ない時) や障害対応のためにメンテナンス作業を行う必要があります。その場合、時間や曜日ベースのアクセス制御ルールを厳しく設定すると、Directory Manager がディレクトリーを適切に管理できなくなる可能性があります。

18.17.2. RootDN アクセス制御プラグインの設定

ルート DN アクセス制御ルールはデフォルトで無効になっています。rootDN アクセス制御プラグインを有効にし、次に適切なアクセス制御ルールを設定できます。

注記

Directory Manager には 1 つのアクセス制御ルールがあり、プラグインエントリーには、ディレクトリー全体のすべてのアクセスに適用されます。

1. *nsslapd-pluginEnabled* 属性を on に設定して、RootDN アクセス制御プラグインを有効にします。以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=RootDN Access Control Plug-in,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

2. アクセス制御命令にバインドルールを設定します。

- *rootdn-open-time* 時間ベースのアクセス制御の場合は *rootdn-close-time* です。
- *rootdn-days-allowed* 日ベースのアクセス制御の場合
- *rootdn-allow-host*ホストベースのアクセス制御用の *rootdn-deny-host*、*rootdn-allow-ip*、および *rootdn-deny-ip*。これらはすべて多値の属性です。

拒否ルールは、許可ルールよりも優先されます。たとえば、*rootdn-allow-host* 属性

が *.example.com に設定され、*rootdn-deny-host* 属性が *.front-office.example.com に設定されている場合、front-office.example.com サブドメインにあるものはすべて、大規模な example.com ドメインが許可されていても Directory Manager としてログインできなくなります。

ワイルドカードは、IP 範囲またはフルドメインを許可するために使用できます。

以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=RootDN Access Control Plug-in,cn=plugins,cn=config
changetype: modify
add: rootdn-open-time
rootdn-open-time: 0600
-
add: rootdn-close-time
rootdn-close-time: 2100
-
add: rootdn-allow-host
rootdn-allow-host: *.example.com
-
add: rootdn-deny-host
rootdn-allow-host: *.remote.example.com
```

3. Directory Server を再起動して、新しいプラグイン設定を読み込みます。

```
# systemctl restart dirsrv@instance
```

18.18. 以前のリリースとの互換性

後方互換性のために、Directory Server で非推奨となった以下の ACI キーワードがサポートされます。

- userdnattr
- groupdnattr

注記

Red Hat は、これらの非推奨の ACI キーワードを使用しないことを推奨します。これらのキーワードは、Directory Server の今後のリリースで削除される予定です。

第19章 ユーザー認証の管理

ユーザーが Red Hat Directory Server に接続すると、最初にユーザーが認証されます。次に、ディレクトリーは認証時に確立されたアイデンティティーに応じてアクセス権限およびリソース制限をユーザーに付与します。

この章では、ディレクトリーのパスワードとアカウントのロックアウトポリシーの設定、ディレクトリーへのアクセスを拒否するユーザーグループの設定、バインド DN に応じてユーザーが利用できるシステムリソースの制限など、ユーザーを管理するためのタスクを説明します。

19.1. ユーザーパスワードの設定

このエントリーは、*userPassword* 属性があり、アクティブではない場合にのみディレクトリーにバインドするために使用できます。ユーザーパスワードはディレクトリーに格納されるため、ldapmodify などの LDAP 操作でユーザーパスワードを設定またはリセットできます。

ディレクトリーエントリーの作成または変更に関する情報は、「3章 *ディレクトリーエントリーの管理*」を参照してください。ユーザーアカウントを非アクティブにする方法は、「ユーザーおよびロールの手動による非アクティブ化」を参照してください。

Red Hat 管理コンソールまたは Directory Server コンソールの Users and Groups 領域でパスワードを設定してリセットすることもできます。管理コンソールの Users および Groups エリアの使用方法は、Red Hat 管理コンソールで利用可能なオンラインヘルプを参照してください。

「パスワード管理者の設定」に説明されているパスワード管理者のみが、事前にハッシュ化されたパスワードを追加できます。これらのユーザーは、パスワードポリシーに違反させることもできます。

警告

パスワード管理者アカウントまたは Directory Manager (root DN) を使用してパスワードを設定すると、パスワードポリシーは回避され、検証されません。通常のユーザーパスワードの管理には、これらのアカウントを使用しないでください。パスワードポリシーの回避が必要なパスワード管理タスクの実行にのみ使用します。

19.2. パスワード管理者の設定

Directory Manager は、*パスワード管理者* ロールをユーザーまたはグループに追加できます。アクセス制御命令 (ACI) は設定する必要があるため、グループで、すべてのパスワード管理者を管理する単一の ACI セットのみを許可することが推奨されます。パスワード管理者は、以下を含むユーザーパスワード操作を実行できます。

- ユーザーがパスワードの変更を強制する
- パスワードポリシーで定義されている異なるストレージスキームへのユーザーのパスワードの変更
- パスワード構文チェックを回避
- ハッシュ化されたパスワードを追加します。

「ユーザーパスワードの設定」で説明されているように、通常のパスワードの更新は、userPassword 属性のみを更新するパーミッションを持つデータベース内の既存のロールで実行することが推奨されます。このような通常のタスクにパスワード管理者アカウントを使用することは推奨されません。

ローカルポリシーのパスワード管理者としてユーザーまたはグループを指定するには、ldapmodify を使用してメインの設定エントリーに passwordAdminDN 属性を設定します。

```
# ldapmodify -h localhost -p 389 -D "cn=Directory Manager" -W
dn:
cn=cn\3DnsPwPolicyEntry\2Cou\3DPeople\2Cdc\3Dexample\2Cdc\3Dcom,cn=nsPwPolicyContainer,ou=
People,dc=example,dc=com
changetype: modify
replace: passwordAdminDN
passwordAdminDN: cn=Passwd Admins,ou=groups,dc=example,dc=com
```

グローバルポリシーで設定する場合：

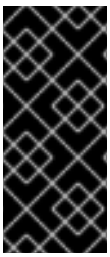
```
# ldapmodify  -h localhost -p 389 -D "cn=Directory Manager" -W
dn: cn=config
changetype: modify
replace: passwordAdminDN
passwordAdminDN: cn=Passwd Admins,ou=groups,dc=example,dc=com
```

19.3. 外部に保存されたパスワードの変更

ほとんどのパスワードは、コンソールおよびその他の Directory Server 機能や ldapmodify 操作から変更できますが、通常の LDAP 操作で変更することはできないパスワードがあります。これらのパスワードは、SASL アプリケーションに保存されているパスワードなど、Directory Server 外に保存できます。これらのパスワードは、パスワード変更拡張操作 で変更できます。

Directory Server は RFC 3062 で定義されているパスワード変更操作をサポートするため、ユーザーは標準に準拠した状態で適切なクライアントを使用してパスワードを変更できます。ldappasswd ユーティリティーは、指定されたユーザーのパスワードの変更を渡します。

```
# ldappasswd -x -D bind_dn -W -p server_port -h server_hostname [-a oldPassword] [-s newPassword] [user]
```



重要

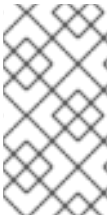
パスワード操作はセキュアな接続（SASL、TLS、または Start TLS）に対して実行する必要があります。LDAP クライアントツールでセキュアな接続を使用する方法は、「証明書を使用した認証」を参照してください。

表19.1 ldappasswd オプション

パラメーター	詳細
-h	Directory Server のホスト名を指定します。
-p	Directory Server のポート番号を指定します。パスワード変更操作に TLS が必要なため、通常は Directory Server の TLS ポートが指定されます。Start TLS の -Z Z または -ZZZ を使用する場合、これは標準ポートになります。
-D	バインド DN を指定します。
-w	バインド DN のパスワードを指定します。
-x	TLS 接続での単純なバインドを許可するように SASL を無効にします。
-a	オプション。変更中の古いパスワードを指定します。
-s	オプション。新しいパスワードを設定します。
user	オプション。パスワードを変更するユーザーエントリーの DN を指定します。

セキュアでないポート上でコマンドを実行する TLS を使用するには、-ZZ オプションと標準の LDAP ポート番号を指定して ldappasswd を実行します。パスワード拡張変更操作の形式は以下のとおりです。

```
# ldappasswd -x -D bind_dn -W -p server_port -h server_hostname -Z [-a oldPassword] [-s newPassword] [user]
```



注記

TLS 接続を機能させるには、「証明書を使用した認証」 の説明に従って TLS 環境変数を設定する必要があります。

-ZZ オプションを使用して、強制的に接続を成功させます。

エントリーのパスワードを変更するには、他の LDAP 操作と同様に ldappasswd を実行します。アカウントがバインド DN で指定されるのと同じ場合は、ユーザー を指定する必要はありません。以下に例を示します。

```
# ldappasswd -x -h ldap.example.com -p 389 -ZZ -D "uid=jsmith,ou=People,dc=example,dc=com" -W -s newpassword
```

バインド認証情報で指定されたエントリー以外のエントリーのパスワードを変更するには、以下のように ldappasswd を実行します。以下のように、ユーザー DN を操作に追加し、別の認証情報を提供します。

```
# ldappasswd -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -ZZ -s newpassword "uid=jsmith,ou=People,dc=example,dc=com"
```

アクセス制御はパスワードの変更操作に対して適用されます。バインド DN に指定のパスワードを変更する権限がない場合、操作は Insufficient rights エラーを出力して失敗します。

19.4. パスワードポリシーの管理

パスワードポリシーは、一定レベルのセキュリティーを強制することで、パスワードを使用するリスクを最小限に抑えることができます。たとえば、パスワードポリシーでは、以下を定義できます。

- ユーザーはスケジュールに応じてパスワードを変更する必要があります。

- ユーザーは、簡単ではないパスワードを提供する必要があります。

- パスワード構文は、特定の複雑な要件を満たす必要があります。

パスワードポリシーの概要は、『デプロイメントガイド』の「セキュアディレクトリーの設計」の章の「パスワードポリシーの設計」『を参照してください』。

警告

パスワード管理者アカウントまたは Directory Manager (root DN) を使用してパスワードを設定すると、パスワードポリシーは回避され、検証されません。通常のユーザーパスワードの管理には、これらのアカウントを使用しないでください。パスワードポリシーの回避が必要なパスワード管理タスクの実行にのみ使用します。

Directory Server は、きめ細かなパスワードポリシーをサポートしており、パスワードポリシーは、ディレクトリー全体 (global パスワードポリシー)、特定のサブツリー (subtree-level または local パスワードポリシー)、または特定のユーザー (user-level または local パスワードポリシー) に適用することができます。

ユーザーアカウントに適用される完全なパスワードポリシーは、以下の要素で構成されます。

- パスワードポリシーチェックのタイプまたはレベル。この情報は、サーバーがグローバルパスワードポリシーまたはローカル (サブツリー/ユーザーレベル) パスワードポリシーを確認および有効にするかどうかを示します。

  パスワードポリシーは、一般的なものから特定のものまで、逆ピラミッド型になっています。グローバルパスワードポリシーは、サブツリーレベルのパスワードポリシーに置き換えられました。これは、ユーザーレベルのパスワードポリシーに置き換えられます。エントリーに対して強制されるパスワードポリシーは 1 つだけで、パスワードポリシーは追加されません。つまり、特定の属性がグローバルレベルまたはサブツリーレベルのポリシーで設定されていて、ユーザーレベルのパスワードポリシーで設定されていない場合、アクティブで適用される

ポリシーはユーザーレベルのポリシーであるため、ログインが試みられてもその属性はユーザーに使用されません。

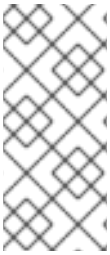
- パスワードの追加および変更の情報。パスワード情報には、パスワードの構文およびパスワード履歴の詳細が含まれます。

- バインド情報バインド情報には、許可された猶予期間の数、パスワードエージング属性、およびバインド失敗の追跡が含まれます。

注記

パスワードポリシーを設定したら、アカウントのロックアウトポリシーを設定すると、ユーザーパスワードを潜在的な脅威から保護できます。アカウントのロックアウトは、ユーザーのパスワードを繰り返し推測することで、ディレクトリーに分割を試行するハッカーから保護されます。

19.4.1. グローバルパスワードポリシーの設定



注記

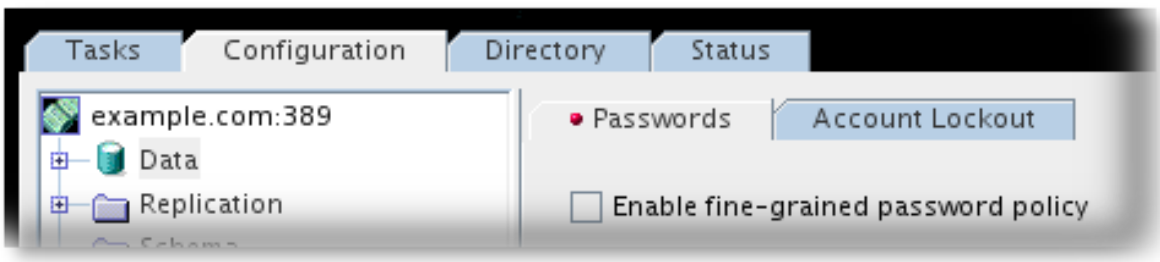
パスワードポリシーを設定したら、アカウントロックアウトポリシーを設定します。詳細は、「パスワードベースのアカウントロックアウトポリシーの設定」を参照してください。

19.4.1.1. コンソールを使用したグローバルパスワードポリシーの設定

グローバルパスワードポリシーは、ディレクトリー全体のすべてのエントリーに適用されます。

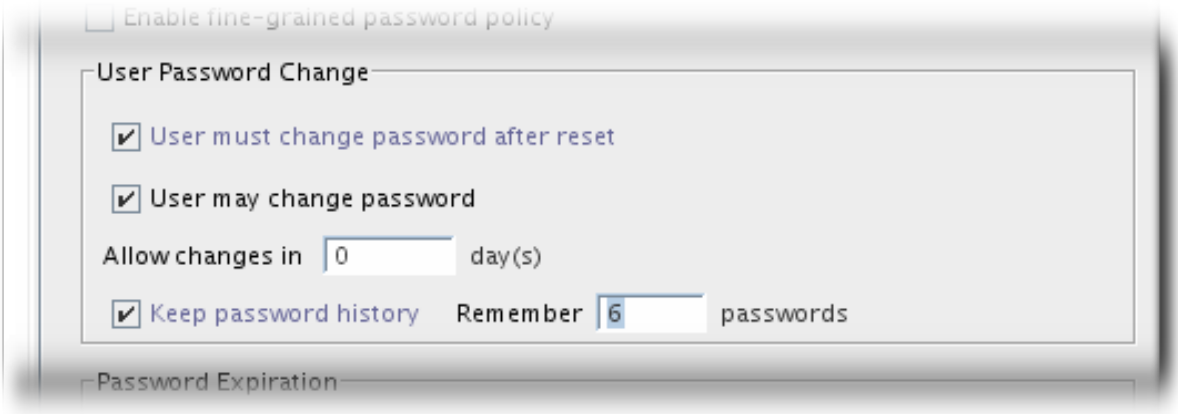
1. **Configuration** タブを選択し、**Data** ノードを選択します。

2. 右側のペインで、**Passwords** タブを選択します。



このタブには、Directory Server 全体のパスワードポリシーが含まれます。

3. ユーザーが独自のパスワードを変更するためにパスワードポリシーを設定します。



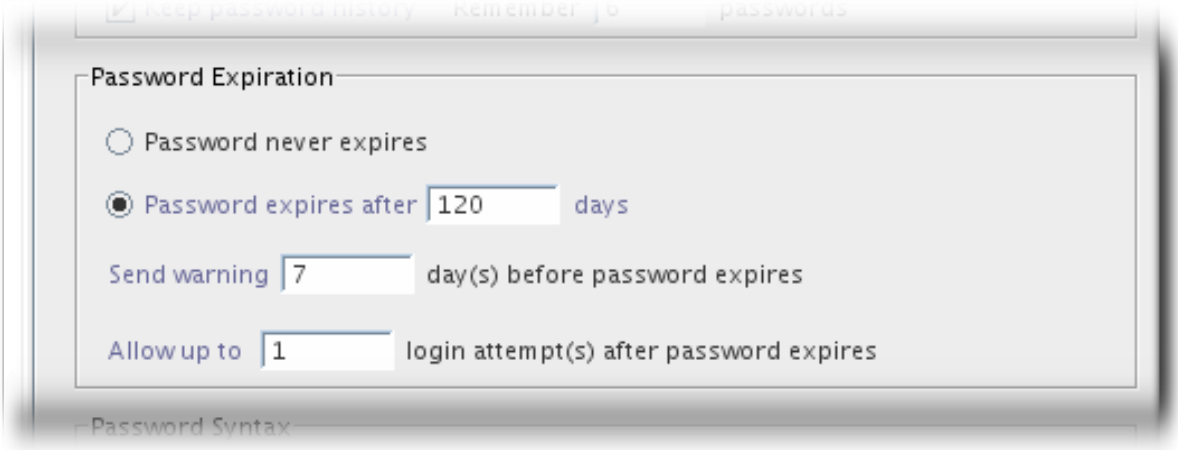
- ユーザーが初めてパスワードを変更する必要がある場合は、リセット後にパスワードを変更する必要があります。

- ユーザーが独自のパスワードを変更できるようにするには、User may change password チェックボックスを選択します。

- ユーザーが特定の期間のパスワードを変更できないようにするには、Allow changes in X day(s) テキストボックスに日数を入力します。これにより、パスワード履歴でパスワードを再利用できるように、パスワード交換を迅速化する必要がなくなります。

- サーバーが各ユーザーが使用するパスワードの履歴リストを維持するには、Keep password history チェックボックスを選択します。Remember X password テキストボックスに、ユーザーごとに保持するサーバーのパスワード 数を入力します。

4. パスワードの有効期限が切れる場合のポリシーを設定します。



- ユーザーパスワードが失効しない場合は、パスワードを失効 しない ラジオボタンを選択します。

- ユーザーがパスワードを定期的に変更する必要がある場合は、X days ラジオボタンの後に Password expires を選択し、ユーザーパスワードが有効な日数を入力します。

  パスワード期間の最大値は、今日の日付から 1 月 18 日の 2038 を引いて派生します。入力した値は最大値に設定できず、最大値に近い値も設定しないでください。値を最大値に設定すると、エポック日が経過した秒数を超えると、Directory Server が起動できなくなる可能性があります。このような場合、エラーログはパスワードの最大期間が無効であることを示します。この問題を解決するには、dse.ldif ファイルの *passwordMaxAge* 属性値を修正します。

  一般的なポリシーでは、パスワードは 30 - 90 日ごとに失効します。デフォルトでは、パスワードの最大期間は 8640000 秒（100 日）に設定されます。

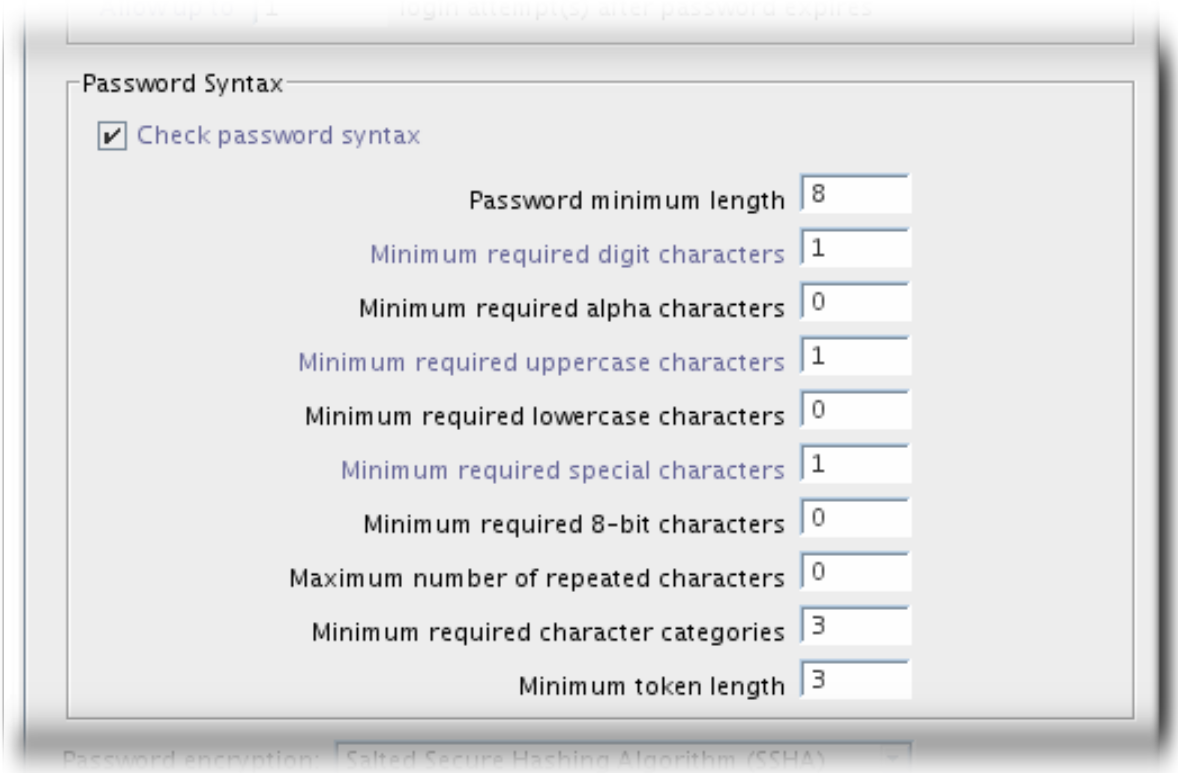
- X days ラジオボタンが終わった後にパスワードを失効 させた場合は、パスワードが失効してからユーザーに警告を送信する期間を指定します。Send Warning X Days Before Password Expires テキストでは、パスワードの有効期限が失効してから警告を送信するまでの日数を入力します。

注記

Directory Server がユーザーに警告を送信するように設定する必要はありません。Directory Server は、次回ユーザーが Directory Server Console にログインしようとすると、パスワードが期限切れになったり、期限切れになったりするときに警告が表示されます。これは、ユーザーのログイン時に "Warning: password will expire in 7 days" を読み取るオペレーティングシステムの警告に似ています。

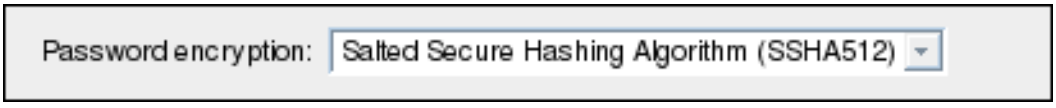
5.

サーバーによるユーザーパスワードの構文をチェックして、パスワードポリシーで設定された最小要件を満たすようにするには、**Check Password Syntax** チェックボックスを選択します。次に、最小の長さや必要な数や特殊文字など、必要なパスワードの複雑性を指定します。



6.

パスワード暗号化 プルダウンメニューから、パスワードを保存する時に使用するサーバーの暗号化方法を選択します。



サポート対象のパスワードストレージスキームの一覧は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス の該当するセクションを参照してください』。

7.

Save をクリックします。

19.4.1.2. コマンドラインを使用したグローバルパスワードポリシーの設定

サブツリーまたはユーザーにパスワードポリシーを設定するには、サブツリーまたはユーザーレベルで必要なエントリーおよび属性を追加し、適切な値をパスワードポリシー属性に設定し、詳細なパスワードポリシーチェックを有効にします。

パスワードポリシー属性はデフォルトで設定されていません。グローバルポリシーを作成するには、各パスワードポリシー属性を cn=config エントリーに手動で追加する必要があります。これらは、ldapmodify で LDIF ファイルを渡すことですべて渡すことができます。

1. LDIF ファイルを作成します。各ステートメントは、stdin による変更の入力と同じです。これは、ダッシュ(-)で区切られた個別の更新ステートメントです。

```
dn: cn=config
changetype: modify
add: passwordChange
passwordChange: on
-
add: passwordExp
passwordExp: on
-
add: passwordMaxAge
passwordMaxAge: 8640000
-
add: passwordCheckSyntax
passwordCheckSyntax: on
-
add: passwordMinCategories
passwordMinCategories: 3
-
add: passwordStorageScheme
passwordStorageScheme: SSHA512
^D
```

以下の表は、パスワードポリシーの設定に使用できる属性を示しています。

表19.2 パスワードポリシー関連の属性

passwordChange	passwordCheckSyntax	passwordExp
passwordGraceLimit	passwordHistory	passwordInHistory
passwordMaxAge	passwordMaxRepeats	passwordMin8bit
passwordMinAge	passwordMinAlphas	passwordMinCategories
passwordMinDigits	passwordMinLength	passwordMinLowers
passwordMinSpecials	passwordMinTokenLength	passwordMinUppers

passwordMustChange	passwordSendExpiringTime	passwordStorageScheme
passwordTrackUpdateTime	passwordWarning	

パラメーターの詳細は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス を参照してください』。

2. ldapmodify コマンドで -f オプションを使用して、LDIF ファイルをサーバーに渡します。

```
# ldapmodify -D "cn=Directory Manager" -W -x -f user-pwdpolicy.ldif
```

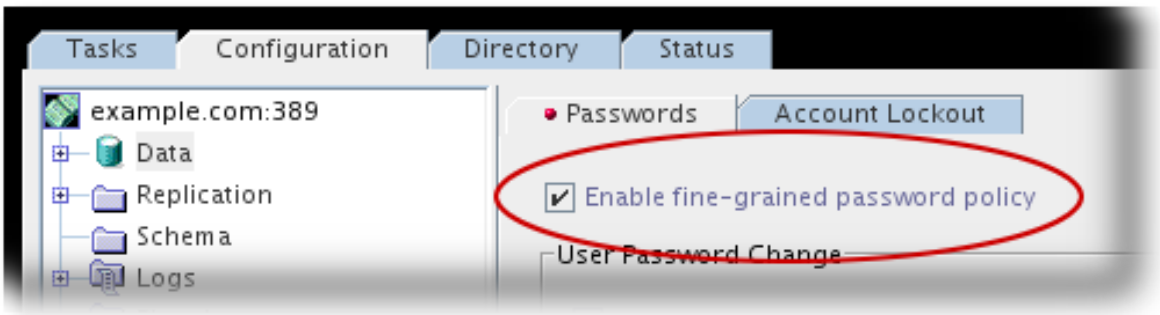
19.4.2. ローカルパスワードポリシーの設定

注記

パスワードポリシーを設定したら、アカウントロックアウトポリシーを設定します。詳細は、「パスワードベースのアカウントロックアウトポリシーの設定」を参照してください。

19.4.2.1. コンソールを使用したサブツリー/ユーザーパスワードポリシーの設定

1. 「コンソールを使用したグローバルパスワードポリシーの設定」 で説明されているように、きめ細かなパスワードポリシーをグローバルに有効にします。ユーザーレベルのパスワードポリシーを許可する場合は、Enable fine-grained password policy チェックボックスにチェックマークを入れてください。



注記

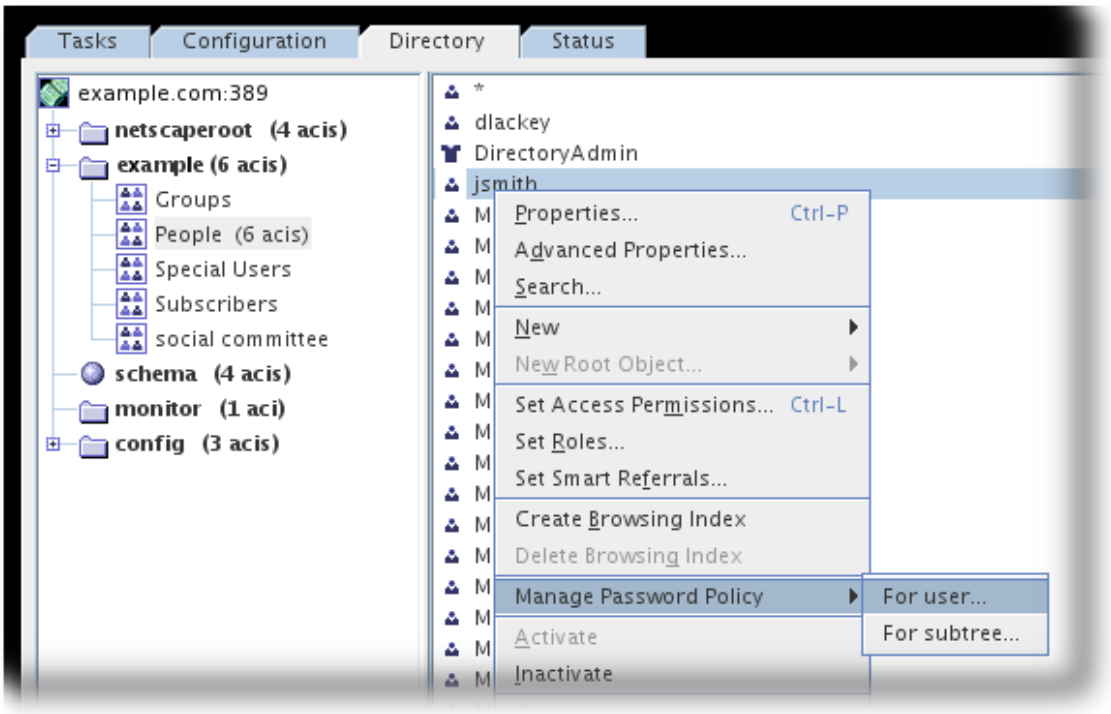
グローバルパスワードポリシーは、異なる場合はローカルポリシーを上書きしません。

2. サブツリーまたはユーザーのローカルパスワードポリシーを作成します。

   a. Directory タブを選択します。

   b. ナビゲーションペインで、パスワードポリシーを設定するサブツリーまたはユーザーエントリーを選択します。

   c. Object メニューから Manage Password Policy オプションを選択し、For user または For subtree を選択します。



   d. Passwords タブで Create subtree/user level password policy のチェックボックスを選択し、必要な属性を追加します。パスワードポリシー設定： リセット、有効期限、構文、および暗号化は、「コンソールを使用したグローバルパスワードポリシーの設定」 のグローバルポリシーと同じです。

e.

Account Lockout タブで、適切な情報を指定して **Save** をクリックします。

19.4.2.2. コマンドラインを使用したサブツリー/ユーザーパスワードポリシーの設定

1.

ns-newpwpolicy.pl スクリプトを実行して、必要な属性をサブツリーまたはユーザーエントリーに追加します。

スクリプトのコマンド構文は以下のとおりです。

```
# ns-newpwpolicy.pl [-D rootDN]  -w password | -w - | -j filename [-p port] [-h host] -U userDN -S suffixDN
```

サブツリーエントリーを更新するには、**- S** オプションを使用します。ユーザーエントリーを更新するには、**- U** オプションを使用します。**ns-newpwpolicy.pl** スクリプトは、一度

に 1 つのユーザーまたはサブツリーエントリーのみを受け入れます。ただし、ユーザーと接尾辞エントリーの両方を同時に使用することも可能です。このスクリプトの詳細は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス を参照してください』。

2. このスクリプトは、ターゲットエントリーがサブツリーまたはユーザーエントリーであるかに応じて、必要な属性を追加します。

サブツリー (例: ou=people,dc=example,dc=com) では、以下のエントリーが追加されます。

- サブツリーとそのすべての子について、さまざまなパスワードポリシー関連のエントリーを保持するためのサブツリーレベルのコンテナーエントリー (*nsPwPolicyContainer*)。以下に例を示します。

```
dn: cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
objectClass: top
objectClass: nsContainer
cn: nsPwPolicyContainer
```

- サブツリーに固有のすべてのパスワードポリシー属性を保持するための実際のパスワードポリシー仕様エントリー (*nsPwPolicyEntry*) です。以下に例を示します。

```
dn: cn="cn=nsPwPolicyEntry,ou=people,dc=example,dc=com",
    cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: ldapsubentry
objectclass: passwordpolicy
```

- 上記の (*nsPwPolicyEntry*) エントリーを指定する *pwdpolicysubentry* 値を持つ CoS テンプレートエントリー (*nsPwTemplateEntry*)。以下に例を示します。

```
dn: cn="cn=nsPwTemplateEntry,ou=people,dc=example,dc=com",
    cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: costemplate
objectclass: ldapsubentry
cosPriority: 1
pwdpolicysubentry: cn="cn=nsPwPolicyEntry,ou=people,dc=example,dc=com",
    cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
```

- サブツリーレベルでの CoS 仕様エントリー。以下に例を示します。

```
dn: cn=newpwdpolicy_cos,ou=people,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=cn=nsPwTemplateEntry\,ou=people\,dc=example,dc=com,
    cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
cosAttribute: pwdpolicysubentry default operational
```

ユーザーの場合（ uid=jdoe,ou=people,dc=example,dc=comなど）、以下のエントリーが追加されます。

- ユーザーとそのすべての子について、さまざまなパスワードポリシー関連のエントリーを保持するための親レベルのコンテナーエントリー (*nsPwPolicyContainer*)。以下に例を示します。

```
dn: cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
objectClass: top
objectClass: nsContainer
cn: nsPwPolicyContainer
```

- ユーザーに固有のパスワードポリシー属性を保持するための実際のパスワードポリシー仕様エントリー (*nsPwPolicyEntry*) です。以下に例を示します。

```
dn: cn="cn=nsPwPolicyEntry,uid=jdoe,ou=people,dc=example,dc=com",
    cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: ldapsubentry
objectclass: passwordpolicy
```

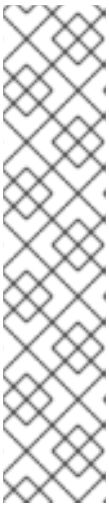
3. 上記のエントリー DN の値をターゲットエントリーの *pwdpolicysubentry* 属性に割り当てます。たとえば、以下はパスワードポリシーをユーザーエントリーに割り当てます。

```
dn: uid=jdoe,ou=people,dc=example,dc=com
changetype: modify
replace: pwdpolicysubentry
pwdpolicysubentry:
cn="cn=nsPwPolicyEntry,uid=jdoe,ou=people,dc=example,dc=com",
    cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
```

4.

適切な値でサブツリーまたはユーザーエントリーのパスワードポリシー属性を設定します。

表19.2「パスワードポリシー関連の属性」 には、パスワードポリシーの設定に使用できる属性が記載されています。ldapmodify ユーティリティーを使用すると、nsPwPolicyEntry オブジェクトクラスが含まれるサブツリーまたはユーザーエントリーのこれらの属性を変更できます。



注記

cn=config エントリーの *nsslapd-pwpolicy-local* 属性は、サーバーが強制するパスワードポリシーのタイプを制御します。デフォルトでは、この属性は無効(off)です。属性が無効になると、サーバーはグローバルパスワードポリシーのみを確認し、強制します。サブツリーおよびユーザーレベルのパスワードポリシーは無視されます。ns-newpwpolicy.pl スクリプトが実行されると、最初に指定されたサブツリーおよびユーザーエントリーをチェックし、それらが存在する場合は変更します。エントリーを正常に更新した後、スクリプトは *nsslapd-pwpolicy-local* 設定パラメーターを on に設定します。サブツリーおよびユーザーレベルのパスワードポリシーを有効にできない場合は、スクリプトの実行後に *nsslapd-pwpolicy-local* を off に設定するようにしてください。

ユーザーおよびサブツリーレベルのパスワードポリシーチェックをオフにするには、cn=config エントリーを変更して *nsslapd-pwpolicy-local* 属性を off に設定します。以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=config
changetype: modify
replace: nsslapd-pwpolicy-local
nsslapd-pwpolicy-local: off
```

この属性は、設定ファイル(dse.ldif)で直接変更して無効にすることもできます。

1. サーバーを停止します。

```
# systemctl stop dirsrv.target instance
```

2. テキストエディターで dse.ldif ファイルを開きます。

3. *nsslapd-pwpolicy-local* の値を off に設定し、保存します。

```
nsslapd-pwpolicy-local: off
```

4. サービスを起動します。

```
# systemctl start dirsrv.target instance
```

19.5. パスワードの有効期限コントロールの概要

ユーザーが有効なパスワードを使用して Directory Server に認証し、パスワードの期限が切れたり、リセットする必要がある場合、サーバーは以下の LDAP コントロールをクライアントに送信します。

- 期限切れのコントロール (2.16.840.1.113730.3.4.4): パスワードの有効期限が切れていることを示しています。Directory Server は以下の状況でこの制御を送信します。
  - パスワードの失効していますが、猶予のあるログインがなくなりました。サーバーは、Error 49 メッセージでバインドを拒否します。
  - パスワードは失効していますが、自動ログインは引き続き利用できます。バインドが許可されます。
  - cn=config エントリーで *passwordMustChange* を有効にし、管理者が変更した後にパスワードをリセットする必要があります。バインドが許可されますが、パスワードの変更以外の後続の操作では、Error 53 メッセージが表示されます。
- 期限切れコントロール (2.16.840.1.113730.3.4.5): パスワードが間もなく期限切れになることを示します。Directory Server は以下の状況でこの制御を送信します。
  - このパスワードは、cn=config エントリーの *passwordWarning* 属性に設定されたパスワード警告期間内に期限切れとなります。
  - cn=config エントリーの *passwordSendExpiringTime* 属性でパスワードポリシー設定オプションが有効になっている場合は、パスワードが警告期間内にいるかどうかにかかわらず、期限切れ制御が常に返されます。
- バインド応答制御 (1.3.6.1.4.1.42.2.27.8.5.1): この制御には、期限切れ間近のパスワード

や、もうすぐ期限切れになるパスワードの状態に関する詳細な情報が含まれています。

注記

Directory Server は、クライアントが要求した場合に限りバインド応答制御を送信します。たとえば、ldapsearch を使用する場合は、*-e ppolicy* パラメーターをコマンドに渡してバインド応答制御を要求する必要があります。

例19.1 Query でのバインド処理コントロールの要求

-e ppolicy パラメーターを ldapsearch コマンドに渡すなどしてバインド応答制御を要求する場合、サーバーはアカウントの有効期限に関する詳細情報を返します。以下に例を示します。

```
# ldapsearch -D "uid=user_name,dc=example,dc=com" -xLLL -W \
    -b "dc=example,dc=com" -e ppolicy
ldap_bind: Success (0); Password expired (Password expired, 1 grace logins remain)
```

19.6. DIRECTORY MANAGER パスワードの管理

Directory Manager は特権データベース管理者であり、Linux の root ユーザーと類似しています。Directory Manager エントリーと対応するパスワードは、インスタンスのインストール時に設定されます。

Directory Manager のデフォルトの識別名 (DN) は cn=Directory Manager です。

警告

パスワードに中括弧 ({}) を使用しないでください。Directory Server は、パスワードを {password-storage-scheme}hashed_password 形式で保存します。サーバーは、中括弧内の文字をパスワードストレージスキームとして解釈します。文字列が無効なストレージスキームであるか、パスワードが正しくハッシュ化されない場合、Directory Manager はサーバーに接続できません。

19.6.1. Directory Manager パスワードのリセット

Directory Manager のパスワードを紛失した場合は、リセットします。

1. Directory Server インスタンスを停止します。

```
# systemctl stop dirsrv@instance_name
```

2. 新しいパスワードハッシュを生成します。以下に例を示します。

```
# pwdhash -D /etc/dirsrv/slapd-instance_name password
{SSHA512}2eyW2uSFhh8LeB/nwZipfvFhSwL2DKZ58kXrCXsxr98Vz0nZl8fhd0W5BbL32
1Sr9Ulhzo3LhiQLiv4iVGF7hEGezlka65kN
```

Directory Server 設定へのパスを指定すると、***nsslapd-rootpwstoragescheme*** 属性に設定されたパスワードストレージスキームが自動的に使用され、新しいパスワードを暗号化します。

3. /etc/dirsrv/slapd-***instance_name***/dse.ldif ファイルを編集し、***nsslapd-rootpw*** 属性を直前の手順で表示された値に設定します。

```
nsslapd-rootpw:
{SSHA512}2eyW2uSFhh8LeB/nwZipfvFhSwL2DKZ58kXrCXsxr98Vz0nZl8fhd0W5BbL321Sr9
Ulhzo3LhiQLiv4iVGF7hEGezlka65kN
```

4. Directory Server インスタンスを起動します。

```
# systemctl start dirsrv@instance_name
```

19.6.2. Directory Manager パスワードの変更

19.6.2.1. コマンドラインを使用した Directory Manager パスワードの変更

コマンドラインを使用して Directory Manager パスワードを変更するには、サーバーは暗号化された接続に対応している必要があります。サーバーが暗号化された接続に対応していない場合は、Directory Server コンソールを使用して Directory Manager パスワードを更新します。「Directory Server コンソールを使用した Directory Manager パスワードの変更」 を参照してください。

サーバーが暗号化された接続に対応している場合は、以下の手順を実行してパスワードを変更します。

1. 新しいパスワードハッシュを生成します。以下に例を示します。

```
# pwdhash -D /etc/dirsrv/slapd-instance_name password
{SSHA512}2eyW2uSFhh8LeB/nwZipfvFhSwL2DKZ58kXrCXsxr98Vz0nZl8fhd0W5BbL32
1Sr9Ulhzo3LhiQLiv4iVGF7hEGezIka65kN
```

Directory Server 設定へのパスを指定すると、*nsslapd-rootpwstoragescheme* 属性に設定されたパスワードストレージスキームが自動的に使用され、新しいパスワードを暗号化します。

2. セキュアな接続 (STARTTLS) を使用して、前のステップで表示される値に *nsslapd-rootpw* 属性を設定します。

```
# ldapmodify -W -x -D "cn=Directory Manager" -p 389 -h server.example.com -x -ZZ

dn: cn=config
changetype: modify
replace: nsslapd-rootpw
nsslapd-rootpw:
{SSHA512}2eyW2uSFhh8LeB/nwZipfvFhSwL2DKZ58kXrCXsxr98Vz0nZl8fhd0W5BbL321Sr9
Ulhzo3LhiQLiv4iVGF7hEGezIka65kN
```

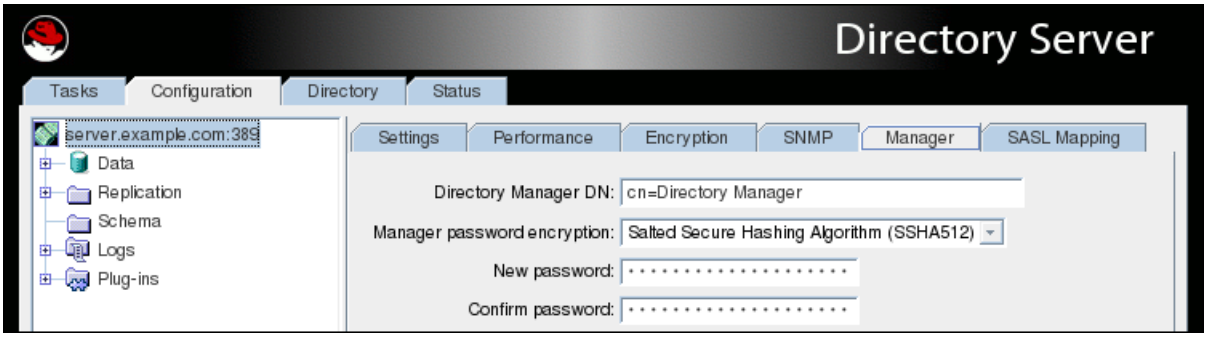
19.6.2.2. Directory Server コンソールを使用した Directory Manager パスワードの変更

管理者として、パスワードを変更するには、以下の手順を実施します。

1. Directory Server コンソールを開きます。詳細は、「Directory Server コンソールを開く」 を参照してください。

2. Configuration タブで、左側のペインのホスト名を選択し、Manager タブをクリックします。

3. 新しいパスワードを入力して確定します。



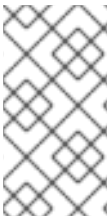
4. Save をクリックします。

19.6.3. Directory Manager パスワードストレージスキームの変更

パスワードストレージスキームでは、Directory Server がどのアルゴリズムをパスワードハッシュに使用するかを指定します。コマンドラインを使用してストレージスキームを変更するには、サーバーは暗号化された接続に対応している必要があります。サーバーが暗号化された接続に対応していない場合は、Directory Server コンソールを使用してストレージスキームを設定します。「コンソールを使用した Directory Manager パスワードストレージスキームの変更」 を参照してください。

Directory Manager(*nsslapd-rootpwstoragescheme*)のストレージスキームは、ユーザーパスワードの暗号化に使用されるスキームと異なる場合があります(*nsslapd-pwstoragescheme*)。

サポート対象のパスワードストレージスキームの一覧は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス の該当するセクションを参照してください』。



注記

Directory Manager のパスワードストレージスキームを変更する場合は、そのパスワードもリセットする必要があります。既存のパスワードは再暗号化できません。

19.6.3.1. コマンドラインを使用した Directory Manager パスワードストレージスキームの変更

サーバーが暗号化された接続に対応している場合は、以下の手順に従ってパスワードストレージのスキームを変更します。

1. 新しいストレージスキームを使用する新しいパスワードハッシュを生成します。以下に例を示します。

```
# pwdhash -s SSHA512 password
{SSHA512}2eyW2uSFhh8LeB/nwZipfvFhSwL2DKZ58kXrCXsxr98Vz0nZl8fhd0W5BbL32
1Sr9Ulhzo3LhiQLiv4iVGF7hEGezIka65kN
```

2. ***nsslapd-rootpwstoragescheme*** 属性をストレージスキームに設定し、***nsslapd-rootpw*** 属性をセキュアな接続 (STARTTLS) を使用して、前の手順で表示した値に設定します。

```
# ldapmodify -W -x -D "cn=Directory Manager" -p 389 -h server.example.com -x -F

dn: cn=config
changetype: modify
replace: nsslapd-rootpwstoragescheme
nsslapd-rootpwstoragescheme: SSHA512
-
replace: nsslapd-rootpw
nsslapd-rootpw:
{SSHA512}2eyW2uSFhh8LeB/nwZipfvFhSwL2DKZ58kXrCXsxr98Vz0nZl8fhd0W5BbL321Sr9
Ulhzo3LhiQLiv4iVGF7hEGezIka65kN
```

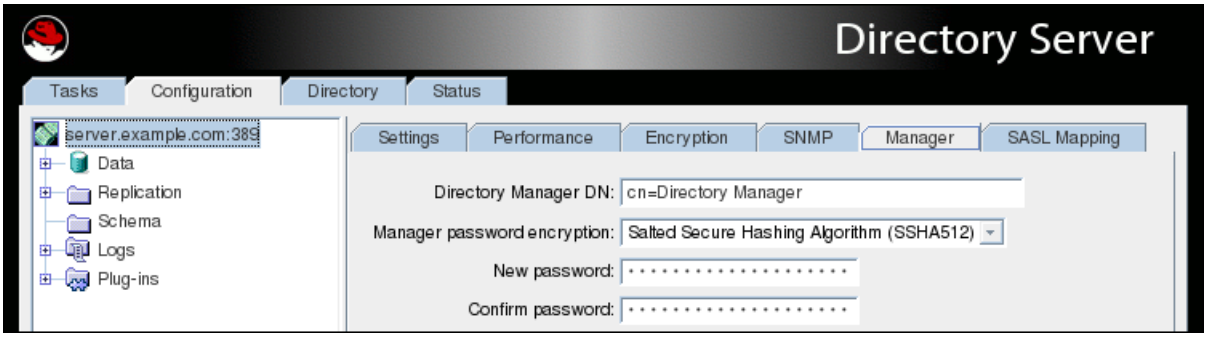
19.6.3.2. コンソールを使用した Directory Manager パスワードストレージスキームの変更

管理者として、Directory Manager パスワードストレージスキームを変更するには、以下の手順を実施します。

1. Directory Server コンソールを開きます。詳細は、「Directory Server コンソールを開く」 を参照してください。

2. Configuration タブで左側のペインのホスト名を選択し、Manager タブをクリックします。

3. Manager password encryption フィールドで新しいパスワードストレージスキームを選択します。



4. 新しいパスワードを入力して確定します。

5. Save をクリックします。

19.6.4. Directory Manager DN の変更

19.6.4.1. コマンドラインを使用した Directory Manager DN の変更

管理者として、Directory Manager DN を cn=New Directory Manager に変更するには、以下の手順を実施します。

```
# ldapmodify -W -x -D "cn=Directory Manager" -p 389 -h server.example.com -x

dn: cn=config
changetype: modify
replace: nsslapd-rootdn
nsslapd-rootdn: cn=New Directory Manager
```

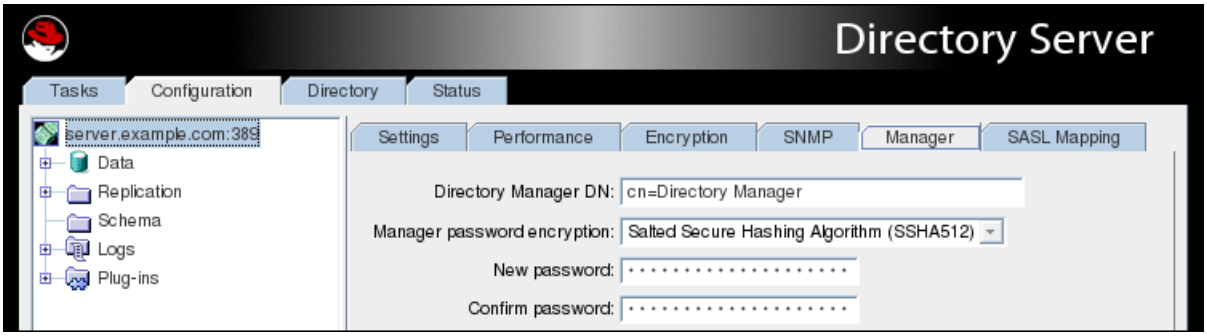
19.6.4.2. コンソールを使用した Directory Manager DN の変更

管理者として、Directory Manager DN を変更するには、以下の手順を実行します。

1. Directory Server コンソールを開きます。詳細は、「Directory Server コンソールを開く」 を参照してください。

2. Configuration タブで左側のペインのホスト名を選択し、Manager タブをクリックします。

3. Directory Manager DN フィールドに、Directory Manager の新しい DN を入力します。



4.
Save をクリックします。

19.7. パスワードなしのアクセスについてのアカウント可用性の確認

多くの場合は、Directory Server がユーザーアカウントの認証情報を返すため、クライアントは実際にそのユーザーとしてバインド (またはバインドを試行) します。また、バインドの試行には、ユーザー認証情報 (通常はパスワードまたは証明書) が必要です。Directory Server は、認証されていないバインドおよび匿名バインドを許可しますが、いずれのバインドもユーザーアカウント情報をすべて返しません。

クライアントが他の操作を行うために、ユーザーアカウントに関する情報、特にそのアカウントの認証を許可するかどうかの情報を必要とする場合がありますが、クライアントは Directory Server にそのユーザーアカウントの認証情報を持っていないか、または使用しています。基本的に、クライアントは、ユーザーアカウント情報 (アカウントにパスワードがある場合は、パスワードの有効期限情報を含む) を取得するために、認証情報なしで認証されたバインド操作を実行する必要があります。

Account Usability Extension Control を渡すと、ldapsearch でこれを行うことができます。このコントロールは、あたかも特定のユーザーに対して認証済みのバインド操作を行い、そのユーザーのアカウントステータスを返すかのように動作しますが、実際にはサーバーにバインドすることはありません。これにより、クライアントはそのアカウントがログインに使用できるかどうかを判断し、そのアカウント情報を PAM などの別のアプリケーションに渡すことができます。

たとえば、Account Usability Extension Control を使用すると、システムが Directory Server を ID バックエンドとして使用でき、認証操作が Directory Server の外部で実行されるスマートカードや SSH 鍵などのパスワードなしの認証方法を使用できます。

19.7.1. アカウントのユーザビリティー拡張制御を使用したエントリーの検索

Account Usability Extension Control は ldapsearch の拡張機能です。アカウントのステータスと、そのアカウントのパスワードポリシーに関する情報を提供する、返された各エントリーの追加行を返します。次に、クライアントまたはアプリケーションはそのステータスを使用して、そのユーザーアカウントの Directory Server 外で行われた認証の試行を評価できます。基本的に、この制御では、認証操作なしにユーザーが認証を許可するかどうかを制御します。

注記

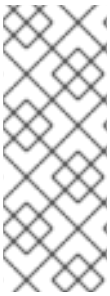
Directory Server が使用する OpenLDAP ツールは、Account Usability Extension Control に対応していません。OpenDS などの他の LDAP ユーティリティーや、制御をサポートする他のクライアントを使用できます。

例えば、OpenDS ツールを使用する場合、コントロールは、コントロールOID (1.3.6.1.4.1.42.2.27.9.5.8) を持つ -J、または accountusability:true フラグを使用して指定することができます。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b "dc=example,dc=com"
-s sub -J "accountusability:true" "(objectclass=*)"
# Account Usability Response Control
# The account is usable
dn: dc=example,dc=com
objectClass: domain
objectClass: top
dc: example
...
```

これは、特定のエントリーに対して実行することもできます。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b
"uid=bjensen,ou=people,dc=example,dc=com" -s base -J "accountusability:true" "(objectclass=*)"
# Account Usability Response Control
# The account is usable
dn: uid=bjensen,ou=people,dc=example,dc=com
...
```



注記

デフォルトでは、Directory Manager のみが Account Usability Extension Control を使用できます。他のユーザーが Account Usability Extension Control を使用できるようにするには、cn=features でサポート対象のコントロールエントリー上の ACI に設定します。「アカウントのユーザビリティー検索の対象を変更」を参照してください。

このコントロールは、アカウントの実際のステータスや、ユーザーアカウントのパスワードポリシー設定 (ユーザーにパスワードがある場合) に応じて異なるメッセージを返します。

表19.3 アカウント信頼性制御の結果メッセージ

アカウントのステータス	コントロール結果メッセージ
有効なパスワードがあるアクティブなアカウント	The account is usable
パスワードが設定されていないアクティブなアカウント	The account is usable
期限切れのパスワード	Password expired

アカウントのステータス	コントロール結果メッセージ
アカウントのパスワードポリシーが変更される	Password expired
アカウントがロックされ、ロックアウト期間がありません	Password reset
アカウントがロックされ、ロックアウト期間があります	**Time** (in seconds) for automatic unlock of the account
最初のログイン時にアカウントのパスワードをリセットする必要があります	Password reset
パスワードの期限が切れており、猶予期間ログインが許可されます	Password expired and **X** grace login is allowed
パスワードの有効期限が切れ、猶予ログイン回数がなくなっています。	Password expired
パスワードの有効期限が切れます (期限切れの警告)。	Password will expire in **X** number of seconds

19.7.2. アカウントのユーザビリティー検索の対象を変更

デフォルトでは、Directory Manager のみが Account Usability Extension Control を使用できます。他のユーザーは、サポートされるコントロールエントリーに適切な ACI を設定することで、Account Usability Extension Control を使用できます。コントロールエントリーは、Account Usability Extension Control OID (1.3.6.1.4.1.42.2.27.9.5.8) に対して名前が付けられます。

たとえば、cn=Administrators,ou=groups,dc=example,dc=com グループのメンバーが、全ユーザーの Account Usability Extension Control を読み取れるようにするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -x

dn: oid=1.3.6.1.4.1.42.2.27.9.5.8,cn=features,cn=config
changetype: modify
add: aci
aci: (targetattr = "*")(version 3.0; acl "Account Usable"; allow (read)(groupdn =
"ldap:///cn=Administrators,ou=groups,dc=example,dc=com");)
```

19.8. パスワードベースのアカウントロックアウトポリシーの設定

アカウントのロックアウトは、ユーザーのパスワードを繰り返し推測することで、ディレクトリーに分割を試行するハッカーから保護されます。パスワードポリシーを設定して、特定のユーザーが指定し

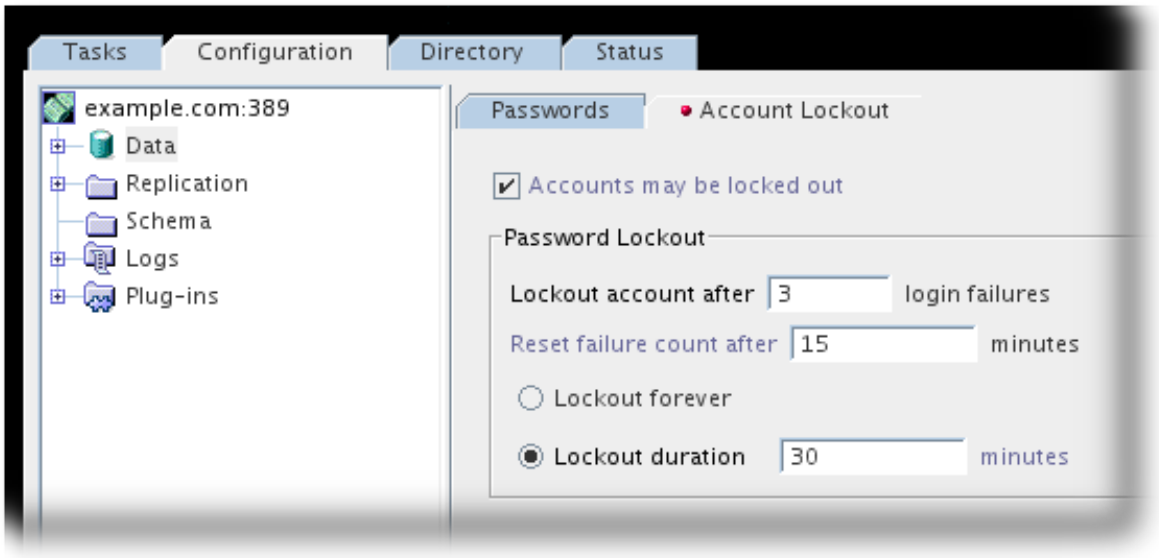
た数のバインドの試行後にディレクトリーからロックされるようにできます。

19.8.1. コンソール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 のアカウントロックアウトポリシーを設定または変更するには、以下を実行します。

1. Configuration タブを選択し、Data ノードを選択します。

2. 右側のペインで、Account Lockout タブを選択します。



3. アカウントのロックアウトを有効にするには、Accounts may be locked out のチェックボックスを選択します。

4. X ログインの失敗テキストボックスの後に、Lockout アカウントに 許可されるバインド失敗の最大数を入力します。サーバーは、ここで指定した制限を超えるユーザーをロックします。

5. X minutes テキストボックスの後に Reset failure カウンターに、サーバーが待機した後に、バインド失敗カウンターがゼロにリセットされるまでの数分を入力します。

6. ユーザーがディレクトリーからロックされる間隔を設定します。

-

パスワードを管理者がリセットするまで ロックアウト ラジオボタンを選択し てユーザーをロックアウトします。

- Lockout Duration ラジオボタンを選択し、テキストボックスに時間（分単位）を入力して、特定のロックアウト期間を設定します。

7. Save をクリックします。

19.8.2. コマンドラインを使用したアカウントロックアウトポリシーの設定

ldapmodify を使用して、cn=config エントリーでアカウントロックアウトポリシーを設定します。以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -x -p 389 -h server.example.com -x
dn: cn=config
changetype: modify
replace: passwordLockout
passwordLockout: on
-
add: passwordMaxFailure
passwordMaxFailure: 4
-
add: passwordLockoutDuration
passwordLockoutDuration: 600
-
```

アカウントロックアウトポリシーに関連する属性は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス に記載されています。』

以下の属性は、アカウントパスワードポリシーを制御します。

- passwordLockout
- passwordMaxFailure
- passwordUnlock

- passwordLockoutDuration
- passwordResetFailureCount

19.8.3. レガシーパスワードロックアウト動作の無効化

パスワードの最大失敗 (*passwordMaxFailure*) に達すると、解釈する方法は複数あります。これは、サーバーが最後の失敗を全体の失敗数にどのようにカウントするかによります。

LDAP クライアントの従来の動作は、制限に達した 後に 障害が発生した場合を想定することです。つまり、失敗回数を 3 回に設定すると、4 回目の失敗でロックアウトされます。これは、4 回目の試みが成功した場合、技術的には失敗の限界に達していたとしても、ユーザーは正常に認証できることを意味します。これは、カウントの n+1 です。

LDAP クライアントは、最大失敗制限を増やして、最後の失敗試行を最終的な試行としてカウントすることを期待します。そのため、障害の上限が 3 に設定されている場合、3 番目の障害によりアカウントはロックされます。4 番目の試行では、正しい認証情報を使用しても失敗します。これはカウントの n です。

最初のシナリオ (試行回数が超過した場合にのみアカウントがロックされる) は過去の動作なので、これは従来のパスワードポリシーの動作と考えられます。Directory Server では、このポリシーはデフォルトで有効になっているため、障害数が n+1 の場合のみアカウントがロックされます。このレガシー動作を無効にして、新しい LDAP クライアントが予想される際にエラー (LDAP_CONSTRAINT_VIOLATION) を受け取るようにすることができます。これは、*passwordLegacyPolicy* パラメーターで設定されます。

以下に例を示します。

```
[root@server ~]# ldapmodify -D "cn=Directory Manager" -x -D "cn=directory manager" -W -p
389 -h server.example.com -x
dn: cn=config
replace: passwordLegacyPolicy
passwordLegacyPolicy: off
```

19.9. 時間ベースのアカウントロックアウトポリシーの構成

認証に失敗した場合にアカウントをロックする以外にも、アカウントの非アクティブ化やアカウントのエイジに基づいてアカウントロックアウトポリシーを定義する方法があります。アカウントポリシープラグインは、相対 時間設定を使用してアカウントをロックする必要があるかどうかを判断します。

注記

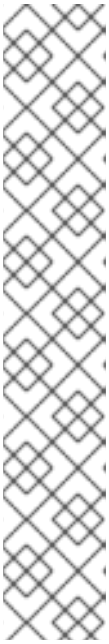
サービスのロールまたはクラスは、絶対 アカウント時間に基づいてアカウントを非アクティブにするのに使用できます。たとえば、特定の日付の前に作成されたすべてのアカウントで CoS を作成できます。

アカウントポリシープラグインには、3 つの設定エントリーが必要です。

- プラグイン自体の設定エントリー。これにより、そのサーバーに設定されたすべてのアカウントポリシーに使用されるグローバル値が設定されます。

- アカウントポリシー設定エントリー。このエントリーはユーザーディレクトリー内にあり、基本的にはユーザーアカウントエントリーに参照および適用されるテンプレートとなります。

- アカウントポリシーエントリーを適用するエントリー。ユーザーアカウントは、直接アカウントポリシーを参照したり、CoS またはロールを使用してアカウントポリシーを自動的にユーザーアカウントのセットに適用することができます。



注記

アカウントポリシーは、*acctPolicySubentry* 属性を介して適用されます。この属性はユーザーアカウントに直接追加できますが、この属性は単値になります。つまり、そのアカウントにはアカウントポリシーを 1 つだけ適用できます。

これはほとんどの場合で問題ありません。しかし、現実的には、2 つのアカウントポリシーを作成することができます。1 つはアカウントの非アクティブ化のため、もう1つは年齢に基づくアカウントの失効のためです。

CoS を使用してアカウントポリシーを適用すると、複数のアカウントポリシーをアカウントに使用できます。

19.9.1. アカウントポリシープラグインの構文

Account Policy プラグイン自体には 2 つの設定属性のみがあります。

- **nsslapd-pluginEnabled:** プラグインが有効かどうかを設定します。この属性はデフォルトで **off** です。

- nsslapd-pluginarg0: プラグイン設定ディレクトリーの DN を参照します。設定エントリーは、通常プラグイン自体の子エントリーです (例: cn=config,cn=Account Policy Plugin,cn=plugins,cn=config)。

そのため、アカウントポリシーは 2 つの部分で定義されます。

- nsslapd-pluginarg0 属性で特定されたプラグイン設定エントリー。これにより、アカウントポリシー設定エントリーの特定やユーザーアカウントエントリーの管理に使用するプラグインのグローバル設定が設定されます。これらの設定はサーバー全体に適用されます。

  設定エントリー属性は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス』 の 『アカウントポリシープラグイン属性』 セクションで説明されています。

- アカウントポリシーの設定エントリー。これは、アカウントポリシーに特定の値を設定するテンプレートエントリーとよく似ています。ユーザーアカウントは、直接または CoS エントリーを介して、このアカウントポリシーエントリを参照します。

  アカウントポリシーとユーザーエントリーの属性については、以下の表で説明されています。

表19.4 アカウントポリシーエントリーおよびユーザーエントリーの属性

属性	定義	設定またはユーザーエントリー
accountpolicy (オブジェクトクラス)	アカウントの無効化または期限切れポリシーのテンプレートエントリーを定義します。	設定
accountInactivityLimit (属性)	アカウントの最終ログイン時刻から、非アクティブ時にアカウントがロックされるまでの時間を秒単位で設定します。	設定

属性	定義	設定またはユーザーエントリー
acctPolicySubentry (属性)	アカウントのポリシー (具体的には、アカウントロックアウトポリシー) に属するエントリーを指定します。この属性の値は、エントリーに適用されるアカウントポリシーの DN を参照します。	ユーザー
createTimestamp (操作属性)	エントリーが最初に作成された日時が含まれます。	ユーザー
lastLoginTime (操作属性)	指定のアカウントがディレクトリーに対して認証された最終時刻のタイムスタンプが含まれます。	ユーザー

詳細は、Red『Hat Directory Server の設定、コマンド、およびファイルリファレンス』の属性の説明を参照してください。

19.9.2. アカウントアクティビティーとアカウントの有効期限

Account Policy プラグインを使用すると、以下を設定できます。

- アカウントの有効期限: アカウントの作成後に一定時間無効になります。
- アカウントの非アクティブ化: 最後にログインに成功してから一定時間が経過すると、アカウントが無効になります。これにより、未使用のアカウントを自動的に無効にできます。

無効にされたアカウントはログインできなくなります。

Account Policy プラグインを設定するには、以下を実行します。

1. アカウントポリシープラグインを有効にします。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
```

```
dn: cn=Account Policy Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

2. nsslapd-pluginarg0 属性を、プラグイン設定エントリーを参照するように設定します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=Account Policy Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginarg0
nsslapd-pluginarg0: cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
```

3. プラグイン設定エントリーを作成します。

- アカウントポリシーで CoS またはロールを使用するには、*alwaysRecordLogin* の値を yes に設定します。これは、*acctPolicySubentry* 属性がない場合でも、すべてのエントリーにログイン時間が記録されることを意味します。

- アカウントポリシー評価に使用するプライマリー属性を *stateAttrName* の値として設定します。アカウントの停止状態の場合は、lastLoginTime 属性を使用します。単純なアカウントの有効期限の場合は、createTimestamp 属性を使用します。

- *altStateAttrName* にセカンダリー属性を設定できます。これは、*stateAttrName* で定義されたプライマリー属性が存在しない場合にチェックできます。属性を指定していない場合は、デフォルト値の createTimestamp が使用されます。

警告

プライマリー属性の値が lastLoginTime と *altStateAttrName* で createTimestamp に設定されていると、既存の環境のユーザーは、アカウントに lastLoginTime 属性がなく、設定した非アクティブ期間よりも *createTimestamp* が古い場合に、既存の環境のユーザーは自動的にロックされます。

この状況に対処するには、代替属性を 1.1 に設定します。これは、代替として属性を使用しないことを明示します。*lastLoginTime* 属性は、ユーザーが次回ログインした後に自動的に作成されます。

- アカウントポリシーを適用するエントリーを表示するのに使用する属性を設定します (acctPolicySubentry)。

- 実際のタイムアウト期間の設定に使用されるアカウントポリシーの属性を秒単位で設定します (accountInactivityLimit)。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=config,cn=Account Policy Plugin,cn=plugins,cn=config

objectClass: top
objectClass: extensibleObject
cn: config
alwaysRecordLogin: yes
stateAttrName: lastLoginTime
altStateAttrName: 1.1
specattrname: acctPolicySubentry
limitattrname: accountInactivityLimit
```

4. サーバーを再起動して、新しいプラグイン設定を読み込みます。

```
# systemctl start dirsrv.target
```

5. アカウントポリシーを定義します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
```

```
dn: cn=Account Inactivation Policy,dc=example,dc=com

objectClass: top
objectClass: ldapsubentry
objectClass: extensibleObject
objectClass: accountpolicy
accountInactivityLimit: 2592000
cn: Account Inactivation Policy
```

6. サービステンプレートエントリーのクラスを作成します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=TempltCoS,dc=example,dc=com

objectClass: top
objectClass: ldapsubentry
objectClass: extensibleObject
objectClass: cosTemplate
acctPolicySubentry: cn=Account Inactivation Policy,dc=example,dc=com
```

アカウントポリシーは、CoS を使用する代わりに、ユーザーエントリーで直接定義できます。しかし、CoS を使用することで、複数のエントリーに対して確実にアカウントポリシーを適用および更新することができ、1 つのエントリーに複数のポリシーを適用することができます。

7. サービス定義エントリーのクラスを作成します。CoS の管理エントリーは、アカウントポリシーの属性 *acctPolicySubentry* です。この例では、CoS をディレクトリーツリー全体に適用します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=DefnCoS,dc=example,dc=com

objectClass: top
objectClass: ldapsubentry
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=TempltCoS,dc=example,dc=com
cosAttribute: acctPolicySubentry default operational-default
```

19.9.3. パスワード失効後の特定期間のアカウントの無効化

Directory Server では、パスワードの期限が切れてから一定期間アカウントを無効にするアカウントポリシーを設定できます。無効にしたアカウントはログインできなくなります。

この設定を設定するには、「アカウントアクティビティーとアカウントの有効期限」の手順に従います。ただし、プラグイン設定エントリーを設定する場合は、代わりに以下の設定を使用してください。

```
dn: cn=config,cn=Account Policy Plugin,cn=plugins,cn=config

objectClass: top
objectClass: extensibleObject
cn: config
alwaysrecordlogin: yes
stateAttrName: non_existent_attribute
altStateAttrName: passwordExpirationTime
specattrname: acctPolicySubentry
limitattrname: accountInactivityLimit
```

この設定では、*stateAttrName* パラメーターにダミー値を使用します。したがって、*altStateAttrName* パラメーターで設定した passwordExpirationTime 属性のみが、アカウントの期限が切れるタイミングを算出するために使用されます。

また、ユーザーエントリーの *lastLoginTime* 属性に最後に成功したログインの時間を記録するため、以下を設定します。

```
dn: cn=config,cn=Account Policy Plugin,cn=plugins,cn=config

alwaysRecordLoginAttr: lastLoginTime
```

この設定では、ユーザーの*passwordExpirationTime* 属性と*accountInactivityLimit* パラメーターの値に設定されている時間の合計が過去になった場合は、アカウントが自動的に無効になります。この設定では、ユーザーの *passwordExpirationTime* 属性と *accountInactivityLimit* パラメーターの値の合計が、*alwaysRecordLoginAttr* 属性が最後に更新されてからの時間を超えた場合は、アカウントが自動的に無効になります。

19.9.4. ロックアウトポリシーを設定しないログイン時間の追跡

アカウントポリシープラグインを使用して、有効期限や非アクティブ期間を設定 しなくても、ユーザーのログイン時間を追跡することもできます。この場合、アカウントポリシープラグインは *lastLoginTime* 属性をユーザーエントリーに追加するために使用されますが、他のポリシールールを設定する必要はありません。

その場合には、アカウントポリシープラグインを通常どおりに設定して、ログイン時間を追跡します。ただし、追跡中のログイン情報に作用する CoS は作成しないでください。

1. アカウントポリシープラグインを有効にします。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=Account Policy Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

2. nsslapd-pluginarg0 属性を、プラグイン設定エントリーを参照するように設定します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=Account Policy Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginarg0
nsslapd-pluginarg0: cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
```

3. ログイン時間を記録するプラグイン設定エントリーを作成します。

   - すべてのエントリーにログイン時間が記録されるように、*alwaysRecordLogin* の値を yes に設定します。
   - *lastLoginTime* 属性をアカウントポリシー (*stateattrname*) に使用する属性として設定します。
   - アカウントポリシーを適用するエントリーを表示するのに使用する属性を設定します (*acctPolicySubentry*)。
   - 実際のタイムアウト期間 (秒単位) を設定するのに使用されるアカウントポリシーの属性を設定します (*accountInactivityLimit*)。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=config,cn=Account Policy Plugin,cn=plugins,cn=config

objectClass: top
objectClass: extensibleObject
cn: config
alwaysRecordLogin: yes
stateattrname: lastLoginTime
```

```
altstateattrname: createTimestamp
specattrname: acctPolicySubentry
limitattrname: accountInactivityLimit
```

4. サーバーを再起動して、新しいプラグイン設定を読み込みます。

```
# systemctl start dirsrv.target
```

19.9.5. Inactive アカウントのロックの解除

アカウントポリシープラグインから非アクティブになっているアカウントは、管理者(ns-activate.pl)またはパスワードポリシーを介して手動で設定されたロックアウトの管理に使用するツールでは管理できません。

非アクティブの制限に達したためにアカウントがロックされている場合は、*lastLoginTime* 属性をリセットして再度アクティブにできます。以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: uid=jsmith,ou=people,dc=example,dc=com
changetype: modify
replace: lastLoginTime
lastLoginTime: 20160610080000Z
```

注記

lastLoginTime は、GMT/UTC 時間（Zulu タイムゾーン）で設定され、タイムスタンプに Z が付加されます。

19.10. アカウントロックアウト属性の複製

アカウントのロックアウトポリシーにより、ログイン試行が失敗した回数を超えると、ユーザー ID が Directory Server にアクセスできなくなります。これにより、ハッカーやその他の悪意のあるユーザーは、パスワードを推測することで Directory Server に不正にアクセスできなくなります。パスワードポリシーはローカルに設定され、通常、アカウントロックアウト属性は各レプリカに対してローカルになります。つまり、アカウントのロックアウト回数に達するまでは、あるレプリカにログインを試み、すぐに別のレプリカで再試行することができるのです。それを防ぐには、アカウントのロックアウト回数に関連する属性をエントリーに複製し、1 つのマスターでログイン試行に失敗した場合に、悪意のあるユーザーが構成内のすべてのサプライヤーとコンシューマーのレプリカからロックアウトされるようにします。

デフォルトでは、他のパスワード属性がある場合でも、3 つのパスワードポリシー属性は複製されません。これらの属性は、ログインの失敗およびロックアウト期間に関連します。

- *passwordRetryCount*
- *retryCountResetTime*
- *accountUnlockTime*

19.10.1. アカウントロックアウトおよびレプリケーションの管理

パスワードとアカウントのロックアウトポリシーの適用は、複製された環境では若干異なります。

- パスワードポリシーはデータマスターで実施されます。
- アカウントロックアウトは、レプリケーションに参加するすべてのサーバーに適用されます。

ディレクトリー内のパスワードポリシー情報の一部は、自動的に複製されます。

- *passwordMinAge* および *passwordMaxAge*
- *passwordExp*
- *passwordWarning*

ただし、設定情報はローカルに保持され、複製されません。この情報には、パスワード構文とパスワード変更の履歴が含まれます。アカウントロックアウトカウンターおよび層は、特にレプリケーション用に設定されていない限り複製されません。

複製された環境でパスワードポリシーを設定する場合は、これらの要素が有効であることを確認し、パスワードポリシーとアカウントロックアウト設定が一貫して実行されるようにします。

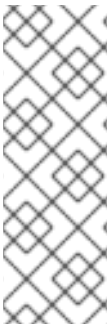
-

パスワードの有効期限が迫っていることを示すサーバーからの警告は、すべてのレプリカで発行されます。この情報は、各サーバーにローカルに保存されるため、ユーザーが複数のレプリカにバインドされた場合は、同じ警告が複数回発行されます。また、ユーザーがパスワードを変更した場合は、その情報がレプリカに反映されるまでに時間がかかることがあります。ユーザーがパスワードを変更してすぐに再バインドすると、レプリカが変更を登録するまでバインドに失敗することがあります。

- サプライヤーやレプリカなど、すべてのサーバーで同じバインド動作が発生する必要があります。各サーバーで同じパスワードポリシー設定情報を作成してください。

- アカウントロックアウトカウンターは、マルチマスター環境で期待どおりに機能しない可能性があります。アカウントのロックアウトカウンターは、デフォルトでは複製されません (ただし、設定は可能です)。アカウントのロックアウト属性がまったく複製されない場合、あるユーザーがあるサーバーからロックアウトされていても、別のサーバーには正常にバインドできる可能性があります (または、あるサーバーではロックが解除されていても、別のサーバーではブロックされている場合もあります)。アカウントロックアウト属性が複製されると、アカウントのロックアウトの変更と、その変更が他のサーバーに伝播されるときにラグが発生することがあります。これはレプリケーションのスケジュールにより異なります。

- レプリケーション用に作成されるエントリー (例: サーバーアイデンティティー) には有効期限のないパスワードが必要です。これらの特別なユーザーに有効期限のないパスワードがあることを確認するには、エントリーに *passwordExpirationTime* 属性を追加し、その値を 20380119031407Z 有効な範囲内に) 指定します。



注記

パスワードポリシーが有効になり、*alwaysRecordLogin* パラメーターが yes に設定されている場合、*lastLoginTime* 属性の値はマスターと読み取り専用レプリカで異なる場合があります。たとえば、ユーザーが読み取り専用のレプリカにログインすると、*lastLoginTime* 属性はローカルに更新されますが、値はマスターサーバーに複製されません。

19.10.2. パスワードポリシー属性を複製する Directory Server の設定

特別なコア設定属性は、パスワードポリシーの操作属性が複製されるかどうかを制御します。これは、コンシューマー Directory Server 設定で有効になっている *passwordIsGlobalPolicy* 属性で、コンシューマーがパスワードポリシーの操作属性を受け入れるようにします。

デフォルトでは、この属性は off に設定されます。

これらの属性を複製できるようにするには、コンシューマーの *passwordIsGlobalPolicy* 設定属性を変更します。

```
# ldapmodify -D "cn=Directory Manager" -W -x -h consumer1.example.com

dn: cn=config
changetype: modify
replace: passwordIsGlobalPolicy
passwordIsGlobalPolicy: on
```

この値を on に変更すると、*passwordRetryCount*、*retryCountResetTime*、および *accountUnlockTime* が複製されます。複製された属性で属性を含めるには、他の設定は必要ありません。

19.10.3. パスワードポリシー属性に対する一部レプリケーションの設定

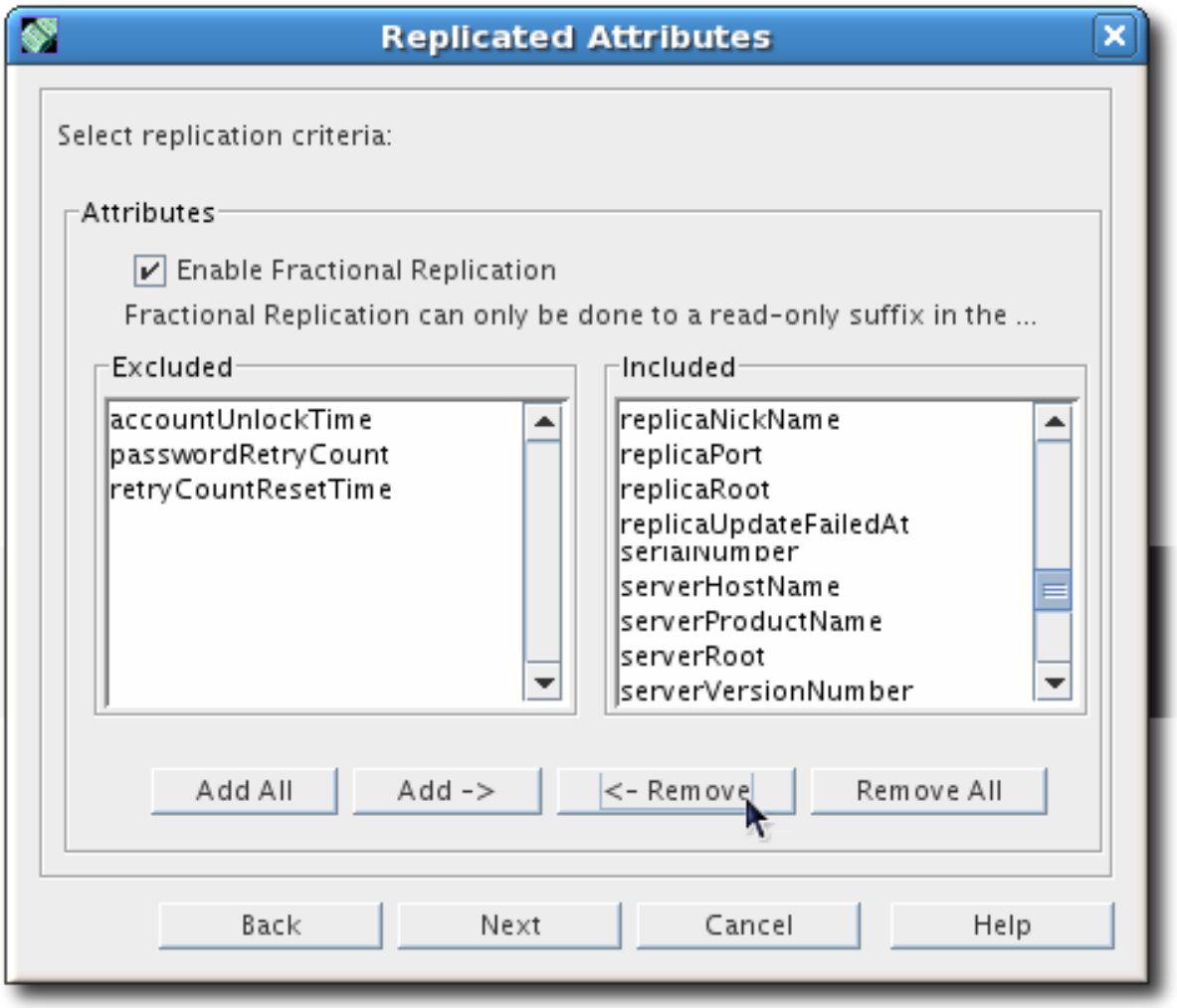
passwordIsGlobalPolicy 属性を設定すると、コンシューマーがそれらの属性への更新を受け取ることができるように、レプリケーションのコンシューマーに影響します。パスワードポリシー属性がサプライヤーによって実際に複製されるかどうかを制御するには、特定のエントリー属性が複製されるものを制御する一部レプリケーションを使用します。

パスワードポリシー属性を複製する必要がある場合は (デフォルトでは設定) 一部レプリカ合意にこれらの属性が含まれていることを確認してください。

コンシューマーで *passwordIsGlobalPolicy* 属性が off に設定されているため、パスワードポリシー属性を複製する必要がない場合は、一部レプリケーション (「一部レプリケーションを使用した属性のサブセットの複製」で説明) を使用してサプライヤーでレプリケーションを強制し、それらの属性をレプリカ合意から明確に除外します。

1. 「レプリカ合意の作成」 で説明したようにサプライヤーにレプリカ合意を設定する場合は、Enable Fractional Replication チェックボックスを選択します。

2. デフォルトでは、すべての属性が Replicated 属性 ボックスに一覧表示されます。*passwordRetryCount*、*retryCountResetTime*、および *accountUnlockTime* パラメーターを選択し、矢印ボタンをクリックして Do Not Replicate ボックスに移動します。



3. レプリカ合意の設定を終了します。

19.11. 異なるタイプのバインドの有効化

エンティティーが Directory Server にログインするかアクセスするたびにディレクトリーに バインド されます。バインド操作にはさまざまな種類があり、バインドの方法に応じたもの (シンプルバインドやオートバインドなど) や、ディレクトリーにバインドするユーザーのアイデンティティーに応じたもの (匿名バインドや未認証バインド) があります。

以下のセクションでは、バインドのセキュリティーを高めたり (「セキュアなバインドの要求」)、バインド操作を効率化したり (「自動バインドの設定」など) するための設定パラメーターを紹介します。

19.11.1. セキュアなバインドの要求

単純なバインドは、エンティティーが単純なバインド DN とパスワードの組み合わせを使用して Directory Server に対して認証される場合です。コマンドラインからパスワードを直接送信するのではなく、パスワードファイルを使用することは可能ですが、いずれの方法でもネットワーク経由で平文の

パスワードを送受信する必要があります。これでは、接続を盗聴された場合に、パスワードが脆弱になってしまいます。

セキュアな接続（TLS または Start TLS）で単純なバインドを行うことが必要になる場合があります。これにより、バインド操作で送信される平文のパスワードを実質的に暗号化できます。(SASL 認証や証明書ベースの認証など、簡易バインドの代わりに使用することも可能です。)

重要

通常ユーザーは、サーバーおよび LDAP 操作にログインすると、単純なバインドにセキュアな接続を要求することで、サーバー間の接続に影響があります。たとえば、レプリケーション、同期、データベースチェーンはすべて、サーバー間で単純なバインドを使用できます。

nsslapd-require-secure-binds 属性が有効にになっている場合は、レプリカ合意、同期合意、およびチェーン設定がセキュアな接続を指定するようにしてください。それ以外の場合、これらの操作は失敗します。

注記

バインド操作のセキュアな接続を 認証バインド にのみ適用する必要があります。パスワードのないバインド操作 (匿名および認証されていないバインド) は、標準の接続を引き継ぐことができます。

1. *nsslapd-require-secure-binds* 属性を cn=config エントリーに追加します。

```
# ldapmodify -D "cn=Directory Manager" -W -x

dn: cn=config
changetype: modify
replace: nsslapd-require-secure-binds
nsslapd-require-secure-binds: on
```

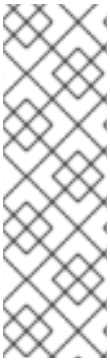
2. サービスを再起動します。

```
# systemctl restart dirsrv.target
```

19.11.2. 匿名バインドの無効化

ユーザー名またはパスワードを指定せずに Directory Server への接続を試みると、これは 匿名バイ

ンド になります。匿名バインドは、ユーザーが最初にディレクトリーに対して認証を行う必要がないため、電話番号や電子メールアドレスをディレクトリーで確認するような、一般的な検索および読み取り操作を簡素化します。



注記

デフォルトでは、匿名バインドは検索操作および読み取り操作に対して許可 (on) されます。これにより、ユーザーおよびグループのエントリーに加えて、root DSE などの設定エントリーを含む 通常のディレクトリーエントリー にアクセスすることができます。別のオプション rootdse により、匿名検索および root DSE 自体への読み取りアクセスが許可されますが、他のすべてのディレクトリーエントリーへのアクセスを制限します。

ただし、匿名バインドにはリスクがあります。機密情報へのアクセスを制限したり、変更や削除などのアクションを許可しないように、適切な ACI を導入する必要があります。さらに、匿名バインドは、サービス拒否攻撃や、悪意のあるユーザーがサーバーへのアクセスを取得するのに使用できます。

「匿名アクセスの付与」は、ACI を設定して匿名ユーザーがアクセスするものを制御する例があり、「匿名バインドでのリソース制限の設定」には、匿名ユーザーのリソース制限の設定に関する情報があります。

このオプションで十分なレベルのセキュリティーが提供されない場合は、匿名バインドを完全に無効にできます。

1. *nsslapd-allow-anonymous-access* 属性を cn=config エントリーに追加します。

```
# ldapmodify -D "cn=Directory Manager" -W -x

dn: cn=config
changetype: modify
replace: nsslapd-allow-anonymous-access
nsslapd-allow-anonymous-access: off
```

2. サービスを再起動します。

```
# systemctl restart dirsrv.target
```

注記

匿名バインドが無効の場合、ユーザーは RDN を使用してログインできません。これは、ログインのために完全な DN を提供する必要があります。

さらに、匿名バインドを無効にする場合は、認証されていないバインドも自動的に無効になります。

19.11.3. 認証されていないバインドの許可

認証されていないバインドは、ユーザーが空のパスワードを提供する Directory Server への接続です。Directory Server では、デフォルト設定を使用すると、セキュリティー上の理由から、このシナリオのアクセスを拒否します。

```
# ldapsearch -w "" -p 389 -h server.example.com -b "dc=example,dc=com" \
    -s sub -x "(objectclass=*)"

ldap_bind: Server is unwilling to perform (53)
 additional info: Unauthenticated binds are not allowed
```

警告

Red Hat は、認証されていないバインドを有効にしないことを推奨します。この認証方法により、Directory Manager を含むアカウントとしてパスワードを指定せずにユーザーがバインドできます。バインド後、ユーザーはバインドに使用されるアカウントのパーミッションを持つすべてのデータにアクセスできます。

セキュアでない非認証バインドを有効にするには、*nsslapd-allow-unauthenticated-binds* を on に設定します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=config
changetype: modify
replace: nsslapd-allow-unauthenticated-binds
nsslapd-allow-unauthenticated-binds: on
```

19.11.4. 自動バインドの設定

Autobind は、ローカルの UNIX 認証情報に基づいて Directory Server に接続する方法です。これは、ディレクトリー自体に保存されたアイデンティティーにマッピングされます。autobind は、以下の 2 つの部分で設定されます。

autobind を設定する前に、まず LDAPI が有効であることを確認してください（「LDAPI の有効化」で）。次に、（「自動バインドの設定」に）autobind マッピングを設定します。

19.11.4.1. Autobind および LDAPI の概要

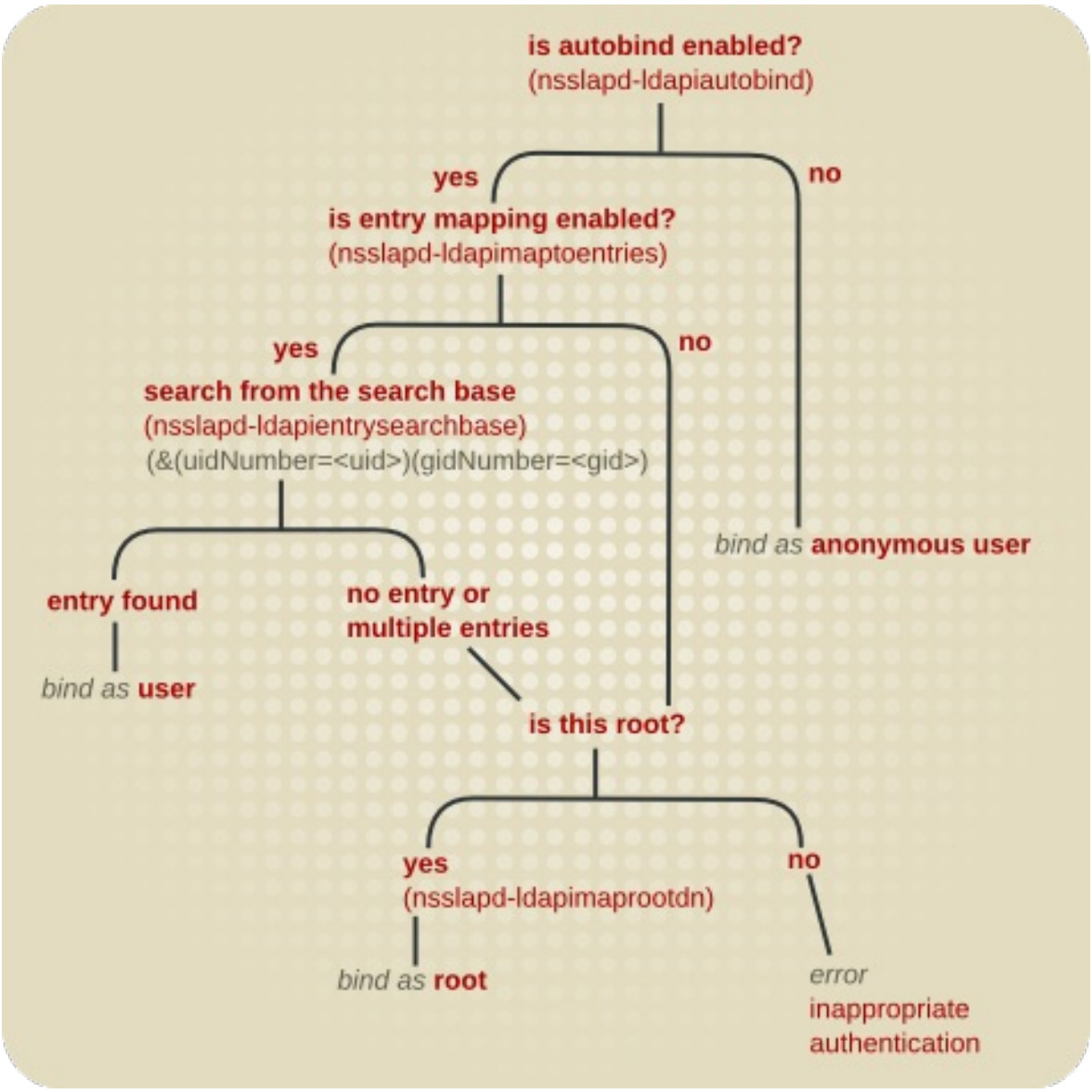
IPC (Inter-process communication) は、Unix マシン上のプロセスやネットワークを区別して相互に直接通信する方法です。LDAPI は、これらの IPC 接続で LDAP 接続を実行する方法です。つまり、LDAP 操作は Unix ソケット上で実行できます。これらの接続は、通常の LDAP 接続よりもはるかに高速で、より安全です。

Directory Server はこの LDAPI 接続を使用して、ユーザーがすぐに Directory Server にバインドしたり、Unix ソケットを介した接続をサポートするツールを使用して Directory Server にアクセスできるようにします。autobind は、Unix ユーザーの uid:gid を使用して、そのユーザーを Directory Server のエントリーにマッピングし、そのユーザーのアクセスを許可します。

autobind では、3 つのディレクトリーエントリーへのマッピングを許可します。

- Unix ユーザーが 1 つのユーザーエントリーに一致した場合はユーザーエントリー
- Unix ユーザーが rootの場合は Directory Manager（または *nsslapd-ldapimaprootdn*で定義されたスーパーユーザー）

図19.1 自動バインド接続パス



特別な自動バインドユーザーのエントリーは、特別な自動バインド接尾辞の下 (一般ユーザーのサブツリー外) にあります。この下のエントリーは、ユーザーおよびグループの ID 番号で識別されます。

gidNumber=*gid***+uidNumber***uid*, *autobindsuffix*

自動バインドが有効になっていないが LDAPI の場合は、他のバインド認証情報を指定しない限り、Unix ユーザーは Directory Server に匿名でバインドされます。

注記

自動バインドを使用すると、バインドユーザー名とパスワードを指定したり、他の SASL 認証メカニズムを使用したりせずに、クライアントが Directory Server に要求を送信できます。LDAP 標準によると、要求でバインド情報が指定されていない場合、サーバーは要求を匿名バインドとして処理します。何らかのバインド情報を必要とする規格に準拠するため、自動バインドを使用するクライアントは SASL/EXTERNAL で要求を送信する必要があります。

SASL の設定に関する詳細は、「SASL Identity マッピングの設定」を参照してください。

19.11.4.2. 自動バインドの設定

自動バインドのみを設定すると、Directory Server への匿名アクセスが可能になります。Unix ユーザーのエントリーへのマッピングを有効にし、root を Directory Manager にマップすることもできます。

1. ldapmodify を実行して Directory Server 設定を更新します。

```
#ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=config
changetype: modify
```

2. autobind を有効にします。

```
replace: nsslapd-ldapiautobind
nsslapd-ldapiautobind: on
```

3. ユーザーエントリーをマッピングするには、以下の 4 つの属性を追加します。

- *nsslapd-ldapimaptoentries* エントリーマッピングを有効にするには、以下を実行します。
- *nsslapd-ldapiuidnumbertype* Directory Server 属性を Unix UID 番号にマッピングするように設定するには、
-

nsslapd-ldapigidnumbertype Unix グループ ID 番号にマップする Directory Server 属性を設定します。

- *nsslapd-ldapientrysearchbase* Directory Server ユーザーエントリーの検索に使用する検索ベースを設定します。

```
add: nsslapd-ldapimaptoentries
nsslapd-ldapimaptoentries: on
-
add: nsslapd-ldapiuidnumbertype
nsslapd-ldapiuidnumbertype: uidNumber
-
add: nsslapd-ldapigidnumbertype
nsslapd-ldapigidnumbertype: gidNumber
-
add: nsslapd-ldapientrysearchbase
nsslapd-ldapientrysearchbase: ou=people,dc=example,dc=com
```

4. root エントリーを Directory Manager にマッピングするには、*nsslapd-ldapimaprootdn* 属性を追加します。

```
add: nsslapd-ldapimaprootdn
nsslapd-ldapimaprootdn: cn=Directory Manager
```

5. サーバーを再起動して、新しい設定を適用します。

```
# systemctl restart dirsrv@instance
```

19.12. パススルー認証の使用

パススルー認証 (PTA) は、ある Red Hat Directory Server インスタンスがバインド要求を認証できるメカニズムです。パススルー認証は PTA プラグインを介して実装されます。有効にすると、Directory Server インスタンスは、ローカルデータベースに保管されないエントリーの単純なバインド操作 (パスワードベース) を受け入れます。

Directory Server は、PTA を使用して、Directory Server の別のインスタンスでユーザーおよび設定ディレクトリーを管理します。

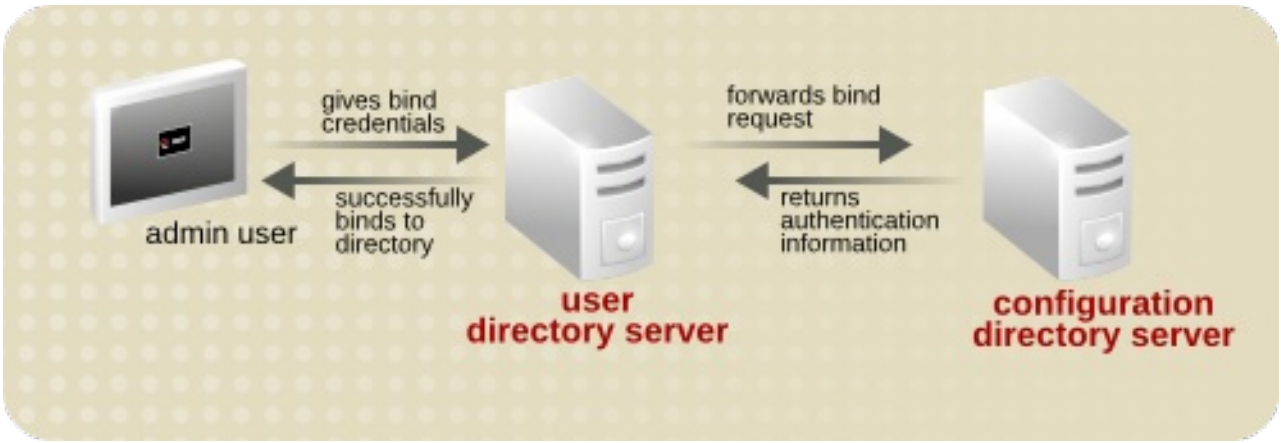
設定ディレクトリーとユーザーディレクトリーが Directory Server の別のインスタンスにインストールされている場合は、設定プログラムは PTA を自動的に設定し、Configuration Administrator ユーザー（通常は admin）が管理役割を実行できるようにします。

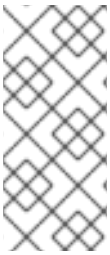
この場合、admin ユーザーエントリーは設定ディレクトリーの o=NetscapeRoot 接尾辞に保存されるため、ここでは PTA が必要です。そのため、通常 admin は失敗するため、ユーザーディレクトリーへのバインドを試みます。PTA により、ユーザーディレクトリーは、認証情報を設定ディレクトリーに送信し、その認証情報を設定ディレクトリーを検証できます。次に、ユーザーディレクトリーにより、admin ユーザーがバインドできるようになります。

この例のユーザーディレクトリーは、別の Directory Server にバインド要求を通過する PTA Directory Server として機能します。設定ディレクトリーは、認証ディレクトリー として機能し、エントリーが含まれるサーバーとして機能し、要求するクライアントのバインド認証情報を検証します。

パススルーサブツリー は、PTA ディレクトリーに 存在しない サブツリーです。ユーザーのバインド DN にこのサブツリーが含まれる場合、ユーザーの認証情報が認証ディレクトリーに渡されます。

図19.2 簡易的な PAM パススルー認証プロセス





注記

PTA プラグインは、同じサーバーインスタンスがユーザーディレクトリーおよび設定ディレクトリーに使用される場合は、Directory Server コンソールには一覧表示されないことがあります。

パススルー認証が機能する仕組みを以下に示します。

1. 設定 Directory Server（認証ディレクトリー）がマシン A にインストールされています。設定ディレクトリーには、設定データベースと接尾辞 o=NetscapeRoot が常に含まれます。この例では、サーバー名は configdir.example.com です。

2. ユーザー Directory Server (PTA ディレクトリー) がマシン B にインストールされます。ユーザーディレクトリーは、dc=example,dc=com などのルート接尾辞を保存します。この例

では、サーバー名は userdir.example.com です。

3.
ユーザーディレクトリーがマシン B に設定されている場合、setup スクリプトはマシン A の設定ディレクトリーの LDAP URL を要求します。

4.
setup プログラムは PTA プラグインを有効にし、設定ディレクトリー LDAP URL を使用するように設定します。

このエントリーには、設定ディレクトリーの LDAP URL が含まれます。以下に例を示します。

```
dn: cn=Pass Through Authentication,cn=plugins,
...
nsslapd-pluginEnabled: on
nsslapd-pluginarg0: ldap://configdir.example.com/o=NetscapeRoot
...
```

ユーザーディレクトリーは、o=NetscapeRoot が含まれる DN を持つエントリーのすべてのバインド要求を設定ディレクトリー configdir.example.com に送信するように設定されるようになりました。

5.
インストールが完了すると、admin ユーザーはユーザーディレクトリーへの接続を試み、ユーザーの追加を開始します。

6.
設定プログラムは、admin ユーザーのエントリーを uid=admin,ou=TopologyManagement,o=NetscapeRoot としてディレクトリーに追加します。そのため、ユーザーディレクトリーは PTA プラグイン設定で定義されている設定ディレクトリーにバインド要求を渡します。

7.
設定ディレクトリーは、ユーザーの認証情報を認証し、情報をユーザーディレクトリーに送信します。

8.
ユーザーディレクトリーを使用すると、admin ユーザーがバインドできるようになります。

19.12.1. PTA プラグインの構文

PTA プラグインの設定情報は、必要な PTA 構文を使用して PTA ディレクトリーの cn=Pass

Through Authentication、cn=plugins,cn=config エントリー (認証ディレクトリーへのバインド要求をパススルーするように設定されたユーザーディレクトリー) に指定されます。このエントリーには大きな属性は 2 つしかありません。

- nsslapd-pluginEnabled: プラグインが有効かどうかを設定します。この属性の値は オンまたはオフに することができます。

- nsslapd-pluginarg 0: 設定ディレクトリーを参照します。この属性の値は、バインド要求を渡すサーバーおよび接尾辞の LDAP URL で、オプションのパラメーターである maxconns、maxops、timeout、ldver、connlifetime、startTLS です。

PTA プラグイン構文の変数コンポーネントは、表19.5「PTA プラグインのパラメーター」に説明されています。

注記

LDAP URL (ldap|ldaps://authDS/subtree) は、1 つの空白で任意のパラメーター (maxconns、maxops、timeout、ldver、connlifetime、startTLS) から分離する必要があります。任意のパラメーターのいずれかが定義される場合は、デフォルト値のみが使用されている場合でも、それらをすべて定義する必要があります。

「複数の認証用 Directory Server の指定」にあるように、*nsslapd-pluginarg* 属性接尾辞を 1 つずつ増やすことで、いくつかの認証ディレクトリーまたはサブツリーを指定できます。以下に例を示します。

```
nsslapd-pluginarg0: LDAP URL for the first server
nsslapd-pluginarg1: LDAP URL for the second server
nsslapd-pluginarg2: LDAP URL for the third server
...
```

任意のパラメーターは、構文で表示される順序で以下の表で説明されています。

表19.5 PTA プラグインのパラメーター

変数	定義
state	プラグインを有効または無効にするかどうかを定義します。使用できる値は on または off です。

変数	定義
ldap|ldaps	2 つの Directory Server 間の通信に TLS を使用するかどうかを定義します。詳細は、「セキュアな接続を使用するようにサーバーを設定」を参照してください。
authDS	認証ディレクトリーのホスト名。Directory Server のポート番号は、コロンとポート番号を追加して指定できます。たとえば、ldap://dirserver.example.com:389/ です。ポート番号が指定されていない場合、PTA サーバーは標準ポートのいずれかを使用して接続を試みます。 URL に ldap:// が指定されている場合のポート 389。 URL に ldaps:// が指定されている場合のポート 636。 詳細は、「認証する Directory Server の指定」を参照してください。
subtree	パススルーサブツリー。PTA Directory Server は、このサブツリーに DN を持つすべてのクライアントから、認証する Directory Server にバインド要求を渡します。詳細は、「パススルーサブツリーの指定」を参照してください。このサブツリーは、このサーバーに存在させることはできません。o=NetscapeRoot のバインド要求を設定ディレクトリーに渡すには、サブツリー o=NetscapeRoot がサーバーに存在しません。
maxconns	任意。PTA ディレクトリーの最大接続数は、認証ディレクトリーに対して同時に開くことができます。デフォルトは 3 です。詳細は、「オプションパラメーターの設定」を参照してください。
maxops	任意。PTA ディレクトリーは単一接続内の認証ディレクトリーに送信できる同時操作の最大数 (通常はバインド要求)。デフォルトは 5 です。詳細は、「オプションパラメーターの設定」を参照してください。
timeout	任意。PTA ディレクトリーが、認証用ディレクトリーサーバーからの応答を待つ時間を秒単位で指定します。このタイムアウトを超えると、サーバーはエラーをクライアントに返します。デフォルトは 300 秒 (5 分) です。ゼロ (0) を指定すると、時間制限をかけないことを示します。詳細は、「オプションパラメーターの設定」を参照してください。
ldver	任意。認証用ディレクトリーへの接続に使用される LDAP プロトコルのバージョン。Directory Server は LDAP バージョン 2 および 3 をサポートします。デフォルトはバージョン 3 です。Red Hat は、古くなり、非推奨になる LDAPv2 を使用 しない ことを強く推奨します。詳細は、「オプションパラメーターの設定」を参照してください。

変数	定義
connlifetime	任意。接続を使用できる制限時間を秒単位で指定します。この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すると、サーバーは接続を閉じ、認証するディレクトリーへの新しい接続を開きます。バインド要求が開始し、ディレクトリーが接続寿命を超えたと判断しない限り、サーバーは接続を閉じません。このオプションを指定しない場合、または 1 つのホストのみが記載されている場合は、接続の有効期間は実行されません。2 つ以上のホストがリストされている場合、デフォルトは 300 秒 (5 分)です。詳細は、「オプションパラメーターの設定」を参照してください。
startTLS	オプション。認証用ディレクトリーへの接続に Start TLS を使用するかどうかを示すフラグ。Start TLS は標準ポート上でセキュアな接続を確立するため、LDAPS の代わりに LDAP を使用して接続するのに便利です。TLS サーバーと CA 証明書の両方がサーバーで使用できる必要があります。 デフォルトは 0 (off) です。Start TLS を有効にするには 、1 に設定します。TLS を使用するには、LDAP URL は ldaps : ではなく ldap: を使用する必要があります。 詳細は、「オプションパラメーターの設定」を参照してください。

19.12.2. PTA プラグインの設定

PTA プラグインを設定する唯一の方法は、エントリー cn=Pass Through Authentication,cn=plugins,cn=config を変更することです。PTA 設定を変更するには、以下を実行します。

1. ldapmodify コマンドを使用して cn=Pass Through Authentication,cn=plugins,cn=config を変更します。
2. Directory Server を再起動します。

PTA プラグインパラメーターを設定する前に、Directory Server に PTA プラグインエントリーが必要です。このエントリーが存在しない場合は、「PTA プラグインの構文」の説明に従って、適切な構文で作成します。

注記

ユーザーと設定ディレクトリーがそのディレクトリーの別のインスタンスにインストールされている場合は、PTA プラグインエントリーがユーザーディレクトリーの設定に自動的に追加され、有効になります。

本セクションでは、以下のセクションでプラグインを設定する方法を説明します。

- 「セキュアな接続を使用するようにサーバーを設定」
- 「認証する Directory Server の指定」
- 「パススルーサブツリーの指定」
- 「オプションパラメーターの設定」

19.12.2.1. セキュアな接続を使用するようにサーバーを設定

PTA ディレクトリーは、PTA ディレクトリーの LDAP URL に LDAPS を指定して、TLS 経由で認証ディレクトリーと通信するように設定できます。以下に例を示します。

```
nsslapd-pluginarg0: ldaps://ldap.example.com:636/o=NetscapeRoot
```

19.12.2.2. 認証する Directory Server の指定

認証用ディレクトリーには、クライアントがバインドしようとしているエントリーのバインド認証情報が含まれます。PTA ディレクトリーは、認証ディレクトリーとして定義されたホストにバインド要求を渡します。認証する Directory Server を指定するには、表19.5「PTA プラグインのパラメーター」で説明するように、PTA ディレクトリーの LDAP URL の authDS を、認証するディレクトリーのホスト名に置き換えます。

1. ldapmodify は PTA プラグインエントリーを編集します。

```
ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=Pass Through Authentication,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginarg0
nsslapd-pluginarg0: ldap://dirserver.example.com/o=NetscapeRoot
```

必要に応じて、ポート番号を含めます。ポート番号が指定されていない場合、PTA Directory Server は ldap:// の標準ポート (389) または ldaps:// のセキュアなポート (636) を使用して接続を試みます。

PTA Directory Server と認証する Directory Server との間の接続が破損するか、接続を開始できない場合は、PTA Directory Server が、指定された次のサーバー (存在する場合) に要求を送信します。最初の Directory Server が利用できない場合にフェイルオーバーを提供するために、必要に応じて認証する複数の Directory Server を指定できます。認証するすべての Directory Server が *nsslapd-pluginarg0* 属性に設定されます。

認証する複数の Directory Server は、以下の形式で、スペース区切りの host:port ペアの一覧で記述されます。

```
ldap|ldaps://host1:port1 host2:port2/subtree
```

2. サービスを再起動します。

```
systemctl restart dirsrv@instance
```

19.12.2.3. パススルーサブツリーの指定

PTA ディレクトリーは、パススルーサブツリーで定義された DN を持つすべてのクライアントからの認証要求を、バインド要求を渡します。サブツリーを指定するには、PTA ディレクトリーの LDAP URL の subtree パラメーターを置き換えて指定します。

パススルーのサブツリーは PTA ディレクトリーに存在すべきではありません。そうすると、PTA ディレクトリーは自分のディレクトリーの内容を使用してバインド要求を解決しようとするため、バインドが失敗します。

1. ldapmodify コマンドを使用して、LDIF ファイルをディレクトリーにインポートします。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
```

```
dn: cn=Pass Through Authentication,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginarg0
nsslapd-pluginarg0: ldap://dirserver.example.com/o=NetscapeRoot
```

この構文の変数コンポーネントの詳細は、表19.5「PTA プラグインのパラメーター」を参照してください。

2. サービスを再起動します。

```
# systemctl restart dirsrv@instance
```

19.12.2.4. オプションパラメーターの設定

PTA 接続の制御に使用する追加のパラメーターは LDAP URL で設定できます。

```
ldap|ldaps://authDS/subtree maxconns, maxops, timeout, ldver, connlifetime, startTLS
```

- PTA Directory Server が認証ディレクトリーに対して同時に開くことができる最大の接続数で、PTA の構文では maxconns で表されます。デフォルト値は 3 です。
- PTA Directory Server が単一の接続内の認証する Directory Server に同時に送信できるバインド要求の最大数。PTA 構文では、このパラメーターは maxops です。デフォルト値は 5 です。
- PTA Directory Server が認証する Directory Server からの応答を待つ時間制限。PTA 構文では、このパラメーターは timeout です。デフォルト値は 300 秒 (5 分) です。
- 認証する Directory Server への接続に使用する PTA Directory Server の LDAP プロトコルのバージョン。PTA 構文では、このパラメーターは ldver です。デフォルトは LDAPv3 です。
- 接続を使用できる制限時間を秒単位で指定します。この時間を過ぎてからクライアントがバインドリクエストを開始すると、そのサーバーは接続を閉じ、認証する Directory Server への新しい接続を開きます。バインド要求が開始し、サーバーがタイムアウトを超えたかどうかを判別しない限り、サーバーは接続を閉じません。このオプションが指定されていない場合や、認証する Directory Server が authDS パラメーターに記載されている場合に限り、時間制限が適用されません。2 つ以上のホストがリストされている場合、デフォルトは 300 秒 (5 分) です。PTA 構文では、このパラメーターは connlifetime になります。

- 接続に Start TLS を使用するかどうか。TLS は、標準の LDAP ポートでセキュアな接続を作成します。Start TLS では、サーバーおよび CA 証明書がインストールされている必要がありますが、TLS で実行する必要はありません。

  デフォルトは 0 で、Start TLS がオフになっていることを意味します。Start TLS を有効にするには 、1 に設定します。TLS を使用するには、LDAP URL は ldaps : ではなく ldap: を使用する必要があります。

1. ldapmodify を使用してプラグインエントリーを編集します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn:  cn=Pass Through Authentication,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginarg0
nsslapd-pluginarg0: ldap://dirserver.example.com/o=NetscapeRoot 3,5,300,3,300,0
```

   (この例では、各オプションのパラメーターはデフォルト値に設定されます。) subtree パラメーターと任意のパラメーターの間にスペースがあることを確認してください。

   注記

   これらのパラメーターは任意ですが、いずれかのパラメーターが定義されている場合は、デフォルト値を使用していてもすべてのパラメーターを定義する必要があります。

2. サービスを再起動します。

```
# systemctl restart dirsrv@instance
```

19.12.3. PTA プラグイン構文の例

本セクションでは、dse.ldif ファイルの PTA プラグイン構文の以下の例を説明します。

- 「Directory Server と 1 つのサブツリーの指定」

- 「複数の認証用 Directory Server の指定」
- 「1 つの Directory Server と複数のサブツリーを指定」
- 「デフォルト以外のパラメーター値の使用」
- 「認証する異なる Directory Server の異なる任意のパラメーターおよびサブツリーの指定」

19.12.3.1. Directory Server と 1 つのサブツリーの指定

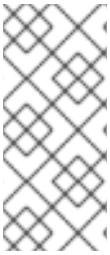
この例では、PTA プラグインを設定して、オプションの変数の全デフォルトを受け入れるように設定します。この設定により、PTA Directory Server は o=NetscapeRoot サブツリーへのすべてのバインド要求に対して、認証する Directory Server に接続します。認証する Directory Server のホスト名は configdir.example.com です。

```
dn: cn=Pass Through Authentication,cn=plugins,cn=config
...
nsslapd-pluginEnabled: on
nsslapd-pluginarg0: ldap://configdir.example.com/o=NetscapeRoot
...
```

19.12.3.2. 複数の認証用 Directory Server の指定

PTA Directory Server と認証する Directory Server との間の接続が破損するか、接続を開始できない場合は、PTA Directory Server が、指定された次のサーバー (存在する場合) に要求を送信します。最初の Directory Server が利用できない場合にフェイルオーバーを提供するために、必要に応じて認証する複数の Directory Server を指定できます。認証するすべての Directory Server が *nsslapd-pluginarg0* 属性に設定されます。認証する複数の Directory Server は、host:port ペアの空白区切りリストに一覧表示されます。以下に例を示します。

```
dn: cn=Pass Through Authentication,cn=plugins,cn=config
...
nsslapd-pluginEnabled: on
nsslapd-pluginarg0: ldap://configdir.example.com:389
config2dir.example.com:1389/o=NetscapeRoot
...
```



注記

nsslapd-pluginarg0 属性は、認証する Directory Server を設定します。追加の *nsslapd-pluginargN* 属性は、使用する PTA プラグインの追加 接尾辞 を設定できますが、追加の ホスト ではありません。

19.12.3.3. 1 つの Directory Server と複数のサブツリーを指定

以下の例では、PTA Directory Server が複数のサブツリーのバインド要求をパススルーするように設定します (パラメーターのデフォルトを使用)。

```
dn: cn=Pass Through Authentication,cn=plugins,cn=config
...
nsslapd-pluginEnabled: on
nsslapd-pluginarg0: ldap://configdir.example.com/o=NetscapeRoot
nsslapd-pluginarg1: ldap://configdir.example.com/dc=example,dc=com
...
```

19.12.3.4. デフォルト以外のパラメーター値の使用

この例では、最大接続数パラメーター maxconns のみに、デフォルト以外の値 (10) を使用しています。その他のパラメーターはデフォルト値に設定されます。ただし、1 つのパラメーターが指定されているため、構文ですべてのパラメーターを明示的に定義する必要があります。

```
dn: cn=Pass Through Authentication,cn=plugins,cn=config
...
nsslapd-pluginEnabled: on
nsslapd-pluginarg0: ldap://configdir.example.com/o=NetscapeRoot 10,5,300,3,300,1
...
```

19.12.3.5. 認証する異なる Directory Server の異なる任意のパラメーターおよびサブツリーの指定

認証する Directory Server ごとに異なるパススルーサブツリーと任意のパラメーター値を指定するには、複数の LDAP URL/任意のパラメーターペアを設定します。LDAP URL/任意のパラメーターペアは、以下のように単一スペースで区切ります。

```
dn: cn=Pass Through Authentication,cn=plugins,cn=config
...
nsslapd-pluginEnabled: on
nsslapd-pluginarg0:ldap://configdir.example.com/o=NetscapeRoot 10,15,30,3,600,0
nsslapd-pluginarg1:ldap://config2dir.example.com/dc=example,dc=com 7,7,300,3,300,1
...
```

19.13. 認証に ACTIVE DIRECTORY 形式のユーザー名の使用

Directory Server に接続する場合は、 uid=*user_name*,ou=People,dc=example,dc=com などのユーザーの識別名 (DN) を指定して認証する必要があります。ただし、DN は記憶しにくくなります。AD DN プラグインを有効にして設定する場合には、DN ではなく *user_name* や *user_name@domain* などの Active Directory 形式のユーザー名を使用できます。

プラグインを有効にし、ユーザーは DN 形式ではないユーザー名を使用してディレクトリーに接続すると、Directory Server はプラグインの設定に基づいて DN を検索します。検索で DN が 1 つ返される場合、Directory Server はこの DN を使用して認証を行います。DN がない場合や複数の DN が返された場合は、認証に失敗します。

注記

コマンドラインで AD DN プラグインのみを有効して設定できます。

example.com をデフォルトドメインとして使用するようにプラグインを有効にして設定するには、以下を行います。

1. cn=addn,cn=plugins,cn=config プラグインエントリーを追加し、デフォルトドメインを設定します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=addn,cn=plugins,cn=config
changetype: add
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: addn
nsslapd-pluginPath: libaddn-plugin
nsslapd-pluginInitfunc: addn_init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled: on
nsslapd-pluginId: addn
nsslapd-pluginVendor: 389 Project
nsslapd-pluginVersion: 1.3.6.0
nsslapd-pluginDescription: Allow AD DN style bind names to LDAP
addn_default_domain: example.com
```

プラグインエントリーで必要な *addn_default_domain* パラメーターにより、デフォルトのドメインが設定されます。認証時に指定されたユーザー名にドメイン名が含まれていない場合、プラグインはこのドメインを追加します。

2. デフォルトドメインの設定エントリーを追加します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=example.com,cn=addn,cn=plugins,cn=config
changetype: add
objectClass: top
objectClass: extensibleObject
cn: example.com
addn_base: ou=People,dc=example,dc=com
addn_filter: (&(objectClass=account)(uid=%s))
```

この例で使用されるパラメーターの詳細は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス』 の説明を参照してください。

警告

デフォルトドメインに、少なくとも設定エントリーを追加する必要があります。エントリーが見つからないと、Directory Server は起動できません。

3. 必要に応じて、前のステップで説明したように、追加のドメイン設定を作成して、複数のドメイン名をサポートすることができます。各ドメイン設定は、異なる検索ベースおよびフィルターを使用できます。

4. Directory Server インスタンスを再起動します。

```
# systemctl restart dirsrv@instance_name
```

19.14. パススルー認証での PAM の使用

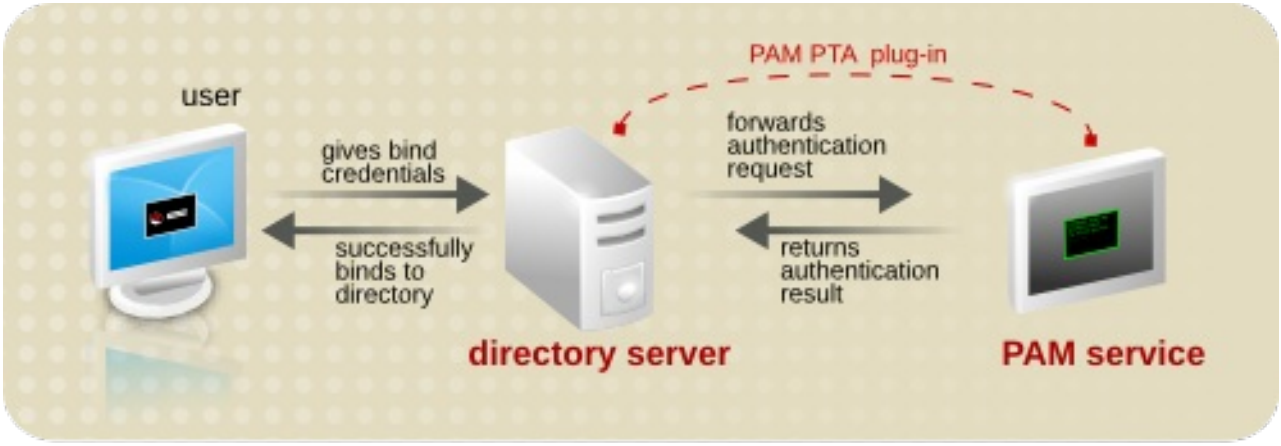
パススルー認証とは、認証要求が 1 つのサーバーから別のサービスに転送されることです。

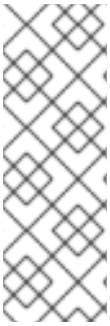
多くのシステムでは、Unix ユーザーおよび Linux ユーザー用の認証メカニズムがすでに含まれています。最も一般的な認証フレームワークの 1 つは、プラグ可能な認証モジュール (PAM) です。既存の認証サービスの多くが利用できるため、管理者はそれらのサービスを引き続き使用したいと思うかもし

れません。PAM モジュールは、Directory Server に対して LDAP クライアントに既存の認証ストアを使用するように指示するように設定できます。

Red Hat Directory Server における PAM パススルー認証は、PAM パススルー認証プラグインを使用します。これにより、Directory Server が PAM サービスと通信して LDAP クライアントを認証できます。

図19.3 PAM パススルー認証プロセス





注記

PAM パススルー認証は、適切なマッピング方法 (ENTRY) が使用されていることを前提に、ユーザーを認証する際にアカウントの無効化と連動します。ただし、PAM パススルー認証では、パスワードは Directory Server ではなくPAM モジュールで設定され保存されるため、グローバルまたはローカルに設定されたパスワードポリシーに照らしてパスワードを検証することはできません。

19.14.1. PAM パススルー認証設定オプション

PAM パススルー認証は、PAM パススルー認証プラグインコンテナーエントリーの下にある子エントリーで構成されます。複数の PAM パススルー認証ポリシーがあり、接尾辞内の異なる接尾辞やエントリーに適用されます。

PAM パススルー用に設定できるエリアはいくつかあります。

- PAM パススルー認証プラグインで制御される接尾辞です。ここでは、除外する接尾辞および含める接尾辞と、欠落した接尾辞の処理方法を説明します。
- 認証設定のターゲットである、設定された接尾辞内の個々のエントリー。デフォルトでは、接尾辞内のすべてのエントリーが認証スコープに含まれますが、複数の異なる PAM パス

スルー認証プラグインインスタンスを設定し、異なるユーザーに異なるプラグイン設定を適用することが可能です。

- PAM 属性マッピング。Directory Server に提示された認証情報は、何らかの方法で LDAP エントリーにマッピングされ、さらに PAM サービスの認証情報に戻される必要があります。これは、マッピングメソッドを定義し、任意で認証情報と一致させるために使用する LDAP 属性を定義します。

- TLS 接続の使用や、使用する PAM サービス、および PAM 認証に失敗した場合の LDAP 認証へのフォールバックなど、一般的な設定。

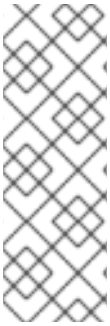
注記

PAM パススルー認証プラグインには、複数の設定インスタンスが存在する場合があります。PAM パススルー認証プラグインのインスタンスは、*pamFilter* 属性を使用して、プラグインで使用する特定のエントリーを検索するよう LDAP フィルターを設定することで、ユーザーエントリーのサブセットに適用できます。

設定可能な属性の一覧は、Red 『Hat Directory Server の設定、コマンド、およびファイルリファレンス』 の 『PAM パススルー認証プラグイン属性』 セクションを参照してください。

19.14.1.1. PAMPTA のターゲットとなるサフィックスの指定

PAM PTA プラグインは、明示的に除外されない限り、デフォルトですべての接尾辞にグローバルに適用されます。接尾辞を除外して組み込むと、ディレクトリーのエリアが LDAP 認証ではなく PAM 認証を使用する場合に役立ちます。



注記

PAM パススルー認証エントリーのターゲットは、任意のサブツリーではなく接尾辞でなければなりません。「接尾辞の作成および維持」 で説明されているように、userRoot に関連する cn=config や userRoot に関連するルート接尾辞 dc=example,dc=com など、特定のバックエンドデータベースに関連付けられるサブツリーです。

pamExcludeSuffix 属性は接尾辞を除外します。デフォルトでは、設定サブツリー (cn=config) のみが除外されます。別の方法では、PAM PTA プラグインは *pamIncludeSuffix* 属性の接尾辞に適用することもできます。これらの属性はいずれも多値で構成されます。

include 属性が設定されている場合、他の接尾辞はすべて自動的に除外されます。同様に、除外属性が設定されている場合、他のすべての接尾辞は自動的に含まれます。

```
pamExcludeSuffix: cn=config
pamExcludeSuffix: o=NetscapeRoot
```

pamIncludeSuffix を使用すると、指定した接尾辞のみが含まれ、その他は自動的に除外されます。この属性は多値であるため、接尾辞を明示的に一覧表示することで、PAM 評価に複数の接尾辞を追加できます。

```
pamIncludeSuffix: ou=Engineering,dc=example,dc=com
pamIncludeSuffix: ou=QE,dc=example,dc=com
```

pamMissingSuffix 属性は、指定された接尾辞 (include または exclude) が存在しない場合に失敗を処理する方法をサーバーに指示します。IGNORE に設定すると、接尾辞が存在しない場合は、プラグインはその接尾辞を省略し、次の試行を試みます。

```
pamMissingSuffix: IGNORE
pamIncludeSuffix: ou=Engineering,dc=example,dc=com
pamIncludeSuffix: ou=Not Real,dc=example,dc=com
```

19.14.1.2. 異なるエントリーへの異なる PAM パススルー認証設定の適用

デフォルトでは、PAM パススルー認証ポリシーは指定の接尾辞内のすべてのエントリーに適用されます。ただし、*pamFilter* 属性で LDAP フィルターを指定することができます。これは、PAM パススルー認証ポリシーを適用する接尾辞内の特定のエントリーを識別します。

これは、複数の PAM パススルー認証ポリシーを使用して、異なる PAM 設定やマッピング方法を異なるユーザータイプに適用する場合に便利です。

19.14.1.3. PAM PTA マッピングの設定

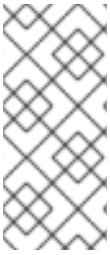
LDAP アイデンティティーを PAM アイデンティティーに接続する方法が必要です。最初に定義するのは、エントリーのマッピングに使用する 方法 です。DN、RDN、および ENTRY の 3 つのオプションがあります。ENTRY はエントリーでユーザー定義の属性を使用します。

複数のマッピング方法を、スペースで区切って順番に並べて指定することができます。プラグインは、認証が成功するまで、またはリストの最後に到達するまで、一覧表示される順序で各マッピングメソッドの使用を試行します。

例えば、このマッピング方法では、まず RDN メソッドをマッピングし、そのメソッドの順に ENTRY、次に DN がマッピングされます。

```
pamIDMapMethod: RDN ENTRY DN
```

異なるマッピングメソッドが、表19.6「PAM 認証のマッピングメソッド」に一覧表示されます。



注記

Directory Server ユーザーアカウントは、ENTRY マッピング方法を使用してのみ検証されます。RDN または DN では、アカウントが非アクティブの Directory Server ユーザーでも、サーバーに正常にバインドされます。

表19.6 PAM 認証のマッピングメソッド

マッピング	説明
RDN	このメソッドは、バインド DN の左端にある RDN から値を使用します。このメソッドのマッピングは、Directory Server で定義されます。指定がない場合は、これがデフォルトのマッピングメソッドになります。
ENTRY	このメソッドは、バインド DN エントリーのユーザー定義の属性から PAM アイデンティティーの値をプルします。identity 属性は *pamIDAttr* 属性で定義されます。例: pamIDAttr: customPamUid
DN	このメソッドは、バインド DN からの完全な識別名を使用します。このメソッドのマッピングは、Directory Server で定義されます。

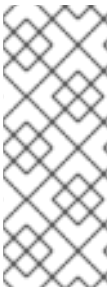
19.14.1.4. 汎用 PAM PTA 設定の設定

PAM 認証には、一般的な 3 つの設定を設定できます。

- PAM (pamService) に送信するサービス名。これは、/etc/pam.d で使用する設定ファイルの名前です。
- セキュアな接続 (pamSecure) を必要とするかどうか。
- PAM 認証に失敗した場合の LDAP 認証にフォールバックするかどうか (pamFallback)

```
pamFallback: false
pamSecure: false
pamService: ldapserver
```

19.14.2. PAM パススルー認証の設定



注記

PAM パススルー認証プラグインには、複数の設定インスタンスが存在する場合があります。PAM パススルー認証プラグインのインスタンスは、*pamFilter* 属性を使用して、プラグインで使用する特定のエントリーを検索するよう LDAP フィルターを設定することで、ユーザーエントリーのサブセットに適用できます。

PAM パススルー認証は、コマンドラインで設定されます。

1. PAM サービスが完全に設定されていることを確認してください。

2. pam_fprintd.so モジュールを PAM 設定ファイルから削除します。

重要

pam_fprintd.so モジュールは、PAM パススルー認証プラグイン設定の *pamService* 属性によって参照される設定ファイルにすることはできません。PAM の fprintd モジュールを使用すると、Directory Server は最大ファイル記述子制限に到達し、Directory Server プロセスが中止する可能性があります。

3. プラグインを有効にします。デフォルトでは無効になっています。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=PAM Pass-Through Auth Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

4. PAM パススルー認証プラグイン設定エントリーを作成します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
```

```
dn: cn=Admin PAM PTA Config,cn=PAM Pass-Through Auth
Plugin,cn=plugins,cn=config
cn: AD PAM PTA Config
```

5. PAM プラグインに使用できる属性を追加します。利用可能な属性は 「PAM パススルー認証設定オプション」 に表示され、例19.2「PAM パススルー認証設定エントリーの例」 にはサンプルエントリーがあります。

6. サーバーを再起動して、新しいプラグイン設定を読み込みます。

```
# systemctl restart dirsrv.target
```

例19.2 PAM パススルー認証設定エントリーの例

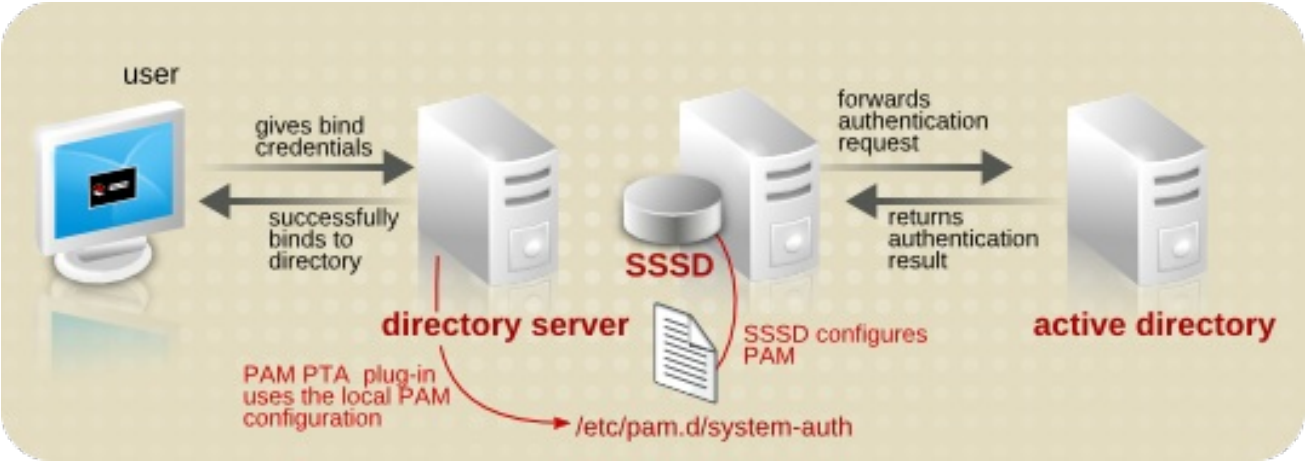
```
dn: cn=Admin PAM PTA Config,cn=PAM Pass Through Auth,cn=plugins,cn=config
objectclass: top
objectclass: pamConfig
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: Admin PAM PTA Config
pamMissingSuffix: ALLOW
pamExcludeSuffix: cn=config
pamExcludeSuffix: o=NetscapeRoot
pamIDMapMethod: RDN ENTRY
pamIDAttr: customPamUid
pamFilter: (manager=uid=bjensen,ou=people,dc=example,dc=com)
pamFallback: FALSE
pamSecure: TRUE
pamService: ldapserver
```

19.14.3. Active Directory をバックエンドとして PAM パススルー認証の使用

PAM パススルー認証では、Directory Server から PAM サービスに認証情報を転送します。1 つのオプションとして、Directory Server 専用の PAM モジュールを設定できます。また、インフラストラクチャーによっては、より再現性が高く便利な方法として、SSSD (System Security Services Daemon) を使用して PAM を設定する方法もあります。SSSD はさまざまなアイデンティティーストアを使用できるため、Active Directory などの認証情報を提供するのに多くの異なるサーバーやサービスを使用できます。

SSSD を介してパススルー認証を使用することはサービスのデイジーチェーンです。PAM PTA プラグインは通常通りに設定されます。使用する特定の PAM サービスファイルを指します。このサービスファイルは SSSD によって管理され、SSSD は複数のプロバイダーであっても必要なアイデンティティープロバイダーに接続するように設定されます。

図19.4 SSSD による PAM パススルー認証



Active Directory で PAM パススルー認証を設定するには、以下を行います。

1. Active Directory サーバーを ID プロバイダーの 1 つとして使用するように SSSD を設定します。

この設定は、『Windows 統合ガイド』 の 『SSSD で Active Directory を ID プロバイダーとして使用』 セクションで説明しています。

2. PAM パススルー認証プラグインを有効にします。これはデフォルトで無効にされています。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=PAM Pass-Through Auth Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

3. PAM パススルー認証プラグイン設定エントリーを作成します。

```
# ldapmodify -a -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=AD PAM PTA Config,cn=PAM Pass-Through Auth Plugin,cn=plugins,cn=config
cn: AD PAM PTA Config
```

4. *pamService* 属性を設定して、SSSD が管理する PAM 設定ファイルを参照します。デフォルトでは、これは /etc/pam.d/system-auth です。

```
pamService: system-auth
```

重要

pam_fprintd.so モジュールは、PAM パススルー認証プラグイン設定の *pamService* 属性によって参照される設定ファイルにすることはできません。PAM の fprintd モジュールを使用すると、Directory Server は最大ファイル記述子制限に到達し、Directory Server プロセスが中止する可能性があります。

5.

ID マップメソッドおよび属性を設定します。Directory Server 環境に応じて、これを行う方法は複数あります。

最も簡単な方法は、RDN マップメソッドを使用して、*uid* 属性（または正しい命名属性）を使用して Directory Server ユーザーを Active Directory ユーザー（Active Directory はアイデンティティープロバイダー）に戻します。

```
pamIDMapMethod: RDN
```

同様に、*samAccountName* 属性を使用して ENTRY マップメソッドで実行できます。Directory Server のユーザーアカウントが、Active Directory のユーザーアカウントの *uid*値と一致する *samAccountName* で作成されると、マッピングは成功します。

```
pamIDMapMethod: ENTRY
pamIDAttr: samAccountName
```

Windows 同期が設定されている場合、ENTRY メソッドは *ntUserDomainId* 属性で使用できます。Directory Server および Active Directory ユーザーアカウントは、その属性値に基づいてすでに同期されているため、PAM マッピングは成功します。

```
pamIDMapMethod: ENTRY
pamIDAttr: ntUserDomainId
```

6.

サーバーを再起動して、新しいプラグイン設定を読み込みます。

```
# systemctl restart dirsrv.target
```

19.15. ユーザーおよびロールの手動による非アクティブ化

1 つのユーザーアカウントまたはアカウントのセットを一時的に非アクティブにできます。アカウン

トが非アクティブになると、ユーザーがディレクトリーにバインドできません。認証操作は失敗します。

ユーザーおよびロールは、動作している属性 *nsAccountLock* を使用して非アクティベートされます。エントリーに値が true の *nsAccountLock* 属性が含まれる場合、サーバーはバインドを拒否します。

同じ手順を使用して、ユーザーとロールを非アクティブ化します。ただし、ロールが非アクティブになると、ロールエントリー自体ではなく ロールのメンバー が非アクティブになります。一般的なロールの詳細、およびロールとアクセスコントロールの関係については、「8章 *エントリーの編成とグループ化*」を参照してください。

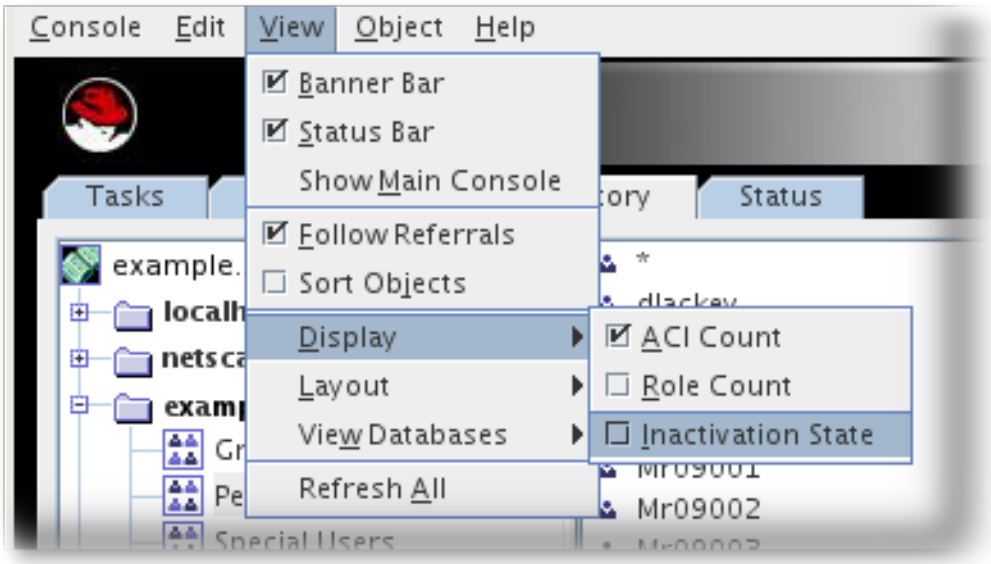
警告

データベースのルートエントリー (ルートまたは部分接尾辞に対応するエントリー) は非アクティブにできません。「3章 *ディレクトリーエントリーの管理*」には、root またはサブ接尾辞のエントリーの作成に関する情報があり、「2章 *ディレクトリーデータベースの設定*」には、root およびサブ接尾辞の作成に関する情報があります。

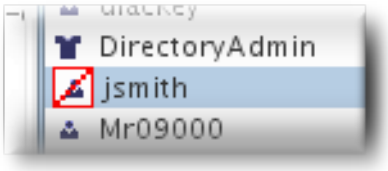
19.15.1. コンソールを使用したアクティブユーザーとロールの表示

1. View メニューを選択し、Display 項目を選択します。

2. 非アクティブ化 の状態 項目を選択します。



非アクティブ化の状態が表示されると、コンソールの右側のペインに、赤色のスラッシュと共にリストされます。



19.15.2. コンソールを使用したユーザーおよびロールのアクティベートおよび非アクティブ化

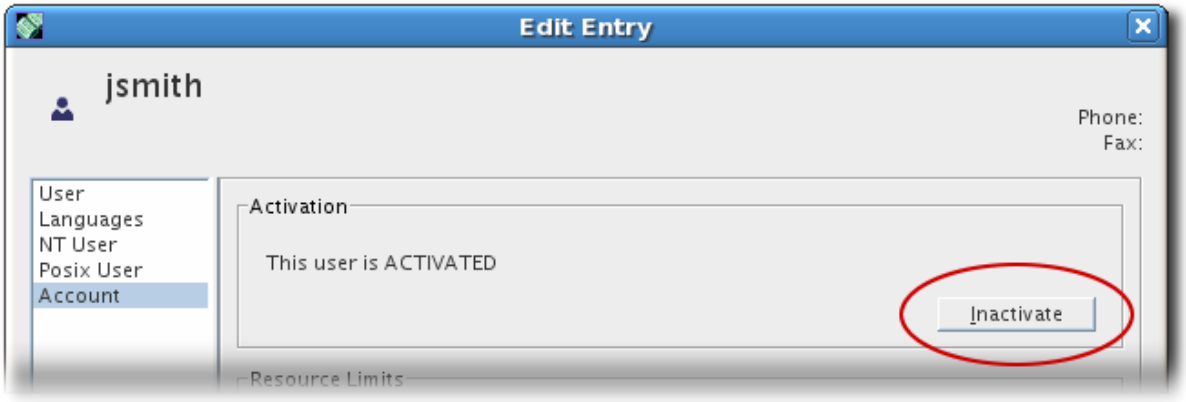
ユーザーとロールのエントリーはすべて、デフォルトでアクティブになります。これらは手動で非アクティブに識別し、非アクティブになっても手動で再アクティブ化する必要があります。

1. Directory タブを選択します。

2. 左側のナビゲーションペインでナビゲーションツリーを参照し、エントリーをダブルクリックします。

   Edit Entry ダイアログボックスが表示されます。

3. 左側のペインで アカウント をクリックします。右側のペインには、ロールまたはユーザーがアクティブになっていると記載されています。非アクティブ ボタンをクリックしてユーザーまたはロール（または アクティベート ボタン ）を非アクティブにして、エントリーを再度有効にします。



4. OK をクリックします。

または、エントリーを強調表示し、Object メニューから Inactivate （または Activate ）を選択します。

19.15.3. コマンドラインを使用したアクティブユーザーおよびロールの表示

ns-accountstatus.pl スクリプトは、アクティブユーザーおよび非アクティブユーザーに関する詳細情報を取得するために使用されます。

単一ユーザーのアカウントステータスを取得するには、以下のようにコマンドを使用できます。

```
# ns-accountstatus.pl -D "cn=Directory Manager" -w password -I
"uid=jsmith,ou=people,dc=example,dc=com"
uid=bjensen,ou=people,dc=example,dc=com  activated.
```

-V オプションを追加して、より詳細な出力を取得します。

```
# ns-accountstatus.pl -D "cn=Directory Manager" -w password -I
"uid=jsmith,ou=people,dc=example,dc=com"
Entry:                  uid=jsmith,ou=People,dc=example,dc=com
Entry Creation Date:    20160204153140Z (02/04/2016 10:31:40)
Entry Modification Date: 20160205163904Z (02/05/2016 11:39:04)
Last Login Date:        20160205163905Z (02/05/2016 11:39:05)
Inactivity Limit:       2592000 seconds (30 days)
Time Until Inactive:    2591688 seconds (29 days, 23 hours, 54 minutes, 48 seconds)
Time Since Inactive:    -
Entry State:            activated
```

上記は、出力の最後の 3 行で表されるアクティブなアカウントの例です。代わりに、非アクティブアカウントは以下のような出力を提供します。

```
# ns-accountstatus.pl -D "cn=Directory Manager" -w password -I
"uid=jsmith,ou=people,dc=example,dc=com"
Entry:                  uid=jsmith,ou=people,dc=example,dc=com
Entry Creation Date:    20160204153140Z (02/04/2016 10:31:40)
Entry Modification Date: 20160204160545Z (02/04/2016 11:05:45)
Last Login Date:        20160204160546Z (01/04/2016 11:05:46)
Inactivity Limit:       2592000 seconds (30 days)
Time Until Inactive:    -
Time Since Inactivated: 85877 seconds (23 hours, 51 minutes, 17 seconds)
Entry State:            inactivated (inactivity limit exceeded)
```

**-I オプションを使用してアカウントを指定する代わりに、- b (search a database suffix)、- f（フィルターを使用）、- s （検索範囲）オプションを使用して検索を作成できます。さらに、- i オプション（非アクティブアカウントのみを返す）または -g *X オプション（次の X 秒で有効期限が切れるアカウントのみ）を使用して検索を改良できます。以下に例を示します。*

```
# ns-accountstatus.pl -D "cn=Directory Manager" -w password -b "ou=people,dc=example,dc=com" -f
"(uid=*)" -V -g 86400
Entry:                  uid=jsmith,ou=people,dc=example,dc=com
Entry Creation Date:    20160204153140Z (02/04/2016 10:31:40)
Entry Modification Date: 20160205163904Z (02/05/2016 11:39:04)
Last Login Date:        20160205163905Z (01/05/2016 11:39:05)
Inactivity Limit:       2592000 seconds (30 days)
Time Until Inactive:    979 seconds (16 minutes, 19 seconds)
Time Since Inactive:    -
Entry State:            activated
```

出力の最後の 3 行から分かるように、このアカウントは現在アクティブですが、まもなく有効期限が切れます。

19.15.4. コマンドラインを使用したユーザーおよびロールの非アクティブ化およびアクティブ化

***Directory Server** は、デュアルスクリプトを使用して、コマンドラインからエントリーを非アクティブまたはアクティブ化します。**ns-inactivate.pl** スクリプトおよび **ns-activate.pl** スクリプトは、*『**Red Hat Directory Server** の設定、コマンド、およびファイルリファレンス に記載されている、変更するエントリーを識別するのと同様のオプションを共有します。』*

たとえば、ユーザーアカウントを非アクティブにするには、次のコマンドを実行します。

```
[root@server ~]# ns-inactivate.pl -Z instance_name -D Directory Manager -w secret -p 389 -h
example.com  -I "uid=jfrasier,ou=people,dc=example,dc=com"
```

次に、アカウントを再度アクティブにできます。

```
# ns-activate.pl -Z instance_name -D Directory Manager -w secret -p 389 -h example.com  -I
"uid=jfrasier,ou=people,dc=example,dc=com"
```

第20章 サーバーおよびデータベースアクティビティーの監視

本章では、データベースおよび Red Hat Directory Server ログの監視を説明します。SNMP を使用して Directory Server を監視する方法は、21章SNMP を使用した Directory Server の監視を参照してください。

20.1. DIRECTORY SERVER ログファイルの種類

Directory Server は以下のログタイプを提供します。

- アクセスログ: クライアント接続および Directory Server インスタンスへの接続試行に関する情報が含まれます。このログタイプは、デフォルトで有効になります。
- エラーログ: エラーや、通常の操作中のディレクトリーエクスペリエンスに関する詳細なエラーメッセージが含まれます。このログタイプは、デフォルトで有効になります。

警告

Directory Server がエラーログに書き込みに失敗した場合、サーバーはエラーメッセージを Syslog サービスに送信し、終了します。このログタイプは、デフォルトで有効になります。

- 監査ログ: 各データベースとサーバー設定に加えられた変更を録画します。このログはデフォルトでは有効になっていません。
- 監査失敗ログ: レコードで監査ログが失敗しました。このログはデフォルトでは有効になっていません。

20.2. ログファイルの表示

コマンドラインおよび Directory Server コンソールを使用して Directory Server ログファイルを表示できます。

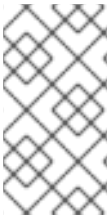
20.2.1. コマンドラインでログファイルの表示

コマンドラインを使用してログファイルを表示するには、less、more、cat などの、Red Hat Enterprise Linux に含まれるユーティリティーを使用します。以下に例を示します。

```
# less /var/log/dirsrv/slapd-instance_name/access
```

ログファイルの場所を表示するには、以下を実行します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 \
    -h server.example.com -x -b "cn=config" -s base \
    nsslapd-accesslog nsslapd-errorlog nsslapd-auditlog nsslapd-auditfaillog
...
nsslapd-accesslog: /var/log/dirsrv/slapd-instance_name/access
nsslapd-errorlog: /var/log/dirsrv/slapd-instance_name/errors
nsslapd-auditlog: /var/log/dirsrv/slapd-instance_name/audit
nsslapd-auditfaillog: /var/log/dirsrv/slapd-instance_name/audit-failure
```



注記

ログタイプのロギングが有効になっていない場合は、対応するログファイルが存在しません。

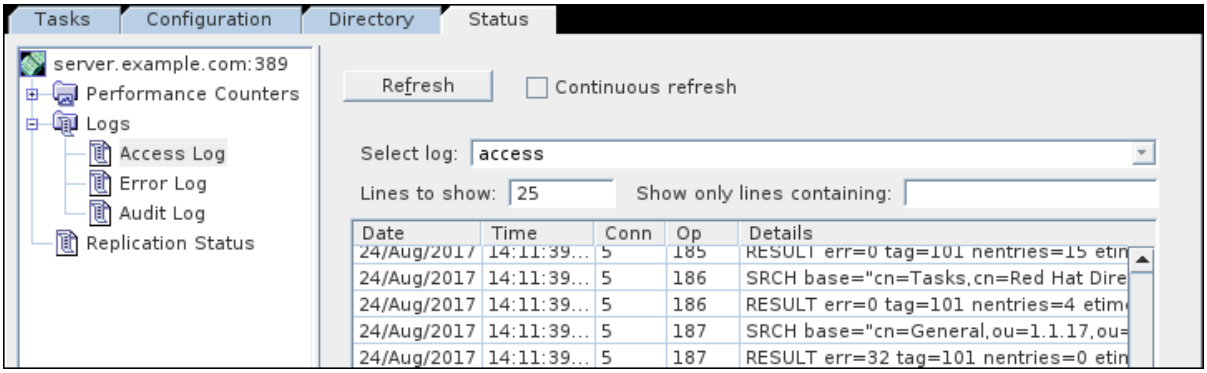
20.2.2. コンソールを使用したログファイルの表示

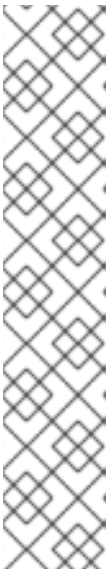
Directory Server ログファイルを表示するには、次のコマンドを実行します。

1. Directory Server コンソールを開きます。詳細は、「Directory Server コンソールを開く」 を参照してください。

2. Status タブを選択します。

3. Logs エントリーを展開し、表示するログを選択します。





注記

コンソールには、Audit Fail ログのログファイルビューアーが含まれていません。または、以下を行うことができます。

- *コマンドラインでこのログを表示します。「コマンドラインでログファイルの表示」 を参照してください。*
- *Audit Fail エントリーを Audit イベントと同じ ファイルに記録するように Directory Server を設定します。*

4. *必要に応じて、以下の設定をログファイルビューアーに適用することができます。*

   - *表示するフィールドに行数 を設定します。*
   - ***Show only lines with the field**（フィールドを含む）にフィルターを設定します。*
   - ***Select log** フィールドでログを選択して、同じタイプの古いログファイルを表示します。*
   - ***Continuous refresh** を選択して、新しいログエントリーを自動的に表示できるようにします。*

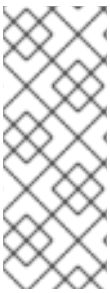
   ***Refresh** ボタンをクリックして変更を適用します。*

20.3. ログファイルの設定

すべてのタイプのログファイルについて、ログの 作成 ポリシーおよびログ削除 ポリシーを設定する必要があります。ログ作成ポリシーは、新規ログファイルの起動時に設定され、古いログファイルが削除される際にログ削除ポリシーが設定されます。

20.3.1. ログの有効化または無効化

アクセスおよびエラーロギングはデフォルトで有効になっています。ただし、監査および監査の失敗ロギングはデフォルトで無効になっています。



注記

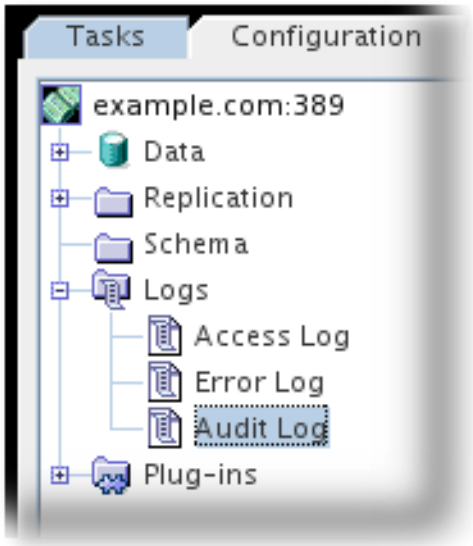
アクセスログを無効にすると、ディレクトリーへの 2000 回のアクセスごとに約 1 メガバイトのログファイルが増加するため、一部のシナリオで有用です。ただし、アクセスログをオフにする前に、この情報で問題のトラブルシューティングを行うことができます。

Directory Server コンソールでのロギングの有効化または無効化

1. Directory Server コンソールにログインします。

2. Configuration タブを選択します。

3. ナビゲーションツリーで Logs フォルダーを展開し、ログのフォルダーを選択して有効または無効にします。



4. ロギングを有効または無効にするには、Enable Logging チェックボックスを選択します。

5. ログが有効な場合は、提供されるフィールドに Directory Server がログインするために使用する完全パスおよびファイル名を入力します。デフォルトのパスは /var/log/dirsrv/slapd-instance/log_type です（例： /var/log/dirsrv/slapd-instance/access ）。

6. Save をクリックします。

コマンドラインを使用したロギングの有効化または無効化

ldapmodify ユーティリティーを使用して、Directory Server のロギング機能を制御する cn=config サブツリーのパラメーターを変更できます。

- アクセスログ: nsslapd-accesslog-logging-enabled
- エラーログ: nsslapd-errorlog-logging-enabled
- 監査ログ: nsslapd-auditlog-logging-enabled
- 監査ログの失敗ログ: nsslapd-auditfaillog-logging-enabled

詳細は、Red 『Hat Directory Server の設定、コマンド、およびファイルリファレンス の該当するセクションを参照してください』。

たとえば、監査ロギングを有効にするには以下を入力します。

```
# ldapmodify -D "cn=Directory Manager" -W -x
dn: cn=config
changetype: modify
replace: nsslapd-auditlog-logging-enabled
nsslapd-auditlog-logging-enabled: on
```

20.3.2. プラグイン固有のロギングの設定

デバッグのために、プラグインが実行する操作についてアクセスおよび監査ロギングを有効にでき

ます。詳細は、Red 『Hat Directory Server の設定、コマンド、およびファイルリファレンス の対応するセクションにある nsslapd-logAccess および nsslapd-logAudit パラメーターを参照してください』。

20.3.3. 高解像度のログタイムスタンプの無効化

Directory Server は、デフォルト設定を使用して、ナノ秒の精度でエントリーをログに記録します。

```
[27/May/2016:17:52:04.754335904 -0500] schemareload - Schema validation passed.
[27/May/2016:17:52:04.894255328 -0500] schemareload - Schema reload task finished.
```

高解像度のログタイムスタンプを無効にするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -x
dn: cn=config
changetype: modify
replace: nsslapd-logging-hr-timestamps-enabled
nsslapd-logging-hr-timestamps-enabled: off
```

注記

高解像度のログのタイムスタンプを無効にするオプションは非推奨で、将来のリリースで削除される予定です。

高解像度のログのタイムスタンプを無効にすると、Directory Server は秒単位の精度でしかログを記録しなくなります。

```
[27/May/2016:17:52:04 -0500] schemareload - Schema validation passed.
[27/May/2016:17:52:04 -0500] schemareload - Schema reload task finished.
```

20.3.4. ログファイルのローテーションポリシーの定義

現在のログファイルを定期的にアーカイブして新しいファイルを作成するには、ログファイルのローテーションポリシーを設定します。Directory Server コンソールまたはコマンドラインを使用して、cn=config サブツリーの設定を更新できます。

以下の設定パラメーターを設定して、ログファイルのローテーションポリシーを制御できます。

アクセスモード

アクセスモードでは、新規に作成されたログファイルにファイル権限を設定します。

- アクセスログ: nsslapd-accesslog-mode
- エラーログ: nsslapd-errorlog-mode
- 監査ログ: nsslapd-auditlog-mode
- 監査ログの失敗ログ: nsslapd-auditfaillog-mode

ログの最大数

保持するログファイルの最大数を設定します。ファイル数に達すると、Directory Server は新しいログファイルを作成する前に、最も古いログファイルを削除します。

- アクセスログ: nsslapd-accesslog-maxlogsperdir
- エラーログ: nsslapd-errorlog-maxlogsperdir
- 監査ログ: nsslapd-auditlog-maxlogsperdir
- 監査ログの失敗ログ: nsslapd-auditfaillog-maxlogsperdir

各ログのファイルサイズ

ログファイルがローテーションされるまでの最大サイズをメガバイト単位で設定します。

- アクセスログ: nsslapd-accesslog-maxlogsize
- エラーログ: nsslapd-errorlog-maxlogsize

- *監査ログ: nsslapd-auditlog-maxlogsize*

- *監査ログの失敗ログ: nsslapd-auditfaillog-maxlogsize*

毎回ログの作成

ログファイルの最大期間を設定します。

- *nsslapd-accesslog-logrotationtime および nsslapd-accesslog-logrotationtimeunit*

- *nsslapd-errorlog-logrotationtime および nsslapd-errorlog-logrotationtimeunit*

- *nsslapd-auditlog-logrotationtime および nsslapd-auditlog-logrotationtimeunit*

- *nsslapd-auditfaillog-logrotationtime および nsslapd-auditfaillog-logrotationtimeunit*

さらに、以下のパラメーターを使用してログファイルがローテーションされるまでの時間を設定することもできます。

- *nsslapd-accesslog-logrotationsynchour および nsslapd-accesslog-logrotationsyncmin*

- *nsslapd-errorlog-logrotationsynchour および nsslapd-errorlog-logrotationsyncmin*

- *nsslapd-auditlog-logrotationsynchour および nsslapd-auditlog-logrotationsyncmin*

- *nsslapd-auditfaillog-logrotationsynchour および nsslapd-auditfaillog-logrotationsyncmin*

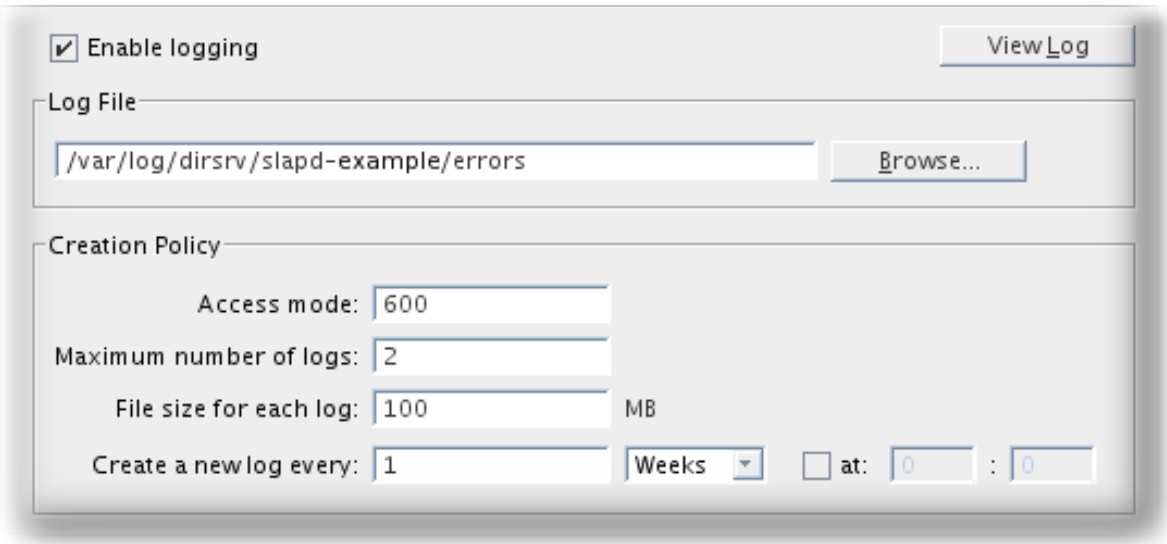
詳細は、Red 『Hat Directory Server の設定、コマンド、およびファイルリファレンスの該当するセクションを参照してください』。

各ログファイルは、ログファイルのアーカイブまたは交換を容易にするため、サーバーのバージョン、ホスト名、およびポートを識別するタイトルで始まります。以下に例を示します。

```
389-Directory/1.3.5.10 B2016.257.1817
server.example.com:389 (/etc/dirsrv/slapd-instance)
```

Directory Server コンソールでログファイルローテーションの設定

1. Directory Server コンソールにログインします。
2. Configuration タブを選択します。
3. ナビゲーションツリーで、Logs フォルダーを展開し、設定を更新するログのフォルダーを選択します。
4. 作成ポリシー エリアでロギング設定を設定します。以下に例を示します。



5. Save をクリックします。

コマンドラインを使用したログファイルローテーションの設定

ldapmodify ユーティリティーを使用して、Directory Server のロギング機能を制御するパラメーターを変更できます。たとえば、エラーログの場合は、アクセスモード 600 を設定して最大 2 を維持し、ログファイルのサイズを 100 MB または 5 日 ごとにローテーションするには、以下を実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -x
dn: cn=config
changetype: modify
replace: nsslapd-errorlog-mode
nsslapd-errorlog-mode: 600
-
replace: nsslapd-errorlog-maxlogsperdir
nsslapd-errorlog-maxlogsperdir: 2
-
replace: nsslapd-errorlog-maxlogsize
nsslapd-errorlog-maxlogsize: 100
-
replace: nsslapd-errorlog-logrotationtime
nsslapd-errorlog-logrotationtime: 5
-
replace: nsslapd-errorlog-logrotationtimeunit
nsslapd-errorlog-logrotationtimeunit: day
```

20.3.5. ログファイルの削除ポリシーの定義

Directory Server は、Deletion Policy を設定すると、アーカイブされた古いログファイルを自動的に削除します。

注記

ログファイルのローテーションポリシーが設定されている場合に限り、ログファイルの削除ポリシーを設定できます。Directory Server は、ログローテーション時に削除ポリシーを適用します。

以下の設定パラメーターを設定して、ログファイルの削除ポリシーを制御できます。

ログサイズの合計

すべてのアクセス、エラー、監査、または監査失敗ログファイルのサイズが設定された値を越えると、最も古いログファイルが自動的に削除されます。

- アクセスログ: nsslapd-accesslog-logmaxdiskspace

- *エラーログ: nsslapd-errorlog-logmaxdiskspace*
- *監査ログ: nsslapd-auditlog-logmaxdiskspace*
- *監査ログ: nsslapd-auditfaillog-logmaxdiskspace*

空きディスク領域が「より少ない」

空きディスク容量がこの値に達すると、最も古いアーカイブファイルが自動的に削除されます。

- *アクセスログ: nsslapd-accesslog-logminfreediskspace*
- *エラーログ: nsslapd-errorlog-logminfreediskspace*
- *監査ログ: nsslapd-auditlog-logminfreediskspace*
- *監査ログ: nsslapd-auditfaillog-logminfreediskspace*

指定した時間よりもファイルが古い場合

ログファイルが設定された時間よりも古い場合は、これが自動的に削除されます。

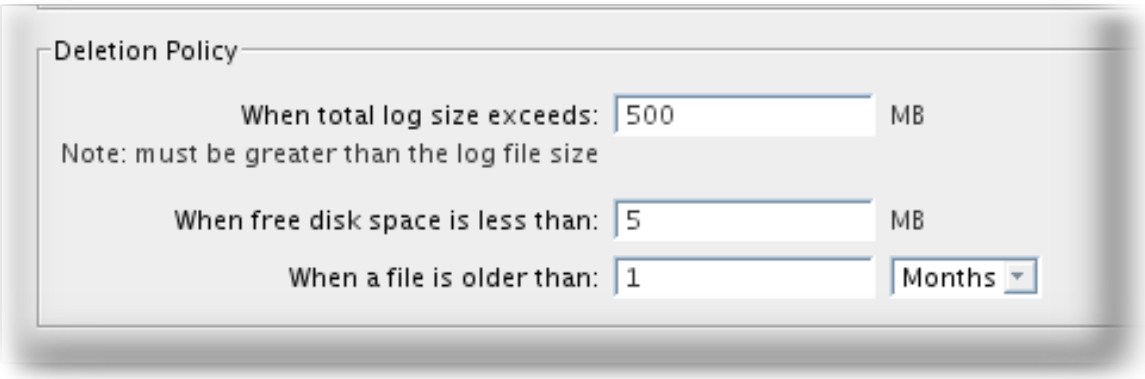
- *アクセスログ: nsslapd-accesslog-logexpirationtime および nsslapd-accesslog-logexpirationtimeunit*
- *エラーログ: nsslapd-errorlog-logminfreediskspace および nsslapd-errorlog-logexpirationtimeunit*
- *監査ログ: nsslapd-auditlog-logminfreediskspace および nsslapd-auditlog-logexpirationtimeunit*

- *監査ログ: nsslapd-auditfaillog-logminfreediskspace および nsslapd-auditfaillog-logexpirationtimeunit*

詳細は、Red 『Hat Directory Server の設定、コマンド、およびファイルリファレンス の該当するセクションを参照してください』。

Directory Server コンソールでのログ削除ポリシーの設定

1. *Directory Server コンソールにログインします。*
2. *Configuration タブを選択します。*
3. *ナビゲーションツリーで、Logs フォルダーを展開し、設定を更新するログのフォルダーを選択します。*
4. *Deletion Policy エリアでロギング設定を設定します。以下に例を示します。*



5. *Save をクリックします。*

コマンドラインを使用したログ削除ポリシーの設定

ldapmodify ユーティリティーを使用すると、Directory Server のロギング機能を制御するパラメーターが変更されます。たとえば、すべてのアクセスログファイルの合計サイズが 500 MB 増加した場合に、最も古いアクセスログファイルを自動的に削除するには、次のように実行します。

```
# ldapmodify -D "cn=Directory Manager" -W -x
dn: cn=config
```

```
changetype: modify
replace: nsslapd-accesslog-logmaxdiskspace
nsslapd-accesslog-logmaxdiskspace: 500
```

20.3.6. 手動ログファイルローテーション

Directory Server は、3 つのすべてのログの自動ログファイルのローテーションをサポートします。ただし、自動ログファイルの作成や削除ポリシーが設定されていない場合には、ログファイルを手動でローテーションすることができます。デフォルトでは、アクセス、エラー、監査、および監査失敗のログファイルは、以下の場所にあります。

```
/var/log/dirsrv/slapd-instance
```

ログファイルを手動でローテーションするには、以下を実行します。

1. サーバーをシャットダウンします。

```
# systemctl stop dirsrv.target instance
```

2. 古いログファイルが今後の参照で利用できるように、ローテーションされるログファイルを移動するか名前を変更します。

3. サービスを再起動します。

```
# systemctl restart dirsrv.target instance
```

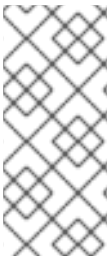
20.3.7. ログレベルの設定

アクセスとエラーログはいずれも、設定されるログレベルに応じて、さまざまな情報を記録できます。

以下の設定パラメーターを設定して、以下のログレベルを制御できます。

- アクセスログ: **nsslapd-accesslog-level**
- エラーログ: **nsslapd-errorlog-level**

詳細情報およびサポートされるログレベルの一覧は、Red 『Hat Directory Server の設定、コマンド、およびファイルリファレンス の該当するセクションを参照してください』。

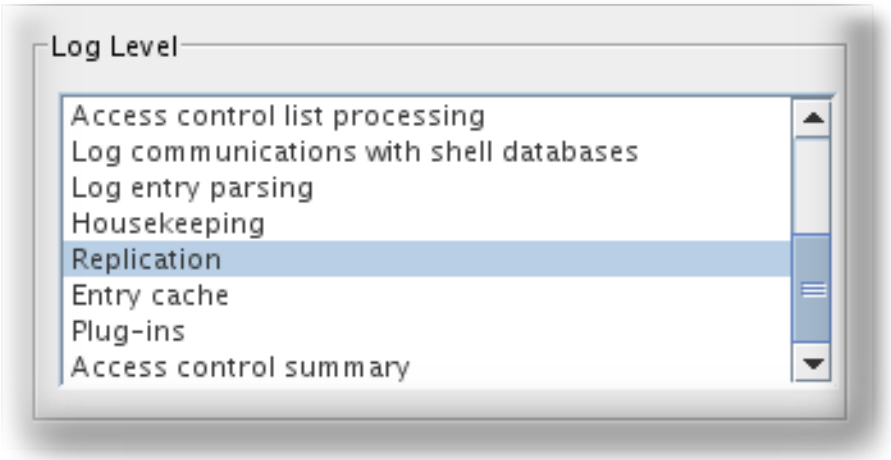


注記

デフォルトからログレベルを変更すると、ログファイルが急速に増大する可能性があります。Red Hat は、Red Hat のテクニカルサポートからの要請がない限り、デフォルト値を変更しないことを推奨します。

Directory Server コンソールでのログレベルの設定

1. Directory Server コンソールにログインします。
2. Configuration タブを選択します。
3. ナビゲーションツリーで、Logs フォルダーを展開し、設定を更新するログのフォルダーを選択します。
4. Log Level エリアにログレベルを設定します。たとえば、エラーログファイルの場合、



5. Save をクリックします。

コマンドラインを使用したログレベルの設定

ldapmodify ユーティリティーを使用してログレベルを設定できます。たとえば、検索フィルターロギング (32) および設定ファイル処理 (64) を有効にするには、nsslapd-errorlog-level パラメーターを

96 (32 + 64) に設定します。

```
# ldapmodify -D "cn=Directory Manager" -W -x
dn: cn=config
changetype: modify
replace: nsslapd-errorlog-level
nsslapd-errorlog-level: 96
```

20.4. アクセスログ統計の取得

logconv.pl スクリプトはアクセスログを解析し、サーバーで実行するさまざまなユーザーおよび操作に関するサマリー情報を返します。

最も簡単な方法は、スクリプトはアクセスログ (単数または複数) を解析するだけです。

```
# logconv.pl /relative/path/to/accessLog
```

このスクリプトはワイルドカードを受け入れて複数のアクセスログを解析できます。これはログローテーションが使用される場合に役立ちます。

```
# logconv.pl /var/log/dirsrv/slapd-instance/access*
```

logconv.pl のさまざまなオプションは、man ページと 『設定、コマンド、およびファイルリファレンス』 で説明されています。

logconv.pl は、アクセスログから一般的な使用状況を引き出すために、いくつかの異なる方法があります。

最も簡単な方法としては、logconv.pl が、総操作数、総接続数、各操作タイプごとのカウント、永続的な検索などの拡張操作のカウント、およびバインド情報のリストを表示します。

```
# logconv.pl /var/log/dirsrv/slapd-instance/access
Access Log Analyzer 8.2
Command: logconv.pl /var/log/dirsrv/slapd-instance/access
Processing 1 Access Log(s)...

[001] /var/log/dirsrv/slapd-instance/access size (bytes):       77532

Total Log Lines Analysed:  527

Start of Logs:    14/Oct/2017:16:15:22.452909568
```

```
End of Logs:     14/Oct/2017:16:39:50.157790196

Processed Log Time:  0 Hours, 24 Minutes, 27.704877056 Seconds

Restarts:                 10
Secure Protocol Versions:
  - TLS1.2 client bound as uid=user_name,ou=people,o=example.com (11 connections)
  - TLS1.2 128-bit AES; client CN=CA Subsystem,O=example.com; issuer CN=Certificate
Authority,O=example.com (11 connections)
  - TLS1.2 128-bit AES-GCM (2 connections)
  - TLS1.2 128-bit AES (3 connections)

Peak Concurrent Connections:  38
Total Operations:           4771
Total Results:              4653
Overall Performance:        97.5%

Total Connections:          249         (0.17/sec)  (10.18/min)
 - LDAP Connections:        107         (0.07/sec)  (4.37/min)
 - LDAPI Connections:       128         (0.09/sec)  (5.23/min)
 - LDAPS Connections:       14          (0.01/sec)  (0.57/min)
 - StartTLS Extended Ops:   2           (0.00/sec)  (0.08/min)

Searches:                  2963        (2.02/sec)  (121.13/min)
Modifications:             649         (0.44/sec)  (26.53/min)
Adds:                      785         (0.53/sec)  (32.09/min)
Deletes:                   10          (0.01/sec)  (0.41/min)
Mod RDNs:                  6           (0.00/sec)  (0.25/min)
Compares:                  0           (0.00/sec)  (0.00/min)
Binds:                     324         (0.22/sec)  (13.25/min)

Proxied Auth Operations:     0
Persistent Searches:         17
Internal Operations:         0
Entry Operations:            0
Extended Operations:         4
Abandoned Requests:          0
Smart Referrals Received:    0

VLV Operations:              30
VLV Unindexed Searches:      0
VLV Unindexed Components:    20
SORT Operations:             22

Entire Search Base Queries:  12
Paged Searches:              2
Unindexed Searches:          0
Unindexed Components:        149

FDs Taken:                   249
FDs Returned:                212
Highest FD Taken:            107

Broken Pipes:                0
Connections Reset By Peer:   0
Resource Unavailable:        0
```

```
Max BER Size Exceeded:        0

Binds:                        324
Unbinds:                      155
---------------------------------
 - LDAP v2 Binds:             41
 - LDAP v3 Binds:             180
 - AUTOBINDs(LDAPI):          103
 - SSL Client Binds:          0
 - Failed SSL Client Binds:   0
 - SASL Binds:                134
   - EXTERNAL: 114
   - GSSAPI: 20
 - Directory Manager Binds:   10
 - Anonymous Binds:           1

Cleaning up temp files...
Done.
```

操作と接続のサマリー情報に加えて、サーバーへのすべての接続のより詳細なサマリー情報を提供します。この情報には、サーバーへの接続に使用された最も一般的な **IP** アドレス、ログインに最も失敗した **DN**、サーバーへのアクセスに使用された合計バインド **DN** 数、最も一般的なエラーコードやリターンコードなどが含まれます。

その他の接続サマリーは単一オプションとして渡されます。たとえば、サーバー (**b**) への接続に使用する **DN** 数と、サーバーによって返された合計接続コード (**c**) を **-bc** として渡します。

```
# logconv.pl -bc /var/log/dirsrv/slapd-instance/access
...
----- Total Connection Codes -----

U1              3    Cleanly Closed Connections
B1              1    Bad Ber Tag Encountered

----- Top 20 Bind DN's -----

Number of Unique Bind DN's: 212

1801            cn=Directory Manager
1297            Anonymous Binds
311             uid=jsmith,ou=people...
87              uid=bjensen,ou=peopl...
85              uid=mreynolds,ou=peo...
69              uid=jrockford,ou=peo...
55              uid=sspencer,ou=peop...
...
```

データは、特定の開始時間 (**-S**) 以降、特定の終了時間 (**-E**) 以降、または範囲内からエントリーに制限することができます。開始時間と終了時間が設定されると、**logconv.pl** が最初に指定の時間範囲を出力し、次にその期間の概要を出力します。

```
# logconv.pl -S "[01/Jul/2016:16:11:47.000000000 -0400]" -E "[01/Jul/2016:17:23:08.999999999 -0400]" /var/log/dirsrv/slapd-instance/access
...
----------- Access Log Output ------------

Start of Logs:    01/Jul/2016:16:11:47
End of Logs:      01/Jul/2016:17:23:08
...
```

開始時間と終了の期間は、合計サマリー数の生成に使用されるデータの時間制限のみを設定します。それでも、集計または合計の数が表示されます。**Directory Server** への接続や操作のパターンを把握するために、**1** 分ごと **(-M)** や **1** 秒ごと **(-m)** のカウントデータを出力することができます。この場合、データは時間単位で、指定した **CSV** 出力ファイルに出力されます。

```
# logconv.pl -m|-M outputFile accessLogFile
```

以下に例を示します。

```
# logconv.pl -M /home/output/statsPerMin.txt /var/log/dirsrv/slapd-instance/access*
```

-M|-m オプションは、**-S** 引数および **-E** 引数と共に使用することで、特定の期間内の分単位または秒単位のカウントを取得することもできます。

ファイルの各行は、分または秒の **1** つの時間単位を表し、その期間の合計数を示しています。**CSV** ファイル **(1** 分ごとの統計および **1** 秒ごと統計の両方**)** には、以下のコラムが順番に含まれます。

```
Time,time_t,Results,Search,Add,Mod,Modrdn,Delete,Abandon,Connections,SSL Conns,Bind,Anon Bind,Unbind,Unindexed
```

CSV ファイルは、**LibreOffice Calc** などのスプレッドシートプログラムや、その他多くのビジネスアプリケーションで操作できます。**CSV** データをインポートし、チャートまたは他のメトリクスを生成する手順は、アプリケーション自体によって異なります。

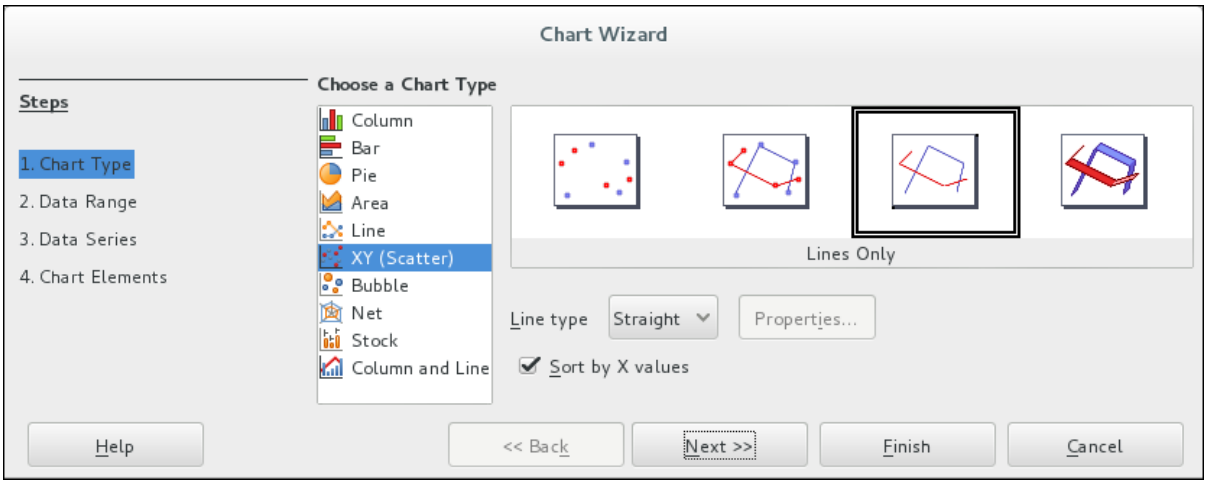
たとえば、**LibreOffice Calc** でチャートを作成するには、次のコマンドを実行します。

1. **CSV** ファイルを開きます。
2. **Insert** メニューをクリックし、**Chart** を選択します。

3. *Chart Type エリアで、チャートタイプを XY (Scatter) に設定します。*

   a. *サブタイプを行のみに設定します。*

   b. *X 値でソートするオプションを選択します。*



4. *他の画面のデフォルト (特に、データ系列を列で使用すること、最初の行と最初の列をラベルとして設定すること) を受け入れて、チャートを作成します。*

20.5. シャットダウンのローカルディスクの監視

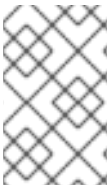
システムで利用可能なディスク領域が小さすぎると、Directory Server プロセス(slapd)がクラッシュします。突然シャットダウンを行うと、データベースが破損したり、ディレクトリーデータが失われるリスクが発生します。

slapd プロセスが使用できるディスク領域を監視できます。ディスク監視スレッドは、nsslapd-disk-monitoring 設定属性を使用して有効になります。これにより、特定の領域で利用可能なディスク領域をチェックするために 10 秒ごとにウェイクする監視スレッドが作成されます。

ディスク領域が定義されたしきい値に近づくと、slapd は一連の手順（デフォルト）を開始し、それが使用するディスク領域の量を削減します。

- *詳細なロギングは無効になっています。*

- アクセスロギングおよびエラーロギングは無効になっています。
- ローテーション（アーカイブ）ログが削除されます。



注記

ログ設定に他の変更が加えられても、常にエラーログメッセージが記録されます。

利用可能なディスク領域が設定済みのしきい値の半分にドロップされ続けると、**slapd** は正常なシャットダウンプロセス（猶予期間内）を開始し、利用可能なディスク領域が 4KB だと、**slapd** プロセスがすぐにシャットダウンします。ディスク領域が解放されると、シャットダウンプロセスが中止され、以前に無効にしたすべてのログ設定が再有効になります。

デフォルトでは、監視スレッドは設定、トランザクションログ、およびデータベースディレクトリーを確認します。ディスク領域を評価する際に、追加の属性(**nsslapd-disk-monitoring-logging-critical**)を設定してログディレクトリーを追加できます。

ディスクの監視はデフォルトで無効にされていますが、適切な設定属性を **cn=config** エントリーに追加することで有効にし、設定できます。表20.1「ディスクモニタリングの設定属性」 すべての設定オプションを一覧表示します。

1. **ldapmodify** を使用して、ディスク監視属性を追加します。少なくとも、**nsslapd-disk-monitoring** 属性をオンにしてディスクの監視を有効にします。デフォルトのしきい値は 2MB です。これは、**nsslapd-disk-monitoring-threshold** 属性で設定できます（任意）。

   以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -x
dn: cn=config
changetype: modify
add: nsslapd-disk-monitoring
nsslapd-disk-monitoring: on
-
add: nsslapd-disk-monitoring-threshold
nsslapd-disk-monitoring-threshold: 3000000
-
add: nsslapd-disk-monitoring-grace-period
nsslapd-disk-monitoring-grace-period: 20
```

2. *Directory Server を再起動して、新しい構成を読み込みます。*

```
[root@server ~]# systemctl restart dirsrv.target
```

表20.1 ディスクモニタリングの設定属性

設定属性	詳細
nsslapd-disk-monitoring	有効にするディスクモニタリング。他の設定オプションで使用できるデフォルト値があるため、これは唯一の必須属性です。
nsslapd-disk-monitoring-grace-period	ディスク領域の制限の半分に達するとサーバーをシャットダウンする前に待機する猶予期間を設定します。これにより、状況に対応するための管理者の時間が提供されます。デフォルト値は 60（分）です。
nsslapd-disk-monitoring-logging-critical	ログディレクトリーがディスク領域の制限に設定された半方向ポイントをパスした場合に、サーバーをシャットダウンするかどうかを設定します。これにより、監視スレッドが監査ロギングを無効にしたり、ローテーションされたログファイルを削除したりできなくなります。
nsslapd-disk-monitoring-threshold	サーバーに十分なディスク領域があるかどうかを評価するために使用するディスク容量（バイト単位）を設定します。スペースがこのしきい値の半分になると、サーバーはシャットダウンプロセスを開始します。デフォルト値は 2000000(2MB)です。

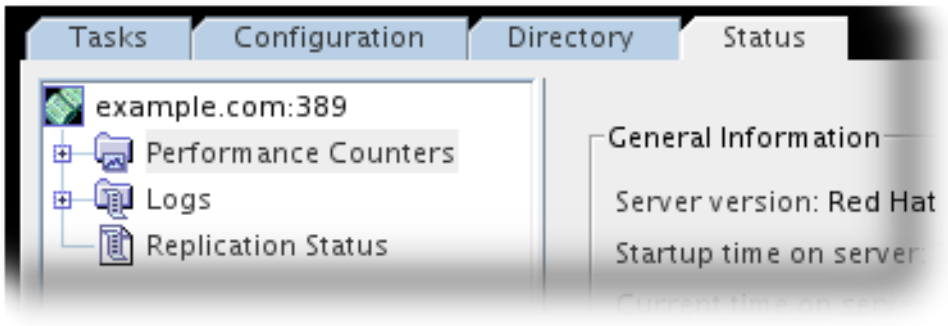
20.6. サーバーアクティビティーの監視

Directory Server の現在のアクティビティーは、Directory Server コンソールまたはコマンドラインから監視できます。また、すべてのデータベースのキャッシュアクティビティーを監視することもできます。

20.6.1. Directory Server コンソールからのサーバーの監視

1. *Status タブを選択します。*

2. *ナビゲーションツリーで Performance Counters を選択します。*



右側のペインの **Status** タブには、サーバーアクティビティーに関する現在の情報が表示されます。サーバーが現在実行していない場合は、このタブではパフォーマンスの監視情報は提供されません。

3.

Refresh をクリックし、現在の表示を更新します。サーバーが表示する情報を継続的に更新するには、**Continuous** チェックボックスを選択します。

General Information の表には、サーバーの基本情報が記載されています。これは、収集された統計に関するベースラインを設定するのに役立ちます。

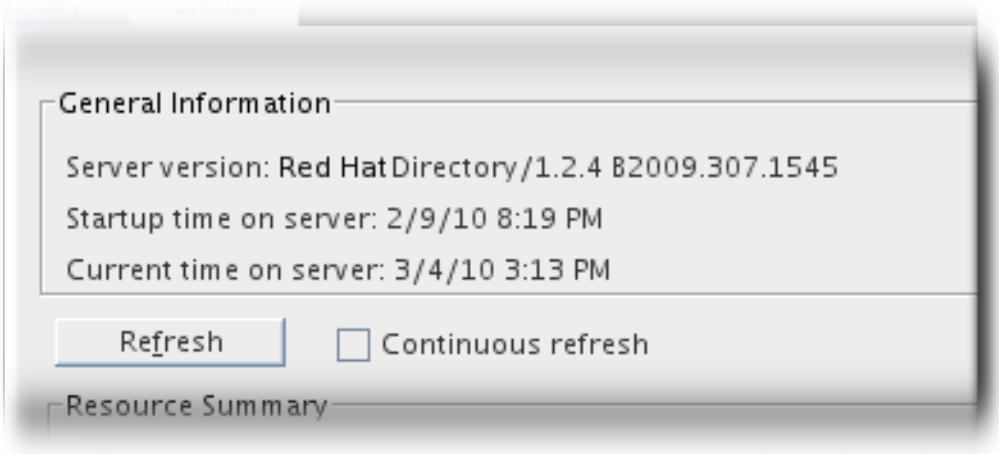
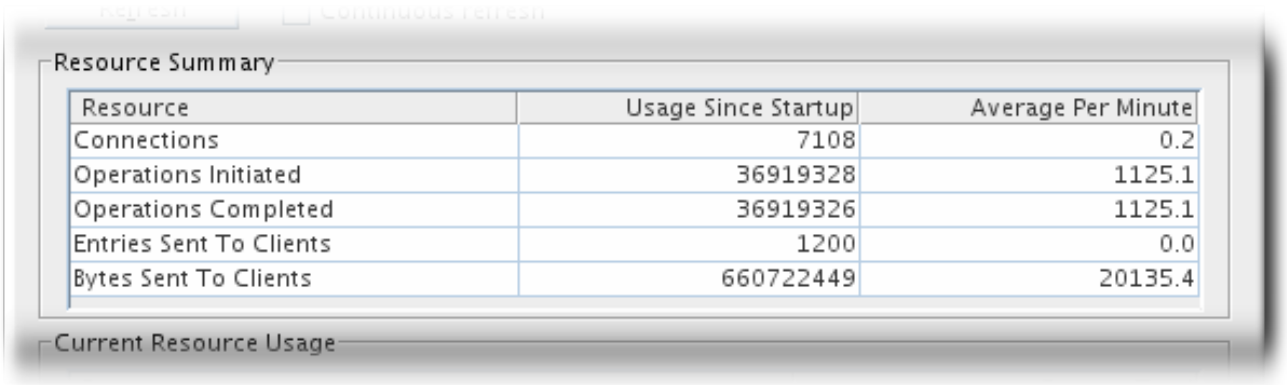


表20.2 一般情報（サーバー）

フィールド	詳細
サーバーバージョン	現在のサーバーバージョンを識別します。
サーバー上の起動時間	サーバーが開始した日時。
サーバーの現在の時間	サーバーの現在の日時。

リソースサマリー の表には、そのインスタンスによって実行されるすべての操作の合計が表示されます。


Resource Summary

Resource	Usage Since Startup	Average Per Minute
Connections	7108	0.2
Operations Initiated	36919328	1125.1
Operations Completed	36919326	1125.1
Entries Sent To Clients	1200	0.0
Bytes Sent To Clients	660722449	20135.4

Current Resource Usage

表20.3 リソースの概要

リソース	起動時の使用方法	1 分あたりの平均数
接続	サーバーの起動以降、このサーバーへの接続の合計数。	サーバーの起動から 1 分あたりの平均接続数
操作開始	サーバーの起動以降に開始された操作の合計数。操作には、検索、追加、変更などのサーバーアクションのクライアント要求が含まれます。多くの場合、接続ごとに複数の操作が開始されます。	サーバーの起動から 1 分あたりの操作の平均。
完了した操作	サーバーの起動以降、サーバーによって完了した操作の合計数。	サーバーの起動から 1 分あたりの操作の平均。
クライアントに送信されるエントリー	サーバーの起動以降にクライアントに送信されるエントリーの合計数。エントリーは、検索要求の結果でクライアントに送信されます。	サーバーの起動時に 1 分あたりにクライアントに送信されるエントリーの平均数。
クライアントに送信されるバイト	サーバーの起動以降にクライアントに送信される合計バイト数。	サーバーの起動時に 1 分あたりにクライアントに送信される平均のバイト数。

現在のリソース使用状況 の表には、サーバー上の現在の要求が表示されます。

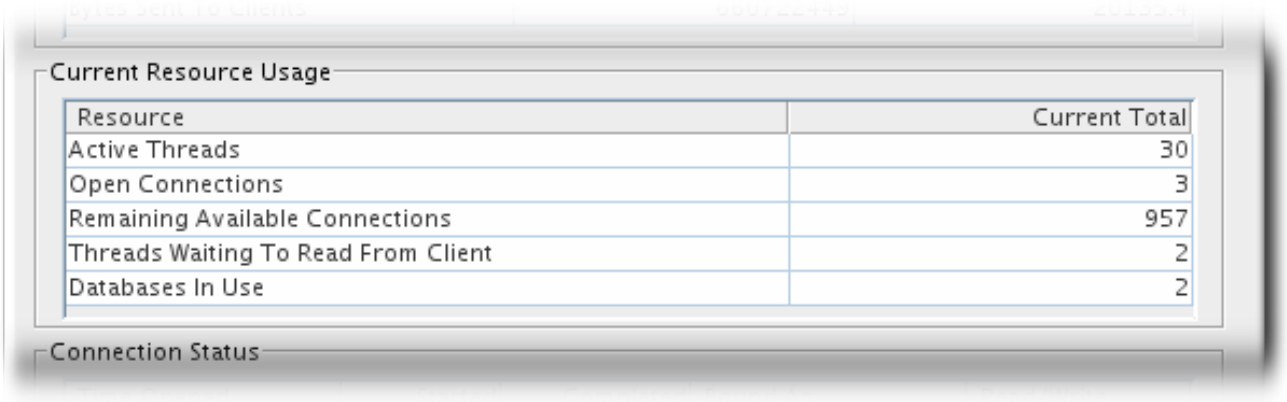


表20.4 現在のリソース使用

リソース	*現在の合計*
アクティブなスレッド	*リクエストの処理に使用される現在のアクティブなスレッドの数。追加のスレッドは、レプリケーションやチェーンなどの内部サーバータスクで作成できます。*
開いている接続	*オープン接続の合計数。各接続は複数の操作のために考慮できるため、複数のスレッドがあります。*
残りの利用可能な接続	*サーバーが同時に開くことのできる残りの接続の合計数。この数は、現在開いている接続の数と、サーバーが開くことのできる同時接続の合計数に基づいています。多くの場合、後者の値はオペレーティングシステムによって決定され、タスクで利用可能なファイル記述子の数で表示されます。*
クライアントへの書き込みを待機するスレッド	*クライアントへの書き込みを待機するスレッドの合計数。スレッドは、クライアントにデータ送信中にサーバーを一時停止する必要がある場合にすぐに書き込みできない場合があります。一時停止の理由には、低速なネットワーク、低速なクライアント、またはクライアントに送信される非常に多くの情報などがあります。*
クライアントからの読み取りを待機するスレッド	*クライアントから読み取りを待機するスレッドの合計数。サーバーがクライアントから要求を受信していった場合、スレッドはすぐに読み取らない場合があり、一部の理由でその要求の送信は停止します。通常、読み取りを待機するスレッドは、低速なネットワークまたはクライアントを示します。*

リソース	現在の合計
使用中のデータベース	サーバーによってサービスされるデータベースの合計数。

Connection Status テーブルには、現在のアクティブな接続が表示されます。これには、関連する接続情報が含まれます。

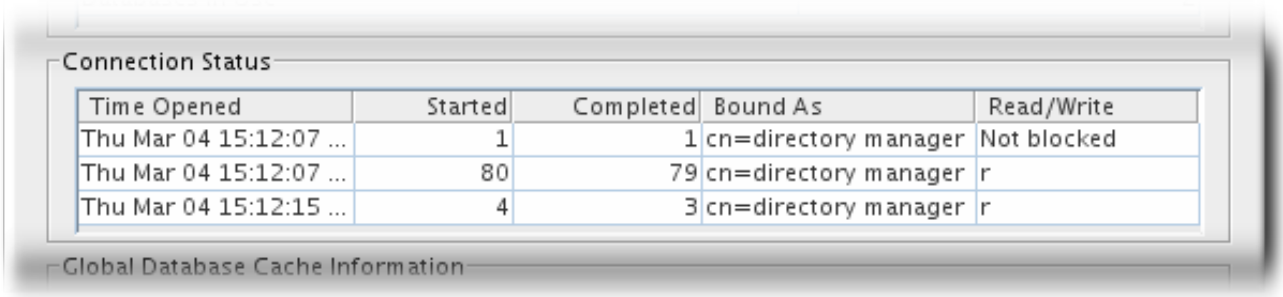
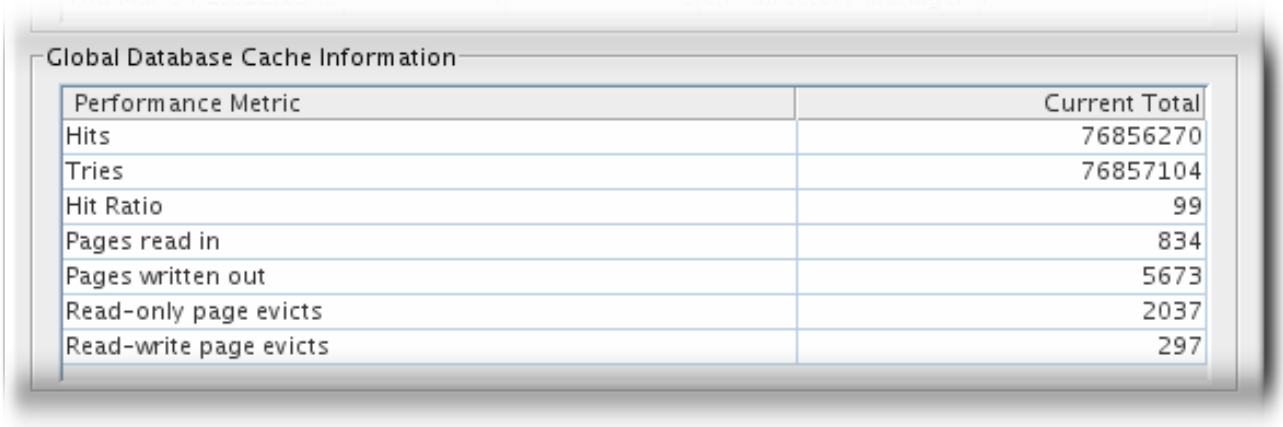


表20.5 接続の状態

テーブルヘッダー	詳細
開かれた時間	接続が最初に開かれたサーバー上の時間。
Started	このコネクションによって開始される操作の数。
完了	この接続のためにサーバーが完了した操作の数。
バインド：	サーバーにバインドするためにクライアントによって使用される識別名。クライアントがサーバーに対して認証されていない場合、サーバーはこのフィールドで バインドされません。

テーブルヘッダー	詳細
読み取り/書き込み	サーバーが現在クライアントへの読み取りまたは書き込みアクセスに対してブロックされているかどうかを示します。以下の 2 つの値を使用できます。 ブロックされないということは、サーバーがアイドル状態であること、データをクライアントにアクティブに送信したり、クライアントからデータをアクティブに読み込むことを意味します。 ブロック とは、サーバーがクライアントへのデータの送信またはクライアントからのデータの読み取りを試みるが、できないことを意味します。おそらく、ネットワークまたはクライアントが遅くなる可能性があります。

Global Database Cache テーブルには、Directory Server インスタンス内のすべてのデータベースのキャッシュ情報が記載されています。


Global Database Cache Information

Performance Metric	Current Total
Hits	76856270
Tries	76857104
Hit Ratio	99
Pages read in	834
Pages written out	5673
Read-only page evicts	2037
Read-write page evicts	297

注記

グローバルデータベースキャッシュのパフォーマンスカウンターは、Directory Server コンソールの他のサーバーパフォーマンスカウンターと共に一覧表示されますが、実際のデータベースキャッシュエントリーは、cn =monitor,cn= database_instance,cn=ldbm database,cn=plugins,cn=config に置かれ、その他のデータベースアクティビティーであるためです。コマンドラインでこれらのエントリーを監視する方法については、「コマンドラインでのデータベースの監視」 を参照してください。

表20.6 グローバルデータベースのキャッシュ情報

テーブルヘッダー	詳細
Hits	ディスクに移動するのではなく、キャッシュからデータを取得することで、サーバーがリクエストを処理できる回数。
tries	サーバー起動後のデータベースアクセスの総数。
Ratio のヒット	キャッシュの比率は、キャッシュヒットの成功を試みます。この数を 100% にすると、より良い数値になります。
ページの読み取り	ディスクからキャッシュに読み取られるページ数。
ページが書き込まれる	キャッシュからディスクに書き込まれたページ数。
読み取り専用ページのエビクション	新規ページのスペースを作成するために、キャッシュから破棄された読み取り専用ページの数。キャッシュから破棄されたページはディスクに書き込まれ、サーバーのパフォーマンスに影響する可能性があります。ページの数が小さいほどエビクトされます。
読み取り/書き込みページのエビクション	新規ページのスペースを作成するために、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。この値は、変更されていない読み取り書き込みページを破棄する点で、Pages Written Out とは異なります。キャッシュから破棄されたページはディスクに書き込まれ、サーバーのパフォーマンスに影響する可能性があります。ページのエビクト数が少ないほど、パフォーマンスが向上します。

20.6.2. コマンドラインでの Directory Server の監視

Directory Server の現在のアクティビティーは、以下の特性を持つ ldapsearch などの LDAP ツールを使用して監視できます。

- 属性 filterobjectclass =* で検索します。
- 検索ベースの cn=monitor を使用します。サーバーの監視属性は cn=monitor エントリーにあります。
- 検索範囲 ベース を使用します。

以下に例を示します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -s base -b
"cn=monitor" "(objectclass=*)"
```

Directory Server の監視属性は cn=monitor エントリーにあります。Directory Server の検索に関する情報は、「ldapsearch の使用」 を参照してください。

表20.7 サーバーモニタリングの属性

属性	説明
バージョン	ディレクトリーの現在のバージョン番号を指定します。
スレッド	リクエストの処理に使用される現在のアクティブなスレッドの数。追加のスレッドは、レプリケーションやチェーンなどの内部サーバータスクで作成できます。

属性	説明
connection:fd:opentime:opsinitiated:opscompleted:binddn:[rw]	*オープン接続ごとに以下の概要情報を提供します（Directory Manager としてディレクトリーにバインドする場合にのみ利用可能）。* **fd***: この接続に使用するファイル記述子。* **opentime -** *接続が開かれた時間。* **opsinitiated***: この接続によって開始される操作の数。* **opscompleted***: 完了した操作の数* **bindDN***: ディレクトリーに接続するためにこの接続によって使用される識別名。* **rw***: 接続が読み取りまたは書き込みのためにブロックされると表示されるフィールド。* *デフォルトでは、この情報は Directory Manager で利用できます。ただし、この情報に関連付けられている ACI は、他の情報にアクセスできるように編集できます。*
currentconnections	*ディレクトリーで現在サービスにある接続の数を特定します。*
totalconnections	*起動してからディレクトリーによって処理される接続の数を特定します。*
dtablesize	*は、ディレクトリーで利用可能なファイル記述子の数を表示します。各接続には、オープンインデックスごとに 1 つのファイル記述子、ログファイル管理用のファイル、および ns-slapd 自体に 1 つ必要です。基本的に、この値は、ディレクトリーが提供できる追加の同時接続の数を示します。ファイル記述子の詳細は、オペレーティングシステムのドキュメントを参照してください。*
readwaiters	*クライアントからデータの読み取りを待機するスレッドの数を特定します。*

属性	説明
opsinitiated	起動後にサーバーが開始した操作の数を特定します。
opscompleted	起動してからサーバーが完了した操作の数を特定します。
entriessent	サーバー起動以降にクライアントに送信されるエントリーの数を特定します。
bytesSent	サーバーが起動してからクライアントに送信されたバイト数を特定します。
currenttime	サーバーのこのスナップショットが作成された時間を指定します。この時間は Greenwich Mean Time(GMT)で UTC 形式で表示されます。
rhncfg	サーバーが起動した時間を指定します。この時間は Greenwich Mean Time(GMT)で UTC 形式で表示されます。
nbackends	サーバーサービスのバックエンド（データベース）の数を指定します。
backendmonitordn	各ディレクトリーデータベースの DN を識別します。

20.7. データベースアクティビティーの監視

データベースの現在のアクティビティーは、Directory Server Console またはコマンドラインから監視できます。

注記

エントリーおよびデータベースキャッシュを調整してサーバーパフォーマンスを向上させるためのヒントは、『『Red Hat Directory Server パフォーマンスチューニングガイド』を参照してください』。

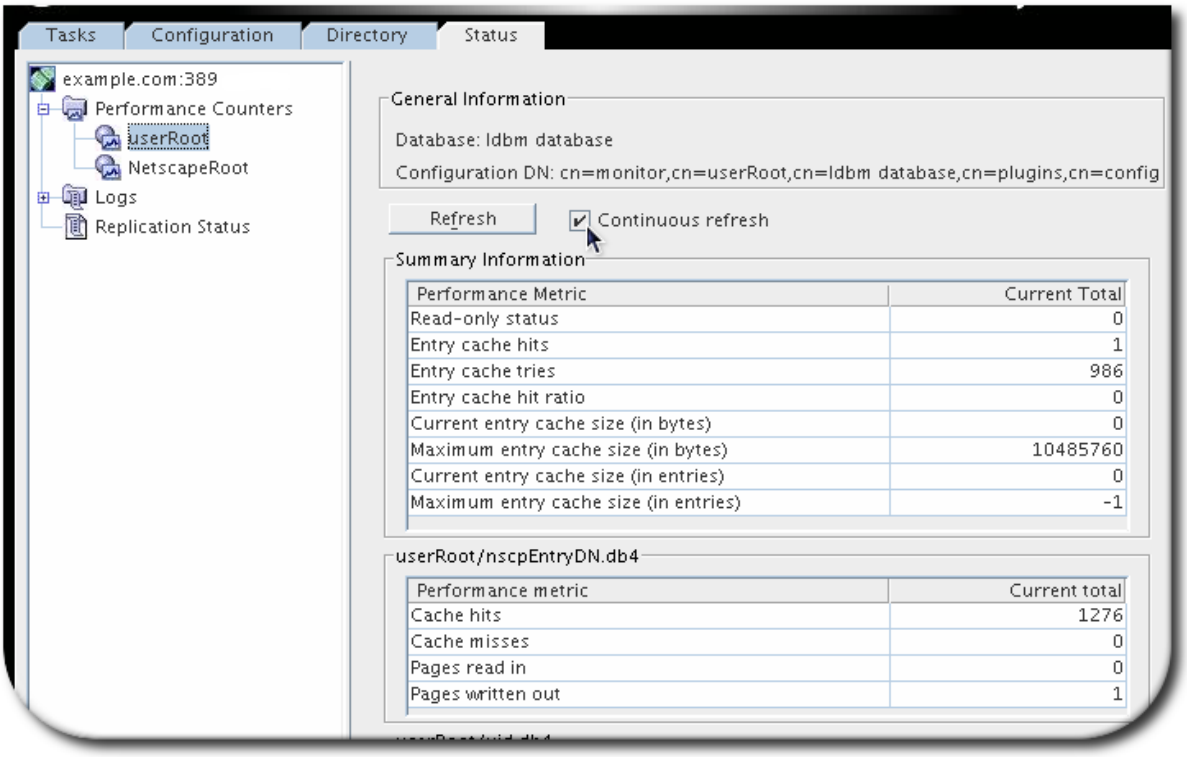
20.7.1. Directory Server コンソールからのデータベースアクティビティーの監視

データベースのアクティビティーを監視するには、以下を実行します。

1. *Directory Server コンソールで、Status タブを選択します。*

2. *ナビゲーションツリーで、Performance Counters フォルダーを展開し、監視するデータベースを選択します。*

   タブには、データベースアクティビティーの現在の情報が表示されます。サーバーが現在実行していない場合は、このタブではパフォーマンスの監視情報は提供されません。



3. *Refresh をクリックし、現在表示されている情報を更新します。ディレクトリーが継続的に表示される情報を更新する場合は、Continuous チェックボックスを選択し、Refresh をクリックします。*

表20.8 一般情報（データベース）

フィールド	*詳細*
データベース	*監視されるデータベースのタイプを特定します。*

フィールド	詳細
設定 DN	ldapsearch コマンドラインユーティリティーを使用して、検索ベースとして使用する必要のある識別名を特定します。

Summary Information セクションは、監視するすべてのデータベースの累積情報と、すべてのデータベースに適用されるキャッシュ関連の設定の一部を示しています。

表20.9 サマリー情報

パフォーマンスメトリック	現在の合計
読み取り専用のステータス	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します。nsslapd-readonly 属性が on に設定されている場合、データベースは読み取り専用モードになります。
エントリーキャッシュヒット	成功したエントリーキャッシュルックアップの合計数。つまり、ディスクに移動するのではなくキャッシュからデータを取得することで、サーバーが検索要求を処理できる合計回数です。
エントリーキャッシュのトリアトリア	ディレクトリーが最後に開始されてからのエントリーキャッシュルックアップの合計数。つまり、サーバー起動以降に要求されるエントリーの合計数です。

パフォーマンスメトリック	現在の合計
エントリー Cache Hit Ratio	エントリーキャッシュの検索の成功を試みる比率。この数は、ディレクトリーが最後に開始された後のルックアップおよびヒットの合計に基づいています。この値を 100% にすると、より良い値になります。操作がエントリーキャッシュに存在しないエントリーの検索を試みるたびに、ディレクトリーがエントリーを取得するためにディスクアクセスを実行する必要があります。そのため、この比率はゼロに対してドロップするため、ディスクアクセスの数は増加し、ディレクトリー検索のパフォーマンス低下します。 比率を改善するには、エントリーキャッシュの自動調整を有効にします。詳細は、『Red 『Hat Directory Server パフォーマンスチューニングガイド』の該当するセクションを参照してください』。
現在のエントリーキャッシュサイズ（バイト）	エントリーキャッシュに現在存在するディレクトリーエントリーの合計サイズ。

パフォーマンスメトリック	現在の合計
最大エントリーキャッシュサイズ（バイト単位）	ディレクトリーが維持するエントリーキャッシュのサイズ。 エントリーキャッシュのサイズは、cn=database _name,cn=ldbm database,cn=plugins,cn=config エントリーの nsslapd-cachememsize 属性に設定されます。パフォーマンスを最適化するには、エントリーキャッシュの自動調整を有効にします。詳細は、『Red 『Hat Directory Server パフォーマンスチューニングガイド』の該当するセクションを参照してください』。
現在のエントリーキャッシュサイズ（エントリー内）	エントリーキャッシュに現在存在するディレクトリーエントリーの数。
エントリーキャッシュサイズ（エントリー内）	非推奨。 エントリーキャッシュで保持できるディレクトリーエントリーの最大数。 許可される最大エントリー数を設定し、キャッシュサイズの管理を試行しないでください。これにより、ホストが RAM を効果的に割り当てることが困難になる可能性があります。

デフォルトでは、エントリーとインデックス化属性の両方でデータベースが維持されるため、デー

タベースの監視ページには多くの異なるデータベースが一覧表示されます。ただし、すべてのデータベースは、カウンターで、同じ種類のキャッシュ情報を監視します。

表20.10 データベースキャッシュ情報

パフォーマンスメトリック	現在の合計
Hits	データベースキャッシュが要求されたページを正常に提供した回数。
tries	データベースキャッシュがページを要求する回数。
Ratio のヒット	データベースキャッシュへのデータベースキャッシュヒットの比率。この値を 100% にすると、より良い値になります。ディレクトリー操作がデータベースキャッシュに存在しないデータベースの一部を見つけようとした場合、ディレクトリーは適切なデータベースページを取得するためにディスクアクセスを実行する必要があります。そのため、この比率はゼロに対してドロップするため、ディスクアクセスの数は増加し、ディレクトリーのパフォーマンス低下します。 比率を改善するには、データベースキャッシュの自動調整を有効にします。詳細は、『Red 『Hat Directory Server パフォーマンスチューニングガイド』の該当するセクションを参照してください』。
ページの読み取り	ディスクからデータベースキャッシュに読み取られるページ数。
ページが書き込まれる	キャッシュからディスクに書き込まれたページ数。データベースページは、読み取り/書き込みページが変更され、その後キャッシュから削除されるたびにディスクに書き込まれます。 キャッシュが満杯になり、ディレクトリー操作に現在キャッシュに保存されていないデータベースページが必要な場合に、ページはデータベースキャッシュから削除されます。

パフォーマンスメトリック	現在の合計
読み取り専用ページのエビクション	新規ページのスペースを作成するために、キャッシュから破棄された読み取り専用ページの数。
読み取り/書き込みページのエビクション	新規ページのスペースを作成するために、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。この値は、変更されていない読み取り書き込みページを破棄する点で、Pages Written Out とは異なります。

表20.11 データベースファイル固有の

パフォーマンスメトリック	現在の合計
Cache Hits	検索によって、この特定のファイルでキャッシュがヒットした回数。つまり、クライアントはこのファイルからデータを必要とする検索を実行し、ディレクトリーはキャッシュから必要なデータを取得します。
キャッシュがない	検索結果が、この特定のファイルのキャッシュに到達できなかった回数。つまり、このファイルから必要なデータが実行されており、必要なデータがキャッシュにあることができませんでした。
ページの読み取り	このファイルからキャッシュに移動するページ数。
ページが書き込まれる	キャッシュからディスクに書き込まれるこのファイルのページ数。

20.7.2. コマンドラインでのデータベースの監視

データベースの現在のアクティビティーは、ldapsearch などの LDAP ツールを使用して監視できます。検索は、LDBM データベースエントリーの監視サブツリー cn=monitor,cn=database_name,cn=ldbm database,cn=plugins,cn=config をターゲットにします。これには、その特定のデータベースインスタンスの監視属性がすべて含まれます。

以下に例を示します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -s base -b
"cn=monitor,cn=database_name,cn=ldbm database,cn=plugins,cn=config" "(objectclass=*)"
```

表20.12 データベースモニタリングの属性

属性	詳細
データベース	現在モニターされているデータベースのタイプを特定します。
readonly	データベースが読み取り専用モードであるかどうかを指定します。0 は 、 サーバーが読み取り専用モードではないことを意味します。1 は、読み取り専用モードであることを意味します。
entrycachehits	成功したエントリーキャッシュルックアップの合計数。つまり、ディスクに移動するのではなくキャッシュからデータを取得することで、サーバーが検索要求を処理できる合計回数です。
entrycachetries	ディレクトリーが最後に開始されてからのエントリーキャッシュルックアップの合計数。つまり、サーバー起動以降にサーバーに対して実行される検索操作の合計数。
entrycachehitratio	エントリーキャッシュの検索の成功を試みる比率。この数は、ディレクトリーが最後に開始された後のルックアップおよびヒットの合計に基づいています。この値を 100% にすると、より良い値になります。検索操作がエントリーキャッシュに存在しないエントリーの検索を試みるたびに、ディレクトリーがエントリーを取得するためにディスクアクセスを実行する必要があります。そのため、この比率はゼロに対してドロップするため、ディスクアクセスの数は増加し、ディレクトリー検索のパフォーマンス低下します。 比率を改善するには、エントリーキャッシュの自動調整を有効にします。詳細は、『Red 『Hat Directory Server パフォーマンスチューニングガイド』の該当するセクションを参照してください』。
currententrycachesize	エントリーキャッシュに現在存在するディレクトリーエントリーの合計サイズ（バイト単位）。

属性	詳細
maxentrycachesize	エントリーキャッシュで保持できるディレクトリーエントリーの最大サイズ（バイト単位）。 エントリーキャッシュのサイズは、cn= database _name,cn=ldbm database,cn=plugins,cn=config エントリーの nsslapd-cachememsize 属性に設定されます。パフォーマンスを最適化するには、エントリーキャッシュの自動調整を有効にします。詳細は、『Red 『Hat Directory Server パフォーマンスチューニングガイド』の該当するセクションを参照してください』。
dbcachehits	ディスクに移動するのではなく、キャッシュからデータを取得することで、サーバーがリクエストを処理できる回数。
dbcachetries	サーバー起動後のデータベースアクセスの総数。
dbcachehitratio	キャッシュの比率は、キャッシュヒットの成功を試みます。この数を 100% にすると、より良い数値になります。
dbcachepagein	ディスクからキャッシュに読み取られるページ数。
dbcachepageout	キャッシュからディスクに書き込まれたページ数。
dbcacheroevict	新規ページのスペースを作成するために、キャッシュから破棄された読み取り専用ページの数。キャッシュから破棄されたページはディスクに書き込まれ、サーバーのパフォーマンスに影響する可能性があります。ページの数が小さいほどエビクトされます。
dbcacherwevict	新規ページのスペースを作成するために、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。この値は、変更されていない読み取り書き込みページを破棄する点で、Pages Written Out とは異なります。キャッシュから破棄されたページはディスクに書き込まれ、サーバーのパフォーマンスに影響する可能性があります。ページの数が小さいほどエビクトされます。
dbfilename-number	ファイルの名前です。数値は、ファイルの連続した整数識別子（0 から開始）を提供します。ファイルに関連付けられたすべての統計には、同じ数値 ID が指定されます。

属性	詳細
***dbfilecachehit*-number**	*検索によって、この特定のファイルでキャッシュがヒットした回数。つまり、クライアントはこのファイルからデータを必要とする検索を実行し、ディレクトリーはキャッシュから必要なデータを取得します。*
***dbfilecachemiss*-number**	*検索結果が、この特定のファイルのキャッシュに到達できなかった回数。つまり、このファイルから必要なデータが実行されており、必要なデータがキャッシュにあることができませんでした。*
***dbfilepagein*-number**	*このファイルからキャッシュに移動するページ数。*
***dbfilepageout*-number**	*キャッシュからディスクに書き込まれるこのファイルのページ数。*
currentdncachesize	*DN キャッシュに現在存在する DN の合計サイズ（バイト単位）。* *DN キャッシュに存在するエントリーのサイズを増やすには、データベースの* **cn= database_name ,** *cn=ldbm database,cn=plugins,cn=config エントリーの nsslapd-dncachememsize 属性の値を増やします。*
maxdncachesize	*DN キャッシュで保持できる DN の最大サイズ（バイト単位）。* *キャッシュに存在するエントリーのサイズを増やすには、データベースの* **cn= database_name、cn=** *ldbm database,cn=plugins,cn=config エントリーの nsslapd-dncachememsize 属性の値を増やします。*
currentdncachecount	*DN キャッシュに現在存在する DN の数。*

20.8. データベースリンクアクティビティーの監視

ldapsearch コマンドラインユーティリティーを使用してコマンドラインからデータベースリンクのアクティビティーを監視し、必要な監視属性を返すことができます。モニタリング属性は cn=monitor,cn=database_link_name、cn=chaining database,cn=plugins,cn=config に保存されます。

たとえば、ldapsearch コマンドラインユーティリティーを使用して、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取得できます。たとえば、以下のコマンドは、DB Link1 というデータベースリンク を監視します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -s sub -b
"cn=monitor,cn=DBLink1,cn=chaining database,cn=plugins,cn=config" "(objectclass=*)" nsAddCount
```

表20.13「データベースリンクの監視属性」 に、監視可能なデータベースリンク監視属性を一覧表示します。

表20.13 データベースリンクの監視属性

属性名	詳細
nsAddCount	受信した追加操作の数。
nsDeleteCount	受信した削除操作の数。
nsModifyCount	受信した変更操作の数。
nsRenameCount	受信した名前変更操作の数。
nsSearchBaseCount	受け取ったベースレベルの検索の数。
nsSearchOneLevelCount	受信した 1 レベル検索の数。
nsSearchSubtreeCount	受信したサブツリー検索の数。
nsAbandonCount	受信した Abandon 操作の数。
nsBindCount	受信したバインド要求の数。
nsUnbindCount	受信したバインド解除の数。
nsCompareCount	受信した比較操作の数。
nsOperationConnectionCount	通常操作のオープン接続の数。

属性名	詳細
nsBindConnectionCount	バインド操作のオープン接続の数。

20.9. カウンターの有効化および無効化

nsslapd-counters 属性により、実行するカウンターが有効になります。ただし、カウンターの実行はパフォーマンスに影響する可能性があるため、カウンターをオフにすることもできます。カウンターがオフの場合、カウンターの値はすべてゼロ (0) になります。

デフォルトでは、カウンターはすでに有効になっています。パフォーマンスカウンターを有効または無効にするには、ldapmodify を使用します。

```
ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=config
changetype: modify
replace: nsslapd-counters
nsslapd-counters: off
```

第21章 SNMP を使用した DIRECTORY SERVER の監視

「20章サーバーおよびデータベースアクティビティーの監視」で説明しているサーバーおよびデータベースアクティビティー監視のログ設定は、Directory Server に固有のものです。また、SNMP (Simple Network Management Protocol) を使用して Directory Server を監視することもできます。SNMP は、ネットワーク活動を監視するのに使用する管理プロトコルで、さまざまな機器をリアルタイムに監視することができます。

Directory Server は、AgentX サブエージェントを使用して SNMP を監視できます。SNMP 監視は、バインド情報、サーバーで実行される操作、キャッシュ情報など、Directory Server に関する有用な情報を収集します。Directory Server SNMP サブエージェントは SNMP トラップをサポートし、サーバーインスタンスの実行状態の変更に関する通知を送信します。

21.1. SNMP の概要

SNMP は広く普及しているため、相互運用性があります。このような相互運用性と、SNMP がさまざまなデバイスクラスに固有の多くのジョブを引き受けることができるという事実により、SNMP はグローバルネットワークの制御と監視のための理想的な標準メカニズムとなっています。SNMP により、ネットワーク管理者はすべてのネットワーク監視活動を統合することができ、Directory Server の監視もその一部となります。

SNMP は、ネットワークアクティビティーに関するデータを交換するために使用されます。SNMP では、ユーザーがネットワークをリモートで管理する管理デバイスとネットワーク管理アプリケーション (NMS) の間でデータが伝送されます。管理デバイスは、ホスト、ルーター、Directory Server などの SNMP を実行するすべてです。NMS 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です。ネットワーク管理アプリケーションは、管理している機器の情報、どの機器が稼働または停止しているのか、どのエラーメッセージをどれだけ受け取ったのか、などをグラフィカルに表示します。

NMS と管理デバイスに関する情報は、サブエージェントと マスターエージェント の 2 種類のエージェントを使用して転送されます。サブエージェントは、管理対象デバイスに関する情報を収集し、情報をマスターエージェントに渡します。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、さまざまなサブエージェントと NMS との間のので報を交換します。マスターエージェントは、通常、リモートマシンで実行できるものの、通信するサブエージェントと同じホストマシンで実行します。

問い合わせ可能な SNMP 属性 (変数とも呼ばれる) の値は、管理対象機器に保持され、必要に応じて NMS に報告されます。各変数は 管理オブジェクト と呼ばれ、エージェントがアクセスして NMS に送信できるものです。すべての管理オブジェクトは、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 (MIB) で定義されます。階層の最上位には、ネットワークに関する最も一般的な情報が含まれます。その下の各ブランチはより具体的で、個別のネットワーク領域を扱っています。

SNMP は、プロトコルデータユニット (PDU) の形式でネットワーク情報を交換します。PDU には、

管理対象デバイスに保存されている変数に関する情報が含まれます。これらの変数は、管理オブジェクトとも呼ばれ、必要に応じて NMS にレポートされる値とタイトルを持ちます。NMS と管理対象機器の間の通信は、NMS が更新情報を送信したり、情報を要求したり、管理対象オブジェクトがサーバーのシャットダウンや起動時に トラップ と呼ばれる通知や警告を送信することで行われます。

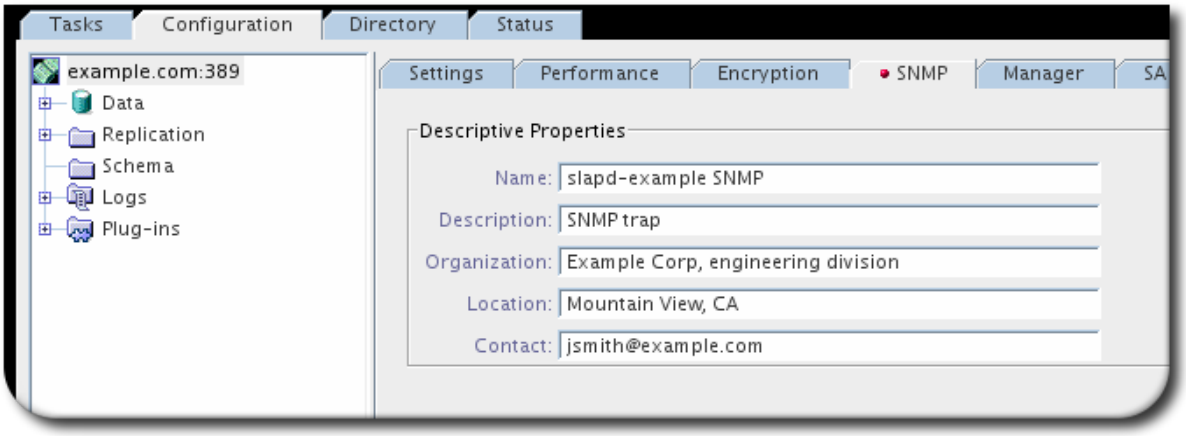
21.2. SNMP 用の DIRECTORY SERVER の設定

デフォルトでは、Directory Server は、サブエージェントが設定されると同時に SNMP を使用して監視することができます。ただし、Directory Server インスタンスに有用な変数がいくつかあります。これは、SNMP を使用して Directory Server インスタンスの特定に役立ちます。Directory Server コンソールからこれらの SNMP 設定を設定するには、以下を実行します。

1. Configuration タブを選択し、左側のペインのナビゲーションツリーで最上位のエントリーを選択します。

2. メインウィンドウで SNMP タブを選択します。

3. Net-SNMP で Directory Server インスタンスを簡単に識別できるように、SNMP 記述子に関する情報を入力します。



- インスタンスの一意の名前および説明。

- ディレクトリーインスタンスが属する企業または組織。

- インスタンスを管理するディレクトリーインスタンスまたは組織の物理的な場所。

- Directory Server インスタンスを維持するユーザーのメールアドレスまたは連絡先番

号。

4. Save をクリックします。

21.3. DIRECTORY SERVER の SNMP AGENT の設定

SNMP プロトコルを使用して Directory Server から情報をクエリーするには、SNMP エージェントを設定します。

1. 389-ds-base-snmp パッケージおよび net-snmp パッケージをインストールします。

```
# yum install 389-ds-base-snmp net-snmp
```

2. SNMP マスターエージェントを設定するには、/etc/snmp/snmpd.conf ファイルを編集し、以下のエントリーを追加してエージェントの拡張性 (AgentX) プロトコルを有効にします。

```
master agentx
```

AgentX プロトコルの詳細は RFC 2741 を参照してください。

3. SNMP サブエージェントを設定するには、/etc/dirsrv/config/ldap-agent.conf ファイルを編集し、監視する各 Directory Server インスタンスに server パラメーターを追加します。以下に例を示します。

```
server slapd-instance_name
```

4. 必要に応じて、SNMP ユーザーアカウントを作成します。

a. snmpd サービスを停止します。

```
# systemctl stop snmpd
```

b. SNMP ユーザーアカウントを作成します。以下に例を示します。

```
# net-snmp-create-v3-user -A authentication_password -a SHA \
    -X private_password -x AES user_name
```

コマンドで使用されるパラメーターの詳細は、net-snmp-create-v3-user(1) の man ページを参照してください。

c. snmpd サービスを起動します。

```
# systemctl start snmpd
```

5. 必要に応じて、Directory Server 記述プロパティーを設定します。詳細は、「SNMP 用の Directory Server の設定」 を参照してください。

6. dirsrv-snmp サービスを起動します。

```
# systemctl start dirsrv-snmp
```

7. 必要に応じて、設定を確認するには、以下を実行します。

a. net-snmp-utils パッケージをインストールします。

```
# yum install net-snmp-utils
```

b. Directory Server オブジェクト識別子 (OID) をクエリーします。以下に例を示します。

```
# snmpwalk -v3 -u user_name -M /usr/share/snmp/mibs:/usr/share/dirsrv/mibs/ \
    -l AuthPriv -m +RHDS-MIB -A authentication_password -a SHA
    -X private_password -x AES server.example.com .1.3.6.1.4.1.2312.6.1.1
```

SNMP の詳細については、『Red Hat システム管理者ガイド』 の 『Net-SNMP によるパフォーマンスの監視』 を参照してください。

21.4. SNMP トラップの設定

SNMP トラップは基本的に、監視対象のサーバーによって問題が発生した場合に通知をトリガーする

しきい値です。トラップを使用するには、マスターエージェントを設定して、トラップを許可し、それらの操作を行うように設定する必要があります。たとえば、トラップは、Directory Server インスタンスの管理者が停止するメール通知をトリガーできます。

サブエージェントは、トラップをマスターエージェントに送信するだけです。マスターエージェントとトラップハンドラーは、使用している SNMP マスターエージェントのドキュメントに従って設定する必要があります。

トラップは Entity Table からの情報に付随します。これには、名前やバージョン番号などの Directory Server インスタンスに固有の情報が含まれます。Entity Table は、「エンティティーテーブル」で説明しています。つまり、マスターエージェントがトラップを受けたときに取るアクションは、あるインスタンスでは dsEntityContact 変数に定義された電子メールアドレスに電子メールを送信する一方で、別のインスタンスでは dsEntityContact 変数に定義されたページャー番号に通知を送信するなど、柔軟に対応することができます。

サブエージェントでサポートされるトラップは 2 つあります。

- DirectoryServerDown.このトラップは、サブエージェントが Directory Server が実行されていないことを検出するたびに生成されます。このトラップは、Directory Server インスタンスの説明、バージョン、物理的な場所、および連絡先情報と共に送信されます。詳細は、dsEntityDescr 変数、dsEntityVers 変数、dsEntityLocation 変数、および dsEntityContact 変数を参照してください。

- DirectoryServerStart.このトラップは、サブエージェントが Directory Server が起動または再起動していることを検出すると常に生成されます。このトラップは、Directory Server インスタンスの説明、バージョン、物理的な場所、および連絡先情報と共に送信されます。詳細は、dsEntityDescr 変数、dsEntityVers 変数、dsEntityLocation 変数、および dsEntityContact 変数を参照してください。

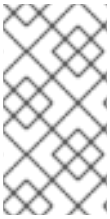
21.5. 管理情報ベースの使用

Directory Server の MIB は、redhat -directory.mib と呼ばれるファイルです。この MIB には、そのディレクトリーのネットワーク管理に関する変数の定義が含まれます。これらの変数は、管理オブジェクトと呼ばれます。ディレクトリー MIB および Net-SNMP を使用すると、ネットワーク上の他の全デバイスと同様にディレクトリーを監視できます。MIB を使用する方法は、「Directory Server の SNMP Agent の設定」を参照してください。

クライアントツールは、Directory Server MIB を読み込み、以下のセクションに記載されている変数名を使用する必要があります。

ディレクトリー MIB を使用すると、管理者は SNMP を使用してディレクトリーの管理情報を確認し、リアルタイムでサーバーを監視できるようになります。ディレクトリー MIB は、管理オブジェクトの 4 つの異なるテーブルに分類されています。

- 「操作表」
- 「エントリー表」
- 「エンティティーテーブル」
- 「対話表」



注記

SNMP に監視される Directory Server 属性はすべて、32 ビットシステムであってもカウンターに 64 ビット整数を使用します。

21.5.1. 操作表

Operations Table は、Directory Server のアクセス、操作、およびエラーに関する統計情報を提供します。表21.1「操作テーブル: 管理オブジェクトと説明」は、redhat-directory.mib ファイルの Operations Table に保存されている管理オブジェクトを説明します。

表21.1 操作テーブル: 管理オブジェクトと説明

管理オブジェクト	説明
dsAnonymousBinds	サーバーの起動以降、ディレクトリーへの匿名バインド数。
dsUnauthBinds	サーバーの起動以降、ディレクトリーへの認証されていないバインド数。
dsSimpleAuthBinds	サーバーが起動してから、単純な認証方法 (パスワード保護など) で確立された、ディレクトリーのバインド数。
dsStrongAuthBinds	サーバーの起動してから、強力な認証方法 (TLS や、Kerberosのような SASL メカニズムなど) を使用して確立されたディレクトリーへのバインドの数。

管理オブジェクト	説明
dsBindSecurityErrors	サーバーの起動以降に、認証失敗または無効な認証情報により、ディレクトリーで拒否されたバインド要求の数。
dsInOps	サーバーの起動以降、別のディレクトリーからこのディレクトリーに転送される操作の数。
dsReadOps	アプリケーションが起動してからこのディレクトリーによる読み取り操作の数。LDAP は検索操作を使用して間接的に読み取り操作を実装するため、このオブジェクトの値は常に 0 になります。
dsCompareOps	サーバーの起動時にこのディレクトリーによる比較操作の数。
dsAddEntryOps	サーバーの起動時にこのディレクトリーによる追加操作の数。
dsRemoveEntryOps	サーバーの起動以降、このディレクトリーがサービス化された削除操作の数。
dsModifyEntryOps	サーバーの起動以降、このディレクトリーがサービス化された変更操作の数。
dsModifyRDNOps	サーバー起動以降、このディレクトリーが処理する RDN 操作の数。
dsListOps	サーバーの起動時にこのディレクトリーによるリスト操作の数。LDAP は検索操作を使用して間接的にリスト操作を実装するため、このオブジェクトの値は常に 0 になります。
dsSearchOps	サーバーの起動以降、このディレクトリーで処理された検索操作の合計数。
dsOneLevelSearchOps	サーバー起動以降、このディレクトリーが処理する 1 レベルの検索操作の数。
dsWholeSubtreeSearchOps	サーバー起動以降、このディレクトリーが指定したサブツリー検索操作全体の数。
dsReferrals	サーバー起動からクライアント要求に対応して、このディレクトリーが返す参照数。
dsSecurityErrors	セキュリティー要件を満たしていないこのディレクトリーに転送される操作の数。

管理オブジェクト	説明
dsErrors	エラー (セキュリティーエラーや参照エラー以外) のためにサービスを提供できなかった要求の数です。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などがあります。部分的に設定されたリクエストはエラーとしてカウントされません。

21.5.2. エントリー表

Entries Table は、ディレクトリーエントリーの内容に関する情報を提供します。表21.2「エントリーテーブル: 管理オブジェクトと説明」は、redhat-directory.mib ファイルの Entries Table に保存されている管理オブジェクトを説明します。

表21.2 エントリーテーブル: 管理オブジェクトと説明

管理オブジェクト	説明
dsMasterEntries	このディレクトリーにマスターエントリーが含まれるディレクトリーエントリーの数。（現在更新が行われないため）このオブジェクトの値は、常に 0 になります。
dsCopyEntries	このディレクトリーのコピーが含まれるディレクトリーエントリーの数。このオブジェクトの値は、常に 0 になります (現在実行された更新は実行されていません)。
dsCacheEntries	ディレクトリーにキャッシュされたエントリーの数。
dsCacheHits	アプリケーションが起動してからローカルに保持されたキャッシュから指定された操作数。
dsSlaveHits	ローカルを保持するレプリケーション (shadow エントリー) で処理されたオペレーションの数。このオブジェクトの値は常に 0 になります。

21.5.3. エンティティーテーブル

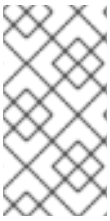
Entity Table には、Directory Server インスタンスに関する識別情報が含まれます。「SNMP 用の Directory Server の設定」 で説明されているように、Entity Table の値が Directory Server コンソールに設定されます。

表21.3「エントリーの表: 管理オブジェクトおよび説明」は、redhat-directory.mib ファイルの Entity Table に保存されている管理オブジェクトを説明します。

表21.3 エントリーの表: 管理オブジェクトおよび説明

管理オブジェクト	説明
dsEntityDescr	Directory Server インスタンスの説明セット。
dsEntityVers	Directory Server インスタンスの Directory Server のバージョン番号。
dsEntityOrg	Directory Server インスタンスに対応する組織。
dsEntityLocation	Directory Server インスタンスの物理的な場所。
dsEntityContact	Directory Server インスタンス担当する担当者の名前と連絡先情報。
dsEntityName	Directory Server インスタンスの名前。

21.5.4. 対話表



注記

Interaction Table はサブエージェントではサポート されていません。サブエージェントはテーブルをクエリーできますが、有効なデータで更新されることはありません。

表21.4「対話テーブル: 管理オブジェクトと説明」は、redhat-directory.mib ファイルの Interaction Table に保存されている管理オブジェクトを説明します。

表21.4 対話テーブル: 管理オブジェクトと説明

管理オブジェクト	説明
dsIntTable	表の各行で、監視される Directory Server とそれぞれのピア Directory Server との対話の履歴に関連する詳細が表示されます。
dsIntEntry	Directory Server と相手の Directory Server との相互作用の詳細を示すエントリー。
dsIntIndex	applIndex と共に一意の鍵の一部であり、(applIndex で参照される) Directory Server と相手の Directory Server との間の (試行された) 相互作用に関する有用な情報を含む概念的な行を特定するためのものです。

管理オブジェクト	説明
dsName	このエントリーが属するピア Directory Server の識別名 (DN)。
dsTimeOfCreation	この行が作成された場合の sysUpTime の値。ネットワーク管理サブシステムが初期化される前にエントリーが作成されると、このオブジェクトにはゼロの値が含まれます。
dsTimeOfLastAttempt	この Directory Server に対する接続最終試行時の sysUpTime の値。ネットワーク管理サブシステムが初期化される前に最後の試行が行われた場合、このオブジェクトにはゼロの値が含まれます。
dsTimeOfLastSuccess	この Directory Server に問い合わせた最後の試行時の sysUpTime の値。このエントリーは、成功した試行がない場合や、最後に成功した試行がネットワーク管理サブシステムの初期化前に行われた場合には、0 の値になります。
dsFailuresSinceLastSuccess	この Directory Server への初回連絡の試行に成功した後の失敗回数。試行に成功しなかった場合、このカウンターには、このエントリーが作成されてからの失敗回数が格納されます。
dsFailures	このエントリーの作成からの累積的な障害。
dsSuccesses	このエントリーの作成以降、累積成功。
dsURL	Directory Server アプリケーションの URL。

第22章 高可用性および障害復旧計画の作成

Directory Server デプロイメントを効率的に実行する場合は、その最悪のケースシナリオに計画されています。この章では、災害復旧計画を作成するための一般的な原則を説明し、災害復旧に役立つ Directory Server の機能を紹介します。

障害復旧 は、ある種の壊滅的な障害が発生した場合に、ある動作環境から別の動作環境へのスムーズな移行を計画および実行する方法です。Directory Server の災害復旧計画は、大規模な事業継続計画の一部である場合もあれば、ディレクトリーサービスの中断に特化した独立した計画である場合もあります。

注記

本章では、障害復旧に関する非常に一般的な概念を説明します。

障害復旧は非常に複雑で、詳細固有の内容になります。Red Hat Directory Server のような機密性の高いサービスやミッションクリティカルなサービスに対する障害復旧の設計、保守、テストには、専門サービスの利用を検討してください。

22.1. 潜在的なシナリオの特定

最初のステップでは、発生する可能性のある問題、サービスへの影響、および実行すべき応答を特定します。『Red Hat Directory Server Deployment Guide』 では、管理者が既存のインフラストラクチャーと提案するインフラストラクチャーのサイトサーベイを行い、どのようなディレクトリーを設計するかを決定しました。災害対策についても同様で、表22.1「障害シナリオおよび応答」のように、データインフラストラクチャーがどこにあるかを特定し、そのコンポーネントが失われた場合の影響を判断し、理想的な対応策を検討します。

表22.1 障害シナリオおよび応答

シナリオ	インフラストラクチャーへの影響	理想的な応答
データの破損	ソフトウェアやハードウェア障害 (または悪意のある攻撃経由) を介して、1つのサイトまたは1つサーバーのデータが破損する可能性があります。その破損したサーバーがマルチマスターレプリケーションのサプライヤーである場合、破損はすぐにデプロイメント全体に伝播してしまいます。	破損していないデータの最新のバックアップにアクセスできる、分離されたサーバーを用意する必要があります。問題が検出された場合は、通常のインフラストラクチャーでのレプリケーションを中断し、このサーバーをオンラインにして、適切なデータでサプライヤーを再初期化することができます。

シナリオ	インフラストラクチャーへの影響	理想的な応答
自然な障害およびその他の大量イベント	自然災害は、長期間の停電だけでなく、オフィスやデータセンター全体を停止させる可能性があります。	ディレクトリー操作は、同じデータを使用して、別の物理の場所にあるミラーリングされたサイトに転送できます。
サーバーまたはマシンの損失	1 つのマシンが失敗する可能性があります。	同じデータを持つ別のマシンは、失われたマシンがあることを想定できます。

22.2. ロールオーバーの種類の定義

災害復旧は、あるシステムから別のシステムに移行するプロセスであり、可能な限りサービスを中断するプロセスです。これは ロールオーバー と呼ばれ、ロールオーバーを行う方法は 3 つあります。

- *ホットロール オーバーは、インフラストラクチャーが別のサイトで完全にミラーリングされ、バックアップサイトは常にプライマリーサイトで最新の状態であることを意味します。これには、プライマリーからバックアップに操作を切り替えるための調整のみが必要になります。*

- *ウォーム ロールオーバーとは、バックアップサイトのすべての要素 (適切なネットワーク接続、必要なすべてのアプリケーションとハードウェア) は整っているが、システムがアクティブに稼働していない、または必ずしも設定されていない状態を指します。これには、マシンを設定し、システムの実行に追加の時間が必要になる場合があります。*

- *コールド ロールオーバーとは、サイトは利用可能だが、それをセットアップするためのリソースがすぐには得られないことを意味します。*

ロールオーバーの種類における明らかな違いは、バックアップサイトの設定に必要な時間と費用です。ホットサイトおよびウォームサイトは、立ち上げや運営にかかる初期費用が高くなります。

計画している特定の障害シナリオに応じて、ロールオーバータイプの組み合わせを使用できます。たとえば、1 台のサーバーが失われた場合のロールオーバー計画では、Directory Server インスタンスの仮想マシンコピーを作成して保持し、数分以内にオンラインにすることで、簡単かつ比較的安価にホットロールオーバーを利用することができます。仮想マシンを別の施設やネットワークで維持する必要もありません。一方、コールドロールオーバーは、データセンターやオフィス全体の損失を想定して計画することができます。

ロールオーバーのプロセスを、災害シナリオの深刻さ、予算と利用可能なリソース、問題発生の可能性に合わせます。

22.3. 障害復旧における便利な DIRECTORY SERVER 機能の特定

復旧で最も難しいのはハードウェアではなく、サーバー内のデータの信頼できるコピーを得ることです。障害復旧用にデータコピーを準備する優れたツールとして、Directory Server には 3 つの機能があります。

- データベースのバックアップおよびバックアップの定期的な検証
- マルチマスターのレプリケーション、チェーン、データベースのバックアップ、および名前付きパイプスクリプトでサーバーの監視
- チェーン

また、名前付きパイプスクリプトや他の Directory Server のパフォーマンスカウンターを使用してサーバを監視することで、特定の重要なイベントを発見し、迅速に対応することができます。

22.3.1. 災害リカバリー用のディレクトリーデータのバックアップ

障害復旧で最も便利なツールは、ディレクトリーインスタンスのバックアップを頻繁に行うことです。アーカイブは、プライマリーデータセンターとは異なる場所で、コールドバックアップの場所で物理メディアに保存できます。バックアップ操作と復元操作は、（db 2bak.plなど）shell または perl スクリプトのいずれかを使用して実行できます。

バックアップは、cron ジョブを使用して定期的に実行するように自動化できます。以下に例を示します。

```
0 7 * * 1 /usr/sbin/db2bak.pl -Z instance_name
```

Perl スクリプト db2bak.pl は、最初にサーバーを停止せずにディレクトリーデータをバックアップします。

注記

Red Hat は、マルチマスターレプリケーション環境のすべてのサーバー上でデータのバックアップを行うことを推奨します。

「データのバックアップおよび復元」 では、ディレクトリーデータベースとディレクトリー設定（dse.ldif ファイル）の両方のバックアップが対象です。

22.3.2. 高可用性のためのマルチ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は、1 台のサーバーや、場合によってはオフィスや部門全体における損失に関する最善の防御策です。少数のサーバーがデータマスターとなる一方で、複数のサーバーがすべて同じデータを保持しており、1 つのレプリケーション環境に数十台のマスターとハブが存在する可能性があります。これにより、複数のサーバーがオフラインであっても、クライアントが情報にアクセスできる状態を維持します。

レプリケーションは、データをサーバーにコピーしたり、より迅速に交換するのに使用できます。

注記

レプリケーションを介して伝播されるデータの破損を保護するには、データベースを頻繁にバックアップします。

レプリケーション設定により、プライマリーサプライヤーがアクセスできない場合に、書き込み操作はフェイルオーバーサーバーと呼ばれます。つまり、サーバーがオフラインであっても、書き込み操作はクライアントの観点から通常通り続行できます。

例22.1 マルチマスターレプリケーションのシナリオ

レプリケーションは、いくつかのシナリオで障害復旧のための汎用的なツールです。

- 単一のサーバー障害の場合、そのインスタンスに保存されているすべてのデータには、他のサーバーからアクセスと取得が可能です。
- オフィス全体やコロケーション施設が失われた場合は、まったく別の物理的な場所にサーバーをミラーリングすることができます (Directory Serverの広域レプリケーションの

> 性能が役立ちます)。最低限の努力で、新しいサーバーをオンラインにすることなく、トラフィックは複製されたサイトにリダイレクトされます。

レプリケーションの設定については、「15章レプリケーションの管理」で説明しています。

22.3.3. 高可用性のデータベースチェーン

チェーン は、クライアントが要求を 1 台のサーバーに送信し、その要求が自動的に別のサーバーに転送されて処理されるという構成です。データベースリンク (またはチェーン) に複数のサーバーを設定して、1 つのサーバーが利用できない場合に自動フェイルオーバーを行うことができます。

> 例22.2 連鎖のシナリオ
>
> チェーンがフェイルオーバーサーバーのリストと組み合わせると、クライアントトラフィックはオフライン時に単一サーバー (またはサーバーのグループも) から自動的にリダイレクトできます。これは復旧には役立ちませんが、プライマリーサーバーからバックアップサーバーへの移行を管理するのに役立ちます。

チェーンデータベースについては、「データベースリンクの作成および維持」で説明しています。

22.4. リカバリープロセスの定義

障害復旧を支援するツールは数多くありますが、効果的な復旧プロセスは、あらゆるシナリオで何をすべきかを明確に定義したプランを持つことに集約されます。少なくとも 2 つの項目を明確に特定する必要があります。

- 障害を示すシグナル。明白なもの (大規模な停電、ネットワーク損失、または火災) もありますが、定義する必要があるものもあります。たとえば、バックアップサーバーをオンラインにする必要があるというメッセージは何ですか。
- 障害に応答する人および方法。災害が発生したら、誰が責任を持って行動しますか。これらのイベントについて通知する方法。何を期待されていますか。

重要

- *災害復旧計画を出力したものをサイト外に保管します。*
- *障害復旧計画を定期的にテストし、設定とインフラストラクチャーの変更後にテストします。*

22.5. 基本的な例: リカバリーの実行

管理者 (John Smith) は、そのディレクトリーデプロイメントの障害復旧計画を作成する必要があります。Example Corp. は、サンフランシスコ、ダラス、アーリントンの 3 箇所にオフィスを構えています。各サイトには 10 台のサーバーがあり、ローカルで相互にレプリケーションを行い、各サイトの 1 台のサーバーが他の 2 つのサイトの別のサーバーにレプリケーションを行います。

各サイトのディレクトリーには、ビジネスに不可欠な顧客データや、人事データが保存されています。請求などの操作を実行するために、データにいくつかの外部アプリケーションにアクセスする必要があります。

John Smith の最初の手順は、サイト Survey を実行することです。彼が探しているのは、ディレクトリーの使用状況 (アクセスするクライアントやサイト全体のトラフィック負荷)、現在の資産、そして取得が必要な資産の 3 つです。これは、Red Hat Directory Server のデプロイ時に実行する初期サイトサーベイと似ています。

次の手順では、障害のシナリオを特定します。3 つのサイトのうち 2 つ (サンフランシシスコとダラス) は、自然災害に対して非常に脆弱です。3 つのサイトはすべて、電源やインターネットアクセスの停止など、通常の中断が生じる可能性がありました。また、各サイトは独自のローカルデータをマスターするため、各サイトはサーバーインスタンスまたはマシンの損失に対して脆弱です。

John Smith は、障害復旧計画を 3 つの部分に分類します。

- *プラン A では、Directory Server における 1 つのインスタンスの損失について扱います。*
- *プラン B は、何らかのデータの破損または攻撃を扱います。*
- *プラン C では、オフィス全体が失われた場合を扱います。*

プラン A と B の場合、John Smith はホットリカバリーを使用して、1 つのインスタンスからバックアップに機能を即座に切り替えることにしました。各サーバーは毎日バックアップされ、cron ジョブを使用してアーカイブをコピーし、次にアーカイブが仮想マシンでコピーされ、復元されます。仮想マシンは別のサブネットに保管されますが、そのピアがオフラインになるとすぐに切り替えることができます。John Smith は、簡単な SNMP トラップを使用して、各 Directory Server インスタンスの可用性を追跡します。

プラン C はより広範囲です。サイト間のレプリケーションとローカルのバックアップに加えて、彼は各サイトのバックアップの物理的なコピーを、ローカルのインスタンスごとに、週に一度、他の 2 つのコロケーション施設に郵送することにしました。また、仮想マシンを使用してサイト全体をリストアするために、十分なインターネットアクセスとソフトウェアライセンスを備えた予備のサーバーを、他の異なるコロケーション施設の 1 つに置いています。彼は、IT スタッフのほとんどがいる Arlington を一次復旧拠点とし、次にサンフランシスコ、最後にダラスと、人員の分布に基づいて指定しています。すべてのイベントにおいて、3 つのサイトの IT 管理者に通知され、管理者は仮想マシンのセットアップ、物理的なバックアップからの Directory Server インスタンスのリストア、クライアントトラフィックの再ルーティングなどの責任を負います。

John Smith は、新しいハードウェアやアプリケーションの変更を考慮して、四半期ごとに計画を見直し、更新する予定です。年に一度、3 つのサイトすべてが、Disaster Plan C の手順に従って、他の 2 つのサイトのリカバリーとデプロイメントの手順を実行しなければなりません。

付録A LDAP クライアントツールの使用

Red Hat Directory Server は、OpenLDAP で提供される LDAP ツール (ldapsearch や ldapmodify など) を使用します。OpenLDAP ツールオプションは、OpenLDAP man ページ http://www.openldap.org/software/man.cgi で説明されています。

この付録では、これらの LDAP ツールを使用する際の一般的な使用例を紹介しています。

ldapsearch を使用するためのより広範な例は、「14章ディレクトリーエントリーの検索」に記載されています。ldapmodify および ldapdelete を使用する例は「「コマンドラインでエントリーの管理」」に記載されています。

A.1. 延長操作の実行

Red Hat Directory Server は、特に拡張検索操作など、さまざまな拡張操作をサポートします。拡張操作は、LDAP 操作と共に追加の操作 (get effective rights 検索やサーバー側のソートなど) を渡します。同様に、LDAP クライアントは、多くの拡張操作に対応する可能性があります。

OpenLDAP LDAP ツールは、2 つの方法で拡張された操作をサポートします。すべてのクライアントツール (ldapmodify、ldapsearch など) は、-e オプションまたは -E オプションのいずれかを使用して拡張操作を送信します。-e 引数は、任意の OpenLDAP クライアントツールと使用でき、パスワードポリシーの処理方法など、操作に関する一般的な指示を送信できます。-E は、ldapsearch でのみ使用され、GER 検索、ソート、ページ情報などのより有用な制御を渡し、その他情報 (not-explicitly-support 拡張操作など) に関する情報を渡します。

さらに、OpenLDAP には別のツール ldapexop があります。これは ldapsearch -E を実行するのと同じように、拡張検索操作を行うためだけに使用されます。

ldapsearch の拡張操作の形式は、通常、以下のとおりです。

```
-E extended_operation_type=operation_parameters
```

拡張操作が OpenLDAP ツールで明示的に処理される場合、extended_operation_type は、逆参照検索の deref やサーバー側ソートの sss のようなエイリアスになります。サポートされる拡張操作では、出力がフォーマットされています。GER 検索などの他の拡張操作は、エイリアスではなく OID を使用して渡されるため、extended_operation_type は OID になります。サポートされていない操作の場合、ツールはサーバーからの応答を認識しないため、出力は形式化されません。

たとえば、pg 拡張操作タイプは、結果を簡易なページの結果に形式化します。

```
# ldapsearch -x -D "cn=Directory Manager" -W -b
"ou=Engineers,ou=People,dc=example,dc=com" -E pg=3 "(objectclass=*)" cn

dn: uid=jsmith,ou=Engineers,ou=People,dc=example,dc=com
  cn: John Smith

dn: uid=bjensen,ou=Engineers,ou=People,dc=example,dc=com
  cn: Barbara Jensen

dn: uid=hmartin,ou=Engineers,ou=People,dc=example,dc=com
  cn: Henry Martin

Results are sorted.
next page size (3): 5
```

ldapexop の同じ操作は、ページングされた簡易な結果操作の OID のみを使用して実行できます (ページごとに 3 つの結果)。

```
ldapexop 1.2.840.113556.1.4.319=3
```

ただし、ldapexop は ldapsearch が実行する検索パラメーターの範囲を受け入れず、柔軟性が低下します。

A.2. エントリーの比較

ldapcompare は、エントリーをチェックして、指定したエントリーに特定値の属性が含まれているかどうかを確認します。たとえば、このチェックは、エントリーに sn 値が Smith であるかを確認します。

```
# ldapcompare -D "cn=Directory Manager" -W -p 389 -h server.example.com -x sn:smith
uid=bjensen,ou=people,dc=example,dc=com
comparing type: "sn" value: "smith" in entry "uid=bjensen,ou=people,dc=example,dc=com"
compare FALSE

ldapcompare -D "cn=Directory Manager" -W -p 389 -h server.example.com -x sn:smith
uid=jsmith,ou=people,dc=example,dc=com
comparing type: "sn" value: "smith" in entry "uid=jsmith,ou=people,dc=example,dc=com"
compare TRUE
```

compare 属性は以下の 3 つの方法のいずれかで指定できます。

- コマンドラインで直接渡される単一の attribute:value 文

  ```
  sn:Smith
  ```

- jpegPhoto などの属性、または証明書または CRL を検証するために、コマンドラインで渡される単一の attribute::base64value ステートメント

  ```
  jpegPhoto:dkdkPDKCDdko0eiofk==
  ```

- 属性の比較値の一覧を含むファイルを参照する attribute:file 文およびスクリプトは、リストを繰り返し処理します。

  ```
  postalCode:/tmp/codes.txt
  ```

compare 操作自体は、特定のエントリーまたはエントリーのグループに対して実行する必要があります。単一のエントリー DN をコマンドラインで渡すか、-f オプションを使用して指定することのできる DN の一覧を指定します。

例A.1 1 つの属性値と 1 つのエントリーの比較

属性と値の比較と DN の両方はスクリプトで渡されます。

```
ldapcompare -D "cn=Directory Manager" -W -p 389 -h server.example.com -x sn:smith
uid=jsmith,ou=people,dc=example,dc=com
comparing type: "sn" value: "smith" in entry "uid=jsmith,ou=people,dc=example,dc=com"
compare TRUE
```

例A.2 ファイルからのリスト属性値の比較

まず、可能な sn 値のファイルを作成します。

```
jensen
johnson
johannson
jackson
jorgenson
```

次に、エントリーのリストを作成して、値を比較します。

```
uid=jen200,ou=people,dc=example,dc=com
uid=dsj,ou=people,dc=example,dc=com
uid=matthewjms,ou=people,dc=example,dc=com
uid=john1234,ou=people,dc=example,dc=com
uid=jack.son.1990,ou=people,dc=example,dc=com
```

次に、スクリプトを実行します。

```
# ldapcompare -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
sn:/tmp/surnames.txt -f /tmp/names.txt
comparing type: "sn" value: "jensen" in entry "uid=jen200,ou=people,dc=example,dc=com"
compare TRUE
```

A.3. パスワードの変更

ldappasswd コマンドは、新しいユーザー定義のパスワードを設定したり、アカウントの新しいパスワードを生成したりできます。表19.1「ldappasswd オプション」 は、コマンドラインを使用してパスワードを設定するための最も重要なパラメーターを一覧表示します。その他の設定 (バインド情報、接続情報、その他のコマンド設定など) は必要なことがあり、OpenLDAP の man ページに記載されています。

```
# ldappasswd -x -D bind_dn -W -p server_port -h server_hostname [-A | -a oldPassword] [-S | -s
newPassword] [user]
```

重要

パスワードの変更操作は、TLS、Start TLS、SASL などのセキュアな接続で実行する必要があります。LDAP クライアントに TLS を設定する方法は、「証明書を使用した認証」を参照してください。

lda passwd のパスワード操作関連のパラメーターの一覧は、表19.1「ldappasswd オプション」 を参照してください。

例A.3 Directory Manager が TLS を介してユーザーのパスワードを変更

Directory Manager は、uid=tuser1,ou=People,dc=example,dc=com ユーザーのパスワードを TLS 経由で new_password に変更します。

```
# ldappasswd -D "cn=Directory Manager" -W -ZZ -p 389 -h server.example.com -x -s
new_password "uid=tuser1,ou=People,dc=example,dc=com"
```

例A.4 ユーザーのパスワードを生成する Directory Manager

Directory Manager は、TLS 経由で uid=tuser2,ou=People,dc=example,dc=com ユーザーのパスワードを生成します。

```
# ldappasswd -D "cn=Directory Manager" -W -ZZ -p 389 -h server.example.com -x
"uid=tuser2,ou=People,dc=example,dc=com"
```

例A.5 ユーザーが自分のパスワードを変更

ユーザー tuser3 は、パスワードを TLS 経由で old_newpassword から new_password に変更します。

```
# ldappasswd -p 389 -h server.example.com -ZZ -x -D
"uid=tuser3,ou=People,dc=example,dc=com" -W -a old_password -s new_password
```

例A.6 DIGEST_MD5 によるユーザー認証および自身のパスワードの変更

ユーザー jsmith が、GSS-API で認証し、パスワードを new_password *に変更します。*

```
# ldappasswd -p 389 -h server.example.com -O noplain,minssf=1,maxbufsize=512 -Y
GSSAPI -U "dn:uid=jsmith,ou=people,dc=example,dc=com" -R EXAMPLE.COM -W -s
new_password
```

例A.7 新しいパスワードに対して Kerberos プロンプトにより認証されるユーザー権限

Kerberos によって認証済みのユーザーにより、新しいパスワードが要求されます。これは TLS 上では実行されません。

```
# ldappasswd -p 389 -h server.example.com -O noplain,minssf=1,maxbufsize=512 -I
```

A.4. LDAP URL の生成

LDAP URL は、さまざまな設定エリアおよび操作で使用されます。参照元およびチェーン、レプリケーション、同期、ACI、ならびにインデックスは開始リストで使用します。間違った URL が間違ったサーバーに接続するか、または単に操作が失敗する可能性があるため、正確な LDAP URL を構築す

ることは重要です。さらに、すべての OpenLDAP ツールでは、-H オプションで、他の接続情報 (ホスト名、ポート、サブツリー、検索ベースなど) の代わりに LDAP URL を渡すことができます。

注記

LDAP URL は、「付録C LDAP URL」で説明されています。

ldapurl コマンドは、以下の 2 つの方法で URL を管理します。

- *指定された LDAP URL をその構成要素に分解*
- *指定要素から新しくて有効な LDAP URL を作成*

URL を操作するパラメーターは、表A.1「ldapurl パラメーター」に一覧表示されています。パラメーターの一覧は OpenLDAP の man ページにあります。

表A.1 ldapurl パラメーター

オプション	説明
URL の分解の場合	
-H "URL"	LDAP URL を渡して要素に分割します。
URL の構築の場合	
-a attributes	検索結果で特に返されるコンマ区切りの属性を指定します。
-b base	URL の検索ベースまたはサブツリーを設定します。
-f filter	使用する検索フィルターを設定します。
-h hostname	Directory Server のホスト名を指定します。
-p port	Directory Server のポートを指定します。
-S ldap|ldaps|ldapi	**ldap**、**ldaps**、**ldapi** など、接続に使用するプロトコルを指定します。

オプション	説明
-s scope	検索条件を指定します。

例A.8 LDAP URL の無効化

ldapurl は、-H オプションを使用して既存の LDAP URL にフィードし、このツールは neat リストの URL の要素を返します。

```
# ldapurl -H "ldap:///:389/dc=example,dc=com?cn,sn?sub?(objectclass=inetorgperson)"
scheme: ldap
port: 389
dn: dc=example,dc=com
selector: cn
selector: sn
scope: sub
filter: (objectclass=inetorgperson)
```

例A.9 LDAP URL の構築

ldapurl の最も便利なアプリケーションは、有効な LDAP URL を手動で構築することです。Directory Server コンソールには、ACI や参照などのエリアに有効な URL を開発するためのツールがありますが、非常に複雑な設定やスクリプト化された操作には、管理者が URL を手動で構築する必要がある場合があります。ldapurl を使用すると、URL が有効であることを確認します。

ldapurl は、検索ベース、スコープ、および属性に対して、すべての LDAP クライアントツールの通常の接続パラメーターと追加の ldapsearch 引数を受け入れますが、このツールは Directory Server インスタンスに接続しないため、バインド情報は必要ありません。接続を受け入れ、検索設定を受け入れ、それらを要素として URL に提供します。

```
ldapurl -a cn,sn -b dc=example,dc=com -s sub -f "(objectclass=inetorgperson)"

ldap:///:389/dc=example,dc=com?cn,sn?sub?(objectclass=inetorgperson)
```

付録B LDAP データ交換形式

Red Hat Directory Server (Directory Server) は、LDAP データ交換形式 (LDIF) を使用して、ディレクトリーおよびディレクトリーエントリーをテキスト形式で記述します。LDIF は、初期ディレクトリーデータベースの構築や、多数のエントリーを一度にディレクトリーに追加するために使用されます。さらに、LDIF はディレクトリーエントリーの変更を説明するのにも使用されます。このため、Directory Server のコマンドラインユーティリティーのほとんどは、入力または出力のいずれかに LDIF を使用します。

LDIF はテキストファイル形式であるため、LDIF は実質的には任意の言語を使用して作成できます。すべてのディレクトリーデータは Unicode の UTF-8 エンコーディングを使用して保存されます。そのため、作成される LDIF ファイルも UTF-8 でエンコードする必要があります。

LDIF を使用してディレクトリーエントリーを変更する方法は、「3章ディレクトリーエントリーの管理」を参照してください。

B.1. LDIF ファイルの形式の概要

LDIF は、空白行で区切られた 1 つ以上のディレクトリーエントリーで構成されます。各 LDIF エントリーは、任意のエントリー ID、必須の識別名、複数のオブジェクトクラス、および複数の属性定義で構成されます。

LDIF 形式は、RFC 2849 の 『The LDAP Data Interchange Format (LDIF)』 で定義されています。Directory Server はこの規格に準拠しています。

LDIF で表されるディレクトリーエントリーの基本的な形式は次のとおりです。

```
dn: distinguished_name
objectClass: object_class
objectClass: object_class
...
 attribute_type[;subtype]:attribute_value
...
```

- すべての LDIF エントリーには、DN と 1 つ以上のオブジェクトクラス定義が必要です。
- エントリーに定義されるオブジェクトクラスで必要な属性を含めます。

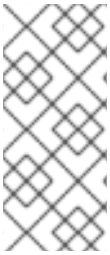
- その他の属性およびオブジェクトクラスはすべて任意です。
- オブジェクトクラスおよび属性は任意の順序で指定できます。
- コロン後のスペースは任意です。

表B.1「LDIF フィールド」では、前の定義で示された LDIF フィールドを説明します。

表B.1 LDIF フィールド

フィールド	定義
[id]	任意。エントリー ID を表す正の 10 進数。データベース作成ツールは、この ID を自動的に生成します。この値は独自に追加したり、編集したりしないでください。
dn: distinguished_name	エントリーの識別名を指定します。
objectClass: object_class	このエントリーで使用するオブジェクトクラスを指定します。オブジェクトクラスは、エントリーに使用可能な属性のタイプ (スキーマ) を識別します。標準オブジェクトクラスの一覧は『、Red Hat Directory Server 10 の設定、コマンド、およびファイルリファレンス』を参照してください。スキーマのカスタマイズに関する情報は、「12章ディレクトリースキーマの管理」を参照してください。
attribute_type	エントリーで使用する説明的な属性を指定します。属性はスキーマで定義する必要があります。標準『属性の一覧は、Red Hat Directory Server 10 の設定、コマンド、およびファイルリファレンス』を参照してください。スキーマのカスタマイズに関する情報は、「12章ディレクトリースキーマの管理」を参照してください。

フィールド	定義
[subtype]	任意。サブタイプ、言語、バイナリー、または発音を指定します。このタグを使用して、対応する属性値が表現されている言語や、属性値がバイナリーであるか、属性値の発音であるかを識別します。属性サブタイプに関する詳細は、「属性サブタイプの追加」を参照してください。サポートされるサブタイプタグの一覧は、表D.1「サポートされる言語サブタイプ」を参照してください。
attribute_value	属性タイプと使用する属性値を指定します。



注記

ディレクトリーのエントリーへの変更を表す LDIF 構文は、表B.1「LDIF フィールド」で説明されている構文とは異なります。LDIF を使用してディレクトリーエントリーを変更する方法は、「3章ディレクトリーエントリーの管理」を参照してください。

B.2. LDIF での行継続

LDIF ファイルでは、行の継続部分をスペース 1 つ分インデントすることで、行を分割して継続する (折りたたみ と呼ばれる) ことができます。たとえば、以下の 2 つのステートメントは同じです。

```
dn: cn=Jake Lupinski,dc=example,dc=com

dn: cn=Jake Lup
 inski,dc=exa
 mple,dc=com
```

LDIF 行を分割して継続する必要はありません。ただし、それを行うことで LDIF ファイルの読みやすさが向上することがあります。通常の規則では、LDIF ファイルには 78 列を超えるテキストが含まれません。

B.3. バイナリーデータの表現

JPEG イメージなどのバイナリーデータは、標準の LDIF 表記または base-64 エンコードの 2 つの方法のいずれかを使用して LDIF で表されます。

B.3.1. 標準の LDIF 表記

標準の LDIF 表記は、データがバイナリーであることを示すために、(<) より小さい記号を使用しま

す。以下に例を示します。

```
jpegphoto: < file:/path/to/photo
```

この標準表記では、ldapmodify -b パラメーターを指定する必要はありません。ただし、標準の表記法では、LDIF ファイルまたは LDIF 更新ステートメントの先頭に以下の行を追加する必要があります。

```
version: 1
```

以下に例を示します。

```
# ldapmodify -x -D userDN -W

version: 1
dn: cn=Barney Fife,ou=People,dc=example,dc=com
changetype: modify
add: usercertificate
usercertificate;binary: < file: BarneysCert
```

B.3.2. Base-64 でエンコード

バイナリデータを Base-64 に変換して LDIF ファイルにすることで、画像から TLS 証明書までさまざまなデータに対応します。ベース 64 でエンコードされたデータは、:: 記号を使用して識別されます。以下に例を示します。

```
jpegPhoto::encoded_data
```

バイナリーデータの他に、base-64 でエンコードする必要のあるその他の値には以下が含まれます。

- *コロン (:) またはスペースで始まる値。*
- *ASCII 以外のデータを含む値 (新しい行を含む)。*

-b パラメーターで ldif コマンドラインユーティリティーを使用して、バイナリーデータを LDIF 形式に変換します。

```
# ldif -b attribute_name
```

attribute_name は、バイナリーデータを指定する属性の名前です。バイナリーデータは標準入力から読み取られ、結果は標準出力に書き込まれます。そのため、入力ファイルと出力ファイルの選択にはリダイレクト演算子を使用します。

ldif コマンドラインユーティリティーは入力を取得し、これを正しい行継続性と適切な属性情報で形式化します。ldif ユーティリティーは、入力に base-64 エンコーディングが必要かどうかも評価します。以下に例を示します。

```
# ldif -b jpegPhoto <  mark.jpg > out.ldif
```

この例では、JPEG 形式のイメージを含むバイナリーファイルを取得し、属性 jpegPhoto の LDIF 形式に変換します。出力は out.ldif に保存されます。

-b オプションは、ldif ユーティリティーが入力全体を単一のバイナリー値として解釈するように指定します。-b が存在しない場合は、各行が個別の入力値であると見なされます。

B.4. LDIF を使用したディレクトリーエントリーの指定

多くのエントリーはディレクトリーに保存できます。本セクションでは、ディレクトリーで使用する最も一般的なエントリーの 3 つ (ドメイン、組織単位、および組織の人のエントリー) を 3 つにまとめる方法を説明します。

エントリーに定義されているオブジェクトクラスは、エントリーがドメインまたはドメインコンポーネント、組織単位、組織人、またはその他のタイプのエントリーを表しているかどうかを示すものです。デフォルトでディレクトリーで使用できるオブジェクトクラスの完全一覧と、最も一般的に使用される属性の一覧は、『Red Hat Directory Server 10 Configuration, Command, and File Referenceを参照してください』。

B.4.1. ドメインエントリーの指定

ディレクトリーにはドメインエントリーが 1 つ以上含まれます。通常、これはディレクトリの最初の、つまり一番上のエントリーです。ドメインエントリーは、多くの場合、ディレクトリーの DNS ホストおよびドメイン名に対応します。たとえば、Directory Server ホストが ldap.example.com と呼ばれている場合は、ディレクトリーのドメインエントリーは dc=ldap,dc=example,dc=com または単に dc=example,dc=com という名前になります。

ドメインの定義に使用される LDIF エントリーは以下のようになります。

```
dn: distinguished_name
```

```
objectClass: top
objectClass: domain
dc: domain_component_name
 list_of_optional_attributes
...
```

以下は、LDIF 形式のドメインエントリーの例です。

```
dn: dc=example,dc=com
objectclass: top
objectclass: domain
dc: example
description: Fictional example company
```

LDIF フォーマットのドメインエントリーの各要素が 表B.2「ドメインエントリーの LDIF 要素」に定義されています。

表B.2 ドメインエントリーの LDIF 要素

LDIF 要素	説明
dn: distinguished_name	必須。エントリーの識別名を指定します。
objectClass: top	必須。top オブジェクトクラスを指定します。
objectClass: domain	domain オブジェクトクラスを指定します。この行は、エントリーをドメインまたはドメインコンポーネントとして定義します。このオブジェクトクラスで利用可能な属性の一覧は、『Red Hat Directory Server 10 Configuration, Command, and File Referenceを参照してください』。
dc: domain_component	ドメイン名を指定する属性。サーバーは通常、dc=hostname,dc=domain,dc=toplevel の形式で接尾辞または命名コンテキストを持つように初期設定時に設定されます。たとえば、dc=ldap,dc=example,dc=com になります。ドメインエントリーは、dc: ldap のように左端の dc の値を使用する必要があります。サフィックスが dc=example,dc=com の場合、dc の値は dc: example になります。サーバーが接尾辞を使用するよう設定されていない場合は、dn: dc=com のエントリーを作成しないでください。

LDIF 要素	説明
list_of_attributes	エントリーに保持する任意の属性の一覧を指定します。このオブジェクトクラスで利用可能な属性の一覧は、『Red Hat Directory Server 10 Configuration, Command, and File Referenceを参照してください』。

B.4.2. 組織単位エントリーの指定

組織ユニットエントリーは、ディレクトリーツリー内のメジャーブランチポイントまたはサブディレクトリーを表すために使用されます。これらのサブツリーは、企業内の主要な、適度に静的なエンティティーに対応しています。例えば、ユーザーを含むサブツリーや、グループを含むサブツリーなどです。

エントリーに含まれる組織単位属性は、マーケティングやエンジニアリングなど、企業内の主要な組織を表す場合もあります。ただし、このスタイルは推奨されません。Red Hat はフラットなディレクトリーツリーの使用を強く推奨しています。

通常は、ディレクトリーツリー内に、複数の組織単位またはブランチがあります。

組織単位エントリーを定義する LDIF は、以下のように表示される必要があります。

```
dn: distinguished_name
objectClass: top
objectClass: organizationalUnit
ou: organizational_unit_name
 list_of_optional_attributes
...
```

以下は、LDIF 形式の組織単位エントリーの例です。

```
dn: ou=people,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
ou: people
description: Fictional example organizational unit
```

表B.3「組織単位エントリーの LDIF 要素」は、LDIF 形式の組織単位エントリーの各要素を定義します。

表B.3 組織単位エントリーの LDIF 要素

LDIF 要素	*説明*
dn: **distinguished_name**	*エントリーの識別名を指定します。DN が必要です。DN にコンマがある場合、コンマはバックスラッシュ (\) でエスケープする必要があります (例: dn: ou=people,dc=example,dc=com)。*
objectClass: top	*必須。top オブジェクトクラスを指定します。*
objectClass: organizationalUnit	*organizationalUnit オブジェクトクラスを指定します。この行は、エントリーを organizational unit として定義します。このオブジェクトクラスで利用可能な属性の一覧は、* 『Red Hat Directory Server 10 Configuration, Command, and File Referenceを参照してください』。
ou: **organizational_unit_name**	*組織単位の名前を指定する属性。*
list_of_attributes	*エントリーに保持する任意の属性の一覧を指定します。このオブジェクトクラスで利用可能な属性の一覧は、* 『Red Hat Directory Server 10 Configuration, Command, and File Referenceを参照してください』。

B.4.3. 組織の個人エントリーの指定

ディレクトリー内のエントリーの大半は、組織の人を表します。

LDIF では、組織担当者の定義は以下のようになります。

```
dn: distinguished_name
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: common_name
sn: surname
 list_of_optional_attributes
```

以下は、LDIF 形式の組織単位エントリーの例です。

```
dn: uid=bjensen,ou=people,dc=example,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Babs Jensen
sn: Jensen
givenname: Babs
uid: bjensen
ou: people
description: Fictional example person
telephoneNumber: 555-5557
userPassword: {SSHA}dkfljlk34r2kljdsfk9
```

表B.4「Person エントリーの LDIF 要素」は、LDIF 担当者エントリーの各要素を定義します。

表B.4 Person エントリーの LDIF 要素

LDIF 要素	説明
dn: distinguished_name	*必須。エントリーの識別名を指定します。たとえば、dn: uid=bjensen,ou=people,dc=example,dc=com になります。DN にコンマがある場合は、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。*
objectClass: top	*必須。top オブジェクトクラスを指定します。*
objectClass: person	*person オブジェクトクラスを指定します。多くの LDAP クライアントは、ユーザーや組織のユーザーの検索操作時にこれを必要とするため、このオブジェクトクラス仕様を含める必要があります。*
objectClass: organizationalPerson	*organizationalPerson オブジェクトクラスを指定します。一部の LDAP クライアントは、組織ユーザーの検索操作時に必要であるため、このオブジェクトクラス仕様を含める必要があります。*

LDIF 要素	説明
objectClass: inetOrgPerson	inetOrgPerson オブジェクトクラスを指定します。このオブジェクトクラスには属性の範囲が広がるため、このオブジェクトクラスには組織の人エントリーの作成には、inetOrgPerson オブジェクトクラスが推奨されます。uid 属性はこのオブジェクトクラスで必要で、このオブジェクトクラスが含まれるエントリーは uid 属性の値に基づいて名前が付けられます。このオブジェクトクラスで利用可能な属性の一覧は、『Red Hat Directory Server 10 Configuration, Command, and File Referenceを参照してください』。
cn: common_name	担当者の一般的な名前を指定します。これは、担当者が一般的に使用するフルネームです。たとえば、cn: Bill Anderson のようになります。少なくとも 1 つの共通名が必要です。
sn: 姓	ユーザーの姓を指定します。たとえば、sn: Anderson のようになります。姓が必要です。
list_of_attributes	エントリーに保持する任意の属性の一覧を指定します。このオブジェクトクラスで利用可能な属性の一覧は、『Red Hat Directory Server 10 Configuration, Command, and File Referenceを参照してください』。

B.5. LDIF を使用したディレクトリーの定義

ディレクトリー全体の内容は、LDIF を使用して定義できます。LDIF の使用は、ディレクトリーに追加するエントリーが多数ある場合に、ディレクトリー作成の効率的な方法です。

LDIF を使用してディレクトリーを作成するには、以下を実行します。

1. LDIF 形式に追加するエントリーを含む ASCII ファイルを作成します。

   各エントリーは、空の行で次のエントリーと区切られていることを確認してください。エントリーの間に 1 行のみを使用し、ファイルの最初の行が空白ではないことを確認します。そうでない場合は、ldapmodify ユーティリティーが終了します。詳細は「LDIF を使用したディレクトリーエントリーの指定」を参照してください。

2.
各ファイルは、データベースの一番上のエントリー (ルート) から始めます。

root エントリーは、データベースに含まれる接尾辞または部分接尾辞を表す必要があります。たとえば、データベースに dc=example,dc=com の接尾辞がある場合、ディレクトリーの最初のエントリーは dn: dc=example,dc=com でなければなりません。

接尾辞の詳細は、『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス』 に記載されている「Suffix」パラメーターを参照してください。

3.
LDIF ファイルのブランチポイントを表すエントリーが、そのブランチで作成するエントリーの前に配置されていることを確認します。

たとえば、エントリーをユーザーやグループのサブツリーに置くには、それらのサブツリー内にエントリーを作成する前に、そのサブツリーの分岐点を作成します。

注記

LDIF ファイルは順番に読み込まれるため、親エントリーが子エントリーの前にある必要があります。

4.
以下の方法のいずれかを使用して、LDIF ファイルからディレクトリーを作成します。

- Directory Server コンソールでデータベースの初期化 インポートするデータベースの規模が小さい (10,000 エントリーより少ない) 場合は、この方法を使用します。「コンソールからのデータベースのインポート」 を参照してください。

警告

このメソッドは破壊的で、接尾辞の既存のデータを削除します。

- ldif2db または ldif2db.pl コマンドラインユーティリティー。インポートするデータベースの規模が大きい (10,000 エントリーより多い) 場合は、この方法を使用しま

す。「ldif2db コマンドラインユーティリティーを使用したインポート」を参照してください。

- ldif2db サーバーが実行している場合のみ使用できます。
- ldif2db.pl サーバーが実行されている場合のみ使用できます。

警告

このメソッドは破壊的で、接尾辞の既存のデータを削除します。

- ldapmodify コマンドラインユーティリティーに -a パラメータをつけたものです。この方法は、既存のデータベースに新しいサブツリーを追加する場合や、削除してはいけない既存のデータが接尾辞にある場合に使用します。LDIF ファイルからディレクトリーを作成する他の方法とは異なり、ldapmodify を使用してサブツリーを追加する前に Directory Server を実行する必要があります。「エントリーの追加」を参照してください。

例B.1 LDIF ファイルの例

この LDIF ファイルには、1 つのドメイン、2 つの組織単位、および 3 つの組織人のエントリーが含まれます。

```
dn: dc=example,dc=com
objectclass: top
objectclass: domain
dc: example
description: Fictional example domain

dn: ou=People,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
ou: People
description: Fictional example organizational unit
tel: 555-5559

dn: cn=June Rossi,ou=People,dc=example,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
```

```
cn: June Rossi
sn: Rossi
givenName: June
mail: rossi@example.com
userPassword: {sha}KDIE3AL9DK
ou: Accounting
ou: people
telephoneNumber: 2616
roomNumber: 220

dn: cn=Marc Chambers,ou=People,dc=example,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: Marc Chambers
sn: Chambers
givenname: Marc
mail: chambers@example.com
userPassword: {sha}jdl2alem87dlacz1
telephoneNumber: 2652
ou: Manufacturing
ou: People
roomNumber: 167

dn: cn=Robert Wong,ou=People,example.com Corp,dc=example,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: Robert Wong
cn: Bob Wong
sn: Wong
givenname: Robert
givenname: Bob
mail: bwong@example.com
userPassword: {sha}nn2msx761
telephoneNumber: 2881
roomNumber: 211
ou: Manufacturing
ou: people

dn: ou=Groups,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
ou: groups
description: Fictional example organizational unit
```

B.6. 複数の言語での情報の保存

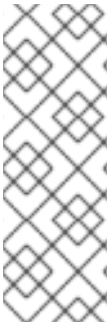
ディレクトリーに単一の言語が含まれる場合は、ディレクトリーに新しいエントリーを追加するための特別な作業を行う必要はありません。ただし、組織が多国籍である場合は、異なるロケールのユー

ザーが自分の言語でディレクトリー情報を閲覧できるように、情報を複数の言語で保存する必要があるかもしれません。

ディレクトリー内の情報が複数の言語で表される場合、サーバーは言語タグを属性値に関連付けます。新しいエントリーが追加されると、RDN (相対識別名、命名属性) で使用される属性値は、言語コードなしで指定する必要があります。

複数の言語を 1 つの属性に保存できます。この場合、属性タイプは同じですが、各値には異なる言語コードがあります。

Directory Server がサポートする言語とその関連する言語タグの一覧は、「サポート対象のロケール」を参照してください。



注記

言語タグは、文字列がディレクトリー内にどのように保存されるかには影響しません。すべてのオブジェクトクラスおよび属性文字列は UTF-8 を使用して保存されます。ユーザーは、LDIF で使用されるデータを UTF-8 に変換します。ほとんどのオペレーティングシステムが提供する iconv または uconv コマンドを使用して、ネイティブ文字セットからデータを UTF-8 に変換できます。

たとえば、アメリカとフランスにオフィスを持つ Example Corporation は、社員が母国語でディレクトリー情報を閲覧できるようにしたいと考えています。ディレクトリーエントリーを追加すると、ディレクトリー管理者は英語とフランス語の両方で属性値を指定します。新規従業員 Babs Jensen のディレクトリーエントリーを追加する際に、管理者は以下を行います。

1. 管理者が、フランスの所在地住所値で street.txt ファイルを作成します。

```
1 rue de l'Université
```

2. ファイルのコンテンツは UTF-8 に変換されます。

```
# iconv -t UTF-8 -o output.txt street.txt
```

3. 以下の LDIF エントリーは、streetAddress;lang-fr の street アドレス値の UTF-8 値を使用して作成されます。

```
dn: uid=bjensen,ou=people,dc=example,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:: AasljdoaAJASl023909jaASJaonasd0ADS
preferredLanguage: fr
```

属性名とサブタイプの後にコロンは、値がバイナリーの base-64 でエンコードされたことを示しています。

推奨言語が英語に設定されている LDAP クライアントを使用して、このディレクトリーエントリーにアクセスすると、1 University Street アドレスが表示されます。設定言語がフランス語に設定された LDAP クライアントを持つディレクトリーにアクセスすると、アドレス 1 rue de l'Université が表示されます。

付録C LDAP URL

LDAP URL は、サイト URL が特定の Web サイトまたは Web ページを特定するのと同様に、Red Hat Directory Server インスタンスを特定します。Directory Server インスタンスの LDAP URL が使用される場合は、以下の 3 つの一般的な時間になります。

- LDAP URL は、Web ベースのクライアントを使用して Directory Server にアクセスする際に特定の Directory Server インスタンスを特定するために使用されます。
- LDAP URL は Directory Server の参照を設定するのに使用されます。
- LDAP URL はアクセス制御の指示を設定するのに使用されます。

注記

LDAP URL 形式は RFC 4516 で説明されています。http://www.ietf.org/rfc/rfc4516.txt を参照してください。

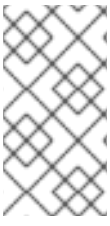
C.1. LDAP URL のコンポーネント

LDAP URL の構文は以下のとおりです。

```
ldap[s]://hostname:port/base_dn?attributes?scope?filter
```

ホスト名の代わりに IPv4 アドレスまたは IPv6 アドレスを使用することもできます。

ldap:// プロトコルは、セキュアでない接続で LDAP サーバーへの接続に使用されます。ldaps:// プロトコルは、TLS 接続を介して LDAP サーバーに接続するために使用されます。表C.1「LDAP URL コンポーネント」は、LDAP URL のコンポーネントを一覧表示します。



注記

LDAP URL 形式は RFC 4516 で説明されています。http://www.ietf.org/rfc/rfc4516.txt を参照してください。

表C.1 LDAP URL コンポーネント

コンポーネント	説明
ホスト名	LDAP サーバーの名前（または IPv4 アドレスまたは IPv6 アドレス）たとえば、ldap .example.com または 192.0.2.90 です。
ポート	LDAP サーバーのポート番号 (例: 696)。ポートが指定されていない場合は、標準の LDAP ポート (389) または LDAPS ポート (636) が使用されます。
base_dn	ディレクトリー内のエントリーの識別名 (DN)。この DN は、検索の開始点であるエントリーを特定します。ベース DN が指定されていない場合、検索はディレクトリーツリーのルートで始まります。
attributes	返される属性。複数の属性を指定するには、コンマを使用して属性を区切ります (例: cn,mail,telephoneNumber)。URL に属性が指定されていない場合は、すべての属性が返されます。
scope	検索の範囲で、以下のいずれかの値になります。 base は、URL に指定された識別名 (base_dn) に関する情報のみを取得します。 one は、URL に指定された識別名 (base_dn) より 1 レベル下のエントリーに関する情報を取得します。ベースエントリーはこのスコープに含まれません。 sub は、URL に指定された識別名 (base_dn) より下のすべてのレベルのエントリーに関する情報を取得します。ベースエントリーはこのスコープに含まれます。 スコープが指定されていない場合、サーバーは base 検索を実行します。
filter	指定された検索範囲内のエントリーに適用する検索フィルター。フィルターを指定しないと、サーバーはフィルター (objectClass=*) を使用します。

属性、スコープ、およびフィルターコンポーネントは URL の位置によって識別されます。属性が指定されていない場合でも、そのフィールドを区切るために疑問符を含める必要があります。

たとえば、(sn=Jensen) を一致させるエントリーの属性をすべて返す dc=example,dc=com から開始するサブツリー検索を指定するには、以下の LDAP URL を使用します。

> *ldap://ldap.example.com/dc=example,dc=com??sub?(sn=Jensen)*

2 つの連続したクエスチョンマーク (??) は、属性が指定されていないことを示します。URL には特定の属性が指定されていないため、検索ではすべての属性が返されます。

C.2. 不安全文字のエスケープ

URL の 不安全 文字はすべてエスケープするか、特殊文字に置き換える必要があります。

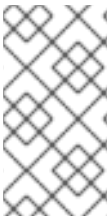
たとえば、空白文字は、URL 内で %20 として表す必要がある不安全な文字です。そのため、識別名 o=example.com corporation は o=example.com%20corporation としてエンコードされる必要があります。

以下の表は、URL 内で不安全とみなされる文字を一覧表示し、不安全な文字の代わりに使用する関連エスケープ文字を提供します。

不安全な文字	エスケープ文字
space	%20
<	%3c
>	%3e
"	%22
#	%23
%	%25
{	%7b
}	%7d
|	%7c
\	%5c
^	%5e
~	%7e

不安全な文字	エスケープ文字
[	%5b
]	%5d
`	%60

C.3. LDAP URL の例



注記

LDAP URL 形式は RFC 4516 で説明されています。http://www.ietf.org/rfc/rfc4516.txt を参照してください。

例 1

以下の LDAP URL は、識別名 dc=example,dc=com のエントリーのベース検索を指定します。

```
ldap://ldap.example.com/dc=example,dc=com
```

- *ポート番号が指定されていないため、標準の LDAP ポート番号 (389) が使用されます。*
- *属性が指定されていないため、検索はすべての属性を返します。*
- *検索条件が指定されていないため、検索はベースエントリー dc=example,dc=com に制限されます。*
- *フィルターが指定されていないため、ディレクトリーはデフォルトのフィルター (objectclass=*) を使用します。*

例 2

以下の LDAP URL は、DN dc=example,dc=com を使用してエントリーの postalAddress 属性を取得します。

```
ldap://ldap.example.com/dc=example,dc=com?postalAddress
```

- *検索条件が指定されていないため、検索はベースエントリー dc=example,dc=com に制限されます。*

- *フィルターが指定されていないため、ディレクトリーはデフォルトのフィルター (objectclass=*) を使用します。*

例 3

以下の LDAP URL は、Barbara Jensen のエントリーの cn 属性、mail 属性、および telephoneNumber 属性を取得します。

```
ldap://ldap.example.com/cn=Barbara%20Jensen,dc=example,dc=com?
cn,mail,telephoneNumber
```

- *検索範囲が指定されていないため、検索はベースエントリー cn=Barbara Jensen,dc=example,dc=com に制限されます。*

- *フィルターが指定されていないため、ディレクトリーはデフォルトのフィルター (objectclass=*) を使用します。*

例 4

以下の LDAP URL は、姓 Jensen を持ち、dc=example,dc=com 下のレベルにあるエントリーの検索を指定します。

```
ldap://ldap.example.com/dc=example,dc=com??sub?(sn=Jensen)
```

- *属性が指定されていないため、検索はすべての属性を返します。*

- *検索範囲は sub であるため、検索にはベースエントリー dc=example,dc=com と、ベースエントリー下の全レベルでエントリーが含まれます。*

例 5

以下の LDAP URL は、dc=example,dc=com 下 1 レベル下にあるすべてのエントリーレベルに対するオブジェクトクラスの検索を指定します。

```
ldap://ldap.example.com/dc=example,dc=com?objectClass?one
```

- *検索範囲は one であるため、検索にはベースエントリー dc=example,dc=com の 1 レベル下に全エントリーが含まれます。検索範囲にはベースエントリーが含まれません。*

- *フィルターが指定されていないため、ディレクトリーはデフォルトのフィルター (objectclass=*) を使用します。*

注記

LDAP URL の構文には、認証情報またはパスワードを指定する方法は含まれません。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する場合を除き、LDAP URL から開始された検索要求は認証されます。

付録D 国際化

Red Hat Directory Server を使用すると、ユーザーはさまざまな言語でエントリーとそれに関連する属性を保存、管理、および検索できます。国際化されたディレクトリーは、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にすぐにアクセスできる、貴重な企業リソースになり得ます。

Directory Server は、ディレクトリーデータが UTF-8 に保存されるため、デフォルトですべての国際文字セットをサポートします。さらに、Directory Server では、検索操作で言語の詳細をもとに、指定したマッチングルールと結合順序を使用できます。

注記

ASCII 文字は属性およびオブジェクトクラス名に必要です。

D.1. ローカルの概要

Directory Server は、ロケール を使用して複数の言語をサポートします。ロケールとは、特定の地域、文化、慣習のユーザーがどのようにデータを表示するかについての言語固有の情報を示すもので、ある言語のデータをどのように解釈するか、データをどのようにソートするか、または 照合 するかなどが含まれます。

さらに、ロケール情報は、特定の言語を表すために使用されるコードページを示します。コードページは、オペレーティングシステムがキーボードキーを文字フォント画面表示に関連付けるために使用する内部テーブルです。

具体的には、ロケールは次の 4 つのことを定義します。

- 照合順序。照合順序は、特定の言語の文字をどのようにソートするかについて、言語および文化に固有の情報を提供します。アルファベットの文字の順序、アクセントのある文字とアクセントのない文字を比較する方法、文字列を比較するときに無視できる文字があるかどうかなどを識別します。照合順序では、言語が読み取られる方向 (左から右、右から左、または上と下) など、言語に関する文化固有の情報も考慮されます。

- 文字タイプ。文字タイプは、アルファベット文字を数字またはその他の文字と区別します。たとえば、ある言語では、パイプ (|) 文字は句読点と見なされますが、他の言語ではアルファベットと見なされます。さらに、大文字への大文字のマッピングを定義します。

- 通貨形式。通貨形式は、特定の地域で使用される通貨記号、その記号がその値の前後にある

かどうか、および通貨単位がどのように表されるかを指定します。

- 時刻/日付の形式。時刻と日付の形式は、その地域の時刻と日付の慣習的な形式を示しています。日時形式は、日付が mm/dd/yy (月、日、年) または dd/mm/yy (日、月、年) の形式で、特定の言語の曜日と日付を指定します。たとえば、1996 年 1 月 10 日の日付は、チェコ語では 10. leden 1996、およびフランス語では 10 janvier 1996 と表示されます。

ロケールは、機械的な言語の違いに加えて、文化的、慣習的、地域的な違いを説明するため、ディレクトリーデータは、ユーザーが理解できる特定の言語に翻訳され、特定の地域のユーザーが期待する方法で表示できます。

D.2. サポート対象のロケール

検索操作など、ロケールを指定する必要があるディレクトリー操作を実行する際に、言語タグや照合順序オブジェクト識別子 (OID) を使用します。

言語タグ は、ISO 標準 639 で定義されているように言語を識別する 2 文字の小文字の言語コードで始まる文字列です。言語で地域の違いを区別する必要がある場合、言語タグに ISO 標準 3166 に定義されている国コードの 2 文字の文字列が含まれる可能性があります。言語コードおよび国コードは、ハイフンで区切られています。たとえば、イギリス英語のロケールの特定に使用される言語タグは en-GB です。

オブジェクト識別子 (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 進数の数値です。国際化されたディレクトリーを検索またはインデックス化する OID は、Directory Server でサポートされる特定の照合順序を識別します。たとえば、OID 2.16.840.1.113730.3.3.2.17.1 はフィンランドの照合順序を識別します。

ディレクトリーで国際検索を実行する場合は、言語タグまたは OID を使用して、使用する照合順序を特定します。ただし、国際的な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。インデックスの詳細は、「13章インデックスの管理」を参照してください。

Directory Server でサポートされる言語タグと OID の一覧は、/etc/dirsrv/config/slapd-collations.conf ファイルを参照してください。

D.3. サポートされる言語サブタイプ

言語のサブタイプはクライアントで使用することで、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用に関する詳細は、「属性サブタイプの追加」 を参照してください。表D.1「サ

ポートされる言語サブタイプ」は、Directory Server でサポートされる言語のサブタイプ一覧を表示します。

表D.1 サポートされる言語サブタイプ

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	チェコ語
da	デンマーク語
de	ドイツ語
el	ギリシャ語
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

言語タグ	言語
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
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	スクロバキア語
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	スウェーデン語
uk	ウクライナ語
zh	中国語

D.4. 国際化されたディレクトリーの検索

検索操作の実行時に、Directory Server は、サーバーがサポートする照合順序を持つ任意の言語に基づいて結果をソートすることができます。ディレクトリーでサポートされている照合順序の一覧は、「サポート対象のロケール」を参照してください。

注記

国際化された検索を実行するには、LDAPv3 検索が必要です。したがって、ldapsearch の呼び出しには LDAPv2 オプションを設定しないでください。

本セクションでは、国際的な属性値を返すためのマッチングルールフィルターの使用に焦点を当てています。一般的な ldapsearch 構文の詳細は、「LDAP 検索フィルター」を参照してください。Red Hat コンソールの Users および Groups 部分を使用して国際化されたディレクトリーを検索する方法は、オンラインヘルプを参照してください。

- 「マッチングルールの形式」
- 「サポートされる検索タイプ」
- 「国際検索の例」

D.4.1. マッチングルールの形式

国際化された検索のマッチングルールのフィルターは、いくつかの方法で表すことができ、どちらを使用するかは好みの問題です。

- 検索のベースとなるロケールの照合順序の OID として。
- 検索のベースとなる照合順序に関連付けられた言語タグとして。
- 照合順序の OID および関係演算子を表す接尾辞として。
- 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として。

これらの各オプションの構文は、以下のセクションで説明されています。

- 「マッチングルールでの OID の使用」
- 「マッチングルールに言語タグの使用」
- 「マッチングルールでの OID および Suffix の使用」
- 「一致するルールに対する言語タグと接尾辞の使用」

D.4.1.1. マッチングルールでの OID の使用

Directory Server で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。Directory Server がサポートする OID の一覧は、/etc/dirsrv/config/slapd-collations.conf ファイルを参照してください。

照合順序 OID は、次のように一致規則フィルターの一致規則部分で使用できます。

```
attr:OID:=(relational_operator value)
```

リレーショナル演算子は文字列の値の部分に含まれ、値はシングルスペースで区切られます。たとえば、スウェーデン語の照合順序で N4709 時またはそれ以降のすべての departmentNumber 属性を検索するには、次のフィルターを使用します。

```
departmentNumber:2.16.840.1.113730.3.3.2.46.1:=>= N4709
```

D.4.1.2. マッチングルールに言語タグの使用

Directory Server でサポートされる各ロケールには、関連する言語タグがあります。Directory Server でサポートされる言語タグの一覧は、/etc/dirsrv/config/slapd-collations.conf ファイルを参照してください。

言語タグは、次のようにマッチングルールフィルターのマッチングルール部分で使用できます。

```
attr:language-tag:=(relational_operator value)
```

リレーショナル演算子は文字列の値の部分に含まれ、値はシングルスペースで区切られます。たとえば、スペイン語の照合順序を使用して、estudiante の値を持つすべての説明属性をディレクトリーで検索するには、次のフィルターを使用します。

```
cn:es:== estudiante
```

D.4.1.3. マッチングルールでの OID および Suffix の使用

関係演算子と値のペアを使用する代わりに、フィルターのマッチングルール部分の OID に特定の演算子を表す接尾辞を追加します。以下のように OID と接尾辞を統合します。

```
attr: OID+suffix:=value
```

注記

この構文は、mozldap ユーティリティーでのみサポートされ、ldapsearch などの OpenLDAP ユーティリティーではサポートされません。

たとえば、ドイツ語の照合順序で softwareprodukte 値を持つ businessCategory 属性を選択するには、次のフィルターを使用します。

```
businessCategory:2.16.840.1.113730.3.3.2.7.1.3:=softwareprodukte
```

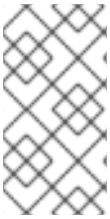
上記の例の .3 は等価接尾辞です。

Directory Server がサポートする OID の一覧は、/etc/dirsrv/config/slapd-collations.conf ファイルを参照してください。リレーショナル演算子とそれらの同等の接尾辞の一覧は、表D.2「検索タイプ、演算子、および接尾辞」を参照してください。

D.4.1.4. 一致するルールに対する言語タグと接尾辞の使用

関係演算子と値のペアを使用する代わりに、フィルターのマッチングルール部分の言語タグに特定の演算子を表す接尾辞を追加します。以下のように、言語タグと接尾辞を組み合わせます。

```
attr: language-tag+suffix:=value
```



注記

この構文は、mozldap ユーティリティーでのみサポートされ、ldapsearch などの OpenLDAP ユーティリティーではサポートされません。

例えば、フランス語の照合順序で La Salle の前後に来るすべての名字を検索するには、次のようなフィルターを使用します。

```
sn:fr.4:=La Salle
```

Directory Server でサポートされる言語タグの一覧は、/etc/dirsrv/config/slapd-collations.conf ファイルを参照してください。リレーショナル演算子とそれらの同等の接尾辞の一覧は、表D.2「検索タイプ、演算子、および接尾辞」を参照してください。

D.4.2. サポートされる検索タイプ

Directory Server は、以下のタイプの国際検索をサポートします。

- *等号 (=)*
- *部分文字列 (*)*
- *より大きい (>)*
- *以上 (>=)*
- *より小さい (<)*
- *以下 (<=)*

近似検索、音声検索、存在検索は英語のみ対応しています。

通常の ldapsearch 検索操作と同様に、国際検索は演算子を使用して検索のタイプを定義します。た

だし、国際的な検索を行う場合は、検索文字列の値の部分に標準的な演算子 (=、>=、>、<、<=) を使用するか、フィルターのマッチングルールの部分に接尾辞 (ディレクトリー接尾辞と混同しないように) と呼ばれる特殊なタイプの演算子を使用してください。表D.2「検索タイプ、演算子、および接尾辞」は、検索の各タイプ、演算子、および同等の接尾辞をまとめています。

表D.2 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
より小さい	<	.1
より小さいか等しい	<=	.2
等号	=	.3
より大きいか等しい	>=	.4
より大きい	>	.5
部分文字列	*	.6

D.4.3. 国際検索の例

以下のセクションでは、ディレクトリーデータで国際検索を実行する方法の例を紹介します。それぞれの例では、マッチングルールのフィルター形式がすべて示されているため、形式に慣れて最適なものを選択することができます。

D.4.3.1. less-than の例

小なり演算子 (<) 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、特定の照合順で指定の属性の前に渡され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苗字 Marquez の前に来るすべての苗字を検索するには、以下のマッチングルールフィルターのいずれかが有効です。

```
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
```

D.4.3.2. less-Than または Equal-to の例

小なりまたは等号演算子 (<=) 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、特定の照合順で指定の属性またはその前に渡され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 に、またはその前にあるすべての部屋番号を検索するには、次のマッチングルールのフィルターのいずれかが機能します。

```
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
```

D.4.3.3. 等価性の例

等号演算子 (=) や接尾辞 (.3) を使用してローカル固有の検索を行うと、特定の照合順序で与えられた属性に一致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値 softwareprodukte を持つすべての businessCategory 属性を検索するには、次のマッチングルールフィルターのいずれかが機能します。

```
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
```

D.4.3.4. より大きいか等しいの例

大なりまたは等号演算子 (>=) 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、特定の照合順で指定の属性またはその後に渡され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 に、またはその後に位置するすべての地域を検索するには、次のようなマッチングルールフィルターが有効です。

```
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
```

D.4.3.5. より大きいの例

大なり演算子 (>) 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、特定の照合順で指定の属性またはその前に渡されるすべての属性値が検索されます。

たとえば、チェコ語の照合順序でホスト schranka4 の後に来るすべてのメールホストを検索するには、次のマッチングルールフィルターのいずれかが機能します。

```
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
```

D.4.3.6. 部分文字列の例

国際的な部分文字列検索を実行すると、指定した照合順序の指定のパターンに一致する値がすべて検索されます。

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ー ID を検索するには、次のマッチングルールフィルターのいずれかを使用します。

```
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
```

modifiersName や memberOf などの DN 値属性を使用する部分文字列検索フィルターは、フィルターに空白文字が 1 つ以上含まれる場合は、エントリーには常に正しく一致しません。

この問題を回避するには、部分文字列ではなくフィルターで DN 全体を使用するか、フィルターの DN 部分文字列が RDN 境界で始まるようにしてください。つまり、DN の type= 部分で始まるようにしてください。たとえば、以下のフィルターは使用しないでください。

```
(memberOf=*Domain Administrators*)
```

ただし、以下のいずれかが正常に動作します。

```
(memberOf=cn=Domain Administrators*)
...
(memberOf=cn=Domain Administrators,ou=Groups,dc=example,dc=com)
```

D.5. マッチングルールのトラブルシューティング

国際的な照合順序のマッチングルールは、一貫した動作をしない場合があります。matching-rule 呼び出しの形式が正しく動作しないため、誤った検索結果を生成します。たとえば、以下のルールは動作しません。

```
# ldapsearch -x -p 389 -D "uid=userID,ou=people,dc=example,dc=com" -W -b
"dc=example,dc=com" "sn:2.16.840.1.113730.3.3.2.7.1:=passin"

ldapsearch -x -p 389 -D "uid=userID,ou=people,dc=example,dc=com" -W -b
"dc=example,dc=com" "sn:de:=passin"
```

ただし、以下に記載のルールは機能します (passin 値の前に .3 が必要)。

```
# ldapsearch -x -p 389 -D "uid=userID,ou=people,dc=example,dc=com" -W -b
"dc=example,dc=com" "sn:2.16.840.1.113730.3.3.2.7.1.3:=passin"

ldapsearch -x -p 389 -D "uid=userID,ou=people,dc=example,dc=com" -W -b
"dc=example,dc=com" "sn:de.3:=passin"
```

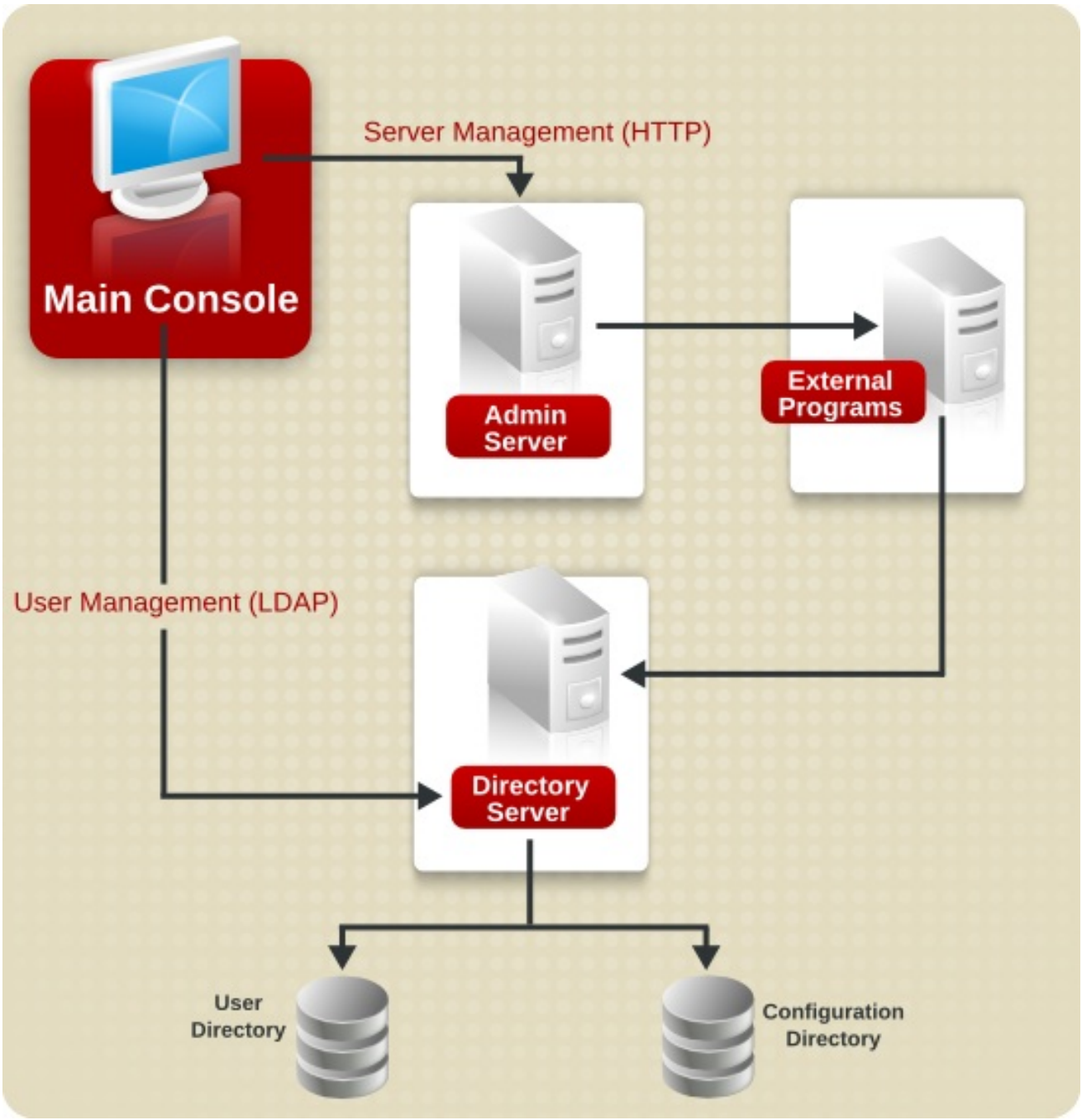
付録E 管理サーバーの管理

E.1. RED HAT 管理サーバーの概要

Red Hat Directory Server での ID 管理およびディレクトリーサービスは、以下の 3 つのコンポーネントを使用します。

- Java ベースの管理コンソール
- Web サーバーとしても機能する管理サーバー
- LDAP ディレクトリーサーバー

図E.1 コンソール、管理サーバー、および Directory Server 間の対話



管理サーバー は Directory Server インスタンスの設定要求を処理し、サーバーインスタンスの停止や起動などの数多くの一般的なサーバータスクを実行します。ディレクトリーサービスは通常、コンソールと管理サーバー 設定を保存する設定 データベースと、ユーザーおよびグループ情報が含まれる Directory Server 設定および ユーザーデータベース という 2 つのカテゴリーに分類されます。これらのデータベースは、同じ Directory Server インスタンスに保持できますが、これらのサービスを個別の Directory Server インスタンスに分割することもできます。この場合、Directory Server インスタンスの設定は、設定 Directory Server と呼ばれる別の Directory Server *に保存され、ユーザーデータは* User Directory Server *に保存されます。Administration Server は Red Hat Directory Server のサーバー設定要求を処理するため、設定 Directory Server インスタンスとユーザー Directory Server インスタンスはどちらも Administration Server 設定で定義されます。*

管理サーバーは Web サーバーとして、コンソールへの接続の処理や Admin Express などの Web アプリケーションのホストなど、Directory Server のすべてのオンライン機能を提供します。管理サー

バーは、TLS が有効な場合、HTTP または HTTPS の両方をサポートするため、クライアントはセキュアおよび標準接続の両方を介して管理サーバーに接続します。

（Red Hat Directory Server に依存する）Red Hat Directory Server または Red Hat Certificate System がインストールされると、管理サーバーが自動的にインストールされ、設定されます。1 台のマシンに複数の Directory Server インスタンスと複数の Certificate System サブシステムが存在する可能性があり、すべて Administration Server の同じインスタンスを使用します。

マシンごとに 1 つの 管理サーバーのみを使用できます。この単一の管理サーバーインスタンスは、Directory Server のインスタンスと、Red Hat Certificate System などの管理サーバーを使用できる他のクライアントの複数のインスタンスを処理できます。

コンソールが管理対象のサーバーインスタンスとは異なるマシン上にある場合でも、Directory Server または Certificate System のインスタンスを管理するためにコンソールを開いた場合は、ローカルの Administration Server インスタンスに接続して要求されたタスクを実行します。たとえば、管理サーバー はプログラムを実行して設定ディレクトリーに保存されているサーバーおよびアプリケーション設定を変更したり、サーバーがリッスンするポート番号を変更したりできます。

管理サーバー自体は、独自の Java ベースのインターフェース、設定ファイルの編集、またはコマンドラインツールを使用して管理できます。

E.2. 管理サーバー設定

Administration Server は Red Hat Directory Server または Red Hat Certificate System の別のサーバーですが、連携しています。管理サーバープロセス、ファイルの場所、および設定オプションも分離します。本章では、管理サーバーの起動および停止、TLS の有効化、ログの表示、サーバーポート番号などの管理サーバー設定プロパティーの変更など、管理サーバー情報について説明します。

E.2.1. ファイルの場所

『Red Hat Directory Server の設定、コマンド、およびファイルリファレンス の該当するセクションを参照してください』。

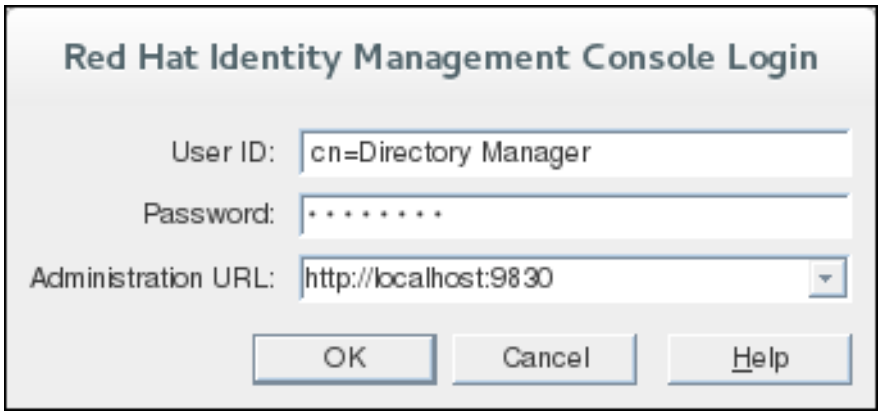
E.2.2. 管理コンソールを開く

メインコンソールを起動する単純なスクリプトがあります。Red Hat Enterprise Linux の場合は、以下を実行します。

```
# /usr/bin/redhat-idm-console
```

ログイン画面を開くと、Administration Server はユーザー名、パスワード、および管理サーバーのロケーションを要求します。管理サーバーの場所は URL です。標準の接続の場合、標準の HTTP プロトコルに対する http: 接頭辞があります。TLS が有効になっている場合は、セキュアな HTTPS プロトコルに https: 接頭辞を使用します。

図E.2 ログインボックス



注記

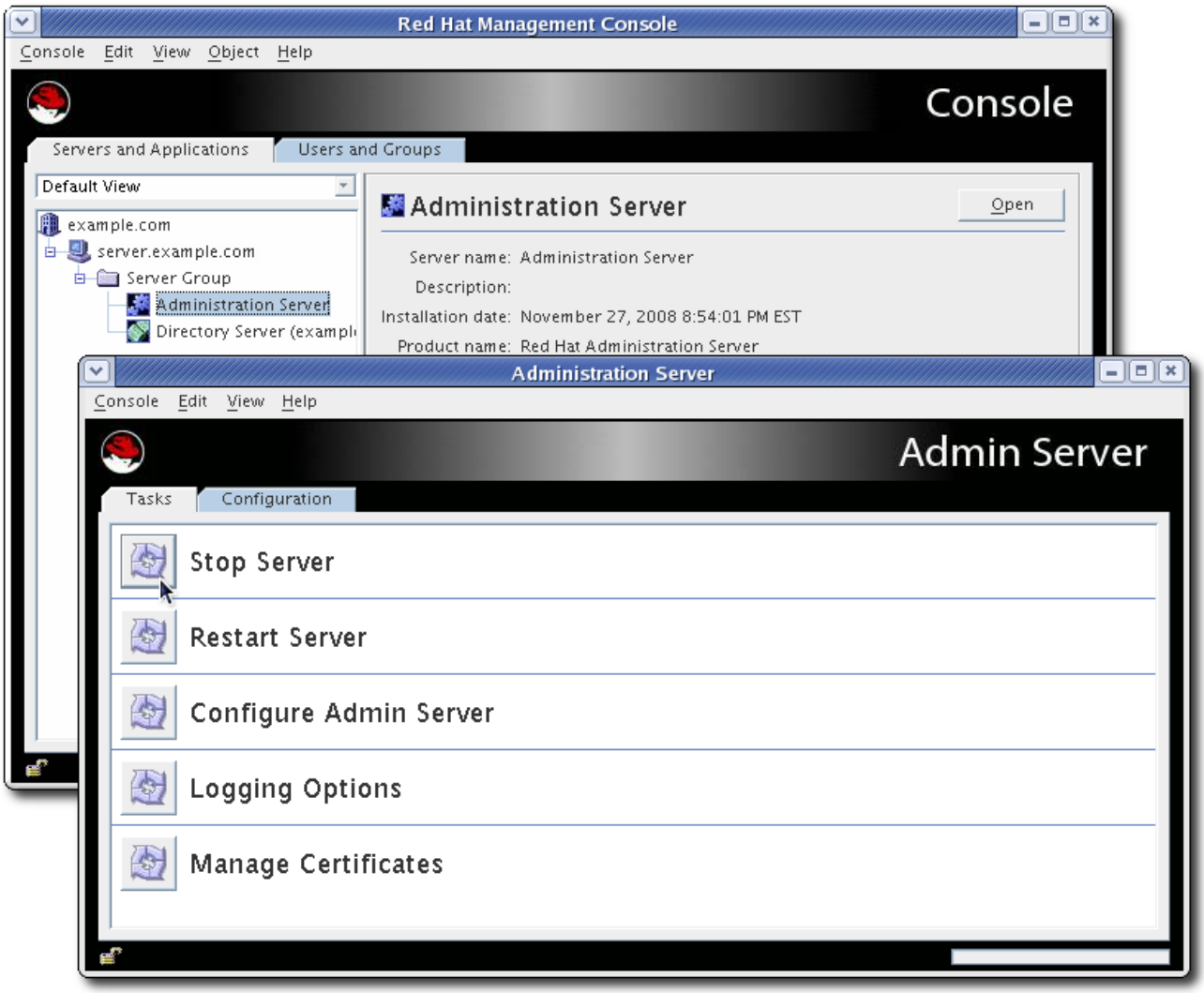
開始スクリプトで管理サーバーの URL およびポートを送信することができます。以下に例を示します。

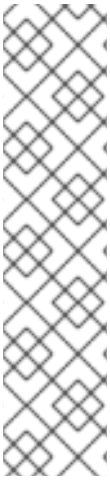
```
# /usr/bin/redhat-idm-console -a http://localhost:9830
```

a オプションは、特に Directory Server に初めてログインする場合に便利なオプションです。後続のログインでは、URL が保存されます。Administration Server のポート番号が redhat-idm-console コマンドで渡されていない場合、サーバーはコンソールのログイン画面にこれを要求します。

これにより、メインの Console ウィンドウが開きます。Administration Server Console を開くには、左側のサーバーグループから Administration Server インスタンスを選択し、ウィンドウの右上にある Open をクリックします。

図E.3 管理コンソール





注記

コンソールを起動する前に、Oracle Java Runtime Environment(JRE)または OpenJDK バージョン 1.8.0 が PATH に設定されていることを確認します。以下のコマンドを実行して、Java プログラムが PATH にあり、バージョンおよびベンダー情報を取得します。

```
java -version
```

E.2.3. ログの表示

ログファイルは管理サーバーのアクティビティーを監視し、サーバーの問題のトラブルシューティングに役立ちます。Administration Server のログは、サーバーに関する情報を提供する広範囲にサポートされている形式である Common Logfile Format を使用します。

Administration Server は、2 種類のログを生成します。

- *アクセスログアクセスログには、管理サーバーからのリクエストおよび応答が表示されます。デフォルトでは、ファイルは /var/log/dirsrv/admin-serv/access にあります。*

- *エラーログ。エラーログは、ログファイルの作成後にサーバーが発生したエラーのメッセージを表示します。また、サーバーの開始時やサーバーにログオンしようとしたユーザーなど、サーバーに関する情報メッセージも含まれます。デフォルトでは、ファイルは /var/log/dirsrv/admin-serv/error にあります。*

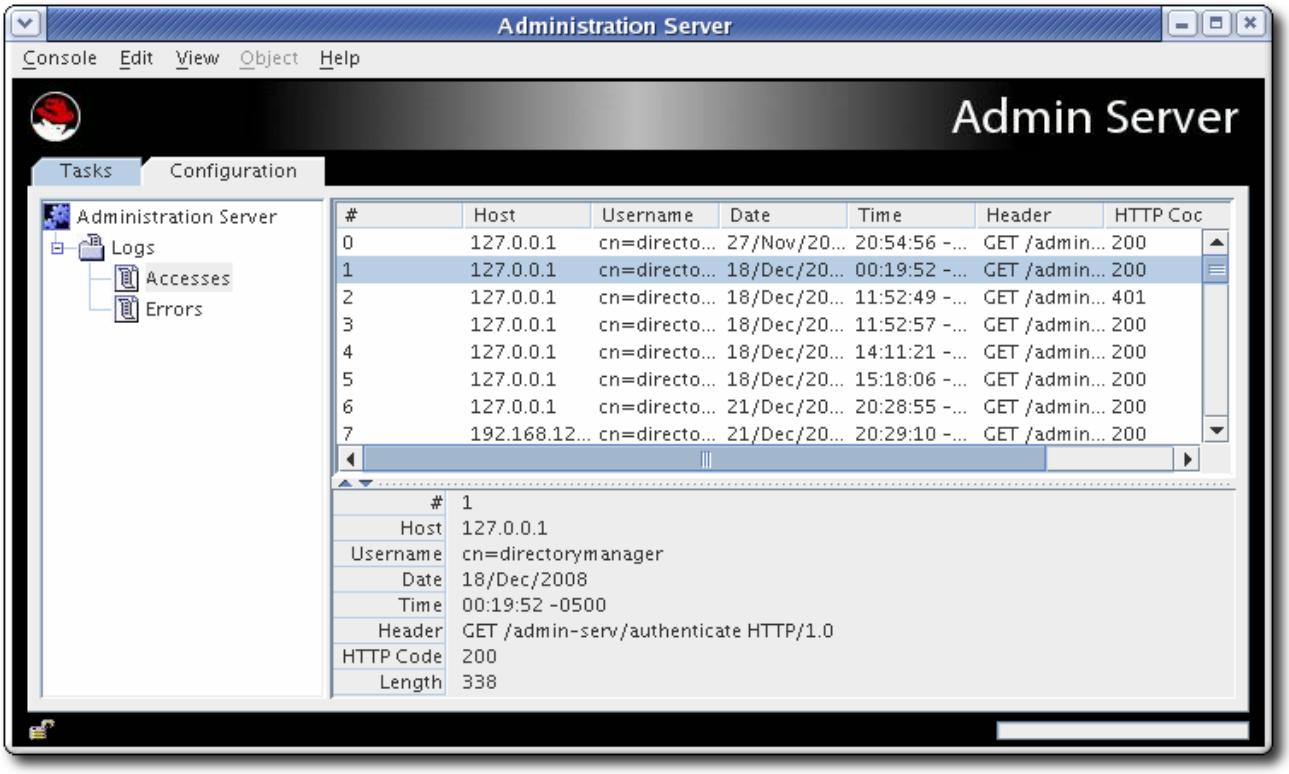
ログは、管理コンソールから表示することも、ログファイルを開いて表示できます。

E.2.3.1. コンソールからのログの表示

1. *Administration Server 管理ウィンドウを開きます。*

2. *Configuration タブをクリックします。*

3. *Logs ディレクトリーを展開し、Access es または Error のいずれかのログファイル名をクリックします。*



E.2.3.2. コマンドラインでのログの表示

デフォルトでは、アクセスログは /var/log/dirsrv/admin-serv/error です。アクセスログを表示するには、vi などのエディターで開きます。

アクセスログには、クライアントの IP アドレス、ユーザー名、およびリクエストが送信されたメソッドに基づいて管理サーバーへの接続が表示されます。各行の形式は以下のとおりです。

> *ip_address - bind_DN [timestamp -0500] "GET|POST cgi" HTTP_response bytes*

ログの例は、例E.1「アクセスログの例」 に表示されます。

> ***例E.1 アクセスログの例***
>
> ```
> 127.0.0.1 - cn=Directory Manager [23/Dec/2008:19:32:52.157345975 -0500] "GET /admin-serv/authenticate HTTP/1.0" 200 338
> 192.168.123.121 - cn=Directory Manager [23/Dec/2008:19:33:14.453724501 -0500] "POST /admin-serv/tasks/Configuration/ServerSetup HTTP/1.0" 200 244
> 192.168.123.121 - cn=Directory Manager [23/Dec/2008:19:33:16.573485244 -0500] "GET /admin-serv/tasks/Configuration/ReadLog?op=count&name=access HTTP/1.0" 200 10
> ```

デフォルトでは、エラーログは /var/log/dirsrv/admin-serv/errors です。エラーログを表示するには、vi などのエディターで開きます。

エラーログは、管理サーバーからの問題応答を記録します。アクセスログと同様に、エラーログは、クライアントの IP アドレスに基づいてエントリーもエラーメッセージのタイプとメッセージテキストを記録します。

> *[timestamp] [severity] [client ip_address error_message*

重大度 メッセージは、管理者の介入に十分なエラーが重要かどうかを示します。[warning]、[error]、および [critical] には、すぐに管理者のアクションが必要です。その他の重大度は、エラーが情報であるか、またはデバッグに関することを意味します。

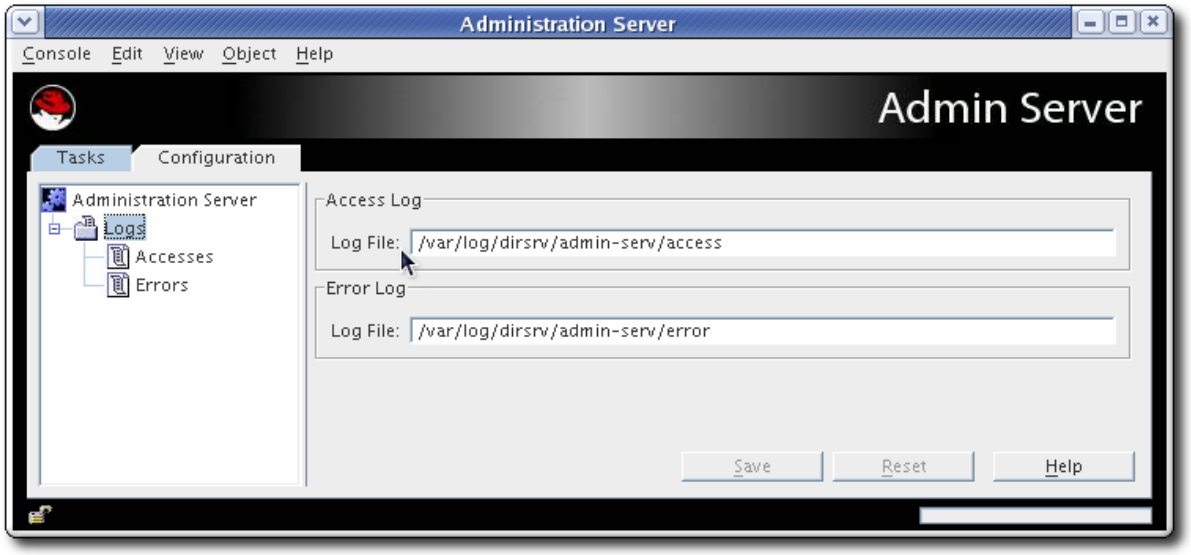
ログの例は、例E.2「エラーログの例」 に表示されます。

> **例E.2 エラーログの例**
>
> ```
> [24/Mar/2017:11:14:27.110314677 +0100] - NOTICE - ldbm_back_start - total cache size:
> 417775616 B;
> [24/Mar/2017:11:14:27.165466639 +0100] - INFO - dblayer_start - Resizing db cache size:
> 1519206400 -> 132562944
> [24/Mar/2017:11:14:27.650899322 +0100] - INFO - slapd_daemon - slapd started.  Listening
> on All Interfaces port 389 for LDAP requests
> [24/Mar/2017:11:14:29.620268885 +0100] - WARN - modify_config_dse - Modification of
> attribute "aci" is not allowed, ignoring!
> ```

E.2.3.3. コンソールでのログ名の変更

アクセスおよびエラーログファイルの名前は、ファイルをローテーションするために変更できます。既存のログファイルが大きすぎる場合は、新しいファイルを作成するには、このローテーションを手動で行う必要があります。

1. Administration Server 管理ウィンドウを開きます。
2. Configuration タブをクリックします。
3. 左側のパネルで Logs をクリックします。
4. 右側の Logs ウィンドウで、新規ログファイル名を入力します。



警告

ログファイルへのパスは絶対的なものであり、変更することはできません。

5. *OK をクリックして変更を保存します。*

6. *Tasks タブを開き、Restart Server ボタンをクリックしてサーバーを再起動して変更を適用します。*

E.2.3.4. コマンドラインでのログ場所の変更

ファイルをローテーションするために、アクセスおよびエラーログファイルの名前および場所を変更できます。既存のログファイルが大きすぎる場合は、新しいファイルを作成するには、このローテーションを手動で行う必要があります。/var/log/dirsrv/admin-serv のデフォルトの場所がアプリケーションのニーズを満たさない場合は、場所を変更できます。

管理サーバー設定は 2 つの場所に保存されます。メインエントリーは、Configuration Directory Server の o=NetscapeRoot データベースの LDAP エントリーです。もう 1 つは console.conf ファイルです。ログ設定の変更には、両方の設定を変更する必要があります。

1. *Configuration Directory Server で Administration Server 設定エントリーを編集します。*

a.

Administration Server エントリーの名前を取得します。Administration Server エントリーには特別なオブジェクトクラス nsAdminConfig があるため、そのオブジェクトクラスを使用して DN を取得することができます。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -b
"o=NetscapeRoot" "(objectclass=nsAdminConfig)" dn

version:1
dn: cn=configuration,cn=admin-serv-example,cn=Red Hat Administration
Server,cn=Server
Group,cn=server.example.com,ou=example.com,o=NetscapeRoot
```

b.

Administration Server エントリーは、ldapmodify を使用して編集できます。アクセスおよびエラーログ設定は、それぞれ nsAccessLogs 属性および nsErrorLogs 属性に保存されます。以下に例を示します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=configuration,cn=admin-serv-example,cn=Red Hat Administration
Server,cn=Server
Group,cn=server.example.com,ou=example.com,o=NetscapeRoot
changetype:modify
replace:nsAccessLog
nsAccessLog:/var/log/dirsrv/admin-serv/access_new
```

Enter を 2 回押して 操作を送信し、Control +C を押して ldapmodify を閉じます。

2.

Administration Server 設定ディレクトリーを開きます。

```
# cd /etc/dirsrv/admin-serv
```

3.

console.conf ファイルを編集します。アクセスログの場合は、CustomLog パラメーターのパスおよびファイル名を編集します。エラーログの場合は、ErrorLog パラメーターのパスおよびファイル名を編集します。

```
CustomLog /var/log/dirsrv/admin-serv/access_new common
ErrorLog /var/log/dirsrv/admin-serv/error_new
```

アクセスログパスの後に 一般的な 用語を残します。これは、アクセスログが Common Log Format にあることを意味します。

4.
管理サーバーを再起動します。

```
# systemctl restart dirsrv-admin.service
```

E.2.3.5. IP アドレスの代わりにホスト名を表示するログの設定

デフォルトでは、ログには管理サーバーに接続するクライアントの IP アドレスが表示されます。これは、すべての接続に DNS ルックアップを行う必要がないため、管理サーバーでは高速になります。管理サーバーは、ホスト名をログで使用するように DNS ルックアップを実行するように設定できます。IP アドレスの代わりにホスト名を使用すると、ホスト名が解決できないという不要なエラーメッセージも削除されています。

管理サーバーが DNS ルックアップを実行するように設定するには、以下を実行します。

1.
管理コンソールの console.conf ファイルを編集します。

```
# cd /etc/dirsrv/admin-serv
# vim console.conf
```

2.
HostnameLookups パラメーターを on に設定します。デフォルトでは、これはオフになっており、IP アドレスはホスト名ではなくログに記録されます。

```
HostnameLookups on
```

E.2.4. ポート番号の変更

ポート番号は、管理サーバーのインスタンスがメッセージをリッスンするかどうかを指定します。

インスタンスを最初にインストールし、setup-ds-admin.pl などの設定スクリプトの実行時に、管理サーバーのデフォルトのポート番号が設定されます。デフォルトのポート番号は 9830 です 。

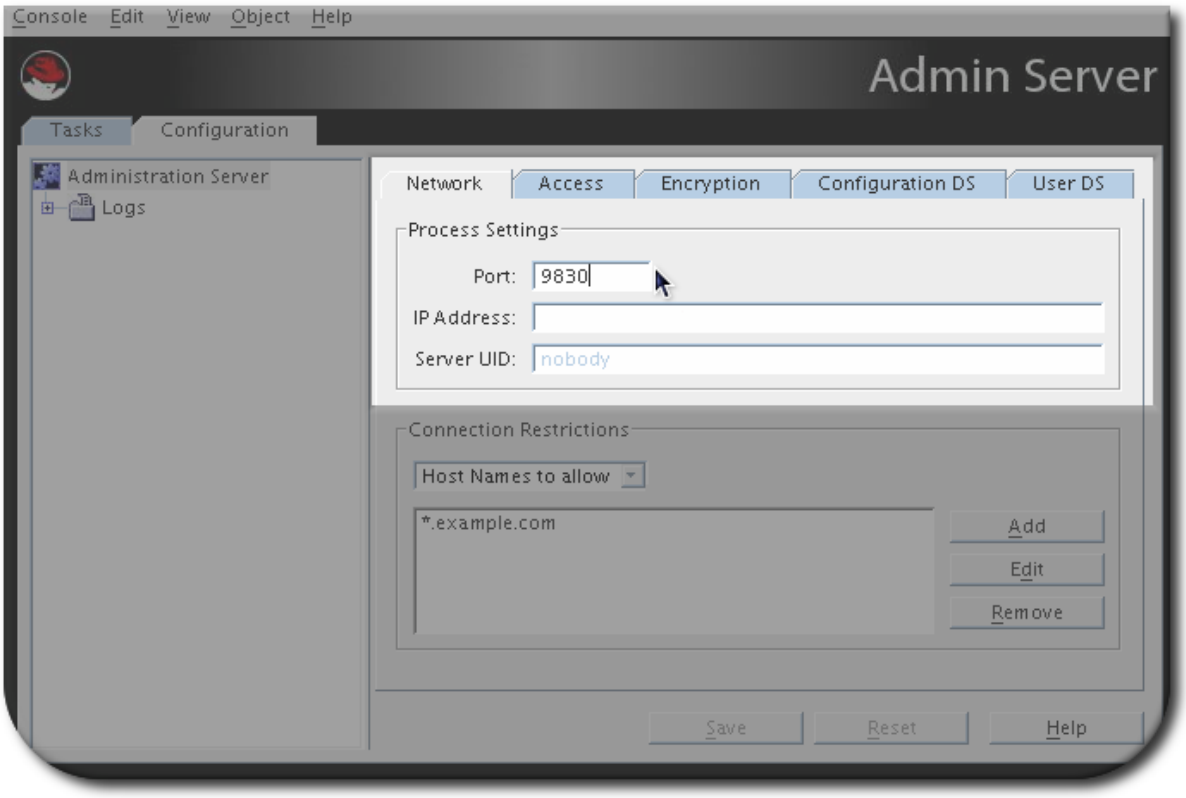
E.2.4.1. コンソールのポート番号の変更

1.
Administration Server 管理ウィンドウを開きます。

2.
Configuration タブをクリックします。

3.

Network タブをクリックします。



4.

Port フィールドに Administration Server インスタンスのポート番号を入力します。管理サーバーのポート番号には、デフォルトの 9830 があります。

5.

OK をクリックします。

6.

Tasks タブを開き、Restart Server ボタンをクリックしてサーバーを再起動して変更を適用します。

7.

コンソールを閉じ、接続 URL に新しい管理サーバーポート番号を指定して、コンソールを再起動します。

E.2.4.2. コマンドラインでのポート番号の変更

Administration サーバーのポート番号はデフォルトで 9830 です。

管理サーバー設定は 2 つの場所に保存されます。メインエントリーは、Configuration Directory Server の o=NetscapeRoot データベースの LDAP エントリーです。もう 1 つは console.conf ファイ

ルです。ポート番号を変更するには、両方の設定を変更する必要があります。

1. Configuration Directory Server で Administration Server 設定エントリーを編集します。

   a. Administration Server エントリーの名前を取得します。Administration Server エントリーには特別なオブジェクトクラス nsAdminConfig があるため、そのオブジェクトクラスを使用して DN を取得することができます。

   ```
   # ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -b
   "o=NetscapeRoot" "(objectclass=nsAdminConfig)" dn

   version:1
   dn: cn=configuration,cn=admin-serv-example,cn=Red Hat Administration
   Server,cn=Server
   Group,cn=server.example.com,ou=example.com,o=NetscapeRoot
   ```

   b. Administration Server エントリーは、ldapmodify を使用して編集できます。ポート番号は nsServerPort 属性に設定されます。以下に例を示します。

   ```
   # ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

   dn: cn=configuration,cn=admin-serv-example,cn=Red Hat Administration
   Server,cn=Server
   Group,cn=server.example.com,ou=example.com,o=NetscapeRoot
   changetype:modify
   replace:nsServerPort
   nsServerPort:10030
   ```

   Enter を 2 回押して 操作を送信し、Control +C を押して ldapmodify を閉じます。

2. Administration Server 設定ディレクトリーを開きます。

   ```
   # cd /etc/dirsrv/admin-serv
   ```

3. console.conf ファイルの Listen パラメーターを編集します。

   ```
   Listen 0.0.0.0:10030
   ```

4.
管理サーバーを再起動します。

```
# systemctl restart dirsrv-admin.service
```

E.2.5. ホスト制限の設定

接続制限は、管理サーバーに接続することのできるホストを指定します。これらのホストは、DNS 名、IP アドレス、またはその両方で一覧表示できます。接続制限パラメーターにリストされているホストマシンのみが管理サーバーへの接続が許可されます。この設定により、ドメインまたは IP アドレス範囲内のワイルドカードにより、接続制限の設定が簡単になります。

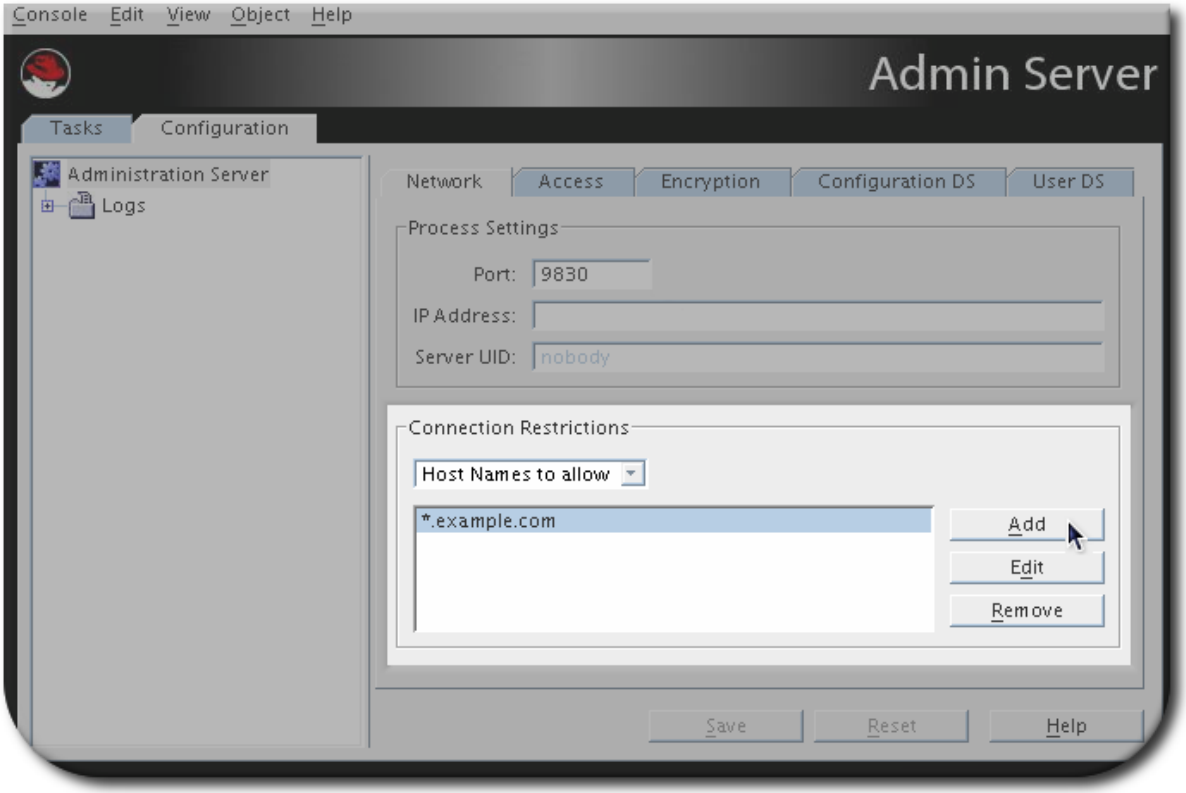
E.2.5.1. コンソールでのホスト制限の設定

1.
Administration Server 管理ウィンドウを開きます。

2.
Configuration タブをクリックします。

3.
Network タブをクリックします。

4.
接続制限 エリアには、管理サーバーへの接続が許可されているホストの一覧が表示されます。ドロップダウンリストは、リストエントリーが DNS 名または IP アドレスで追加されるかどうかを指定します。この一覧は最初にホスト名で評価され、次に IP アドレスで評価されます。

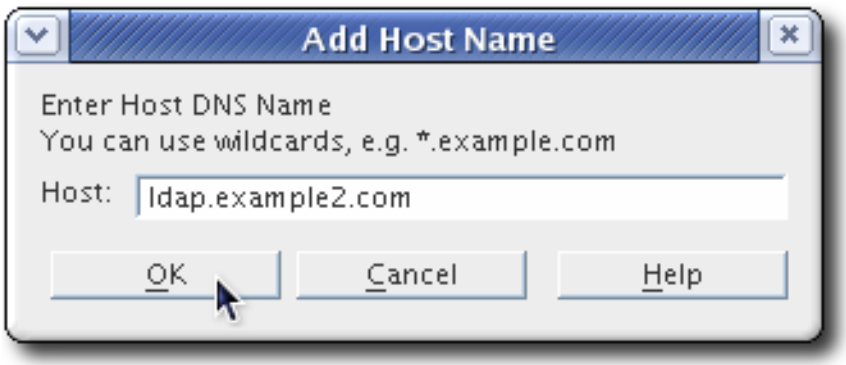


5.

追加 ボタンをクリックして、許可されたコンピューターの一覧に別のホストを追加します。ホスト名を追加するには、上部のドロップダウンリストで、許可するホスト名 を読み取るようにしてください。IP アドレスを追加するには、許可する IP アドレス を選択します。

6.

ホスト名または IPv4 アドレスまたは IPv6 アドレスのいずれかでホスト情報を入力します。



** ワイルドカードを使用すると、ホストのグループを指定できます。たとえば、*.example.com は、example .com ドメイン内のすべてのマシンがインスタンスにアクセスできるようにします。205.12.* を入力すると、IP アドレスが 205.12 で始まるすべてのホストがインスタンスにアクセスできるようになります。*

IP アドレスの制限を指定する場合には、3 つすべて分離ドットを含めます。これがない場合、Administration Server はエラーメッセージを返します。

7.
OK をクリックして Add... を閉じてから、保存 ボタンをクリックして新規ホストを保存します。

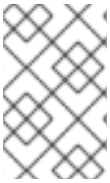
8.
Tasks タブを開き、Restart Server ボタンをクリックしてサーバーを再起動して変更を適用します。

一覧表示されるホストまたは IP アドレスの情報を変更するには、編集 ボタンをクリックして、指定の情報を変更します。許可されるホストまたは IP アドレスを削除するには、一覧からホストを選択し、Remove をクリックします。管理サーバー。

E.2.5.2. コマンドラインでのホスト制限の設定

ホストの制限により、ネットワーククライアントが管理サーバーに接続可能なルールを設定します。したがって、管理サーバーを使用するサービスにルールを設定します。ホストの制限には、ホストの制限があり、ホスト名またはドメイン名および制限に基づいて制限が IP アドレスに基づきます。

Administration Server ホストの制限は、Configuration Directory Server の o=NetscapeRoot データベースのメイン設定エントリーに設定されます。IP アドレスとホスト名には、ホストの制限を設定する属性は 2 つあります。それぞれ IP アドレスとホスト名には nsAdminAccessAddresses と nsAdminAccessHosts の 2 つの属性があります。



注記

管理サーバーは、IPv4 アドレスと IPv6 アドレスの両方をサポートします。

Administration Server エントリーは、ldapmodify を使用して編集できます。

ホストの制限を設定するには、以下を実行します。

1.
Administration Server エントリーの名前を取得します。Administration Server エントリーには特別なオブジェクトクラス nsAdminConfig があるため、そのオブジェクトクラスを使用して DN を取得することができます。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -b
"o=NetscapeRoot" "(objectclass=nsAdminConfig)" dn
```

```
version:1
dn: cn=configuration,cn=admin-serv-example,cn=Red Hat Administration
Server,cn=Server Group,cn=server.example.com,ou=example.com,o=NetscapeRoot
```

2. IP アドレスベースの制限を設定するには、nsAdminAccessAddresses 属性を編集します。IPv4 アドレスまたは IPv6 アドレスのいずれかを使用できます。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=configuration,cn=admin-serv-example,cn=Red Hat Administration
Server,cn=Server Group,cn=server.example.com,ou=example.com,o=NetscapeRoot
changetype:modify
replace:nsAdminAccessAddresses
nsAdminAccessAddresses:72.5.*.*
```

Enter を 2 回押して 操作を送信し、Control +C を押して ldapmodify を閉じます。

nsAdminAccessAddresses 値では、ワイルドカードを使用して範囲を許可することができます。IPv4 アドレスまたは IPv6 アドレスのいずれかを使用できます。

たとえば、すべての IP アドレスを許可するには、以下を実行します。

```
nsAdminAccessAddresses:*
```

ローカルネットワーク上のアドレスのサブセットのみを許可するには、以下を実行します。

```
nsAdminAccessAddresses:192.168.123.*
```

3. ホスト名またはドメインベースの制限を設定するには、nsAdminAccessHosts 属性を編集します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=configuration,cn=admin-serv-example,cn=Red Hat Administration
Server,cn=Server Group,cn=server.example.com,ou=example.com,o=NetscapeRoot
changetype:modify
replace:nsAdminAccessHosts
nsAdminAccessHosts:*.example.com
```

Enter を 2 回押して 操作を送信し、Control +C を押して ldapmodify を閉じます。

4.
Administration Server を再起動して変更を適用します。

```
# systemctl restart dirsrv-admin.service
```

E.2.6. 管理ユーザーのパスワードの変更

インストール時に、設定管理者 のユーザー名とパスワードの入力を求められます。このユーザーは、設定 ディレクトリー全体にアクセスして変更権限のあるユーザーです。Configuration Administrator エントリーは、以下の DN 配下のディレクトリーに保存されます。

```
uid=userID,ou=Administrators,ou=TopologyManagement,o=NetscapeRoot
```

設定管理者のユーザー名とパスワードは Directory Server で管理され、LDAP エントリーで表示されます。これは、『『Red Hat Directory Server 管理ガイド』』 で説明されています。

インストール時に、設定管理者のユーザー名とパスワードを使用して管理サーバー管理者 を自動的に作成します。このユーザーは、ローカルサーバーグループでのサーバーの起動、停止、再起動など、限られた数の管理タスクを実行できます。Administration Server Administrator は、Directory Server が実行されていない場合にコンソールにログインする目的で作成されます。

Administration Server Administrator には LDAP エントリーがありません。これは、ローカル設定ファイル /etc/dirsrv/admin-serv/admpw のエンティティーとしてのみ存在します。

これらはインストール時に同時に作成されますが、その時点で Configuration Administrator と Administration Server Administrator は 2 つの異なるエンティティーです。コンソールでユーザー名またはパスワードを変更すると、コンソールによって自動的に同じ変更は加えられません。

管理サーバー管理者は、管理サーバーのすべての設定に完全アクセスできます。admin ユーザーの情報は、コンソールの Access タブに設定されます。

注記

Administration Server 管理者のユーザー名およびパスワードは /etc/dirsrv/admin-serv/admpw ファイルに保存されます。以下に例を示します。

```
admin:{SHA}W6ph5Mm5Pz8GgiULbPgzG37mj9g=
```

パスワードは暗号化され、admpw ファイルで直接変更することはできません。ユーザー名はこのファイルで変更できますが、最初にコンソールでパスワードが更新されない限り、コンソールへのログインには使用できません。このため、管理コンソールからのみ管理サーバー管理者のユーザー名およびパスワードを編集することをお勧めします。

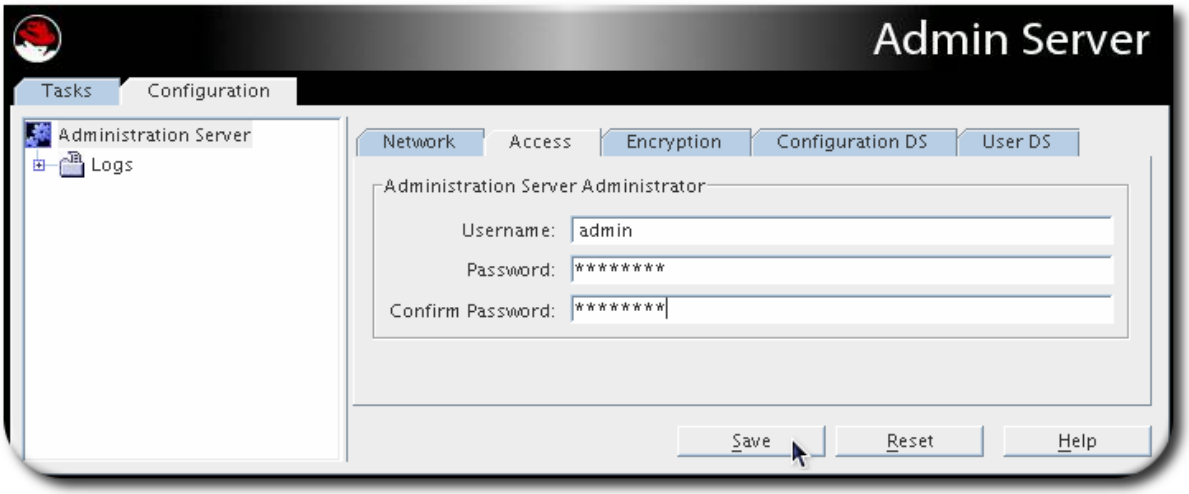
管理サーバーの管理者の ID またはパスワードを変更するには、以下を実行します。

1. *Administration Server 管理ウィンドウを開きます。*

2. *Configuration タブをクリックします。*

3. *Access タブをクリックします。*

4. *admin ユーザーまたはパスワードを変更します。ユーザー名は、管理サーバーにログインするために提供された ID です。*



5. *Save をクリックします。*

E.2.7. TLS の使用

管理サーバーは、サーバーで TLS が有効になっている場合に HTTPS（セキュアな HTTP）上で実行できます。TLS を有効にする手順があります。

1. 証明書要求の生成および送信。
2. 証明書の受信およびインストール。
3. 証明書を発行した認証局(CA)を信頼すること。
4. 管理サーバー設定を変更して TLS 接続を許可します。

E.2.7.1. 管理サーバーの証明書の管理

管理コンソールの証明書を要求およびインストールするには、Directory Server コンソールの手順に従います。以下を参照してください。

- 「Directory Server インスタンスの NSS データベースの作成」
- 「証明書署名要求の作成」

  管理サーバーと同じ Directory Server の証明書を使用するには、「管理サーバーの Directory Server プライベートキーおよび証明書の使用」 を参照してください。

- 「CA 証明書のインストール」

  管理サーバーと同じ Directory Server の証明書を使用するには、「管理サーバーの Directory Server プライベートキーおよび証明書の使用」 を参照してください。

- 「証明書のインストール」

- 「証明書の更新」
- 「CA 信頼オプションの変更」
- 「NSS データベースのパスワードの変更」
- 「証明書失効リストの追加」

重要

使用する場合は、以下を行います。

- グラフィカルユーザーインターフェース。Directory Server コンソールではなく、管理コンソールの 証明書の管理 メニューで手順を実行します。
- Network Security Services(NSS)データベースを管理する場合は、/etc/dirsrv/slapd-instance_name/ ディレクトリーの代わりに /etc /dirsrv/admin-serv / コマンドを使用します。

E.2.7.1.1. 管理サーバーの Directory Server プライベートキーおよび証明書の使用

管理サーバーおよび Directory Server は、異なる PKI データベースを使用します。Directory Server の Certificate Request Wizard が渡されると、自動生成された秘密鍵は Directory Server の PKI データベースに保存されます。ただし、同じ秘密鍵が両方のデータベースに存在しないため、発行した証明書は他のデータベースにインストールできません。

以下のコマンドを実行して Directory Server の秘密鍵および証明書をエクスポートし、それらを Administration Server のデータベースにインポートします。

1. 管理サーバーをシャットダウンします。

```
# systemctl stop dirsrv-admin
```

2. Directory Server をシャットダウンします。

```
# systemctl stop dirsrv@instance
```

3.

Directory Server NSS データベースの内容を一覧表示します。

```
# certutil -L -d /etc/dirsrv/admin-serv/

Certificate Nickname                    Trust Attributes
                                        SSL,S/MIME,JAR/XPI

Demo CA                                 CT,,
server-cert                             u,u,u
```

4.

Directory Server の PKI データベースから、server-cert という名前の秘密鍵と証明書をエクスポートします。

```
# pk12util -o /tmp/keys.pk12 -n server-cert -d /etc/dirsrv/slapd-instance/
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
```

Directory Server のキーストアパスワードを入力します。オプションで、プロンプトが表示されたら、一時的にエクスポートされたファイルの新しいパスワードを入力します。

5.

秘密鍵および証明書を管理サーバーの PKI データベースにインポートします。

```
# pk12util -i /tmp/keys.pk12 -d /etc/dirsrv/admin-serv/
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
Enter password for PKCS12 file:
pk12util: PKCS12 IMPORT SUCCESSFUL
```

pk12util は、管理サーバーのキーストアのパスワードを設定するよう要求します。事前にこのデータベースに 1 つ設定されていた場合は、代わりにこのパスワードを入力するよう求められます。前の手順でエクスポートしたファイルにパスワードを設定すると、このパスワードを入力するよう求められます。

6.

一時的にエクスポートされたファイルを削除します。

```
# rm /tmp/keys.pk12
```

7. Demo CA を信頼します。

```
# certutil -M -d /etc/dirsrv/admin-serv/ -n "Demo CA" -t CT,,
```

8. Directory Server を起動します。

```
# systemctl start dirsrv@instance
```

9. 管理サーバーを起動します。

```
# systemctl start dirsrv-admin
```

E.2.7.2. TLS の有効化

「管理サーバーでの TLS の有効化」 を参照してください。

E.2.7.3. 管理サーバーのパスワードファイルの作成

通常、TLS が有効になっている場合、管理サーバーが再起動されると、サーバーはセキュリティーパスワードを要求します。

```
Starting dirsrv-admin:
Please enter password for "internal" token:
```

管理サーバーは、TLS が有効な場合にパスワードファイルを使用できます。これにより、セキュリティーパスワードを要求せずにサーバーが警告なしで再起動されます。

警告

このパスワードはパスワードファイル内にクリアテキストに保存されるため、その使用は重大なセキュリティーリスクを表します。サーバーがセキュアでない環境で実行している場合は、パスワードファイルを使用しないでください。

1.

以下の内容で /etc/dirsrv/admin-serv/password.conf ファイルを作成します。

- FIPS(Federal Information Processing Standard)モードが無効になっているシステムの場合：

```
internal:password
```

- FIPS モードが有効になっているシステムの場合は、次のコマンドを実行します。

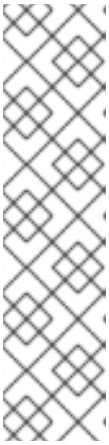
```
internal:password
NSS FIPS 140-2 Certificate DB:password
```

このファイルの行は、token_name:password の形式を使用します。

NSS ソフトウェア暗号モジュール（デフォルトのソフトウェアデータベース）では、トークンは常に internal と呼ばれます。FIPS モードを有効にすると、証明書データベースの追加トークンは NSS FIPS 140-2 Certificate DB と呼ばれます。

2.

他のユーザー(mode 0400)にはアクセスなく、管理サーバーユーザーがパスワードファイルを所有し、admin Server ユーザーで読み取り専用に設定する必要があります。



注記

Administration Server ユーザー ID を確認するには、Administration Server 設定ディレクトリーで grep を実行します。

```
# grep "^User" /etc/dirsrv/admin-serv/console.conf
User dirsrv
```

パーミッションを設定するには、以下を入力します。

```
# chown dirsrv:root /etc/dirsrv/admin-serv/password.conf
# chmod 0400 /etc/dirsrv/admin-serv/password.conf
```

3.

/etc/dirsrv/admin-serv/nss.conf ファイルを編集し、新しいパスワードファイルの場所を参照します。

```
#   Pass Phrase Dialog:
#   Configure the pass phrase gathering process.
#   The filtering dialog program (`builtin' is a internal
#   terminal dialog) has to provide the pass phrase on stdout.
NSSPassPhraseDialog  file://etc/dirsrv/admin-serv/password.conf
```

4.

管理サーバーを再起動します。

```
# systemctl restart dirsrv-admin.service
```

TLS を有効にした後、管理サーバーは HTTPS を使用してのみ接続できます。Administration Server およびそのサービスへ接続するための以前の HTTP（標準）URL はすべて機能しなくなります。これは、コンソールを使用して管理サーバーに接続するか、または Web ブラウザーを使用する場合でも当てはまります。

E.2.8. Directory Server 設定の変更

管理サーバー は、Directory Server 設定ディレクトリー（インスタンス設定 情報を保存する）および Directory Server User Directory （実際のディレクトリーエントリーを保存する）に関する情報を保存します。これは同じディレクトリーインスタンスになる可能性もありますが、必須ではありません。これらの両方のデータベースの設定は、管理サーバー設定で編集して、別の Directory Server インスタンスと通信できます。

E.2.8.1. 設定ディレクトリーホストまたはポートの変更

設定データは、設定 ディレクトリーの o=NetscapeRoot に保存されます。設定データベースには、ネットワークトポロジー情報やサーバーインスタンスエントリーなどのサーバー設定が含まれます。サーバー設定の変更が設定ディレクトリーサブツリーに保存されている場合。

警告

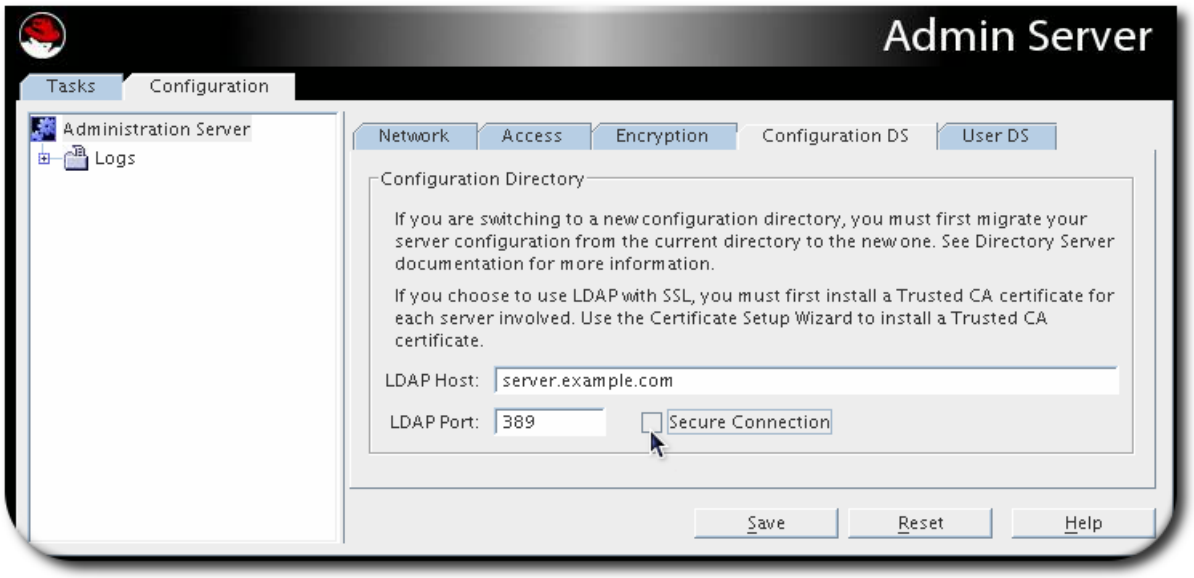
Directory Server ホスト名またはポート番号を変更すると、サーバーグループの残りのサーバーに影響します。ここで設定を変更すると、サーバーグループのすべてのサーバーに同じ変更を加える必要があります。

1. Administration Server 管理ウィンドウを開きます。

2. Configuration タブをクリックします。

3. Configuration DS タブをクリックします。

4. Configuration Directory Server 接続情報を設定します。



- LDAP ホストは、設定 Directory Server マシンのホスト名、IPv4、または IPv6 アド

レスです。

- LDAP Port は Directory Server インスタンスに使用するポート番号です。通常の LDAP ポートは 389 で、デフォルトの LDAPS（セキュア）ポート番号は 636 です。

- Secure Connection チェックボックスにチェックを入れて、セキュアなポートを使用します。このボックスをチェックする前に、設定 Directory Server が TLS を有効にしていることを確認します。

5. Save をクリックします。

E.2.8.2. ユーザーディレクトリーホストまたはポートの変更

ユーザーディレクトリーは、認証、ユーザー管理、およびアクセス制御に使用されます。すべてのユーザーおよびグループデータ、アカウントデータ、グループリスト、およびアクセス制御命令(ACI)を保存します。

複数のユーザーディレクトリーを使用すると、地理的に分散している組織の全体的なパフォーマンスが向上します。または、個別のディレクトリーでは、個別のディレクトリーの利点が広がります。

管理サーバーは、複数のユーザーディレクトリーに対してユーザーを認証するように設定できます。

ユーザーディレクトリーの情報を変更するには、以下を実行します。

1. Administration Server 管理ウィンドウを開きます。

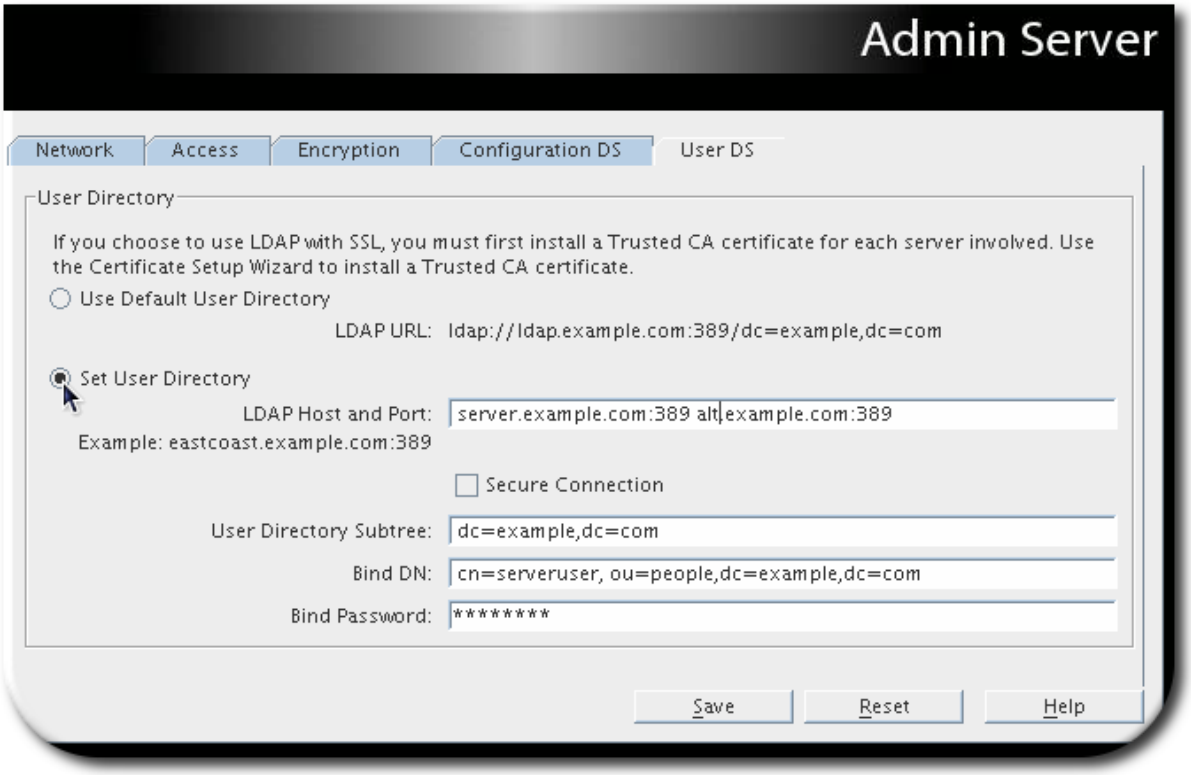
2. Configuration タブをクリックします。

3. User DS タブをクリックします。

4. User Directory Server の接続情報を設定します。

5.

ユーザーディレクトリー情報を編集します。



***Use Default User Directory** ラジオボタンは、ドメインに関連付けられたデフォルトのユーザーディレクトリーを使用します。複数の **Directory Server** インスタンスを使用するか、別のインスタンスを使用するには、**Set User Directory** ラジオボタンを選択して、必要な情報を設定します。*

- *LDAP ホストおよびポート フィールドは、IPv4 アドレスまたは IPv6 アドレスで* **hostname:port** *または* **ip_address:port** *形式を使用して、ユーザーディレクトリーインスタンスの場所を指定します。*

  認証およびその他のディレクトリー機能用に、ユーザーディレクトリーに複数の場所を設定できます。各場所はスペースで区切ります。以下に例を示します。

  ```
  server.example.com:389 alt.example.com:389
  ```

  注記

  LDAP ホストおよびポート フィールドに複数の場所 を指定すると、残りのフィールドの設定はそれらのインスタンスすべてに適用されます。

-

TLS を使用してユーザーディレクトリーに接続するには、Secure Connection ボックスにチェックを入れます。Directory Server がすでに TLS を使用するように設定されている場合のみこれ を選択します。

- User Directory サブツリー を指定します。以下に例を示します。

```
dc=example,dc=com
```

LDAP Host および Port フィールドに一覧表示されるすべての場所には、そのサブツリーが含まれる必要があり、サブツリーにはユーザー情報が含まれている必要があります。

- 必要に応じて、バインド DN と、ユーザーディレクトリーを接続するユーザーの Password を入力します。

6. Save をクリックします。

付録F ADMIN EXPRESS の使用

F.1. ADMIN EXPRESS でのサーバーの管理

Admin Express は、サーバーの基本管理を行うための、簡単な Web ベースのゲートウェイを提供します。Admin Express では 3 つのタスクを実行できます。

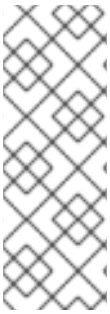
- サーバーの停止および起動
- サーバーのアクセス、エラー、監査ログの確認
- Directory Server との間のレプリケーションの進捗および情報の監視

F.1.1. Admin Express を開く

管理サーバーのサービスページの URL は、管理サーバーホスト（ホスト名、IPv4 アドレス、または IPv6 アドレス）およびポートです。以下に例を示します。

```
http://ldap.example.com:9830/
```

Admin Express ページは常にその URL で利用できます。



注記

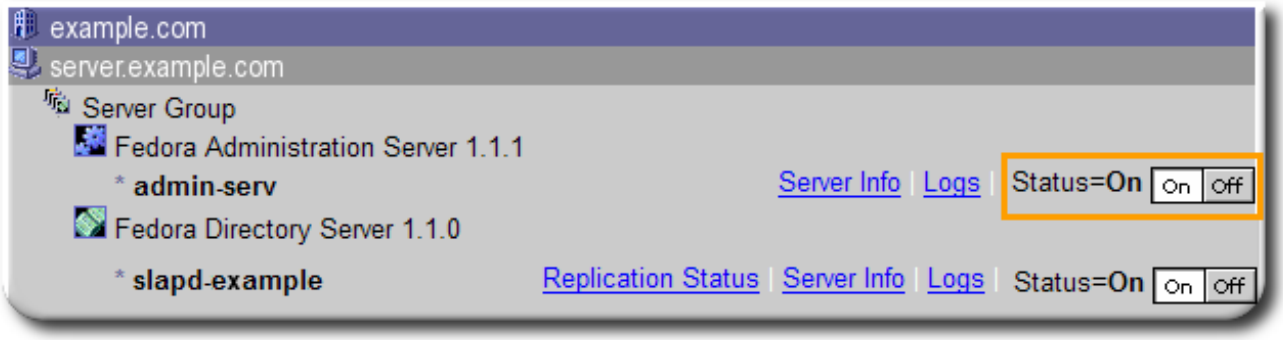
管理サーバーで TLS が有効になっている場合、URL はプレフィックス https と同じポート番号を使用する必要があります。標準の HTTP URL は機能しません。

```
https://ldap.example.com:9830/
```

F.1.2. サーバーの起動と停止

メインの Admin Express ページには、サーバーをオフにしてオンするボタンがあります。

図F.1 サーバーの停止および停止



重要

Administration Server または Configuration Directory Server のいずれかが Admin Express ページからオフになっている場合は、Admin Express On/Off ボタンではなく、コマンドラインで再起動する必要があります。これは、Admin Express が Administration Server と Configuration Directory Server の両方にアクセスする必要があるためです。

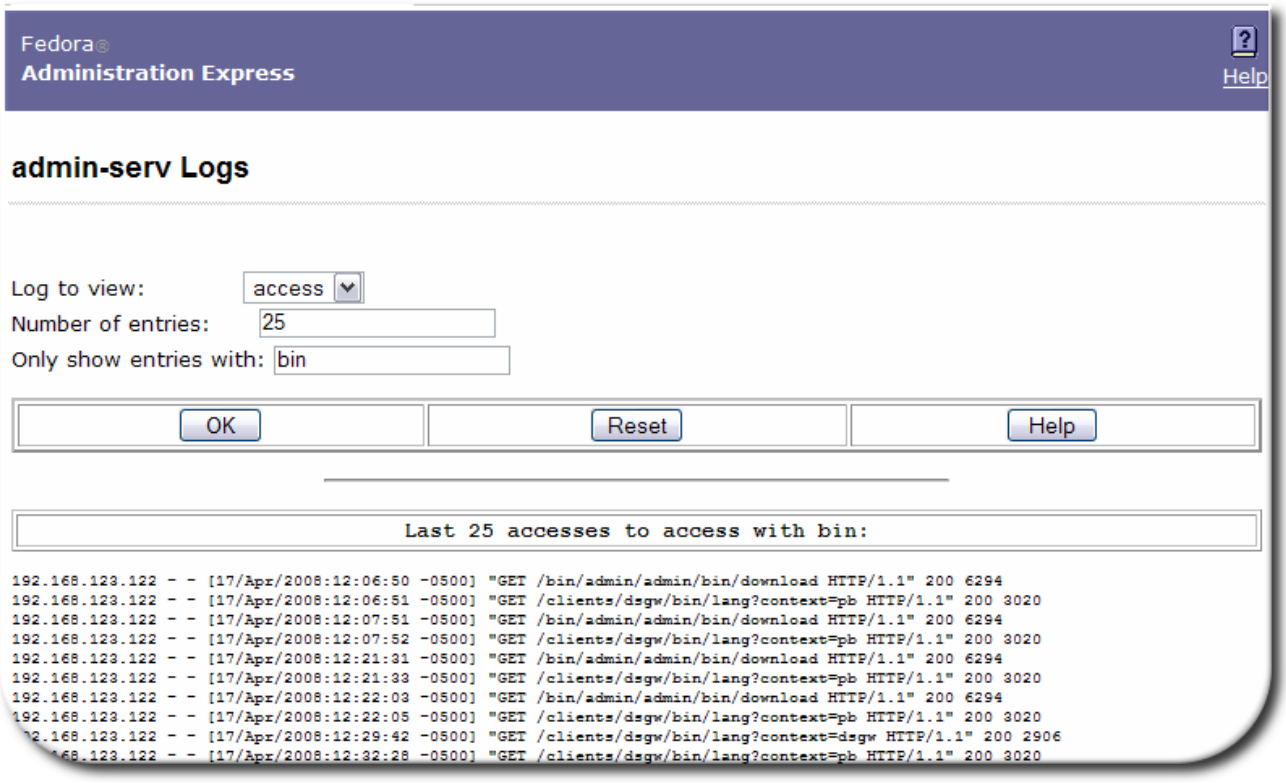
他の Directory Server インスタンスは、Admin Express を介して安全に停止および再起動できます。

F.1.3. サーバーログの表示

Admin Express は、Directory Server および Administration Server のアクセスおよびエラーログと、Directory Server の監査ログを表示および検索できます。

1. *Admin Express ページで、サーバー名の Logs リンクをクリックします。*
2. *表示するログタイプ、返す行数、検索する文字列数を選択し、OK をクリックします。*

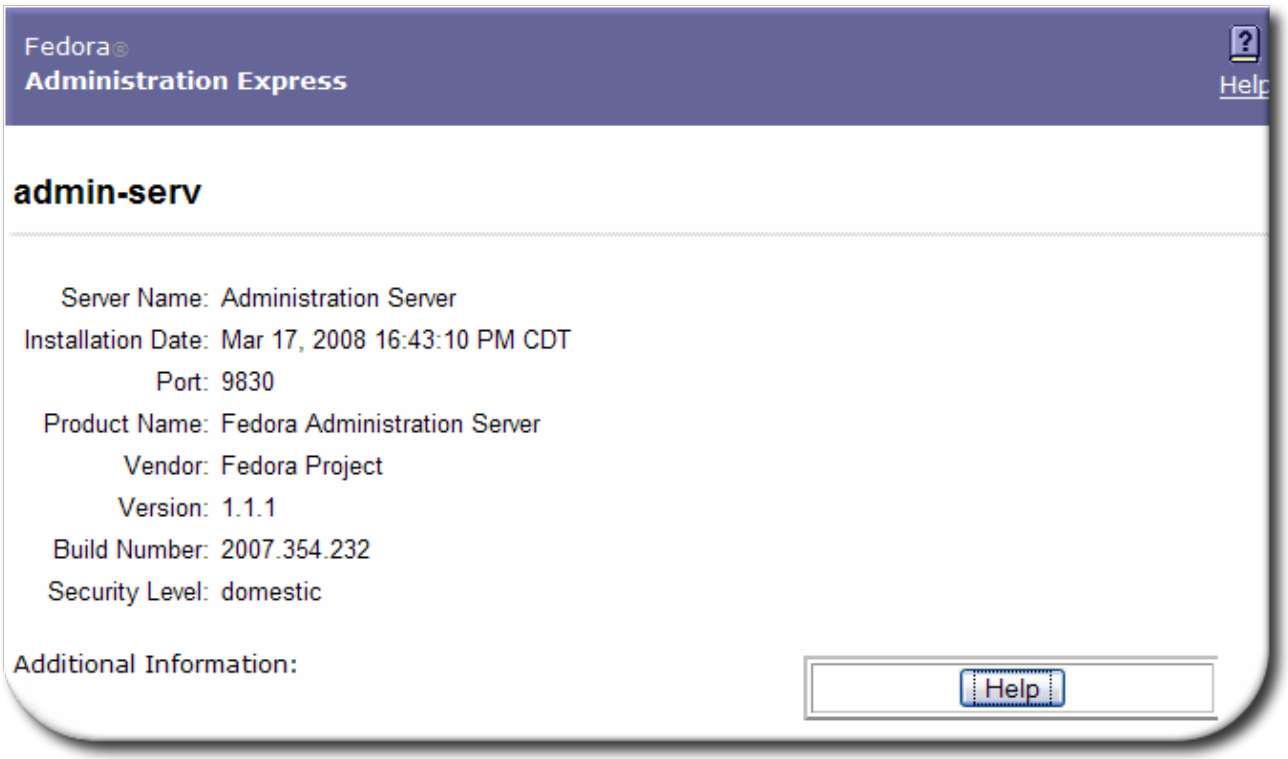
図F.2 ログの確認



F.1.4. サーバー情報の表示

Admin Express ページの Server Info リンクは、ビルド番号、インストール日、サーバーポート番号などのサーバーインスタンスの基本記述を含むページを開きます。これは、インスタンスの選択時にコンソールに表示される情報と同じです。

図F.3 サーバー情報の確認



Directory Server 情報は、/etc/dirsrv/slapd-instance/dse.ldif ファイルにあります。管理サーバー情報は、/etc/dirsrv/admin-serv ディレクトリーの .conf ファイルにあります。

F.2. ADMIN EXPRESS の設定

Admin Express はページの外観に対して編集できますが、ほとんどの機能は Web サーバーまたは Administration Server 設定で制御されるため、設定ファイルを直接編集して編集する必要があります。

F.2.1. Admin Express ファイルの場所

すべての Admin Express 設定ファイルのディレクトリーは 表F.1「Admin Express File directoryies」 に一覧表示されます。特定のファイルは、異なる Admin Express ページ設定を記述する各セクションで説明されています。

表F.1 Admin Express File directoryies

ディレクトリー	詳細
/etc/dirsrv/admin-serv/	管理サーバーを定義し、Web サーバーを設定する local.conf、httpd.conf、およびその他の設定ファイルが含まれます。
/usr/share/dirsrv/html/	Admin Express の外観に使用される HTML ファイルとグラフィックが含まれます。

F.2.2. Admin Express 設定ファイル

Admin Express の動作は主に Web サーバー設定を介して設定されるため、編集しないでください。他の Admin Express 設定は、データまたはフォームフィールドを挿入するディレクティブを介して設定されます。

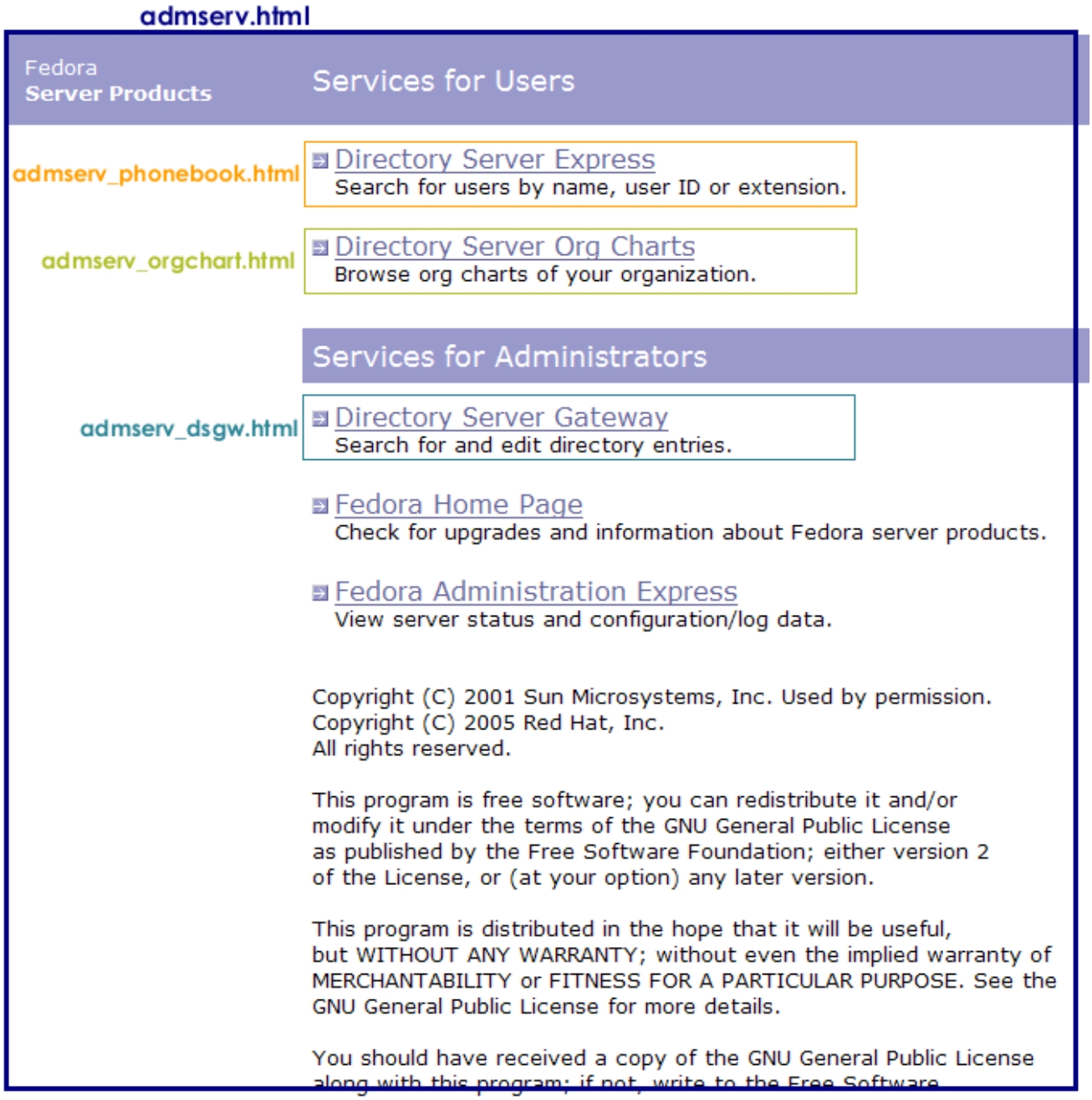
Admin Express のページのフォーマットを一元化するためのスタイルシート(CSS)ファイルはカスケードしていません。すべての形式は、タグまたはページヘッドの <style> タグでインラインで実行されます。インラインタグの編集に関する詳細は、を参照してください http://directory.fedoraproject.org/docs/389ds/administration/htmlediting.html。

F.2.2.1. 管理サーバーの Welcome ページのファイル

Admin Express 用の introductory ページの設定ファイルは、/etc/dirsrv/admin-serv ディレクトリーにあります。1 つのファイルはフォーマット、著作権テキスト、および一部の Web アプリケー

ションテキスト(admserv.html)を設定します。

図F.4 ページ要素の導入



ページのフォーマットはすべてインラインに設定されます。テキストファイルは INCLUDEIFEXISTS ディレクティブを使用して挿入されます。

```
<tr valign="TOP">
  <td> </td>
  <td bgcolor="#9999cc" colspan="4"> <font color="white" size="+1"><font face="Verdana,
sans-serif">Services
    for Administrators</font></font></td>
  <td> </td>
</tr>
<tr valign="TOP">
  <td> </td>
```

```
    <td colspan="4">
      <table border="0" cellspacing="0" cellpadding="0">
        <tr valign="TOP">
          <td><img src="/icons/spacer.gif" width="6" height="6"></td>
          <td></td>
        </tr>
<!-- INCLUDEIFEXISTS admserv_dsgw.html -->
```

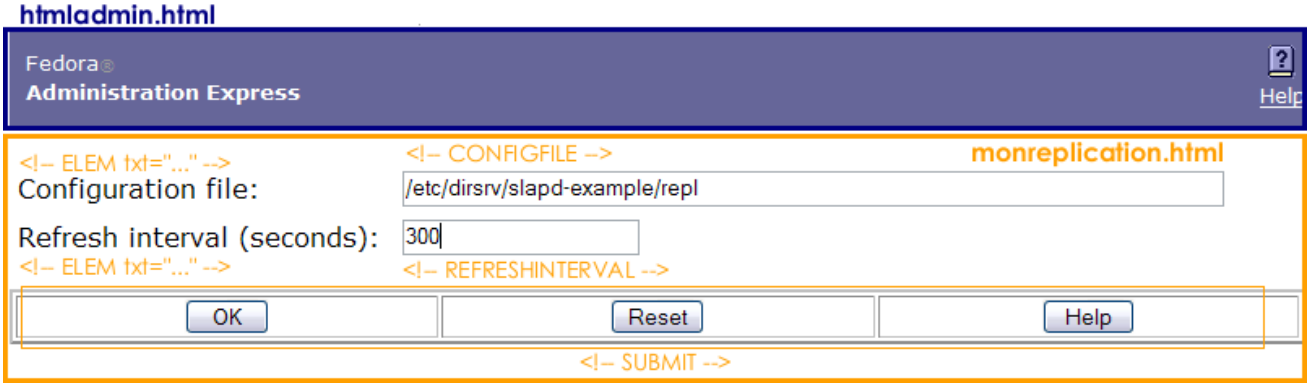
テキストファイル自体には、挿入されたテーブルの行へのインラインフォーマットがあります。

F.2.2.2. Replication Status Appearance のファイル

レプリケーションステータスをモニターするための 2 つのページがあります。これは、以下の 2 つのファイルが必要な設定ページです。

- ページの本文 /usr/share/dirsrv/html/monreplication.html
- ページの /usr/share/dirsrv/html/htmladmin.html

図F.5 レプリケーション設定のページ要素の監視

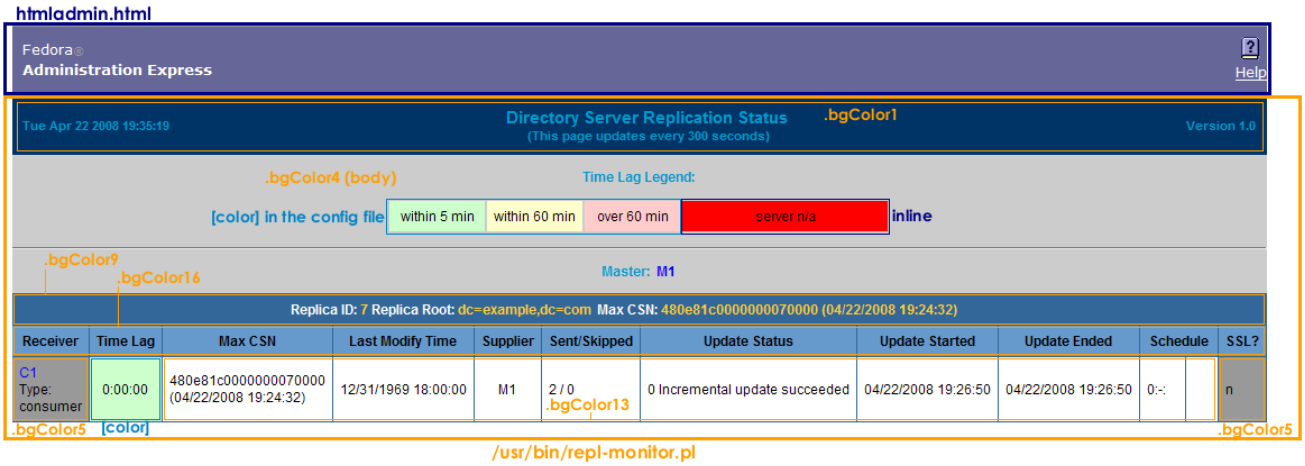


Replication Status ページでは、2 つのスクリプト関連の設定ファイルが使用されます。

- ページの本文。レプリケーション監視スクリプト /usr/bin/repl-monitor.plで設定されます。
- オプションで、レプリケーション監視の設定ファイル。この設定ファイルは、時間ラグ色を [colors] セクションで設定できます。

- ページの /usr/share/dirsrv/html/htmladmin.html

図F.6 レプリケーションビューのページ要素の監視



テーブルの見出し、ラベル、およびページセクションのテキストは Perl スクリプトに設定されます。以下に例を示します。

```
#Print the header of consumer
   print "\n<tr class=bgColor16>\n";
   print "<th nowrap>Receiver</th>\n";
   print "<th nowrap>Time Lag</th>\n";
   print "<th nowrap>Max CSN</th>\n";
....
   print "</tr>\n";
```

Replication Status ページのスタイルが、HTML ヘッダーの <style> タグの Perl スクリプトに出力されます。クラスの多くは、他の Web アプリケーションの style.css と同じです。これらは Perl スクリプトで編集することも、スタイルシートの参照のコメントを解除して CSS ファイルを指定して編集できます。以下に例を示します。

```
 # print the HTML header

   print "Content-type: text/html\n\n";
   print "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 3.2//EN\"><html>\n";
   print "<head><title>Replication Status</title>\n";
   # print "<link type=text/css rel=stylesheet href=\"master-style.css\">\n";
   print "<style text/css>\n";
   print "Body, p, table, td, ul, li {color: #000000; font-family: Arial, Helvetica, sans-serif; font-
size: 12px;}\n";
   print "A {color:blue; text-decoration: none;}\n";
   print "BODY {font-family: Arial, Helvetica, sans-serif}\n";
   print "P {font-family: Arial, Helvetica, sans-serif}\n";
   print "TH {font-weight: bold; font-family: Arial, Helvetica, sans-serif}\n";
   print "TD {font-family: Arial, Helvetica, sans-serif}\n";
   print ".bgColor1  {background-color: #003366;}\n";
   print ".bgColor4  {background-color: #cccccc;}\n";
```

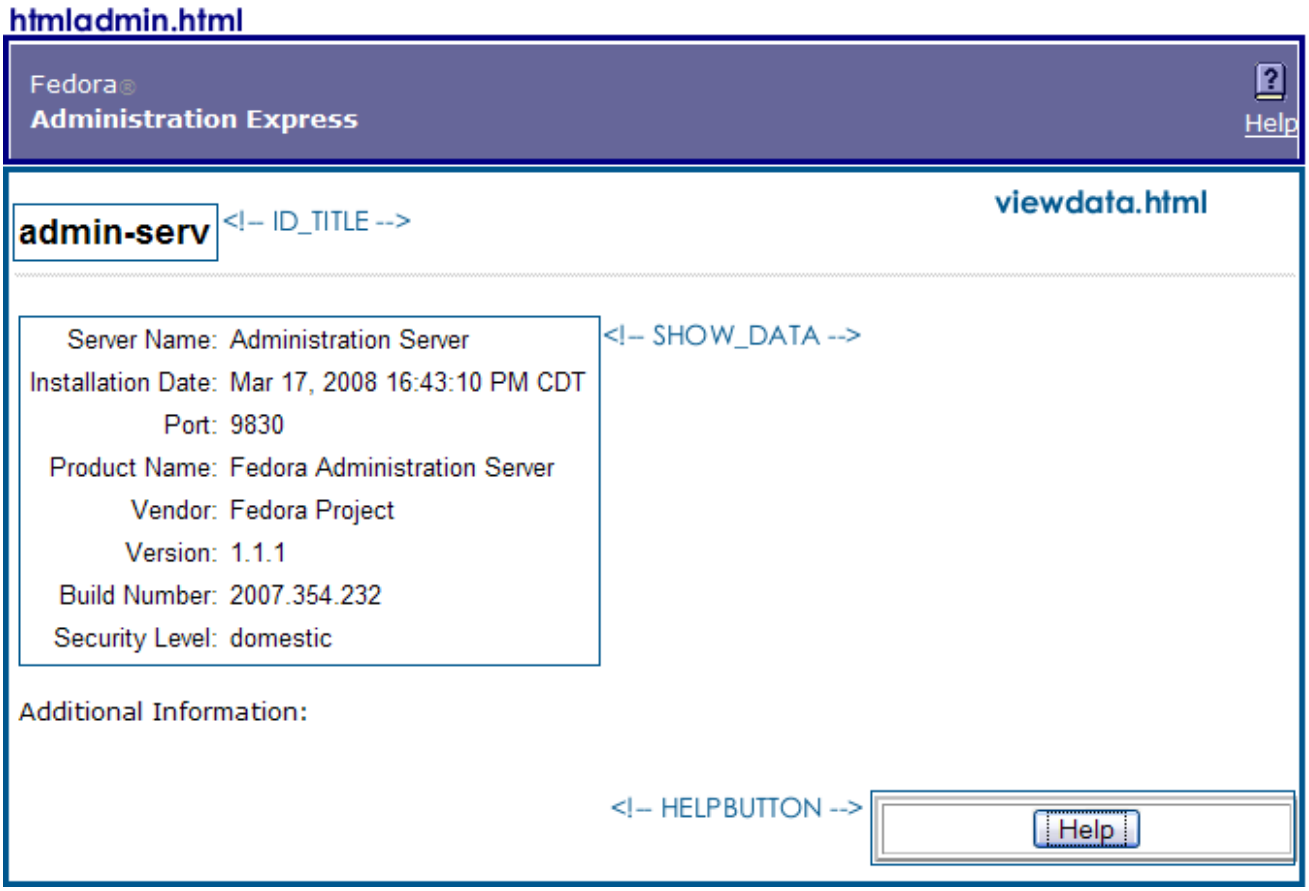
```
    print ".bgColor5  {background-color: #999999;}\n";
    print ".bgColor9  {background-color: #336699;}\n";
    print ".bgColor13 {background-color: #ffffff;}\n";
    print ".bgColor16 {background-color: #6699cc;}\n";
    print ".text8  {color: #0099cc; font-size: 11px; font-weight: bold;}\n";
    print ".text28 {color: #ffcc33; font-size: 12px; font-weight: bold;}\n";
    print ".areatitle {font-weight: bold; color: #ffffff; font-family: Arial, Helvetica, sans-serif}\n";
    print ".page-title {font-weight: bold; font-size: larger; font-family: Arial, Helvetica, sans-
serif}\n";
    print ".page-subtitle {font-weight: bold; font-family: Arial, Helvetica, sans-serif}\n";

    print "</style></head>\n<body class=bgColor4>\n";
```

F.2.2.3. サーバー情報ページのファイル

サーバー情報ページのフォーマットには、以下の 2 つのファイルがあります。

- ページ /usr/share/dirsrv/html/viewdata.html
- ページの /usr/share/dirsrv/html/htmladmin.html

図F.7 サーバー情報のページ要素



viewdata.html ファイルは、2 つのディレクティブのみを使用してサーバーデータを挿入したり、その他のディレクティブを使用して他の情報を挿入したりすることが非常に簡単です。Administration Server では、SHOW_DATA ディレクティブは /etc/dirsrv/admin-serv/local.conf ファイルから情報を取得します。Directory Server では、/etc/dirsrv/slapd-instance/dse.ldif ファイルからデータを取得します。ID_TITLE はサーバーインスタンスの名前です。

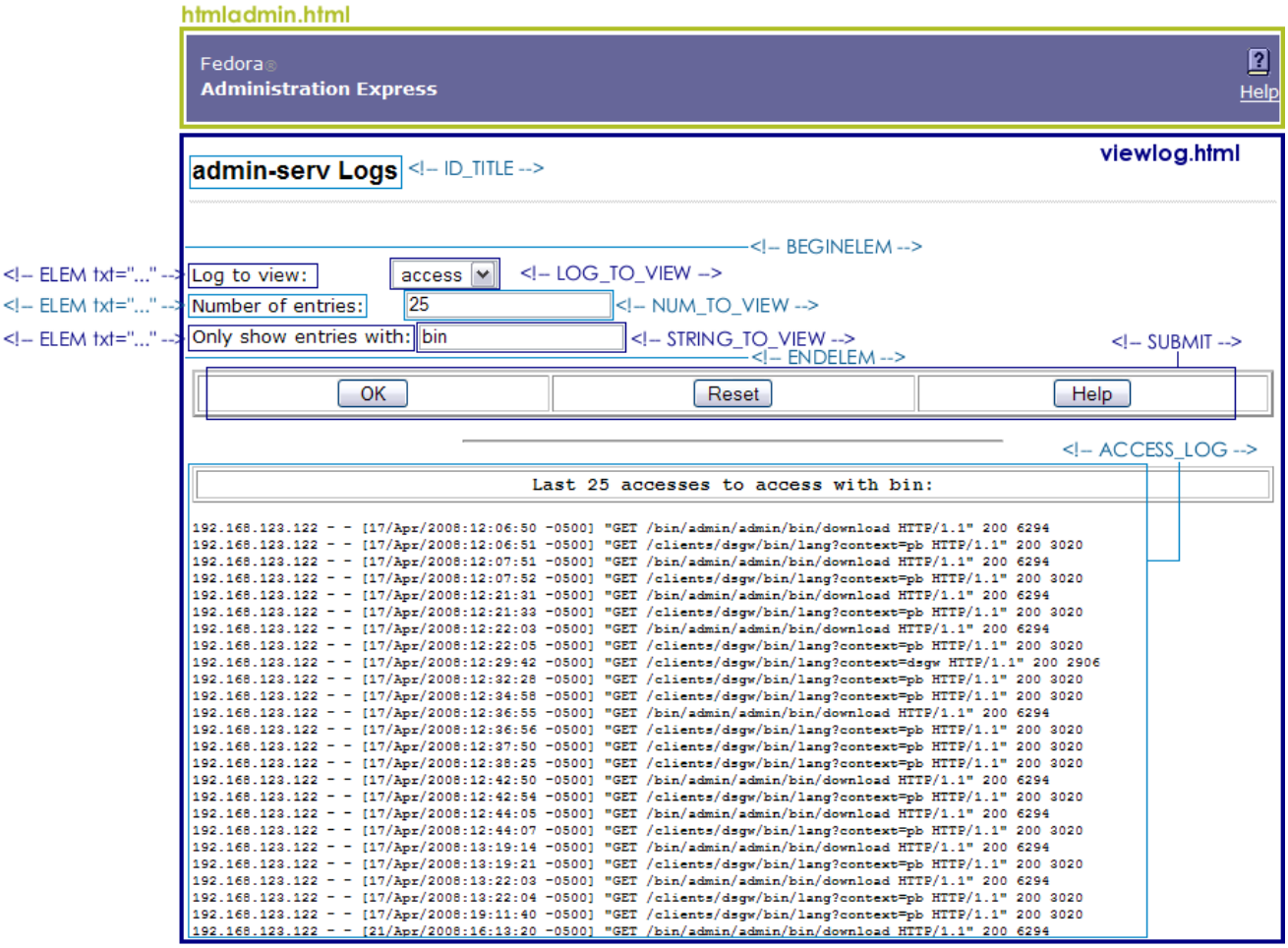
```
<body text="#000000" bgcolor="#FFFFFF" link="#666699" vlink="#666699" alink="#333366">

<br>
<table BORDER=0 CELLSPACING=2 CELLPADDING=2 WIDTH="100%">
<!-- ID_TITLE -->
<p>
<!-- SHOW_DATA -->
<p>
<font face="PrimaSans BT, Verdana, sans-serif"><font size=-1>Additional Information:</font>
</font>
<p>
<!-- CHECK_UPGRADE -->
<p>
<!-- SHOW_URL -->
</table>

<!-- HELPBUTTON -->

</body>
```

F.2.2.4. サーバーログページのファイル

サーバーログページをフォーマットする 2 つのファイルがあります。

- *ページ /usr/share/dirsrv/html/viewlog.html*
- *ページの /usr/share/dirsrv/html/htmladmin.html*

図F.8 ログ表示ページ要素



ページ情報は、挿入されたディレクティブで設定されます。サーバーインスタンス名は ID_TITLE ディレクティブに設定されます。ログは、ACCESS _LOG ディレクティブに表示されます。上部のフォームはディレクティブのペアでフォーマットされ、説明テキストとその他のフィールドタイプを挿入します。たとえば、これによりログタイプメニューが設定されます。

```
<form method=GET action=ViewLog>
<font face="PrimaSans BT, Verdana, sans-serif"><font size=-1>
<!-- BEGINELEM -->
<!-- ELEM txt="Log to view:          " -->
<!-- LOG_TO_VIEW -->
....
<!-- SUBMIT -->
</font></font>
</form>
```

F.2.3. Admin Express ディレクティブ

Admin Express ディレクティブは、CGI スクリプトによって解釈される HTML コメントです。これらのディレクティブは、フォームフィールドを設定し、サーバー設定およびログファイルからデータを取得するために使用されます。

表F.2 Admin Express ディレクティブ

ディレクティブ	詳細	例
ACCESS_LOG	サーバーログファイルを挿入します。	<!-- ACCESS_LOG -->
ADMURL		<!-- ADMURL -->
BEGINELEM	フォームの入力要素のオープンとマークします。これは、常に ENDELEM のペアです。	<!-- BEGINELEM -->
CHECK_UPGRADE		<!-- CHECK_UPGRADE -->
ELEM	テキスト要素を挿入します。これには、使用するテキストを定義する引数である txt= が 1 つあります。	<!-- ELEM txt="Field name here: " -->
ELEMADD	テキスト要素を挿入します。これには、使用するテキストを定義する引数である txt= が 1 つあります。	<!-- ELEMADD txt="Field name here: " -->
ENDELEM	フォームの入力要素の最後をマークします。これは、BEGINELEM と常にペアになります。	<!-- ENDELEM -->
HELP_BUTTON	コンテキスト固有のヘルプを開くボタンを挿入します。	<!-- HELP_BUTTON -->
HELPLINK	一般的な Admin Express ヘルプファイルへのリンクを挿入します。	<!-- HELPLINK -->
HIDDEN_ID		<!-- HIDDEN_ID -->
ID_TITLE	admin-serv や example などのサーバーインスタンスの名前を挿入します（Directory Server インスタンス名が slapd-example の場合）。	<!-- ID_TITLE -->

ディレクティブ	詳細	例
INCLUDEIFEXISTS	HTML ファイルの内容を挿入します。挿入されたファイルには、テキストと HTML マークアップの両方を含める必要があります。	<!-- INCLUDEIFEXISTS "file.html" -->
LOG_TO_VIEW	表示可能なログの種類でドロップダウンメニューを挿入します。	<!-- LOG_TO_VIEW -->
NUM_TO_VIEW	フォームフィールドを挿入し、返す行数を設定します。	<!-- NUM_TO_VIEW -->
REFRESHINTERVAL	フォームフィールドを挿入し、レプリケーション監視の更新間隔（秒単位）を設定します。	<!-- REFRESHINTERVAL -->
SERVHOST		<!-- SERVHOST -->
SERVPORT		<!-- SERVPORT -->
SHOW_DATA	ポート番号、インストール日、ビルド番号など、設定ファイルからサーバーデータを挿入します。	<!-- SHOW_DATA -->
SHOW_URL		<!-- SHOW_URL -->
SITEROOT		<!-- SITEROOT -->
STRING_TO_VIEW	ログの検索文字列を設定するのに使用するフォームフィールドを挿入します。	<!-- STRING_TO_VIEW -->
送信	3 つのボタンセットを挿入する： フォームを保存または送信してください。フォームをリセットして、ヘルプトピックを開きます。	<!-- SUBMIT -->

付録G コンソールの使用

G.1. DIRECTORY SERVER コンソールの概要

Red Hat Management Console は、Red Hat Directory Server および管理サーバー設定およびディレクトリー情報を管理するユーザーインターフェースです。サーバーを管理する単一のメインコンソールウィンドウがあります（管理ドメインで集めて識別されます）。メインコンソールを使用すると、サーバー固有のコンソールを開き、個別のインスタンスの設定と情報を管理できます。

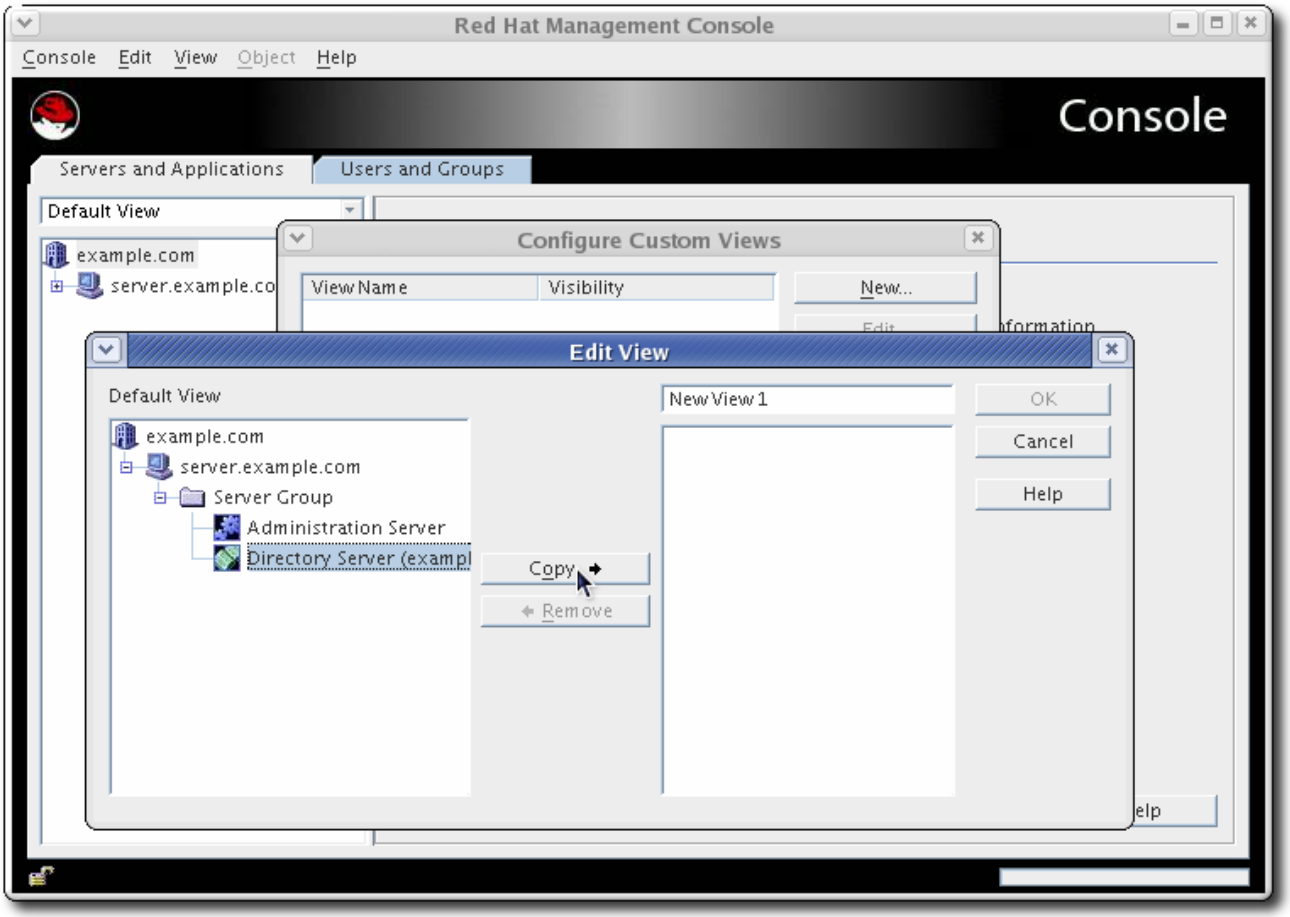
本章では、コンソールが Directory Server と管理サーバーと相互作用する方法の概要と、コンソールのウィンドウとオプションを説明します。

G.1.1. コンソール、Directory Server、および管理サーバーの機能

Red Hat Console は独立した Java アプリケーションで、Red Hat Directory Server および Administration Server のインスタンスと連携します。ほとんどのサーバー管理機能は、Directory Server および Administration Server のサーバー固有のコンソールウィンドウで実行されます。Red Hat Console は、Red Hat Directory Server インスタンスと管理サーバーを管理するシステムの一部であるため、ディレクトリーの情報となります。Red Hat Directory Server、Red Hat Management Console、および Red Hat Administration Server は相互に連携しますが、サーバー、アプリケーション、ユーザーの管理に特定の役割を果たします。

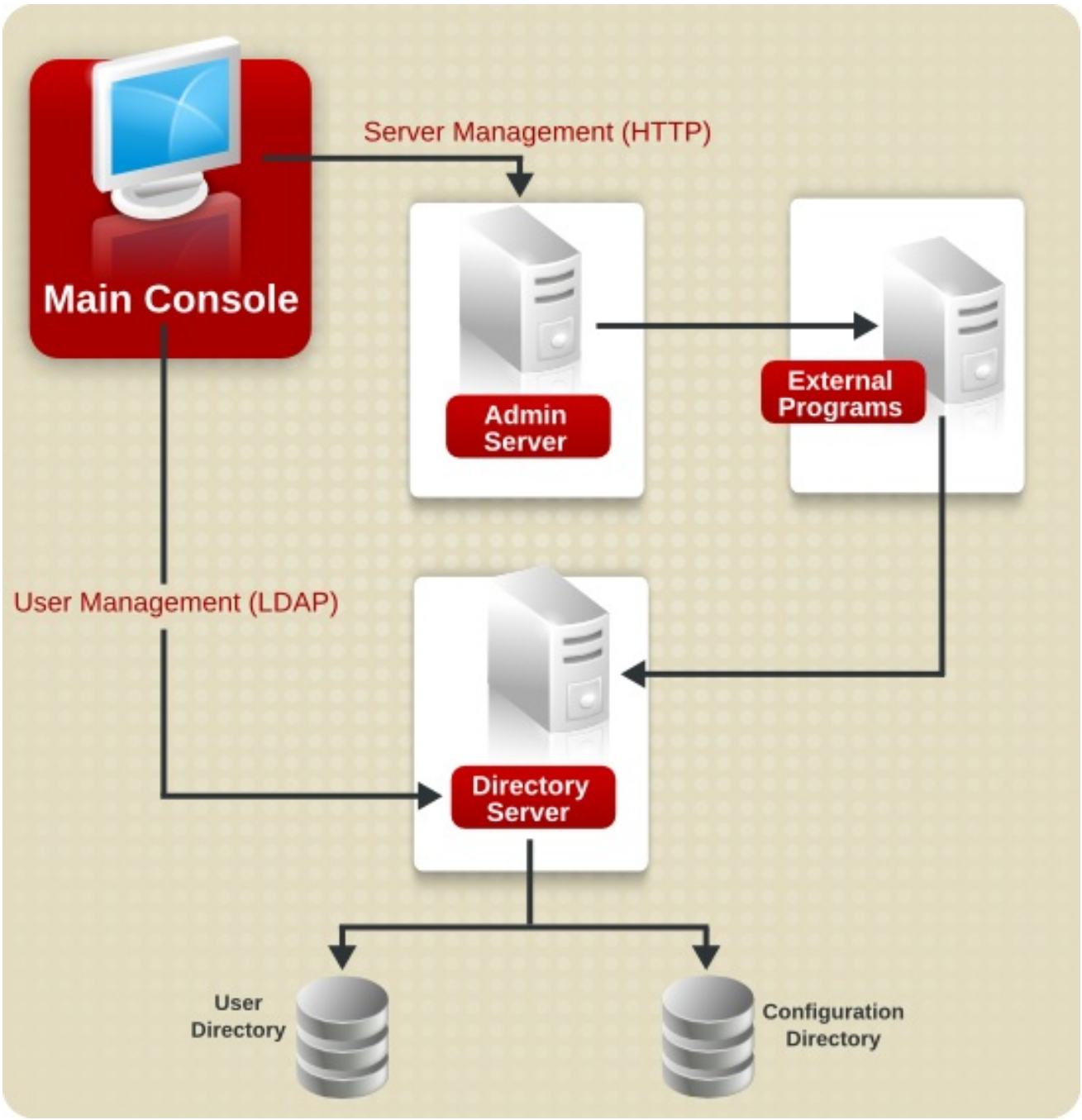
Red Hat Management Console は、Red Hat Directory Server のフロントエンド管理アプリケーションです。これは、設定ディレクトリーに登録されているすべてのサーバーおよびアプリケーションを検索し、グラフィカルインターフェースで表示し、そのサーバーを管理および設定できます。また、メインコンソールは、ユーザーディレクトリーでユーザーおよびグループのエントリーを検索、作成、および編集することもできます。

図G.1 Red Hat 管理コンソールインターフェース



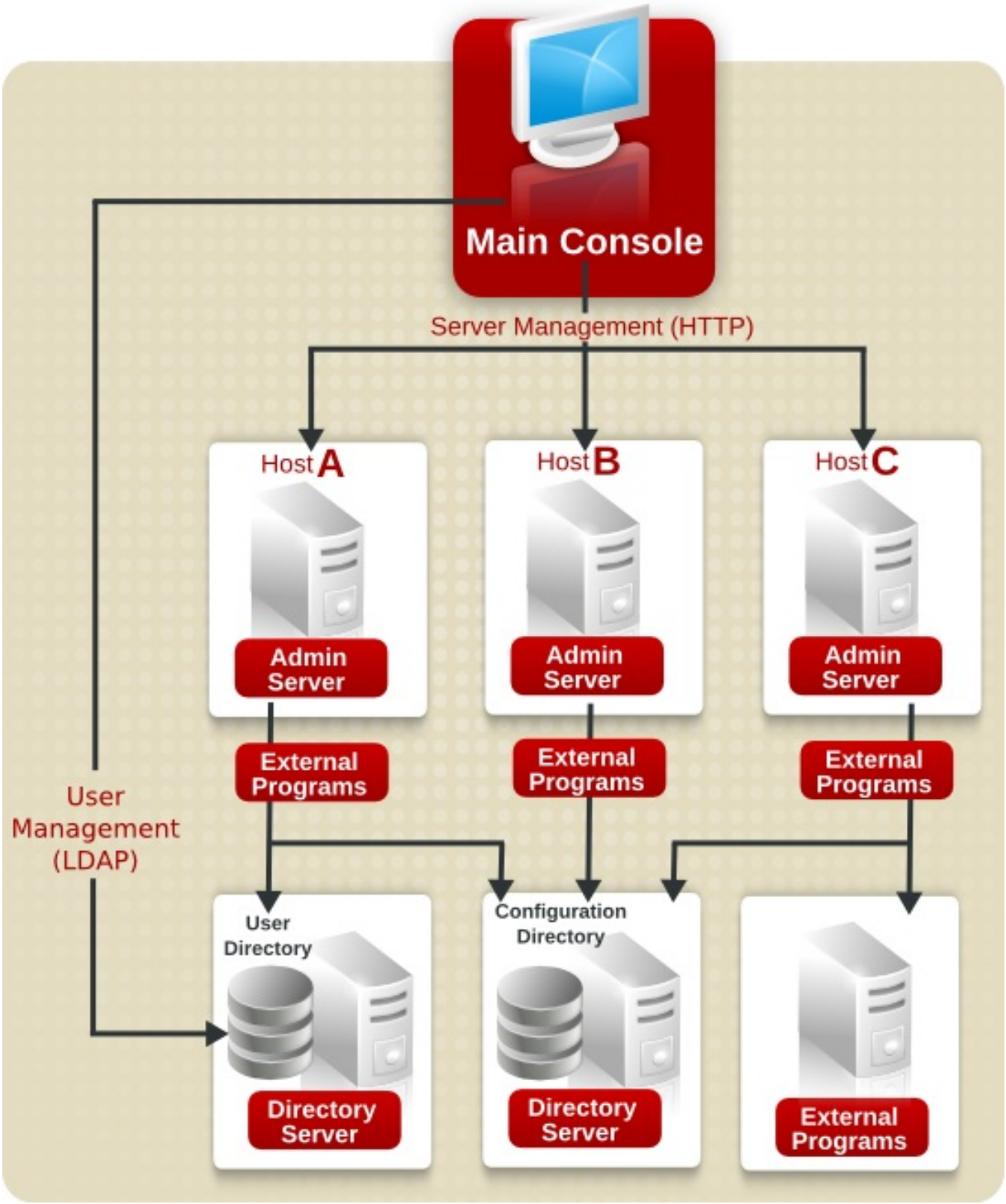
ユーザーが Red Hat 管理コンソールにログインすると、コンソールは Hypertext Transfer Protocol(HTTP)経由で管理サーバーに接続します。管理サーバー は、さまざまな Directory Server インスタンスを管理するための要求を受け取り、ポート番号の変更など、設定への変更を実行します。Red Hat 管理コンソールに要求を送信すると、ユーザーエントリーの追加または編集が行われると、コンソールは Lightweight Directory Access Protocol(LDAP)メッセージを直接 Directory Server に送信し、ユーザーディレクトリーを更新します。

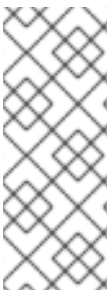
図G.2 Red Hat 管理コンソールを使用したシンプルなシステム



Red Hat Directory Server は、サーバーおよびアプリケーションの設定とユーザー情報を保存します。通常、アプリケーションとサーバー設定情報は Red Hat Directory Server のサブツリーに保存されますが、ユーザーエントリーおよびグループエントリーは別のサブツリーに保存されます。ただし、企業が大きい場合は、設定およびユーザー情報を Directory Server の別のインスタンス に保存できます（同じホストマシン上、または 2 つの異なるホストマシン上にあります）。図G.2「Red Hat 管理コンソールを使用したシンプルなシステム」 は比較的シンプルな Red Hat Directory Server システムを示しています。企業が拡大および変更されると、コンソールの管理コンソールの管理対象ドメインに追加のホストおよび Directory Server を追加できます。これにより、単一コンソールで複数の Directory Server を管理できます。

図G.3 より詳細な複雑なシステム





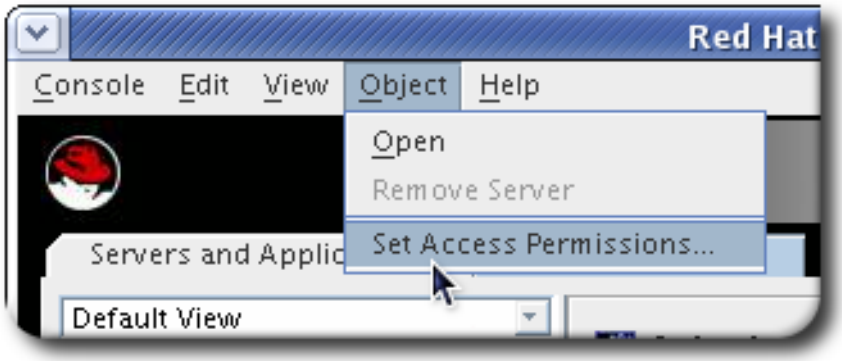
注記

本書で設定ディレクトリー とユーザーディレクトリーが使用される場合は、Directory Server の単一インスタンスのサブツリーか、Directory Server の 2 つのインスタンスにあるかに関係なく、設定情報とユーザー情報が保存される場所を定義します。

G.1.2. Red Hat 管理コンソールメニュー

コンソールのトップメニューには、5 つのメニュー項目があります。これらのメニューの各オプションは、コンソールウィンドウ（メインコンソール、Directory Server Console、または 管理コンソール）とそのサーバーエリアで利用可能なオブジェクトタイプによって異なります。

図G.4 メインコンソールメニュー



表G.1 コンソールメニュー

メニュー	詳細
コンソール	ウィンドウを閉じるか、セッションを完全に終了したりなど、コンソールセッションを管理します。 • メインウィンドウでは、このメニューを使用して admin ドメインを追加および削除できます。 • Directory Server コンソールでは、ユーザーは別のユーザーとしてログインできます。 • 管理コンソールの場合は、証明書やトークンなどのセキュリティー問題を管理します。
Edit	3 つのコンソールすべてに表示設定を設定します。Directory Server コンソールでは、ディレクトリーエントリーまたはテキストをコピー、貼り付け、削除する方法も提供します。

メニュー	詳細
表示	トップバナー、メニュー、サイドナビゲーションペインなど、コンソールウィンドウの特定部分を表示するかどうかを設定します。これにより、現在の表示も更新されます。Directory Server コンソールでは、このメニューはディレクトリーまたは表示するデータベースの一部を設定します。
オブジェクト	アクティブオブジェクトで利用可能な操作を提供します。これは、アクティブエリアまたはエントリーの右クリックメニューと同じです。 • メインウィンドウでは、このメニューがサーバーインスタンスを開くか、または削除します。 • Directory Server コンソールでは、高度なプロパティーエディターや新規エントリーの作成など、ディレクトリーエントリーのすべての設定オプションが提供されます。 • 管理コンソールのコンソールでは、設定エディターが開き、サーバーが起動し、停止します。
ヘルプ	現在のコンソールエリアのコンテキスト固有のヘルプを開きます。

G.1.3. Red Hat Management Console タブ

メインのコンソールウィンドウには、以下の 2 つのタブがあります。

- Directory Server インスタンス および管理 サーバーインスタンスを管理するサーバーおよびアプリケーション
- ユーザーおよびグループ。 Directory Server でユーザーエントリーおよびグループエントリーを検索および作成する

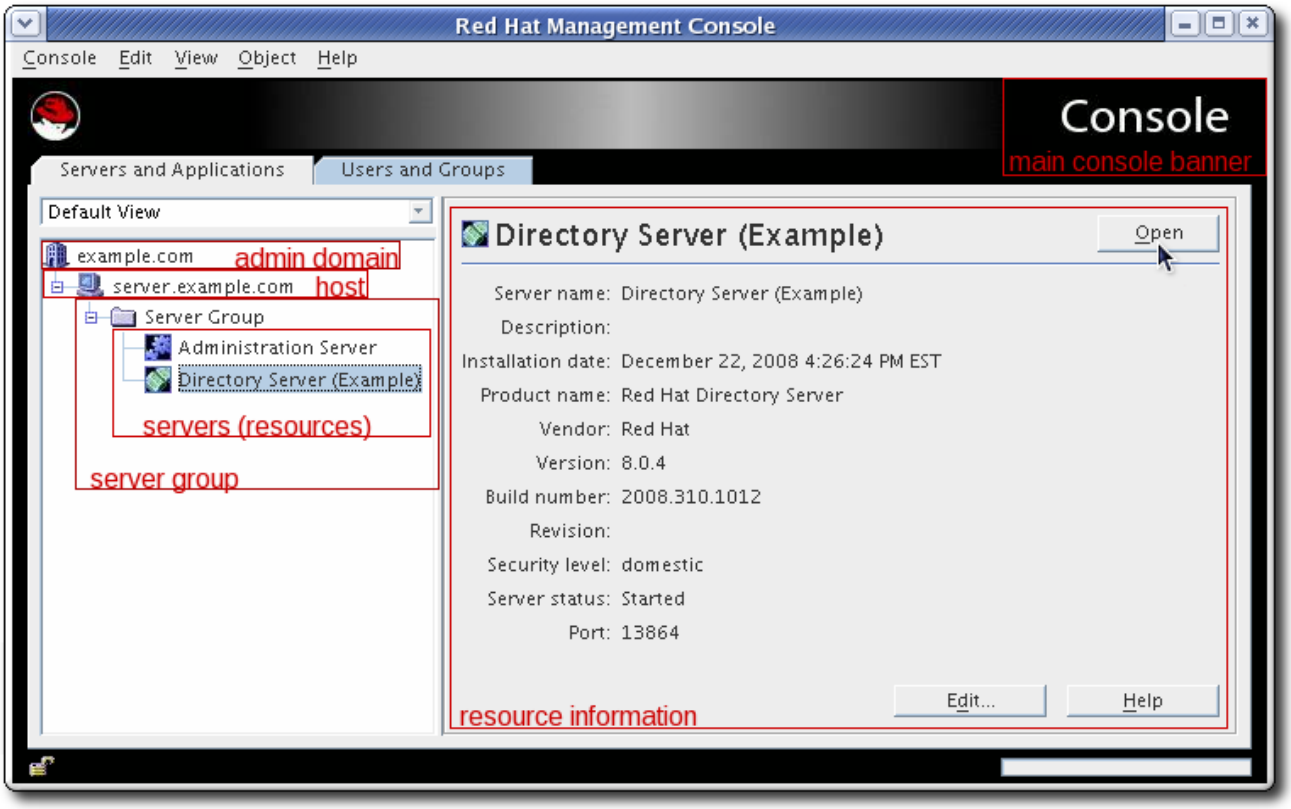
G.1.3.1. 「Servers and Applications」タブ

デフォルトでは、Servers and Applications タブは、ホスト、ディレクトリー、および管理サーバー、および中央情報パネルを表示するために、左側にナビゲーションツリーがあります。Directory

Server インスタンス、ディレクトリー情報、または 管理 Server にアクセスするには、ナビゲーションツリーに記載されているサーバーリソースを開きます。ビルド番号やポート番号などのサーバーインスタンスの情報

ナビゲーションツリーは、設定ディレクトリーに登録されているすべてのリソース（サーバーやホストなど）の階層的な表現である Red Hat Directory Server トポロジー を表示します。

図G.5 「Servers and Applications」タブ



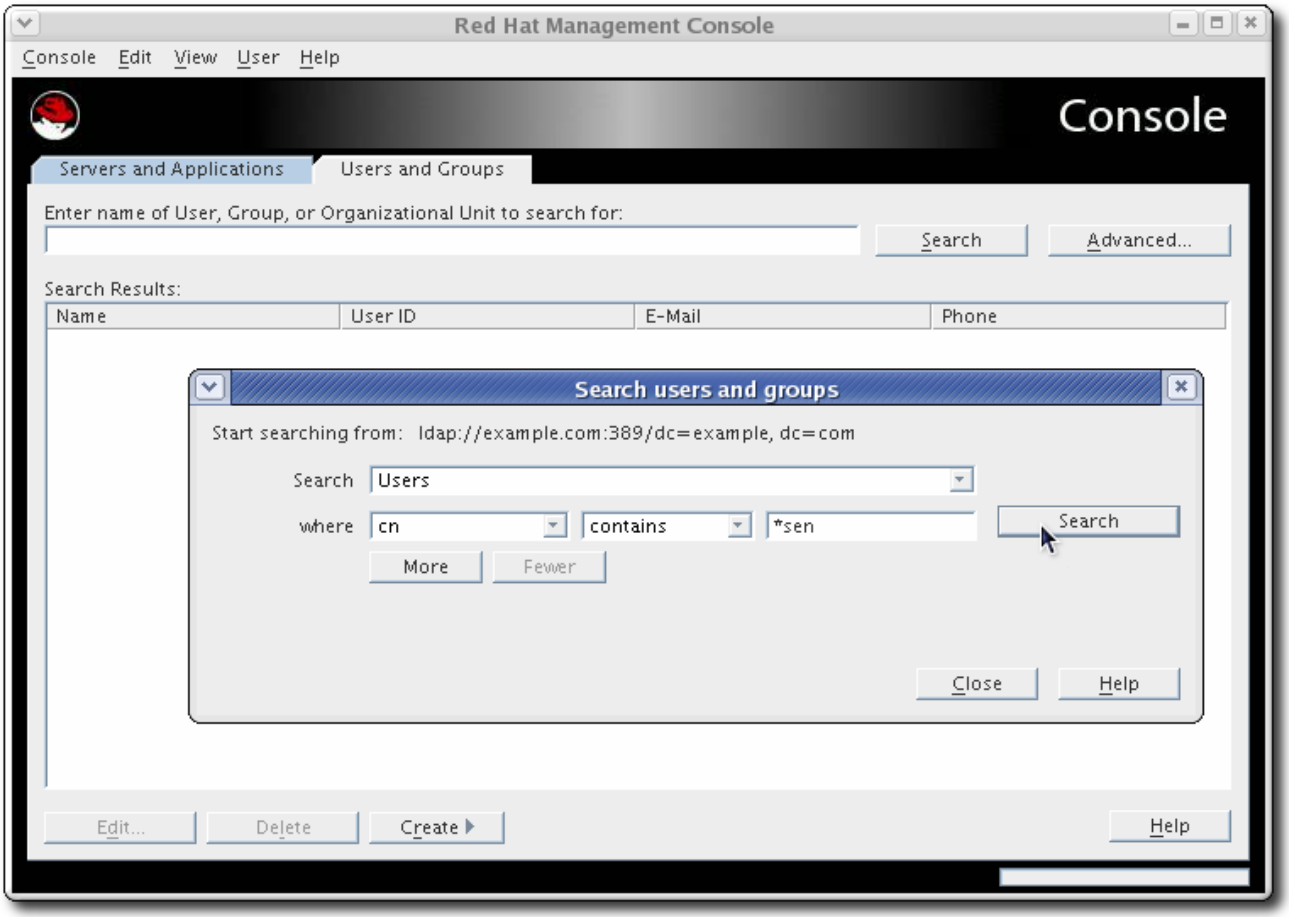
トポロジーの最上部は管理ドメインです。管理ドメインは、同じユーザーディレクトリーを共有するホストシステムおよびサーバーのコレクションです。Directory Server インスタンスまたは Administration Server インスタンスをホストするサーバーは管理ドメインに属し、ホストです。

サーバーグループ は、共通の管理サーバーが管理するすべての Directory Server で構成されます。管理ドメイン内に多くのサーバーグループが存在する可能性があります。

G.1.3.2. ユーザーおよびグループタブ

Users and Groups タブで、コンソールが管理する Directory Server のユーザーエントリーおよびグループエントリーを検索できます。このタブで、返されるエントリーのいずれかを編集または削除できます。新しいエントリーは、Users タブおよび Groups タブから作成することもできます。

図G.6 ユーザーおよびグループタブ



「ユーザーおよびグループの検索」 *で説明されているように、検索されるディレクトリーの切り替え、または* **Users** *メニューのオプションでエントリーが追加される場所を切り替えます。*

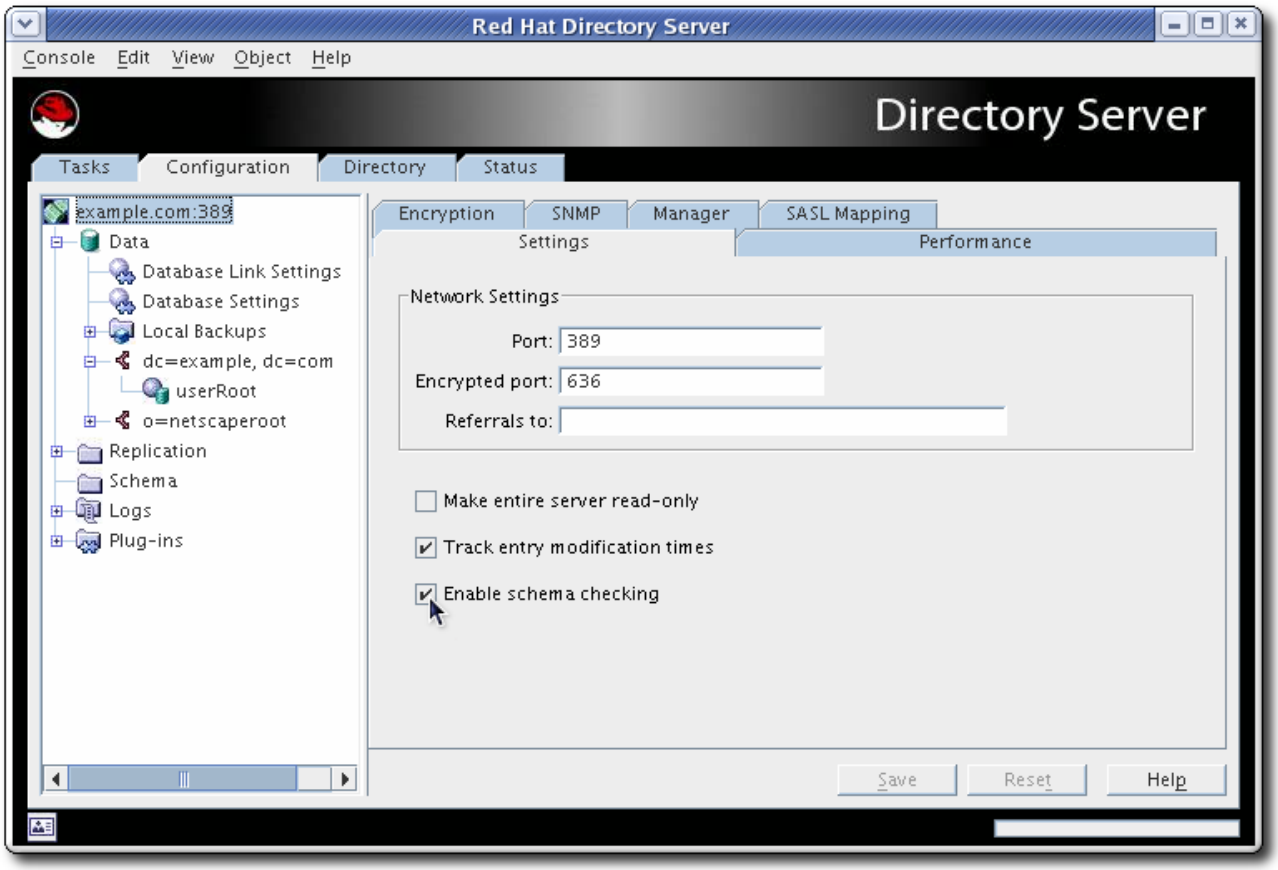
G.1.4. サーバー固有のコンソール

メインコンソールは 2 つのサーバー固有のウィンドウに開き、管理サーバーと **Directory Server** *を管理できます。これらのウィンドウは、ナビゲーションエリアのサーバー名をクリックしてから、リソースエリアの* **Open** *ボタンをクリックして開きます。*

G.1.4.1. Directory Server コンソール

***Directory Server** コンソールは、ポート番号、TLS 設定、ロギングなどの特定の **Directory Server** インスタンス設定を管理します。**Directory Server** コンソールは、データベースのインポートおよびエクスポート、接尾辞の作成、スキーマの拡張などのディレクトリー情報（エントリー）とディレクトリー操作も管理します。*

図G.7 Directory Server コンソール



Directory Server コンソールには、4 つのタブがあります。

- *Directory Server インスタンスの起動/停止、データベースのインポートおよびエクスポート、TLS 証明書の管理など、一般的なサーバー操作へのショートカットがある タスク*
- *SASL および TLS 認証 、 ポート番号、レプリケーション、同期、データベースおよびサフィックス、ロギング、プラグインなど、すべてのサーバー設定を定義する設定*
- *ユーザーエントリーおよび 全グループエントリー（ロール、サービスクラス、ビュー、およびグループを含む）などのディレクトリー情報にアクセスし、管理します。*
- *status: サーバーパフォーマンスを監視し、Directory Server およびデータベースに異なる監視およびパフォーマンスカウンターを表示します。*

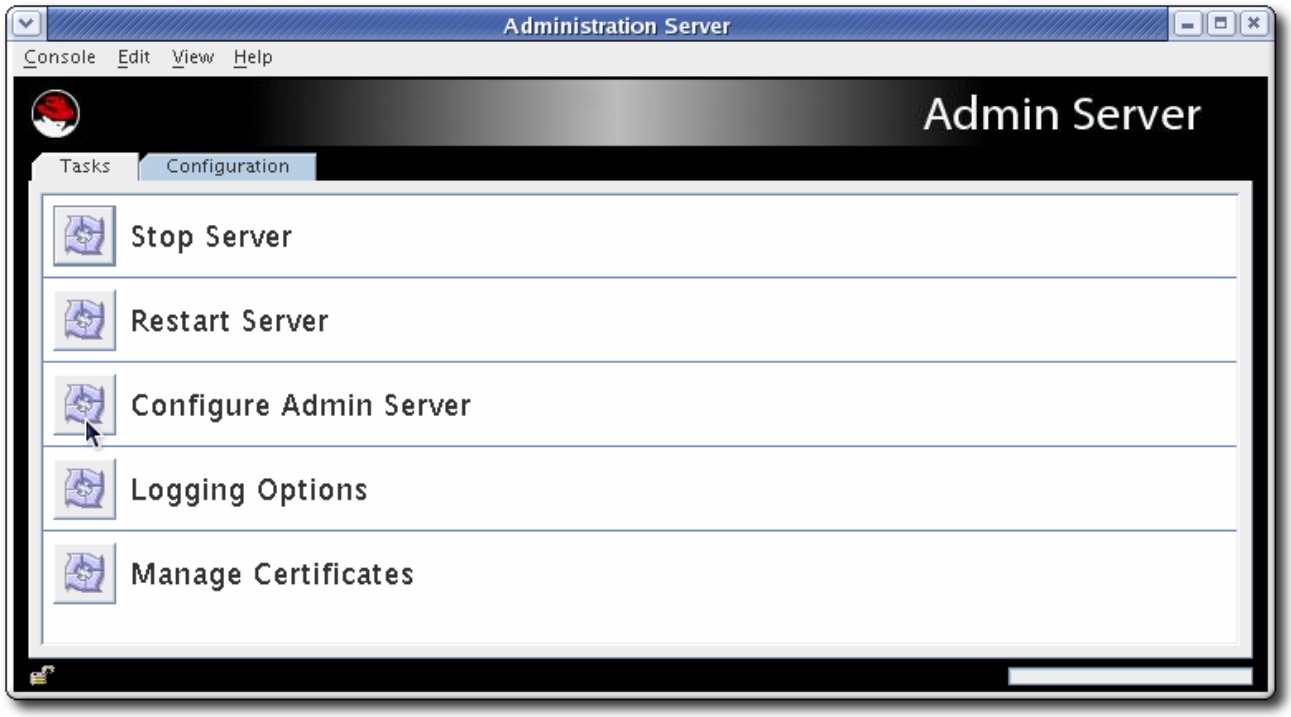
メインコンソールと同様に、Directory Server Console タブには、左側のナビゲーション領域と、アクティブな設定、エントリー、またはデータベースに関する情報を表示する中央パネルがあります。

Directory Server コンソールを使用して Directory Server 設定およびディレクトリーエントリーを管理する方法は、『 『Red Hat Directory Server 管理ガイド』』 で説明しています。

G.1.4.2. 管理コンソール

管理サーバー自体は、特にサーバーグループの設定やユーザーディレクトリーなど、他のサーバーの設定を管理します。管理コンソールは、管理コンソールの設定とこれら 2 つの Directory Server ディレクトリーの設定を管理します。Directory Server 設定で設定を変更するたびに、サーバーはこれらのサーバーを適切に管理するために管理サーバー設定に行う必要があります。

図G.8 管理コンソール



管理コンソールは Directory Server コンソールよりも簡単です。以下の 2 つのタブのみがあります。

- *タスク （管理サーバーインスタンスの起動および停止、ロギングの設定、TLS 証明書の管理など、一般的なサーバー操作へのショートカットを含む）*
- *TLS 認証 、 ポート番号、ロギングなどの管理サーバー構成設定や、管理サーバー がディレクトリーサービスに接続するために使用される Configuration Directory Server および User Directory Server 設定など、すべての管理 Server 設定を定義する設定*

管理コンソールを使用して管理サーバー設定と関連するディレクトリーサービスを管理する手順は、『 『Using the Admin Server』』 で説明しています。

G.2. コンソールアプリケーションの変更

コンソールの異なる要素に使用されるフォントを編集できます。フォント設定およびフォントプロファイルが保存される場所をカスタマイズできます。デフォルトのフォント設定は簡単に復元できます。

本セクションでは、コンソールの外観の他の側面を制御する方法を説明します。たとえば、テーブルの列を簡単に再配置できます。表示されるサーバーインスタンス（ナビゲーションビューと呼ばれる）を制御することもできます。これにより、サーバーインスタンスを簡単にソートし、見つけることができます。

アクセス制御の手順は、「アクセス制御の設定」で説明されているユーザーインターフェース要素に適用できます。

- 「プロファイルの場所の変更」
- 「デフォルトのフォント設定の復元」
- 「コンソールフォントの変更」
- 「テーブル列の並べ替え」
- 「メインウィンドウのカスタマイズ」

G.2.1. プロファイルの場所の変更

コンソールのフォーマットは プロファイル に保存されます。エントリーのプロファイルはローカルに保存できます。つまり、特定のワークステーションでのみ利用可能なか、設定ディレクトリーに格納できるため、どこからでもアクセスできます。

プロファイルの場所を設定するには、以下を実行します。

1.

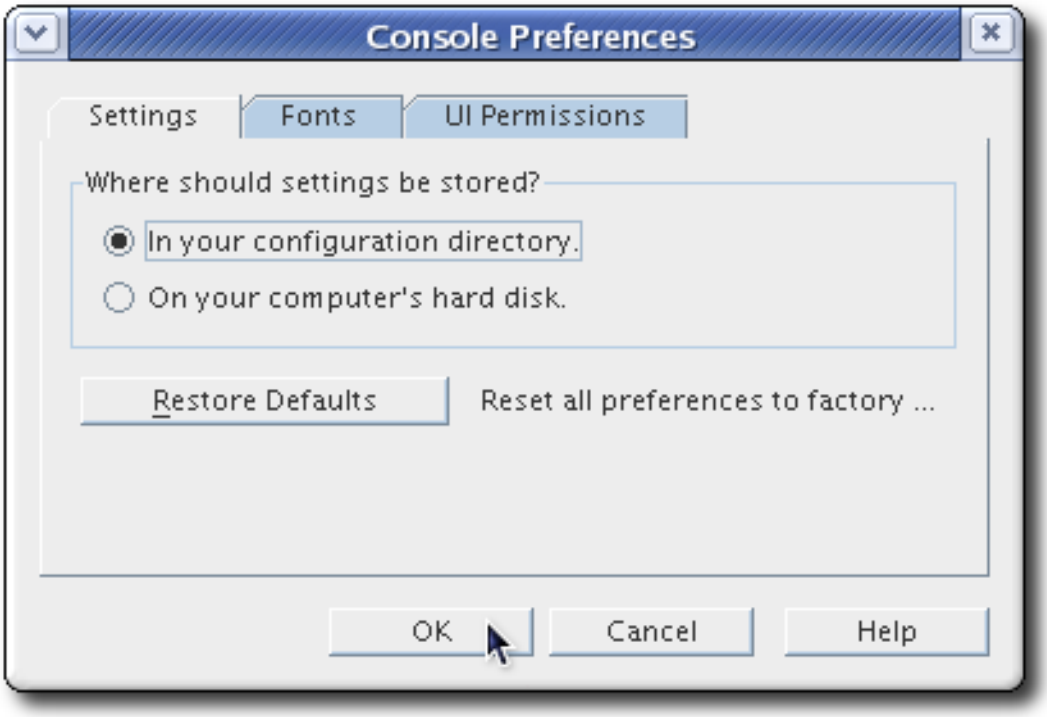
トップメニューで **Edit** をクリックし、**Preferences** を選択します。

2.

Settings タブをクリックします。

3.

設定を保存する場所のラジオボタンを選択します。



- 設定ディレクトリーでは、設定が **Directory Server** 設定に保存され、コンソールにどこからでも利用できるようにすることができます。

- お使いのコンピューターのハードディスクでは、設定プロファイルをローカルに保存します。これは主に、ワークステーションやラップトップなど、異なるコンソールで使用される特定の設定が必要な場合に有用です。

4.

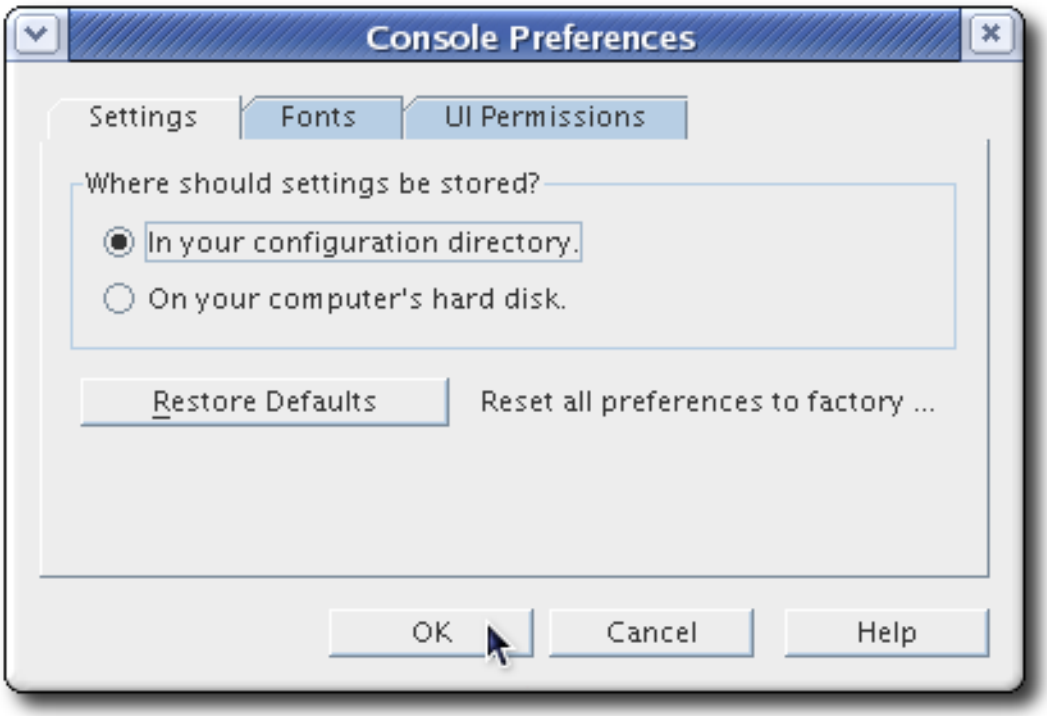
OK をクリックします。

G.2.2. デフォルトのフォント設定の復元

1. トップメニューで **Edit** をクリックし、**Preferences** を選択します。

2. **Settings** タブをクリックします。

3. **Restore Defaults** ボタンをクリックして、デフォルトの表示設定に戻ります。



4. **OK** をクリックします。

G.2.3. コンソールフォントの変更

テーブルの見出しや通常のテキストなど、コンソールの異なる部分には異なるフォント設定があります。フォント設定は プロファイル に保存されます。プロファイルは、すべてのテキスト要素のフォントファミリー、サイズ、およびフォーマットを定義します。複数のフォントプロファイルを利用でき

ます。また、フォントプロファイルは、すべてのユーザーがそれらにアクセスできるように、特定のユーザーまたはグループの設定やパブリックなどのプライベートにすることができます。

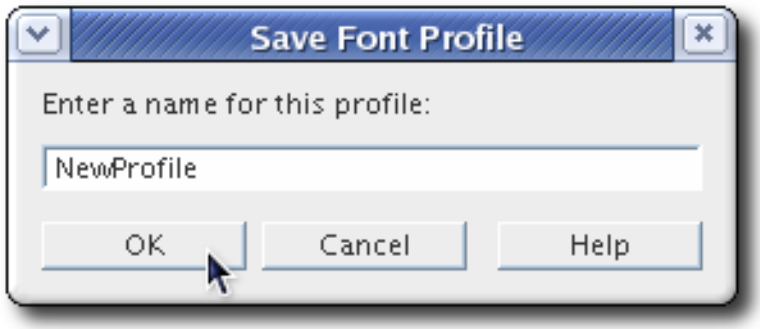
デフォルトのプロファイルは、新規プロファイルを作成せずに編集できます。

フォントプロファイルを編集または作成するには、以下を実行します。

1. メインの **Red Hat Management Console** ウィンドウで、**Edit** メニューから **Preferences** を選択します。

2. **Fonts** タブをクリックします。

3. 新規設定を新規プロファイルとして保存するには、**Save As** ボタンをクリックして、プロファイル名を入力します。

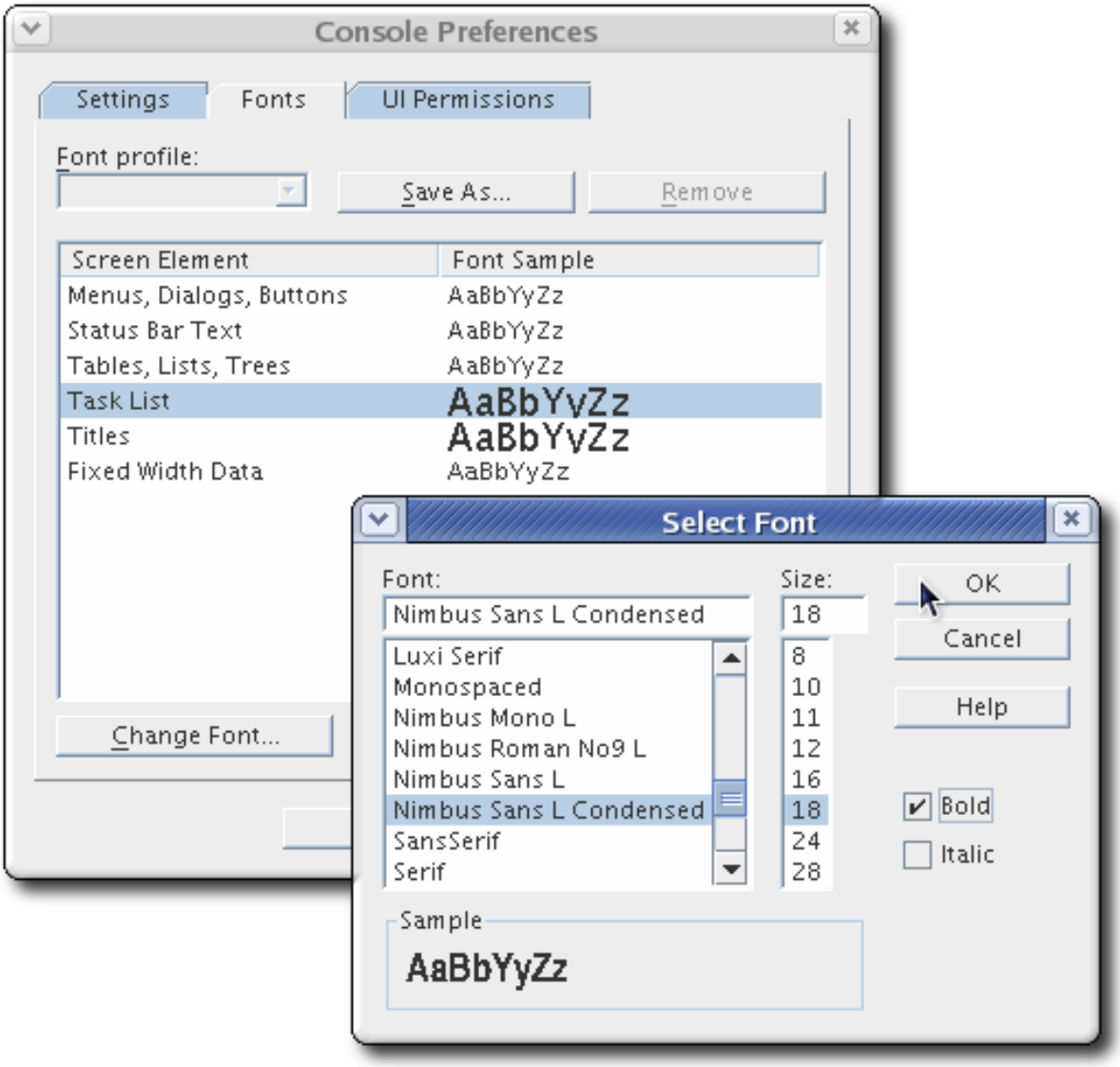


デフォルト（または現在の）プロファイルを編集するには、フォントの編集を開始します。

4. スクリーン要素 のコラムで編集する画面要素をクリックし、**Change Font** ボタンをクリックします。

5.

その特定の要素のフォントを編集します。フォントファミリー、サイズ、およびフォーマット（太字またはイタリック）の 3 つの設定が可能です。



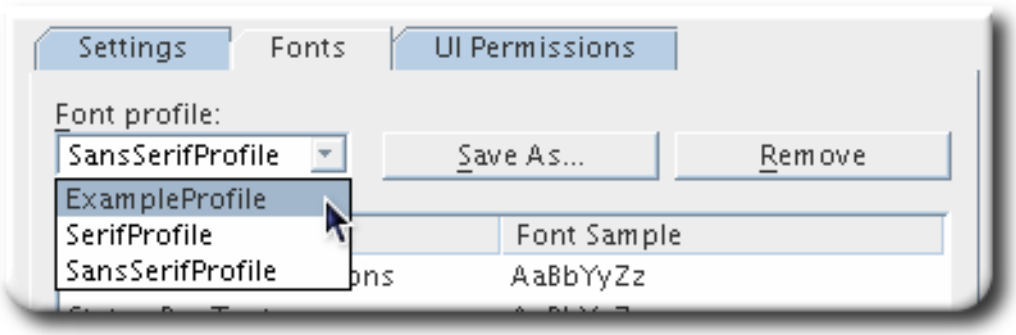
6.

OK をクリックしてプロファイルを保存します。

7.

コンソールを再起動して変更を適用します。

保存されたフォントプロファイルをロードおよび使用するには、Preference ダイアログの Font タブを開いて、使用するフォントプロファイルを選択し、OK をクリックします。

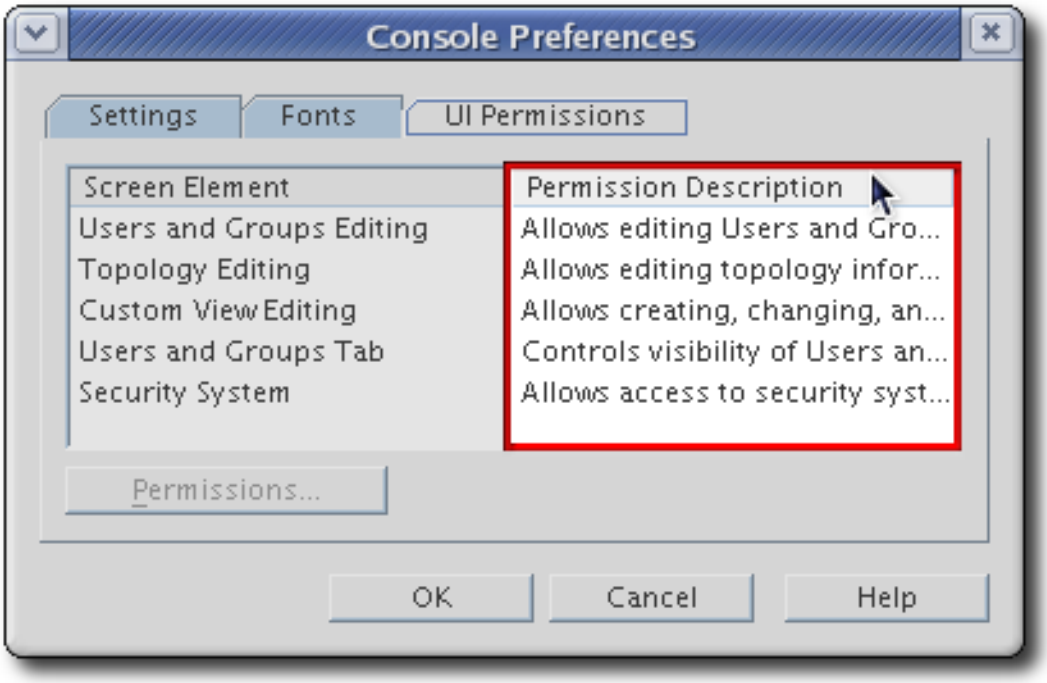


*フォントプロファイルを削除するには、**Font s** タブのドロップダウンメニューから選択されているようにし、**Remove** ボタンをクリックします。*

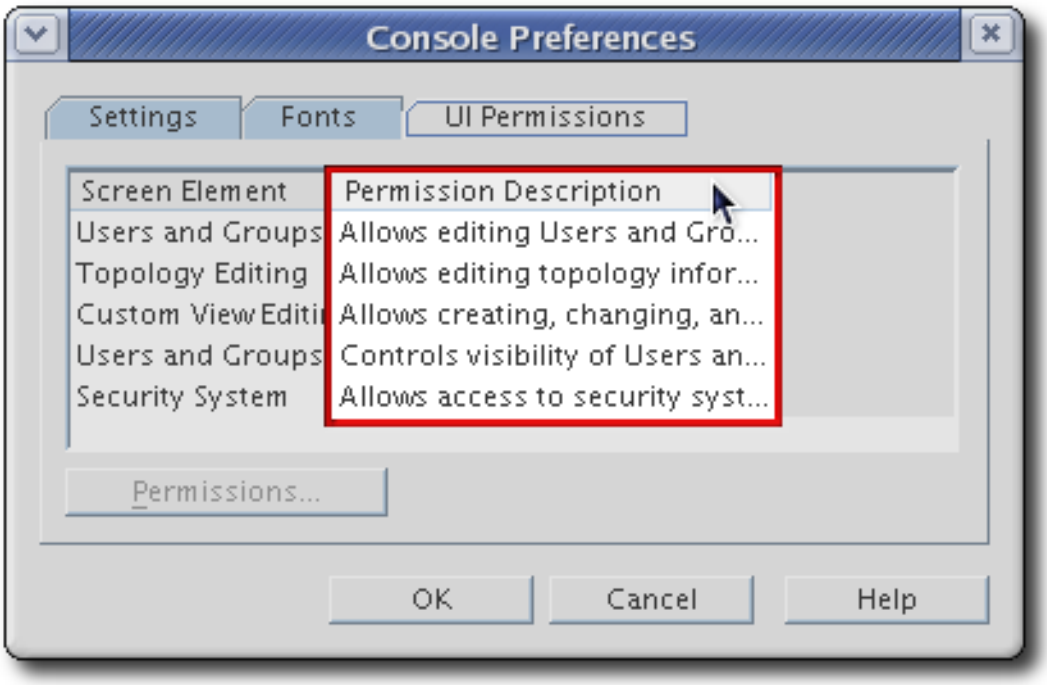
G.2.4. テーブル列の並べ替え

テーブルの列は、新しい位置にドラッグしることで再編成できます。

1. *テーブル見出しをクリックします。*

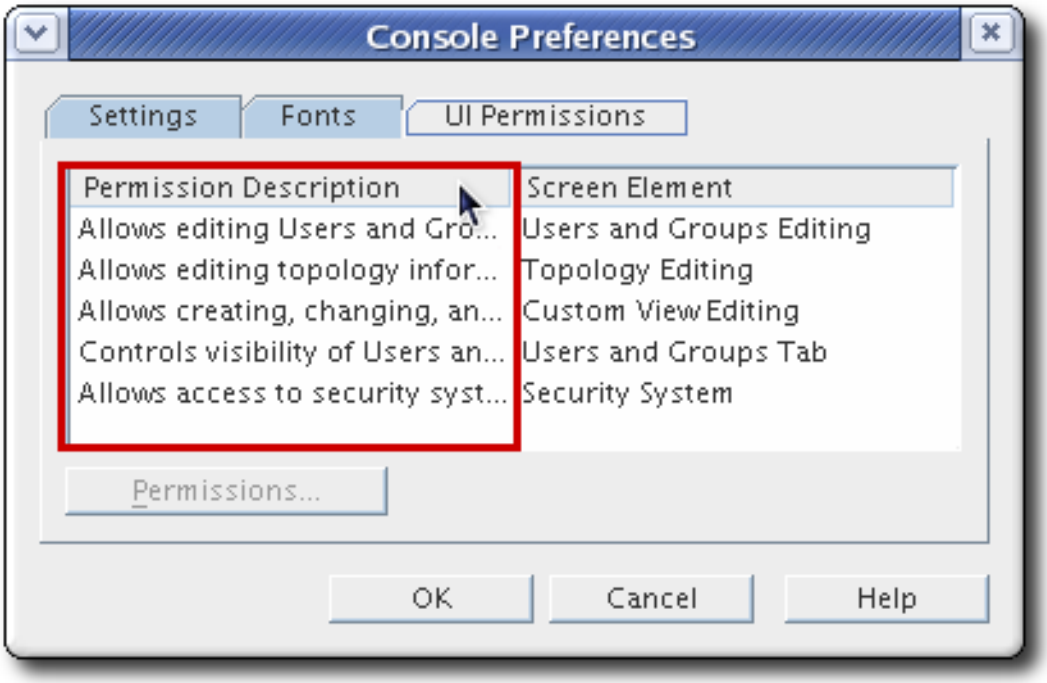


2. *左のマウスを押し続けると、その列を新しい場所にドラッグします。他のテーブル列は、自動的に新しい位置に移動します。*



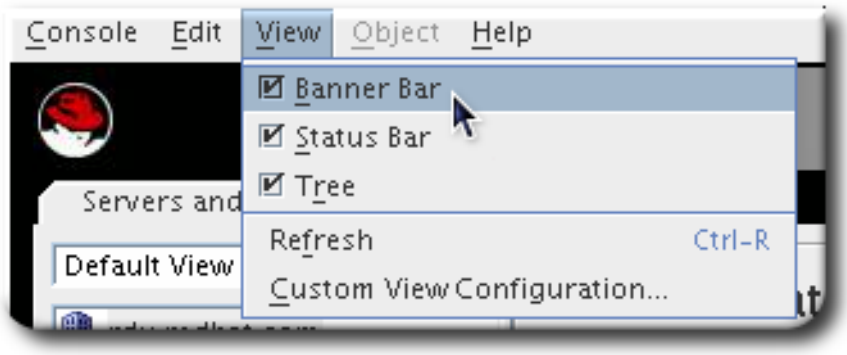
3.

マウスボタンをリリースすると、列 snap を新しい位置に送ります。

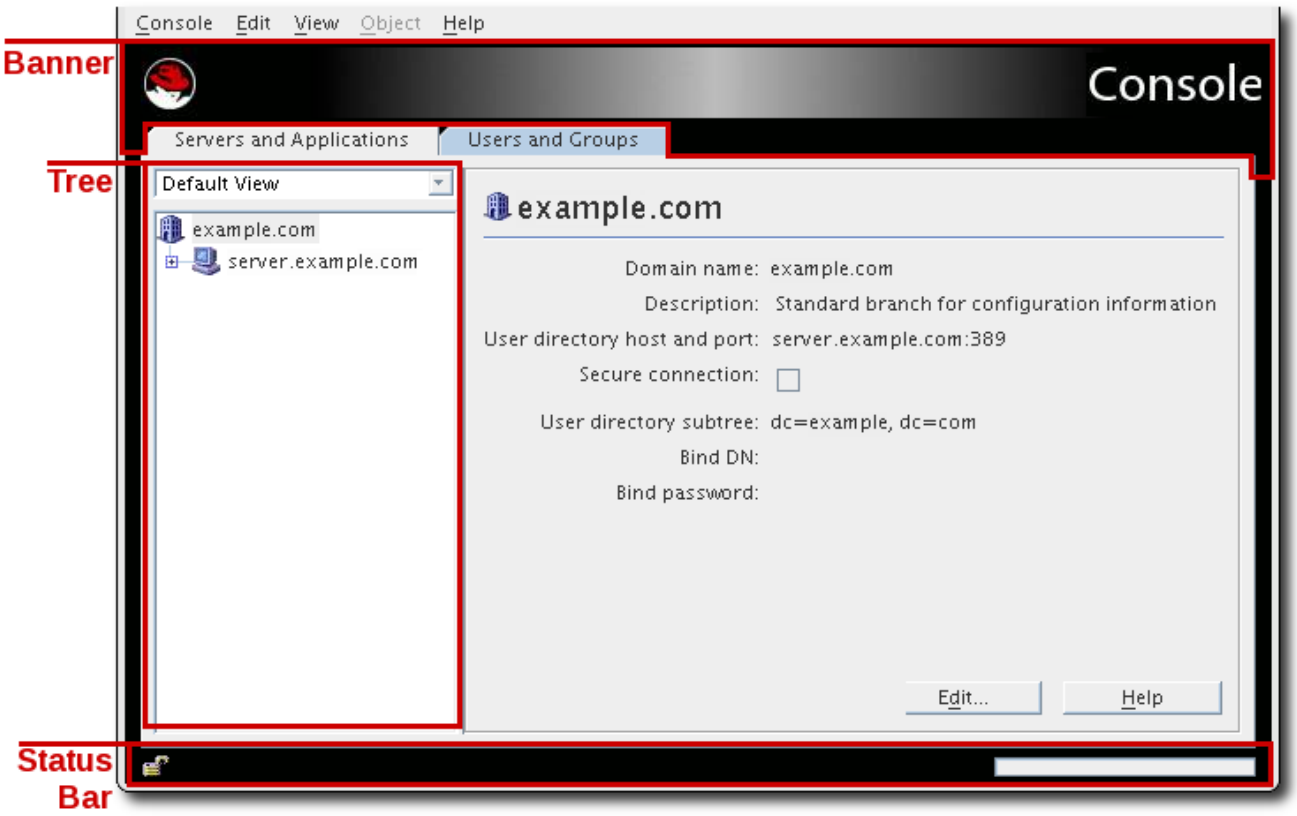


G.2.5. メインウィンドウのカスタマイズ

メインの Red Hat Management Console 画面のさまざまな要素を表示または非表示にすることができます。これは View メニューのチェックボックスによって設定されます。



コンソールには、ナビゲーションツリー（コンソールウィンドウの左側にある小規模なパネル）、コンソールウィンドウの上部にあるデコック背景およびバナー、コンソールの下部にあるステータスバーの 3 つの部分があります。



G.2.6. カスタムビューの使用

コンソールでは、異なるビューを作成して、Red Hat 管理コンソールウィンドウで異なるサーバーおよびドメインエントリーを表示できます。ビューは、定義されたサーバーエントリーセットのみを表示します。これにより、多数のインスタンスを維持したり、特定のタスクをすばやく実行できるようになります。

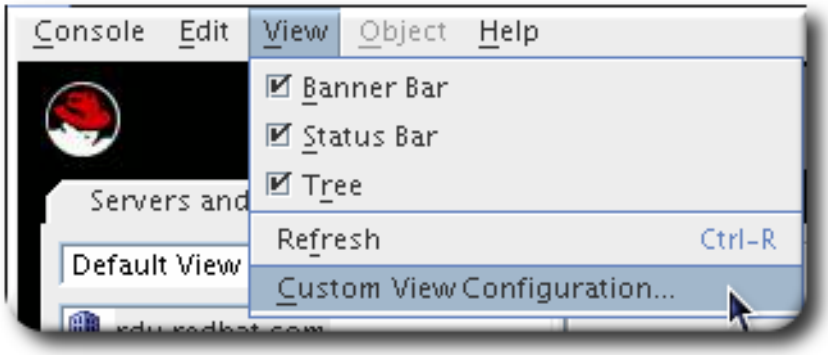
G.2.6.1. カスタムビューの作成

カスタムビューでは、定義した異なるサーバーインスタンスが表示されます。ビューはパブリック

またはプライベートです。プライベートビューはユーザー自身に表示されますが、プライベートビューは作成したユーザーのみに表示されます。

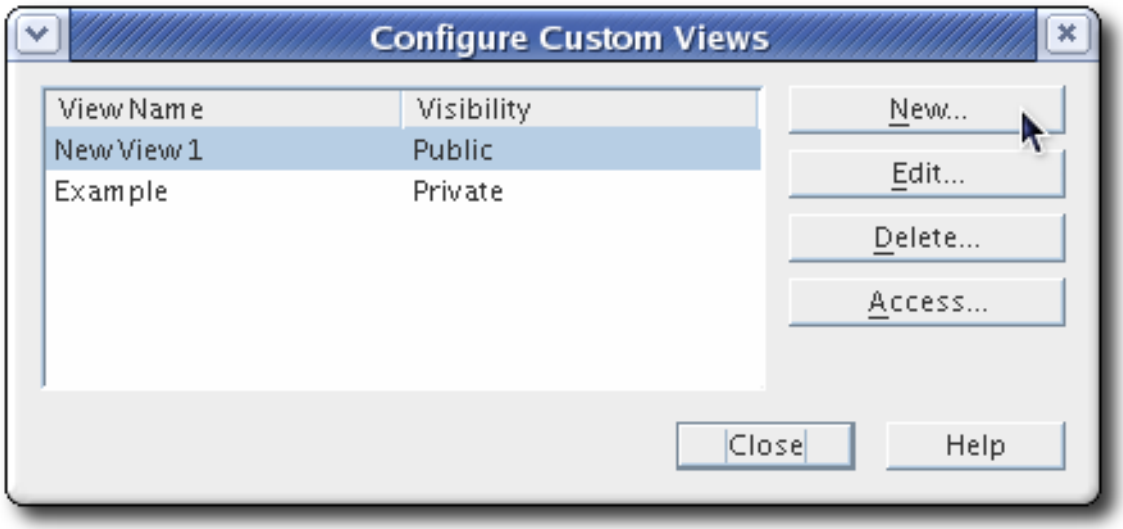
1.

View メニューで、Custom View Configuration を選択します。



2.

New をクリックします。



3.

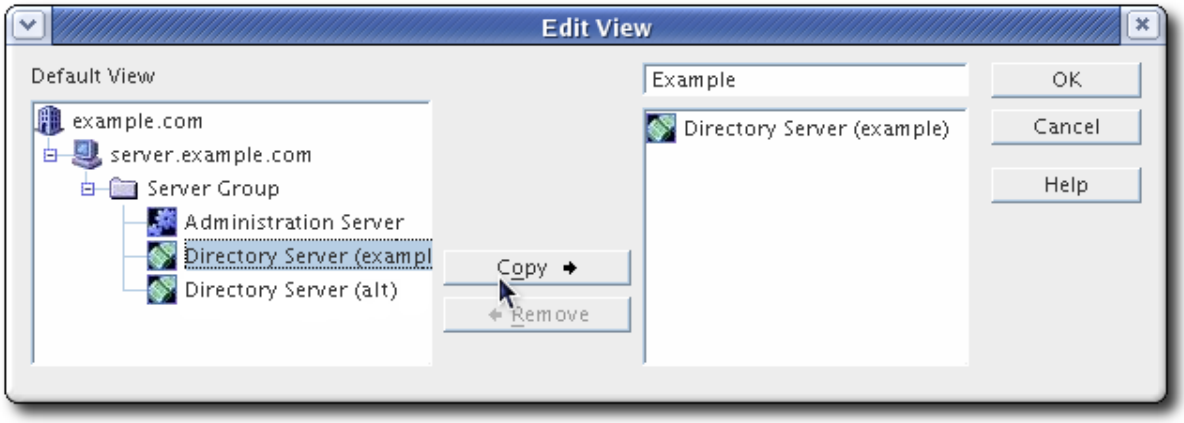
新規ビューをパブリックまたはプライベートにするかを選択して、OK をクリックします。

- デフォルトでは、パブリックビューはすべてのコンソールユーザーに表示されますが、アクセス制御命令(ACI)をアクセスを制限するように設定できます。詳細は、「パブリックビューのアクセスパーミッションの設定」 を参照してください。

- プライベートビューは、設定するユーザーにのみ表示され、ACI を設定してそのアクセスを変更することは設定できません。

4. Edit View ウィンドウで、このビューを説明する名前を入力します。

5. 左側の Default View ナビゲーションツリーからリソースを選択します。Copy をクリックして、右側のパネルに一覧表示し、ビューに追加します。



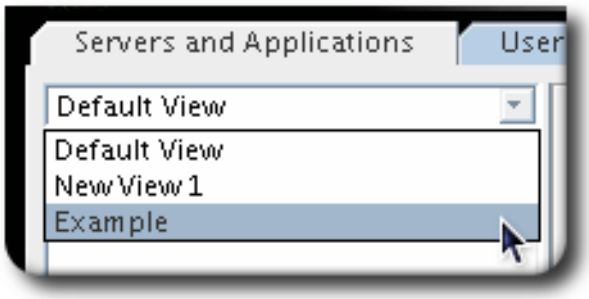
リソースの範囲を選択するには、/ SHIFT キーをクリックし、最初のエントリーと最後のエントリーを選択します。Ctrl キーを押してエントリーを選択して、複数のリソースを選択します。

カスタムビューを編集するには、一覧からこれを選択し、Edit ボタンをクリックして名前またはリソースに変更を加えます。

カスタムビューを削除するには、一覧から選択して、削除 ボタンをクリックします。

G.2.6.2. カスタムビューへの切り替え

Servers and Applications タブのドロップダウンリストから必要なカスタムビューを選択します。

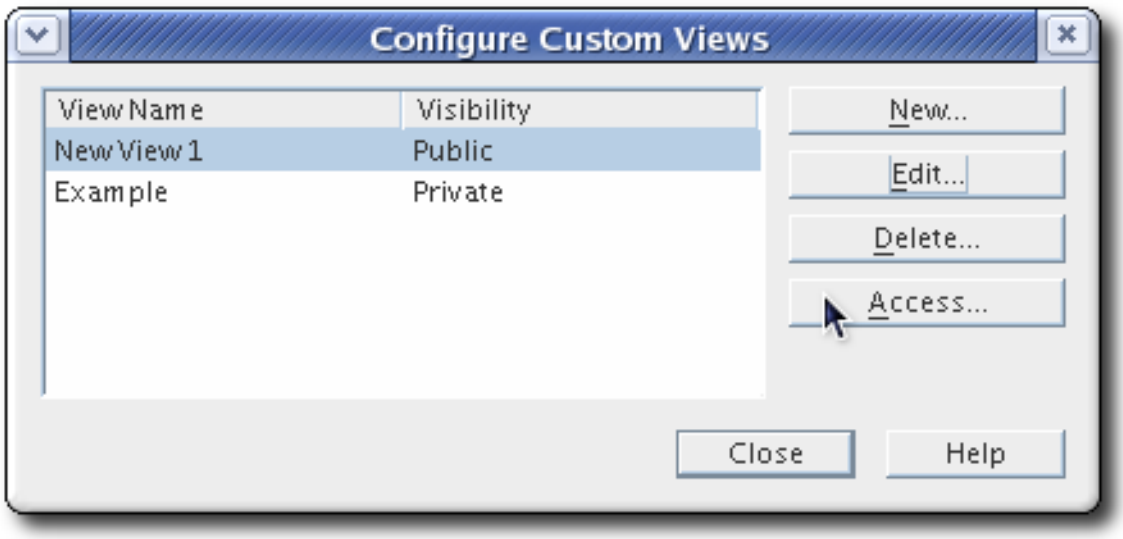


デフォルトビューに戻るには、ドロップダウンリストから Default View を選択します。

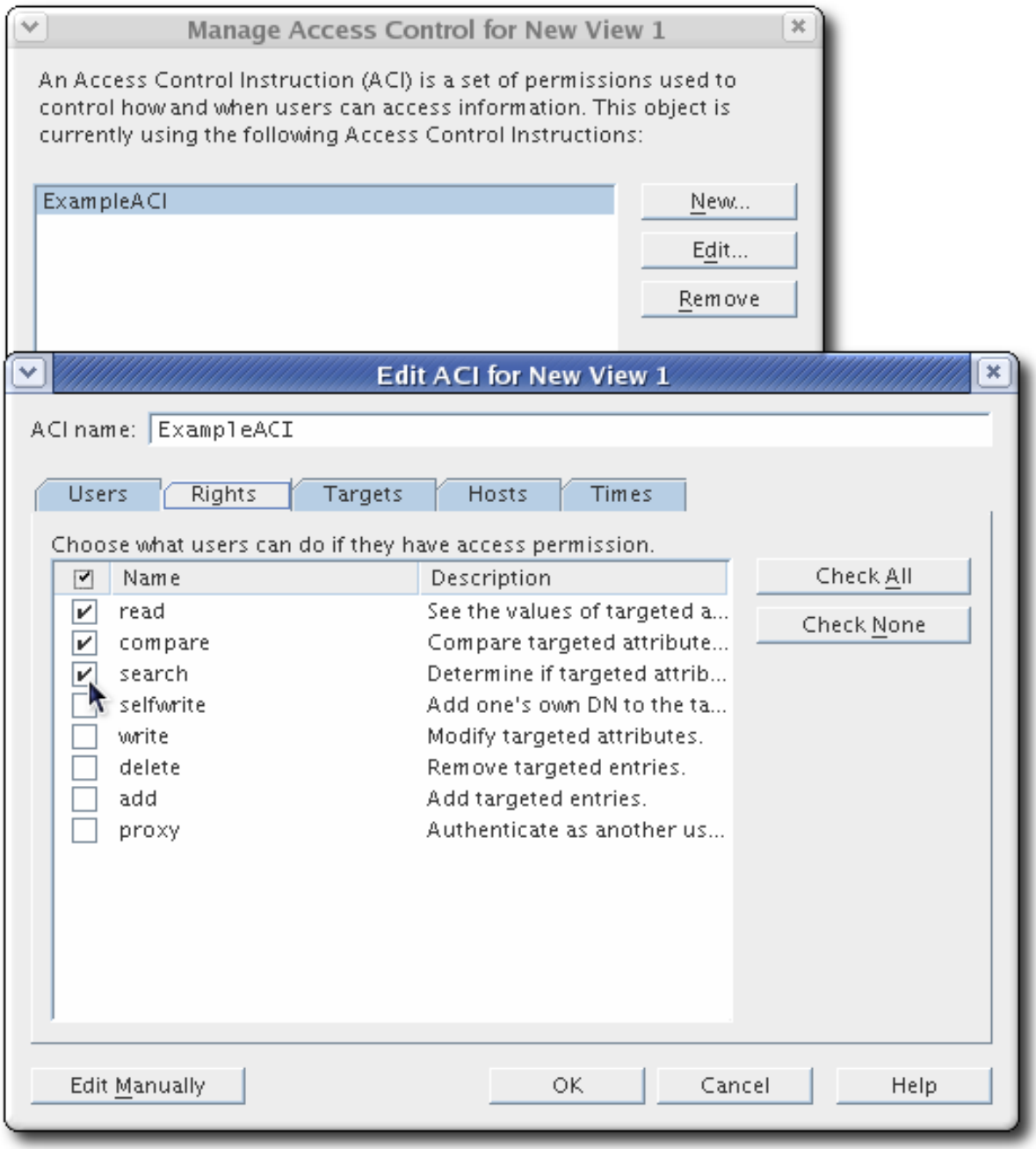
G.2.6.3. パブリックビューのアクセスパーミッションの設定

1. *View メニューから Custom View Configuration を選択します。*

2. *一覧からパブリック Custom View を選択し、Access をクリックします。*



3. *アクセス制御手順を設定します。*



4.

OK をクリックして ACI を保存します。

アクセス権の設定およびアクセス制御手順の詳細は、「アクセス制御の設定」 を参照してください。

G.3. サーバーインスタンスの管理

Red Hat 管理コンソールが管理するサーバーインスタンスは、階層に編成されます。上部の admin ドメインです。ドメイン内のホストは、異なるサーバーマシンを表します。各ホストにはサーバーグループがあり、同じ管理サーバーインスタンスを使用して Directory Server の相互関連グループを識別

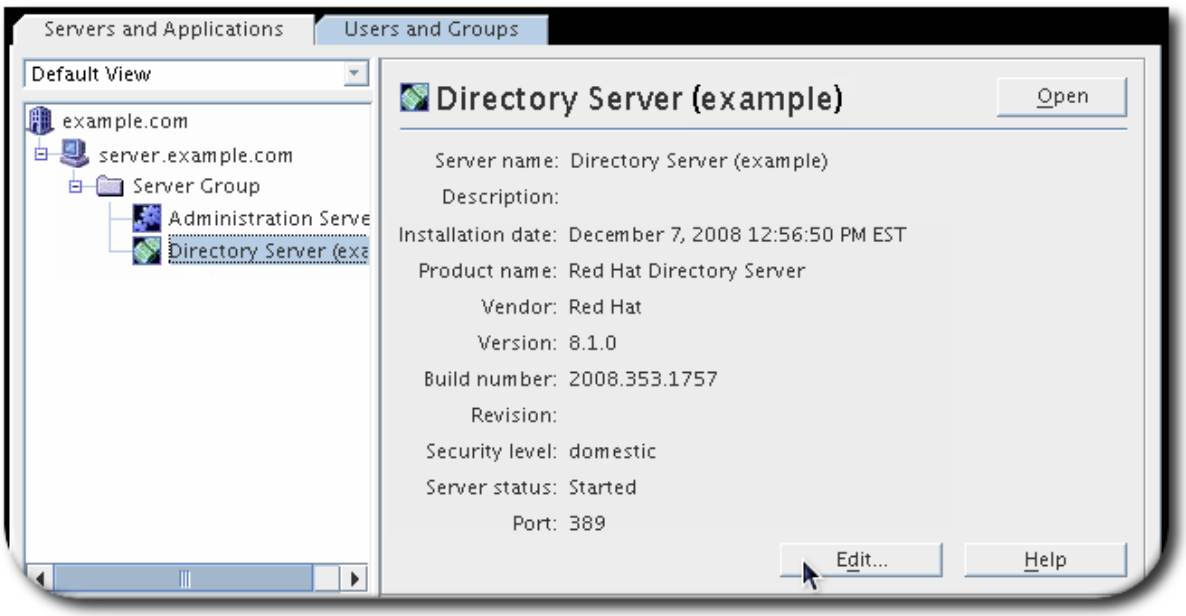
します。個別の Directory Server インスタンスと単一の管理サーバーインスタンスは、サーバーグループ内に所属します。サーバーグループごとに 1 つの管理サーバーインスタンスのみを使用できます。

このハイレベルエントリーは、Red Hat 管理コンソール で作成および管理できます。

G.3.1. ドメイン、ホスト、サーバーグループ、およびインスタンス情報の編集

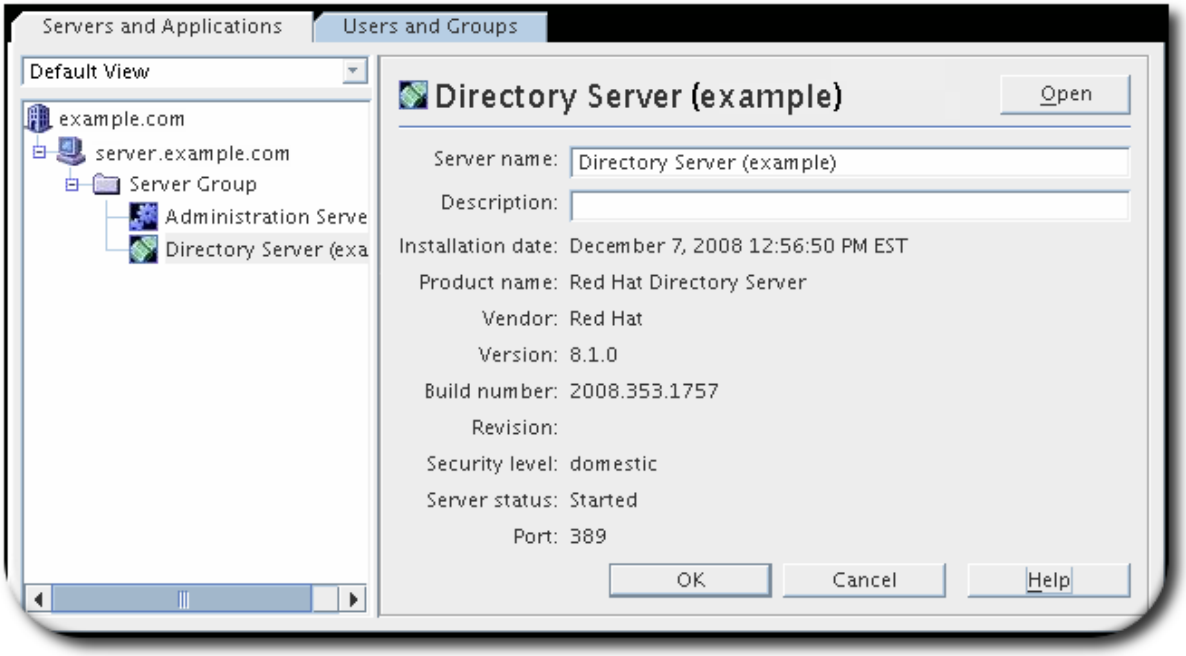
Red Hat コンソールは、すべての管理ドメイン、ホスト、グループ、およびサーバーインスタンスに関する情報を表示します。この情報のほとんどは、インストール日やビルド番号など、編集できませんが、その情報もいくつかあります。

1. *Servers and Applications タブで、変更するエントリーを選択します。*



2. *編集 をクリックします。*

3. *インスタンスの情報を編集します。すべてのエントリーには、名前と説明を変更するオプションがあります。インスタンスがインストールされている物理マシンであるホストには、場所を変更するオプションもあります。*



4.
OK をクリックします。

G.3.2. 管理ドメインの作成と削除

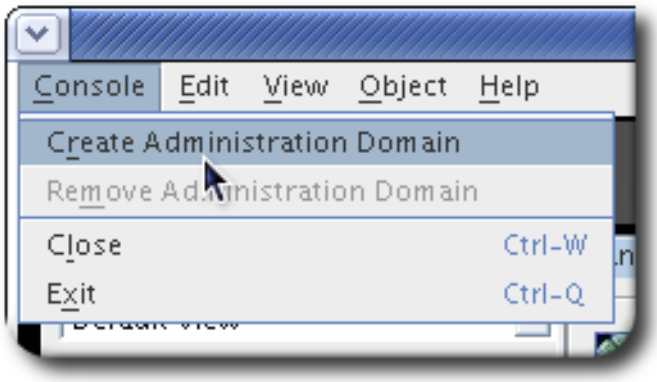
管理ドメインは、サーバーグループのコンテナーエントリーです（各サーバーグループには、同じ Configuration Directory Server と、サーバーグループでも機能するよう設定された Directory Server インスタンスが含まれます）。

G.3.2.1. 管理対象ドメインの作成および編集

新しい admin ドメインを作成するには、以下を実行します。

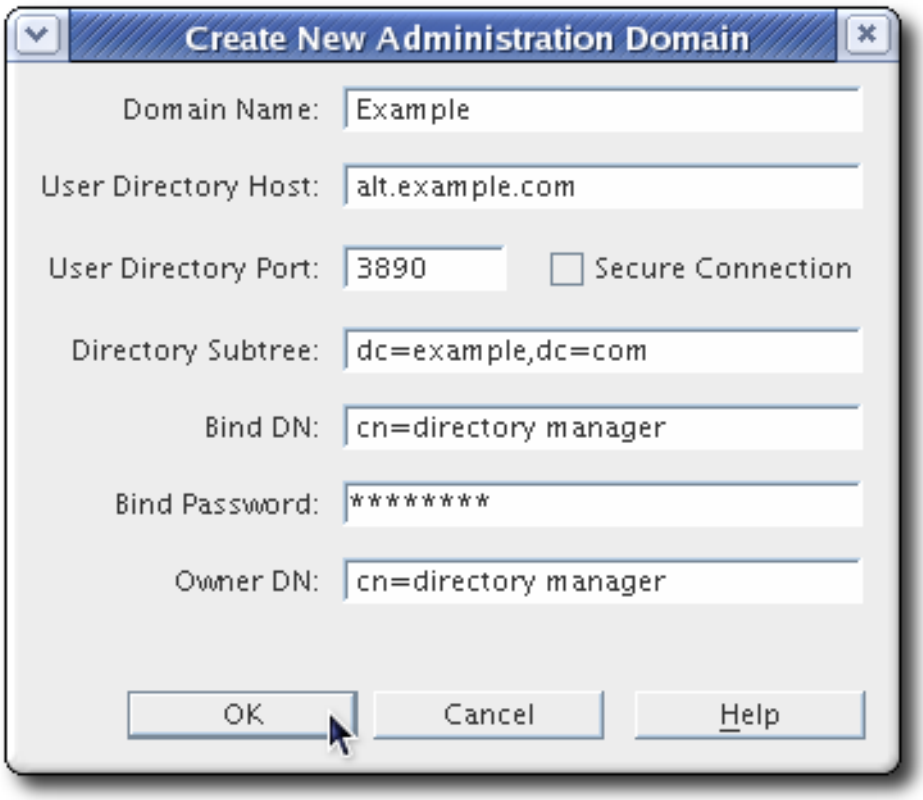
1.
トップメニューで、Console メニューアイテムをクリックし ます。

2.
Create New Administration Domain を選択します。



3.

新しい Directory Server インスタンスの情報など、管理ドメインの情報を入力します。



4.

OK をクリックします。

管理ドメインを編集するには、サーバーウィンドウでエントリーを選択し、Edit ボタンをクリックします。

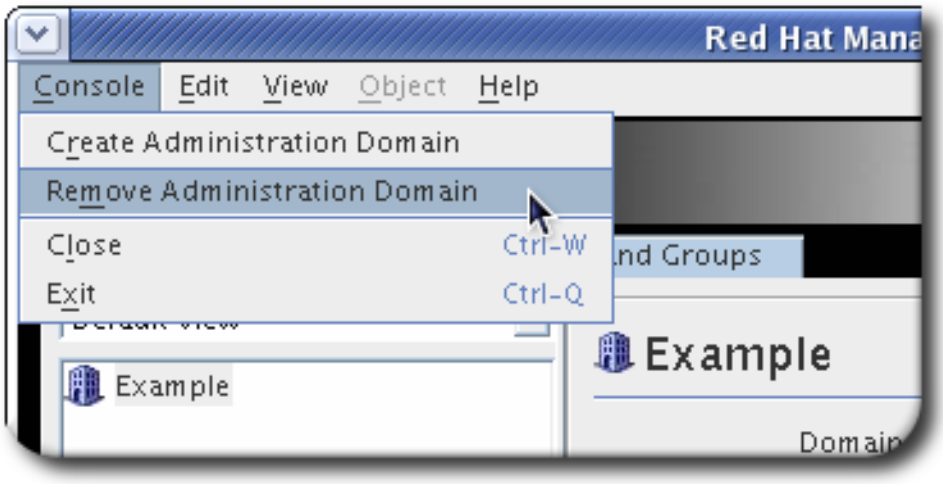
警告

管理ドメイン設定は、ドメイン内のすべてのサーバーに影響します。admin ドメイン設定に変更を加えると、ドメインのすべてのサーバーを再起動する必要があります。

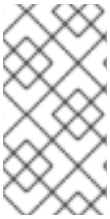
G.3.2.2. 管理対象ドメインの削除

管理ドメインを削除するには、以下を実行します。

1. *ナビゲーションツリーで削除する管理ドメインを強調表示します。*
2. *トップメニューで、Console メニューアイテムをクリックし ます。*
3. *管理ドメイン の削除 を選択します。*



4. *Yes をクリックします。*



注記

ドメインを削除する前に、ドメイン内のサーバーグループおよびサーバーを削除する必要があります。

G.4. DIRECTORY SERVER のユーザーおよびグループの管理

複数の Red Hat Directory Server インスタンスと管理サーバーの両方のユーザーの場合、Red Hat 管理コンソールで作成、編集、検索が可能です。メインのコンソールウィンドウは、組織単位およびグループを作成し、新しい ous および groups にエントリーを追加するためにも使用できます。

「アクセス制御の設定」 では、アクセス権限やその他のセキュリティー情報を設定する際に、ユーザーおよびグループの情報を使用する方法を説明します。

G.4.1. ユーザーおよびグループの検索

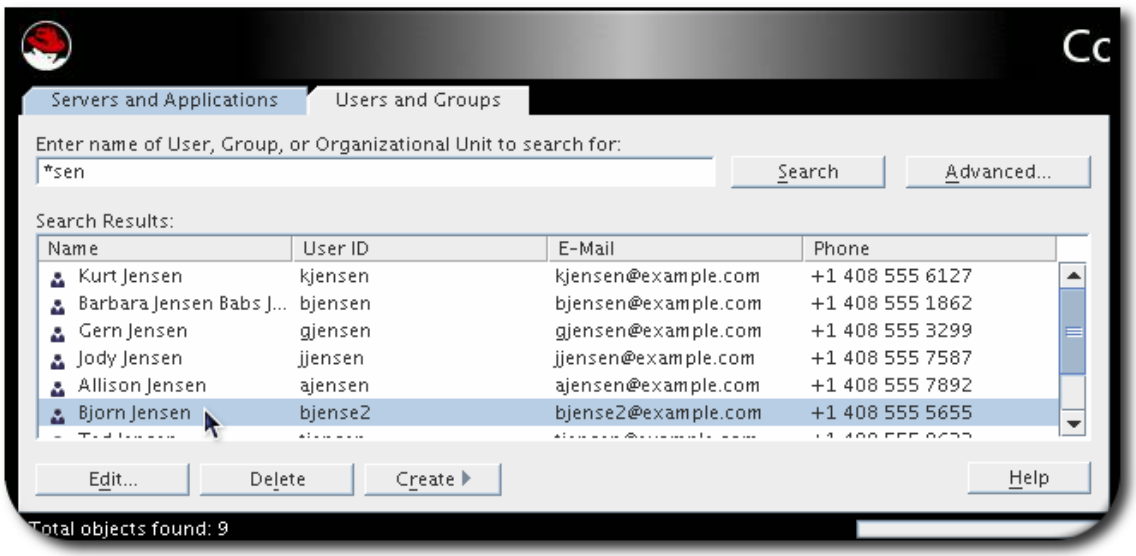
ユーザーおよびグループはディレクトリーエントリーを検索します。デフォルトでは、Administration Server 用に設定されたデフォルトのユーザーディレクトリーを検索しますが、任意の Red Hat Directory Server インスタンスにディレクトリーを変更できます。

ディレクトリーを検索するには、以下を実行します。

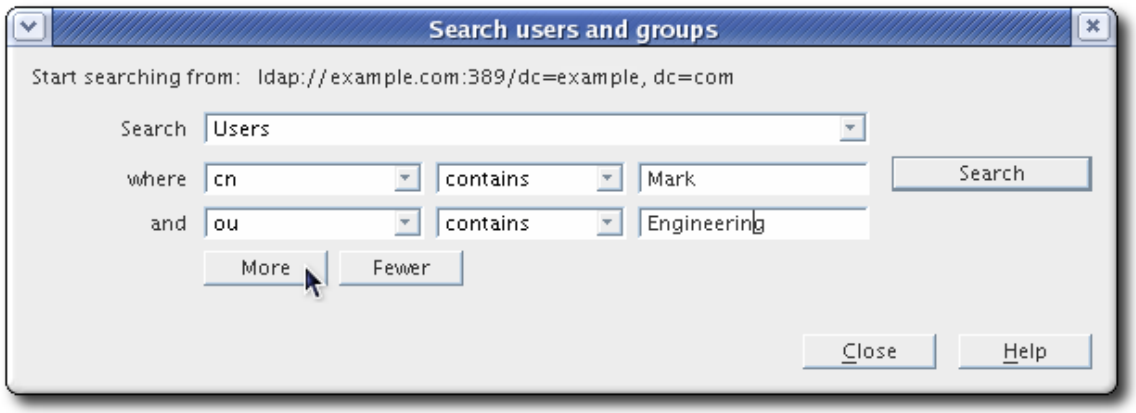
1. Users and Groups タブをクリックします。

2. 検索条件を入力し、検索 をクリックします。

- 簡単な検索には、テキストボックスにエントリー名 all または part を入力します。すべてのエントリーを返すには、検索フィールドを空白のままにするか、アスタリスク(*)を入力します。



- より複雑な検索またはフォーカス検索には、**Advanced** ボタンをクリックして検索する属性（cn、givenname、または ou）、検索の種類、検索用語など）を入力します。検索条件を追加または削除するには、**More and Fewer** ボタンをクリックします。

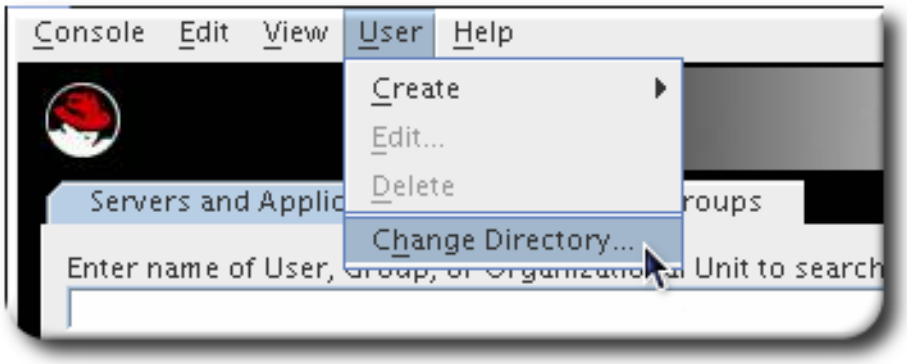


3. **Search** をクリックします。結果はリストボックスに表示されます。

検索ディレクトリーを変更するには、以下を実行します。

1. **Users and Groups** タブをクリックします。

2. トップメニューで **User** メニューアイテムを選択し、**Change Directory** を選択します。



3. ユーザーディレクトリーの情報を入力します。

- *ユーザーディレクトリーホスト。Directory Server インスタンスの完全修飾ホスト名。*

- *ユーザーディレクトリー ポートおよび セキュアな接続接続のポート番号と、これが TLS(LDAPS)であるかどうか。*

- *ユーザーディレクトリーサブツリー。ディレクトリーを検索するサブツリーの DN です。たとえば、サブツリーの場合は dc=example,dc=com （ベース DN または ou=Marketing, dc=example,dc=com ）になります。*

- *バインド DN およびバインドパスワード。ディレクトリーへの認証に使用する認証情報。*

4. *OK をクリックします。*

G.4.2. ディレクトリーエントリーの作成

Red Hat 管理コンソールは、Users and Groups タブのユーザー、グループ、および組織単位を追加、編集、および削除できます。エントリー作成のさまざまな種類のエントリーおよびオプションは、『『Red Hat Directory Server 管理ガイド』』 で詳しく説明しています。

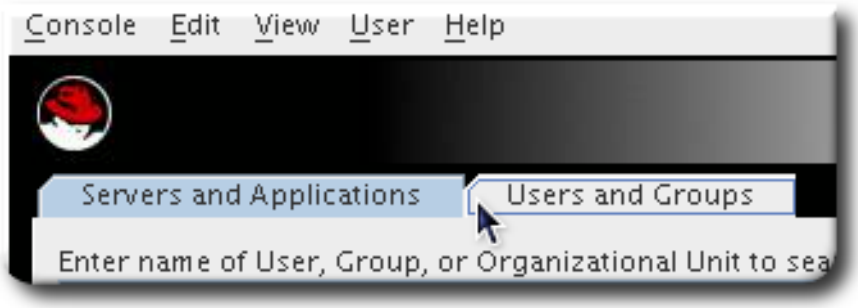
G.4.2.1. ディレクトリーおよび管理ユーザー

注記

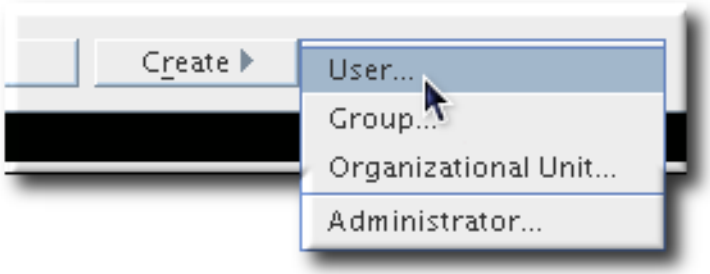
ユーザーは、コンソールを介して Directory Server ユーザーデータベースに追加できます。また、ユーザーを Administration Server 管理者として追加できます。このプロセスはほぼ同一ですが、2 つの例外があります。

- *Directory Server ユーザーは、Create ボタンをクリックしてから Users オプションをクリックすると追加されます。一方、管理者は Administrator オプションを選択すると作成されます。*
- *管理者は ou=Groups,ou=Topology,o=NetscapeRoot に自動的に追加されます。*

1. *Users and Groups タブをクリックします。*

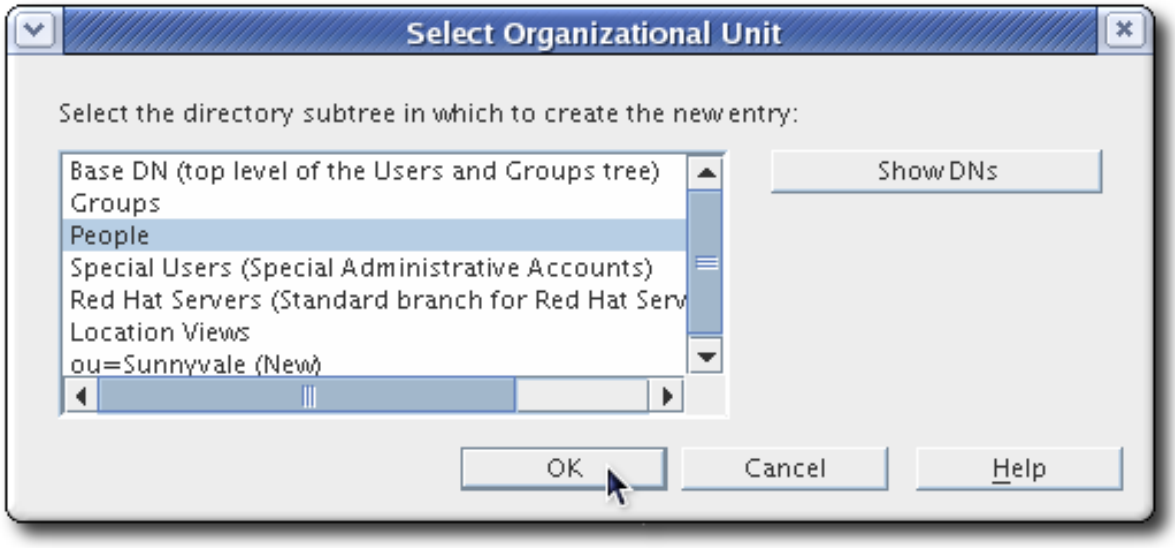


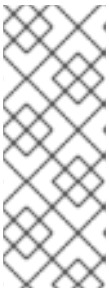
2. *Create ボタンをクリックし、User を選択します。*



または、トップメニューで User オプションを開き、Create > User を選択します。

3. *エントリーが作成されるディレクトリーツリーにあるものを選択します。*





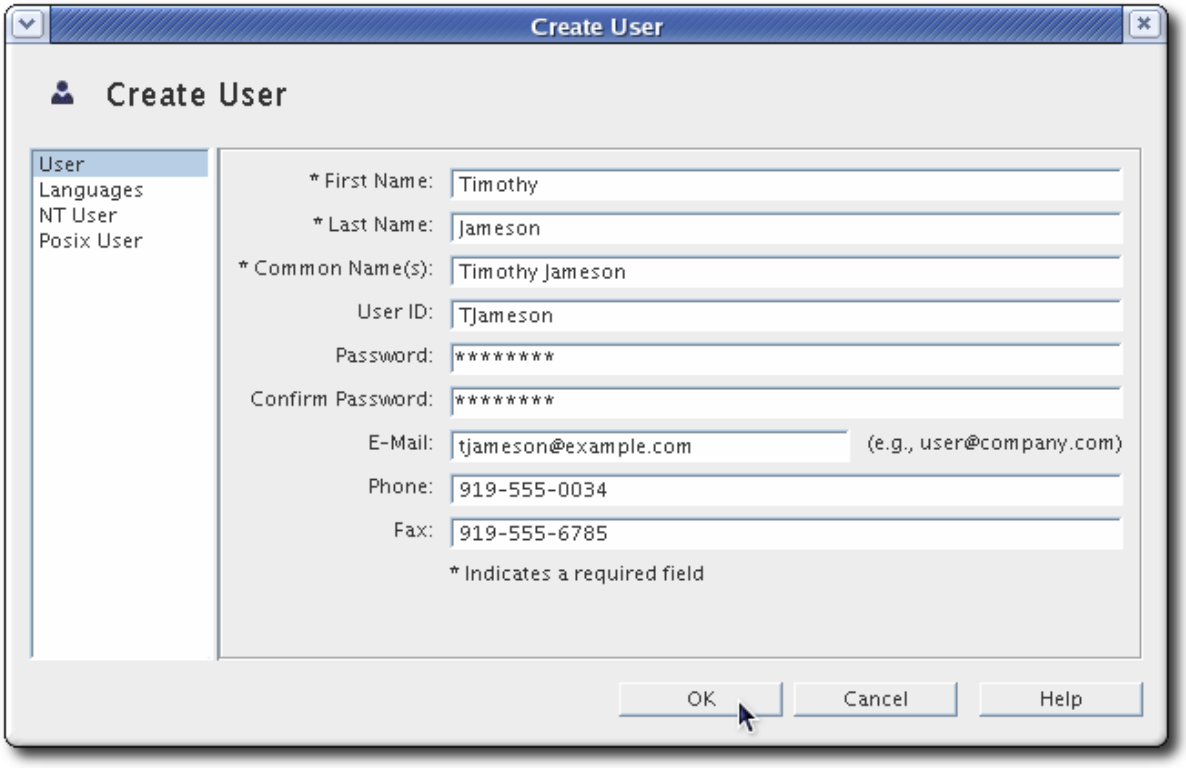
注記

管理者の作成時に、通常の Directory Server ユーザーとともにユーザーを追加する ou を選択するオプションはありません。これは、管理者が admin ユーザーで ou=Groups,ou=Topology,o=NetscapeRoot に追加されるためです。

ビューがディレクトリーに追加されている場合は、エントリーを ou またはビューに追加できます。

4.

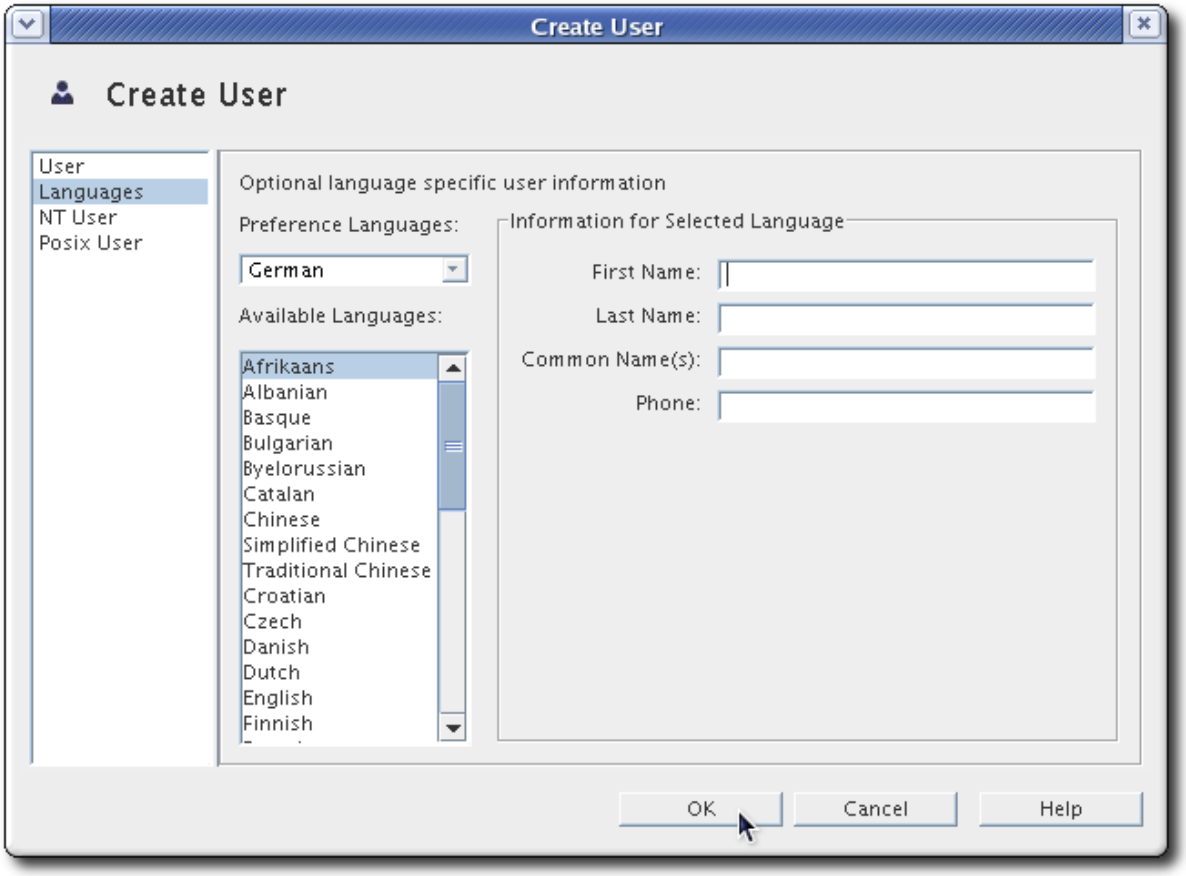
Create User ウィンドウで、ユーザー情報を入力します。Common Name フィールドおよび User ID フィールドには、First Name フィールドと Last Name フィールドに結合された値が自動的に入力されます。これらの最初、姓、および共通名フィールドは必須です。Directory Server とコンソールにログインできるパスワードも必要ですが、必須の属性ではありません。



5.

*必要に応じて、左側の **Languages** リンクをクリックし、代替言語を選択して、共通の属性に対して国際化された値を入力します。*

*このオプションを使用すると、国際ユーザーは英語以外の言語を選択でき、優先言語でその名前を表すことができます。**pronunciation** 属性を使用すると、国際名属性に対する電話番号検索が可能になります。*



6.

OK をクリックします。

G.4.2.2. グループ

グループは、共通の属性を共有したり、リストの一部であるユーザーで構成されます。Red Hat Directory Server は、静的、動的、および証明書の 3 種類のグループをサポートします。各グループは、ユーザーまたは メンバーを追加する 方法によって異なります。

- *静的グループには、手動で追加するメンバーがあるため、管理者がユーザーを手動で追加または削除しない限り、メンバーは変更しないため、静的 になります。*

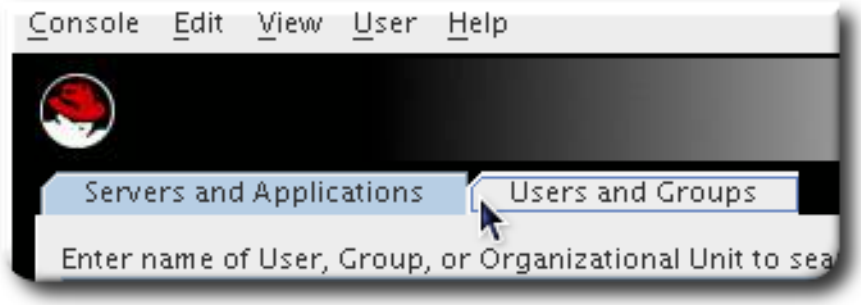
- *動的グループには、エントリーの 1 つ以上の属性に基づいてユーザーが自動的に含まれます。属性と値は LDAP URL を使用して決定されます。たとえば、動的グループは、属性や値 st=California および department= sales を含むエントリーを検索する LDAP フィルターを使用できます。この 2 つの属性を持つディレクトリーにエントリーが追加されると、ユーザーは動的グループにメンバーとして自動的に追加されます。これらの属性がエントリーから削除されると、エントリーはグループから削除されます。*

- *証明書グループには、証明書のサブジェクト名に特定の属性と値のペアがあるすべてのユーザーが含まれます。たとえば、証明書グループは、サブジェクト名に st=California,ou=Sales,ou=West の文字列に基づいています。ユーザーが証明書のそれらの属*

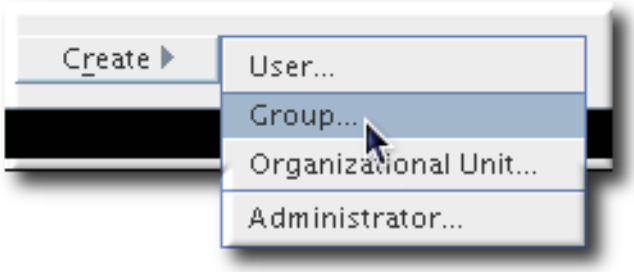
性を持つ証明書を使用してサーバーにログインすると、そのユーザーはグループに自動的に追加され、そのグループのすべてのアクセス権限が付与されます。

グループを作成するには、以下を実行します。

1. **Users and Groups** タブをクリックします。

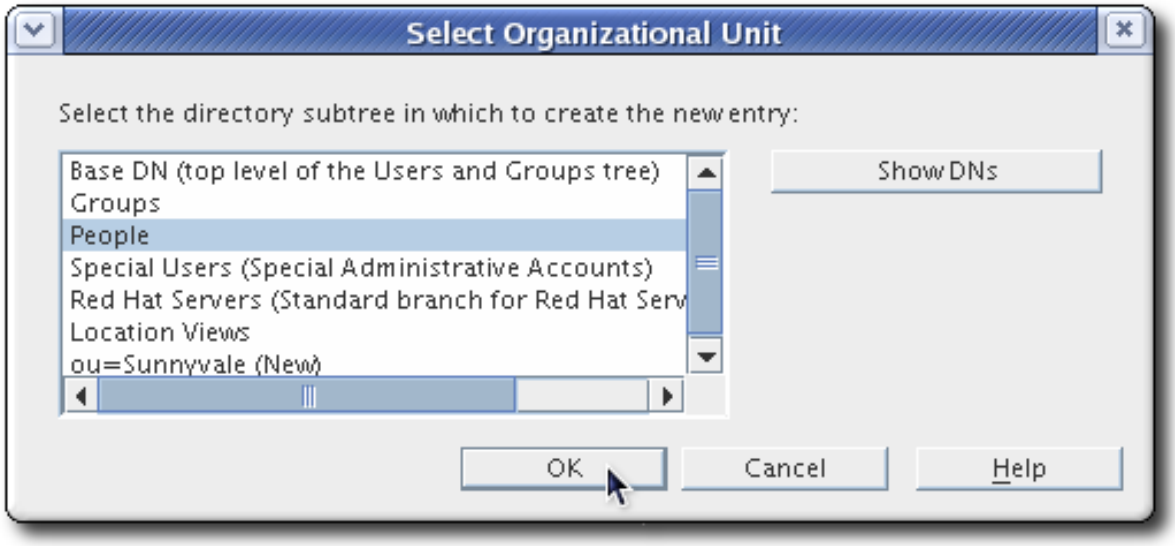


2. **Create** ボタンをクリックし、**Group** を選択します。



または、トップメニューで **User** オプションを開き、**Create > Group** を選択します。

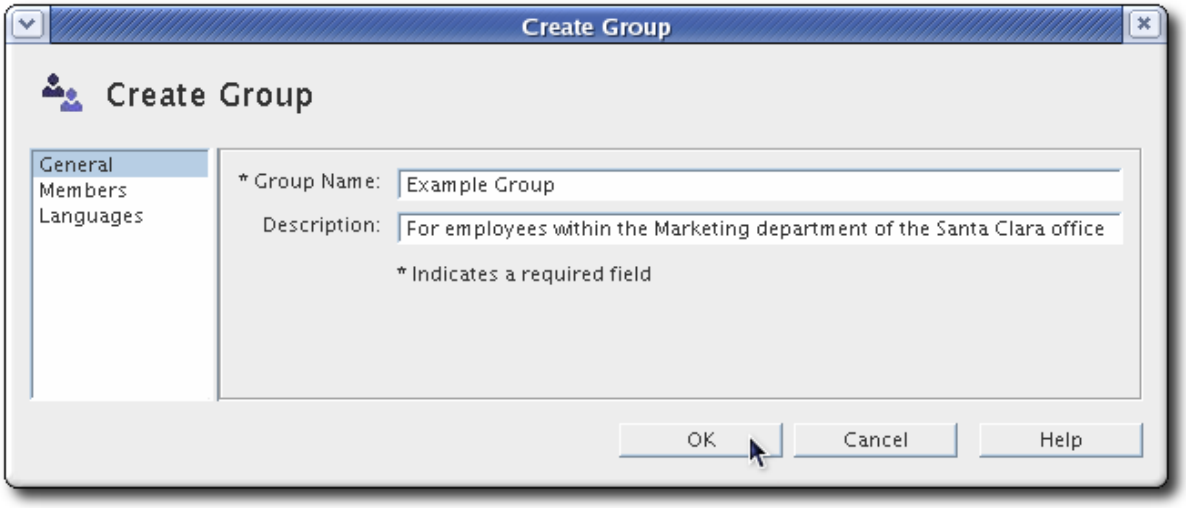
3. エントリーが作成されるディレクトリーツリーにあるものを選択します。



サブツリーエントリーは、ディレクトリーに追加された場合に ou またはビューになります。

4.

グループの名前と説明を入力します。



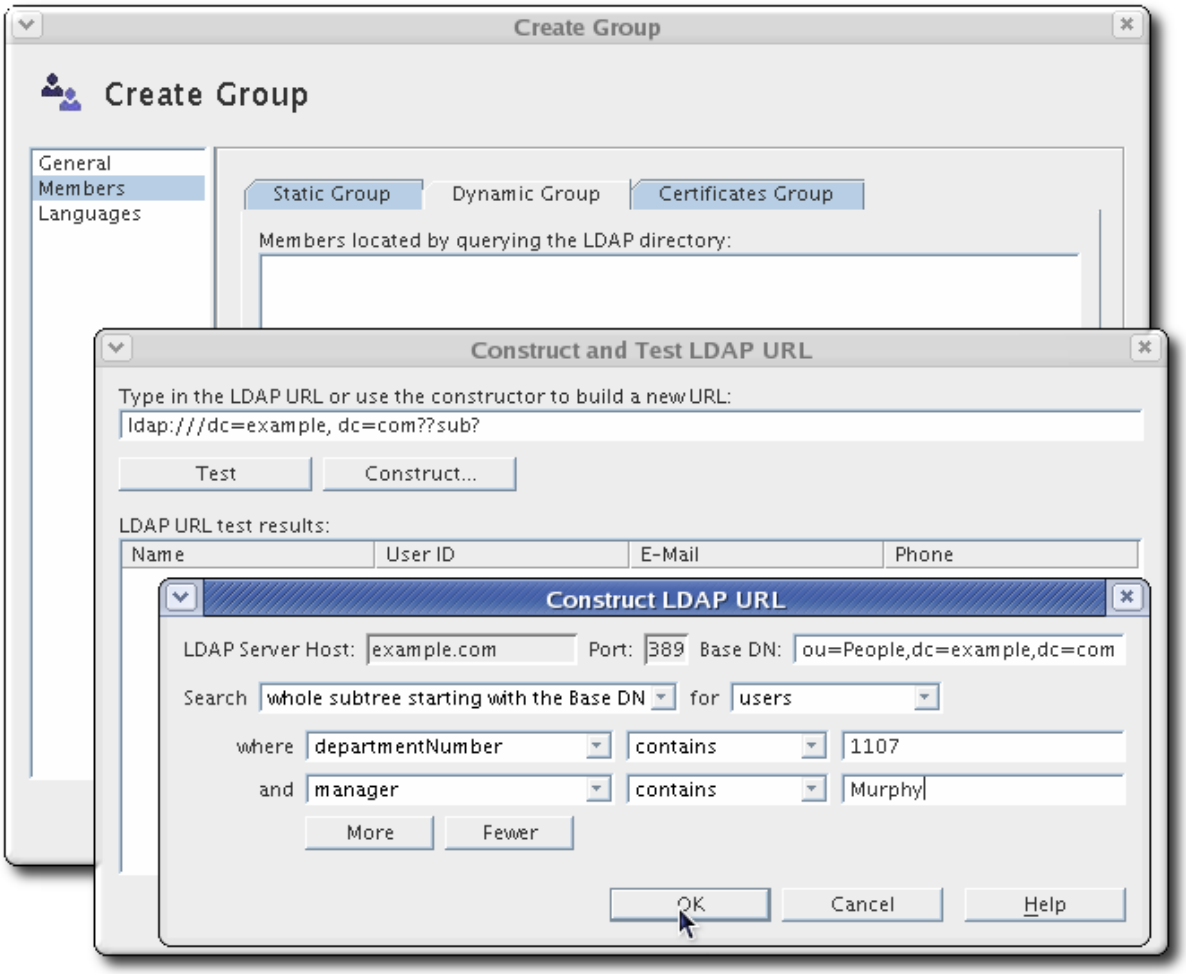
この時点で、メンバーを追加せずに、新しいグループエントリーを保存できます。OK をクリックします。

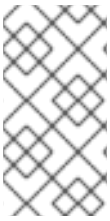
5.

Members リンクをクリックしてグループにメンバーを追加し、グループメンバーシップ、静的、動的、または証明書のタイプのタブをクリックします。

6.

メンバーを設定します。静的グループの場合は、ユーザーを手動で検索して追加します。動的グループの場合は、エントリーの検索に使用する LDAP URL を作成し、証明書グループの場合は、ユーザー証明書サブジェクト名で検索する値を入力します。





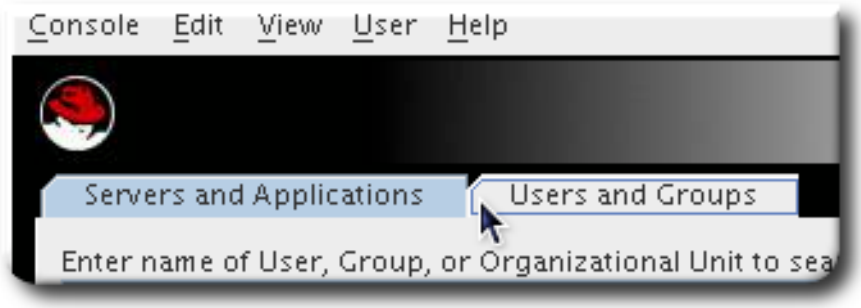
注記

さまざまな種類のグループおよび設定方法は、『『Red Hat Directory Server 管理ガイド』』で詳しく説明しています。

G.4.2.3. 組織単位

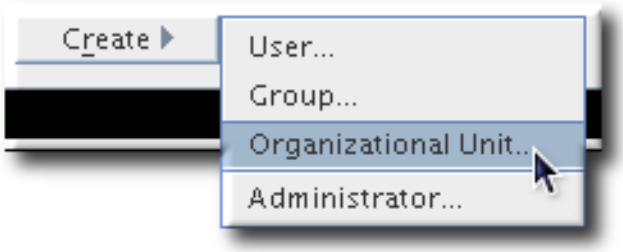
組織単位には、多数のグループとユーザーを含めることができます。組織ユニットは、通常、組織内の個別の論理部門（部門や地理的な場所など）を表します。各 organizationalUnitName (ou)はディレクトリーツリーの新しいサブツリーブランチです。これは、サブエントリーの識別名の一部である ou=People,dc=example,dc=com など、ou の相対識別名に反映されます。

1. *Users and Groups タブをクリックします。*



2.

Create ボタンをクリックし、Organizational Unit を選択します。



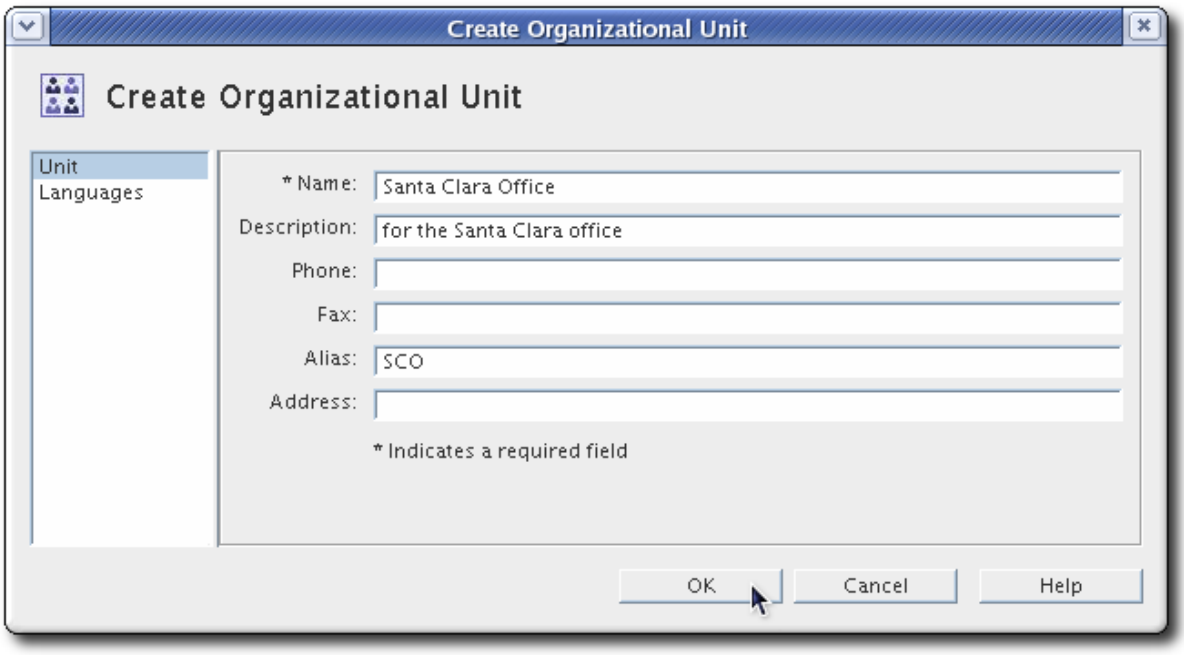
または、トップメニューで User オプションを開き、Create > Organizational Unit を選択します。

3.

新しい組織単位を検索するディレクトリーサブツリーを選択します。

4.

組織単位情報を記入します。Alias は、フルネームの代わりに使用できる組織単位の代替名を指定します。



5.
OK をクリックします。

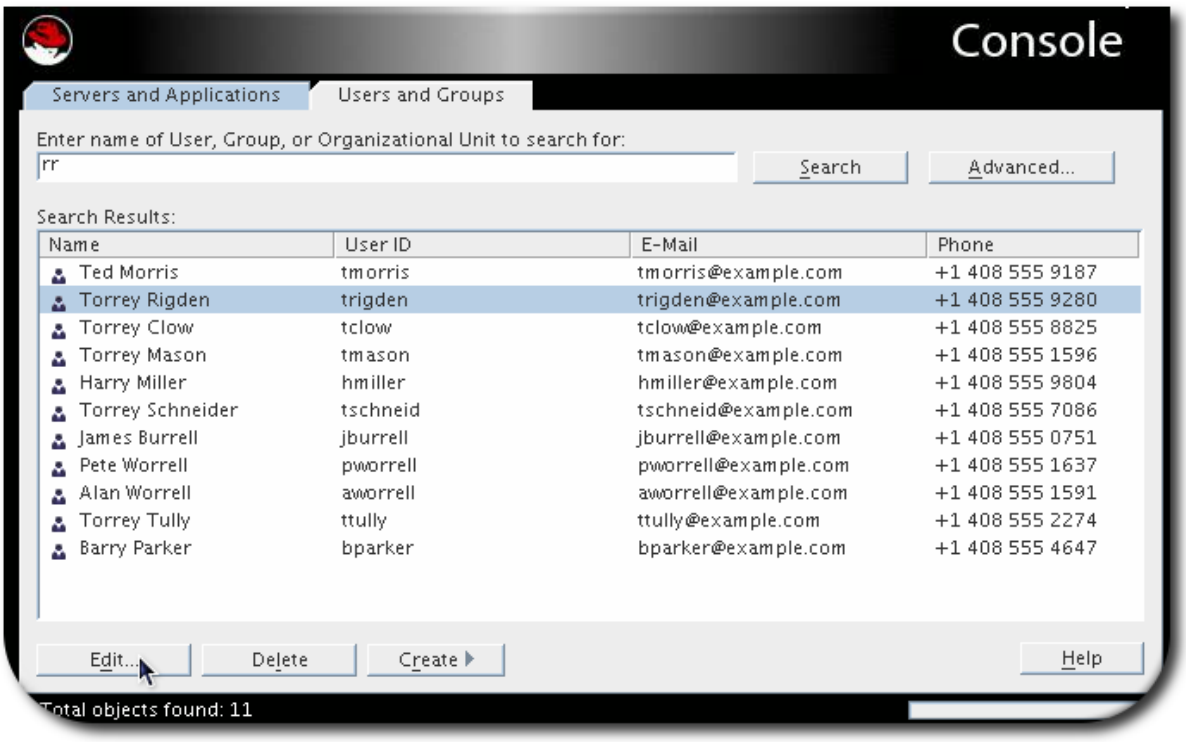
G.4.3. ディレクトリーエントリーの変更

G.4.3.1. エントリーの編集

1.
編集するエントリーを検索します。

エントリーの検索に関する詳細は、「ユーザーおよびグループの検索」 を参照してください。

2.
エントリーを選択し、Edit をクリックします。



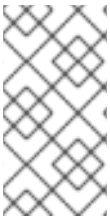
3.
エントリー情報を編集し、OK をクリックして変更を保存します。

G.4.3.2. エントリーの同期属性の許可

Red Hat Directory Server および Active Directory は、Unix および Windows 固有のディレクトリー属性を統合し、Active Directory に Directory Server エントリーを使用するには、エントリーに ntUser 属性が必要です。（それ以外の場合、Windows エントリーには posixAccount 属性が必要です。）

Windows(NT)属性はエントリーで有効にする必要があります。デフォルトでは、これらの属性は個別のエントリーに手動で追加されます。ユーザー編集ウィンドウには、NT ユーザー用の左側のリンクがあり、Directory Server エントリーに同期用に Windows 固有の属性を含めることができます。

すべての新規エントリーが ntUser オブジェクトクラスを自動的に所有するようにサーバーを設定することもできます。これは、『Red Hat Directory Server 管理ガイドのDirectory Server-Active Directory』 同期の章で説明されています。



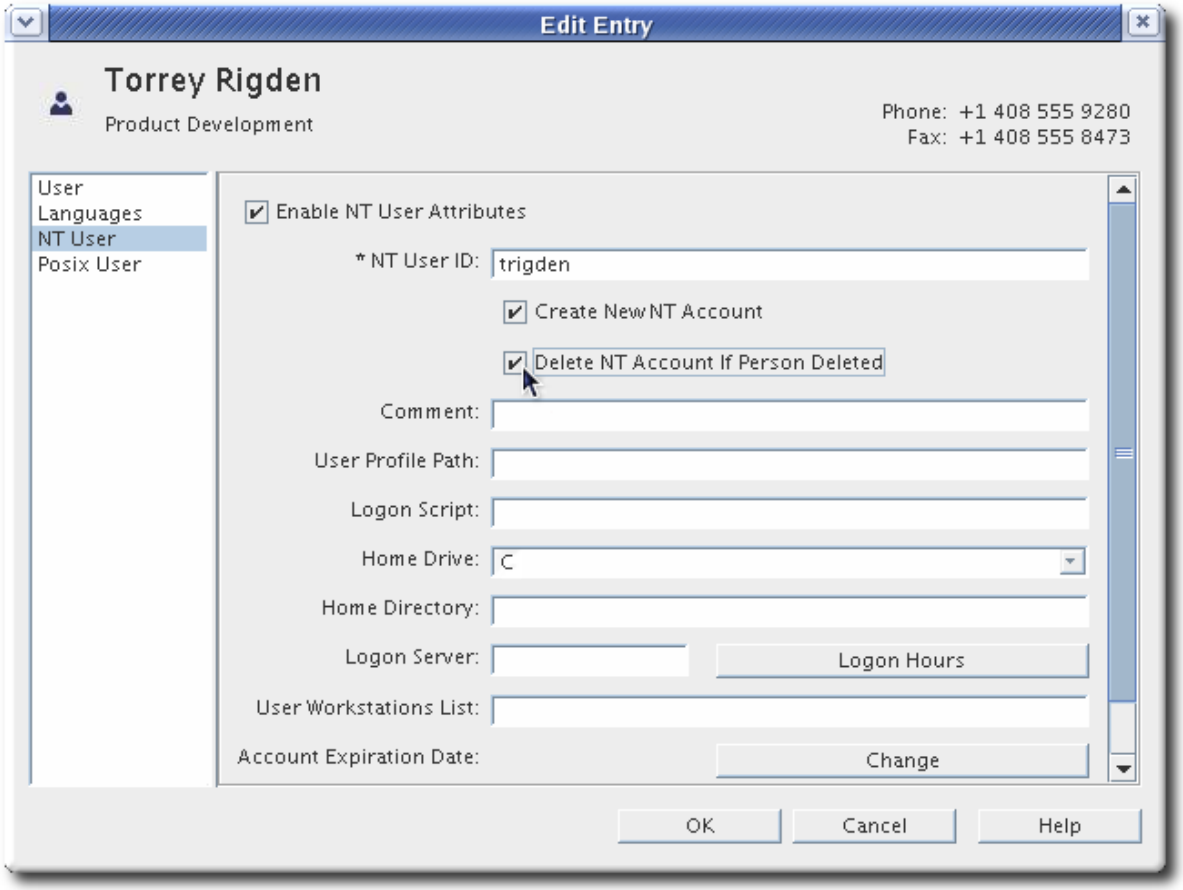
注記

Red Hat Directory Server エントリーには、ntUser オブジェクトクラスと、Active Directory に同期するために必要な属性を追加する必要があります。

同期を有効にするには、以下を実行します。

1. *ユーザーを選択または作成し、NT User リンクをクリックします。*

2. *NT アカウントを有効にし、エントリーの同期方法を確認します（つまり、新しいエントリーが作成されるかどうか、および Directory Server を削除する場合に、Active Directory でそのエントリーを削除するかどうか）を確認します。*



3.

OK をクリックします。

G.4.3.3. 管理者エントリーの変更

管理コンソールをインストールすると、コンソールの管理者アクセスで 2 つのエントリーが作成されます。メインエントリーは **Configuration Administrator** *で、設定ディレクトリー全体 (o=NetscapeRoot)にアクセスして変更することを承認されています。Configuration Administrator エントリーは、uid=*username*,ou= Administrators,ou=TopologyManagement,o=NetscapeRootエントリーに保存されます。*

Configuration Administrator のユーザー名とパスワードは、サーバーの起動、停止、再起動など、限られた数のタスクを実行できる管理 サーバー管理者（管理サーバー管理者）を作成するために自動的に使用されます。Directory Server の実行中にユーザーが Red Hat 管理コンソールにログインするように、Administration Server Administrator が作成されます。管理サーバー管理者には LDAP エントリーがありません。管理サーバーの設定ファイル /usr/share/dirsrv/properties/admpw に存在します。

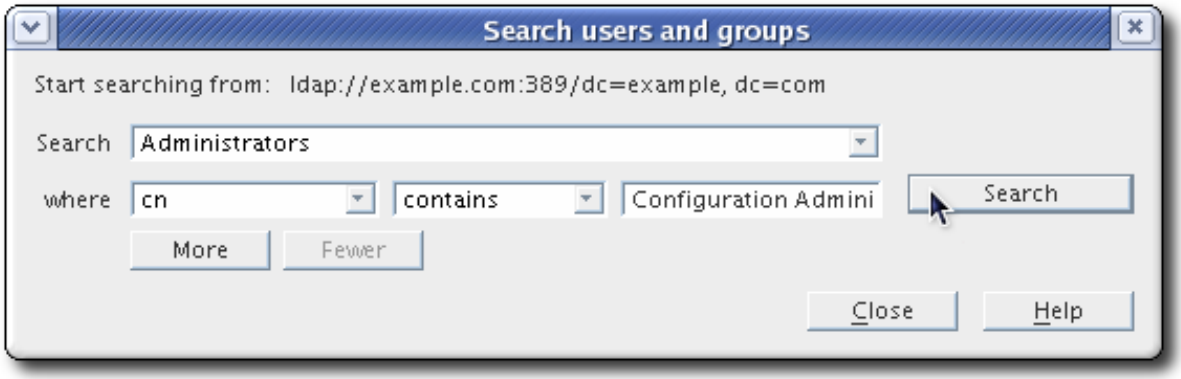
重要

これらはインストール時に同時に作成されますが、その時点で Configuration Administrator と Administration Server Administrator は 2 つの異なるエンティティーです。Red Hat 管理コンソールは、ユーザー名またはパスワードが変更された場合に、他について同じ変更を自動的に加えません。

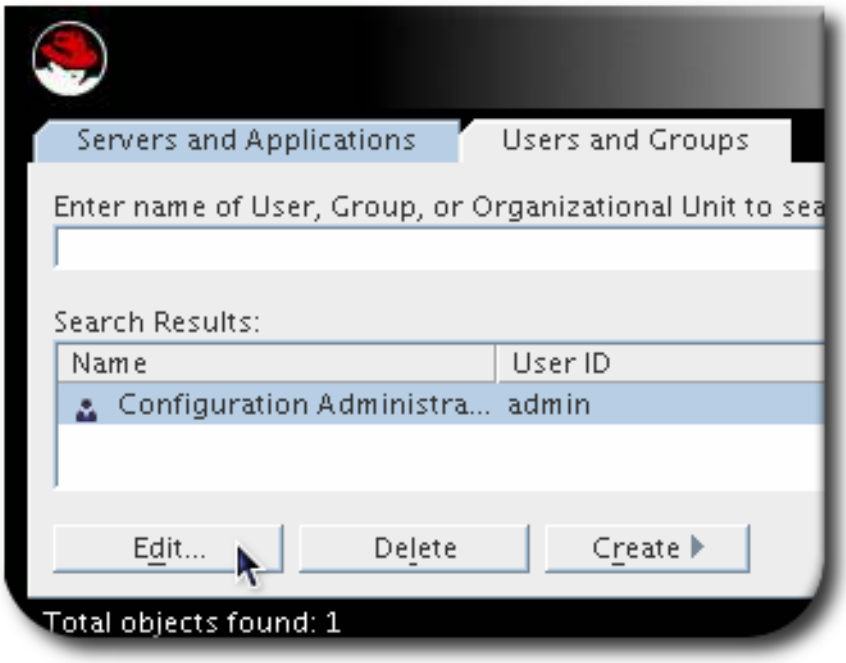
- *「設定管理者およびパスワードの変更」*
- *「管理者パスワードの変更」*
- *「管理者管理者グループへのユーザーの追加」*

G.4.3.3.1. 設定管理者およびパスワードの変更

1. *Users and Groups で、Advanced をクリックします。*
2. *Configuration Administrator を検索します。Administrators オブジェクトを選択し、デフォルトで管理者のユーザー名である Configuration Administrator を入力します。*



3. *検索結果の一覧から Configuration Administrator を選択し、Edit をクリックします。*



4.

管理者の uid およびパスワードを変更します。uid は、コンソールにログインしてコマンドを実行するために使用する naming 属性です。

5.

OK をクリックします。

注記

Configuration Administrator エントリーを編集時に Configuration Administrator としてコンソールにログインしている場合は、ディレクトリーのログイン情報を更新します。

1. *Users and Groups タブで、トップメニューの User メニューをクリックし、Change Directory を選択します。*

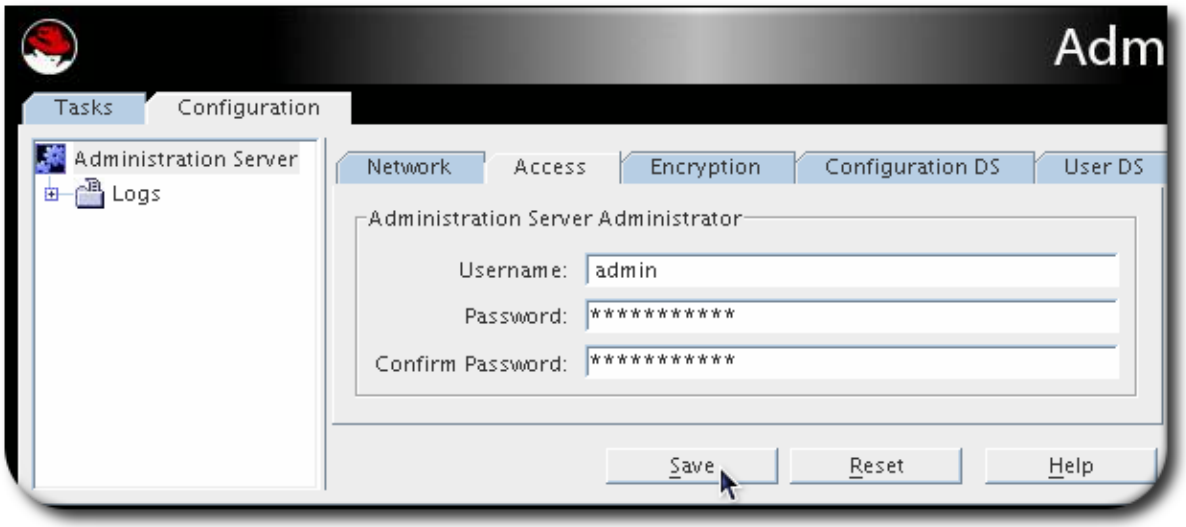
2. *Bind DN および Bind Password フィールドを Configuration Administrator の新しい情報で更新し、OK をクリックします。*

G.4.3.3.2. 管理者パスワードの変更

1. *Servers and Applications タブで Administration Server を選択し、Open をクリックします。*

2. *Configuration タブをクリックし、Access タブを開き ます。*

3. *新しいパスワードを設定します。*



警告

admin ユーザー名を変更しないでください。

4.

Save をクリックします。

5.

管理サーバーを再起動します。

```
systemctl restart dirsrv-admin.service
```

G.4.3.3.3. 管理者管理者グループへのユーザーの追加

1.

Users and Groups タブで、トップメニューの User メニューをクリックし、Change Directory を選択します。

2.

設定情報および Configuration Administrators グループが含まれる o=NetscapeRoot サブツリーに切り替えます。

3.

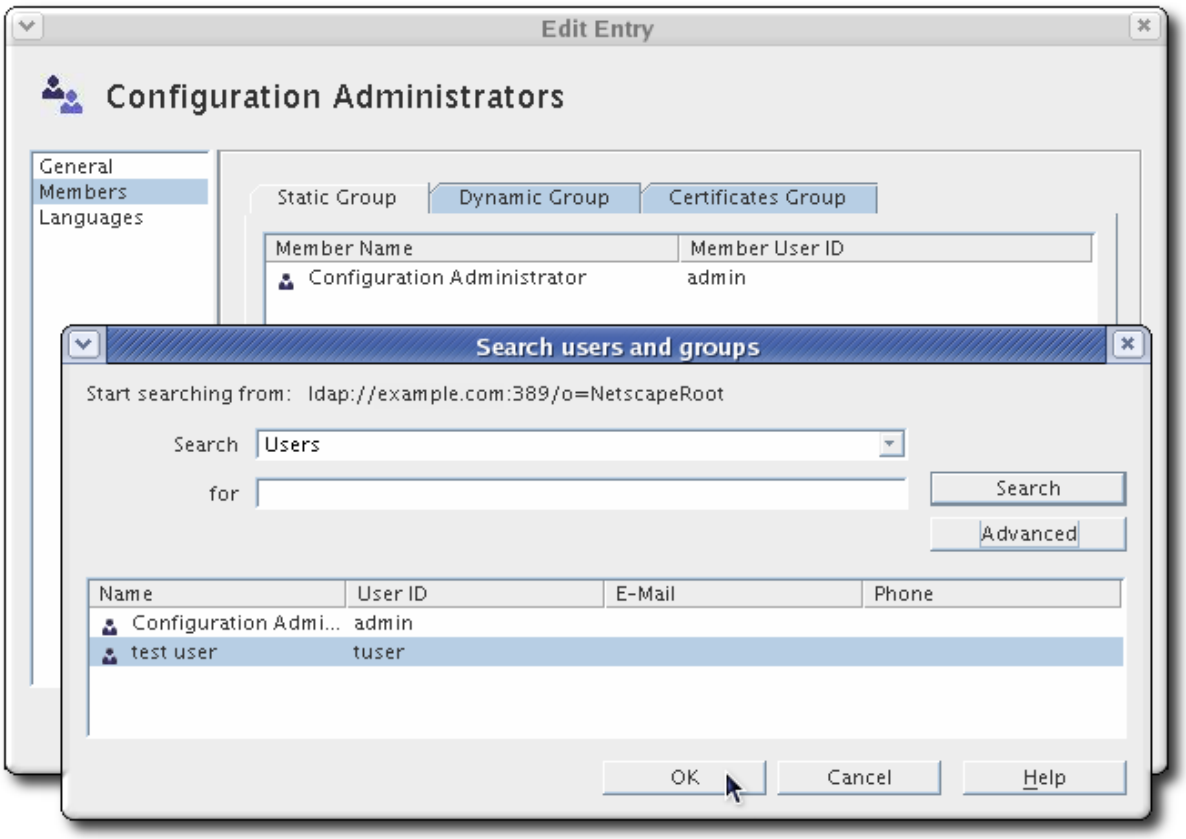
Configuration Administrators グループを検索し、Edit をクリックします。

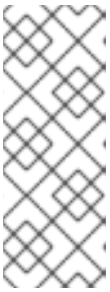
4.

編集ウィンドウの左側にある Members リンクをクリックします。

5.

Add をクリックして、グループに追加するユーザーを検索します。





注記

o=NetscapeRoot データベースのユーザーだけが Configuration Administrators グループに追加できます。つまり、コンソールを介して追加される場合に、エントリーは管理者として作成する必要があります。「ディレクトリーおよび管理ユーザー」 を参照してください。

G.4.3.4. ディレクトリーからのエントリーの削除

1.

削除するエントリーを検索します。

エントリーの検索に関する詳細は、「ユーザーおよびグループの検索」 を参照してください。

注記

すべてのエントリーは、組織ユニットから削除してから削除する必要があります。

2. *結果一覧でエントリーを選択し、Delete をクリックします。OK をクリックして削除を確定します。*

G.5. アクセス制御の設定

アクセス制御手順(ACI)は、Red Hat 管理コンソールに設定して、ユーザーが確認できる操作と、コンソールで管理される Red Hat Directory Server および管理 Server インスタンスで実行できる操作を設定できます。

ACI は、Red Hat Directory Server または管理サーバーの特定のインスタンスを使用してユーザーが実行できる操作を定義します。ACI は、アクセスまたは変更が可能なサブツリーの領域、サーバーへのアクセスに使用できるホストや日アクセスが許可される時間、許可されるサブツリーの領域にルールを設定します。

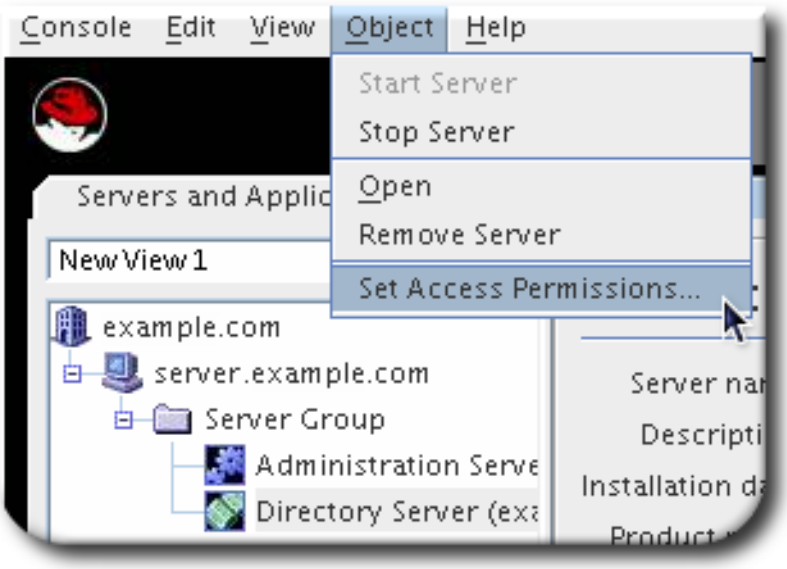
Red Hat 管理コンソールでは、アクセス制御 を使用して、管理権限を特定のユーザーに付与し、ディレクトリーの検索、ユーザーおよびグループの追加や編集、サーバーまたはコンソール設定の編集など、メインのコンソールのさまざまな側面に制限を設定できます。

G.5.1. Directory Server および管理サーバーのユーザーへの管理者権限の付与

ユーザーには管理権限を付与できます。管理ユーザーの admin ユーザーと、Directory Server の cn=Directory Manager ユーザーに似ています（特殊ユーザーである Directory Manager と全く同じではありません）。

1. *コンソールのナビゲーションツリーでサーバーを強調表示します。*

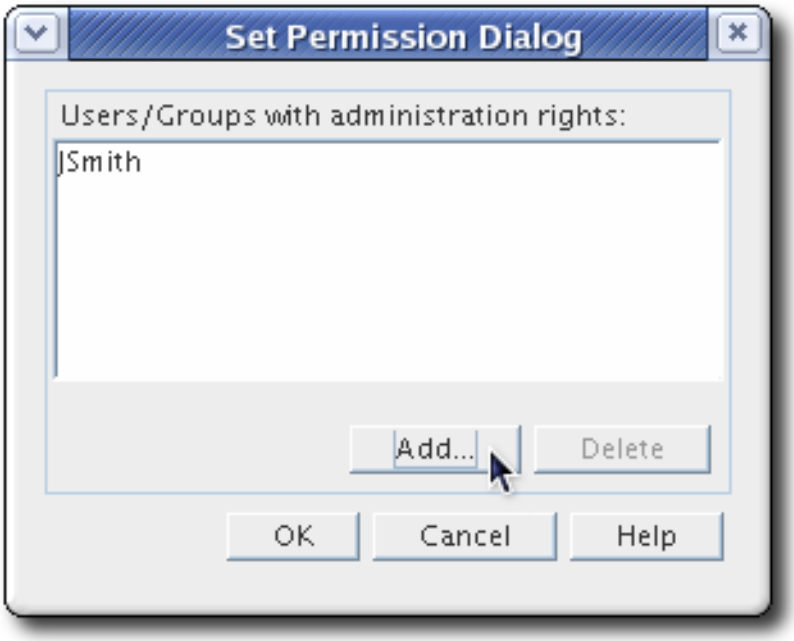
2. *Object メニューを選択し、Set Access Permissions を選択します。*



*または、エントリーを右クリックし、**Set Access Permissions** を選択します。*

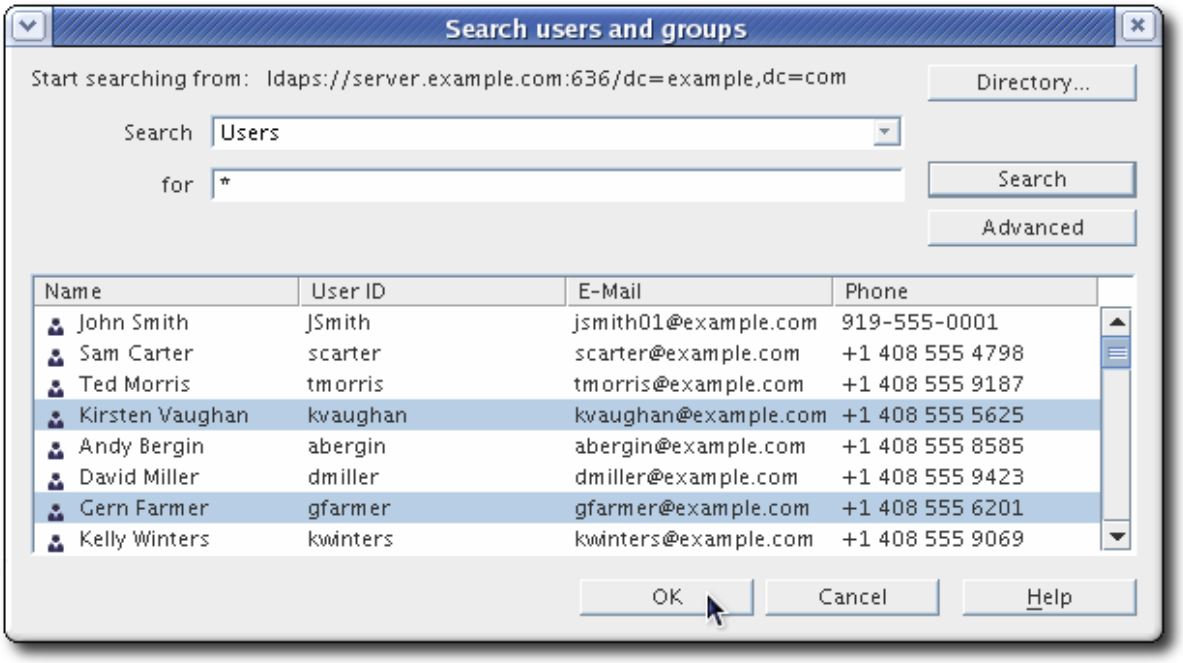
3.

Add をクリックして、サーバーの管理者一覧に新規ユーザーを追加します。管理サーバーの Directory Server の Directory Manager と admin の Directory Manager は、Set Permissions Dialog ボックスに記載されていません。



4.

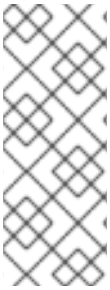
ユーザーを検索して管理者として追加します。その結果、選択したユーザーを強調表示し、Add をクリックして管理者一覧に追加します。



ユーザーおよびグループの検索に関する詳細は、「ユーザーおよびグループの検索」 を参照してください。

5.

OK をクリックして名前を Set Permissions Dialog 一覧に追加し、再度 OK をクリックして変更を保存し、ダイアログを閉じます。



注記

サーバーを管理する権限をユーザーに付与すると、そのユーザーにより、他のユーザーが同じ権利を自動的に付与することはできません。ユーザーが他のユーザーに管理者権限を付与するようにするには、「管理者管理者グループへのユーザーの追加」 の説明に従って、そのユーザーを Configuration Administrators グループに追加します。

G.5.2. コンソール要素でのアクセスパーミッションの設定

アクセス制御ルールは、コンソールで定義されている要素が 5 つあります。

- *ユーザーおよびグループタブ（表示）*
- *ユーザーおよびグループのタブ（編集）*
- *Topology タブ（編集）*

- *カスタムビュータブ（編集）*
- *サーバーセキュリティー（編集）*

デフォルトでは、これらの各 Console 要素には 5 つの継承 ACI があります。

- *匿名アクセスの有効化*
- *デフォルトの匿名アクセス*
- *設定管理者の変更*
- *グループ拡張の有効化*
- *SIE（ホスト）グループパーミッション*

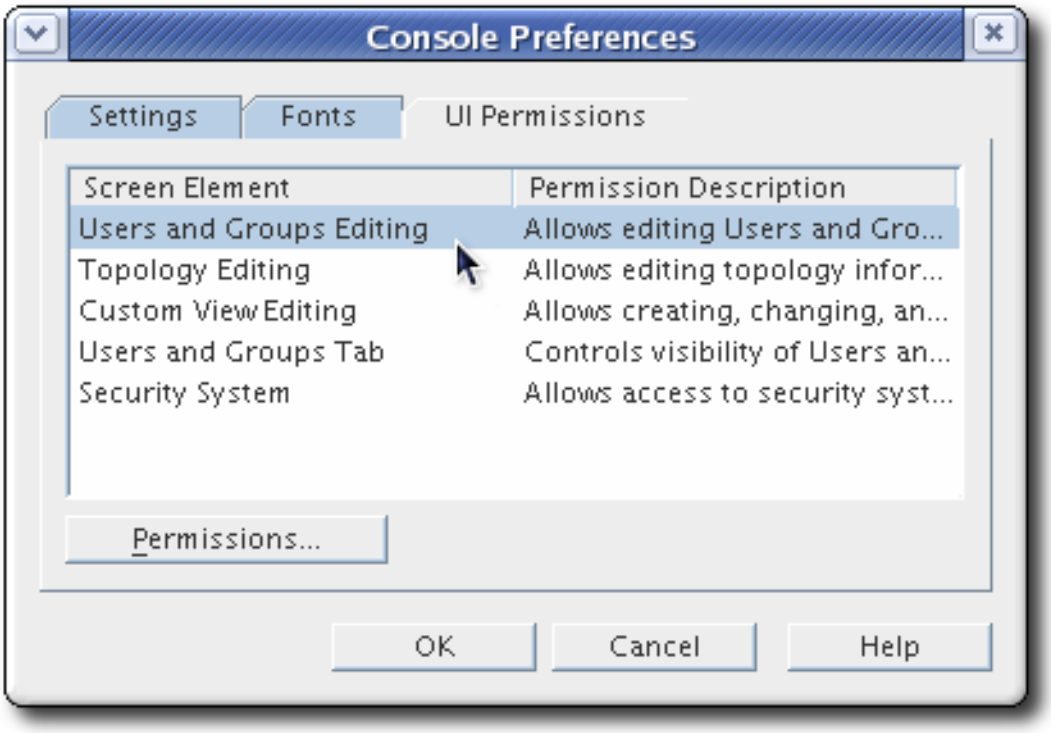
継承された ACI は編集できませんが、これらのデフォルトに加えて、各 Console 要素に新しい ACI を追加できます。追加の ACI は、たとえば、Red Hat 管理コンソール内の他のパーミッションを制限したり、変更したりでき、Directory Server インスタンスおよび管理コンソールインスタンスへのアクセスに影響します。

新規 ACI を作成するには、以下を実行します。

1. *トップメニューで Edit を選択し、Preferences を選択します。*

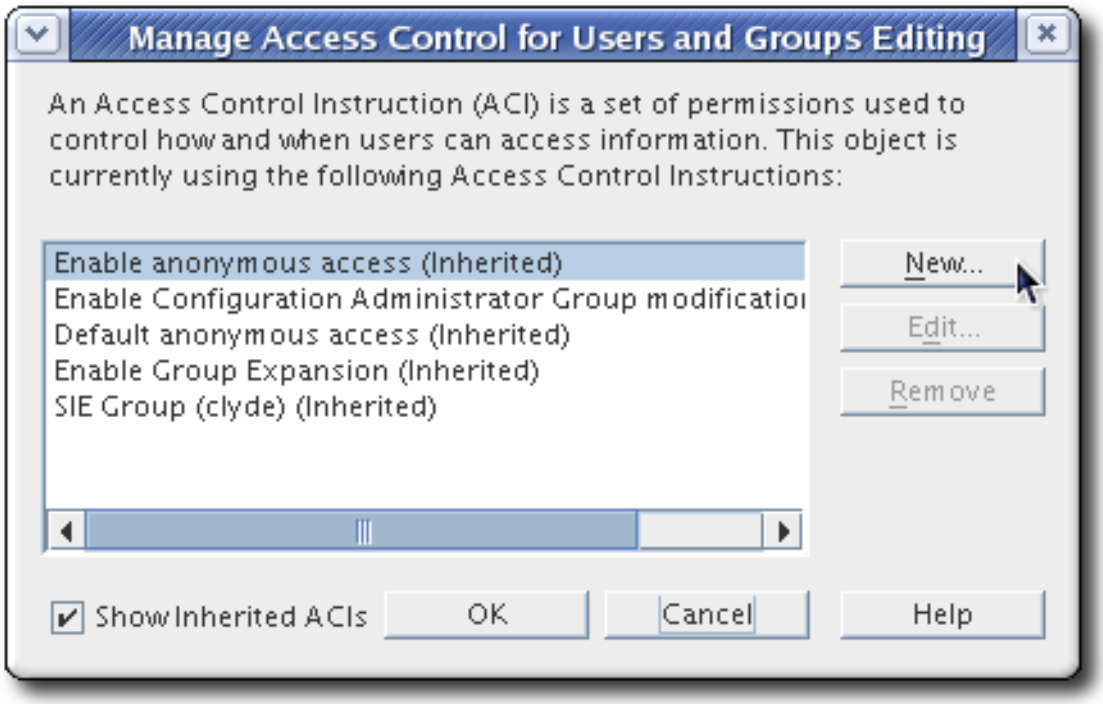
2.

*一覧から **Console** 要素を選択し、**Permissions** ボタンをクリックします。*



3.

*ACI Manager ウィンドウで、**New** ボタンをクリックします。*



*継承された **ACI** はデフォルトでは表示されません。一覧表示を表示するには、**Show inherited ACI** チェックボックスをクリックします。*

4.

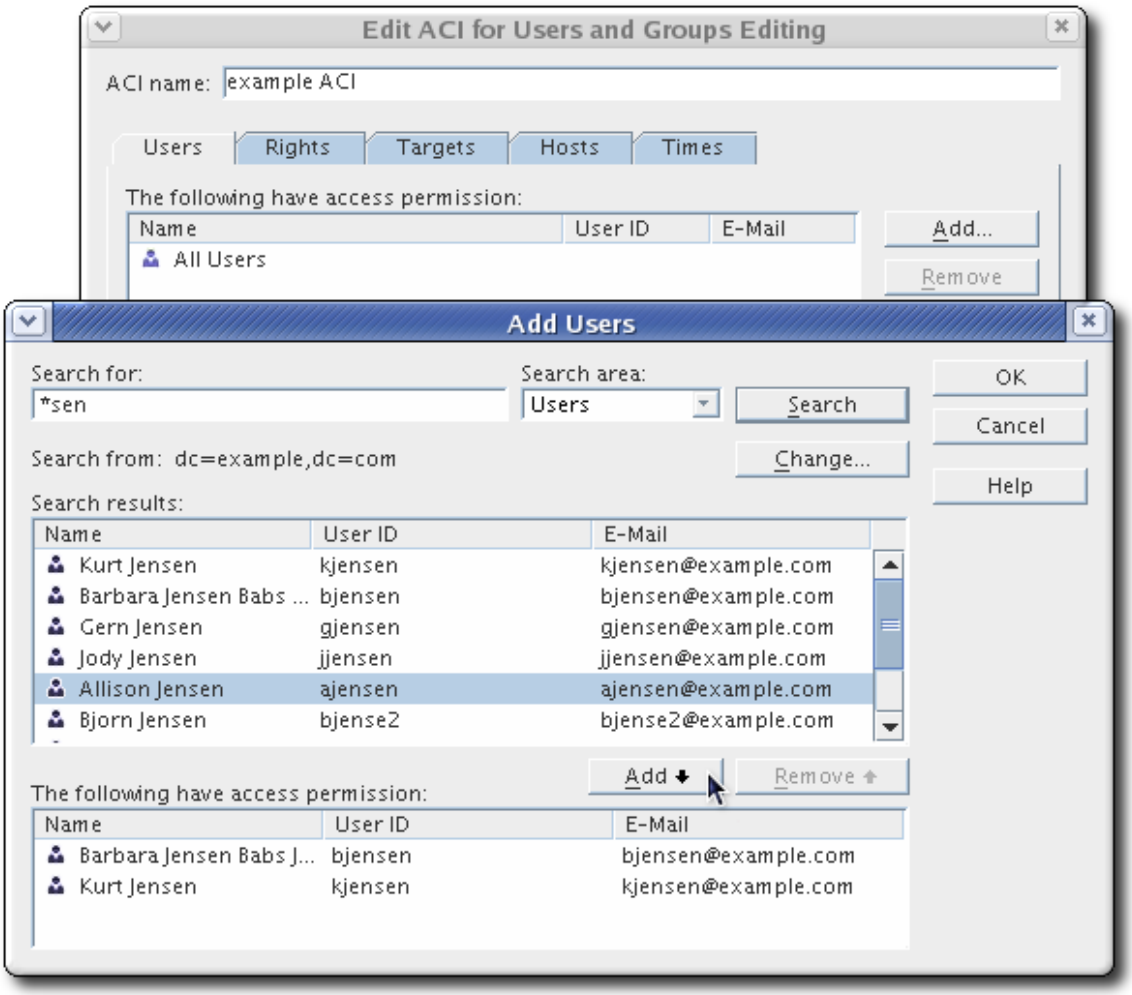
ACI は、少なくとも、適用するユーザーと、許可される権限を設定して設定します。ウィザードで ACI を設定するには（通常）、以下を実行します。

a.

ACI Name フィールドに ACI の名前を入力します。

b.

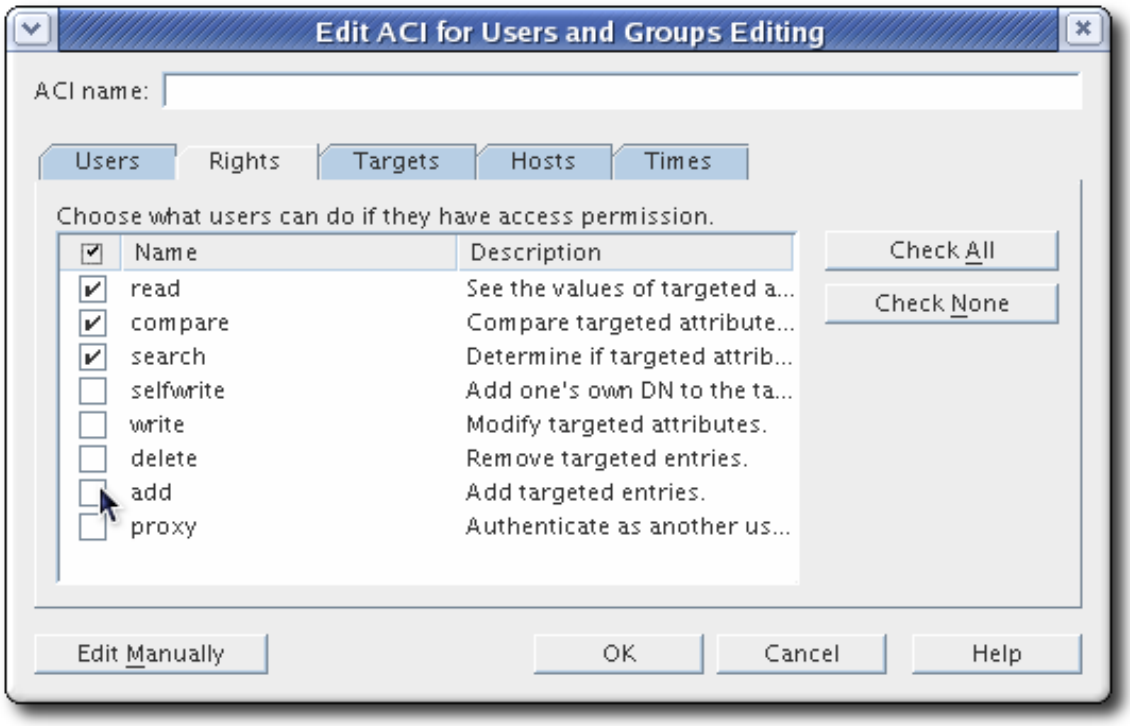
Users/Groups タブで 追加 ボタンをクリックして検索ウィンドウを開きます。ACI を適用するユーザーを検索し、追加します。



結果の一覧からユーザーを選択し、追加 ボタンをクリックして 追加します。OK をクリックして一覧を保存します。

c.

Rights タブで、この ACI の一部として許可される操作を指定します。

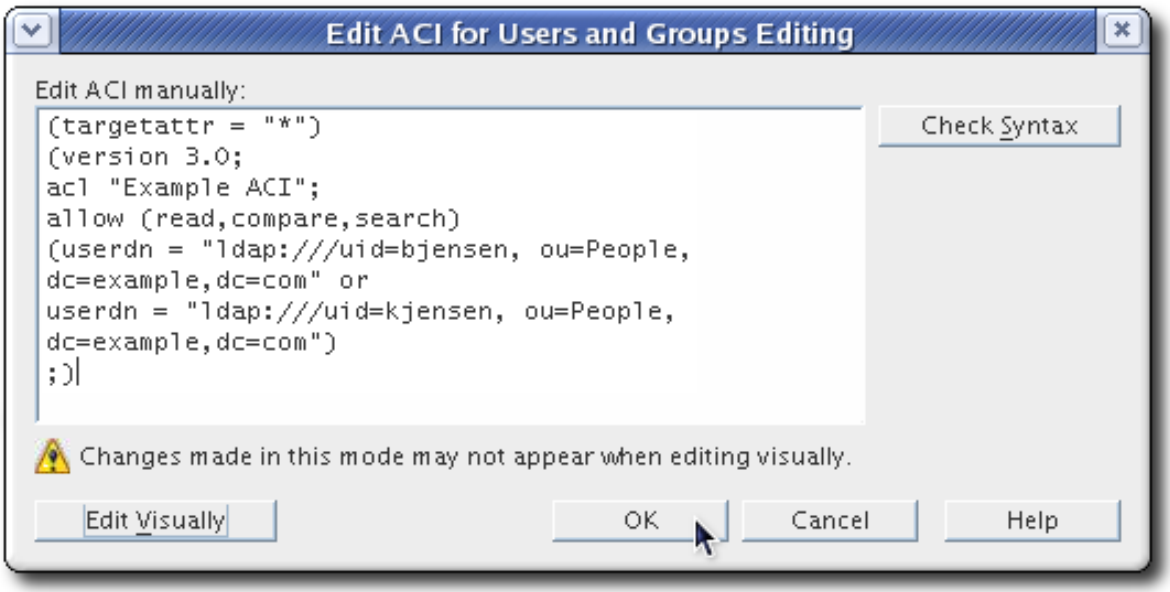


*選択したユーザー、グループ、およびホストから **Console** 要素を完全に非表示にするには、**Check None** をクリックしてすべてのアクセスをブロックします。*

d.

*必要に応じて、**ACI** が有効になるサブツリー、ホスト名、または時間にターゲットエントリーを設定します。*

*より複雑な **ACI** は視覚的に編集することができません。この場合は、**Edit Manually** ボタンをクリックして **ACI** エントリーを直接設定します。*



Check syntax ボタンを使用して ACI を検証します。

5. *OK をクリックして ACI を保存します。*

6. *Red Hat 管理コンソールを再起動して、新しい ACI を適用します。*

索引

付録H 改訂履歴

改訂番号は本ガイドに関するものであり、Red Hat Directory Server のバージョン番号ではありません。

改訂 10.6-2 **Mon Dec 07 2020** **Marc Muehlfeld**

このドキュメントは非推奨となり、維持されなくなったステートメントを追加しました。

改訂 10.6-1 **Tue Aug 11 2020** **Marc Muehlfeld**

Red Hat Directory Server 10.6 版のガイドをリリース

改訂 10.5-1 **Tue Mar 31 2020** **Marc Muehlfeld**

Red Hat Directory Server 10.5 版のガイドをリリース

改訂 10.4-2 **Mon Aug 26 2019** **Marc Muehlfeld**

『Directory Server が使用する CA 証明書の Red Hat Enterprise Linux セクションのトラストストアへの追加』を追加。

改訂 10.4-1 **Tue Aug 06 2019** **Marc Muehlfeld**

Red Hat Directory Server 10.4 版のガイドをリリース

改訂 10.3-2 **Wed Jun 05 2019** **Marc Muehlfeld**

誤った ACI サンプルが修正されました。

改訂 10.3-1 ***Wed Oct 24 2018*** ***Marc Muehlfeld***

Red Hat Directory Server 10.3 版のガイドをリリース

改訂 10.2-1 ***Tue Apr 10 2018*** ***Marc Muehlfeld***

*バージョン 10.2 の場合：『*2 つの Directory Server インスタンスの比較、『パスワードポリシーの制御作業』、『一般的なレプリケーションの競合、および命名の競合』 について*』 更新しました 『。』*

改訂 10.1-17 ***Mon Mar 12 2018*** ***Marc Muehlfeld***

『アクセス制御の管理』 の章の大きな部分が書き換えられます。

改訂 10.1-16 ***Wed Feb 14 2018*** ***Marc Muehlfeld***

*『*Directory Server の SNMP Agent のセットアップを書き換え*』 ます。『*自己署名証明書の生成およびインストールの追加。*』*

改訂 10.1-15 ***Wed Jan 17 2018*** ***Marc Muehlfeld***

*『*TLS の有効化 セクションを書き換え*』 ます。『*証明書ベースの認証を使用するようにレプリケーションパートナー*』 を設定する 機能を追加*

改訂 10.1-14 ***Tue Dec 05 2017*** ***Marc Muehlfeld***

証明書ベースの 『クライアント『認証および暗号化』 『暗号の設定』 セクションを使用して、Directory Server が使用する NSS』 データベースの管理を強化します。

改訂 10.1-13 ***Mon Nov 06 2017*** ***Marc Muehlfeld***

『レプリケーションセッションフック セクションの設定』 セクションを 『プラグインガイドに移しました』。複数のマイナー更新。

改訂 10.1-11 ***Tue Aug 08 2017*** ***Marc Muehlfeld***

『Command Line および 『Enforcing 属性 Uniqueness』 セクションを使用したエントリーの管理』 を書き換えます。

改訂 10.1-10 ***Tue Aug 01 2017*** ***Marc Muehlfeld***

バージョン 10.1.1 の場合： 『FIPS モードサポートの管理』 の章を追加。 『Directory Manager パスワードの管理』 を解除し、 『既存の属性値の構文を有効にします』。複数のマイナー更新。

改訂 10.1-9 ***Mon Jul 31 2017*** ***Marc Muehlfeld***

rewrote: 『参照整合性セクションの有効化および無効化』。レプリケーション関連の画面キャプチャーを更新しました。複数のマイナー更新。

改訂 10.1-8 ***Wed Jul 12 2017*** ***Marc Muehlfeld***

追加されたセクション 『: プラグインを設定し、』 『Directory Server 管理コンソールを起動する』。複数のマイナー更新。

改訂 10.1-7 ***Mon Jun 26 2017*** ***Marc Muehlfeld***

Rebootrote セクション：『レプリケーション changelog』 および 『レプリケーション changelog ディレクトリーの移動を行います』。

改訂 10.1-6 ***Mon May 29 2017*** ***Marc Muehlfeld***

更新されたセクション 『: VLV 情報のアクセス制御を設定します』。

改訂 10.1-5 ***Tue Mar 14 2017*** ***Marc Muehlfeld***

Added section: 『レプリケーション Keep-alive エントリー』。

改訂 10.1-4 ***Fri Feb 24 2017*** ***Marc Muehlfeld***

セクションの追加： 『Fine Grained ID List Size』, 『Trimming the Replication Changelog』, and 『Setting up Content Synchronization with an RFC 4533-aware LDAP servers』その他の若干の修正。

改訂 10.1-3 ***Wed Jan 11 2017*** ***Marc Muehlfeld***

書き換え セクション： 『ログファイルの設定』 replicate_now.sh の例を更新しました。

改訂 10.1-1 ***Fri Nov 11 2016*** ***Marc Muehlfeld***

サポートされる JRE バージョンを更新しました。レガシーレプリケーションのサポートが削除されました。その他の若干の修正。

改訂 10.1-0 ***Mon Oct 31 2016*** ***Marc Muehlfeld***

Red Hat Directory Server 10.1 ガイドのリリース

改訂 10.0-3 ***Wed Jun 22 2016*** ***Petr Bokoč***

更新されたセクション： 『Directory Server、Administration Server、および Console で TLS/SSL を有効にします』。*その他の若干の更新。*

改訂 10.0-2 ***Thu Mar 24 2016*** ***Petr Bokoč***

更新されたセクション： 『スキーマレプリケーション』。*セクションの追加：* 『High-resolution Time Stamps』*その他の若干の修正。*

改訂 10.0-1 ***Wed Jun 17 2015*** ***Tomáš Čapek***

更新セクション： 『システム要件』

改訂 10.0-0 ***Tue Jun 09 2015*** ***Tomáš Čapek***

Red Hat Directory Server 10 ガイドのリリース