



# Red Hat Directory Server 11

設定、コマンド、およびファイルリファレンス

Directory Server を設定するためのリファレンスガイド



# Red Hat Directory Server 11 設定、コマンド、およびファイルリファレンス

---

Directory Server を設定するためのリファレンスガイド

## 法律上の通知

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 概要

これは、Red Hat Directory Server の設定パラメーター、サーバースキーマ、ファイル、およびコマンドラインユーティリティーのリファレンスです。

## 目次

前書き .....	5
法的通知 .....	5
多様性を受け入れるオープンソースの強化 .....	6
このリファレンスについて .....	7
1. DIRECTORY SERVER の概要 .....	7
第1章 概要 .....	8
1.1. DIRECTORY SERVER 設定 .....	8
1.2. DIRECTORY SERVER インスタンスファイルのリファレンス .....	8
1.3. DIRECTORY SERVER コマンドラインユーティリティーの使用 .....	8
第2章 ファイルの場所の概要 .....	9
2.1. DIRECTORY SERVER インスタンスに依存しないファイルおよびディレクトリー .....	9
2.2. DIRECTORY SERVER インスタンス固有のファイルおよびディレクトリー .....	9
第3章 CORE SERVER 設定リファレンス .....	21
3.1. コアサーバー設定の属性リファレンス .....	21
3.2. 設定オブジェクトクラス .....	231
3.3. ROOT DSE 属性 .....	245
3.4. レガシーの属性 .....	249
第4章 プラグイン実装サーバー機能リファレンス .....	253
4.1. サーバープラグインの機能リファレンス .....	253
4.2. すべてのプラグインに共通の属性のリスト .....	293
4.3. 特定のプラグインで使用できる属性 .....	298
4.4. データベースプラグインの属性 .....	301
4.5. データベースリンクプラグイン属性 (チェーン属性) .....	348
4.6. PAM パススルー認証プラグイン属性 .....	361
4.7. アカウントポリシープラグインの属性 .....	365
4.8. AD DN プラグインの属性 .....	369
4.9. AUTO MEMBERSHIP プラグインの属性 .....	370
4.10. DISTRIBUTED NUMERIC ASSIGNMENT プラグインの属性 .....	375
4.11. LINKED ATTRIBUTES 属性プラグインの属性 .....	386
4.12. MANAGED ENTRIES プラグインの属性 .....	388
4.13. MEMBEROF プラグイン属性 .....	390
4.14. ATTRIBUTE UNIQUENESS プラグインの属性 .....	393
4.15. POSIX WINSYNC API プラグインの属性 .....	396
4.16. RETRO CHANGELOG プラグインの属性 .....	398
4.17. ROOTDN アクセス制御プラグインの属性 .....	402
4.18. REFERENTIAL INTEGRITY プラグインの属性 .....	405
第5章 ディレクトリーエントリースキーマのリファレンス .....	407
5.1. DIRECTORY SERVER スキーマについて .....	407
5.2. エントリー属性の参照 .....	416
5.3. エントリーオブジェクトクラスの参照 .....	527
第6章 運用属性とオブジェクトクラス .....	623
6.1. ACCOUNTUNLOCKTIME .....	623
6.2. ACI .....	623
6.3. ALTSERVER .....	623
6.4. CREATETIMESTAMP .....	624

6.5. CREATORSNAME	624
6.6. DITCONTENTRULES	624
6.7. DITSTRUCTURERULES	624
6.8. ENTRYUSN	625
6.9. INTERNALCREATORSNAME	625
6.10. INTERNALMODIFIERSNAME	626
6.11. HASSUBORDINATES	626
6.12. LASTLOGINTIME	626
6.13. LASTMODIFIEDBY	627
6.14. LASTMODIFIEDTIME	627
6.15. LDAPSUBENTRY	627
6.16. LDAPSYNTAXES	628
6.17. MATCHINGRULES	628
6.18. MATCHINGRULEUSE	628
6.19. MODIFYTIMESTAMP	629
6.20. MODIFIERSNAME	629
6.21. NAMEFORMS	629
6.22. NSACCOUNTLOCK	630
6.23. NSAIMSTATUSGRAPHIC	630
6.24. NSAIMSTATUSTEXT	630
6.25. NSBACKENDSUFFIX	630
6.26. NSCPENTRYDN	631
6.27. NSDS5REPLCONFLICT	631
6.28. NSICQSTATUSGRAPHIC	631
6.29. NSICQSTATUSTEXT	632
6.30. NSIDLETIMEOUT	632
6.31. NSIDLISTSCANLIMIT	632
6.32. NSLOOKTHROUGHLIMIT	632
6.33. NSPAGEDIDLISTSCANLIMIT	633
6.34. NSPAGEDLOOKTHROUGHLIMIT	633
6.35. NSPAGEDSIZELIMIT	633
6.36. NSPARENTUNIQUEID	634
6.37. NSROLE	634
6.38. NSROLEDN	634
6.39. NSROLEFILTER	635
6.40. NSSCHEMACSN	635
6.41. NSSIZELIMIT	636
6.42. NSTIMELIMIT	636
6.43. NSTOMBSTONE (オブジェクトクラス)	636
6.44. NSUNIQUEID	637
6.45. NSYIMSTATUSGRAPHIC	637
6.46. NSYIMSTATUSTEXT	637
6.47. NUMSUBORDINATES	638
6.48. PASSWORDGRACEUSERTIME	638
6.49. PASSWORDRETRYCOUNT	638
6.50. PWDPOLICYSUBENTRY	639
6.51. PWDUPDATETIME	639
6.52. SUBSCHEMASUBENTRY	639
6.53. GLUE(オブジェクトクラス)	639
6.54. PASSWORDOBJECT (オブジェクトクラス)	640
6.55. SUBSCHEM (オブジェクトクラス)	641
<b>第7章 ログファイルのリファレンス</b> .....	<b>642</b>

---

7.1. アクセスログリファレンス	642
7.2. エラーログ参照	654
7.3. 監査ログリファレンス	663
7.4. LDAP の結果コード	664
7.5. ログファイルの名前付きパイプへの置き換え	667
<b>第8章 設定ファイルのリファレンス</b> .....	<b>672</b>
8.1. CERTMAP.CONF	672
<b>第9章 コマンドラインユーティリティー</b> .....	<b>675</b>
9.1. DS-REPLCHECK	675
9.2. LDIF	675
9.3. DBSCAN	675
9.4. DS-LOGPIPE.PY	677
9.5. DN2RDN	679
9.6. PWDHASH	680
<b>付録A DIRECTORY SERVER で利用可能なスクリプトのテスト</b> .....	<b>681</b>
A.1. LDCLT (負荷ストレステスト)	681
A.2. RSEARCH(検索ストレステスト)	697
<b>付録B レプリカ合意のステータス</b> .....	<b>709</b>
<b>用語集</b> .....	<b>713</b>
<b>付録C 更新履歴</b> .....	<b>724</b>





## 前書き

Directory Server を設定するためのリファレンスガイド

### 法的通知

Copyright 2021 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

**Linux** is the registered trademark of Linus Torvalds in the United States and other countries.

**Java** is a registered trademark of Oracle and/or its affiliates.

**XFS** is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

**MySQL** is a registered trademark of MySQL AB in the United States, the European Union and other countries.

**Node.js** is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The **OpenStack** Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、用語の置き換えは、今後の複数のリリースにわたって段階的に実施されます。詳細は、[弊社の CTO、Chris Wright のメッセージ](#) を参照してください。

## このリファレンスについて

Red Hat Directory Server (Directory Server) は、業界標準の Lightweight Directory Access Protocol (LDAP) をベースとした強力でスケーラブルな分散ディレクトリーサーバーです。Directory Server は、イントラネット、取引先とのエクストラネット、またはパブリックインターネットを介して顧客に到達するために使用できる一元化された分散データリポジトリを構築するための基礎です。

このリファレンスは、サーバー設定とコマンドラインユーティリティーに対応しています。これは主に、ディレクトリー管理者向けに設計されており、そのディレクトリーにアクセスするためにコマンドラインを使用する経験のあるディレクトリーユーザー向けに設計されています。サーバーの設定後、この参照を使用してサーバーを維持します。

Directory Server は、グラフィカルユーザーインターフェイスである Directory Server コンソールから管理できます。**Red Hat Directory Server 管理ガイド**では、これを行う方法と、個々の管理タスクについて詳細に説明しています。

### 1. DIRECTORY SERVER の概要

Directory Server の主なコンポーネントには、以下が含まれます。

- LDAP サーバー: LDAP v3 準拠のネットワークデーモン
- Directory Server Console: ディレクトリーサービスの設定と保守の労力を大幅に削減するグラフィカルな管理コンソール。
- SNMP エージェント: Simple Network Management Protocol (SNMP) を使用して Directory Server を監視できます。

# 第1章 概要

Directory Server は、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。Directory Server は、大規模なスケールディレクトリーを管理し、ユーザーおよびリソースのエンタープライズ全体ディレクトリー、エクストラネット、およびインターネットでの e-commerce アプリケーションをサポートするよう設計されている、堅牢かつスケーラブルなサーバーです。Directory Server は、マシンの **ns-slaped** プロセスまたはサービスとして実行します。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。

ほとんどの Directory Server の管理タスクは、Directory Server コンソール、Directory Server で提供されるグラフィカルユーザーインターフェイスから実行できます。Directory Server コンソールの使用に関する情報は、**Red Hat Directory Server 管理ガイド**を参照してください。

このリファレンスは、コマンドラインとコマンドラインユーティリティーおよびスクリプトを使用してサーバー設定属性を変更し、Directory Server を管理する他の方法を処理します。

## 1.1. DIRECTORY SERVER 設定

Directory Server の設定情報およびすべてのサーバー属性のリストを保存する形式および方法は、[3章 Core Server 設定リファレンス](#) と [4章 プラグイン実装サーバー機能リファレンス](#) の2つの章を参照してください。

## 1.2. DIRECTORY SERVER インスタンスファイルのリファレンス

「[Directory Server インスタンスに依存しないファイルおよびディレクトリー](#)」には、Directory Server の各インスタンスに保存されるファイルおよび設定情報の概要があります。これは、管理者がディレクトリーアクティビティーの過程で変更や変更がない場合に理解するのに役立つリファレンスです。セキュリティの観点からは、通常の変更と異常な動作を強調表示し、エラーと侵入を検知するのに役立ちます。

## 1.3. DIRECTORY SERVER コマンドラインユーティリティーの使用

Directory Server には、ディレクトリー内のエントリーの検索や変更、サーバーの管理が可能な一連の設定可能なコマンドラインユーティリティーが含まれています。[9章 コマンドラインユーティリティー](#)では、このコマンドラインユーティリティーを説明し、ユーティリティーを保存する場所と、そのアクセス方法を説明します。

## 第2章 ファイルの場所の概要

Red Hat Directory Server は、ファイルシステム階層標準 (FHS) と互換性があります。FHS の詳細は <http://refspecs.linuxfoundation.org/fhs.shtml> を参照してください。

### 2.1. DIRECTORY SERVER インスタンスに依存しないファイルおよびディレクトリー

Directory Server のインスタンスに依存しないデフォルトファイルおよびディレクトリーの場所を以下に示します。

型	場所
コマンドラインユーティリティー	<code>/usr/bin/</code> <code>/usr/sbin/</code>
systemd ユニットファイル	<code>/usr/lib/systemd/system/dirsrv.target</code> <code>/etc/systemd/system/dirsrv.target.wants/</code>

### 2.2. DIRECTORY SERVER インスタンス固有のファイルおよびディレクトリー

同じホストで実行されている複数のインスタンスを分離するには、特定のファイルおよびディレクトリーにはインスタンスの名前が含まれます。Directory Server の設定中にインスタンス名を設定します。デフォルトでは、これはドメイン名のないホスト名です。たとえば、完全修飾ドメイン名が **server.example.com** の場合、デフォルトのインスタンス名は **server** になります。

Directory Server のインスタンス固有のデフォルトファイルおよびディレクトリーの場所を以下に示します。

型	場所
バックアップファイル	<code>/var/lib/dirsrv/slapd-instance_name/bak/</code>
設定ファイル	<code>/etc/dirsrv/slapd-instance_name/</code>
証明書および鍵のデータベース	<code>/etc/dirsrv/slapd-instance_name/</code>
データベースファイル	<code>/var/lib/dirsrv/slapd-instance_name/db/</code>
LDIF ファイル	<code>/var/lib/dirsrv/slapd-instance/ldif/</code>
ロックファイル	<code>/var/lock/dirsrv/slapd-instance_name/</code>
ログファイル	<code>/var/log/dirsrv/slapd-instance_name/</code>

型	場所
PID ファイル	<code>/var/run/dirsrv/instance_name.pid</code>
systemd ユニットファイル	<code>/etc/systemd/system/dirsrv.target.wants/dirsrv@instance_name.service</code>

## 2.2.1. 設定ファイル

各 Directory Server インスタンスは、設定ファイルを `/etc/dirsrv/slaped-instance` ディレクトリーに保存します。

Red Hat Directory Server の設定情報は、そのディレクトリー内に LDAP エントリーとして保存されます。そのため、単純に設定ファイルを編集するのではなく、サーバー設定への変更はサーバー自体を使用して実装する必要があります。この設定ストレージの方法の主な利点は、ディレクトリー管理者が LDAP を使用してサーバーを再設定できることです。これにより、ほとんどの設定変更のためにサーバーをシャットダウンする必要がなくなります。

### 2.2.1.1. Directory Server 設定の概要

Directory Server が設定されると、デフォルト設定が、サブツリー `cn=config` のディレクトリー内にある一連の LDAP エントリーとして保存されます。サーバーが起動すると、`cn=config` サブツリーの内容は、LDIF 形式のファイル (`dse.ldif`) から読み込まれます。`dse.ldif` ファイルには、すべてのサーバー設定情報が含まれます。このファイルの最新バージョンは `dse.ldif` と呼ばれ、最後の変更前のバージョンは `dse.ldif.bak` と呼ばれ、サーバーが正常に起動する最新のファイルが `dse.ldif.startOK` と呼ばれます。

Directory Server の機能の多くは、コアサーバーに接続するための個別モジュールとして設計されています。各プラグインの内部設定の詳細は、`cn=plugins,cn=config` 配下の個別のエントリーに含まれます。たとえば、Telephone 構文プラグインの設定は、以下のエントリーに含まれています。

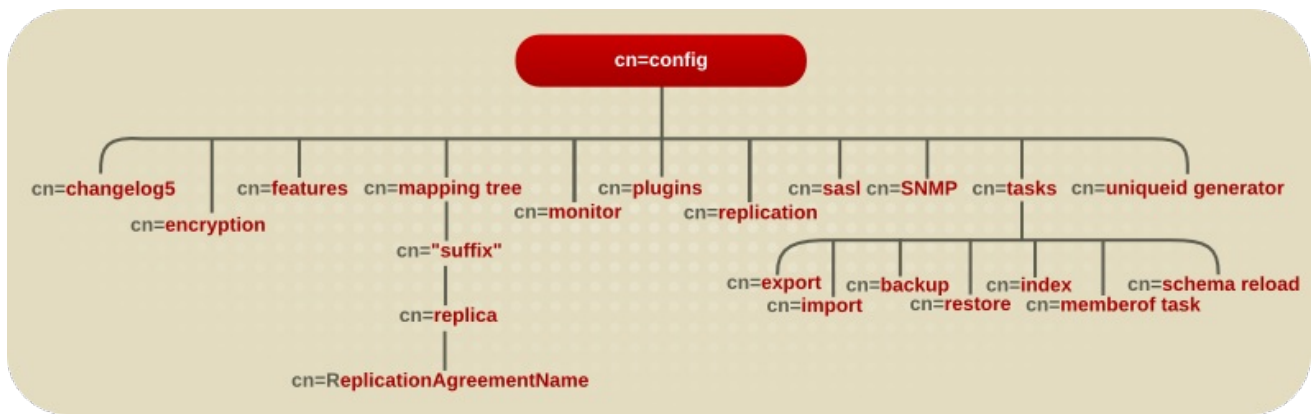
```
cn=Telephone Syntax,cn=plugins,cn=config
```

同様に、データベース固有の設定は以下に保存されます。

`cn=ldb database,cn=plugins,cn=config` (ローカルデータベースの場合) および `cn=chaining database,cn=plugins,cn=config` (データベースリンクの場合)

以下の図は、`cn=config` ディレクトリー情報ツリー内で設定データがどのように適合するかを示しています。

図2.1 設定データを示すディレクトリー情報ツリー



### 2.2.1.1.1. LDIF およびスキーマ設定ファイル

Directory Server の設定データは、`/etc/dirsrv/slapd-instance` ディレクトリーの LDIF ファイルに保存されます。そのため、サーバー識別子が `phonebook` で、Directory Server の場合は、設定 LDIF ファイルはすべて `/etc/dirsrv/slapd-phonebook` の下に保存されます。

このディレクトリーには、他のサーバーインスタンス固有の設定ファイルも含まれます。

スキーマ設定は LDIF 形式でも保存され、これらのファイルは `/etc/dirsrv/schema` ディレクトリーに置かれます。

以下の表は、Directory Server で提供されるすべての設定ファイルを表しています。その設定ファイルには、他の互換性のあるサーバーのスキーマも含まれます。各ファイルの前には、読み込む順序を示す番号が付いています (数値の昇順、次にアルファベットの昇順)。

表2.1 Directory Server LDIF 設定ファイル

設定ファイル名	目的
dse.ldif	サーバーの起動時にディレクトリーによって作成されたフロントエンドのディレクトリー固有のエントリーが含まれます。これには、Root DSE ("" ) および <b>cn=config</b> および <b>cn=monitor</b> の内容が含まれません (acis のみ)。
00core.ldif	最低限の機能セット (ユーザースキーマなし、コア以外の機能のスキーマなし) でサーバーを起動するために必要なスキーマ定義のみが含まれます。ユーザー、機能、およびアプリケーションが使用するその他のスキーマは <b>01common.ldif</b> と他のスキーマファイルにあります。このファイルは変更しないでください。

設定ファイル名	目的
01common.ldif	<b>subschemaSubentry</b> 、RFC2256 (X.520/X.521 ベース) で定義された LDAPv3 標準のユーザーおよび組織スキーマ、 <b>inetOrgPerson</b> などの広く使われている属性や、Directory Server の設定で使われる運用属性など、LDAPv3 標準の運用スキーマが含まれています。このファイルを変更すると、相互運用性の問題が発生します。ユーザー定義の属性は Directory Server コンソールを使用して追加する必要があります。
05rfc2247.ldif	RFC 2247、および Using Domains in LDAP/X500 Distinguished Names.の関連コレクションスキーマのスキーマ。
05rfc2927.ldif	RFC 2927 からのスキーマ MIME Directory Profile for LDAP Schema。 <b>subschema</b> サブエントリに表示する属性に必要な <b>ldapSchemas</b> 操作属性が含まれます。
10presence.ldif	レガシー。インスタントメッセージングプレゼンス (オンライン) 情報のスキーマ。このファイルには、ユーザーがインスタントメッセージングプレゼンス情報を利用できるようにするためにユーザーのエントリに追加する必要のある、許可された属性を持つデフォルトのオブジェクトクラスがリスト表示されます。
10rfc2307.ldif	RFC 2307 からのスキーマ LDAP をネットワーク情報サービスとして使用するためのアプローチ。これは、そのスキーマが使用可能になる <b>10rfc2307bis</b> 、 <b>rfc2307</b> の新バージョンに、取って代わられます。
20subscriber.ldif	新しいスキーマ要素と Nortel サブスクライバーの相互運用性仕様が含まれています。以前は <b>50ns-delegated-admin.ldif</b> ファイルに保存されていた、 <b>adminRole</b> 属性、 <b>memberOf</b> 属性、および <b>inetAdmin</b> オブジェクトクラスが含まれます。
25java-object.ldif	RFC 2713 のスキーマ Schema for Representing Java® Objects in an LDAP Directory
28pilot.ldif	RFC 1274 のパイロットディレクトリースキーマが含まれていますが、これは新しいデプロイメントには推奨されなくなりました。RFC 1274 を成功する今後の RFC は、すべて <b>28pilot.ldif</b> 属性タイプおよびクラスを非推奨とする場合があります。



設定ファイル名	目的
30ns-common.ldif	Directory Server コンソールフレームワークに共通するオブジェクトクラスおよび属性が含まれるスキーマ。
50ns-admin.ldif	Red Hat 管理サーバーによって使用されるスキーマ。
50ns-certificate.ldif	Red Hat Certificate Management System のスキーマ。
50ns-directory.ldif	Directory Server 4.12 以前のバージョンで使用される追加の設定スキーマがディレクトリーに含まれており、これは現在のバージョンの Directory Server には適用されなくなりました。このスキーマは、Directory Server 4.12 と現在のリリース間の複製に必要です。
50ns-mail.ldif	メールサーバーがメールユーザーおよびメールグループを定義するのに Netscape Messaging Server が使用するスキーマ。
50ns-value.ldif	サーバーの値のアイテム属性のスキーマ。
50ns-web.ldif	Netscape Web Server のスキーマ。
60pam-plugin.ldif	将来の使用のために予約されています。
99user.ldif	サプライヤーからの属性とオブジェクトクラスが含まれる Directory Server レプリケーションコンシューマーによって維持されるユーザー定義のスキーマ。

#### 2.2.1.1.2. サーバー設定の組織化方法

**dse.ldif** ファイルには、データベースに関連するエントリーなど、サーバーの起動時にディレクトリー固有のエントリーを含むすべての設定情報が含まれます。このファイルには、root Directory Server エントリー (または "" という名前の DSE) と **cn=config** および **cn=monitor** のコンテンツが含まれます。

サーバーが **dse.ldif** ファイルを生成すると、エントリーが **cn=config** の下のディレクトリーに表示される順番にリスト表示されます。これは、ベース **cn=config** のサブツリースコープの LDAP 検索の順序と同じです。

**dse.ldif** には **cn=monitor** エントリーも含まれています。このエントリーは主に読み取り専用ですが、ACI を設定できます。



## 注記

**dse.ldif** ファイルには、**cn=config** のすべての属性は含まれません。管理者によって属性が設定されておらず、デフォルト値がある場合は、サーバーはその属性を **dse.ldif** に書き込みません。**cn=config** のすべての属性を表示するには、**ldapsearch** を使用します。

## 設定属性

設定エントリー内では、各属性は属性名として表されます。属性の値は属性の設定に対応します。

以下のコード例は、Directory Server の **dse.ldif** ファイルの一部になります。この例では、スキーマチェックが有効になっていると表示されます。これは、**nsslapd-schemacheck** 属性で表され、値は **on** になります。

```
dn: cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsslapdConfig
nsslapd-accesslog-logging-enabled: on
nsslapd-enquote-sup-oc: off
nsslapd-localhost: phonebook.example.com
nsslapd-schemacheck: on
nsslapd-port: 389
nsslapd-localuser: dirsrv
...
```

## プラグイン機能の設定

Directory Server プラグイン機能の各設定には、独自のエントリーと、サブツリー **cn=plugins,cn=config** 下の属性セットがあります。以下のコード例は、プラグインの例である Telephone Syntax プラグインの設定エントリーの例です。

```
dn: cn=Telephone Syntax,cn=plugins,cn=config
objectclass: top
objectclass: nsSlapdPlugin
objectclass: extensibleObject
cn: Telephone Syntax
nsslapd-pluginType: syntax
nsslapd-pluginEnabled: on
```

これらの属性の一部はすべてのプラグインに共通するものであり、特定のプラグインに固有の場合もあります。**cn=config** サブツリーで **ldapsearch** を実行して、特定のプラグインで現在使用されている属性を確認します。

Directory Server がサポートするプラグイン、一般的なプラグイン設定情報、プラグイン設定属性の参照、および設定変更に必要なプラグインのリストは、[4章 プラグイン実装サーバー機能リファレンス](#) を参照してください。

## データベースの設定

データベースプラグインエントリーの **cn=UserRoot** サブツリーには、セットアップ時に作成されたデフォルトの接尾辞を含むデータベースの設定データが含まれます。

これらのエントリーとその子には、キャッシュサイズ、インデックスファイルおよびトランザクションログへのパス、監視および統計情報のためのエントリーおよび属性など、さまざまなデータベース設定を設定するために使用される多くの属性があります。

## インデックスの設定

インデックスの設定情報は、以下の information-tree ノード配下の Directory Server のエントリーとして保存されます。

- `cn=index,cn=UserRoot,cn=ldb database,cn=plugins,cn=config`
- `cn=default indexes,cn=config,cn=ldb database,cn=plugins,cn=config`

一般インデックスの詳細は、[Red Hat Directory Server 管理ガイド](#)を参照してください。インデックス設定属性の詳細は、「[cn=config,cn=ldb database,cn=plugins,cn=config](#) 下のデータベース属性」を参照してください。

### 2.2.1.2. サーバー設定へのアクセスおよび変更

このセクションでは、設定エントリーのアクセス制御について説明し、サーバー設定を表示および変更できる各種の方法を説明します。また、変更を有効にするためにサーバーを再起動する必要がある属性に加え、変更の種類に制限についても扱います。

#### 2.2.1.2.1. 設定エントリーのアクセス制御

Directory Server がインストールされると、`cn=config` 下のすべてのエントリーに対して、デフォルトのアクセス制御命令 (ACI) が実装されます。以下のコードサンプルは、これらのデフォルト ACI の例です。

```
aci: (targetattr = "*" )(version 3.0; acl "Local Directory Administrators Group"; allow (all)
  groupdn = "ldap:///ou=Directory Administrators,dc=example,dc=com");
```

これらのデフォルト ACI により、以下のユーザーがすべての LDAP 操作がすべての設定属性に対して実行できます。

- Configuration Administrators グループのメンバー
- 管理者として機能するユーザーは、セットアップで設定した **admin** アカウントです。デフォルトでは、これはコンソールにログインしているユーザーアカウントと同じです。
- ローカルの Directory Administrators グループのメンバー。
- SIE (Server Instance Entry) グループは、通常 **Set Access Permissions** プロセスを使用してメインコンソールに割り当てられます。

アクセス制御の詳細は、[Red Hat Directory Server 管理ガイド](#)を参照してください。

#### 2.2.1.2.2. 設定属性の変更

サーバー属性は、3つの方法 (Directory Server コンソール、`ldapsearch` コマンドおよび `ldapmodify` コマンドの実行、または `dse.ldif` ファイルの手動編集) のいずれかで表示および変更できます。



#### 注記

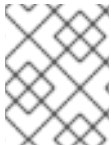
`dse.ldif` ファイルを編集する前にサーバーを停止する **必要** があります。それ以外の場合は、変更が失われます。`dse.ldif` ファイルの編集は、動的に変更できない属性の変更のみに推奨されます。詳細は [サーバー再起動を必要とする設定変更](#) を参照してください。

次のセクションでは、LDAP を使用して (Directory Server Console とコマンドラインの両方を使用して) エントリーを変更する方法、エントリーの変更に適用される制限、属性の変更に適用される制限、および再起動が必要な設定の変更を説明します。

## LDAP を使用した設定エントリーの変更

ディレクトリーの設定エントリーは、Directory Server Console を使用するか、他のディレクトリーエントリーと同じ方法で **ldapsearch** 操作および **ldapmodify** 操作を実行して LDAP を使用して検索および変更できます。LDAP を使用してエントリーを変更する利点は、サーバーの実行中に変更可能です。

詳細は、**Red Hat Directory Server 管理ガイド**のディレクトリーエントリーの作成の章を参照してください。ただし、特定の変更を考慮するには、サーバーを再起動する必要があります。詳細は [サーバー再起動を必要とする設定変更](#) を参照してください。



### 注記

設定ファイルセットと同様に、Directory Server 機能に影響を与えるリスクがあるため、**cn=config** サブツリーのノードを変更または削除する場合には注意が必要です。

常にデフォルト値を取る属性を含む設定全体を表示するには、**cn=config** サブツリーの **ldapsearch** 操作を実行します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b "cn=config" -s sub -x "(objectclass=*)"
```

- **bindDN** は、サーバーのインストール時に Directory Manager に対して選択される DN です (デフォルトでは **cn=Directory Manager**)。
- **Password** は、Directory Manager に選択されるパスワードです。

プラグインを無効にするには、**ldapmodify** を使用して **nsslapd-pluginEnabled** 属性を編集します。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=Telephone Syntax,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: off
```

## 設定エントリーおよび属性を変更する制限

サーバーエントリーおよび属性を変更する場合、特定の制限が適用されます。

- The **cn=monitor** エントリーとその子エントリーは読み取り専用で、ACI の管理以外は変更できません。
- **cn=config** に属性が追加されると、サーバーは属性を無視します。
- 属性に無効な値が入力されると、サーバーはそれを無視します。
- **ldapdelete** はエントリー全体を削除するために使用されているため、**ldapmodify** を使用してエントリーから属性を削除します。

## サーバー再起動を必要とする設定変更

サーバーの実行中に一部の設定属性を変更することはできません。このような場合、変更を反映するには、サーバーをシャットダウンして再起動する必要があります。この変更は、Directory Server コン

ソールを使用するか、手動で **dse.ldif** ファイルを編集して行う必要があります。変更を反映するためにサーバーを再起動する必要がある属性の一部を以下に示します。このリストは網羅的ではありません。完全なリストを表示するには、**ldapsearch** を実行し、**nsslapd-requiresrestart** 属性を検索します。以下に例を示します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b "cn=config" -s sub -x "(objectclass=*)" | grep nsslapd-requiresrestart
```

<b>nsslapd-cachesize</b>	<b>nsslapd-certdir</b>
<b>nsslapd-dbcachesize</b>	<b>nsslapd-dbncache</b>
<b>nsslapd-plugin</b>	<b>nsslapd-changelogdir</b>
<b>nsslapd-changelogmaxage</b>	<b>nsslapd-changelogmaxentries</b>
<b>nsslapd-port</b>	<b>nsslapd-schemadir</b>
<b>nsslapd-saslpath</b>	<b>nsslapd-secureport</b>
<b>nsslapd-tmpdir</b>	<b>nsSSLclientauth</b>
<b>nsSSLSessionTimeout</b>	<b>nsslapd-conntablesz</b>
<b>nsslapd-lockdir</b>	<b>nsslapd-maxdescriptors</b>
<b>nsslapd-reservedescriptors</b>	<b>nsslapd-listenhost</b>
<b>nsslapd-schema-ignore-trailing-spaces</b>	<b>nsslapd-securelistenhost</b>
<b>nsslapd-workingdir</b>	<b>nsslapd-return-exact-case</b>
<b>nsslapd-maxbersize</b> <sup>[a]</sup>	

[a] この属性には再起動が必要ですが、検索では返されません。

## 設定属性の削除

**/etc/dirsrv/slapd-instance-name/dse.ldif** ファイルに書き込まれていない場合でも、コア設定属性はすべてサーバーで使用されるデフォルト値を持つためにあります。

コア設定属性と削除できない属性のリストの詳細は、[Red Hat Directory Server 管理ガイド](#)の該当するセクションを参照してください。

### 2.2.2. データベースファイル

各 Directory Server インスタンスには、すべてのデータベースファイルを保存する **/var/lib/dirsrv/slapd-instance/db** ディレクトリーが含まれます。以下は、**/var/lib/dirsrv/slapd-instance/db** ディレクトリーの内容のサンプルです。

## 例2.1 データベースディレクトリーの内容

```
db.001 db.002 __db.003 DBVERSION log.0000000001 userroot/
```

- **db.00x** ファイル - データベースによって内部で使用されるため、いかなる方法でも移動、削除、または変更しないでください。
- **log.xxxxxxxxxx** ファイル - データベースごとにトランザクションログを保存するために使用されます。
- **DBVERSION** - データベースのバージョンを保存するために使用します。
- **userRoot** - 設定で作成されるユーザー定義の接尾辞 (ユーザー定義のデータベース) を保存します。たとえば、**dc=example,dc=com** を保存します。



### 注記

ディレクトリーツリーを新しい接尾辞の下に保存する新規データベース (例: **testRoot**) が作成されると、**testRoot** という名前のディレクトリーも **/var/lib/dirsrv/slapped-instance/db** ディレクトリーに表示されます。

以下は、**userRoot** ディレクトリーの内容の例です。

## 例2.2 UserRoot データベースのディレクトリーコンテンツ

```
ancestorid.db
DBVERSION
entryrdn.db
id2entry.db
nsuniqueid.db
numsubordinates.db
objectclass.db
parentid.db
```

**userroot** サブディレクトリーには以下のファイルが含まれます。

- **ancestorid.db** - エントリーの先祖の ID を検索する ID のリストが含まれています。
- **entrydn.db** - ID を検索する完全な DN のリストが含まれています。
- **id2entry.db** - 実際のディレクトリーデータベースエントリーが含まれます。他のすべてのデータベースファイルは、必要に応じてこのデータベースファイルから再作成できます。
- **nsuniqueid.db** - ID を検索する一意の ID のリストが含まれています。
- **numsubordinates.db** - 子エントリーを持つ ID が含まれます。
- **objectclass.db** - 特定のオブジェクトクラスを持つ ID のリストが含まれます。
- **parentid.db** - 親の ID を検索する ID のリストが含まれます。

### 2.2.3. LDIF ファイル

LDIF ファイルの例は、LDIF 関連のファイルを保存する `/var/lib/dirsrv/slapped-instance/ldif` ディレクトリーに保存されます。例2.3「LDIF ディレクトリーの内容」は、`/ldif` ディレクトリーの内容をリスト表示します。

#### 例2.3 LDIF ディレクトリーの内容

```
European.ldif
Example.ldif
Example-roles.ldif
Example-views.ldif
```

- **European.ldif:** ヨーロッパの文字サンプルが含まれます。
- **example.ldif:** LDIF ファイルのサンプルです。
- **example-roles.ldif:** **Example.ldif** と同様に LDIF ファイルの例です。ただし、ディレクトリー管理者にアクセス制御およびリソース制限を設定するグループの代わりに、サービスのロールとクラスを使用する点が異なります。



#### 注記

インスタンスディレクトリーの **db2ldif** スクリプトまたは **db2ldif.pl** スクリプトがエクスポートした LDIF ファイルは、`/var/lib/dirsrv/slapped-instance/ldif` に保存されます。

### 2.2.4. ロックファイル

各 Directory Server インスタンスには、ロック関連のファイルを保存する `/var/lock/dirsrv/slapped-instance` ディレクトリーが含まれます。以下は、**locks** ディレクトリーのコンテンツのリスト表示例です。

#### 例2.4 ディレクトリーコンテンツのロック

```
exports/ imports/ server/
```

ロックメカニズムは、Directory Server プロセスのコピーを1つで実行できる数を制御します。たとえば、インポートジョブがある場合は、ロックが **imports/** ディレクトリーに格納され、他の **ns-slapd** (通常)、**ldif2db** (他のインポート)、または **db2ldif** (エクスポート) の操作が実行されないようにします。サーバーが通常通りに実行されている場合は、**server/** ディレクトリーにロックがあり、インポート操作は妨げられ(エクスポート操作は妨げられない)、エクスポート操作がある場合、**exports/** ディレクトリーのロックは、通常のサーバー操作を許可しますが、インポート操作を防ぎます。

利用可能なロックの数は、Directory Server の全体的なパフォーマンスに影響を及ぼす可能性があります。ロックの数は、**nsldapd-db-locks** 属性に設定されます。属性値の調整については、**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。

### 2.2.5. ログファイル

各 Directory Server インスタンスには、ログファイルを保存する `/var/log/dirsrv/slapped-instance` ディレクトリーが含まれます。以下は、**/logs** ディレクトリーの内容をリスト表示した例です。

## 例2.5 ログディレクトリーのコンテンツ

```
access          access.20200228-171925 errors
access.20200221-162824 access.rotationinfo  errors.20200221-162824
access.20200223-171949 audit              errors.rotationinfo
access.20200227-171818 audit.rotationinfo slapd.stats
```

- **access**、**audit**、および **error** のログファイルの内容は、ログ設定によって異なります。
- **slapd.stats** ファイルはメモリーにマッピングされたファイルで、エディターで読み込むことができません。これには、Directory Server SNMP データ収集コンポーネントによって収集されるデータが含まれます。このデータは SNMP 属性クエリーに対応して SNMP サブエージェントによって読み取られ、Directory Server SNMP 要求を処理する SNMP マスターエージェントと通信します。

[7章 ログファイルのリファレンス](#) アクセス、エラー、監査ログファイル形式の概要と、それらの情報を説明します。

### 2.2.6. PID ファイル

**slapd-serverID.pid** ファイルおよび **slapd-serverID.startpid** ファイルは、サーバーの稼働時に **/var/run/dirsrv** ディレクトリーに作成されます。どちらのファイルもサーバーのプロセス ID を保存します。

### 2.2.7. バックアップファイル

各 Directory Server インスタンスには、バックアップ関連のファイルを保存するための以下のディレクトリーとファイルが含まれます。

- **/var/lib/dirsrv/slapd-instance/bak**: これには、インスタンス、データベースバックアップの日時 (例: インスタンスの-2020\_05\_02\_16\_56\_05/ など) がデータベースのバックアップのコピーを保持します。
- **/etc/dirsrv/slapd-instance/dse\_original.ldif**: これは、インストール時からの **dse.ldif** 設定ファイルのバックアップコピーです。



## 第3章 CORE SERVER 設定リファレンス

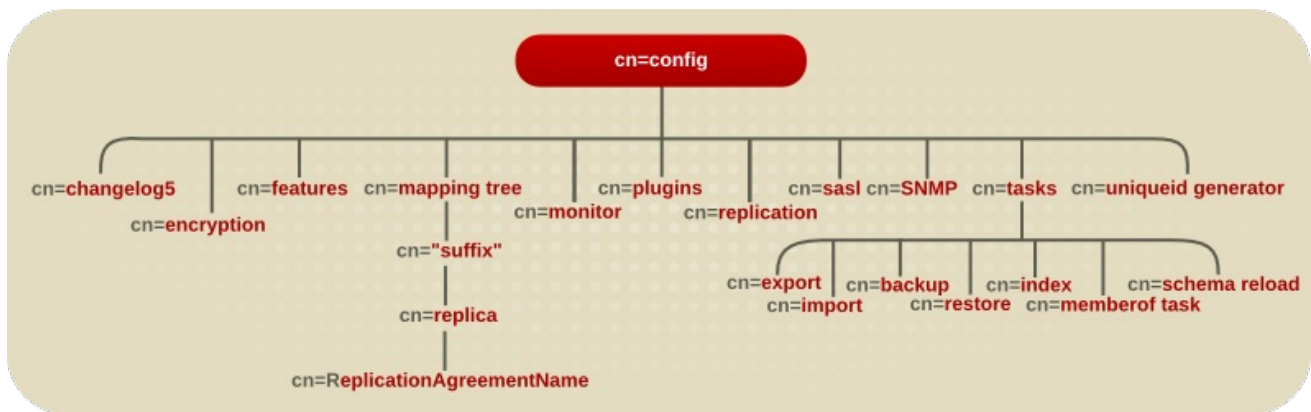
本章では、すべてのコア (サーバー関連) 属性のアルファベットの参照を提供します。「[Directory Server 設定の概要](#)」 Red Hat Directory Server 設定ファイルに関する適切な概要が含まれています。

### 3.1. コアサーバー設定の属性リファレンス

このセクションには、コアサーバーの機能に関連する設定属性の参照情報が記載されています。サーバー設定の変更に関する詳細は、「[サーバー設定へのアクセスおよび変更](#)」を参照してください。プラグインとして実装されるサーバー機能のリストは、「[サーバープラグインの機能リファレンス](#)」を参照してください。カスタムサーバー機能の実装に関するヘルプは、Directory Server サポートにお問い合わせください。

以下の図のように、**dse.ldif** ファイルに保存された設定情報は、一般的な設定エントリー **cn=config** 配下にある情報ツリーとして編成されます。

図3.1 設定データを示すディレクトリー情報ツリー



この設定ツリーのノードのほとんどは、以下のセクションで説明されています。

**cn=plugins** ノードは [4章 プラグイン実装サーバー機能リファレンス](#) で説明されています。各属性の説明には、ディレクトリーエントリーの DN、デフォルト値、値の有効な範囲、その使用方法などが含まれます。



#### 注記

本章で説明するエントリーおよび属性の一部は、製品の今後のリリースで変更される可能性があります。

#### 3.1.1. cn=config

一般的な設定エントリーは **cn=config** エントリーに保存されます。**cn=config** エントリーは、**nsslapdConfig** オブジェクトクラスのインスタンスで、**extensibleObject** オブジェクトクラスを継承します。

##### 3.1.1.1. nsslapd-accesslog(アクセスログ)

この属性は、各 LDAP アクセスを記録するために使用されるログのパスおよびファイル名を指定します。ログファイルには、デフォルトで以下の情報を記録します。

- データベースにアクセスするクライアントマシンの IP アドレス (IPv4 または IPv6)。
- 実行される操作 (検索、追加、変更など)。

- アクセス権の結果 (返されるエントリーの数やエラーコードなど)。

アクセスログをオフにする方法は、**Red Hat Directory Server 管理ガイド**のサーバーおよびデータベースアクティビティーの監視の章を参照してください。

アクセスロギングを有効にするには、この属性に有効なパスとパラメーターが必要で、**nsslapd-accesslog-logging-enabled** 設定属性を **on** に切り替える必要があります。以下の表は、これらの2つの設定属性と、アクセスロギングを無効化または有効にするという2つの設定属性に対して可能な4つの値の組み合わせを示しています。

表3.1 dse.ldif ファイル属性

属性	値	ロギングの有効化または無効化
nsslapd-accesslog-logging-enabled nsslapd-accesslog	on 空の文字列	Disabled
nsslapd-accesslog-logging-enabled nsslapd-accesslog	on filename	有効
nsslapd-accesslog-logging-enabled nsslapd-accesslog	off 空の文字列	Disabled
nsslapd-accesslog-logging-enabled nsslapd-accesslog	off filename	Disabled

パラメーター	説明
エントリー DN	cn=config
有効な値	有効なファイル名。
デフォルト値	/var/log/dirsrv/slapd- <b>instance</b> /access
構文	DirectoryString
例	nsslapd-accesslog: /var/log/dirsrv/slapd- <b>instance</b> /access

### 3.1.1.2. nsslapd-accesslog-level(アクセスログレベル)

この属性は、アクセスログにログ記録する内容を制御します。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	<ul style="list-style-type: none"> <li>* 0 - アクセスロギングなし</li> <li>* 4 - 内部アクセス操作のロギング</li> <li>* 256 - 接続、操作、および結果の記録</li> <li>* 512 - エントリーおよび参照情報にアクセスするためのロギング</li> <li>* これらの値を一緒に追加して、必要なロギングタイプを提供します。たとえば、<b>516</b> (4 + 512) を使用して内部アクセス操作、エントリーアクセス、参照ロギングを取得します。</li> </ul>
デフォルト値	256
構文	整数
例	nsslapd-accesslog-level: 256

### 3.1.1.3. nsslapd-accesslog-list(アクセスログファイルの List)

設定できないこの読み取り専用属性は、アクセスログのローテーションで使用されるアクセスログファイルのリストを提供します。

パラメーター	説明
エントリー DN	cn=config
有効な値	
デフォルト値	なし
構文	DirectoryString
例	nsslapd-accesslog-list: accesslog2,accesslog3

### 3.1.1.4. nsslapd-accesslog-logbuffering(Log Buffering)

**off** に設定すると、サーバーはすべてのアクセスログエントリーを直接ディスクに書き込みます。バッファを使用すると、パフォーマンスに影響を与えずに負荷が大きい場合でもサーバーがアクセスロギングを使用できます。ただし、デバッグ時には、操作と、ログエントリーがファイルにフラッシュされるのを待たずに、バッファを無効にしても、すぐに結果を見ることがあります。ログバッファリングを無効にすると、負荷の高いサーバーのパフォーマンスに深刻な影響を与える可能性があります。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-accesslog-logbuffering: off

### 3.1.1.5. nsslapd-accesslog-logexpirationtime(アクセスログの有効期限)

この属性は、削除前にログファイルに到達できる最大期間を指定します。この属性はユニット数のみを提供します。ユニットは、**nsslapd-accesslog-logexpirationtimeunit** 属性で指定されます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1 から最大 32 ビットの整数値 (2147483647) -1 または 0 の値は、ログが期限切れになることはありません。
デフォルト値	-1
構文	整数
例	nsslapd-accesslog-logexpirationtime: 2

### 3.1.1.6. nsslapd-accesslog-logexpirationtimeunit(アクセスログの有効期限時間単位)

この属性は、**nsslapd-accesslog-logexpirationtime** 属性の単位を指定します。ユニットがサーバーによって認識されていない場合は、ログが期限切れになることはありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	month   week   day
デフォルト値	month
構文	DirectoryString

パラメーター	説明
例	nsslapd-accesslog-logexpirationtimeunit: week

### 3.1.1.7. nsslapd-accesslog-logging-enabled(アクセスログの有効化ロギング)

accesslog ロギングを無効にして有効にしますが、各データベースへのアクセスを記録するのに使用されるログのパスおよびパラメーターを指定する **nsslapd-accesslog** 属性と併せてのみ有効です。

アクセスロギングを有効にするには、この属性を **on** に切り替え、**nsslapd-accesslog** 設定属性に有効なパスとパラメーターが必要です。以下の表は、これらの2つの設定属性と、アクセスロギングを無効化または有効にするという2つの設定属性に対して可能な4つの値の組み合わせを示しています。

表3.2 dse.ldif Attributes

属性	値	ログの有効化または無効化
nsslapd-accesslog-logging-enabled nsslapd-accesslog	on 空の文字列	Disabled
nsslapd-accesslog-logging-enabled nsslapd-accesslog	on <b>filename</b>	有効
nsslapd-accesslog-logging-enabled nsslapd-accesslog	off 空の文字列	Disabled
nsslapd-accesslog-logging-enabled nsslapd-accesslog	off <b>filename</b>	Disabled

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-accesslog-logging-enabled: off

### 3.1.1.8. nsslapd-accesslog-logmaxdiskspace(Access Log Maximum Disk Space)

この属性は、アクセスログが消費できる最大ディスク容量 (メガバイト単位) を指定します。この値を超えると、最も古いアクセスログが削除されます。

最大ディスク容量を設定する場合は、ログファイルのローテーションにより作成可能なログファイルの合計数を考慮してください。また、Directory Server が管理する 3 つの異なるログファイル (アクセスログ、監査ログ、およびエラーログ) があり、それぞれがディスク領域を消費することに注意してください。これらの考慮事項を、アクセスログのディスク領域の合計量と比較します。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1 1 から最大 32 ビットの整数値 (2147483647)。値が -1 の場合は、アクセスログに許可されるディスク領域のサイズが無制限であることを意味します。
デフォルト値	500
構文	整数
例	nsslapd-accesslog-logmaxdiskspace: 500

### 3.1.1.9. nsslapd-accesslog-logminfreediskspace(アクセスログ最小空きディスク容量)

この属性は、許可される最小空きディスク容量をメガバイト単位で設定します。空きディスク容量がこの属性で指定された値を下回ると、この属性を満たすために十分なディスク領域が解放されるまで、最も古いアクセスログが削除されます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1 1 から最大 32 ビットの整数値 (2147483647)
デフォルト値	-1
構文	整数
例	nsslapd-accesslog-logminfreediskspace: -1

### 3.1.1.10. nsslapd-accesslog-logrotationsync-enabled(アクセスログローテーション同期の有効化)

この属性は、アクセスログのローテーションが、特定の日に同期されるかどうかを設定します。この方法でログローテーションを同期すると、毎日深夜から深夜までなど、1 日の指定された時間にログファイルを生成できます。これにより、ログファイルがカレンダーに直接マップされるため、ログファイルの分析がはるかに簡単になります。

アクセスログのローテーションを時刻で同期するには、この属性を **nsslapd-accesslog-logrotationsynchour** 属性値および **nsslapd-accesslog-logrotationsyncmin** 属性値をログファイルのローテーションの時間と分に設定して、この属性を有効にする必要があります。

たとえば、アクセスログファイルを毎日真夜中にローテーションするには、この属性の値を **on** に設定して有効にし、**nsslapd-accesslog-logrotationsynchour** 属性および **nsslapd-accesslog-logrotationsyncmin** 属性の値を **0** に設定します。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-accesslog-logrotationsync-enabled: on

### 3.1.1.11. nsslapd-accesslog-logrotationsynchour(アクセスログローテーション同期時間)

この属性は、アクセスログをローテーションする時刻を設定します。この属性は、**nsslapd-accesslog-logrotationsync-enabled** 属性および **nsslapd-accesslog-logrotationsyncmin** 属性と共に使用する必要があります。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 - 23
デフォルト値	0
構文	整数
例	nsslapd-accesslog-logrotationsynchour: 23

### 3.1.1.12. nsslapd-accesslog-logrotationsyncmin(アクセスログローテーション同期確認確認)

この属性は、アクセスログをローテーションする日数を設定します。この属性は、**nsslapd-accesslog-logrotationsync-enabled** 属性および **nsslapd-accesslog-logrotationsynchour** 属性と共に使用する必要があります。

パラメーター	説明
エントリー DN	cn=config

パラメーター	説明
Valid Range	0 - 59
デフォルト値	0
構文	整数
例	nsslapd-accesslog-logrotationsyncmin: 30

### 3.1.1.13. nsslapd-accesslog-logrotationtime(アクセスログローテーション時間)

この属性は、アクセスログファイルのローテーションの間隔を設定します。この属性はユニット数のみを提供します。単位 (day、week、month など) は **nsslapd-accesslog-logrotationtimeunit** 属性で指定します。

Directory Server は、ログのサイズにかかわらず、設定された間隔が過ぎた後の最初の書き込み操作でログをローテーションします。

パフォーマンス上の理由から、ログローテーションを指定しないことは推奨していませんが、ログが無期限に増大するため、これを指定する方法は 2 つあります。 **nsslapd-accesslog-maxlogsperdir** 属性値を **1** に設定するか、 **nsslapd-accesslog-logrotationtime** 属性を **-1** に設定します。サーバーは最初に **nsslapd-accesslog-maxlogsperdir** 属性をチェックして、この属性の値が **1** を超える場合、サーバーは **nsslapd-accesslog-logrotationtime** 属性をチェックします。詳細は、 [「nsslapd-accesslog-maxlogsperdir\(アクセスログの最大数\)」](#) を参照してください。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1 1 から最大 32 ビットの整数値 (2147483647)。値が -1 の場合は、アクセスログファイルのローテーションの間隔が無制限になります。
デフォルト値	1
構文	整数
例	nsslapd-accesslog-logrotationtime: 100

### 3.1.1.14. nsslapd-accesslog-logrotationtimeunit(アクセスログローテーション時間単位)

この属性は、 **nsslapd-accesslog-logrotationtime** 属性の単位を設定します。

パラメーター	説明
エントリー DN	cn=config



パラメーター	説明
有効な値	month   week   day   hour   minute
デフォルト値	day
構文	DirectoryString
例	nsslapd-accesslog-logrotationtimeunit: week

### 3.1.1.15. nsslapd-accesslog-maxlogsize(アクセスログの最大サイズ)

この属性は、最大アクセスログサイズをメガバイト単位で設定します。この値に達すると、アクセスログはローテーションされます。つまり、サーバーがログ情報を新しいログファイルに書き込むことを意味します。**nsslapd-accesslog-maxlogsperdir** 属性が **1** に設定されている場合、サーバーはこの属性を無視します。

最大ログサイズを設定する場合は、ログファイルのローテーションにより作成可能なログファイルの合計数を考慮してください。また、Directory Server が管理する 3 つの異なるログファイル (アクセスログ、監査ログ、およびエラーログ) があり、それぞれがディスク領域を消費することに注意してください。これらの考慮事項を、アクセスログのディスク領域の合計量と比較します。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1 1 から最大 32 ビットの整数値 (2147483647)。値が -1 の場合は、ログファイルのサイズが無制限になります。
デフォルト値	100
構文	整数
例	nsslapd-accesslog-maxlogsize: 100

### 3.1.1.16. nsslapd-accesslog-maxlogsperdir(アクセスログの最大数)

この属性は、アクセスログが保存されるディレクトリーに格納できるアクセスログの合計数を設定します。アクセスログがローテーションされるたびに、新しいログファイルが作成されます。アクセスログディレクトリーに含まれるファイルの数がこの属性に保存されている値を超えると、ログファイルの最も古いバージョンが削除されます。パフォーマンス上の理由から、サーバーがログをローテーションせず、ログが無制限に大きくなるため、Red Hat はこの値を **1** に設定 **しない** ことを推奨します。

この属性の値が **1** よりも大きい場合は、**nsslapd-accesslog-logrotationtime** 属性をチェックして、ログローテーションが指定されているかどうかを確認します。**nsslapd-accesslog-logrotationtime** 属性の値が **-1** の場合、ログローテーションはありません。詳細は、「[nsslapd-accesslog-logrotationtime\(アクセスログローテーション時間\)](#)」を参照してください。

`nsslapd-accesslog-logminfreediskspace` および `nsslapd-accesslog-maxlogsize` に設定した値によっては、実際のログ数は `nsslapd-accesslog-maxlogspendir` で設定する数よりも少なくなる可能性があります。たとえば、`nsslapd-accesslog-maxlogspendir` がデフォルトの (10 ファイル) を使用し、`nsslapd-accesslog-logminfreediskspace` を 500 MB に、`nsslapd-accesslog-maxlogsize` を 100 MB に設定すると、Directory Server は 5 つのアクセスファイルのみを保持します。

パラメーター	説明
エントリー DN	cn=config
Valid Range	1 から最大 32 ビットの整数値 (2147483647)
デフォルト値	10
構文	整数
例	nsslapd-accesslog-maxlogspendir: 10

### 3.1.17. nsslapd-accesslog-mode(アクセスログファイルのパーミッション)

この属性は、アクセスログファイルが作成されるアクセスモードまたはファイルパーミッションを設定します。有効な値は、**000** から **777** の組み合わせ (番号または絶対 UNIX ファイルのパーミッション) です。値は、3桁の数字である必要があります。数字は **0** から **7** まで変わります。

- **0** - なし
- **1** - 実行のみ
- **2** - 書き込みのみ
- **3** - 書き込みおよび実行
- **4** - 読み取り専用
- **5** - 読み取りおよび実行
- **6** - 読み取りおよび書き込み
- **7** - 読み取り、書き込み、および実行

3桁の数値では、最初の数字は所有者の権限を表し、2つ目の数字はグループのパーミッションを表し、3つ目の数字は全員のパーミッションを表します。デフォルト値を変更する場合、**000** はログへのアクセスを許可せず、すべてのユーザーに書き込み権限を許可すると、ログが上書きまたは削除される可能性があることに注意してください。

新しく設定されたアクセスモードは、作成された新しいログにのみ影響します。モードは、ログが新しいファイルにローテーションするときに設定されます。

パラメーター	説明
エントリー DN	cn=config

パラメーター	説明
Valid Range	000 から 777 まで
デフォルト値	600
構文	整数
例	nsslapd-accesslog-mode: 600

### 3.1.1.18. nsslapd-allow-anonymous-access

バインド DN 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。

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。

匿名バインドを無効にしてセキュリティーを強化できます (オフ)。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。3 つ目のオプション **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。

必要に応じて、「[nsslapd-anonlimitsdn](#)」で説明されているように **nsslapd-anonlimitsdn** 属性を使用して、リソース制限を匿名バインドに配置できます。

この値の変更は、サーバーが再起動するまで反映されません。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off   rootdse
デフォルト値	on
構文	DirectoryString
例	nsslapd-allow-anonymous-access: on

### 3.1.1.19. nsslapd-allow-hashed-passwords

このパラメーターは、事前にハッシュ化されたパスワードチェックを無効にします。デフォルトでは、Directory Server では、事前にハッシュ化されたパスワードは Directory Manager 以外のユーザーによって設定できません。この権限を Password Administrators グループに追加すると、他のユーザーに

委任できます。ただし、レプリケーションパートナーが、事前にハッシュ化されたパスワードチェックをすでに制御している場合など、この機能は Directory Server で無効にする必要があります。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-allow-hashed-passwords: off

### 3.1.1.20. nsslapd-allow-unauthenticated-binds

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、セキュリティ上の理由から、このシナリオのアクセスを拒否します。



#### 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。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-allow-unauthenticated-binds: off

### 3.1.1.21. nsslapd-allowed-sasl-mechanisms

デフォルトでは、ルート DSE は SASL ライブラリーがサポートするすべてのメカニズムをリスト表示します。ただし、一部の環境では、特定の環境だけが優先されます。**nsslapd-allowed-sasl-mechanisms** 属性を使用すると、定義した SASL メカニズムのみを有効にできます。

メカニズム名は大文字、数字、およびアンダースコアで設定される必要があります。各メカニズムはコンマまたはスペースで区切ることができます。



### 注記

**EXTERNAL** メカニズムは SASL プラグインによって実際に使用されません。これはサーバーの内部であり、主に TLS クライアント認証に使用されます。したがって、**EXTERNAL** メカニズムは制限または制御できません。**nsslapd-allowed-sasl-mechanisms** 属性に設定されているかどうかに関わらず、常にサポートされているメカニズムリストに表示されます。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	有効な SASL メカニズム
デフォルト値	None(すべての SASL メカニズムが許可される)
構文	DirectoryString
例	nsslapd-allowed-sasl-mechanisms: GSSAPI, DIGEST-MD5, OTP

### 3.1.1.22. nsslapd-anonlimitsdn

リソース制限は、認証されたバインドに設定できます。リソース制限では、検索の単一の操作 (**nsslapd-sizeLimit**)、時間制限 (**nsslapd-timelimit**)、およびタイムアウト期間 (**nsslapd-idletimeout**)、ならびに検索可能なエントリーの合計数 (**nsslapd-lookthroughlimit**) で検索可能なエントリー数の上限を設定できます。このリソース制限により、サービス拒否攻撃がディレクトリーリソースを結合し、全体的なパフォーマンスを向上させることができます。

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、リソース制限が匿名操作には適用されません。

匿名バインドにリソース制限を設定するには、適切なリソース制限でテンプレートエントリーを作成できます。**nsslapd-anonlimitsdn** 設定属性を追加して、このエントリーを指定し、リソース制限を匿名バインドに適用できます。

パラメーター	説明
エントリー DN	cn=config
有効な値	任意の DN

パラメーター	説明
デフォルト値	なし
構文	DirectoryString
例	nsslapd-anonlimitsdn: cn=anon template,ou=people,dc=example,dc=com

### 3.1.1.23. nsslapd-attribute-name-exceptions

この属性を使用すると、属性名の標準以外の文字を、スキーマ定義属性の "\_" など、古いサーバーと後方互換性に使用できます。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-attribute-name-exceptions: on

### 3.1.1.24. nsslapd-auditlog(監査ログ)

この属性は、各データベースに加えられた変更を記録するために使用されるログのパスおよびファイル名を設定します。

パラメーター	説明
エントリー DN	cn=config
有効な値	有効なファイル名
デフォルト値	/var/log/dirsrv/slapd- <b>instance</b> /audit
構文	DirectoryString
例	nsslapd-auditlog: /var/log/dirsrv/slapd- <b>instance</b> /audit

監査ロギングを有効にするには、この属性に有効なパスとパラメーターが必要で、**nsslapd-auditlog-logging-enabled** 設定属性を **on** に切り替える必要があります。以下の表は、これらの2つの設定属性

と、監査ロギングを無効化または有効にするという 2 つの設定属性に対して可能な 4 つの値の組み合わせを示しています。

表3.3 nsslapd-auditlog の可能な組み合わせ

dse.ldif の属性	値	ロギングの有効化または無効化
nsslapd-auditlog-logging-enabled	on	Disabled
nsslapd-auditlog	空の文字列	
nsslapd-auditlog-logging-enabled	on	有効
nsslapd-auditlog	filename	
nsslapd-auditlog-logging-enabled	off	Disabled
nsslapd-auditlog	空の文字列	
nsslapd-auditlog-logging-enabled	off	Disabled
nsslapd-auditlog	filename	

### 3.1.1.25. nsslapd-auditlog-display-attrs

**nsslapd-auditlog-display-attrs** 属性を使用すると、Directory Server が監査ログに表示する属性を設定して、変更されるエントリーに関する有用な識別情報を提供できます。監査ログに属性を追加すると、エントリー内の特定の属性の現在の状態とエントリーの更新の詳細を確認できます。

次のオプションのいずれかを選択して、ログ内の属性を表示できます。

- Directory Server が変更するエントリーの特定の属性を表示するには、属性名を値として指定します。
- 複数の属性を表示するには、スペースで区切られた属性名のリストを値として指定します。
- エントリーのすべての属性を表示するには、値としてアスタリスク (\*) を使用します。

Directory Server が監査ログに表示する必要がある属性のスペース区切りのリストを指定するか、値としてアスタリスク (\*) を使用して、変更されるエントリーのすべての属性を表示します。

たとえば、監査ログ出力に **cn** 属性を追加するとします。**nsslapd-auditlog-display-attrs** 属性を **cn** に設定すると、監査ログに次の出力が表示されます。

```
time: 20221027102743
dn: uid=73747737483,ou=people,dc=example,dc=com
#cn: Frank Lee
result: 0
changetype: modify
replace: description
description: Adds cn attribute to the audit log
-
replace: modifiersname
modifiersname: cn=dm
```

```
-
replace: modifytimestamp
modifytimestamp: 20221027142743Z
```

パラメーター	説明
エントリー DN	cn=config
有効な値	有効な属性名。監査ログ内のエントリーのすべての属性を表示する場合は、アスタリスク (*) を使用します。
デフォルト値	なし
構文	DirectoryString
例	nsslapd-auditlog-display-attrs: cn ou

### 3.1.1.26. nsslapd-auditlog-list

監査ログファイルのリストを提供します。

パラメーター	説明
エントリー DN	cn=config
有効な値	
デフォルト値	なし
構文	DirectoryString
例	nsslapd-auditlog-list: auditlog2,auditlog3

### 3.1.1.27. nsslapd-auditlog-logexpirationtime(監査ログの有効期限)

この属性は、ログファイルが削除される前に許可される最大期間を設定します。この属性はユニット数のみを提供します。単位 (day、week、month など) は **nsslapd-auditlog-logexpirationtimeunit** 属性で指定します。

パラメーター	説明
エントリー DN	cn=config



パラメーター	説明
Valid Range	-1から最大 32 ビットの整数値 (2147483647)  -1または0の値は、ログが期限切れになることはありません。
デフォルト値	-1
構文	整数
例	nsslapd-auditlog-logexpirationtime: 1

### 3.1.1.28. nsslapd-auditlog-logexpirationtimeunit(監査ログの有効期限時間単位)

この属性は、**nsslapd-auditlog-logexpirationtime** 属性の単位を設定します。ユニットがサーバーによって認識されていない場合は、ログが期限切れになることはありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	month   week   day
デフォルト値	週
構文	DirectoryString
例	nsslapd-auditlog-logexpirationtimeunit: day

### 3.1.1.29. nsslapd-auditlog-logging-enabled(監査ログの有効化)

監査ロギングをオンおよびオフにします。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-auditlog-logging-enabled: off

監査ロギングを有効にするには、この属性に有効なパスとパラメーターが必要で、ns **slapd-auditlog-logging-enabled** 設定属性を **on** に切り替える必要があります。以下の表は、これらの 2 つの設定属性と、監査ロギングを無効化または有効にするという 2 つの設定属性に対して可能な 4 つの値の組み合わせを示しています。

表3.4 nsslapd-auditlog と nsslapd-auditlog-logging-enabled の組み合わせ

属性	値	ロギングの有効化または無効化
nsslapd-auditlog-logging-enabled	on	Disabled
nsslapd-auditlog	空の文字列	
nsslapd-auditlog-logging-enabled	on	有効
nsslapd-auditlog	filename	
nsslapd-auditlog-logging-enabled	off	Disabled
nsslapd-auditlog	空の文字列	
nsslapd-auditlog-logging-enabled	off	Disabled
nsslapd-auditlog	filename	

### 3.1.1.30. nsslapd-auditlog-logmaxdiskpace(監査ログの最大ディスク領域)

この属性は、監査ログが消費できる最大ディスク容量をメガバイト単位で設定します。この値を超えると、最も古い監査ログが削除されます。

最大ディスク容量を設定する場合は、ログファイルのローテーションにより作成可能なログファイルの合計数を考慮してください。また、Directory Server が維持する 3 つの異なるログファイル (アクセスログ、監査ログ、およびエラーログ) があり、それぞれがディスク領域を消費することに注意してください。これらの考慮事項を、監査ログのディスク領域の合計量と比較します。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1 1 から最大 32 ビットの整数値 (2147483647)。値が -1 の場合は、監査ログに許可されるディスク領域のサイズが無制限であることを意味します。
デフォルト値	-1
構文	整数
例	nsslapd-auditlog-logmaxdiskpace: 10000

### 3.1.1.31. nsslapd-auditlog-logminfreediskpace(監査ログの最小ディスク領域)

この属性は、許容できる最小ディスク容量をメガバイト単位で設定します。空きディスク容量がこの属性によって指定された値を下回ると、この属性を満たすために十分なディスク領域が解放されるまで、最も古い監査ログが削除されます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1(無制限)   1から 32 ビットの整数値 (2147483647)
デフォルト値	-1
構文	整数
例	nsslapd-auditlog-logminfreediskspace: -1

### 3.1.1.32. nsslapd-auditlog-logrotationsync-enabled(監査ログローテーション同期の有効化)

この属性は、監査ログのローテーションが特定の時刻と同期するかどうかを設定します。この方法でログローテーションを同期すると、毎日深夜から深夜までなど、1日の指定された時間にログファイルを生成できます。これにより、ログファイルがカレンダーに直接マップされるため、ログファイルの分析がはるかに簡単になります。

監査ロギングのローテーションを時刻で同期するには、この属性を **nsslapd-auditlog-logrotationsynchour** 属性値および **nsslapd-auditlog-logrotationsyncmin** 属性値をログファイルのローテーションの時間および分に設定して、この属性を有効にする必要があります。

たとえば、毎日深夜に監査ログファイルをローテーションするには、その値を **on** に設定してこの属性を有効にしてから、**nsslapd-auditlog-logrotationsynchour** 属性および **nsslapd-auditlog-logrotationsyncmin** 属性の値を **0** に設定します。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-auditlog-logrotationsync-enabled: on

### 3.1.1.33. nsslapd-auditlog-logrotationsynchour(監査ログローテーション同期時間)

この属性は、監査ログのローテーションを行う時刻を設定します。この属性は、**nsslapd-auditlog-logrotationsync-enabled** 属性および **nsslapd-auditlog-logrotationsyncmin** 属性と共に使用する必要があります。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 - 23
デフォルト値	なし ( <b>nsslapd-auditlog-logrotationsync-enabled</b> がオフであるため)
構文	整数
例	nsslapd-auditlog-logrotationsynchour: 23

### 3.1.1.34. nsslapd-auditlog-logrotationsyncmin(監査ログローテーション同期数)

この属性は、監査ログのローテーションに使用する日数を設定します。この属性は、**nsslapd-auditlog-logrotationsync-enabled** 属性および **nsslapd-auditlog-logrotationsynchour** 属性と共に使用する必要があります。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 - 59
デフォルト値	なし ( <b>nsslapd-auditlog-logrotationsync-enabled</b> がオフであるため)
構文	整数
例	nsslapd-auditlog-logrotationsyncmin: 30

### 3.1.1.35. nsslapd-auditlog-logrotationtime(監査ログローテーション時間)

この属性は、監査ログファイルのローテーションの間隔を設定します。この属性はユニット数のみを提供します。単位 (day、week、month など) は **nsslapd-auditlog-logrotationtimeunit** 属性で指定します。**nsslapd-auditlog-maxlogsperdir** 属性が 1 に設定されていると、サーバーはこの属性を無視します。

Directory Server は、ログのサイズにかかわらず、設定された間隔が過ぎた後の最初の書き込み操作でログをローテーションします。

パフォーマンス上の理由から、ログローテーションを指定しないことは推奨しませんが、ログは無期限に大きくなるため、これを指定する方法は 2 つあります。**nsslapd-auditlog-maxlogsperdir** 属性値を 1 に設定するか、**nsslapd-auditlog-logrotationtime** 属性を -1 に設定します。サーバーは最初に **nsslapd-auditlog-maxlogsperdir** 属性をチェックして、この属性の値が 1 よりも大きい場合、サーバーは **nsslapd-auditlog-logrotationtime** 属性をチェックします。詳細は、「[nsslapd-auditlog-maxlogsperdir\(監査ログの最大数\)](#)」を参照してください。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1 1 から最大 32 ビットの整数値 (2147483647)。値が -1 の場合は、監査ログファイルのローテーションの間隔が無制限になります。
デフォルト値	1
構文	整数
例	nsslapd-auditlog-logrotationtime: 100

### 3.1.1.36. nsslapd-auditlog-logrotationtimeunit(監査ログローテーション時間単位)

この属性は、**nsslapd-auditlog-logrotationtime** 属性の単位を設定します。

パラメーター	説明
エントリー DN	cn=config
有効な値	month   week   day   hour   minute
デフォルト値	週
構文	DirectoryString
例	nsslapd-auditlog-logrotationtimeunit: day

### 3.1.1.37. nsslapd-auditlog-maxlogsize(監査ログの最大サイズ)

この属性は、最大監査ログサイズをメガバイト単位で設定します。この値に達すると、監査ログはローテーションされます。つまり、サーバーがログ情報を新しいログファイルに書き込むことを意味します。**nsslapd-auditlog-maxlogspedir** を 1 にすると、サーバーはこの属性を無視します。

最大ログサイズを設定する場合は、ログファイルのローテーションにより作成可能なログファイルの合計数を考慮してください。また、Directory Server が管理する 3 つの異なるログファイル (アクセスログ、監査ログ、およびエラーログ) があり、それぞれがディスク領域を消費することに注意してください。これらの考慮事項を、監査ログのディスク領域の合計量と比較します。

パラメーター	説明
エントリー DN	cn=config

パラメーター	説明
Valid Range	-1 1から最大 32 ビットの整数値 (2147483647)。値が -1 の場合は、ログファイルのサイズが無制限になります。
デフォルト値	100
構文	整数
例	nsslapd-auditlog-maxlogsize: 50

### 3.1.1.38. nsslapd-auditlog-maxlogspersdir(監査ログの最大数)

この属性は、監査ログが保存されるディレクトリーに格納できる監査ログの合計数を設定します。監査ログがローテーションされるたびに、新しいログファイルが作成されます。監査ログディレクトリーに含まれるファイルの数がこの属性に保存されている値を超えると、ログファイルの最も古いバージョンは削除されます。デフォルトは 1 ログです。このデフォルト値が許可されると、サーバーはログをローテーションせず、無制限に増加します。

この属性の値が 1 よりも大きい場合は、**nsslapd-auditlog-logrotationtime** 属性をチェックして、ログローテーションが指定されているかどうかを確認します。**nsslapd-auditlog-logrotationtime** 属性の値が -1 の場合、ログローテーションはありません。詳細は、「[nsslapd-auditlog-logrotationtime\(監査ログローテーション時間\)](#)」を参照してください。

パラメーター	説明
エントリー DN	cn=config
Valid Range	1 から最大 32 ビットの整数値 (2147483647)
デフォルト値	1
構文	整数
例	nsslapd-auditlog-maxlogspersdir: 10

### 3.1.1.39. nsslapd-auditlog-mode(監査ログファイルのパーミッション)

この属性は、監査ログファイルが作成されるアクセスモードまたはファイルパーミッションを設定します。有効な値は、番号がミラーリングされるか、絶対 UNIX ファイルのパーミッションになるため、**000** から **777** の組み合わせです。値は、3 桁の数字の組み合わせである必要があります。数字は **0** から **7** によって異なります。

- 0 - なし
- 1 - 実行のみ
- 2 - 書き込みのみ

- 3 - 書き込みおよび実行
- 4 - 読み取り専用
- 5 - 読み取りおよび実行
- 6 - 読み取りおよび書き込み
- 7 - 読み取り、書き込み、および実行

3桁の数値では、最初の数字は所有者の権限を表し、2つ目の数字はグループのパーミッションを表し、3つ目の数字は全員のパーミッションを表します。デフォルト値を変更する場合、**000**はログへのアクセスを許可せず、すべてのユーザーに書き込み権限を許可すると、ログが上書きまたは削除される可能性があることに注意してください。

新しく設定されたアクセスモードは、作成された新しいログにのみ影響します。モードは、ログが新しいファイルにローテーションするときに設定されます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	000 から 777 まで
デフォルト値	600
構文	整数
例	nsslapd-auditlog-mode: 600

#### 3.1.1.40. nsslapd-auditfaillog(送信失敗ログ)

この属性は、失敗した LDAP の変更を記録するために使用されるログのパスおよびファイル名を設定します。

**nsslapd-auditfaillog-logging-enabled** が有効にされており、**nsslapd-auditfaillog** が設定されていない場合、監査の失敗イベントは **nsslapd-auditlog** で指定されたファイルに記録されます。

**nsslapd-auditfaillog** パラメーターを **nsslapd-auditlog** と同じパスに設定すると、いずれも同じファイルに記録されます。

パラメーター	説明
エントリー DN	cn=config
有効な値	有効なファイル名
デフォルト値	/var/log/dirsrv/slapd- <b>instance</b> /audit
構文	DirectoryString

パラメーター	説明
例	nsslapd-auditfaillog: /var/log/dirsrv/slapd- <b>instance</b> /audit

監査の失敗ログを有効にするには、この属性に有効なパスが必要で、**nsslapd-auditfaillog-logging-enabled** 属性を **on** に設定する必要があります。

### 3.1.1.41. nsslapd-auditfaillog-list

監査失敗のログファイルのリストを提供します。

パラメーター	説明
エントリー DN	cn=config
有効な値	
デフォルト値	なし
構文	DirectoryString
例	nsslapd-auditfaillog-list: auditfaillog2,auditfaillog3

### 3.1.1.42. nsslapd-auditfaillog-logexpirationtime(監査ログの有効期限)

この属性は、削除前のログファイルの最大期間を設定します。ユニット数に提供されます。**nsslapd-auditfaillog-logexpirationtimeunit** 属性の day、week、month など、単位を指定します。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1 から最大 32 ビットの整数値 (2147483647)  -1 または 0 の値は、ログが期限切れになることはありません。
デフォルト値	-1
構文	整数
例	nsslapd-auditfaillog-logexpirationtime: 1

### 3.1.1.43. nsslapd-auditfaillog-logexpirationtimeunit(Audit Fail Log Expiration Time Unit)



この属性は、**nsslapd-auditfaillog-logexpirationtime** 属性に単位を設定します。ユニットがサーバーによって認識されていない場合は、ログが期限切れになることはありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	month   week   day
デフォルト値	週
構文	DirectoryString
例	nsslapd-auditfaillog-logexpirationtimeunit: day

#### 3.1.1.44. nsslapd-auditfaillog-logging-enabled(Audit Fail Log Enable Logging)

失敗した LDAP 変更のロギングをオンまたはオフにします。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-auditfaillog-logging-enabled: off

#### 3.1.1.45. nsslapd-auditfaillog-logmaxdiskspace(監査ログの最大ディスク領域)

この属性は、監査ログが消費できる最大ディスク容量をメガバイト単位で設定します。サイズが制限を超えると、最も古い監査ログが削除されます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1   1 から最大 32 ビットの整数値 (2147483647)。値が -1 の場合は、監査ログに許可されるディスク領域のサイズが無制限であることを意味します。
デフォルト値	100
構文	整数

パラメーター	説明
例	nsslapd-auditfaillog-logmaxdiskpace: 10000

### 3.1.1.46. nsslapd-auditfaillog-logminfreediskpace(監査失敗ログ最小空きディスク容量)

この属性は、許容できる最小ディスク容量をメガバイト単位で設定します。空きディスク容量が指定された値よりも小さい場合、十分なディスク領域が解放されるまで最も古い監査ログが削除されます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1(無制限)   1から 32 ビットの整数値 (2147483647)
デフォルト値	-1
構文	整数
例	nsslapd-auditfaillog-logminfreediskpace: -1

### 3.1.1.47. nsslapd-auditfaillog-logrotationsync-enabled(監査失敗ログローテーション同期有効)

この属性は、監査失敗のログローテーションが、特定の日に同期されるかどうかを設定します。この方法でログローテーションを同期すると、毎日深夜から深夜までなど、1日の指定された時間にログファイルを生成できます。これにより、ログファイルがカレンダーに直接マップされるため、ログファイルの分析ははるかに簡単になります。

監査失敗ロギングのローテーションを時刻で同期するには、この属性を **nsslapd-auditfaillog-logrotationsynchour** 属性値および **nsslapd-auditfaillog-logrotationsyncmin** 属性値をログファイルのローテーションの時間および分に設定して、この属性を有効にする必要があります。

たとえば、毎日深夜に監査失敗ログファイルをローテーションするには、その値を **on** に設定してこの属性を有効にしてから、**nsslapd-auditfaillog-logrotationsynchour** 属性および **nsslapd-auditfaillog-logrotationsyncmin** 属性の値を **0** に設定します。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-auditfaillog-logrotationsync-enabled: on

### 3.1.1.48. nsslapd-auditfaillog-logrotationsynchour(監査ログローテーション同期時間)

この属性は、監査ログがローテーションされる時刻を設定します。この属性は、**nsslapd-auditfaillog-logrotationsync-enabled** 属性および **nsslapd-auditfaillog-logrotationsyncmin** 属性と共に使用する必要があります。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 - 23
デフォルト値	なし ( <b>nsslapd-auditfaillog-logrotationsync-enabled</b> がオフであるため)
構文	整数
例	nsslapd-auditfaillog-logrotationsynchour: 23

### 3.1.1.49. nsslapd-auditfaillog-logrotationsyncmin(監査ログローテーション同期数)

この属性は、監査ログがローテーションされる分を設定します。この属性は、**nsslapd-auditfaillog-logrotationsync-enabled** 属性および **nsslapd-auditfaillog-logrotationsynchour** 属性と共に使用する必要があります。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 - 59
デフォルト値	なし ( <b>nsslapd-auditfaillog-logrotationsync-enabled</b> がオフであるため)
構文	整数
例	nsslapd-auditfaillog-logrotationsyncmin: 30

### 3.1.1.50. nsslapd-auditfaillog-logrotationtime(監査ログローテーション時間)

この属性は、監査失敗ログファイルのローテーションの間隔を設定します。この属性はユニット数のみを提供します。単位 (day、week、month など) は **nsslapd-auditfaillog-logrotationtimeunit** 属性で指定します。**nsslapd-auditfaillog-maxlogspersdir** 属性が 1 に設定されている場合、サーバーはこの属性を無視します。

Directory Server は、ログのサイズにかかわらず、設定された間隔が過ぎた後の最初の書き込み操作でログをローテーションします。

パフォーマンス上の理由から、ログローテーションを指定しないことは推奨しませんが、ログは無期限に大きくなるため、これを指定する方法は 2 つあります。**nsslapd-auditfaillog-maxlogspersdir** 属性値

を 1 に設定するか、`nsslapd-auditfaillog-logrotationtime` 属性を -1 に設定します。サーバーは最初に `nsslapd-auditfaillog-maxlogsperdir` 属性をチェックして、この属性の値が 1 を超える場合は、サーバーでは `nsslapd-auditfaillog-logrotationtime` 属性を確認します。詳細は、「[nsslapd-auditfaillog-maxlogsperdir\(監査ログの最大数\)](#)」を参照してください。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1 1 から最大 32 ビットの整数値 (2147483647)。値が -1 の場合は、監査失敗ログファイルのローテーションの間隔が無制限になります。
デフォルト値	1
構文	整数
例	nsslapd-auditfaillog-logrotationtime: 100

### 3.1.1.51. nsslapd-auditfaillog-logrotationtimeunit(監査失敗ログローテーション時間単位)

この属性は、`nsslapd-auditfaillog-logrotationtime` 属性の単位を設定します。

パラメーター	説明
エントリー DN	cn=config
有効な値	month   week   day   hour   minute
デフォルト値	週
構文	DirectoryString
例	nsslapd-auditfaillog-logrotationtimeunit: day

### 3.1.1.52. nsslapd-auditfaillog-maxlogsize(監査ログの最大サイズ)

この属性は、最大監査失敗ログサイズをメガバイト単位で設定します。この値に達すると、監査ログはローテーションされます。つまり、サーバーがログ情報を新しいログファイルに書き込むことを意味します。`nsslapd-auditfaillog-maxlogsperdir` パラメーターが 1 に設定されている場合、サーバーはこの属性を無視します。

パラメーター	説明
エントリー DN	cn=config

パラメーター	説明
Valid Range	-1 1から最大 32 ビットの整数値 (2147483647)。値が -1 の場合は、ログファイルのサイズが無制限になります。
デフォルト値	100
構文	整数
例	nsslapd-auditfaillog-maxlogsize: 50

### 3.1.1.53. nsslapd-auditfaillog-maxlogspdir(監査ログの最大数)

この属性は、監査ログが保存されるディレクトリーに格納できる監査ログの合計数を設定します。監査失敗のログがローテーションされるたびに、新しいログファイルが作成されます。監査ログディレクトリーに含まれるファイルの数がこの属性に保存されている値を超えると、ログファイルの最も古いバージョンは削除されます。デフォルトは 1 ログです。このデフォルト値が許可されると、サーバーはログをローテーションせず、無制限に増加します。

この属性の値が 1 よりも大きい場合は、**nsslapd-auditfaillog-logrotationtime** 属性をチェックして、ログローテーションが指定されているかどうかを設定します。**nsslapd-auditfaillog-logrotationtime** 属性の値が -1 の場合は、ログローテーションがありません。詳細は、[「nsslapd-auditfaillog-logrotationtime\(監査ログローテーション時間\)」](#) を参照してください。

パラメーター	説明
エントリー DN	cn=config
Valid Range	1 から最大 32 ビットの整数値 (2147483647)
デフォルト値	1
構文	整数
例	nsslapd-auditfaillog-maxlogspdir: 10

### 3.1.1.54. nsslapd-auditfaillog-mode(監査失敗ログファイルパーミッション)

この属性は、監査失敗ログファイルが作成されるアクセスモードまたはファイルパーミッションを設定します。有効な値は、番号がミラーリングされるか、絶対 UNIX ファイルのパーミッションになるため、**000** から **777** の組み合わせです。値は、3桁の数字の組み合わせである必要があります。数字は **0** から **7** によって異なります。

- 0 - なし
- 1 - 実行のみ

- 2 - 書き込みのみ
- 3 - 書き込みおよび実行
- 4 - 読み取り専用
- 5 - 読み取りおよび実行
- 6 - 読み取りおよび書き込み
- 7 - 読み取り、書き込み、および実行

3桁の数値では、最初の数字は所有者の権限を表し、2つ目の数字はグループのパーミッションを表し、3つ目の数字は全員のパーミッションを表します。デフォルト値を変更する場合、**000**はログへのアクセスを許可せず、すべてのユーザーに書き込み権限を許可すると、ログが上書きまたは削除される可能性があることに注意してください。

新しく設定されたアクセスモードは、作成された新しいログにのみ影響します。モードは、ログが新しいファイルにローテーションするときに設定されます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	000 から 777 まで
デフォルト値	600
構文	整数
例	nsslapd-auditfaillog-mode: 600

### 3.1.1.55. nsslapd-bakdir(デフォルトのバックアップディレクトリー)

このパラメーターは、デフォルトのバックアップディレクトリーへのパスを設定します。Directory Server ユーザーには、設定されたディレクトリーに書き込みパーミッションが必要です。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	任意のローカルディレクトリーパス。
デフォルト値	/var/lib/dirsrv/slapd- <b>instance</b> /bak
構文	DirectoryString
例	nsslapd-bakdir: /var/lib/dirsrv/slapd- <b>instance</b> /bak

### 3.1.1.56. nsslapd-certdir(証明書およびキーデータベースディレクトリー)

このパラメーターは、Directory Server がインスタンスの Network Security Services(NSS) データベースを保存するために使用するディレクトリーへの完全パスを定義します。このデータベースには、インスタンスの秘密鍵と証明書が含まれます。

フォールバックとして Directory Server は、秘密鍵と証明書をこのディレクトリーに抽出します。サーバーがプライベート名前空間の **/tmp/** ディレクトリーに抽出できません。プライベート名スペースの詳細は、**systemd.exec(5)** の man ページの **PrivateTmp** パラメーターの説明を参照してください。

**nsslapd-certdir** で指定したディレクトリーはサーバーのユーザー ID で所有され、このユーザー ID のみがこのディレクトリーに読み取り/書き込みパーミッションを持っている必要があります。セキュリティー上の理由から、他のユーザーには、このディレクトリーに読み書きするパーミッションがありません。

この属性への変更を反映するには、サービスを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=config
有効な値	絶対パス
デフォルト値	/etc/dirsrv/slapd- <b>instance_name</b> /
構文	DirectoryString
例	nsslapd-certdir: /etc/dirsrv/slapd- <b>instance_name</b> /

### 3.1.1.57. nsslapd-certmap-basedn(証明書マップ検索ベース)

この属性は、**/etc/dirsrv/slapd-*instance\_name*/certmap.conf** ファイルで設定される security サブシステム証明書マッピングの制限を回避するために、TLS 証明書を使用してクライアント認証を実行する場合に使用できます。このファイルの設定によっては、証明書マッピングは、ルート DN に基づいてディレクトリーサブツリー検索を使用して実行できます。検索がルート DN をベースとする場合、**nsslapd-certmap-basedn** 属性は、ルート以外のエントリーに基づいて検索を強制的に実行する可能性があります。この属性の有効な値は、証明書マッピングに使用する接尾辞またはサブツリーの DN です。

パラメーター	説明
エントリー DN	cn=config
有効な値	任意の有効な DN
デフォルト値	
構文	DirectoryString
例	nsslapd-certmap-basedn: ou=People,dc=example,dc=com

### 3.1.1.58. nsslapd-config

この読み取り専用属性は設定 DN です。

パラメーター	説明
エントリー DN	cn=config
有効な値	任意の有効な設定 DN
デフォルト値	
構文	DirectoryString
例	nsslapd-config: cn=config

### 3.1.1.59. nsslapd-cn-uses-dn-syntax-in-dns

このパラメーターを使用すると、CNF 値内で DN を有効にできます。

Directory Server の DN ノーマライザーは [RFC4514](#) に従い、RDN 属性タイプが DN 構文にベースでない場合は空白を保持します。ただし、Directory Server の設定エントリーは、**cn** 属性を使用して DN 値を保存することがあります。たとえば、**dn: cn="dc=A,dc=com", cn=mapping tree,cn=config** の場合は、DN 構文に従って **cn** を正規化する必要があります。

この設定が必要な場合は、**nsslapd-cn-uses-dn-syntax-in-dns** パラメーターを有効にします。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-cn-uses-dn-syntax-in-dns: off

### 3.1.1.60. nsslapd-connection-buffer: 1

この属性は、接続バッファの動作を設定します。値:

- **0:** バッファを無効にします。PDU(単一のプロトコルデータユニット)のみが一度に読み込まれます。
- **1:** 512 バイトの通常の固定サイズ **LDAP\_SOCKET\_IO\_BUFFER\_SIZE**。
- **2:** 適応可能なバッファサイズ



値が **2** の場合は、クライアントが大量のデータを一度に送信する場合にパフォーマンスが向上します。たとえば、大規模な追加や変更操作の場合や、多くの非同期リクエスト数がレプリケーション中に単一の接続で受信される場合などがこれに該当します。

パラメーター	説明
エントリー DN	cn=config
有効な値	0   1   2
デフォルト値	1
構文	整数
例	nsslapd-connection-buffer: 1

### 3.1.1.61. nsslapd-connection-nocanon

このオプションを使用すると、SASL **NOCANON** フラグを有効または無効にできます。無効にすると、Directory Server は、送信接続の DNS 逆引きエントリーを検索しないようにします。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-connection-nocanon: on

### 3.1.1.62. nsslapd-conntablesize

この属性は接続テーブルサイズを設定し、サーバーによってサポートされる接続の総数を決定します。

接続スロットが不足しているため、Directory Server が接続を拒否する場合は、この属性の値を増やします。この状況が発生すると、Directory Server のエラーログファイルは、**Not listening for new connections — too many fds open** メッセージを記録します。

オープンファイルの数とプロセスごとのオープンファイル数の上限を増やす必要がある場合があります。Directory Server を起動するシェルでオープンファイルの数 (**ulimit -n**) の **ulimit** を増やす必要があります。

接続テーブルのサイズは、**nsslapd-maxdescriptor** で上限です。詳細は、「[nsslapd-maxdescriptors\(最大ファイル記述子\)](#)」を参照してください。

この属性の変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=config
Valid Values	オペレーティングシステムに依存します
Default Value	Directory Server プロセスが開くことのできるファイルの最大数。 <b>getdtablesize()</b> <b>glibc</b> 関数を参照してください。
Syntax	整数
Example	nsslapd-conntablesizesize: 4093

### 3.1.1.63. nsslapd-counters

**nsslapd-counters** 属性は、Directory Server データベースおよびサーバーパフォーマンスカウンターを有効および無効にします。

大きなカウンターを追跡すると、パフォーマンスに影響する可能性があります。カウンターの 64 ビットの整数をオフにすると、パフォーマンスが最小限に抑えられますが、長期統計追跡に悪影響を及ぼします。

このパラメーターは、デフォルトで有効になっています。カウンターを無効にするには、Directory Server を停止し、直接 **dse.ldif** ファイルを編集し、サーバーを再起動します。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-counters: on

### 3.1.1.64. nsslapd-csnlogging

この属性は、利用可能な場合は変更シーケンス番号 (CSN) がアクセスログに記録されるかどうかを設定します。デフォルトでは、CSN ロギングがオンになっています。

パラメーター	説明
エントリー DN	cn=config

パラメーター	説明
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-csnlogging: on

### 3.1.1.65. nsslapd-defaultnamingcontext

この属性は、クライアントがデフォルトで検索ベースとして使用する、設定されたすべての命名コンテキストを示します。この値は、**defaultNamingContext** 属性としてルート DSE にコピーされます。これにより、クライアントはルート DSE にクエリーを実行してコンテキストを取得し、適切なベースで検索を開始できます。

パラメーター	説明
エントリー DN	cn=config
有効な値	ルート接尾辞 DN
デフォルト値	デフォルトのユーザー接尾辞
構文	DN
例	nsslapd-defaultnamingcontext: dc=example,dc=com

### 3.1.1.66. nsslapd-disk-monitoring

この属性は、ディスクで利用可能なディスク領域を確認するか、Directory Server データベースが実行している場所をマウントするために 10 秒ごとに実行されるスレッドを有効にします。利用可能なディスク領域が設定されたしきい値を下回ると、サーバーはロギングレベルの削減、アクセスまたは監査ログの無効化、ローテーションされたログの削除を行います。利用可能な領域が十分でない場合は、サーバーは正常にシャットダウンします (ウェイアと猶予期間後)。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString

パラメーター	説明
例	nsslapd-disk-monitoring: on

### 3.1.1.67. nsslapd-disk-monitoring-grace-period

「[nsslapd-disk-monitoring-threshold](#)」に設定されたディスク領域制限の半分に達すると、サーバーをシャットダウンするまで待機する猶予期間を設定します。これにより、管理者がディスクをクリーンアップし、シャットダウンを防ぐことができます。

パラメーター	説明
エントリー DN	cn=config
有効な値	任意の整数値 (分単位)
デフォルト値	60
構文	整数
例	nsslapd-disk-monitoring-grace-period: 45

### 3.1.1.68. nsslapd-disk-monitoring-logging-critical

ログディレクトリーがディスク領域の制限「[nsslapd-disk-monitoring-threshold](#)」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。

これを有効にすると、ロギングは無効ではなく、サーバーによるディスク使用量を減らす手段としてローテーションされたログは削除されません。サーバーは単にシャットダウンプロセスを実行します。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-disk-monitoring-logging-critical: on

### 3.1.1.69. nsslapd-disk-monitoring-readonly-on-threshold

空きディスク領域が **nsslapd-disk-monitoring-threshold** パラメーターに設定した値の半分に達すると、Directory Server は、**nsslapd-disk-monitoring-grace-period** に設定された猶予期間後にインスタ

ンスをシャットダウンします。ただし、インスタンスが停止する前にディスクの容量が不足すると、データが破損する可能性があります。この問題を回避するには、しきい値に達した場合に **nsslapd-disk-monitoring-readonly-on-threshold** パラメーターを有効にします。Directory Server は、しきい値に達したときにインスタンスを読み取り専用モードに設定します。



### 重要

この設定では、空きディスク領域が **nsslapd-disk-monitoring-threshold** で設定したしきい値の半分を下回ると、Directory Server が起動しません。

この属性への変更を反映するには、サービスを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-disk-monitoring-readonly-on-threshold: off

#### 3.1.1.70. 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しきい値をバイト単位で設定します。領域がこのしきい値の半分に達すると、サーバーはシャットダウンプロセスを開始します。

たとえば、しきい値が 2MB(デフォルト) の場合、利用可能なディスク領域が 1MB になると、サーバーはシャットダウンを開始します。

デフォルトでは、しきい値は Directory Server インスタンスの設定、トランザクション、およびデータベースディレクトリーによって使用されるディスク領域に対して評価されます。[「nsslapd-disk-monitoring-logging-critical」](#) 属性が有効な場合は、ログディレクトリーが評価に含まれます。

パラメーター	説明
エントリー DN	cn=config
有効な値	* 32 ビットシステムの 0 から 32 ビットの整数値 (2147483647)  * 64 ビットシステムの 0 から 64 ビットの整数値 (9223372036854775807)
デフォルト値	2000000 (2MB)
構文	DirectoryString

パラメーター	説明
例	nsslapd-disk-monitoring-threshold: 2000000

### 3.1.1.71. nsslapd-dn-validate-strict

「[nsslapd-syntaxcheck](#)」属性により、サーバーは新規または変更された属性値がその属性に必要な構文と一致することを確認できます。

ただし、DN の構文ルールでは、厳格さが増大しています。[RFC 4514](#) で DN 構文ルールを適用しようとすると、古い構文定義を使用して多くのサーバーが破損する可能性があります。デフォルトでは、[nsslapd-syntaxcheck](#) は [RFC 1779](#) または [RFC 2253](#) を使用して DN を検証します。

[nsslapd-dn-validate-strict](#) 属性は、[RFC 4514](#) のセクション 3 に従って、DN の厳密な構文検証を明示的に有効にします。この属性を **off** (デフォルト) に設定すると、サーバーは、構文違反があるかどうかをチェックする前に値を正規化します。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-dn-validate-strict: off

### 3.1.1.72. nsslapd-ds4-compatible-schema

[cn=schema](#) のスキーマを Directory Server の 4.x バージョンと互換性を持たせるようにします。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-ds4-compatible-schema: off

### 3.1.1.73. nsslapd-enable-turbo-mode

Directory Server の Turbo モードは、ワーカースレッドを接続専用にし、その接続からの受信操作を継続的に読み取ることができる機能です。これにより、非常にアクティブな接続でパフォーマンスを向上でき、この機能はデフォルトで有効になります。

ワーカースレッドは、サーバーによって受信される LDAP 操作を処理します。ワーカースレッドの数は **nsslapd-threadnumber** パラメーターで定義されます。各ワーカースレッドは、現在の接続のアクティビティーレベルが、確立されたすべての接続間で最大1つであるかどうかを評価します。Directory Server は、最後のチェック以降に開始される操作の数としてアクティビティーを測定し、現在の接続の動作が最も高い場合は turbo モードでワーカースレッドを切り替えます。

1秒以上など、バインド操作の実行時間が長い (ログファイルの **etime** 値) 場合は、ターボモードを無効にするとパフォーマンスが向上する可能性があります。ただし、場合によっては、バインド時間がネットワークやハードウェアの問題の現象となる場合があります。このような状況では、turbo モードを無効にするとパフォーマンスが向上しません。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-enable-turbo-mode: on

#### 3.1.1.74. nsslapd-enable-upgrade-hash

単純なバインド時に、バインド操作の性質上、Directory Server はプレーンテキストのパスワードにアクセスできます。**nsslapd-enable-upgrade-hash** パラメーターが有効で、ユーザーを認証した場合、Directory Server は、ユーザーの **userPassword** 属性が **passwordStorageScheme** 属性に設定されたハッシュアルゴリズムを使用するかどうかを確認します。アルゴリズムが異なる場合、サーバーは **passwordStorageScheme** のアルゴリズムでプレーンテキストのパスワードをハッシュ化し、ユーザーの **userPassword** 属性の値を更新します。

たとえば、弱いアルゴリズムを使用してハッシュ化されたパスワードを持つユーザーエントリーをインポートする場合、サーバーは **passwordStorageScheme**(デフォルトでは **PBKDF2\_SHA256**) に設定したアルゴリズムを使用して、ユーザーの最初のログインでパスワードを自動的にハッシュ化します。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString

パラメーター	説明
例	nsslapd-enable-upgrade-hash: on

### 3.1.1.75. nsslapd-enquote-sup-oc (上級オブジェクトクラスエンクォーティングの有効化)

この属性は非推奨となり、Directory Server の今後のバージョンで削除されます。

この属性は、**cn=schema** エントリーに含まれる **objectclass** 属性の引用が、インターネットのドラフト RFC 2252 によって指定された引用に準拠するかどうかを制御します。デフォルトでは、Directory Server は RFC 2252 に準拠しており、この値は引用符で囲まれていないことを示します。非常に古いクライアントのみでは、この値を **on** に設定する必要があります。したがって、この値は **off** のままにします。

この属性をオンまたはオフにしても、Directory Server コンソールには影響を及ぼしません。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-enquote-sup-oc: off

### 3.1.1.76. nsslapd-entryusn-global

**nsslapd-entryusn-global** パラメーターは、**USN** プラグインがすべてのバックエンドデータベースまたは各データベースに個別に生成される更新シーケンス番号 (USN) を割り当てるかどうかを定義します。すべてのバックエンドデータベースで一意的な USN の場合は、このパラメーターを **on** に設定します。

詳細は「[entryusn](#)」を参照してください。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString



パラメーター	説明
例	nsslapd-entryusn-global: off

### 3.1.1.77. nsslapd-entryusn-import-initval

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 (レプリケーション用のデータベースの初期化の際を含む)。デフォルトでは、インポートされたエントリーのエントリー USN はゼロに設定されます。

**nsslapd-entryusn-import-initval** を使用して、エントリー USN に別の初期値を設定できます。これは、すべてのインポートされたエントリーに使用される開始 USN を設定します。

**nsslapd-entryusn-import-initval** には 2 つの値があります。

- 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。
- **next**。つまり、インポートされたエントリーはすべて、インポート操作の前にサーバー上にあった最大のエントリー USN 値を、1 つずつインクリメントして使用します。

パラメーター	説明
エントリー DN	cn=config
有効な値	任意の整数   次へ
デフォルト値	
構文	DirectoryString
例	nsslapd-entryusn-import-initval: next

### 3.1.1.78. nsslapd-errorlog(エラーログ)

この属性は、Directory Server が生成するエラーメッセージを記録するために使用されるログのパスおよびファイル名を設定します。これらのメッセージはエラー状態を記述することができますが、多くの場合、以下のような情報条件が含まれます。

- サーバーの起動およびシャットダウン時間。
- サーバーが使用するポート番号。

このログの情報量は、Log Level 属性の現在の設定により異なります。詳細は、「[nsslapd-errorlog-level\(エラーログレベル\)](#)」を参照してください。

パラメーター	説明
エントリー DN	cn=config

パラメーター	説明
有効な値	有効なファイル名
デフォルト値	/var/log/dirsrv/slapd- <b>instance</b> /errors
構文	DirectoryString
例	nsslapd-errorlog: /var/log/dirsrv/slapd- <b>instance</b> /errors

エラーロギングを有効にするには、この属性に有効なパスとファイル名が必要で、**nsslapd-errorlog-logging-enabled** 設定属性を **on** に切り替える必要があります。以下の表は、これらの2つの設定属性と、エラーロギングを無効化または有効にするという2つの設定属性に対して可能な4つの値の組み合わせを示しています。

表3.5 Nsslapd-errorlog 設定属性に考えられる組み合わせ

dse.ldif の属性	値	ロギングの有効化または無効化
nsslapd-errorlog-logging-enabled	on	Disabled
nsslapd-errorlog	空の文字列	
nsslapd-errorlog-logging-enabled	on	有効
nsslapd-errorlog	<b>filename</b>	
nsslapd-errorlog-logging-enabled	off	Disabled
nsslapd-errorlog	空の文字列	
nsslapd-errorlog-logging-enabled	off	Disabled
nsslapd-errorlog	<b>filename</b>	

### 3.1.1.79. nsslapd-errorlog-level(エラーログレベル)

この属性は、Directory Server のロギングレベルを設定します。ログレベルは加算されます。つまり、**3** の値を指定するとレベル **1** と **2** の両方が含まれます。

**nsslapd-errorlog-level** のデフォルト値は **16384** です。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config

パラメーター	説明
有効な値	<p>* 1 - 追跡関数の呼び出し。サーバーに入る際にメッセージをログに記録し、関数を終了します。</p> <p>* 2 - デバッグパケット処理。</p> <p>* 4 - ヘビートレース出力デバッグ。</p> <p>* 8 - 接続管理。</p> <p>* 16 - 送信/受信パケットの出力。</p> <p>* 32 - 検索フィルター処理。</p> <p>* 64 - 設定ファイル処理。</p> <p>* 128 - アクセス制御リスト処理。</p> <p>* 1024 - シェルデータベースとのログ通信。</p> <p>* 2048 - デバッグを解析するログエントリ。</p> <p>* 4096 - ハウスキーピングスレッドのデバッグ。</p> <p>* 8192 - レプリケーションのデバッグ。</p> <p>* 16384 - 重大なエラーや、常にエラーログに書き込まれるその他のメッセージに使用されるデフォルトのロギングレベル (例: サーバー起動メッセージ) このレベルのメッセージは、ログレベルの設定に関係なく、常にエラーログに含まれます。</p> <p>* 32768 - データベースキャッシュのデバッグ</p> <p>* 65536 - サーバープラグインのデバッグ。サーバープラグインが <b>slapi-log-error</b> を呼び出す際に、ログファイルにエントリを書き込みます。</p> <p>* 262144 - アクセス制御サマリー情報。レベル <b>128</b> よりも詳細度が低くなります。この値は、アクセス制御処理の概要が必要な場合に推奨されます。非常に詳細な処理メッセージには <b>128</b> を使用します。</p> <p>* 524288 - LMDB データベースのデバッグ。</p>
デフォルト値	16384
構文	整数
例	nsslapd-errorlog-level: 8192

## 3.1.1.80. nsslapd-errorlog-list

この読み取り専用属性は、エラーログファイルのリストを提供します。

パラメーター	説明
エントリー DN	cn=config
有効な値	
デフォルト値	なし
構文	DirectoryString
例	nsslapd-errorlog-list: errorlog2,errorlog3

### 3.1.1.81. nsslapd-errorlog-logexpirationtime(エラーログの有効期限)

この属性は、削除前にログファイルが到達できる最大期間を設定します。この属性はユニット数のみを提供します。ユニット (day、week、month など) は **nsslapd-errorlog-logexpirationtimeunit** 属性で指定します。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1 から最大 32 ビットの整数値 (2147483647) -1 または 0 の値は、ログが期限切れになることはありません。
デフォルト値	-1
構文	整数
例	nsslapd-errorlog-logexpirationtime: 1

### 3.1.1.82. nsslapd-errorlog-logexpirationtimeunit(エラーログの有効期限時間単位)

この属性は、**nsslapd-errorlog-logexpirationtime** 属性の単位を設定します。ユニットがサーバーによって認識されていない場合は、ログが期限切れになることはありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	month   week   day
デフォルト値	month

パラメーター	説明
構文	DirectoryString
例	nsslapd-errorlog-logexpirationtimeunit: week

### 3.1.1.83. nsslapd-errorlog-logging-enabled(エラーロギングの有効化)

エラーロギングのオンとオフを切り替えます。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-errorlog-logging-enabled: on

### 3.1.1.84. nsslapd-errorlog-logmaxdiskpace(エラーログの最大ディスク領域)

この属性は、エラーログが消費できる最大ディスク容量をメガバイト単位で設定します。この値を超えると、最も古いエラーログが削除されます。

最大ディスク容量を設定する場合は、ログファイルのローテーションにより作成可能なログファイルの合計数を考慮してください。また、Directory Server が管理する 3 つの異なるログファイル (アクセスログ、監査ログ、およびエラーログ) があり、それぞれがディスク領域を消費することに注意してください。これらの考慮事項を、エラーログのディスク領域の合計量と比較します。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1 1 から最大 32 ビットの整数値 (2147483647)。値が -1 の場合は、エラーログに許可されるディスク領域のサイズが無制限であることを意味します。
デフォルト値	100
構文	整数
例	nsslapd-errorlog-logmaxdiskpace: 10000

### 3.1.1.85. nsslapd-errorlog-logminfreediskpace(エラーログの最小ディスク領域)

この属性は、許可される最小空きディスク容量をメガバイト単位で設定します。空きディスク容量がこの属性で指定された値を下回ると、この属性を満たすために十分なディスク領域が解放されるまで、最も古いアクセスログが削除されます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1(無制限)   1 から 32 ビットの整数値 (2147483647)
デフォルト値	-1
構文	整数
例	nsslapd-errorlog-logminfreediskspace: -1

### 3.1.1.86. nsslapd-errorlog-logrotationsync-enabled(エラーログローテーション同期が有効)

この属性は、エラーログのローテーションが特定の時刻と同期するかどうかを設定します。この方法でログローテーションを同期すると、毎日深夜から深夜までなど、1日の指定された時間にログファイルを生成できます。これにより、ログファイルがカレンダーに直接マップされるため、ログファイルの分析がはるかに簡単になります。

エラーロギングのローテーションを時刻で同期するには、この属性を **nsslapd-errorlog-logrotationsynchour** 属性値および **nsslapd-errorlog-logrotationsyncmin** 属性値をログファイルのローテーションの時間および分に設定して、この属性を有効にする必要があります。

たとえば、毎日深夜にエラーログファイルをローテーションするには、その値を **on** に設定してこの属性を有効にしてから、**nsslapd-errorlog-logrotationsynchour** 属性および **nsslapd-errorlog-logrotationsyncmin** 属性の値を **0** に設定します。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-errorlog-logrotationsync-enabled: on

### 3.1.1.87. nsslapd-errorlog-logrotationsynchour(エラーログローテーション同期時間)

この属性は、エラーログをローテーションする時刻を設定します。この属性は、**nsslapd-errorlog-logrotationsync-enabled** 属性および **nsslapd-errorlog-logrotationsyncmin** 属性と共に使用する必要があります。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 - 23
デフォルト値	0
構文	整数
例	nsslapd-errorlog-logrotationsynchour: 23

### 3.1.1.88. nsslapd-errorlog-logrotationsyncmin(エラーログローテーション同期確認)

この属性は、エラーログをローテーションするために1日分を設定します。この属性は、**nsslapd-errorlog-logrotationsync-enabled** 属性および **nsslapd-errorlog-logrotationsynchour** 属性と併用する必要があります。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 - 59
デフォルト値	0
構文	整数
例	nsslapd-errorlog-logrotationsyncmin: 30

### 3.1.1.89. nsslapd-errorlog-logrotationtime(エラーログローテーション時間)

この属性は、エラーログファイルのローテーションの間隔を設定します。この属性はユニット数のみを提供します。ユニット (day、week、month など) は **nsslapd-errorlog-logrotationtimeunit** (エラーログローテーション時間単位) 属性で指定します。

Directory Server は、ログのサイズにかかわらず、設定された間隔が過ぎた後の最初の書き込み操作でログをローテーションします。

パフォーマンス上の理由から、ログローテーションを指定しないことは推奨しませんが、ログは無期限に大きくなるため、これを指定する方法は2つあります。**nsslapd-errorlog-maxlogsperdir** 属性値を **1** に設定するか、**nsslapd-errorlog-logrotationtime** 属性を **-1** に設定します。サーバーは最初に **nsslapd-errorlog-maxlogsperdir** 属性をチェックして、この属性の値が **1** よりも大きい場合、サーバーは **nsslapd-errorlog-logrotationtime** 属性をチェックします。詳細は、「[nsslapd-errorlog-maxlogsperdir\(最大エラーログファイル数\)](#)」を参照してください。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1 1 から最大 32 ビットの整数値 (2147483647)。値が -1 の場合は、エラーログファイルのローテーションの間隔が無制限になります。
デフォルト値	1
構文	整数
例	nsslapd-errorlog-logrotationtime: 100

### 3.1.1.90. nsslapd-errorlog-logrotationtimeunit(エラーログローテーション時間単位)

この属性は、**nsslapd-errorlog-logrotationtime** (エラーログローテーション時間) の単位を設定します。ユニットがサーバーによって認識されていない場合は、ログが期限切れになることはありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	month   week   day   hour   minute
デフォルト値	週
構文	DirectoryString
例	nsslapd-errorlog-logrotationtimeunit: day

### 3.1.1.91. nsslapd-errorlog-maxlogsize(最大ログサイズ)

この属性は、最大エラーログサイズをメガバイト単位で設定します。この値に達すると、エラーログがローテーションされ、サーバーはログ情報の新しいログファイルへの書き込みを開始します。**nsslapd-errorlog-maxlogspedir** が 1 に設定されている場合、サーバーはこの属性を無視します。

最大ログサイズを設定する場合は、ログファイルのローテーションにより作成可能なログファイルの合計数を考慮してください。また、Directory Server が管理する 3 つの異なるログファイル (アクセスログ、監査ログ、およびエラーログ) があり、それぞれがディスク領域を消費することに注意してください。これらの考慮事項を、エラーログのディスク領域の合計量と比較します。

パラメーター	説明
エントリー DN	cn=config



パラメーター	説明
Valid Range	-1 1から最大 32 ビットの整数値 (2147483647) で、値が -1 の場合は、ログファイルのサイズが無制限になります。
デフォルト値	100
構文	整数
例	nsslapd-errorlog-maxlogsize: 100

### 3.1.1.92. nsslapd-errorlog-maxlogsperdir(最大エラーログファイル数)

この属性は、エラーログが保存されるディレクトリーに格納できるエラーログの合計数を設定します。エラーログがローテーションされるたびに、新しいログファイルが作成されます。エラーログディレクトリーに含まれるファイルの数がこの属性に保存されている値を超えると、ログファイルの最も古いバージョンは削除されます。デフォルトは 1 ログです。このデフォルト値が許可されると、サーバーはログをローテーションせず、無制限に増加します。

この属性の値が 1 よりも大きい場合は、**nsslapd-errorlog-logrotationtime** 属性をチェックして、ログローテーションが指定されているかどうかを確認します。**nsslapd-errorlog-logrotationtime** 属性の値が -1 の場合、ログローテーションはありません。詳細は、「[nsslapd-errorlog-logrotationtime\(エラーログローテーション時間\)](#)」を参照してください。

パラメーター	説明
エントリー DN	cn=config
Valid Range	1 から最大 32 ビットの整数値 (2147483647)
デフォルト値	1
構文	整数
例	nsslapd-errorlog-maxlogsperdir: 10

### 3.1.1.93. nsslapd-errorlog-mode(エラーログファイルのパーミッション)

この属性は、エラーログファイルが作成されるアクセスモードまたはファイルパーミッションを設定します。有効な値は、番号がミラーリングされるか、絶対 UNIX ファイルのパーミッションになるため、**000** から **777** の組み合わせです。つまり、値は 3 桁の数字の組み合わせである必要があり、数字は **0** から **7** によって異なります。

- 0 - なし
- 1 - 実行のみ

- 2 - 書き込みのみ
- 3 - 書き込みおよび実行
- 4 - 読み取り専用
- 5 - 読み取りおよび実行
- 6 - 読み取りおよび書き込み
- 7 - 読み取り、書き込み、および実行

3桁の数値では、最初の数字は所有者の権限を表し、2つ目の数字はグループのパーミッションを表し、3つ目の数字は全員のパーミッションを表します。デフォルト値を変更する場合、**000**はログへのアクセスを許可せず、すべてのユーザーに書き込み権限を許可すると、ログが上書きまたは削除される可能性があることに注意してください。

新しく設定されたアクセスモードは、作成された新しいログにのみ影響します。モードは、ログが新しいファイルにローテーションするときに設定されます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	000 から 777 まで
デフォルト値	600
構文	整数
例	nsslapd-errorlog-mode: 600

### 3.1.1.94. nsslapd-force-sasl-external

TLS 接続を確立すると、クライアントは最初に証明書を送信し、SASL/EXTERNAL メカニズムを使用して BIND 要求を発行します。SASL/EXTERNAL を使用すると、Directory Server に対して、TLS ハンドシェイクの証明書の認証情報を使用するように指示します。ただし、一部のクライアントは BIND 要求を送信するときに SASL/EXTERNAL を使用しないため、Directory Server は簡易認証要求または匿名要求としてバインドを処理し、TLS 接続が失敗します。

**nsslapd-force-sasl-external** 属性は、証明書ベースの認証でクライアントを強制し、SASL/EXTERNAL メソッドを使用して BIND 要求を送信します。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off

パラメーター	説明
構文	String
例	nsslapd-force-sasl-external: on

### 3.1.1.95. nsslapd-groupevalnestlevel

この属性は非推奨になり、これまでの目的でのみ説明されます。

アクセス制御プラグインは **nsslapd-groupevalnestlevel** 属性で指定された値を使用して、アクセス制御がグループ評価用に実行するネストのレベル数を設定します。その代わりに、ネスト化のレベルの数は **5** としてハードコーディングされます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 から 5
デフォルト値	5
構文	整数
例	nsslapd-groupevalnestlevel: 5

### 3.1.1.96. nsslapd-haproxy-trusted-ip (HAProxy 信頼済み IP)

**nsslapd-haproxy-trusted-ip** 属性は、信頼できるプロキシサーバーのリストを設定します。nsslapd-haproxy-trusted-ip を設定すると、Directory Server は HAProxy プロトコルを使用して追加の TCP ヘッダー経由でクライアント IP アドレスを受信し、アクセス制御命令 (ACI) を正しく評価してクライアントトラフィックをログに記録します。

信頼されていないプロキシサーバーがバインド要求を開始した場合、Directory Server は要求を拒否し、エラーログファイルに次のメッセージを記録します。

```
[time_stamp] conn=5 op=-1 fd=64 Disconnect - Protocol error - Unknown Proxy - P4
```

パラメーター	説明
エントリー DN	cn=config
Valid Range	IPv4 または IPv6 アドレス。
デフォルト値	

パラメーター	説明
構文	DirectoryString
例	nsslapd-haproxy-trusted-ip: 127.0.0.1

### 3.1.1.97. nsslapd-idletimeout(デフォルトのアイドルタイムアウト)

この属性は、アイドル状態の LDAP クライアント接続がサーバーによって閉じられるまでの秒数を設定します。**0** に設定すると、サーバーはアイドル状態の接続を閉じなくなります。この設定は、すべての接続およびすべてのユーザーに適用されます。**Poll()** が **0** を返さない場合、接続テーブルがウォークされたときにアイドル状態のタイムアウトが強制されます。そのため、1つの接続を持つサーバーはアイドル状態のタイムアウトを強制しません。

ユーザーエントリーに追加できる **nslidleTimeout** 操作属性を使用して、この属性に割り当てられた値を上書きします。詳細は、Red Hat Directory Server 管理ガイドのバインド DN に基づいたリソース制限の設定セクションを参照してください。



#### 注記

非常に大規模なデータベースの場合、この属性には、エントリーへの接続がタイムアウトすると、オンライン初期化プロセスが完了するか、レプリケーションが失敗するまで十分な値が必要です。または、**nslidleTimeout** 属性を、サプライヤーバインド DN として使用するエントリーの高い値に設定できます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 から最大 32 ビットの整数値 (2147483647)
デフォルト値	3600
構文	整数
例	nsslapd-idletimeout: 3600

### 3.1.1.98. nsslapd-ignore-virtual-attrs

このパラメーターを使用すると、検索エントリーで仮想属性ルックアップを無効にできます。

仮想属性が必要ない場合は、検索結果で仮想属性ルックアップを無効にして、検索の速度を増やすことができます。

パラメーター	説明
エントリー DN	cn=config

パラメーター	説明
有効な値	on   off
デフォルト値	off
構文	DirectoryString
Example	nsslapd-ignore-virtual-attrs: off

### 3.1.1.99. nsslapd-instancedir(インスタンスディレクトリー)

この属性は非推奨になりました。**nsslapd-certdir**、**nsslapd-lockdir** などのインスタンス固有のパス用に個別の設定パラメーターになりました。設定された特定のディレクトリーパスのドキュメントを参照してください。

### 3.1.1.100. nsslapd-ioblocktimeout(IO ブロックのタイムアウト)

この属性は、停止した LDAP クライアントへの接続を閉じるまでの時間をミリ秒単位で設定します。LDAP クライアントは、読み取りまたは書き込み操作の I/O の進捗が全くない場合には停止されたと思なされます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	ティックにおける 0 から最大 32 ビットの整数値 (2147483647)
デフォルト値	10000
構文	整数
例	nsslapd-ioblocktimeout: 10000

### 3.1.1.101. nsslapd-lastmod(トラッキング変更時間)

この属性は、Directory Server が新しく作成または更新されたエントリーの操作属性 **creatorsName**、**createTimestamp**、**modifiersName**、および **modifyTimestamp** を維持するかどうかを設定します。



#### 重要

Red Hat は、これらの属性の追跡を無効にしないことを推奨します。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。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-lastmod: on

### 3.1.1.102. nsslapd-ldapiautobind(Enable Autobind)

**nsslapd-ldapiautobind** は、サーバーが LDAPi を使用して Directory Server に自動バインドできるようにするかどうかを設定します。自動バインドは、システムユーザーの UID または GUID 数を Directory Server ユーザーにマッピングし、これらの認証情報に基づいて Directory Server に対してユーザーを自動的に認証します。Directory Server 接続は UNIX ソケット上で実行されます。

自動バインドを有効にするとともに、自動バインドを設定するには、マッピングエントリーを設定する必要があります。**nsslapd-ldapimaprootdn** は、システム上の root ユーザーを Directory Manager にマッピングします。**nsslapd-ldapimaptentries** は、**nsslapd-ldapiuidnumbertype**、**nsslapd-ldapigidnumbertype**、および **nsslapd-ldapientrysearchbase** 属性で定義されたパラメーターに基づいて、通常のユーザーを Directory Server ユーザーにマッピングします。

autobind は LDAPi が有効な場合にのみ有効にできます。つまり、**nsslapd-ldapilisten** が **on** になり、**nsslapd-ldapifilepath** 属性が LDAPi ソケットに設定されます。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-ldapiautobind: off

### 3.1.1.103. nsslapd-ldapientrysearchbase(LDAPi 認証エントリーの検索ベース)

自動バインドでは、システムユーザーの UID および GUID 番号に基づいて、システムユーザーを Directory Server ユーザーエントリーにマッピングできます。これには、UID 番号 (**nsslapd-ldapiuidnumbertype**) および GUID 番号 (**nsslapd-ldapigidnumbertype**) に使用する属性の Directory Server パラメーターを設定し、一致する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する必要があります。

**nsslapd-ldapientrysearchbase** で、自動バインドに使用するユーザーエントリーを検索するサブツリーを指定します。

パラメーター	説明
エントリー DN	cn=config
有効な値	DN
デフォルト値	サーバーインスタンスの作成時に作成された接尾辞 (例: <b>dc=example,dc=com</b> )
構文	DN
例	nsslapd-ldapientrysearchbase: ou=people,dc=example,dc=om

#### 3.1.1.104. nsslapd-ldapifilepath(LDAPI ソケットのファイルの場所)

LDAPI は、TCP ではなく UNIX ソケットを介して LDAP サーバーに接続します。LDAPI を設定するには、UNIX ソケットを介して通信するようにサーバーを設定する必要があります。使用する UNIX ソケットは、**nsslapd-ldapifilepath** 属性に設定されます。

パラメーター	説明
エントリー DN	cn=config
有効な値	任意のディレクトリーパス
デフォルト値	/var/run/dirsrv/slapd-example.socket
構文	大文字と小文字を区別する文字列
例	nsslapd-ldapifilepath: /var/run/slapd-example.socket

#### 3.1.1.105. nsslapd-ldapigidnumbertype(System GUID 番号の属性マッピング)

自動バインドを使用して、システムユーザーを自動的に認証し、UNIX ソケットを使用してサーバーに接続できます。システムユーザーを、認証のために Directory Server ユーザーにマッピングするには、システムユーザーの UID および GUID 番号を Directory Server 属性にマッピングする必要があります。**nsslapd-ldapigidnumbertype** 属性は、システム GUID をユーザーエントリーにマッピングする Directory Server 属性を示します。

LDAPI が有効になっている場合 (**nsslapd-ldapilisten** および **nsslapd-ldapifilepath**)、自動バインドが有効になっている場合 (**nsslapd-ldapiautobind**)、および通常の利用者に対して自動バインドマッピングが有効になっている場合 (**nsslapd-ldapimaptoentries**) にのみ、ユーザーは自動バインドを使用してサーバーに接続できます。

パラメーター	説明
エントリー DN	cn=config
有効な値	Directory Server の属性
デフォルト値	gidNumber
構文	DirectoryString
例	nsslapd-ldapigidnumbertype: gidNumber

### 3.1.1.106. nsslapd-ldapilisten(LDAPI の有効化)

**nsslapd-ldapilisten** は、Directory Server への LDAPI 接続を有効にします。LDAPI を使用すると、ユーザーは標準の TCP ポートではなく UNIX ソケットを介して Directory Server に接続できるようになります。**nsslapd-ldapilisten** を **on** に設定して LDAPI を有効にすることに加えて、**nsslapd-ldapifilepath** 属性に LDAPI 用に設定された UNIX ソケットも必要です。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-ldapilisten: on

### 3.1.1.107. nsslapd-ldapimaprootdn (root ユーザー用の自動バインドマッピング)

自動バインドにより、システムユーザーは Directory Server ユーザーにマッピングされ、UNIX ソケットを介して Directory Server に対して自動的に認証されます。

root システムユーザー (UID が 0 のユーザー) は、**nsslapd-ldapimaprootdn** 属性で指定した Directory Server エントリーにマッピングされます。

パラメーター	説明
エントリー DN	cn=config
有効な値	任意の DN
デフォルト値	cn=Directory Manager



パラメーター	説明
構文	DN
例	nsslapd-ldapimaprootdn: cn=Directory Manager

### 3.1.1.108. nsslapd-ldapimaptoentries(通常ユーザーの自動バインドマッピングの有効化)

自動バインドにより、システムユーザーは Directory Server ユーザーにマッピングされ、UNIX ソケットを介して Directory Server に対して自動的に認証されます。このマッピングは root ユーザーに対して自動化されますが、**nsslapd-ldapimaptoentries** 属性を介して通常のシステムユーザーに対して有効にする必要があります。この属性を **on** に設定すると、通常のシステムユーザーを Directory Server エントリーにマッピングできます。この属性が有効になっていない場合は、root ユーザーのみが autobind を使用して Directory Server に対して認証し、他のすべてのユーザーは匿名で接続できます。

マッピング自体は、**nsslapd-ldapiuidnumbertype** 属性および **nsslapd-ldapigidnumbertype** 属性で設定され、Directory Server 属性をユーザーの UID および GUID の番号にマップします。

LDAPAPI が有効になっている場合 (**nsslapd-ldapilisten** および **nsslapd-ldapifilepath**)、自動バインドが有効になっている場合 (**nsslapd-ldapiautobind**) にのみ、ユーザーは自動バインドを使用してサーバーに接続できます。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-ldapimaptoentries: on

### 3.1.1.109. nsslapd-ldapiuidnumbertype

自動バインドを使用して、システムユーザーを自動的に認証し、UNIX ソケットを使用してサーバーに接続できます。システムユーザーを、認証のために Directory Server ユーザーにマッピングするには、システムユーザーの UID および GUID 番号を Directory Server 属性にマッピングする必要があります。**nsslapd-ldapiuidnumbertype** 属性は、システム UID をユーザーエントリーにマップするために Directory Server 属性を示します。

LDAPAPI が有効になっている場合 (**nsslapd-ldapilisten** および **nsslapd-ldapifilepath**)、自動バインドが有効になっている場合 (**nsslapd-ldapiautobind**)、および通常のユーザーに対して自動バインドマッピングが有効になっている場合 (**nsslapd-ldapimaptoentries**) にのみ、ユーザーは自動バインドを使用してサーバーに接続できます。

パラメーター	説明
エントリー DN	cn=config
有効な値	Directory Server の属性
デフォルト値	uidNumber
構文	DirectoryString
例	nsslapd-ldapiuidnumbertype: uidNumber

### 3.1.1.110. nsslapd-ldifdir

Directory Server は、**db2ldif** または **db2ldif.pl** を使用する場合に、このパラメーターで設定したディレクトリーに LDAP データ交換形式 (LDIF) 形式のファイルをエクスポートします。ディレクトリーは Directory Server のユーザーおよびグループが所有する必要があります。このユーザーおよびグループは、このディレクトリーに読み取りおよび書き込みアクセスを持つ必要があるだけです。

この属性への変更を反映するには、サービスを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=config
有効な値	Directory Server ユーザーが書き込み可能なディレクトリー
デフォルト値	/var/lib/dirsrv/slapd- <b>instance_name</b> /ldif/
構文	DirectoryString
例	nsslapd-ldifdir: /var/lib/dirsrv/slapd- <b>instance_name</b> /ldif/

### 3.1.1.111. nsslapd-listen-backlog-size

この属性は、ソケット接続のバックログの最大数を設定します。listen サービスは、受信接続を受け付けるソケット数を設定します。Backlog 設定は、接続を拒否する前にソケット (sockfd) のキューを拡張する最大期間を設定します。

パラメーター	説明
エントリー DN	cn=config
有効な値	最大 64 ビットの整数値 (9223372036854775807)

パラメーター	説明
デフォルト値	128
構文	整数
例	nsslapd-listen-backlog-size: 128

### 3.1.1.112. nsslapd-listenhost(IP アドレスに追加)

この属性により、複数の Directory Server インスタンスがマルチホームのマシンで実行できるようになります (または、マルチホームマシンの1つのインターフェイスのリッスンを制限することができます)。1つのホップに関連する複数の IP アドレスが存在する可能性があります。これらの IP アドレスは、IPv4 と IPv6 の両方の組み合わせになります。このパラメーターを使用して、Directory Server インスタンスを単一の IP インターフェイスに制限することができます。

ホスト名が **nsslapd-listenhost** 値として指定される場合、Directory Server はホスト名に関連付けられたすべてのインターフェイスについて要求に応答します。単一 IP インターフェイス (IPv4 または IPv6) が **nsslapd-listenhost** の値として指定される場合、Directory Server は、その特定のインターフェイスに送信された要求のみに応答します。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。

この属性の変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=config
有効な値	ローカルホスト名、IPv4 アドレスまたは IPv6 アドレス
デフォルト値	
構文	DirectoryString
例	nsslapd-listenhost: ldap.example.com

### 3.1.1.113. nsslapd-localhost(ローカルホスト)

この属性は、Directory Server を実行するホストマシンを指定します。この属性は、MMR プロトコルの一部を設定する参照 URL を作成します。フェイルオーバーノードのある高可用性設定では、その参照はローカルホスト名ではなく、クラスターの仮想名を参照する必要があります。

パラメーター	説明
エントリー DN	cn=config
有効な値	完全修飾ホスト名。

パラメーター	説明
デフォルト値	インストールされたマシンのホスト名。
構文	DirectoryString
例	nsslapd-localhost: phonebook.example.com

### 3.1.1.114. nsslapd-localuser(ローカルユーザー)

この属性は、Directory Server を実行するユーザーを設定します。ユーザーを実行するグループは、ユーザーのプライマリグループを調べてこの属性から派生します。ユーザーの変更により、このインスタンス向けのインスタンス固有のファイルおよびディレクトリーはすべて **chown** などのツールを使用して、新規ユーザーによって所有されるように変更する必要があります。

サーバーインスタンスの設定時に **nsslapd-localuser** の値が最初に設定されます。

パラメーター	説明
エントリー DN	cn=config
有効な値	有効なユーザー
デフォルト値	
構文	DirectoryString
例	nsslapd-localuser: dirsrv

### 3.1.1.115. nsslapd-lockdir(サーバーロックファイルディレクトリー)

これは、サーバーがロックファイルに使用するディレクトリーへの完全パスです。デフォルト値は **/var/lock/dirsrv/slapd-instance** です。この値の変更は、サーバーが再起動するまで反映されません。

パラメーター	説明
エントリー DN	cn=config
有効な値	サーバー ID への書き込みアクセスのあるサーバーユーザー ID が所有するディレクトリーへの絶対パス
デフォルト値	<b>/var/lock/dirsrv/slapd-instance</b>
構文	DirectoryString
例	nsslapd-lockdir: /var/lock/dirsrv/slapd-instance

### 3.1.1.116. nsslapd-localssf

**nsslapd-localssf** パラメーターは、LDAPAPI 接続のセキュリティー強度係数 (SSF) を設定します。Directory Server は、**nsslapd-localssf** に設定した値が **nsslapd-minssf** パラメーターに設定した値と同じか、それ以上の場合にのみ LDAPAPI 接続を許可します。そのため、LDAPAPI の接続は **nsslapd-minssf** の最小 SSF セットに対応します。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	0 から最大 32 ビットの整数値 (2147483647)
デフォルト値	71
構文	整数
例	nsslapd-localssf: 71

### 3.1.1.117. nsslapd-logging-hr-timestamps-enabled(高解像度ログタイムスタンプの有効化または無効化)

ログがナノ秒の精度で高解像度のタイムスタンプを使用するか、1秒の精度で標準解決タイムスタンプを使用するかどうかを制御します。デフォルトでは有効です。ログのタイムスタンプを1秒の精度に戻すには、このオプションを **off** に設定します。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	<b>on   off</b>
デフォルト値	<b>on</b>
構文	DirectoryString
例	nsslapd-logging-hr-timestamps-enabled: on

### 3.1.1.118. nsslapd-maxbersize(最大メッセージサイズ)

受信メッセージに許可される最大サイズ (バイト単位) を定義します。これにより、Directory Server で処理できる LDAP 要求のサイズが制限されます。要求のサイズを制限すると、さまざまな DoS 攻撃を防ぎます。

この制限の対象は LDAP 要求の合計サイズです。たとえば、エントリーの追加要求で、要求のエントリーがデフォルトの設定値よりも大きい場合に、この追加要求は拒否されます。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。この属性の変更には細心の注意を払ってください。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 - 2 (ギガバイト) - (2,147,483,647 バイト)  0 がゼロの場合は、デフォルト値を使用する必要があります。
デフォルト値	2097152
構文	整数
例	nsslapd-maxbersize: 2097152

### 3.1.1.119. nsslapd-maxdescriptors(最大ファイル記述子)

この属性は、Directory Server が使用しようとするファイル記述子の最大数を設定します。ファイル記述子は、クライアントがサーバーに接続するたびに使用されます。ファイル記述子は、アクセスログ、エラーログ、監査ログ、データベースファイル (インデックスおよびトランザクションログ)、およびレプリケーションおよびチェーンの他のサーバーへの発信接続のソケットとして使用されます。

TCP/IP がクライアント接続に使用できる記述子の数は、**nsslapd-conntablesize** 属性によって決まります。この属性のデフォルト値は、ファイル記述子のソフト制限に設定されており、デフォルトは 1024 です。ただし、この属性を手動で設定すると、サーバーはプロセスファイル記述子のソフト制限が同じになるように更新します。

この値が設定されすぎると、Directory Server はオペレーティングシステムに許容可能な最大値をクエリーしてから、その値を使用します。また、エラーログに情報メッセージが出力されます。Directory Server Console または **ldapmodify** を使用して、この値をリモートで無効な値に設定すると、サーバーは新しい値を拒否し、古い値を維持し、エラーで応答します。

一部のオペレーティングシステムでは、ユーザーがプロセスで使用できるファイル記述子の数を設定できます。ファイル記述子の制限と設定に関する詳細は、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。**dsktune** プログラム (Red Hat Directory Server インストールガイドで説明) は、必要に応じてファイル記述子の数を増やすなど、システムカーネルまたは TCP/IP チューニング属性への変更を提案するために使用できます。ファイル記述子が不足しているため、Directory Server が接続を拒否する場合は、この属性の値を増加させます。これが発生すると、以下のメッセージが Directory Server のエラーログファイルに書き込まれます。

```
Not listening for new connections -- too many fds open
```

受信接続の数を増やす方法は、「[nsslapd-conntablesize](#)」を参照してください。



## 注記

UNIX シェルには通常、ファイル記述子の数に対する設定可能な制限があります。**limit** および **ulimit** についての詳細は、オペレーティングシステムのドキュメンテーションを参照してください。これらの制限により、多くの場合で問題が発生する可能性があります。

この属性の変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=config
Valid Range	1 から 65535
Default Value	4096
Syntax	整数
Example	nsslapd-maxdescriptors: 4096

### 3.1.1.120. nsslapd-maxsasliosize (最大 SASL パケットサイズ)

ユーザーが SASL GSS-API 経由で Directory Server に対して認証される場合、サーバーはクライアントに一定量のメモリーをクライアントに割り当て、クライアント要求するメモリーの量に応じて、LDAP 操作を実行する必要があります。攻撃者は、このようなサイズの大きいパケットサイズを送信して、Directory Server がクラッシュするか、サービス拒否攻撃の一部として無限に連携する可能性があります。

Directory Server が SASL クライアントに許可するパケットサイズは、**nsslapd-maxsasliosize** 属性を使用して制限できます。この属性は、サーバーが許可する SASL IO パケットの最大サイズを設定します。

受信 SASL IO パケットが **nsslapd-maxsasliosize** の制限よりも大きい場合、サーバーは即座にクライアントを切断し、メッセージをエラーログに記録し、管理者が必要に応じて設定を調節できるようにします。

この属性値はバイト単位で指定されます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	* 32 ビットシステムの -1(32 ビットの整数値) (2147483647)  * 64 ビットシステムの -1(64 ビット整数値) から最大 64 ビットの整数値 (9223372036854775807)
デフォルト値	2097152 (2MB)

パラメーター	説明
構文	整数
例	nsslapd-maxsasliosize: 2097152

### 3.1.1.121. nsslapd-maxthreadspersconn(接続あたりの最大スレッド)

コネクションが使用する必要のあるスレッドの最大数を定義します。クライアントがバインドし、バインドを解除する前に1つまたは2つの操作のみを実行する通常の操作では、デフォルト値を使用します。クライアントが多くのリクエストをバインドして同時に発生する場合は、この値を増やして、各接続ですべての操作を実行できるようにします。この属性は、サーバーコンソールでは使用できません。

パラメーター	説明
エントリー DN	cn=config
Valid Range	1 から最大 threadnumber
デフォルト値	5
構文	整数
例	nsslapd-maxthreadspersconn: 5

### 3.1.1.122. nsslapd-minssf

**セキュリティ強度係数** は、接続の強度が鍵強度に応じてどのように近いかに基づく相対測定です。SSF は、TLS または SASL 接続の保護方法を決定します。**nsslapd-minssf** 属性は、サーバーへの接続に最低限 SSF 要件を設定します。接続試行は、最低 SSF よりも弱い接続を拒否します。

TLS および SASL 接続は、Directory Server への接続で混在できます。通常、これらの接続にはそれぞれ異なる SSF があります。2 つの SSF が高いほど、最小 SSF 要件との比較に使用されます。

SSF 値を 0 に設定すると、最低限の設定はありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	正の整数
デフォルト値	0 (off)
構文	DirectoryString
例	nsslapd-minssf: 128



### 3.1.1.123. nsslapd-minssf-exclude-rootdse

セキュリティー強度係数は、接続の強度が鍵強度に応じてどのように近いかについての相対測定です。SSF は、TLS または SASL 接続の保護方法を決定します。

**nsslapd-minssf-exclude-rootdse** 属性は、ルート DSE のクエリーを除き、サーバーへの任意の接続の SSF 要件を設定します。これにより、ほとんどの接続に対して適切な SSF 値が強制され、最初にセキュアな接続を確立しなくても、クライアントがルート DSE からサーバー設定に関する情報を取得することができます。

パラメーター	説明
エントリー DN	cn=config
有効な値	正の整数
デフォルト値	0 (off)
構文	DirectoryString
例	nsslapd-minssf-exclude-rootdse: 128

### 3.1.1.124. nsslapd-moddn-aci

このパラメーターは、ディレクトリーエントリーがあるサブツリーから別のサブツリーに移動し、moddn 操作でソースおよびターゲット制限を使用するときに ACI チェックを制御します。後方互換性のために、ACI チェックを無効にできます。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-moddn-aci: on

### 3.1.1.125. nsslapd-malloc-mmap-threshold

Directory Server インスタンスが **systemctl** ユーティリティーを使用してサービスとして起動すると、**/etc/sysconfig/dirsrv** ファイルまたは **/etc/sysconfig/dirsrv-instance\_name** ファイルに設定しない限り、環境変数はサーバーに渡されません。詳細は、**systemd.exec(3)** の man ページを参照してください。

サービスファイルを手動で編集して **M\_MMAP\_THRESHOLD** 環境変数を設定する代わりに、**nsslapd-malloc-mmap-threshold** パラメーターを使用すると、Directory Server 設定で値を設定します。詳細は、**mallopt(3)** の man ページの **M\_MMAP\_THRESHOLD** パラメーターの説明を参照してください。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 - 33554432
デフォルト値	<code>malloc(3)</code> の man ページの <b>M_MMAP_THRESHOLD</b> パラメーターの説明を参照してください。
構文	整数
例	nsslapd-malloc-mmap-threshold: 33554432

### 3.1.1.126. nsslapd-malloc-mxfast

Directory Server インスタンスが `systemctl` ユーティリティーを使用してサービスとして起動すると、`/etc/sysconfig/dirsrv` ファイルまたは `/etc/sysconfig/dirsrv-instance_name` ファイルに設定しない限り、環境変数はサーバーに渡されません。詳細は、`systemd.exec(3)` の man ページを参照してください。

サービスファイルを手動で編集して **M\_MXFAST** 環境変数を設定する代わりに、**nsslapd-malloc-mxfast** パラメーターを使用すると、Directory Server の設定に値を設定できます。詳細は、`malloc(3)` の man ページの **M\_MXFAST** パラメーターの説明を参照してください。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 - 80 * (sizeof(size_t) / 4)
デフォルト値	<code>malloc(3)</code> の man ページの <b>M_MXFAST</b> パラメーターの説明を参照してください。
構文	整数
例	nsslapd-malloc-mxfast: 1048560

### 3.1.1.127. nsslapd-malloc-trim-threshold

Directory Server インスタンスが `systemctl` ユーティリティーを使用してサービスとして起動すると、`/etc/sysconfig/dirsrv` ファイルまたは `/etc/sysconfig/dirsrv-instance_name` ファイルに設定しない限り、環境変数はサーバーに渡されません。詳細は、`systemd.exec(3)` の man ページを参照してください。

サービスファイルを手動で編集して **M\_TRIM\_THRESHOLD** 環境変数を設定する代わりに、**nsslapd-malloc-trim-threshold** パラメーターを使用すると、Directory Server 設定で値を設定します。詳細は、**mallopt(3)** の man ページの **M\_TRIM\_THRESHOLD** パラメーターの説明を参照してください。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 から $2^{31}-1$
デフォルト値	<b>mallopt(3)</b> の man ページの <b>M_TRIM_THRESHOLD</b> パラメーターの説明を参照してください。
構文	整数
例	nsslapd-malloc-trim-threshold: 131072

### 3.1.1.128. nsslapd-nagle

この属性の値が **off** の場合、**TCP\_NODELAY** オプションが設定されます。これにより、LDAP 応答 (エントリーや結果メッセージなど) が即座にクライアントに送られます。属性が有効な場合、デフォルトの TCP 動作が適用されます。特に、データの送信は遅延され、通常はイーサネットの場合は1つのパケット (通常は 1500 バイト) にデータをグループ化できます。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-nagle: off

### 3.1.1.129. nsslapd-ndn-cache-enabled

識別名 (DN) の正規化は、リソース集約型タスクです。**nsslapd-ndn-cache-enabled** パラメーターが有効になっている場合、Directory Server はメモリーに正規化された DN をキャッシュします。**nsslapd-ndn-cache-max-size** パラメーターを更新して、このキャッシュの最大サイズを設定します。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-ndn-cache-enabled: on

### 3.1.1.130. nsslapd-ndn-cache-max-size

識別名 (DN) の正規化は、リソース集約型タスクです。 **nsslapd-ndn-cache-enabled** パラメーターが有効になっている場合、Directory Server はメモリーに正規化された DN をキャッシュします。 **nsslapd-ndn-cache-max-size** パラメーターは、このキャッシュの最大サイズを設定します。

要求された DN がキャッシュされていない場合は、正規化され、追加されます。キャッシュサイズの制限を超過すると、Directory Server は、キャッシュから最も最近使用された 10,000 DN を削除します。ただし、少なくとも 10,000 の DN は常にキャッシュされます。

パラメーター	説明
エントリー DN	cn=config
有効な値	0 から最大 32 ビットの整数値 (2147483647)
デフォルト値	20971520
構文	整数
例	nsslapd-ndn-cache-max-size: 20971520

### 3.1.1.131. nsslapd-outbound-ldap-io-timeout

この属性は、すべてのアウトバウンド LDAP 接続の I/O 待機時間を制限します。デフォルトは **300000** ミリ秒 (5 分) です。値が **0** の場合は、サーバーが I/O の待機時間に制限を課さないことを意味します。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 から最大 32 ビットの整数値 (2147483647)
デフォルト値	300000

パラメーター	説明
構文	DirectoryString
例	nsslapd-outbound-ldap-io-timeout: 300000

### 3.1.1.132. nsslapd-pagedsizelimit(シンプルページ結果検索のサイズ制限)

この属性は、**簡単なページ結果制御を使用する** 検索操作から返すエントリーの最大数を設定します。これにより、ページ検索の **nsslapd-sizelimit** 属性がオーバーライドされます。

この値がゼロに設定されている場合、**nsslapd-sizelimit** 属性は、ページ検索と非ページ検索に使用されます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1 から最大 32 ビットの整数値 (2147483647)
デフォルト値	
構文	整数
例	nsslapd-pagedsizelimit: 10000

### 3.1.1.133. nsslapd-plug-in

この読み取り専用属性は、サーバーによって読み込まれる構文および一致するルールプラグインのプラグインエントリーの DN をリスト表示します。

### 3.1.1.134. nsslapd-plugin-binddn-tracking

操作自体がサーバープラグインによって開始された場合でも、エントリーの修飾子として操作に使用されるバインド DN を設定します。操作を実行する特定のプラグインは、別の運用属性 **internalModifiersname** にリスト表示されます。

もう1つの変更は、ディレクトリーツリーの他の自動変更をトリガーできます。たとえば、ユーザーが削除されると、そのユーザーは Referential Integrity 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。ユーザーの初回削除は、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ますが、グループへの更新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、更新を開始したユーザーに関する情報はありません。**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。以下に例を示します。

```
dn: cn=my_group,ou=groups,dc=example,dc=com
modifiersname: uid=jsmith,ou=people,dc=example,dc=com
internalModifiersname: cn=referential integrity plugin,cn=plugins,cn=config
```

この属性はデフォルトで無効にされています。

パラメーター	説明
エントリー DN	cn=config
Valid Range	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-plugin-binddn-tracking: on

### 3.1.1.135. nsslapd-plugin-logging

デフォルトでは、アクセスログが内部操作を記録するように設定されている場合でも、プラグイン内部操作はアクセスログファイルに記録されません。各プラグインの設定でロギングを有効にする代わりに、このパラメーターを使用してグローバルに制御することができます。

有効にすると、プラグインはこのグローバル設定およびログアクセスおよび監査イベント (有効な場合) を使用します。

**nsslapd-plugin-logging** が有効で、**nsslapd-accesslog-level** が内部操作を記録するように設定されている場合は、インデックスされていない検索とその他の内部操作がアクセスログファイルに記録されません。

**nsslapd-plugin-logging** が設定されていない場合、プラグインからインデックスされていない検索は Directory Server エラーログに記録されます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-plugin-logging: off

### 3.1.1.136. nsslapd-port(ポート番号)

この属性は、標準の LDAP 通信に使用される TCP/IP ポート番号を提供します。このポートで TLS を実行するには、Start TLS 拡張操作を使用します。この選択したポートは、ホストシステムで一意でなければなりません。他のアプリケーションが同じポート番号を使用しないようにしてください。ポート番号が **1024** 未満の場合は、Directory Server を **root** で起動する必要があります。

サーバーは、起動後にその **uid** を **nsslapd-localuser** 値に設定します。設定ディレクトリーのポート番号を変更する場合は、設定ディレクトリーの対応するサーバーインスタンスエントリーを更新する必要があります。

ポート番号の変更を考慮してサーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 から 65535
デフォルト値	389
構文	整数
例	nsslapd-port: 389



### 注記

LDAPS ポートが有効な場合は、ポート番号をゼロ (0) に設定して LDAP ポートを無効にします。

#### 3.1.1.137. nsslapd-privatenamespaces

この読み取り専用属性には、プライベート命名コンテキスト **cn=config**、**cn=schema**、および **cn=monitor** のリストが含まれます。

パラメーター	説明
エントリー DN	cn=config
有効な値	cn=config, cn=schema, and cn=monitor
デフォルト値	
構文	DirectoryString
例	nsslapd-privatenamespaces: cn=config

#### 3.1.1.138. nsslapd-pwpolicy-inherit-global(グローバルパスワード構文の継承)

粒度の細かいパスワード構文が設定されていない場合、グローバルパスワード構文が設定されている場合でも、新規または更新されたパスワードは確認されません。粒度の細かいパスワード構文を継承する場合は、この属性を **on** に設定します。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off

パラメーター	説明
デフォルト値	off
構文	DirectoryString
例	nsslapd-pwpolicy-inherit-global: off

### 3.1.1.139. nsslapd-pwpolicy-local(サブツリーおよびユーザーレベルパスワードポリシーの有効化)

粒度の細かい (サブツリーおよびユーザーレベル) パスワードポリシーをオンまたはオフにします。

この属性の値が **off** の場合、ディレクトリー内のすべてのエントリー (**cn=Directory Manager** を除く) はグローバルパスワードポリシーの対象になります。サーバーは、定義されたサブツリー/ユーザーレベルのパスワードポリシーを無視します。

この属性の値が **on** の場合、サーバーはサブツリーおよびユーザーレベルでパスワードポリシーをチェックし、これらのポリシーを適用します。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-pwpolicy-local: off

### 3.1.1.140. nsslapd-readonly(読み取り専用)

この属性は、サーバー全体が読み取り専用モードであるかどうかを設定します。つまり、データベースにはデータも設定情報も変更できません。データベースを読み取り専用モードで変更しようとする、サーバーが操作を実施しないようにするエラーが返されます。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString



パラメーター	説明
例	nsslapd-readonly: off

### 3.1.1.141. nsslapd-referral (Referral)

この多値属性は、サーバーがローカルツリーに属さないエントリーの要求を受信するときに接尾辞によって返される LDAP URL を指定します。つまり、接尾辞を持つエントリーは接尾辞属性で指定された値と一致しません。たとえば、サーバーにエントリーのみが含まれるとします。

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

ただし、このエントリーに対する要求は、以下のとおりです。

```
ou=Groups,dc=example,dc=com
```

この場合、参照は LDAP クライアントが要求されたエントリーが含まれるサーバーを見つけようと試みます。Directory Server インスタンスごとに1つの参照のみが許可されますが、この参照は複数の値を持つことができます。



#### 注記

TLS 通信を使用するには、参照属性は **ldaps://server-location** の形式で指定する必要があります。

Start TLS は参照をサポートしません。

参照の管理に関する詳細は、**Red Hat Directory Server 管理ガイド**のディレクトリーデータベースの設定の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
有効な値	有効な LDAP URL
デフォルト値	
構文	DirectoryString
例	nsslapd-referral: ldap://ldap.example.com/dc=example,dc=com

### 3.1.1.142. nsslapd-referralmode (参照モード)

これが設定されている場合、この属性は接尾辞の任意のリクエストの参照を返します。

パラメーター	説明
エントリー DN	cn=config
有効な値	有効な LDAP URL
デフォルト値	
構文	DirectoryString
例	nsslapd-referralmode: ldap://ldap.example.com

### 3.1.1.143. nsslapd-require-secure-binds

このパラメーターでは、ユーザーは、通常の接続ではなく、TLS、StartTLS、SASL などの保護された接続でディレクトリーに対して認証する必要があります。



#### 注記

これは認証されたバインドにのみ適用されます。**nsslapd-require-secure-binds** がオンの場合でも、匿名バインドと非認証バインドは、標準チャンネルで完了できます。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-require-secure-binds: on

### 3.1.1.144. nsslapd-requiresrestart

このパラメーターには、変更後にサーバーを再起動する必要があるその他のコア設定属性がリスト表示されます。これは、**nsslapd-requiresrestart** にリスト表示される属性が変更された場合、サーバーが再起動するまで新しい設定が有効にならないことを意味します。属性のリストは、**ldapsearch** で返すことができます。

```
ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b "cn=config" -s sub -x "(objectclass=*)" | grep nsslapd-requiresrestart
```

この属性は多値です。

パラメーター	説明
エントリー DN	cn=config
有効な値	コアサーバー設定属性
デフォルト値	
構文	DirectoryString
例	nsslapd-requiresrestart: nsslapd-cachesize

### 3.1.1.145. nsslapd-reservedescriptors(予約ファイル記述子)

この属性は、インデックス管理やレプリケーションの管理など、Directory Server がクライアント接続以外の管理用に予約するファイル記述子の数を指定します。サーバーがこの目的のファイル記述子に対して予約するファイル記述子の数は、LDAP クライアント接続を提供するために利用可能なファイル記述子の合計数 (「[nsslapd-maxdescriptors\(最大ファイル記述子\)](#)」を参照) から減ります。

Directory Server のほとんどのインストールでは、この属性を変更する必要はありません。ただし、以下がすべて該当する場合には、この属性の値を増やすことを検討してください。

- サーバーは、多数のコンシューマーサーバーに複製する (10 以上)、またはサーバーが多数のインデックスファイル (30 以上) を維持している。
- サーバーは多数の LDAP 接続を提供します。
- サーバーがファイル記述子を開けないことを報告するエラーメッセージがあります (実際のエラーメッセージは、サーバーが実行しようとしている操作によって異なりますが、これらのエラーメッセージは **クライアントの LDAP 接続の管理とは関係ありません**)。

この属性の値を増やすと、より多くの LDAP クライアントがディレクトリーにアクセスできない可能性があります。したがって、この属性の値は増加し、**nsslapd-maxdescriptors** 属性の値も増やします。オペレーティングシステムでプロセスの使用を許可するファイル記述子の最大数を使用している場合は、**nsslapd-maxdescriptors** 値を増やすことができない可能性があります。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。この場合、LDAP クライアントが代替ディレクトリーレプリカを検索することで、サーバーの負荷を軽減します。受信接続のファイル記述子の使用については、「[nsslapd-conntablesize](#)」を参照してください。

この属性に設定されたファイル記述子の数を計算するには、以下の式を使用します。

$$\text{nsslapd-reservedescriptor} = 20 + (\text{NldbmsBackends} * 4) + \text{NglobalIndex} + \text{ReplicationDescriptor} + \text{ChainingBackendDescriptors} + \text{PTADescriptors} + \text{SSLDescriptors}$$

- **NldbmsBackends** は、ldbm データベースの数です。
- **NglobalIndex** は、システムインデックスを含むすべてのデータベースに設定されたインデックスの合計数です。(デフォルトでは 8 のシステムインデックスと、データベースごとに追加インデックス 17)
- **ReplicationDescriptor** は、8 に加えて、サプライヤーやハブとして機能するサーバー内のレプリカの数 (NSupplierReplica) です。

- **ChainingBackendDescriptors** は、**NchainingBackend** に **nsOperationConnectionsLimit** (チェーンまたはデータベースリンクの設定属性で、デフォルトは **10**) をかけたものです。
- **PTADescriptors** は、PTA が設定されている場合は **3**、PTA が設定されていない場合は **0** です。
- TLS が設定されている場合、**SSLDescriptors** は **5** (4 ファイル + 1 listensocket) であり、TLS が設定されていない場合には **0** になります。

この属性の変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=config
Valid Range	1 から 65535
デフォルト値	64
構文	整数
例	nsslapd-reservedescriptors: 64

### 3.1.1.146. nsslapd-return-exact-case (完全に一致したケースを返す)

クライアントによって要求される属性タイプ名の正確なケースを返します。LDAPv3 準拠のクライアントは属性名のケースを無視する必要がありますが、一部のクライアントアプリケーションは、検索や変更操作の結果として Directory Server によって属性が返される際に、その属性がスキーマに記載されているとおりに属性名が一致する必要があります。ただし、ほとんどのクライアントアプリケーションは属性に大文字と小文字を無視します。したがって、デフォルトではこの属性は無効になっています。サーバーから返される属性名のケースを確認するレガシークライアントがない限り、変更しないでください。

この属性の変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-return-exact-case: off

### 3.1.1.147. nsslapd-rewrite-rfc1274

この属性は非推奨となり、今後のバージョンで削除されます。

この属性は、RFC 1274 の名前で属性タイプを返す必要がある LDAPv2 クライアントにのみ使用されません。これらのクライアントで値を **on** に設定します。デフォルトは **off** です。

### 3.1.1.148. nsslapd-rootdn (マネージャー DN)

この属性は、アクセス制御の制限を受けないエントリーの識別名 (DN)、ディレクトリーの操作に対する管理制限、または一般的にリソース制限を設定します。この DN に対応するエントリーは必要ありません。デフォルトではこの DN のエントリーはありません。したがって、**cn=Directory Manager** などの値は受け入れ可能です。

ルート DN の変更に関する詳細は、**Red Hat Directory Server 管理ガイド** のディレクトリーエントリーの作成の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
有効な値	有効な識別名
デフォルト値	
構文	DN
例	nsslapd-rootdn: cn=Directory Manager

### 3.1.1.149. nsslapd-rootpw (Root パスワード)

この属性は、Manager DN に関連付けられたパスワードを設定します。root パスワードを指定すると、**nsslapd-rootpwstoragescheme** 属性に選択した暗号化方法に従って暗号化されます。サーバーコンソールから表示すると、この属性に値 \* が表示されます。**dse.ldif** ファイルから表示すると、この属性には、暗号化方法の後にパスワードの暗号化された文字列が表示されます。この例は、実際のパスワードではなく、**dse.ldif** ファイルに表示されるパスワードを示しています。



#### 警告

ルート DN がサーバーの設定になっている場合は、root パスワードが必要です。ただし、ファイルを直接編集して、root パスワードを **dse.ldif** から削除できます。この場合、ルート DN は、匿名のアクセスに対してはディレクトリーへの同じアクセスのみを取得できます。Root DN がデータベースに対して設定されている場合、root パスワードが **dse.ldif** で定義されているようにしてください。**pwdhash** コマンドラインユーティリティは、新しい root パスワードを作成できます。詳細は、「[pwdhash](#)」を参照してください。



## 重要

コマンドラインから Directory Manager のパスワードをリセットする場合は、パスワードに中括弧 ({} ) を使用しないでください。Root パスワードは {password-storage-scheme}hashed\_password 形式で保存されます。中括弧内の文字は、サーバーによって root パスワードストレージスキームとして解釈されます。そのテキストが有効なストレージスキームではない場合や、次のパスワードが適切にハッシュ化されない場合、Directory Manager はサーバーにバインドできません。

パラメーター	説明
エントリー DN	cn=config
有効な値	<a href="#">「パスワードストレージスキーム」</a> で説明されている暗号化方法のいずれかで暗号化されている有効なパスワード。
デフォルト値	
構文	DirectoryString {encryption_method} {encrypted_Password}
例	nsslapd-rootpw: {SSHA}9Eko69APCJfF

### 3.1.1.150. nsslapd-rootpwstoragescheme (Root パスワードストレージスキーム)

この属性は、**nsslapd-rootpw** 属性に保存されている Directory Server のマネージャーパスワードを暗号化する方法を設定します。強固なパスワードストレージスキームなどの詳細は、[「パスワードストレージスキーム」](#) を参照してください。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	<a href="#">「パスワードストレージスキーム」</a> を参照してください。
Default Value	PBKDF2_SHA256
Syntax	DirectoryString
Example	nsslapd-rootpwstoragescheme: PBKDF2_SHA256

### 3.1.1.151. nsslapd-rundir

このパラメーターは、Directory Server が PID ファイルなどのランタイム情報を保存するディレクトリーへの絶対パスを設定します。ディレクトリーは Directory Server のユーザーおよびグループが所有

する必要があります。このユーザーおよびグループは、このディレクトリーに読み取りおよび書き込みアクセスを持つ必要があるだけです。

この属性への変更を反映するには、サービスを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=config
有効な値	Directory Server ユーザーが書き込み可能なディレクトリー
デフォルト値	/var/run/dirsrv/
構文	DirectoryString
例	nsslapd-rundir: /var/run/dirsrv/

### 3.1.1.152. nsslapd-sasl-mapping-fallback

デフォルトでは、最初に一致する SASL マッピングのみがチェックされます。このマッピングに失敗すると、機能する可能性のあるマッピングが他にあっても、バインド操作は失敗します。SASL マッピングフォールバックは、一致するすべてのマッピングをチェックし続けます。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-sasl-mapping-fallback: off

### 3.1.1.153. nsslapd-sasl-max-buffer-size

この属性は、最大 SASL バッファサイズを設定します。

パラメーター	説明
エントリー DN	cn=config
有効な値	0 から最大 32 ビットの整数値 (2147483647)

パラメーター	説明
デフォルト値	67108864 (64 キロバイト)
構文	整数
例	nsslapd-sasl-max-buffer-size: 67108864

### 3.1.1.154. nsslapd-saslpath

Cyrus-SASL SASL2 プラグインを含むディレクトリーに絶対パスを設定します。この属性を設定すると、サーバーはカスタムまたは非標準の SASL プラグインライブラリーを使用できます。通常、これはインストール時に正しく設定され、Red Hat ではこの属性を変更しないことを強く推奨します。属性が存在しないか、値が空の場合は、Directory Server は、正しいバージョンであるシステムによって提供される SASL プラグインライブラリーを使用していることを意味します。

このパラメーターが設定されている場合、サーバーは SASL プラグインを読み込むために指定されたパスを使用します。このパラメーターが設定されていない場合、サーバーは **SASL\_PATH** 環境変数を使用します。**nsslapd-saslpath** または **SASL\_PATH** が設定されていない場合、サーバーはデフォルトの場所である **/usr/lib/sasl2** から SASL プラグインの読み込みを試行します。

この属性への変更は、サーバーが再起動するまで反映されません。

パラメーター	説明
エントリー DN	cn=config
有効な値	プラグインディレクトリーへのパス。
デフォルト値	プラットフォーム依存
構文	DirectoryString
例	nsslapd-saslpath: /usr/lib/sasl2

### 3.1.1.155. nsslapd-schema-ignore-trailing-spaces(オブジェクトクラス名の後続スペースを無視する)

オブジェクトクラス名の末尾を無視します。デフォルトでは、属性はオフになっています。ディレクトリーに、1つ以上のスペースで終わるオブジェクトクラス値を持つエントリーが含まれている場合は、この属性をオンにします。LDAP 標準では許可しないため、末尾のスペースを削除することが推奨されます。

パフォーマンス上の理由から、変更を反映するには、サーバーを再起動する必要があります。

末尾のスペースを含むオブジェクトクラスがエントリーに追加されると、デフォルトでエラーが返されます。さらに、(オブジェクトクラスの拡張および不明な場合)add、modify、および import などの操作時に、末尾のスペースは無視されます (適切な場合)。これは、**nsslapd-schema-ignore-trailing-**



**spaces** を **on** にした場合でも、**top** がすでに存在している場合に、**top** のような値が追加されないことを意味します。オブジェクトクラスが見つからない場合にエラーメッセージをログに記録し、クライアントに返し、末尾のスペースが含まれます。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-schema-ignore-trailing-spaces: on

### 3.1.1.156. nsslapd-schemacheck (スキーマチェック)

この属性は、エントリーが追加または変更されたときにデータベーススキーマを適用するかどうかを設定します。この属性の値が **on** の場合、Directory Server は変更されるまで既存のエントリーのスキーマを確認しません。データベーススキーマでは、データベースで許可される情報のタイプを定義します。デフォルトのスキーマは、オブジェクトクラスおよび属性タイプを使用して拡張できます。Directory Server コンソールを使用してスキーマを拡張する方法は、**Red Hat Directory Server 管理ガイド**のディレクトリースキーマの拡張の章を参照してください。



#### 警告

Red Hat は、スキーマチェックをオフにしないことを強く推奨します。これにより、深刻な相互運用性の問題が発生する可能性があります。これは通常、Directory Server にインポートする必要がある、非常に古い、または標準以外の LDAP データに使用されます。この問題の影響を受けるエントリーが多数ある場合は、これらのエントリーに **extensibleObject** オブジェクトクラスを使用してエントリーごとにスキーマチェックを無効にすることを検討してください。



#### 注記

スキーマのチェックは、**ldapmodify** などの LDAP クライアントを使用してデータベースが変更された場合や、**ldif2db** を使用して LDIF からデータベースをインポートする際にデフォルトで機能します。スキーマチェックをオフにする場合は、すべてのエントリーを手動で検証して、スキーマに準拠することを確認する必要があります。スキーマチェックが有効な場合、サーバーはエラーメッセージをリストし、スキーマに一致しないエントリーをリスト表示します。LDIF ステートメントで作成された属性とオブジェクトクラスの両方がスペルが正しく、**dse.ldif** で識別されていることを確認します。スキーマディレクトリーに LDIF ファイルを作成するか、その要素を **99user.ldif** に追加します。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-schemacheck: on

### 3.1.1.157. nsslapd-schemadir

これは、Directory Server インスタンス固有のスキーマファイルを含むディレクトリーへの絶対パスです。サーバーが起動すると、このディレクトリーからスキーマファイルを読み取ります。スキーマが LDAP ツールで変更されると、このディレクトリーのスキーマファイルが更新されます。このディレクトリーはサーバーユーザー ID で所有され、そのユーザーにはディレクトリーへの読み書きパーミッションがなければなりません。

この属性への変更は、サーバーが再起動するまで反映されません。

パラメーター	説明
エントリー DN	cn=config
有効な値	有効なパス
デフォルト値	/etc/dirsrv/ <b>instance_name</b> /schema
構文	DirectoryString
例	nsslapd-schemadir: /etc/dirsrv/ <b>instance_name</b> /schem

### 3.1.1.158. nsslapd-schemamod

オンラインスキーマの変更には、パフォーマンスに影響するロック保護が必要です。スキーマの変更が無効になっている場合は、このパラメーターを **off** に設定するとパフォーマンスが向上します。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on

パラメーター	説明
構文	DirectoryString
例	nsslapd-schemamod: on

### 3.1.1.159. nsslapd-schemareplace

**cn=schema** エントリーで属性値を置き換える変更操作が許可されるかどうかを決定します。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off   replication-only
デフォルト値	replication-only
構文	DirectoryString
例	nsslapd-schemareplace: replication-only

### 3.1.1.160. nsslapd-search-return-original-type-switch

検索に渡される属性リストにスペースと他の文字が含まれる場合には、同じ文字列がクライアントに戻されます。以下に例を示します。

```
# ldapsearch -b <basedn> "(filter)" "sn someothertext"
dn: <matched dn>
sn someothertext: <sn>
```

この動作はデフォルトでは無効にされますが、この設定パラメーターを使用して有効にできます。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-search-return-type-switch: off

### 3.1.1.161. nsslapd-securelistenhost

この属性により、複数の Directory Server インスタンスがマルチホームのマシンで実行できるようになります (または、マルチホームマシンの1つのインターフェイスのリッスンを制限することができます)。単一のホスト名に関連付けられる IP アドレスは複数あり、これらの IP アドレスは、IPv4 と IPv6 の両方の組み合わせになります。このパラメーターを使用して、Directory Server インスタンスを単一の IP インターフェイスに制限することができます。また、このパラメーターは、通常の LDAP 接続ではなく、TLS トラフィックに使用するインターフェイスを設定します。

ホスト名が **nsslapd-securelistenhost** 値として指定される場合、Directory Server はホスト名に関連付けられたすべてのインターフェイスについて要求に応答します。単一の IP インターフェイス (IPv4 または IPv6) が **nsslapd-securelistenhost** の値として指定される場合、Directory Server は、その特定のインターフェイスに送信された要求にのみ応答します。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。

この属性の変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=config
有効な値	セキュアなホスト名、IPv4 アドレスまたは IPv6 アドレス
デフォルト値	
構文	DirectoryString
例	nsslapd-securelistenhost: ldaps.example.com

### 3.1.1.162. nsslapd-securePort (暗号化されたポート番号)

この属性は、TLS 通信に使用される TCP/IP ポート番号を設定します。この選択したポートは、ホストシステムで一意でなければなりません。他のアプリケーションが同じポート番号を使用しないようにしてください。ポート番号が **1024** 未満の場合は、Directory Server を **root** で起動する必要があります。サーバーは、起動後にその **uid** を **nsslapd-localuser** 値に設定します。

サーバーは、秘密鍵と証明書で設定され、**nsslapd-security** が **on** に設定されている場合にのみこのポートをリッスンします。それ以外の場合は、このポートでリッスンしません。

ポート番号の変更を考慮してサーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=config
Valid Range	1 から 65535
デフォルト値	636
構文	整数

パラメーター	説明
例	nsslapd-securePort: 636

### 3.1.1.163. nsslapd-security(セキュリティー)

この属性は、Directory Server が暗号化されたポートで TLS 通信を受け入れるかどうかを設定します。この属性は、セキュアな接続に対して **on** に設定する必要があります。セキュリティー上で実行するには、他の TLS 設定に加えて、秘密鍵とサーバー証明書でサーバーを設定する必要があります。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-security: off

### 3.1.1.164. nsslapd-sizelimit (サイズ制限)

この属性は、検索操作から返すエントリーの最大数を設定します。この制限に達すると、**ns-slapd** は検索要求に一致するエントリーと、超過サイズ制限エラーを返します。

制限が設定されていない場合、**ns-slapd** は見つかった数に関係なく、一致するすべてのエントリーをクライアントに返します。Directory Server が検索が完了するまで無期限に待機する制限値を設定するには、**dse.ldif** ファイルのこの属性に **-1** の値を指定します。

この制限は、組織に関係なくすべてのユーザーに適用されます。



#### 注記

Dse **.ldif** ファイルのこの属性に対する **-1** の値は、サーバーコンソールで属性を空白のままにしておくのと同じため、制限は使用されません。これは有効な整数ではないため、**dse.ldif** ファイルには null 値を指定できません。**0** に設定すると、検索ごとに **size limit exceeded** が返されます。

対応するユーザーレベルの属性は **nsSizeLimit** です。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1 から最大 32 ビットの整数値 (2147483647)

パラメーター	説明
デフォルト値	2000
構文	整数
例	nsslapd-sizelimit: 2000

### 3.1.1.165. nsslapd-snmpp-index

このパラメーターは、Directory Server インスタンスの SNMP インデックス番号を制御します。

ポート 389 で、ポート 389 がすべてリッスンしている同じホストに複数の Directory Server インスタンスがある場合、このパラメーターを使用すると、インスタンスごとに異なる SNMP インデックス番号を設定できます。

パラメーター	説明
エントリー DN	cn=config
有効な値	0 から最大 32 ビットの整数値 (2147483647)
デフォルト値	0
構文	整数
例	nsslapd-snmpp-index: 0

### 3.1.1.166. nsslapd-SSLclientAuth



#### 注記

**Nsslapd-SSLclientAuth** パラメーターは今後のリリースで非推奨となり、現時点では後方互換性のために維持されます。代わりに **cn=encryption,cn=config** に保存されている新しいパラメーター **nsSSLClientAuth** を使用してください。 [「nsSSLClientAuth」](#) を参照してください。

### 3.1.1.167. nsslapd-ssl-check-hostname(アウトバウンド接続のホスト名を確認)

この属性は、提示される証明書のサブジェクト名 (**subjectDN** フィールド) の共通名 (**cn**) 属性に割り当てられた値に対してホスト名を照合することにより、TLS 対応の Directory Server が要求の信頼性を検証するかどうかを設定します。デフォルトでは、属性は **on** に設定されます。有効で、ホスト名が証明書の **cn** 属性と一致しない場合は、適切なエラーと監査メッセージがログに記録されます。

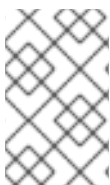
たとえば、複製された環境では、ピアサーバーのホスト名が証明書で指定された名前と一致しないと、以下のようなメッセージがサプライヤーサーバーのログファイルに記録されます。

```
[DATE] - SSL alert: ldap_sasl_bind("",LDAP_SASL_EXTERNAL) 81 (Netscape runtime error -12276 -
Unable to communicate securely with peer: requested domain name does not
```

match the server's certificate.)

```
[DATE] NSMMReplicationPlugin - agmt="cn=SSL Replication Agreement to host1"
(host1.example.com:636):
Replication bind with SSL client authentication failed:
LDAP error 81 (Can't contact LDAP server)
```

Red Hat は、MITM (MITM) 攻撃で、Directory Server のアウトバウンド TLS 接続を保護するために、この属性をオンにすることを推奨します。



### 注記

これを機能させるには、DNS と逆引き DNS が正しく設定されている必要があります。そうしないと、サーバーは、証明書のサブジェクト名に対してピア IP アドレスを、証明書のサブジェクト DN で解決できません。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-ssl-check-hostname: on

### 3.1.1.168. nsslapd-syntaxcheck

この属性は、エントリー属性に対するすべての変更を検証し、新規または変更された値がその属性タイプに必要な構文に準拠することを確認します。この属性が有効になっていると、適切な構文に準拠しない変更は拒否されます。すべての属性値は [RFC 4514](#) の構文定義に対して検証されます。

デフォルトでは、これはオンになっています。

構文の検証は、新規または変更された属性に対してのみ実行されます。既存の属性値の構文は検証されません。追加や変更などの LDAP 操作の構文検証がトリガーされます。元のサプライヤーで属性構文の有効性をチェックする必要があるため、レプリケーションなどの操作後には起こりません。

これは、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、Directory Server でサポートされる属性タイプをすべて検証します。未検証の構文は以下のとおりです。

- Fax (バイナリー)
- OctetString (binary)
- JPEG (バイナリー)
- バイナリー (標準以外)

- スペースに依存しない文字列 (非標準)
- URI (標準以外)

**Nsslapd-syntaxcheck** 属性は、属性の変更を検証および拒否するかどうかを設定します。これは、「[nsslapd-syntaxlogging](#)」属性とともに使用して、無効な属性値に関する警告メッセージをエラーログに書き込むことができます。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-syntaxcheck: on

### 3.1.1.169. nsslapd-syntaxlogging

この属性は、構文検証の失敗をエラーログに記録するかどうかを設定します。デフォルトでは、これはオフになっています。

「[nsslapd-syntaxcheck](#)」属性が有効 (デフォルト) で、**nsslapd-syntaxlogging** 属性も有効になっている場合、無効な属性の変更は拒否され、エラーログに書き込まれます。**nsslapd-syntaxlogging** のみが有効で、**nsslapd-syntaxcheck** が無効になっている場合は、無効な変更を続行できますが、警告メッセージがエラーログに書き込まれます。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-syntaxlogging: off

### 3.1.1.170. nsslapd-threadnumber (しきい値)

このパフォーマンスチューニング関連の値は、起動時に Directory Server が作成するスレッドの数を設定します。値が **-1** (デフォルト) に設定されている場合、Directory Server は利用可能なハードウェアに基づいて最適化された自動チューニングを有効にします。Auto-tuning が有効になっている場合、**nsslapd-threadnumber** は、Directory Server の実行中に自動生成されたスレッド数を表示することに注意してください。





## 注記

Red Hat は、パフォーマンスを最適化するために自動チューニング設定を使用することを推奨します。

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。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1からシステムのスレッドおよびプロセッサでサポートされるスレッドの最大数。
デフォルト値	-1
構文	整数
例	nsslapd-threadnumber: -1

### 3.1.1.171. nsslapd-timelimit(時間制限)

この属性は、検索要求に割り当てられる最大秒数を設定します。この制限に達すると、Directory Server は、検索要求に一致するエントリーと、超過時間制限エラーを返します。

制限が設定されていない場合、**ns-slapd** は完了する時間に関係なく、一致するすべてのエントリーをクライアントに返します。Directory Server が検索が完了するまで無期限に待機する制限値を設定するには、**dse.ldif** ファイルでこの属性に **-1** の値を指定します。**0** ゼロの値を指定すると、検索に時間が許可されません。最小の時間制限は1秒です。



## 注記

**dse.ldif** のこの属性に対する **-1** の値は、サーバーコンソールで属性を空白のままにしておくのと同じため、制限が使用されなくなります。ただし、サーバーコンソールではこのフィールドで負の整数を設定できず、有効な整数ではないため、**dse.ldif** エントリーに null 値を使用することはできません。

対応するユーザーレベルの属性は **nsTimeLimit** です。

パラメーター	説明
エントリー DN	cn=config
Valid Range	-1から最大 32 ビットの整数値 (2147483647)(秒単位)
デフォルト値	3600

パラメーター	説明
構文	整数
例	nsslapd-timelimit: 3600

### 3.1.1.172. nsslapd-tmpdir

これは、サーバーが一時ファイルに使用するディレクトリーの絶対パスです。ディレクトリーはサーバーユーザー ID で所有され、ユーザーには読み取りおよび書き込みアクセスが必要です。他のユーザー ID はディレクトリーに読み取りや書き込みを行うべきではありません。デフォルト値は **/tmp** です。

この属性への変更は、サーバーが再起動するまで反映されません。

### 3.1.1.173. nsslapd-unhashed-pw-switch

**userPassword** 属性を更新すると、Directory Server はパスワードを暗号化し、**userPassword** に保存します。ただし、Active Directory (AD) とパスワードを同期する場合など、特定の状況では、Directory Server は暗号化されていないパスワードをプラグインに渡す必要があります。この場合、サーバーは、**entry extension** と呼ばれる一時的な **unhashed#user#password** 属性に暗号化されていないパスワードを保存し、シナリオに応じて changelog にも格納します。Directory Server は、サーバーのハードディスクに **unhashed#user#password** 属性を保存しないことに注意してください。

**nsslapd-unhashed-pw-switch** パラメーターは、Directory Server が暗号化されていないパスワードを保存するかどうかと方法を制御します。たとえば、Directory Server から Active Directory にパスワードを同期するには、**nsslapd-unhashed-pw-switch** を **on** に設定する必要があります。

パラメーターは以下のいずれかの値に設定できます。

- **off**: Directory Server は、暗号化されていないパスワードをエントリー拡張や changelog に保存しません。AD とパスワードの同期を使用しない場合や、暗号化されていないパスワードへのアクセスを必要とするプラグインを使用する場合は、この値を設定します。
- **on**: Directory Server は、暗号化されていないパスワードをエントリー拡張と changelog に保存します。AD とパスワードの同期を設定する場合は、この値を設定します。
- **nolog**: Directory Server は、暗号化されていないパスワードをエントリー拡張にのみ保存しますが、changelog には保存しません。ローカルの Directory Server プラグインが暗号化されていないパスワードへのアクセスを必要とするが、AD とパスワードの同期を設定しない場合は、この値を設定します。

パラメーター	説明
エントリー DN	cn=config
有効な値	off   on   nolog
デフォルト値	off
構文	DirectoryString

パラメーター	説明
例	nsslapd-unhashed-pw-switch: off

### 3.1.1.174. nsslapd-validate-cert

Directory Server が TLS で実行されるように設定され、証明書の有効期限が切れると、Directory Server を起動できません。**nsslapd-validate-cert** パラメーターは、期限切れの証明書で起動しようとすると Directory Server がどのように応答するかを設定します。

- **warn** により、Directory Server は期限切れの証明書で正常に起動できますが、証明書の有効期限が切れているという警告メッセージが送信されます。これはデフォルト設定です。
- **on** では、証明書を検証します。また、証明書の有効期限が切れるとサーバーが再起動できなくなります。これにより、期限切れの証明書のハード障害が設定されます。
- **off** は、すべての証明書の有効期限の検証を無効にするため、サーバーは警告をログに記録せずに期限切れの証明書で起動できるようにします。

パラメーター	説明
エントリー DN	cn=config
有効な値	warn   on   off
デフォルト値	warn
構文	DirectoryString
例	nsslapd-validate-cert: warn

### 3.1.1.175. nsslapd-verify-filter-schema

**nsslapd-verify-filter-schema** パラメーターは、Directory Server がスキーマで指定されていない属性で検索フィルターを検証する方法を定義します。

**nsslapd-verify-filter-schema** を以下のオプションのいずれかに設定できます。

- **reject-invalid**: Directory Server は、不明な要素が含まれる場合、エラーを出してフィルターを拒否します。
- **process-safe**: Directory Server は不明なコンポーネントを空のセットに置き換え、警告を `/var/log/dirsrv/slapd-instance_name/access` ログファイルの **notes=F** フラグで記録します。**nsslapd-verify-filter-schema** を **warn-invalid** または **off** から **process-safe** に切り替える前に、アクセスログを監視し、**notes=F** フラグでログエントリを発生させるアプリケーションからのクエリーを修正してください。そうしないと、操作結果が変更され、Directory Server が一致するすべてのエントリーを返さない可能性があります。
- **warn-invalid**: Directory Server は、`/var/log/dirsrv/slapd-instance_name/access` ログファイル内の **notes=F** フラグで警告を記録し、完全なデータベースをスキャンし続けます。

- **off**: Directory Server はフィルターを検証しません。

たとえば、`nsslapd-verify-filter-schema` を `warn-invalid` または `off` に設定した場合、`(&(non_existent_attribute=example)(uid=user_name))` などのフィルターは `uid=user_name` エントリーを評価し、`non_existent_attribute=example` が含まれている場合にのみそれを返すことに注意してください。`nsslapd-verify-filter-schema` を `process-safe` に設定した場合、Directory Server はそのエントリーを評価せず、返しません。



### 注記

`nsslapd-verify-filter-schema` を `reject-invalid` または `process-safe` に設定すると、スキーマで指定されていない属性のインデックス付けされていない検索による高負荷を防ぐことができます。

パラメーター	説明
エントリー DN	cn=config
有効な値	reject-invalid、process-safe、warn-invalid、off
Default Value	warn-invalid
Syntax	DirectoryString
Example	nsslapd-verify-filter-schema: warn-invalid

### 3.1.1.176. nsslapd-versionstring

この属性は、サーバーのバージョン番号を設定します。バージョン文字列が表示されると、ビルドデータが自動的に追加されます。

パラメーター	説明
エントリー DN	cn=config
有効な値	有効なサーバーのバージョン番号。
デフォルト値	
構文	DirectoryString
例	nsslapd-versionstring: Red Hat-Directory/11.3

### 3.1.1.177. nsslapd-workingdir

これは、サーバーが起動後に現在の作業ディレクトリーとして使用するディレクトリーの絶対パスです。これは、サーバーが `getcwd()` 関数の値として返す値であり、システムプロセステーブルが現在の作業ディレクトリーとして表示する値です。これは、コアファイルが生成されるディレクトリーです。

サーバーのユーザー ID には、ディレクトリーへの読み取りおよび書き込みアクセス権が必要です。また、他のユーザー ID には、ディレクトリーへの読み取りまたは書き込みアクセス権がありません。この属性のデフォルト値は、エラーログを含む同じディレクトリーであり、通常は `/var/log/dirsrv/slapd-instance` です。

この属性への変更は、サーバーが再起動するまで反映されません。

### 3.1.1.178. passwordAllowChangeTime

この属性は、ユーザーがパスワードを変更できるようになるまでに経過する必要がある時間の長さを指定します。

パスワードポリシーの詳細は、Red Hat Directory Server 管理ガイドのユーザー認証の管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
有効な値	任意の整数
デフォルト値	
構文	DirectoryString
例	passwordAllowChangeTime: 5h

### 3.1.1.179. passwordChange (パスワード変更)

ユーザーがパスワードを変更できるかどうかを示します。

これは、`pwdAllowUserChange` と省略できます。

パスワードポリシーの詳細は、Red Hat Directory Server 管理ガイドのユーザー認証の管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	passwordChange: on

### 3.1.1.180. passwordCheckSyntax (パスワード構文チェック)

この属性は、パスワードを保存する前にパスワード構文をチェックするかどうかを設定します。パスワードの構文チェックメカニズムは、パスワードがパスワードの最小長要件を満たしているかどうか、また、文字列にユーザー名やユーザー ID、ユーザーのディレクトリーエントリーの **uid**、**cn**、**sn**、**givenName**、**ou**、または **mail** 属性に格納されている属性値など、簡単な単語が含まれていないかどうかをチェックします。

パスワード構文には、チェック用のいくつかの異なるカテゴリーが含まれています。

- パスワード内の普通の単語をチェックするときに比較するのに使用する文字列またはトークンの長さ (たとえば、トークンの長さが 3 の場合、パスワードに使用するユーザーの UID、名前、電子メールアドレス、またはその他のパラメーターに 3 つの連続する文字の文字列を含めることはできません)
- 数字の最小文字数 (0~9)
- 大文字の ASCII アルファベットの最小数
- 小文字の ASCII アルファベットの最小数
- **!@#\$** などの特殊 ASCII 文字の最小数
- 8 ビット文字の最小数
- パスワードごとに必要な文字カテゴリーの最小数。カテゴリーは大文字、小文字、特殊文字、数字、または 8 ビット文字

これは、**pwdCheckSyntax** と省略できます。

パスワードポリシーの詳細は、**Red Hat Directory Server 管理ガイド**のユーザー認証の管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	passwordCheckSyntax: off

### 3.1.1.181. passwordDictCheck

**on** に設定すると、**passwordDictCheck** パラメーターはパスワードを **CrackLib** デクショナリーと照合します。新しいパスワードに辞書の単語が含まれている場合、Directory Server はパスワードを拒否します。

パラメーター	説明
エントリー DN	cn=config

パラメーター	説明
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	passwordDictCheck: off

### 3.1.1.182. passwordExp (パスワードの有効期限)

指定された秒数後にユーザーパスワードの有効期限が切れるかどうかを示します。デフォルトでは、ユーザーパスワードは期限切れになりません。パスワードの有効期限が有効になったら、**passwordMaxAge** 属性を使用して、パスワードの有効期限が切れるまでの秒数を設定します。

パスワードポリシーの詳細は、**Red Hat Directory Server 管理ガイド**のユーザーアカウントの管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	passwordExp: on

### 3.1.1.183. passwordExpirationTime

この属性は、ユーザーのパスワードの有効期限が切れるまでに経過する時間の長さを指定します。

パラメーター	説明
エントリー DN	cn=config
有効な値	任意の日付 (整数)
デフォルト値	none
構文	GeneralizedTime

パラメーター	説明
例	passwordExpirationTime: 202009011953

### 3.1.1.184. passwordExpWarned

この属性は、パスワードの有効期限の警告がユーザーに送信されたことを示します。

パラメーター	説明
エントリー DN	cn=config
有効な値	true   false
デフォルト値	none
構文	DirectoryString
例	passwordExpWarned: true

### 3.1.1.185. passwordGraceLimit (パスワードの有効期限)

この属性は、パスワードの有効期限が有効になっている場合にのみ適用されます。ユーザーのパスワードの有効期限が切れると、サーバーはユーザーがパスワードを変更する目的で接続できるようにします。これは、**猶予ログイン**と呼ばれます。サーバーは、ユーザーを完全にロックアウトする前に、特定の回数の試行のみを許可します。この属性は、許可される猶予ログインの数です。**0**の値は、サーバーが猶予ログインを許可しないことを意味します。

パラメーター	説明
エントリー DN	cn=config
有効な値	0 (オフ) から任意の妥当な整数
デフォルト値	0
構文	整数
例	passwordGraceLimit: 3

### 3.1.1.186. passwordHistory (パスワード履歴)

パスワード履歴を有効にします。パスワード履歴は、ユーザーがパスワードの再利用を許可されているかどうかを示します。デフォルトでは、パスワード履歴は無効になっており、ユーザーはパスワードを再利用できます。この属性が **on** に設定されている場合、ディレクトリーは指定された数の古いパス



ワードを保存し、ユーザーが保存されたパスワードを再利用できないようにします。**passwordInHistory** 属性を使用して、Directory Server が保存する古いパスワードの数を設定します。

パスワードポリシーの詳細は、**Red Hat Directory Server 管理ガイド**のユーザー認証の管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	passwordHistory: on

### 3.1.1.187. passwordInHistory (覚えておくべきパスワードの数)

Directory Server が履歴に保存するパスワードの数を示します。履歴に保存されているパスワードは、ユーザーが再利用することはできません。デフォルトでは、パスワード履歴機能は無効になっています。つまり、Directory Server は古いパスワードを保存しないため、ユーザーはパスワードを再利用できます。**passwordHistory** 属性を使用してパスワード履歴を有効にします。

ユーザーが追跡されるパスワードの数をすばやく循環するのを防ぐには、**passwordMinAge** 属性を使用します。

これは、**pwdInHistory** と省略できます。

パスワードポリシーの詳細は、**Red Hat Directory Server 管理ガイド**のユーザー認証の管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
Valid Range	1 から 24 個のパスワード
デフォルト値	6
構文	整数
例	passwordInHistory: 7

### 3.1.1.188. passwordIsGlobalPolicy (パスワードポリシーとレプリケーション)

この属性は、パスワードポリシー属性を複製するかどうかを制御します。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	passwordIsGlobalPolicy: off

### 3.1.1.189. passwordLegacyPolicy

従来のパスワードの動作を有効にします。古い LDAP クライアントは、最大障害制限を **超える** と、ユーザーアカウントをロックするためのエラーを受け取ると予想されていました。たとえば、制限が 3 回失敗した場合は、4 回目の失敗でアカウントがロックされました。ただし、新しいクライアントは、障害制限に達したときにエラーメッセージを受信することを期待しています。たとえば、制限が 3 回失敗した場合、3 回目の失敗でアカウントをロックする必要があります。

障害制限を超えたときにアカウントをロックすることは古い動作であるため、レガシー動作と見なされます。これはデフォルトで有効になっていますが、無効にして、新しい LDAP クライアントが予想される時間にエラーを受信できるようにすることができます。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	passwordLegacyPolicy: on

### 3.1.1.190. passwordLockout (アカウントロックアウト)

バインドの試行が一定回数失敗した後、ユーザーがディレクトリーからロックアウトされているかどうかを示します。デフォルトでは、一連のバインド試行が失敗した後、ユーザーはディレクトリーからロックアウトされません。アカウントのロックアウトが有効になっている場合は、**passwordMaxFailure** 属性を使用して、ユーザーがロックアウトされるまでに失敗したバインドの試行回数を設定します。

これは、**pwdLockOut** と省略できます。

パスワードポリシーの詳細は、Red Hat Directory Server **管理ガイド**のユーザー認証の管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	passwordLockout: off

### 3.1.1.191. passwordLockoutDuration (Lockout Duration)

アカウントのロックアウト後にユーザーがディレクトリーからロックアウトされるまでの時間を秒単位で示します。アカウントのロックアウト機能は、ユーザーのパスワードを繰り返し推測してディレクトリーに分割しようとするハッカーから保護します。**passwordLockout** 属性を使用して、アカウントのロックアウト機能を有効または無効にします。

これは、**pwdLockoutDuration** と省略できます。

パスワードポリシーの詳細は、**Red Hat Directory Server 管理ガイド**のユーザー認証の管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
Valid Range	1 から最大 32 ビット整数値 (2147483647) (秒単位)
デフォルト値	3600
構文	整数
例	passwordLockoutDuration: 3600

### 3.1.1.192. passwordMaxAge (パスワードの最大年齢)

ユーザーパスワードの有効期限が切れるまでの秒数を示します。この属性を使用するには、**passwordExp** 属性を使用してパスワードの有効期限を有効にする必要があります。

これは、**pwdMaxAge** と省略できます。

パスワードポリシーの詳細は、**Red Hat Directory Server 管理ガイド**のユーザー認証の管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
Valid Range	1 から最大 32 ビット整数値 (2147483647) (秒単位)
デフォルト値	8640000 (100 日)
構文	整数
例	passwordMaxAge: 100

### 3.1.1.193. passwordBadWords

**passwordBadWords** パラメーターは、ユーザーがパスワードで使用できない文字列のコンマ区切りリストを定義します。

Directory Server は文字列の大文字と小文字を区別しません。

パラメーター	説明
エントリー DN	cn=config
有効な値	任意の文字列
デフォルト値	""
構文	DirectoryString
例	passwordBadWords: example

### 3.1.1.194. passwordMaxClassChars

**passwordMaxClassChars** パラメーターを **0** よりも大きな値に設定する場合に、Directory Server では、パラメーターに設定した値と同じカテゴリーの文字を連続して指定することができなくなります。有効にすると、Directory Server は以下のカテゴリーに含まれる、連続した文字をチェックします。

- 数字
- 英字
- 小文字
- 大文字

たとえば、**passwordMaxClassChars** を **3** に設定した場合には、**jdif** や **1947** などのパスワードは使用できません。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0(無効) から最大 32 ビットの整数 (2147483647)
デフォルト値	0
構文	整数
例	passwordMaxClassChars: 0

### 3.1.1.195. passwordMaxFailure (最大パスワードの失敗回数)

バインドの試行を何回失敗するとユーザーがディレクトリーからロックアウトされるかを指定します。デフォルトでは、アカウントのロックアウトは無効になっています。**passwordLockout** 属性を変更して、アカウントのロックアウトを有効にします。

これは、**pwdMaxFailure** と省略できます。

パスワードポリシーの詳細は、**Red Hat Directory Server 管理ガイド**のユーザー認証の管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
Valid Range	最大バインド失敗数 (1 以上の整数)
デフォルト値	3
構文	整数
例	passwordMaxFailure: 3

### 3.1.1.196. passwordMaxRepeats (パスワード構文)

パスワードに同じ文字を連続して指定できる最大回数。ゼロ (0) はオフです。整数値は、ある文字を指定の回数以上に使用したパスワードを拒否します。たとえば、**1** を指定すると、文字が複数回使用された場合 (**aa**)、**2** を指定すると、ある文字を複数回使用した場合に (**aaa**) 拒否されます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 から 64

パラメーター	説明
デフォルト値	0
構文	整数
例	passwordMaxRepeats: 1

### 3.1.1.197. passwordMaxSeqSets

**passwordMaxSeqSets** パラメーターを **0** よりも大きな値に設定すると、Directory Server は、このパラメーターで設定した長さを超えて、同じ文字列が複数回出現するパスワードを拒否します。たとえば、**passwordMaxSeqSets** を **2** に設定した場合には、パスワード **azXYZ\_XYZ-g** は、パスワードの中に **XYZ** が 2 回出現するため使用できません。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 (無効) から最大 32 ビット 整数値 (2147483647)
デフォルト値	0
構文	整数
例	passwordMaxSeqSets: 0

### 3.1.1.198. passwordMaxSequence

**passwordMaxSequence** パラメーターを **0** よりも大きな値に設定すると、Directory Server は、**passwordMaxSequence** に設定された値を超えて、同じ文字種が連続して出現するパスワードを拒否します。たとえば、パラメーターを **3** に設定すると、Directory Server は **1234** や **dcba** などの文字列を含むパスワードを拒否します。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 (無効) から最大 32 ビット 整数値 (2147483647)
デフォルト値	0
構文	整数
例	passwordMaxSequence: 0

### 3.1.1.199. passwordMin8Bit (パスワード構文)

これにより、パスワードに含める必要のある 8 ビット文字の最小数が設定されます。



#### 注記

これを使用するには、**userPassword** の 7 ビットチェックを無効にする必要があります。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 から 64
デフォルト値	0
構文	整数
例	passwordMin8Bit: 0

### 3.1.1.200. passwordMinAge (パスワードの最短有効期限)

ユーザーが次にパスワードを変更するまでに待機する必要がある秒数を指定します。この属性は **passwordInHistory** (記憶するパスワードの数) 属性と合わせて使用して、すぐにパスワードを循環して、以前のパスワードをもう一度使用できないようにします。0 の値は、ユーザーがすぐにパスワードを変更できることを示しています。

これは、**pwdMaxFailure** と省略できます。

パスワードポリシーの詳細は、**Red Hat Directory Server 管理ガイド**のユーザー認証の管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 から有効な最大整数
デフォルト値	0
構文	整数
例	passwordMinAge: 150

### 3.1.1.201. passwordMinAlphas (パスワード構文)

この属性は、英数字のパスワードに含める必要がある最小数を設定します。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 から 64
デフォルト値	0
構文	整数
例	passwordMinAlphas: 4

### 3.1.1.202. passwordMinCategories (パスワード構文)

これにより、パスワードで使用される文字カテゴリーの最小数が設定されます。カテゴリーは以下のとおりです。

- 小文字の英字
- 大文字の英字
- 数値
- \$ や punctuation marks など、特別な ASCII 文字
- 8 ビット文字

たとえば、この属性の値が **2** に設定され、ユーザーがパスワードを **aaaaa** に変更しようとする、小文字しか含まれないので、サーバーはパスワードを拒否します。**aAaAaA** のパスワードには大文字と小文字の2つのカテゴリーからの文字が含まれるため、このパスワードは指定できます。

デフォルトは **3** です。つまり、パスワード構文チェックが有効な場合は、有効なパスワードには3つの文字カテゴリーが必要です。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 から 5
デフォルト値	0
構文	整数
例	passwordMinCategories: 2

### 3.1.1.203. PasswordMinDigits (パスワード構文)

これにより、パスワードに含める必要のある数字の最小数が設定されます。



パラメーター	説明
エントリー DN	cn=config
Valid Range	0 から 64
デフォルト値	0
構文	整数
例	passwordMinDigits: 3

#### 3.1.1.204. passwordMinLength (パスワードの最小長)

この属性は、Directory Server ユーザーパスワード属性で使用する必要がある文字の最小数を指定します。一般的に、パスワードが短いほど解読されやすくなります。Directory Server では、強制的にパスワードの最小長を 8 文字にします。これは、解読が難しく、ユーザーがパスワードを書き留めなくても覚えられる長さです。

これは、**pwdMinLength** と省略できます。

パスワードポリシーの詳細は、**Red Hat Directory Server 管理ガイド**のユーザー認証の管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
Valid Range	2 - 512 文字
デフォルト値	8
構文	整数
例	passwordMinLength: 8

#### 3.1.1.205. PasswordMinLowers (パスワード構文)

この属性は、小文字のパスワードに含める必要のある最小数を設定します。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 から 64
デフォルト値	0

パラメーター	説明
構文	整数
例	passwordMinLowers: 1

### 3.1.1.206. PasswordMinSpecials (パスワード構文)

この属性は、パスワードに含める必要がある **特殊** 文字 (または英数字以外の文字) の最小数を設定します。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 から 64
デフォルト値	0
構文	整数
例	passwordMinSpecials: 1

### 3.1.1.207. PasswordMinTokenLength (パスワード構文)

この属性は、**簡単な** 単語チェックに使用される最小の属性値の長さを設定します。たとえば、**PasswordMinTokenLength** が **3** に設定されている場合には、ポリシーで **givenName** の **DJ** は、パスワードに **DJ** が含まれていても拒否されず、**givenName** の **Bob** が含まれるパスワードは拒否されません。

Directory Server は、以下の属性の値に対してトークンの最小長をチェックします。

- **uid**
- **cn**
- **sn**
- **givenName**
- **mail**
- **ou**

Directory Server が追加の属性を確認する必要がある場合は、**passwordUserAttributes** パラメーターで設定できます。詳細は [「passwordUserAttributes」](#) を参照してください。

パラメーター	説明
エントリー DN	cn=config
Valid Range	1 から 64
デフォルト値	3
構文	整数
例	passwordMinTokenLength: 3

### 3.1.1.208. PasswordMinUppers (パスワード構文)

これにより、パスワードに含める必要のある大文字の最小数が設定されます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	0 から 64
デフォルト値	0
構文	整数
例	passwordMinUppers: 2

### 3.1.1.209. passwordMustChange (パスワードを変更する必要があります)

Directory Server への初回のバインド時、または Manager DN でパスワードのリセット時に、ユーザーがパスワードを変更する必要があるかどうかを示します。

これは、**pwdMustChange** と省略できます。

パスワードポリシーの詳細は、**Red Hat Directory Server 管理ガイド**のユーザー認証の管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString

パラメーター	説明
例	passwordMustChange: off

### 3.1.1.210. passwordPalindrome

**passwordPalindrome** パラメーターを有効にすると、新しいパスワードに回文が含まれる場合に、Directory Server はパスワードを拒否します。

回文とは、**abc11cba** など、上から読んでも、下から読んでも同じである文字列を指します。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	passwordPalindrome: off

### 3.1.1.211. passwordResetFailureCount (パスワードの失敗回数のリセット)

パスワード障害カウンターがリセットされるまでの時間 (秒単位) を指定します。無効なパスワードがユーザーのアカウントから送信されるたびに、パスワードの失敗カウンターがインクリメントされます。**passwordLockout** 属性を **on** に設定すると、カウンターが **passwordMaxFailure** 属性で指定された失敗数に達すると、ユーザーはディレクトリーからロックされます (デフォルトでは **600** 秒)。**passwordLockoutDuration** 属性で指定された時間が経過すると、失敗カウンターはゼロ (**0**) にリセットされます。

これは、**pwdFailureCountInterval** と省略できます。

パスワードポリシーの詳細は、Red Hat Directory Server **管理ガイド**のユーザー認証の管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
Valid Range	1 から最大 32 ビット整数値 (2147483647) (秒単位)
デフォルト値	600
構文	整数
例	passwordResetFailureCount: 600

### 3.1.1.212. passwordUserAttributes

デフォルトでは、**passwordMinTokenLength** パラメーターにトークンの最小長を設定すると、Directory Server は特定の属性に対してのみトークンをチェックします。詳細は「[PasswordMinTokenLength \(パスワード構文\)](#)」を参照してください。

**passwordUserAttributes** パラメーターを使用すると、Directory Server がチェックする必要のある属性を追加でコンマ区切りリストとして設定できます。

パラメーター	説明
エントリー DN	cn=config
有効な値	任意の文字列
デフォルト値	""
構文	DirectoryString
例	passwordUserAttributes: telephoneNumber, l

### 3.1.1.213. passwordSendExpiringTime

クライアントがパスワードの期限切れの制御を求めると、パスワードが警告期間内にある場合にのみ、Directory Server は有効期限までの時間の値を返します。パスワードの有効期限が警告期間内にあるかどうかにかかわらず、この値を常に返す必要のある既存のクライアントとの互換性を確保するために、**passwordSendExpiringTime** パラメーターを **on** に設定できます。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	passwordSendExpiringTime: off

### 3.1.1.214. passwordStorageScheme (パスワードの保存スキーム)

この属性は、**userPassword** 属性に保存されているユーザーパスワードを暗号化する方法を設定します。強固なパスワードストレージスキームなどの詳細は、「[パスワードストレージスキーム](#)」を参照してください。



## 注記

Red Hat は、この属性を設定しないことを推奨します。値が設定されていないと、Directory Server は、最も強力なパスワードストレージスキームを自動的に使用します。今後の Directory Server の更新で、セキュリティーを向上させる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。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	<a href="#">「パスワードストレージスキーム」</a> を参照してください。
Default Value	PBKDF2_SHA256
Syntax	DirectoryString
Example	passwordStorageScheme: PBKDF2_SHA256

### 3.1.1.215. passwordTPRDelayExpireAt

**passwordTPRDelayExpireAt** 属性はパスワードポリシーの一部です。管理者が一時パスワードをユーザーアカウントに設定した後に、**passwordTPRDelayExpireAt** は一時パスワードが期限切れになるまでの時間を秒単位で定義します。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	-1(無効) から最大 32 ビット 整数値 (2147483647)
デフォルト値	-1
構文	整数
例	passwordTPRDelayExpireAt: 3600

### 3.1.1.216. passwordTPRDelayValidFrom

**passwordTPRDelayValidFrom** 属性はパスワードポリシーの一部です。管理者が一時的なパスワードをユーザーアカウントに設定した後に、**passwordTPRDelayValidFrom** は一時パスワードを使用するまでの時間を秒単位で定義します。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	-1(無効) から最大 32 ビット 整数値 (2147483647)
デフォルト値	-1
構文	整数
例	passwordTPRDelayValidFrom: 60

### 3.1.1.217. passwordTPRMaxUse: 5

**passwordTPRMaxUse** 属性はパスワードポリシーの一部です。属性は、一時パスワードが期限切れになる前にユーザーが正常に認証できる回数を設定します。認証に成功すると、Directory Server では、パスワードの変更を行わないと、それ以外の操作ができないようになっています。ユーザーがパスワードを変更しないと、操作が終了します。認証が成功したかどうかにかかわらず、認証試行の回数が増えます。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=config
有効な値	-1(無効) から最大 32 ビット 整数値 (2147483647)
デフォルト値	-1
構文	整数
例	passwordTPRMaxUse: 5

### 3.1.1.218. passwordTrackUpdateTime

入力パスワードを変更した最終時間専用、別のタイムスタンプを記録するかどうかを設定します。これを有効にすると、**pwdUpdateTime** 操作属性をユーザーアカウントエントリーに追加します (**modifyTime** などの他の更新時間と区別)。

このタイムスタンプを使用すると、Active Directory など、さまざまな LDAP ストア間でパスワードの変更の同期が容易になります。

パスワードポリシーの詳細は、Red Hat Directory Server 管理ガイドのユーザー認証の管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	passwordTrackUpdateTime: off

### 3.1.1.219. passwordUnlock (アカウントのロック解除)

指定期間ディレクトリーからロックアウトされるか、ロックアウトされてから管理者がパスワードをリセットするまでロックアウトするかを指定します。アカウントのロックアウト機能は、ユーザーのパスワードを繰り返し推測してディレクトリーに分割しようとするハッカーから保護します。この **passwordUnlock** 属性を **off** に設定し、操作属性 **accountUnlockTime** の値が **0** である場合に、アカウントは期限なしにロックされます。

パスワードポリシーの詳細は、Red Hat Directory Server 管理ガイドのユーザー認証の管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	passwordUnlock: off

### 3.1.1.220. passwordWarning (警告の送信)

ユーザーのパスワードが失効するまで (ユーザーが次の LDAP 操作でパスワード失効の警告制御を受信するまで) の時間 (秒数) を指定します。LDAP クライアントによっては、警告の送信時にパスワードの変更を求めるプロンプトが表示される場合もあります。

これは、**pwdExpireWarning** と省略できます。

パスワードポリシーの詳細は、Red Hat Directory Server 管理ガイドのユーザー認証の管理の章を参照してください。



パラメーター	説明
エントリー DN	cn=config
Valid Range	1 から最大 32 ビット整数値 (2147483647) (秒単位)
デフォルト値	86400 (1日)
構文	整数
例	passwordWarning: 86400

### 3.1.1.221. passwordAdminSkipInfoUpdate

新しい **passwordAdminSkipInfoUpdate: on/off** 設定を **cn=config** エントリーに追加すると、パスワード管理者が実行するパスワード更新をきめ細かく制御できます。この設定を **on** に設定すると、パスワードのみが変更され、ユーザーエントリーのパスワード状態属性は更新されません。このような属性には、たとえ

ば、**passwordHistory**、**passwordExpirationTime**、**passwordRetryCount**、**pwdReset**、**passwordExpWarned** があります。

パラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	passwordAdminSkipInfoUpdate: on



#### 注記

パスワード管理者は、passwordAdminSkipInfoUpdate: on/off 設定を使用することで、パスワード構文の確認を回避できるだけでなく、グローバルおよびローカルのパスワードポリシーで設定され、**expiration timestamp (passwordExpirationTime)** 属性および **must change the password (pwdMustChange)** 属性を使用するパスワード有効期限設定も回避できます。

### 3.1.1.222. retryCountResetTime

**retryCountResetTime** 属性には UTC 形式の日時が含まれ、**passwordRetryCount** 属性が **0** にリセットされます。

パラメーター	説明
エントリー DN	cn=config
Valid Range	UTC 形式の有効なタイムスタンプ
デフォルト値	none
構文	一般化時間
例	retryCountResetTime: 20190618094419Z

### 3.1.2. cn=changelog5,cn=config

複数のサプライヤーレプリケーション変更ログの設定エントリーは **cn=changelog5** エントリーに保存されます。**cn=changelog5,cn=config** エントリーは、**extensibleObject** オブジェクトクラスのインスタンスです。

**cn=changelog5** エントリーには以下のオブジェクトクラスを含める必要があります。

- **top**
- **extensibleObject**



#### 注記

Directory Server では、2 種類の changelogs が維持されます。ここに保存され、**Changelog (変更ログ)** と呼ばれる最初のタイプは、マルチサプライヤーレプリケーションによって使用されます。2 番目の変更ログは、実際にはプラグインで、**retro changelog** と呼ばれ、一部のレガシーアプリケーションとの互換性確保用です。Retro Changelog プラグインの詳細は、「[Retro Changelog プラグイン](#)」を参照してください。

#### 3.1.2.1. cn

この必須属性は、changelog エントリーの相対識別名 (RDN) を設定します。

パラメーター	説明
Entry DN	cn=changelog5,cn=config
有効な値	任意の文字列
デフォルト値	changelog5
Syntax	DirectoryString
Example	cn=changelog5

### 3.1.2.2. nsslapd-changelogcompactdb-interval

データベースが明示的に圧縮されない限り、Berkeley データベースは空きページを再利用しません。compact 操作は未使用のページをファイルシステムに返し、データベースファイルサイズを縮小します。このパラメーターは、changelog データベースが機能する間隔を秒単位で定義します。データベース圧縮はリソース集約型であるため、頻繁には実行しないでください。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
Entry DN	cn=changelog5,cn=config
有効な値	0 (圧縮なし) から 2147483647 秒
デフォルト値	2592000 (30 日)
構文	整数
例	nsslapd-changelogcompactdb-interval: 2592000

### 3.1.2.3. nsslapd-changelogdir

この必須属性は、changelog エントリーの作成先のディレクトリー名を指定します。changelog 設定エントリーが作成されるたびに、有効なディレクトリーを追加する必要があります。そうでない場合は、操作は拒否されます。デフォルトで GUI により、このエントリーが `/var/lib/dirsrv/slapped-instance/changelogdb/` に保存されるように提案されます。



#### 警告

**cn=changelog5** エントリーが削除されると、サブディレクトリーを含む、**nsslapd-changelogdir** パラメーターに指定されたディレクトリーが削除され、すべてのコンテンツが削除されます。

この属性の変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
Entry DN	cn=changelog5,cn=config
有効な値	changelog を保存するディレクトリーへの有効なパス
デフォルト値	なし

パラメーター	説明
構文	DirectoryString
Example	nsslapd-changelogdir: /var/lib/dirsrv/slapd- <b>instance</b> /changelogdb/

### 3.1.2.4. nsslapd-changelogmaxage (Changelog 最大経過時間)

コンシューマーと同期する場合には、Directory Server は各更新をタイムスタンプ付きの変更ログに保存します。**nsslapd-changelogmaxage** パラメーターは、changelog に保存するレコードの最大期間を設定します。すべてのレプリカに正常に転送された古いレコードは自動的に削除されます。デフォルトでは、Directory Server は 8 日以上経過しているレコードを削除します。ただし、**nsslapd-changelogmaxage** および **nsslapd-changelogmaxentries** パラメーターを無効にした場合に、Directory Server はすべてのレコードを変更ログに保持するため、変更ログファイルが過度に大きくなる可能性があります。



#### 注記

Retro changelog には、セクション [Retro changelog nsslapd-changelogmaxage](#) で説明されている独自の **nsslapd-changelogmaxage** 属性があります。

trim 操作は、**nsslapd-changelogtrim-interval** パラメーターに設定される間隔で実行されます。

この属性の変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
Entry DN	cn=changelog5,cn=config
Valid Range	0 (エントリーはその経過時間に応じて削除されない) から最大 32 ビット整数 (2147483647)
デフォルト値	7d
構文	DirectoryString IntegerAgeID. AgeID の <b>s (S)</b> は秒、 <b>m (M)</b> は分、 <b>h (H)</b> は時間、 <b>d (D)</b> は日、 <b>w (W)</b> は週を示します。
例	nsslapd-changelogmaxage: 4w

### 3.1.2.5. nsslapd-changelogmaxentries (changelog の最大レコード)

コンシューマーと同期するとき、Directory Server は各更新を変更ログに保存します。**nsslapd-changelogmaxentries** パラメーターは、changelog に保存されているレコードの最大数を設定します。全レプリカに正常に転送されたレコードで一番古いものの数が **nsslapd-changelogmaxentries** の値を超えた場合に、Directory Server はそれらのレコードを自動的に変更ログから削除します。**nsslapd-changelogmaxentries** および **nsslapd-changelogmaxage** パラメーターを無効にした場合、Directory Server はすべてのレコードを変更ログに保持するため、変更ログファイルが過度に大きくなる可能性があります。



## 注記

**nsslapd-changelogmaxentries** パラメーターに低い値を設定した場合に、Directory Server はレプリケーション変更ログのファイルサイズを自動的に縮小しません。詳細は、[Red Hat Directory 管理ガイド](#)の該当するセクションを参照してください。

Directory Server は、**nsslapd-changelogtrim-interval** パラメーターで設定された間隔でトリム操作を実行します。

この属性の変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
Entry DN	cn=changelog5,cn=config
Valid Range	0(最大の上限がディスクサイズの場合) から最大 32 ビット整数 (2147483647)
デフォルト値	0
構文	整数
例	nsslapd-changelogmaxentries: 5000

### 3.1.2.6. nsslapd-changelogtrim-interval (レプリケーションの changelog のトリミング間隔)

Directory Server は、changelog でトリミングプロセスを繰り返し実行します。2 つの実行の間隔を変更するには、**nsslapd-changelogtrim-interval** パラメーターを更新し、間隔を秒単位で設定します。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
Entry DN	cn=changelog5,cn=config
Valid Range	0 から最大 32 ビットの整数値 (2147483647)
デフォルト値	300 (5 分)
構文	DirectoryString
例	nsslapd-changelogtrim-interval: 300

### 3.1.2.7. nsslapd-encryptionalgorithm (暗号化アルゴリズム)

この属性は、changelog の暗号化に使用される暗号化アルゴリズムを指定します。changelog 暗号化を有効にするには、サーバー証明書を Directory Server にインストールする必要があります。changelog の詳細は、[「nsslapd-changelogdir」](#) を参照してください。

この属性の変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
Entry DN	cn=changelog5,cn=config
Valid Range	AES または 3DES
デフォルト値	なし
構文	DirectoryString
例	nsslapd-encryptionalgorithm: AES

### 3.1.2.8. nsSymmetricKey

この属性は、内部で生成された対称鍵を保存します。changelog の詳細は、[「nsslapd-changelogdir」](#) を参照してください。

この属性の変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
Entry DN	cn=changelog5,cn=config
Valid Range	Base64 でエンコードされたキー
デフォルト値	なし
構文	DirectoryString
例	None

### 3.1.3. changelog 属性

changelog 属性には、changelog に記録されている変更が含まれます。

#### 3.1.3.1. changes

この属性には、LDIF 形式で追加操作と変更操作のエントリーに加えられた変更が含まれます。

OID	2.16.840.1.113730.3.1.8
構文	Binary
多値または単一値	複数値

定義される場所	Changelog インターネットドラフト
---------	-----------------------

### 3.1.3.2. changeLog

この属性には、サーバーの changelog を設定するエントリーのセットを含む、エントリーの識別名が含まれます。

OID	2.16.840.1.113730.3.1.35
構文	DN
多値または単一値	複数値
定義される場所	Changelog インターネットドラフト

### 3.1.3.3. changeNumber

この属性は常に存在します。これには、ディレクトリーエントリーに加えられた各変更を一意に識別する整数が含まれます。この数は、変更が発生した順序に関係します。数値が大きいほど、変更は遅くなります。

OID	2.16.840.1.113730.3.1.5
構文	整数
多値または単一値	複数値
定義される場所	Changelog インターネットドラフト

### 3.1.3.4. changeTime

この属性は、エントリーの追加時に **YYMMDDHHMMSS** 形式で時間を定義します。

OID	2.16.840.1.113730.3.1.77
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 3.1.3.5. changeType

この属性は、LDAP 操作のタイプ、**追加**、**削除**、**変更**、または **modrdn** を指定します。以下に例を示します。

changeType: modify

OID	2.16.840.1.113730.3.1.7
構文	DirectoryString
多値または単一値	複数値
定義される場所	Changelog インターネットドラフト

### 3.1.3.6. deleteOldRdn

**modrdn** 操作の場合に、この属性は古い RDN が削除されたかどうかを指定します。

ゼロ (0) の値は、古い RDN を削除します。0 以外の値は古い RDN を維持します。(ゼロ以外の値は、負または正の整数にすることができます。)

OID	2.16.840.1.113730.3.1.10
構文	Boolean
多値または単一値	複数値
定義される場所	Changelog インターネットドラフト

### 3.1.3.7. filterInfo

これは、レプリケーションの処理時に changelog で使用します。

OID	2.16.840.1.113730.3.1.206
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 3.1.3.8. newRdn

**modrdn** 操作の場合、この属性はエントリーの新しい RDN を指定します。

OID	2.16.840.1.113730.3.1.9
構文	DN
多値または単一値	複数値



定義される場所	Changelog インターネットドラフト
---------	-----------------------

### 3.1.3.9. newSuperior

**modrdn** 操作の場合、この属性は移動したエントリーの新しい親 (補助) エントリーを指定します。

OID	2.16.840.1.113730.3.1.11
構文	DN
多値または単一値	複数値
定義される場所	Changelog インターネットドラフト

### 3.1.3.10. targetDn

この属性には、LDAP 操作の影響を受けるエントリーの DN が含まれます。**modrdn** 操作の場合、**targetDn** 属性には変更または移動前のエントリーの DN が含まれます。

OID	2.16.840.1.113730.3.1.6
構文	DN
多値または単一値	複数値
定義される場所	Changelog インターネットドラフト

## 3.1.4. cn=encryption

暗号化関連の属性は、**cn=encryption,cn=config** エントリーに保存されます。**cn=encryption,cn=config** エントリーは、**nsldapdEncryptionConfig** オブジェクトクラスのインスタンスです。

### 3.1.4.1. allowWeakCipher

この属性は、弱い暗号を許可または拒否するかどうかを指定します。デフォルトは、**nsSSL3Ciphers** パラメーターに設定した値により異なります。

暗号は、以下の場合に弱いとみなされます。

- これらはエクスポート可能です。  
エクスポートする暗号には、暗号名に **EXPORT** というラベルが付いています。たとえば、**TLS\_RSA\_EXPORT\_WITH\_RC4\_40\_MD5** の場合は以下ようになります。
- この暗号は対称的であり、3DES アルゴリズムよりも弱いです。  
対称暗号は、暗号化と復号化の両方に同じ暗号鍵を使用します。
- キーの長さは 128 ビットより短いです。

この属性への変更を反映するには、サーバーを再起動する必要があります。

エントリー DN	cn=encryption,cn=config
有効な値	on   off
デフォルト値	<b>off</b> ( <b>nsSSL3Ciphers</b> パラメーターの値が <b>+all</b> または <b>default</b> に設定されている場合)。  <b>on</b> ( <b>nsSSL3Ciphers</b> パラメーターの値にユーザー固有の暗号化リストが含まれる場合)
構文	DirectoryString
例	allowWeakCipher: on

### 3.1.4.2. allowWeakDHParam

Directory Server にリンクするネットワークセキュリティーサービス (NSS) ライブラリーには、最低 2048 ビット Diffie-Hellman(DH) パラメーターが必要です。ただし、Java 1.6 や 1.7 クライアントなどの Directory Server に接続する一部のクライアントは、1024 ビットの DH パラメーターのみをサポートします。**allowWeakDHParam** パラメーターを使用すると、Directory Server で弱い 1024 ビットの DH パラメーターのサポートを有効にできます。

この属性への変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=encryption,cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	allowWeakDHParam: off

### 3.1.4.3. nsSSL3Ciphers

この属性は、暗号化された通信中に Directory Server が使用する TLS 暗号化暗号のセットを指定します。

このパラメーターに設定する値は、**allowWeakCipher** パラメーターのデフォルト値に影響します。詳細は「[allowWeakCipher](#)」を参照してください。

パラメーター	説明
エントリー DN	cn=encryption,cn=config
有効な値	<p>NSS 対応の暗号のコンマ区切りリスト。さらに、以下のパラメーターを使用することもできます。</p> <p>* デフォルト - 弱い暗号以外の NSS でアドバタイズされるデフォルトの暗号を有効にします。詳細は <a href="#">List supported cipher suites for SSL connections</a> を参照してください。</p> <p>* +all: すべての暗号が有効になります。<b>allowWeakCipher</b> パラメーターが有効な場合は、弱い暗号が含まれます。</p> <p>* -all: すべての暗号が無効になります。</p>
デフォルト値	default
構文	<p>DirectoryString</p> <p>無効にするにはプラス記号 (+) 記号を使用するか、マイナス (-) 記号で無効にし、その後に暗号を使用します。暗号のリストでは、空白スペースは使用できません。</p> <p>すべての暗号を有効にするには (具体的に呼び出す必要がある <b>rsa_null_md5</b> を除く)、<b>+all</b> を指定します。</p>
例	<pre>nsSSL3Ciphers: +TLS_RSA_AES_128_SHA,+TLS_RSA_AES_256_SHA, +TLS_RSA_WITH_AES_128_GCM_SHA256,- RSA_NULL_SHA</pre>

対応している暗号のリストの詳細は、[Red Hat Directory Server 管理ガイド](#)の該当するセクションを参照してください。

#### 3.1.4.4. nsSSLActivation

この属性は、TLS 暗号ファミリーが特定のセキュリティーモジュールに対して有効になっているかどうかを示します。

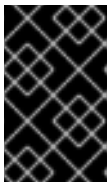
エントリー DN	cn= <b>encryptionType</b> ,cn=encryption,cn=config
有効な値	on   off
デフォルト値	
構文	DirectoryString

例	nsSSLActivation: on
---	---------------------

### 3.1.4.5. nsSSLClientAuth

この属性は、DirectoryServer がクライアント認証を実施する方法を示します。次の値を取ります。

- **off** - Directory Server ではクライアント認証は使用できません。
- **allowed** (デフォルト)- Directory Server でクライアント認証は使用できますが、必須ではありません。
- **required**: すべてのクライアントはクライアント認証を使用する必要があります。



#### 重要

Directory Server コンソールは、クライアント認証をサポートしません。したがって、**nsSSLClientAuth** 属性を **required** に設定すると、このコンソールを使用してインスタンスを管理できません。

この属性の変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=config
有効な値	off   allowed   required
デフォルト値	allowed
構文	DirectoryString
例	nsSSLClientAuth: allowed

### 3.1.4.6. nsSSEnabledCiphers

Directory Server は、複数値の **nsSSEnabledCiphers** 属性を自動的に生成します。属性は読み取り専用で、現在使用している Directory Server 暗号を表示します。このリストは、**nsSSL3Ciphers** 属性に設定したものとは異なる場合があります。たとえば、**nsSSL3Ciphers** 属性に弱い暗号を設定し、**allowWeakCipher** が無効な場合には、**nsSSEnabledCiphers** 属性は、弱い暗号をリスト表示せず、Directory Server ではその暗号を使用しません。

パラメーター	説明
エントリー DN	cn=config
有効な値	この属性の値は自動生成され、読み取り専用です。
デフォルト値	

パラメーター	説明
構文	DirectoryString
例	nsSSLClientAuth: TLS_RSA_WITH_AES_256_CBC_SHA::AES::SHA1::256

### 3.1.4.7. nsSSLPersonalitySSL

この属性には、SSL に使用する証明書名が含まれます。

エントリー DN	cn=encryption,cn=config
有効な値	証明書のニックネーム
デフォルト値	
構文	DirectoryString
以下に例を示します。	nsSSLPersonalitySSL: Server-Cert

### 3.1.4.8. nsSSLSessionTimeout

この属性は、TLS 接続の有効期間を設定します。最小タイムアウト値は **5 秒** です。小さい値を設定すると、自動的に **5 秒** に置き換えられます。以下の有効な範囲内の最大値より大きい値は、範囲内の最大値に置き換えられます。

この属性の変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=encryption,cn=config
Valid Range	5 秒から 24 時間
デフォルト値	0(これは、上記の有効な範囲の最大値を使用することを意味します)。
構文	整数
例	nsSSLSessionTimeout: 5

### 3.1.4.9. nsSSLSupportedCiphers

この属性には、サーバーでサポートされる暗号が含まれます。

エントリー DN	cn=encryption,cn=config
有効な値	特定のファミリー、暗号、および強度の文字列
デフォルト値	
構文	DirectoryString
以下に例を示します。	nsSSLSupportedCiphers: TLS_RSA_WITH_AES_256_CBC_SHA::AES::SHA1::256

### 3.1.4.10. nsSSLToken

この属性には、サーバーによって使用されるトークン (セキュリティーモジュール) の名前が含まれません。

エントリー DN	cn=encryption,cn=config
有効な値	モジュール名
デフォルト値	
構文	DirectoryString
以下に例を示します。	nsSSLToken: 内部 (ソフトウェア)

### 3.1.4.11. nsTLS1

TLS バージョン 1 を有効にします。TLS で使用される暗号は、**nsSSL3Ciphers** 属性で定義されます。

**sslVersionMin** パラメーターおよび **sslVersionMax** パラメーターが **nsTLS1** と組み合わせて設定されている場合、Directory Server はこれらのパラメーターから最も安全な設定を選択します。

この属性の変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=encryption,cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsTLS1: on

### 3.1.4.12. nsTLSEnableClientRenegotiation

Directory Server は、**SSL\_ENABLE\_RENEGOTIATION** オプションを使用した **SSL\_OptionSet()** ネットワークセキュリティーサービス (NSS) 機能を使用して、NSS の TLS 再ネゴシエーション動作を制御します。

**nsTLSEnableClientRenegotiation** 属性は、Directory Server が **SSL\_ENABLE\_RENEGOTIATION** オプションに渡す値を制御します。

- **nsTLSEnableClientRenegotiation** に指定すると、Directory Server は **SSL\_RENEGOTIATE\_REQUIRES\_XTN** を **SSL\_ENABLE\_RENEGOTIATION** オプションに渡します。この場合、NSS は [RFC 5746](#) を使用したセキュアな再ネゴシエーション試行を許可します。
- **nsTLSEnableClientRenegotiation: off** に指定すると、Directory Server は **SSL\_RENEGOTIATE\_NEVER** を **SSL\_ENABLE\_RENEGOTIATION** オプションに渡します。この場合、NSS は、安全なものでもすべての再ネゴシエーションの試行を拒否します。

NSS TLS 再ネゴシエーション動作の詳細は、[Is Red Hat affected by TLS renegotiation MITM attacks \(CVE-2009-3555\)?](#) の記事の [The RFC 5746 implementation in NSS \(Network Security Services\)](#) セクションを参照してください。

この属性への変更を反映するには、サービスを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=encryption,cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsTLSEnableClientRenegotiation: on

### 3.1.4.13. sslVersionMin

**sslVersionMin** パラメーターは、Directory Server が使用する TLS プロトコルの最小バージョンを設定します。ただし、デフォルトでは、Directory Server は、システム全体の暗号化ポリシーに基づいてこのパラメーターを自動的に設定します。`/etc/crypto-policies/config` ファイルでポリシープロファイルを以下のように設定します。

- **DEFAULT**、**FUTURE**、または **FIPS**、Directory Server は **sslVersionMin** を **TLS1.2** に設定します。
- **LEGACY**、Directory Server は **sslVersionMin** を **TLS1.0** に設定します。

または、**sslVersionMin** は、crypto ポリシーで定義されている値よりも高い値に手動で設定できます。

この属性への変更を反映するには、サービスを再起動する必要があります。

エントリー DN	cn=encryption,cn=config
有効な値	<b>TLS1.2</b> などの TLS プロトコルバージョン
デフォルト値	設定したシステム全体の暗号化ポリシープロファイルによって異なります。
構文	DirectoryString
以下に例を示します。	sslVersionMin: TLS1.2

#### 3.1.4.14. sslVersionMax

使用する TLS プロトコルの最大数を設定します。デフォルトでは、この値はシステムにインストールされている NSS ライブラリーで利用可能な最新のプロトコルバージョンに設定されます。

この属性の変更を反映するには、サーバーを再起動する必要があります。

**sslVersionMin** パラメーターおよび **sslVersionMax** パラメーターが **nsTLS1** と組み合わせて設定されている場合、Directory Server はこれらのパラメーターから最も安全な設定を選択します。

エントリー DN	cn=encryption,cn=config
有効な値	<b>TLS1.0</b> などの TLS プロトコルバージョン
デフォルト値	システムにインストールされている NSS ライブラリーで利用可能な最新のプロトコルバージョン
構文	DirectoryString
以下に例を示します。	sslVersionMax: TLS1.2

#### 3.1.5. cn=features

**cn=features** エントリー自体には属性がありません。このエントリーは、オブジェクトクラスが **nsContainer** となっている、親コンテナエントリーとしてのみ使用されます。

子エントリーには、機能および **directoryServerFeature** オブジェクトクラスを識別する **oid** 属性が含まれ、特定の ACL などの機能に関する識別情報のオプションも含まれます。以下に例を示します。以下に例を示します。

```
dn: oid=2.16.840.1.113730.3.4.9,cn=features,cn=config
objectClass: top
objectClass: directoryServerFeature
oid: 2.16.840.1.113730.3.4.9
cn: VLV Request Control
aci: (targetattr != "aci")(version 3.0; aci "VLV Request Control"; allow( read, search, compare, proxy )
userdn = "ldap:///all";)
creatorsName: cn=server,cn=plugins,cn=config
```



```
modifiersName: cn=server,cn=plugins,cn=config
createTimestamp: 20200129132357Z
modifyTimestamp: 20200129132357Z
```

### 3.1.5.1. oid

**oid** 属性には、ディレクトリーサービス機能に割り当てられたオブジェクト識別子が含まれます。**OID** は、これらのディレクトリー機能の命名属性として使用されます。

OID	2.16.840.1.113730.3.1.215
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 3.1.6. cn=mapping ツリー

- 接尾辞、レプリケーション、および Windows 同期の設定属性は **cn=mapping tree,cn=config** に保存されます。接尾辞に関連する設定属性は、接尾辞サブエントリー **cn=suffix, cn=mapping tree,cn=config** にあります。たとえば、**suffix** はディレクトリーツリーの root エントリーです (例: **dc=example,dc=com**)。
- レプリケーション設定属性は、**cn=replica,cn=suffix, cn=mapping tree,cn=config** に保存されます。
- レプリカ合意属性は **cn=replicationAgreementName, cn=replica,cn=suffix,cn=mapping tree,cn=config** に保存されます。
- Windows 同期合意属性は、**cn=syncAgreementName, cn=replica,cn=suffix,cn=mapping tree,cn=config** に保存されます。

### 3.1.7. cn=suffix\_DN 下の接尾辞設定属性

接尾辞の設定は、**cn="suffix\_DN",cn=mapping tree,cn=config** エントリーに保存されます。これらのエントリーは、**nsMappingTree** オブジェクトクラスのインスタンスです。**extensibleObject** オブジェクトクラスは、所属するエントリーが任意のユーザー属性を保持できるようにします。サーバーが接尾辞設定属性を考慮に入れるには、**最上位** のオブジェクトクラスに加えて、これらのオブジェクトクラスがエントリーに存在する必要があります。

接尾辞 DN には等号 (=)、コンマ (,)、空白文字などの文字が含まれるため、引用符で囲む必要があります。引用符を使用すると、DN が別の DN の値として正しく表示されます。例:

```
cn="dc=example,dc=com",cn=mapping tree,cn=config
```

詳細は、[Directory Server 管理ガイド](#) の該当するセクションを参照してください。

#### 3.1.7.1. cn

この必須属性は、新しい接尾辞の相対識別名 (RDN) を設定します。

パラメーター	説明
エントリー DN	cn=suffix_DN,cn=mapping tree,cn=config
有効な値	有効な LDAP DN
デフォルト値	
構文	DirectoryString
例	cn: dn=example,dc=com

### 3.1.7.2. nsslapd-backend

このパラメーターは、要求の処理に使用されるデータベースまたはデータベースリンクの名前を設定します。これは複数値であり、値ごとに1つのデータベースまたはデータベースリンクがあります。この属性は、**nsslapd-state** 属性の値が、**backend** ま **referral on update** に設定されている場合に必要です。

この値は、**cn=ldbm database,cn=plugins,cn=config** の下にあるバックエンドデータベースエントリーインスタンスの名前に設定します。例: **o=userroot,cn=ldbm database,cn=plugins,cn=config**

パラメーター	説明
エントリー DN	cn=suffix_DN,cn=mapping tree,cn=config
有効な値	有効なパーティション名
デフォルト値	
構文	DirectoryString
例	nsslapd-backend: userRoot

### 3.1.7.3. nsslapd-distribution-function

**nsslapd-distribution-function** パラメーターは、カスタムディストリビューション関数の名前を設定します。**nsslapd-backend** 属性に複数のデータベースを設定する場合は、この属性を設定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詳細は、[Directory Server 管理ガイド](#)の該当するセクションを参照してください。

パラメーター	説明
エントリー DN	cn=suffix_DN,cn=mapping tree,cn=config

パラメーター	説明
有効な値	有効なディストリビューション機能
デフォルト値	
構文	DirectoryString
例	nsslapd-distribution-plugin: distribution_function_name

#### 3.1.7.4. nsslapd-distribution-plugin

**nsslapd-distribution-plugin** は、カスタムディストリビューション関数で使用する共有ライブラリーを設定します。**nsslapd-backend** 属性に複数のデータベースを設定する場合は、この属性を設定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詳細は、[Directory Server 管理ガイド](#)の該当するセクションを参照してください。

パラメーター	説明
エントリー DN	cn=suffix_DN,cn=mapping tree,cn=config
有効な値	有効なディストリビューションプラグイン
デフォルト値	
構文	DirectoryString
例	nsslapd-distribution-plugin: /path/to/shared/library

#### 3.1.7.5. nsslapd-parent

サブ接尾辞を作成する場合は、**nsslapd-parent** 属性を使用して親接尾辞を定義します。

属性が設定されていない場合、新しい接尾辞が root 接尾辞として作成されます。

パラメーター	説明
エントリー DN	cn=suffix_DN,cn=mapping tree,cn=config
有効な値	有効なパーティション名
デフォルト値	
構文	DirectoryString

パラメーター	説明
例	nsslapd-parent-suffix: dc=example,dc=com

### 3.1.7.6. nsslapd-referral

この属性は、接尾辞で返される参照の LDAP URL を設定します。**nssldap-referral** 属性を複数回追加して、複数の参照 URL を設定できます。

**nsslapd-state** パラメーターを **referral** に設定した場合や、**更新** 時にこの属性を設定する必要があります。

パラメーター	説明
エントリー DN	cn=suffix_DN,cn=mapping tree,cn=config
有効な値	有効な LDAP URL
デフォルト値	
構文	DirectoryString
例	nssldap-referral: ldap://example.com/

### 3.1.7.7. nsslapd-state

このパラメーターは、接尾辞が操作を処理する方法を決定します。属性は以下の値を取ります。

- **backend**: バックエンドデータベースはすべての操作を処理します。
- **disabled**: 操作を処理するのにデータベースは利用できません。サーバーは、クライアントアプリケーションからの要求に応じて、**No such search object** エラーを返します。
- **referral**: Directory Server は、この接尾辞への要求の参照 URL を返します。
- **referral on update**: データベースはすべての操作に使用されます。更新要求のみが送信される参照元です。

パラメーター	説明
エントリー DN	cn=suffix_DN,cn=mapping tree,cn=config
有効な値	更新におこえる backend   disabled   referral   referral
デフォルト値	バックエンド
構文	DirectoryString

パラメーター	説明
例	nsslapd-state: backend

### 3.1.8. cn=replica,cn=suffixDN,cn=mapping tree,cn=config 下のレプリケーション属性

レプリケーション設定属性は、**cn=replica,cn=suffix,cn=mapping tree,cn=config** に保存されます。**cn=replica** エントリーは、**nsDS5Replica** オブジェクトクラスのインスタンスです。サーバーがレプリケーション設定属性を考慮に入れるには、**最上位** のオブジェクトクラスに加えて、これらのオブジェクトクラスがエントリーに存在する必要があります。レプリケーションの詳細は、**Red Hat Directory Server 管理ガイド**のレプリケーションの管理の章を参照してください。

**cn=replica,cn=suffix,cn=mapping tree,cn=config** エントリーには、以下のオブジェクトクラスが含まれている必要があります。

- **top**
- **extensibleObject**
- **nsds5replica**

#### 3.1.8.1. cn

レプリカの命名属性を設定します。**cn** 属性は **replica** に設定する必要があります。

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	この値は <b>replica</b> に設定する必要があります。
デフォルト値	replica
構文	DirectoryString
例	cn=replica

#### 3.1.8.2. nsds5DebugReplicaTimeout

この属性で、レプリケーションがデバッグロギングで実行される場合に使用する別のタイムアウトの期間を指定します。これにより、時間だけ、または時間とデバッグレベル両方を設定できます。

**nsds5debugreplicatimeout: seconds[:debuglevel]**

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config

パラメーター	説明
有効な値	数値文字列
デフォルト値	
構文	DirectoryString
例	nsds5debugreplicatimeout: 60:8192

### 3.1.8.3. nsDS5Flags

この属性は、フラグで以前に定義されたレプリカプロパティを設定します。現時点では、ログが変更されるかどうかを設定するフラグは1つのみ存在します。

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	0   1  * 0: レプリカは changelog に書き込みません。これはコンシューマーのデフォルトです。  * 1: レプリカは変更ログに書き込みます。これは、ハブとサプライヤーのデフォルトです。
デフォルト値	0
構文	整数
例	nsDS5Flags: 0

### 3.1.8.4. nsDS5ReplConflict

この属性は **cn=replica** エントリーにはありませんが、レプリケーションと併用されます。この複数値属性は、同期プロセスで自動解決できない変更で競合があるエントリーに含まれます。管理者の介入を必要とするレプリケーションの競合を確認するには、LDAP 検索を実行します (**nsDS5ReplConflict=\***)。以下に例を示します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -s sub -b
dc=example,dc=com "((objectclass=nsTombstone)(nsDS5ReplConflict=*))" dn nsDS5ReplConflict
nsUniqueID
```

検索フィルター "**(objectclass=nsTombstone)**" を使用すると、tombstone(削除済み) エントリーも表示されます。**nsDS5ReplConflict** の値には、競合しているエントリーの詳細情報が含まれます。通常、**nsUniqueID** でそのエントリーを参照します。**nsUniqueID** で tombstone エントリーを検索できません。以下に例を示します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -s sub -b
dc=example,dc=com "((objectclass=nsTombstone)(nsUniqueID=66a2b699-1dd211b2-807fa9c3-
a58714648))"
```

### 3.1.8.5. nsDS5ReplicaAutoReferral

この属性は、Directory Server がデータベースの設定済みの参照に従うかどうかを設定します。

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	on   off
デフォルト値	
構文	DirectoryString
例	nsDS5ReplicaAutoReferral: on

### 3.1.8.6. nsState

この属性は、クロックの状態に関する情報を格納します。これは、サーバーがバックワードクロックエラーの検出に必要な既存のシーケンス番号よりも低い変更シーケンス番号 (**csn**) を生成できないようにするための内部使用専用で設計されています。

### 3.1.8.7. nsDS5ReplicaAbortCleanRUV

この読み取り専用属性は、廃止または欠落しているサプライヤーの古い RUV エントリーを削除するバックグラウンドタスクを中止するかかどうかを指定します。このタスクの詳細は、「[cn=abort cleanallruv](#)」を参照してください。値が **0** の場合は、タスクが非アクティブであることを示します。値が **1** の場合は、タスクがアクティブであることを示します。

この属性は、サーバーの再起動後に中止タスクを再開できるように存在します。タスクが完了すると、属性が削除されます。

この値が手動で設定されている場合、サーバーは変更要求を無視します。

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	0   1
デフォルト値	なし
構文	整数
例	nsDS5ReplicaAbortCleanRUV: 1

### 3.1.8.8. nsds5ReplicaBackoffMin および nsds5ReplicaBackoffMax

これらの属性は、更新をできるだけ早く送信する必要があるレプリケーショントラフィックがある環境で使用されます。

デフォルトでは、リモートレプリカがビジー状態になると、レプリケーションプロトコルはバックオフ状態になり、バックオフタイマーの次の間隔で更新の送信を再試行します。デフォルトでは、タイマーは3秒から開始し、最大待機時間は5分です。特定の状況ではこれらのデフォルト設定では不十分な場合があるため、**nsds5ReplicaBackoffMin** および **nsds5ReplicaBackoffMax** を使用して、最小および最大待機時間を設定できます。

この設定は、サーバーがオンラインの状態であれば適用でき、サーバーを再起動する必要はありません。無効な設定が使用されると、代わりにデフォルト値が使用されます。設定は CLI ツールを使用して処理する必要があります。

### 3.1.8.9. nsDS5ReplicaBindDN

この複数値属性は、バインディング時に使用する DN を指定します。この **cn=replica** エントリーには複数の値がありますが、レプリカ合意ごとに1つのサプライヤーバインド DN のみを使用できます。各値は、コンシューマーサーバーのローカルエントリーの DN である必要があります。レプリケーションサプライヤーがクライアント証明書ベースの認証を使用してコンシューマーに接続する場合は、証明書の **subjectDN** をローカルエントリーにマップするようにコンシューマーの証明書マッピングを設定します。



#### 重要

セキュリティ上の理由から、この属性は **cn=Directory Manager** に設定しないでください。

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	任意の有効な DN
デフォルト値	
構文	DirectoryString
例	nsDS5ReplicaBindDN: cn=replication manager,cn=config

### 3.1.8.10. nsDS5ReplicaBindDNGroup

**nsDS5ReplicaBindDNGroup** 属性はグループ DN を指定します。次に、このグループをデプロイメントとして、サブグループのメンバーを含むメンバーが起動時またはレプリカオブジェクトの変更時に **replicaBindDNs** 属性に追加されます。これにより、グループ DN を設定できるため、**nsDS5ReplicaBindDN** 属性によって提供される現在の機能が拡張されます。



パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	有効なグループ DN
デフォルト値	
構文	DirectoryString
例	nsDS5ReplicaBindDNGroup: cn=sample_group,ou=groups,dc=example,dc=com

### 3.1.8.11. nsDS5ReplicaBindDNGroupCheckInterval

Directory Server は、**nsDS5ReplicaBindDNGroup** 属性で指定されたグループの変更をチェックし、それに応じて **replicaBindDN** パラメーターのリストを自動的に再構築します。これらの操作はパフォーマンスに悪影響を与えるため、**nsDS5ReplicaBindDNGroupCheckInterval** 属性で指定された間隔でのみ実行されます。

この属性は、次の値を受け入れます。

- **-1**: 実行時の動的チェックを無効にします。管理者は、**nsDS5ReplicaBindDNGroup** 属性が変更された場合にインスタンスを再起動する必要があります。
- **0**: Directory Server は、グループの変更直後にリストを再ビルドします。
- 正の 32 ビットの整数値: 最後にリビルドされてから経過する必要のある最小期間 (秒数)。

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	-1 - 32 ビットの最大整数 (2147483647)
デフォルト値	-1
構文	整数
例	nsDS5ReplicaBindDNGroupCheckInterval: 0

### 3.1.8.12. nsDS5ReplicaChangeCount

この読み取り専用属性は、変更ログ内のエントリーの総数と、それらがまだレプリケートされていないかどうかを示します。changelog がページされると、まだレプリケートされていないエントリーのみが残ります。

ページ操作プロパティの詳細は、「[nsDS5ReplicaPurgeDelay](#)」および「[nsDS5ReplicaTombstonePurgeInterval](#)」を参照してください。

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Range	-1 - 32 ビットの最大整数 (2147483647)
デフォルト値	
構文	整数
例	nsDS5ReplicaChangeCount: 675

### 3.1.8.13. nsDS5ReplicaCleanRUV

この読み取り専用属性は、廃止または欠落しているサプライヤーの古い RUV エントリーを削除するバックグラウンドタスクがアクティブかどうかを指定します。このタスクの詳細は、「[cn=cleanallruv](#)」を参照してください。値が **0** の場合は、タスクが非アクティブであることを示します。値が **1** の場合は、タスクがアクティブであることを示します。

この属性は、サーバーの再起動後にクリーンアップタスクを再開できるように存在します。タスクが完了すると、属性が削除されます。

この値が手動で設定されている場合、サーバーは変更要求を無視します。

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	0   1
デフォルト値	なし
構文	整数
例	nsDS5ReplicaCleanRUV: 0

### 3.1.8.14. nsDS5ReplicaId

この属性は、特定のレプリケーション環境のサプライヤーに一意的 ID を設定します。

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Range	サプライヤーの場合: <b>1 から 65534</b> コンシューマーおよびハブの場合: <b>65535</b>

パラメーター	説明
デフォルト値	
構文	整数
例	nsDS5ReplicaId: 1

### 3.1.8.15. nsDS5ReplicaLegacyConsumer

この属性がない場合や、値が **false** の場合、レプリカがレガシーコンシューマーではないことを意味します。

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	true   false
デフォルト値	false
構文	DirectoryString
例	nsDS5ReplicaLegacyConsumer: false

### 3.1.8.16. nsDS5ReplicaName

この属性は、内部操作の一意的識別子を割り当ててレプリカの名前を指定します。指定のない場合は、この一意的識別子は、レプリカの作成時にサーバーにより割り当てられます。



#### 注記

サーバーでこの名前の生成を許可することを推奨します。ただし、レプリカロールの変更 (ハブなど) など、特定の状況では、この値を指定する必要があります。それ以外の場合は、サーバーは正しい changelog データベースを使用しないので、レプリケーションに失敗します。

この属性は内部使用のみを対象とします。

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	
デフォルト値	

パラメーター	説明
構文	DirectoryString (UID はレプリカを識別)
例	nsDS5ReplicaName: 66a2b699-1dd211b2-807fa9c3-a58714648

### 3.1.8.17. nsds5ReplicaProtocolTimeout

サーバーを停止したり、レプリカを無効にしたり、レプリカ合意を削除したりすると、サーバーに負荷がかかっているときにレプリケーションを停止するまでの待機時間のタイムアウトがあります。**nsds5ReplicaProtocolTimeout** 属性はこのタイムアウトを設定するために使用され、デフォルト値は 120 秒です。

2 分のタイムアウトが長すぎる、または十分に長くないシナリオが存在する可能性があります。たとえば、特定のレプリカ合意は、シャットダウン中にレプリケーションセッションを終了する前により多くの時間が必要になる場合があります。

この属性は、バックエンドの主要レプリケーション設定エントリーに追加できます。

パラメーター	説明
エントリー DN	cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
Valid Range	0 から最大 32 ビット 整数 (2147483647) (秒単位)
デフォルト値	120
構文	整数
例	nsds5ReplicaProtocolTimeout: 120

**nsds5ReplicaProtocolTimeout** 属性をレプリカ合意に追加することもできます。レプリカ合意プロトコルのタイムアウトは、メインのレプリカ設定エントリーに設定されたタイムアウトをオーバーライドします。これにより、レプリカ合意ごとに異なるタイムアウトが可能になります。レプリケーションセッションが進行中、新しいタイムアウトによってそのセッションが中断され、サーバーのシャットダウンが許可されます。

### 3.1.8.18. nsDS5ReplicaPurgeDelay

この属性は、削除されたエントリー (tombstone エントリー) および状態情報の最大期間を制御します。

Directory Server は、トゥームストーンエントリーと状態情報を格納するため、マルチサプライヤーレプリケーションプロセスで競合が発生した場合に、サーバーは、変更シーケンス番号に格納されているタイムスタンプとレプリカ ID に基づいて競合を解決します。

内部 Directory Server のハウスキーピング操作では、この属性の値 (秒単位) よりも古い tombstone エントリーが定期的に削除されます。状態情報を含むエントリーが変更されると、**nsDS5ReplicaPurgeDelay** 値よりも古い状態情報が削除されます。

マルチサプライヤーレプリケーションでは、属性の値より古い場合でも、サーバーがプライムレプリケーションに対して少数の最新の更新を保持する必要がある場合があるため、すべての tombstone および状態情報が削除されるわけではありません。

この属性は、エントリーで内部パージ操作を実行する間隔を秒単位で指定します。この属性の設定時には、レプリケーションの競合を解決するのに十分な情報を保持し、異なるサーバーに格納されているデータのコピーが分岐しないように、パージ遅延がレプリケーションポリシーの最長のレプリケーションサイクルよりも長いことを確認してください。

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Range	0 (永続保持) から最大 32 ビットの整数 (2147483647)
デフォルト値	604800 [1 week (60x60x24x7)]
構文	整数
例	nsDS5ReplicaPurgeDelay: 604800

### 3.1.8.19. nsDS5ReplicaReapActive

この読み取り専用属性は、データベースから古い tombstones (削除されたエントリー) を削除するバックグラウンドタスクがアクティブであるかどうかを指定します。このタスクの詳細は、「[nsDS5ReplicaTombstonePurgeInterval](#)」を参照してください。値が **0** の場合は、タスクが非アクティブであることを示します。値が **1** の場合は、タスクがアクティブであることを示します。この値が手動で設定されている場合、サーバーは変更要求を無視します。

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	0   1
デフォルト値	
構文	整数
例	nsDS5ReplicaReapActive: 0

### 3.1.8.20. nsDS5ReplicaReferral

この複数値属性は、ユーザー定義の参照を指定します。これは、コンシューマーでのみ定義する必要があります。ユーザーの参照は、クライアントが読み取り専用コンシューマーのデータを変更しようとし

た場合にのみ返されます。このオプションの参照は、レプリケーションプロトコルのコンシューマーによって自動設定される参照を上書きします。

URL の形式は `ldap[s]://host_name:port_number` または `ldap[s]://IP_address:port_number` (IPv4 または IPv6 アドレス) 形式をすることができます。

パラメーター	説明
エントリー DN	<code>cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
有効な値	有効な LDAP URL
デフォルト値	
構文	DirectoryString
例	<code>nsDS5ReplicaReferral: ldap://server.example.com:389</code>

### 3.1.8.21. nsDS5ReplicaReleaseTimeout

この属性は、複数のサプライヤーのシナリオでサプライヤーとハブで使用される場合、サプライヤーがレプリカをリリースするまでのタイムアウト期間 (秒単位) を決定します。これは、ネットワーク接続が遅いなどの問題で、1つのサプライヤーがレプリカへのアクセスを取得して長期間確保し、他のすべてのサプライヤーがレプリカにアクセスして更新を送信できない場合に役立ちます。この属性が設定されている場合には、レプリカは指定された期間後にサプライヤーによって開放されるため、レプリケーションのパフォーマンスが向上します。

この属性を **0** に設定するとタイムアウトが無効になります。他の値の場合には、タイムアウトの長さが秒単位で決定されます。



#### 重要

この属性は、**1** から **30** までの値に設定しないでください。多くの場合、タイムアウトが短い場合にはレプリケーションのパフォーマンスが低下します。

パラメーター	説明
エントリー DN	<code>cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
有効な値	0 から最大 32 ビット 整数 (2147483647) (秒単位)
デフォルト値	60
構文	整数
例	<code>nsDS5ReplicaReleaseTimeout: 60</code>

### 3.1.8.22. nsDS5ReplicaRoot

この属性は、複製された領域のルートに DN を設定します。この属性は、レプリケートされるデータベースの接尾辞と同じ値であり、変更することはできません。

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	複製されるデータベースの接尾辞 (接尾辞 DN)
デフォルト値	
構文	DirectoryString
例	nsDS5ReplicaRoot: "dc=example,dc=com"

### 3.1.8.23. nsDS5ReplicaTombstonePurgeInterval

この属性は、パージ操作サイクルの間隔 (秒単位) を指定します。

サーバーは定期的に内部ハウスキeping操作を実行し、changelog およびメインのデータベースから古い更新および状態情報を削除します。[[nsDS5ReplicaPurgeDelay](#)] を参照してください。

この属性を設定するときは、特にサーバーがクライアントやサプライヤーからの削除操作を多数処理する場合に、パージ操作に時間がかかることに注意してください。

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Range	0 から最大 32 ビット 整数 (2147483647) (秒単位)
デフォルト値	86400 (1 日)
構文	整数
例	nsDS5ReplicaTombstonePurgeInterval: 86400

### 3.1.8.24. nsDS5ReplicaType

このレプリカと他のレプリカ間で存在するレプリケーション関係のタイプを定義します。

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config

パラメーター	説明
有効な値	0 1 2 3 * 0 は不明を意味します * 1はプライマリー (まだ使用されていない) を意味します * 2 はコンシューマーを意味します (読み取り専用) * 3 コンシューマー/サプライヤー (更新可能)
デフォルト値	
構文	整数
例	nsDS5ReplicaType: 2

### 3.1.8.25. nsds5Task

この属性は、データベースのコンテンツを LDIF ファイルにダンプしたり、レプリケーショントポロジから古くなったサプライヤーを削除するなど、レプリケーションタスクを起動します。

**nsds5Task** 属性を以下の値のいずれかに設定できます。

- **cl2ldif**: `/var/lib/dirsrv/slaped-instance_name/changelogdb/` ディレクトリーの LDIF ファイルに changelog をエクスポートします。
- **ldif2cl**: `/var/lib/dirsrv/slaped-instance_name/changelogdb/` ディレクトリーに保存されている LDIF ファイルから changelog をインポートします。
- **cleanruv**: 操作を実行するサプライヤーからレプリカ更新ベクトル (RUV) を削除します。
- **cleanallruv**: レプリケーショントポロジ内のすべてのサーバーから RUV を削除します。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	* <b>cl2ldif</b> * <b>ldif2cl</b> * <b>cleanruv</b> * <b>cleanallruv</b>
デフォルト値	



パラメーター	説明
構文	DirectoryString
例	nsds5Task: cleanallruv

### 3.1.9. cn=ReplicationAgreementName,cn=replica,cn=suffixName,cn=mapping tree,cn=config 下のレプリケーション属性

レプリカ合意に関連するレプリケーション属性

は、**cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config** に保存されます。**cn=ReplicationAgreementName** エントリーは、**nsDS5ReplicationAgreement** オブジェクトクラスのインスタンスです。レプリカ合意は、サプライヤーレプリカでのみ設定されます。

#### 3.1.9.1. cn

この属性は命名に使用されます。この属性が設定されたら、それを変更することはできません。この属性は、レプリカ合意の設定に必要です。

パラメーター	説明
エントリー DN	<b>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</b>
有効な値	任意の有効な <b>cn</b>
デフォルト値	
構文	DirectoryString
例	cn: SupplierAtoSupplierB

#### 3.1.9.2. description

レプリカ合意のフリーフォームテキストの説明。この属性は変更できます。

パラメーター	説明
エントリー DN	<b>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</b>
有効な値	任意の文字列
デフォルト値	
構文	DirectoryString

パラメーター	説明
例	説明: サーバー A とサーバー B 間のレプリカ合意

### 3.1.9.3. nsDS5ReplicaBindDN

この属性は、レプリケーション中にコンシューマーにバインドする時に使用する DN を設定します。この属性の値は、コンシューマーレプリカの **cn=replica** にあるものと同じである必要があります。証明書ベースの認証が使用されている場合、これは空になる可能性があります。この場合、使用される DN は証明書のサブジェクト DN であり、コンシューマーは適切なクライアント証明書マッピングを有効にする必要があります。これは変更することもできます。

パラメーター	説明
エントリー DN	cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	有効な DN(クライアント証明書を使用する場合は空にすることができます)
デフォルト値	
構文	DirectoryString
例	nsDS5ReplicaBindDN: cn=replication manager,cn=config

### 3.1.9.4. nsDS5ReplicaBindMethod

この属性は、サーバーがコンシューマーサーバーにバインドするのに使用するメソッドを設定します。

**nsDS5ReplicaBindMethod** は以下の値をサポートします。

- 空白または **SIMPLE**: サーバーはパスワードベースの認証を使用します。このバインドメソッドを使用する場合は、**nsds5ReplicaBindDN** パラメーターおよび **nsds5ReplicaCredentials** パラメーターをユーザー名とパスワードを指定します。
- **SSLCLIENTAUTH**: サプライヤーとコンシューマー間の証明書ベースの認証を有効にします。このため、コンシューマーサーバーには、サプライヤーの証明書を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ように設定された証明書マッピングが必要です。
- **SASL/GSSAPI**: SASL を使用した Kerberos 認証を有効にします。これには、サプライヤーサーバーに Kerberos キータブがあり、コンシューマーサーバーは、サプライヤーの Kerberos プリンシパルを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ように設定されている SASL マッピングエントリーが必要です。  
詳細は、Red Hat Directory Server 管理ガイドの以下のセクションを参照してください。

- [KDC サーバーおよびキータブの概要](#)

- [コンソールからの SASL アイデンティティマッピングの設定](#)

- **SASL/DIGEST-MD5: DIGEST-MD5** メカニズムで SASL を使用したパスワードベースの認証を有効にします。このバインドメソッドを使用する場合は、**nsds5ReplicaBindDN** パラメーターおよび **nsds5ReplicaCredentials** パラメーターをユーザー名とパスワードを指定します。

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	SIMPLE   SSLCLIENTAUTH   SASL/GSSAPI   SASL/DIGEST
デフォルト値	SIMPLE
構文	DirectoryString
例	nsDS5ReplicaBindMethod: SIMPLE

### 3.1.9.5. nsds5ReplicaBootstrapBindDN

**nsds5ReplicaBootstrapBindDN** パラメーターは、**LDAP\_INVALID\_CREDENTIALS (err=49)**、**LDAP\_INAPPROPRIATE\_AUTH (err=48)**、または **LDAP\_NO\_SUCH\_OBJECT (err=32)** エラーにより、サブライヤーがコンシューマーへのバインドに失敗した場合に、Directory Server が使用するフォールバックバインド識別名 (DN) を設定します。

このような場合、Directory Server は、**nsds5ReplicaBootstrapBindDN** パラメーター、**nsds5ReplicaBootstrapCredentials** パラメーター、**nsds5ReplicaBootstrapBindMethod** パラメーター、および **nsds5ReplicaBootstrapTransportInfo** パラメーターを使用して接続を確立します。これらのブートストラップ設定を使用してサーバーが接続を確立できない場合、サーバーは接続の試行を停止します。

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	任意の有効な DN
デフォルト値	
構文	DirectoryString
例	nsds5ReplicaBootstrapBindDN: cn=replication manager,cn=config

### 3.1.9.6. nsds5ReplicaBootstrapBindMethod

**nsds5ReplicaBootstrapBindMethod** パラメーターは、**LDAP\_INVALID\_CREDENTIALS (err=49)**、**LDAP\_INAPPROPRIATE\_AUTH (err=48)**、または **LDAP\_NO\_SUCH\_OBJECT (err=32)** エ

ラーにより、サブライヤーがコンシューマーへのバインドに失敗した場合に、Directory Server が使用するフォールバックログインメカニズムのパスワードを設定します。

このような場合、Directory Server は、**nsds5ReplicaBootstrapBindDN** パラメーター、**nsds5ReplicaBootstrapCredentials** パラメーター、**nsds5ReplicaBootstrapBindMethod** パラメーター、および **nsds5ReplicaBootstrapTransportInfo** パラメーターを使用して接続を確立します。これらのブートストラップ設定を使用してサーバーが接続を確立できない場合、サーバーは接続の試行を停止します。

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn= <b>suffixDN</b> ,cn=mapping tree,cn=config
有効な値	SIMPLE   SSLCLIENTAUTH   SASL/GSSAPI   SASL/DIGEST
デフォルト値	
構文	DirectoryString
例	nsds5ReplicaBootstrapBindMethod: SIMPLE

### 3.1.9.7. nsds5ReplicaBootstrapCredentials

**nsds5ReplicaBootstrapCredentials** パラメーターは、**LDAP\_INVALID\_CREDENTIALS (err=49)**、**LDAP\_INAPPROPRIATE\_AUTH (err=48)**、または **LDAP\_NO\_SUCH\_OBJECT (err=32)** エラーにより、サブライヤーがコンシューマーへのバインドに失敗した場合に、Directory Server が使用するフォールバックバインド識別名 (DN) のパスワードを設定します。

このような場合、Directory Server は、**nsds5ReplicaBootstrapBindDN** パラメーター、**nsds5ReplicaBootstrapCredentials** パラメーター、**nsds5ReplicaBootstrapBindMethod** パラメーター、および **nsds5ReplicaBootstrapTransportInfo** パラメーターを使用して接続を確立します。これらのブートストラップ設定を使用してサーバーが接続を確立できない場合、サーバーは接続の試行を停止します。

Directory Server は、クリアテキストでパラメーターを設定すると、AES リバーシブルパスワードの暗号化アルゴリズムを使用してパスワードを自動的にハッシュ化します。

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn= <b>suffixDN</b> ,cn=mapping tree,cn=config
有効な値	有効な文字列
デフォルト値	
構文	DirectoryString

パラメーター	説明
例	nsds5ReplicaBootstrapCredentials: password

### 3.1.9.8. nsds5ReplicaBootstrapTransportInfo

**nsds5ReplicaBootstrapTransportInfo** パラメーターは、**LDAP\_INVALID\_CREDENTIALS (err=49)**、**LDAP\_INAPPROPRIATE\_AUTH (err=48)**、または **LDAP\_NO\_SUCH\_OBJECT (err=32)** のエラーにより、サプライヤーがコンシューマーへのバインドに失敗したときに、Directory Server が使用するフォールバック接続用のレプリカとの間の接続の暗号化方式を設定します。

このような場合、Directory Server は、**nsds5ReplicaBootstrapBindDN** パラメーター、**nsds5ReplicaBootstrapCredentials** パラメーター、**nsds5ReplicaBootstrapBindMethod** パラメーター、および **nsds5ReplicaBootstrapTransportInfo** パラメーターを使用して接続を確立します。これらのブートストラップ設定を使用してサーバーが接続を確立できない場合、サーバーは接続の試行を停止します。

属性は以下の値を取ります。

- **TLS**: 接続は **StartTLS** コマンドを使用して暗号化を開始します。
- **ssl**: コネクションは TLS 暗号化で LDAPS を使用します。
- **LDAP**: 接続は暗号化されていません。

パラメーター	説明
エントリー DN	cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	TLS   SSL   LDAP
デフォルト値	
構文	DirectoryString
例	nsds5ReplicaBootstrapTransportInfo: SSL

### 3.1.9.9. nsDS5ReplicaBusyWaitTime

この属性は、コンシューマーがビジー応答を送信してから、コンシューマーがアクセスの取得を試みるまで、サプライヤーが待機する時間を秒単位で設定します。デフォルト値は 3 秒です。属性を負の値に設定すると、Directory Server はメッセージおよび **LDAP\_UNWILLING\_TO\_PERFORM** エラーコードをクライアントに送信します。

**nsDS5ReplicaBusyWaitTime** 属性は、**nsDS5ReplicaSessionPauseTime** 属性とともに動作します。この 2 つの属性は、**nsDS5ReplicaSessionPauseTime** の間隔が、**nsDS5ReplicaBusyWaitTime** に指定された間隔よりも常に 1 秒以上長くなるように設計されています。間隔が長くなると、待機中のサプライヤーは、前のサプライヤーがコンシューマーに再度アクセスできるようになる前に、コンシューマーにアクセスできる可能性が高くなります。

**changetype:modify** を **replace** 操作で使用して、任意のタイミングで **nsDS5ReplicaBusyWaitTime** 属性を設定します。更新セッションがすでに進行中であれば、次の更新セッションで変更が有効になります。

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn= <b>suffixDN</b> ,cn=mapping tree,cn=config
有効な値	有効な整数
デフォルト値	3
構文	整数
例	nsDS5ReplicaBusyWaitTime: 3

### 3.1.9.10. nsDS5ReplicaChangesSentSinceStartup

この読み取り専用属性は、サーバーが起動してからこのレプリカに送信された変更の数を示します。属性の実際の値は、バイナリーblobとして保存されます。Directory Server コンソールでは、この値は比率で、**replica\_id:changes\_sent/changes\_skipped**の形式を取ります。たとえば、100 個の変更が送信され、レプリカ7でスキップされた変更がなかった場合に、属性値はコンソールに 7:100/0 として表示されます。

コマンドラインでは、属性値はバイナリー形式で表示されます。以下に例を示します。

```
nsds5replicaChangesSentSinceStartup:: MToxLzAg
```

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn= <b>suffixDN</b> ,cn=mapping tree,cn=config
Valid Range	0 - 32 ビットの最大整数 (2147483647)
デフォルト値	
構文	整数
例	nsds5replicaChangesSentSinceStartup:: MToxLzAg

### 3.1.9.11. nsDS5ReplicaCredentials

この属性は、**nsDS5ReplicaBindDN** 属性で指定されたバインド DN の認証情報を設定します。Directory Server はこのパスワードを使用してコンシューマーに接続します。

以下の例は、実際のパスワードではなく、**/etc/dirsrv/slapd-instance\_name/dse.ldif** ファイルに保存されている暗号化値を示しています。値を設定するには、これをクリアテキストで設定します (例:

**nsDS5ReplicaCredentials: password**)。Directory Server は、値の保存時に AES リバーシブルパスワードの暗号化スキーマを使用してパスワードを暗号化します。

証明書ベースの認証を使用する場合に、この属性には値が設定されません。

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	有効なパスワード
デフォルト値	
構文	DirectoryString {AES-Base64-algorithm-id}encoded_password
例	nsDS5ReplicaCredentials: {AES-TUhNRONT...} {VogIUB8GG5A...

### 3.1.9.12. nsds5ReplicaEnabled

この属性は、レプリカ合意がアクティブかどうか (つまり、対象の合意に合わせてレプリケーションを実行するか) を設定します。デフォルトでは **on** になっており、レプリケーションが有効化されています。

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsds5ReplicaEnabled: off

### 3.1.9.13. nsds5ReplicaFlowControlPause

このパラメーターは、**nsds5ReplicaFlowControlWindow** パラメーターに設定されたエントリーおよび更新の数に達すると一時停止する時間をミリ秒単位で設定します。**nsds5ReplicaFlowControlWindow** パラメーターおよび **nsds5ReplicaFlowControlPause** パラメーターの両方を更新すると、レプリケーションのスループットを微調整できます。詳細は「[nsds5ReplicaFlowControlWindow](#)」を参照してください。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=replication_agreement_name,cn=replica,cn=suffix_DN,cn=mapping tree,cn=config
有効な値	0 から最大 64 ビットの長さ
デフォルト値	2000
構文	整数
例	nsds5ReplicaFlowControlPause: 2000

### 3.1.9.14. nsds5ReplicaFlowControlWindow

この属性は、サプライヤーが送信し、コンシューマーにより確認されないエントリーおよび更新の最大数を設定します。制限に達すると、サプライヤーは **nsds5ReplicaFlowControlPause** パラメーターに設定された時間、レプリカ合意を一時停止します。**nsds5ReplicaFlowControlWindow** パラメーターおよび **nsds5ReplicaFlowControlPause** パラメーターの両方を更新すると、レプリケーションのスループットを微調整できます。

サプライヤーがエントリーおよび更新をコンシューマーがデータをインポートまたは更新できるよりも早く送信した場合に、この設定を更新します。この場合、サプライヤーのエラーログファイルに以下のメッセージが表示されます。

```
Total update flow control gives time (2000 msec) to the consumer before sending more entries [
msgid sent: xxx, rcv: yyy])
If total update fails you can try to increase nsds5ReplicaFlowControlPause and/or decrease
nsds5ReplicaFlowControlWindow in the replica agreement configuration
```

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=replication_agreement_name,cn=replica,cn=suffix_DN,cn=mapping tree,cn=config
有効な値	0 から最大 64 ビットの長さ
デフォルト値	1000
構文	整数
例	nsds5ReplicaFlowControlWindow: 1000

### 3.1.9.15. nsDS5ReplicaHost

この属性は、コンシューマーレプリカを含む、リモートサーバーのホスト名を設定します。この属性が設定されたら、それを変更することはできません。



パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn=suffi xDN,cn=mapping tree,cn=config
有効な値	有効なホストサーバー名
デフォルト値	
構文	DirectoryString
例	nsDS5ReplicaHost: ldap2.example.com

### 3.1.9.16. nsDS5ReplicaLastInitEnd

このオプションの読み取り専用属性は、コンシューマーレプリカの初期化がいつ終了したかを示します。

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn=suffi xDN,cn=mapping tree,cn=config
有効な値	YYYYMMDDhhmmssZ は、接続が開いた時間 (一般化時間) 形式の日時です。この値は、グリニッジ標準時との関係で時間を示します。時間は 24 時間クロックで設定されます。末尾の Z は、時間がグリニッジ標準時を基準にしていることを示します。
デフォルト値	
構文	GeneralizedTime
例	nsDS5ReplicaLastInitEnd: 20200504121603Z

### 3.1.9.17. nsDS5ReplicaLastInitStart

このオプションの読み取り専用属性は、コンシューマーレプリカの初期化がいつ開始したかを示します。

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn=suffi xDN,cn=mapping tree,cn=config

パラメーター	説明
有効な値	YYYYMMDDhhmmssZ は、接続が開いた時間 (一般化時間) 形式の日時です。この値は、グリニッジ標準時との関係で時間を示します。時間は 24 時間クロックで設定されます。末尾の <b>Z</b> は、時間がグリニッジ標準時を基準にしていることを示します。
デフォルト値	
構文	GeneralizedTime
例	nsDS5ReplicaLastInitStart: 20200503030405

### 3.1.9.18. nsDS5ReplicaLastInitStatus

この読み取り専用属性 (任意) は、コンシューマーの初期化のステータスを指定します。通常、数値コードの後にステータスを説明する短い文字列が続きます。ゼロ (**0**) は成功を意味します。

パラメーター	説明
エン트리 DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn= <b>suffixDN</b> ,cn=mapping tree,cn=config
有効な値	0 (Consumer Initialization Succeeded) (その後に他のステータスメッセージが表示されます)
デフォルト値	
構文	String
例	nsDS5ReplicaLastInitStatus: 0 Consumer Initialization Succeeded

### 3.1.9.19. nsDS5ReplicaLastUpdateEnd

この読み取り専用属性は、最新のレプリケーションスケジュールの更新が終了すると表示されます。

パラメーター	説明
エン트리 DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn= <b>suffixDN</b> ,cn=mapping tree,cn=config

パラメーター	説明
有効な値	YYYYMMDDhhmmssZ は、接続が開いた時間 (一般化時間) 形式の日時です。この値は、グリニッジ標準時との関係で時間を示します。時間は 24 時間クロックで設定されます。末尾の Z は、時間がグリニッジ標準時を基準にしていることを示します。
デフォルト値	
構文	GeneralizedTime
例	nsDS5ReplicaLastUpdateEnd: 20200502175801Z

### 3.1.9.20. nsDS5ReplicaLastUpdateStart

この読み取り専用属性は、最新のレプリケーションスケジュールの更新が開始されるタイミングを示します。

パラメーター	説明
エントリー DN	cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	YYYYMMDDhhmmssZ は、接続が開いた時間 (一般化時間) 形式の日時です。この値は、グリニッジ標準時との関係で時間を示します。時間は 24 時間クロックで設定されます。末尾の Z は、時間がグリニッジ標準時を基準にしていることを示します。
デフォルト値	
構文	GeneralizedTime
例	nsDS5ReplicaLastUpdateStart: 20200504122055Z

### 3.1.9.21. nsds5replicaLastUpdateStatus

各レプリカ合意の読み取り専用 **nsds5replicaLastUpdateStatus** 属性に、Directory Server は、契約の最新ステータスを表示します。ステータスのリストは、[付録B レプリカ合意のステータス](#)を参照してください。

パラメーター	説明
エントリー DN	cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config

パラメーター	説明
有効な値	<a href="#">付録B レプリカ合意のステータス</a> を参照してください。
デフォルト値	
構文	DirectoryString
例	nsds5replicaLastUpdateStatus: Error (0) Replica acquired successfully: Incremental update succeeded

### 3.1.9.22. nsDS5ReplicaPort

この属性は、レプリカを含むリモートサーバーのポート番号を設定します。この属性が設定されたら、それを変更することはできません。

パラメーター	説明
エントリー DN	cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	レプリカを含むリモートサーバーのポート番号
デフォルト値	
構文	整数
例	nsDS5ReplicaPort:389

### 3.1.9.23. nsDS5ReplicaReapActive

この読み取り専用属性は、データベースから古い tombstones (削除されたエントリー) を削除するバックグラウンドタスクがアクティブであるかどうかを指定します。このタスクの詳細は、「[nsDS5ReplicaTombstonePurgeInterval](#)」を参照してください。値がゼロ (0) の場合は、タスクが非アクティブであることを示します。値が 1 の場合は、タスクがアクティブであることを示します。この値が手動で設定されている場合、サーバーは変更要求を無視します。

パラメーター	説明
エントリー DN	cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	0   1
デフォルト値	

パラメーター	説明
構文	整数
例	nsDS5ReplicaReapActive: 0

### 3.1.9.24. nsDS5BeginReplicaRefresh

レプリカを初期化します。この属性はデフォルトでは指定されていません。ただし、この属性に **start** の値が追加されると、サーバーはレプリカを初期化し、属性値を削除します。初期化手順のステータスを監視するには、この属性をポーリングします。初期化が完了すると、属性はエントリーから削除され、他の監視属性を使用して詳細なステータス照会を行うことができます。

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn= <b>suffi xDN</b> ,cn=mapping tree,cn=config
有効な値	stop   start
デフォルト値	
構文	DirectoryString
例	nsDS5BeginReplicaRefresh: start

### 3.1.9.25. nsDS5ReplicaRoot

この属性は、複製された領域のルートに DN を設定します。この属性は、レプリケートされるデータベースの接尾辞と同じ値であり、変更することはできません。

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn= <b>suffi xDN</b> ,cn=mapping tree,cn=config
有効な値	複製されるデータベースの接尾辞 (上記の suffixDN と同じ)
デフォルト値	
構文	DirectoryString
例	nsDS5ReplicaRoot: "dc=example,dc=com"

### 3.1.9.26. nsDS5ReplicaSessionPauseTime

この属性は、次に行われる更新セッションまでの間に、サプライヤーが待機する時間を秒単位で設定します。デフォルト値は **0** です。属性を負の値に設定すると、Directory Server はメッセージおよび **LDAP\_UNWILLING\_TO\_PERFORM** エラーコードをクライアントに送信します。

**nsDS5ReplicaSessionPauseTime** 属性は、**nsDS5ReplicaBusyWaitTime** 属性とともに動作します。この 2 つの属性は、**nsDS5ReplicaSessionPauseTime** の間隔が、**nsDS5ReplicaBusyWaitTime** に指定された間隔よりも常に 1 秒以上長くなるように設計されています。間隔が長くなると、待機中のサプライヤーは、前のサプライヤーがコンシューマーに再度アクセスできるようになる前に、コンシューマーにアクセスできる可能性が高くなります。

- どちらかの属性が指定され、両方が指定されていない場合には、**nsDS5ReplicaSessionPauseTime** は自動的に **nsDS5ReplicaBusyWaitTime** よりも **1 秒** より大きい値に設定されます。
- 両方の属性が指定されていても、**nsDS5ReplicaSessionPauseTime** が **nsDS5ReplicaBusyWaitTime** 以下の場合には、**nsDS5ReplicaSessionPauseTime** は **nsDS5ReplicaBusyWaitTime** よりも **1 秒以上大きい値** に自動的に設定されます。

値の設定時には、**nsDS5ReplicaSessionPauseTime** の間隔が **nsDS5ReplicaBusyWaitTime** に指定した間隔よりも **1 秒以上長くなっている**ことを確認します。サプライヤーの間で許容できる程度にコンシューマーアクセスが分散されるまで、必要に応じてこの間隔を増やします。

**changetype:modify** を **replace** 操作で使用して、任意のタイミングで **nsDS5ReplicaSessionPauseTime** 属性を設定します。更新セッションがすでに進行中であれば、次の更新セッションで変更が有効になります。

Directory Server が **nsDS5ReplicaSessionPauseTime** の値を自動的にリセットする必要がある場合に、値は内部でだけ変更されます。この変更はクライアントには表示されず、設定ファイルには保存されません。外部から見ると、属性値は最初に設定されたとおりに表示されます。

パラメーター	説明
エントリー DN	cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	有効な整数
デフォルト値	0
構文	整数
例	nsDS5ReplicaSessionPauseTime: 0

### 3.1.9.27. nsds5ReplicaStripAttrs

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リストが許可されます。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。

**nsds5ReplicaBootstrapBindMethod** 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リストを追加します。論理的には、**modifiersName** のような操作属性が含まれます。

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。

パラメーター	説明
エントリー DN	cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Range	サポートされるディレクトリー属性のスペース区切りリスト
デフォルト値	
構文	DirectoryString
例	nsds5ReplicaStripAttrs: modifiersname modifytimestamp

### 3.1.9.28. nsDS5ReplicatedAttributeList

使用可能な属性は、コンシューマーサーバーにレプリケートされない属性を指定します。部分的なレプリケーションでは、データベースを低速の接続で複製したり、機密情報を保護しながらも、安全性の低いコンシューマーに複製したりできます。デフォルトでは、すべての属性がレプリケートされ、この属性は存在しません。分数レプリケーションの詳細は、Red Hat Directory Server 管理ガイドのレプリケーションの管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Range	
デフォルト値	
構文	DirectoryString
例	nsDS5ReplicatedAttributeList: (objectclass=*) \$ EXCLUDE accountlockout memberof

### 3.1.9.29. nsDS5ReplicatedAttributeListTotal

使用可能な属性は、全更新中にコンシューマーサーバーにレプリケートされない属性を指定します。

部分的なレプリケーションは、指定した属性のみをレプリケートします。これにより、ネットワークの全体的なパフォーマンスが向上します。ただし、管理者が増分更新時に部分的なレプリケーションを使用して一部の属性を制限する場合があります、これらの属性を全体更新時(またはその逆)に複製することもできます。

デフォルトでは、すべての属性が複製されます。**nsDS5ReplicatedAttributeList** は増分レプリケーションリストを設定します。**nsDS5ReplicatedAttributeList** のみが設定されている場合には、このリストは更新全体にも適用されます。

**nsDS5ReplicatedAttributeListTotal** は、全更新から除外する属性のリストを設定します。

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Range	
デフォルト値	
構文	DirectoryString
例	nsDS5ReplicatedAttributeListTotal: (objectclass=*) \$ EXCLUDE accountlockout

### 3.1.9.30. nsDS5ReplicaTimeout

使用可能な属性は、タイムアウトおよび失敗する前に、リモートレプリカからの応答を待つ秒単位のアウトバウンド LDAP 操作の数を指定します。サーバーがエラーログファイルに **Warning: timed out waiting** のメッセージを書き込む場合は、この属性の値を増やします。

リモートマシンのアクセスログを調べて、操作が実際に継続した時間を調べ、それに応じて **nsDS5ReplicaTimeout** 属性を設定して、パフォーマンスを最適化します。

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Range	0 から最大 32 ビットの整数値 (2147483647)(秒単位)
デフォルト値	120
構文	整数
例	nsDS5ReplicaTimeout: 120

### 3.1.9.31. nsDS5ReplicaTransportInfo

この属性は、レプリカとの間のデータ転送に使用される転送ポートのタイプを設定します。この属性は、設定後は変更できません。

属性は以下の値を取ります。



- **StartTLS**: 接続は、**StartTLS** コマンドで暗号化を使用します。
- **LDAPS**: 接続は TLS 暗号化を使用します。
- **LDAP**: 接続は暗号化されていない LDAP プロトコルを使用します。この値は、**nsDS5ReplicaTransportInfo** 属性が設定されていない場合にも使用されます。

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn= <b>suffixDN</b> ,cn=mapping tree,cn=config
有効な値	StartTLS   LDAPS   LDAP
デフォルト値	absent
構文	DirectoryString
例	nsDS5ReplicaTransportInfo: StartTLS

### 3.1.9.32. nsDS5ReplicaUpdateInProgress

この読み取り専用属性は、レプリケーションの更新が進行中であることを示します。

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn= <b>suffixDN</b> ,cn=mapping tree,cn=config
有効な値	true   false
デフォルト値	
構文	DirectoryString
例	nsDS5ReplicaUpdateInProgress: true

### 3.1.9.33. nsDS5ReplicaUpdateSchedule

この複数値属性はレプリケーションスケジュールを指定し、変更できます。この属性に加えられた変更は即座に有効になります。この値を変更すると、レプリケーションを一時停止して後で再開するのに便利です。たとえば、この値を **0000-0001 0** にすると、サーバーがこのレプリカ合意の更新の送信を停止します。サーバーは、後で再生できるように保存し続けます。値が後で **0000-2359 0123456** に戻された場合は、レプリケーションがすぐに再開し、保留中のすべての変更が送信されます。

パラメーター	説明
--------	----

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Range	<b>XXXX-YYYY 0123456</b> として表示される時間スケジュール。ここでは、 <b>XXXX</b> は開始時間、 <b>YYYY</b> は終了時間、番号 <b>0123456</b> は曜日 (日曜から開始) を表します。
デフォルト値	0000-2359 0123456 (常時)
構文	整数
例	nsDS5ReplicaUpdateSchedule: 0000-2359 0123456

### 3.1.9.34. nsDS5ReplicaWaitForAsyncResults

レプリケーション環境では、**nsDS5ReplicaWaitForAsyncResults** パラメーターは、コンシューマーが準備状態にない場合に待機する時間をミリ秒単位で設定します。

パラメーターを **0** に設定すると、デフォルト値が使用される点に注意してください。

パラメーター	説明
エントリー DN	cn= <b>ReplicationAgreementName</b> ,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Range	0 - 32 ビットの最大整数 (2147483647)
デフォルト値	100
構文	整数
例	nsDS5ReplicaWaitForAsyncResults: 100

### 3.1.9.35. nsDS50ruv

この属性は、このレプリカ合意のコンシューマーから読み取られた最後のレプリカ更新ベクトル (RUV) を保存します。常に存在し、変更してはなりません。

### 3.1.9.36. nsruvReplicaLastModified

この属性には、レプリカのエントリーが変更され、changelog が更新された最新の時間が含まれます。

### 3.1.9.37. nsds5ReplicaProtocolTimeout

サーバーを停止したり、レプリカを無効にしたり、レプリカ合意を削除したりすると、サーバーに負荷がかかっているときにレプリケーションを停止するまでの待機時間のタイムアウトがあります。

す。**nsds5ReplicaProtocolTimeout** 属性はこのタイムアウトを設定するために使用され、デフォルト値は 120 秒です。

2 分のタイムアウトが長すぎる、または十分に長くないシナリオが存在する可能性があります。たとえば、特定のレプリカ合意は、シャットダウン中にレプリケーションセッションを終了する前により多くの時間が必要になる場合があります。

この属性は、バックエンドの主要レプリケーション設定エントリーに追加できます。

パラメーター	説明
エントリー DN	cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
Valid Range	0 から最大 32 ビット整数 (2147483647) (秒単位)
デフォルト値	120
構文	整数
例	nsds5ReplicaProtocolTimeout: 120

**nsds5ReplicaProtocolTimeout** 属性をレプリカ合意に追加することもできます。レプリカ合意プロトコルのタイムアウトは、メインのレプリカ設定エントリーに設定されたタイムアウトをオーバーライドします。これにより、レプリカ合意ごとに異なるタイムアウトが可能になります。レプリケーションセッションが進行中、新しいタイムアウトによってそのセッションが中断され、サーバーのシャットダウンが許可されます。

### 3.1.10. cn=syncAgreementName,cn=WindowsReplica,cn=suffixName,cn=mapping tree,cn=config 下の同期属性

同期合意に関連する同期属性

は、**cn=syncAgreementName,cn=WindowsReplica,cn=suffixDN,cn=mapping tree,cn=config** に保存されます。**cn=syncAgreementName** エントリーは、**nsDSWindowsReplicationAgreement** オブジェクトクラスのインスタンスです。サーバーが同期合意の設定属性を考慮に入れるには、**最上位** のオブジェクトクラスに加えて、これらのオブジェクトクラスがエントリーに存在する必要があります。同期合意は、Windows Active Directory サーバーとの同期が有効なデータベースでのみ設定されます。

表3.6 レプリカと同期合意との間で共有される属性のリスト

cn	nsDS5ReplicaLastUpdateEnd
description	nsDS5ReplicaLastUpdateStart
nsDS5ReplicaBindDN (Windows 同期マネージャー ID)	nsDS5ReplicaLastUpdateStatus
nsDS5ReplicaBindMethod	nsDS5ReplicaPort
nsDS5ReplicaBusyWaitTime	nsDS5ReplicaRoot

nsDS5ReplicaChangesSentSinceStartup	nsDS5ReplicaSessionPauseTime
nsDS5ReplicaCredentials (Windows 同期マネージャーのパスワード)	nsDS5ReplicaTimeout
nsDS5ReplicaHost (Windows ホスト)	nsDS5ReplicaTransportInfo
nsDS5ReplicaLastInitEnd	nsDS5ReplicaUpdateInProgress
nsDS5ReplicaLastInitStart	nsDS5ReplicaUpdateSchedule
nsDS5ReplicaLastInitStatus	nsDS50ruv
winSyncMoveAction	winSyncInterval
nsds5ReplicaStripAttrs	

### 3.1.10.1. nsds7DirectoryReplicaSubtree

同期している Directory Server サブツリーの接尾辞または DN。

パラメーター	説明
エントリー DN	cn= <b>syncAgreementName</b> ,cn=replica,cn= <b>suffixDN</b> ,cn=mapping tree,cn=config
有効な値	有効な接尾辞またはサブ接尾辞
デフォルト値	
構文	DirectoryString
例	nsDS7DirectoryReplicaSubtree: ou=People,dc=example,dc=com

### 3.1.10.2. nsds7DirsyncCookie

この文字列は Active Directory DirSync により作成され、最終同期時の Active Directory Server の状態を示します。以前の cookie は、各 Directory Server の更新のたびに Active Directory に送信され、新しい Cookie が Windows ディレクトリーデータとともに返されます。これは、最後の同期が取得されてから変更されたエントリーだけを指します。

パラメーター	説明
エントリー DN	cn= <b>syncAgreementName</b> ,cn=replica,cn= <b>suffixDN</b> ,cn=mapping tree,cn=config

パラメーター	説明
有効な値	任意の文字列
デフォルト値	
構文	DirectoryString
例	nsDS7DirsyncCookie::khDKJFBZsjBDSCkjsdhIU74D JJVBXDhfvjmfvbhzxj

### 3.1.10.3. nsds7NewWinGroupSyncEnabled

この属性は、Directory Server で新しいグループを作成して、Windows 同期ピアで作成された新しいグループを自動的に同期するかどうかを設定します。

パラメーター	説明
エントリー DN	cn=syncAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	on   off
デフォルト値	
構文	DirectoryString
例	nsDS7NewWinGroupSyncEnabled: on

### 3.1.10.4. nsds7NewWinUserSyncEnabled

この属性は、Directory Server で新しいエントリーを作成して、Windows 同期ピアで作成された新しいエントリーを自動的に同期するかどうかを設定します。

パラメーター	説明
エントリー DN	cn=syncAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
有効な値	on   off
デフォルト値	
構文	DirectoryString
例	nsDS7NewWinUserSyncEnabled: on

### 3.1.10.5. nsds7WindowsDomain

この属性は、Windows 同期ピアが属する Windows ドメインの名前を設定します。

パラメーター	説明
エントリー DN	cn= <b>syncAgreementName</b> ,cn=replica,cn= <b>suffixDN</b> ,cn=mapping tree,cn=config
有効な値	有効なドメイン名
デフォルト値	
構文	DirectoryString
例	nsDS7WinndowsDomain: DOMAINWORLD

### 3.1.10.6. nsds7WindowsReplicaSubtree

同期している Windows サブツリーの接尾辞または DN。

パラメーター	説明
エントリー DN	cn= <b>syncAgreementName</b> ,cn=replica,cn= <b>suffixDN</b> ,cn=mapping tree,cn=config
有効な値	有効な接尾辞またはサブ接尾辞
デフォルト値	
構文	DirectoryString
例	nsDS7WindowsReplicaSubtree: cn=Users,dc=domain,dc=com

### 3.1.10.7. oneWaySync

この属性は、同期を実行する方向を設定します。これは、Active Directory サーバーから Directory Server へ、または Directory Server から Active Directory サーバーのいずれかになります。

この属性がない場合 (デフォルト)、同期合意は **双方向** であるため、両方のドメインで行った変更が同期されます。

パラメーター	説明
エントリー DN	cn= <b>syncAgreementName</b> ,cn=replica,cn= <b>suffixDN</b> ,cn=mapping tree,cn=config

パラメーター	説明
有効な値	toWindows   fromWindows   null
デフォルト値	
構文	DirectoryString
例	oneWaySync: fromWindows

### 3.1.10.8. winSyncInterval

この属性は、Directory Server が Windows 同期ピアをポーリングして Active Directory エントリーの変更を検索する頻度を秒単位で設定します。このエントリーが設定されていない場合には、Directory Server は Windows サーバーを 5 分ごとにチェックします。つまり、デフォルト値は **300** (300 秒) です。

この値を低く設定すると、Active Directory の変更をディレクトリーサーバーにすばやく書き込むことができ、ディレクトリー検索に時間がかかりすぎる場合は、この値を高く設定できます。

パラメーター	説明
エントリー DN	cn= <b>syncAgreementName</b> ,cn=replica,cn= <b>suffixDN</b> ,cn=mapping tree,cn=config
有効な値	1 から最大 32 ビットの整数値 (2147483647)
デフォルト値	300
構文	整数
例	winSyncInterval: 600

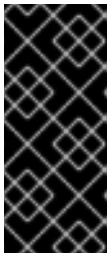
### 3.1.10.9. winSyncMoveAction

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** 属性に基づいて関連されます。Synchronization プラグインは、以前に同期されたエントリー (**samAccount/uid** の関係に基づく) 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は、Synchronization プラグインはエントリーが同期されなくなったことを認識します。

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、何も発生しません (これはデフォルトの動作です)。

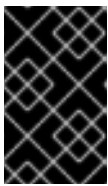
- **unsync** は、Directory Server エントリーから同期関連の属性 (**ntUser** または **ntGroup**) を削除しますが、Directory Server エントリーはそのまま残されます。ActiveDirectory と Directory Server のエントリーは連携して存在します。



### 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。

パラメーター	説明
エントリー DN	cn= <b>syncAgreementName</b> ,cn=replica,cn= <b>suffixDN</b> ,cn=mapping tree,cn=config
有効な値	none   delete   unsync
デフォルト値	none
構文	DirectoryString
例	winSyncMoveAction: unsync

### 3.1.11. cn=monitor

サーバーの監視に使用される情報は **cn=monitor** 下に保存されます。このエントリーとその子は読み取り専用で、クライアントは直接変更できません。サーバーはこの情報を自動的に更新します。本セクションでは、**cn=monitor** 属性を説明します。アクセス制御の設定向けにユーザーが変更できる属性は **aci** 属性のみです。

**cn=config** の **nsslapd-counters** 属性が **on** (デフォルト設定) に設定されている場合は、32 ビットマシンや、32 ビットバージョンの Directory Server でも、Directory Server インスタンスにより保持されるすべてのカウンターは、64 ビットの整数を使用して増分します。**cn=monitor** エントリーでは、**opsinitiated**、**opscompleted**、**entriessent**、および **bytessent** カウンターで 64 ビットの整数が使用されます。





## 注記

**nsslapd-counters** 属性は、これらの特定のデータベースおよびサーバーカウンターの 64 ビットサポートを有効にします。64 ビットの整数を使用するカウンターは設定できません。64 ビットの整数は、許可されるすべてのカウンターに対して有効であるか、許可されているすべてのカウンターに対して無効にされます。

## connection

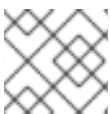
この属性は、開放されている接続および関連するステータスおよびパフォーマンス関連情報および値をリスト表示します。これらは次の形式で提供されます。

```
connection: A:YYYYMMDDhhmmssZ:B:C:D:E:F:G:H:I:IP_address
```

以下に例を示します。

```
connection: 69:20200604081953Z:6086:6086:-
:cn=proxy,ou=special_users,dc=example,dc=test:0:11:27:7448846:ip=192.0.2.1
```

- **A** は接続番号で、この接続に関連する接続テーブルのロット数です。これは、この接続が開放されたときに、アクセスログメッセージで **slot=A** としてログに記録された数字で、通常は接続に関連付けられたファイル記述子に対応します。**dTableSize** 属性は、接続テーブルの合計サイズを表示します。
- **YYYYMMDDhhmmssZ** は、接続が開いた時間 (GeneralizedTime) 形式の日時です。この値は、グリニッジ標準時との関係で時間を示します。
- **b** は、この接続で受信する操作の数です。
- **C** は、完了した操作の数です。
- サーバーがネットワークから BER を読み取る処理中である場合、**D** は **r** であり、それ以外は空になります。(この例のように) この値は、通常空です。
- **e** はバインド DN です。これは空であるか、匿名接続の **NULLDN** の値を持つことがあります。
- **f** は接続の最大スレッド状態です。**1** では最大スレッドにあり、**0** では最大スレッドにありません。
- **G** は、このスレッドが最大スレッド値に達した回数です。
- **H** は、最大スレッド数によってブロックされた操作の数です。
- **I** は、ログに **conn=connection\_ID** として報告される接続 ID です。
- **ip\_address** は、LDAP クライアントの IP アドレスです。



## 注記

開始操作および完了した操作の **B** および **C** は同等であるべきです。

## currentConnections

この属性は、現在開いていてアクティブな Directory Server 接続の数を表示します。

## totalConnections

この属性は、Directory Server 接続の合計数を表示します。この数には、**currentConnections** に加えて、サーバーが最後に起動されてから開かれた接続と閉じられた接続が含まれます。

### dTableSize

この属性は、Directory Server 接続テーブルのサイズを表示します。各接続はこのテーブルのロットに関連付けられ、通常はこの接続で使用されるファイル記述子に対応します。詳細は、「[nsslapd-conntablesize](#)」を参照してください。

### readWaiters

この属性は、一部の要求が保留中であり、現在 DirectoryServer のスレッドで処理されていない接続の数を示します。

### opsinitiated

この属性は、開始された Directory Server 操作の数を表示します。

### opsCompleted

この属性は、完了した Directory Server 操作の数を表示します。

### entriesSent

この属性は、Directory Server で送信されるエントリーの数を表示します。

### bytesSent

この属性は、Directory Server が送信するバイト数を表示します。

### currentTime

この属性は、グリニッジ標準時 (**20200202131102Z** など、**generalizedTime** 構文 **Z** 表記で表される) で指定されている現在の時間を表示します。

### startTime

この属性は、グリニッジ標準時で指定した Directory Server の起動時間を表示します。これは、**generalizedTime** 構文 **Z** 表記で示されます。たとえば、**20200202131102Z** です。

### version

この属性は、Directory Server のベンダー、バージョン、およびビルド番号を表示します。たとえば、**Red Hat/11.3.1 B2020.274.08** です。

### threads

この属性は、Directory Server が使用するスレッド数を表示します。これは **cn=config** の **nsslapd-threadnumber** に対応している必要があります。

### nbackEnds

この属性は、Directory Server データベースバックエンドの数を示します。

### backendMonitorDN

この属性は、各 Directory Server データベースバックエンドの DN を示します。データベースの監視に関する詳細は、以下のセクションを参照してください。

- 「[cn=attributeName,cn=encrypted attributes,cn=database\\_name,cn=ldbm database,cn=plugins,cn=config](#) 下のデータベース属性」

- 「cn=database,cn=monitor,cn=ldbm database,cn=plugins,cn=config 下のデータベース属性」
- 「cn=monitor,cn=database instance name,cn=chaining database,cn=plugins,cn=config 下のデータベースリンク属性」

### 3.1.12. cn=replication

このエントリーには属性がありません。レガシーレプリケーションを設定する場合には、これらのエントリーはプレースホルダーとして機能する **cn=replication** ノードに保存されます。

### 3.1.13. cn=sasl

SASL マッピング設定を含むエントリーは、**cn=mapping,cn=sasl,cn=config** に保存されます。**cn=sasl** エントリーは、**nsContainer** オブジェクトクラスのインスタンスです。各マッピングは、**nsSaslMapping** オブジェクトクラスのインスタンスです。

#### 3.1.13.1. nsSaslMapBaseDNTemplate

この属性には、SASL ID マッピングで使用される検索ベース DN テンプレートが含まれます。

パラメーター	説明
エントリー DN	cn= <b>mapping_name</b> ,cn=mapping,cn=sasl,cn=config
有効な値	任意の有効な DN
デフォルト値	
構文	IA5String
例	nsSaslMapBaseDNTemplate: ou=People,dc=example,dc=com

#### 3.1.13.2. nsSaslMapFilterTemplate

この属性には、SASL ID マッピングで使用される検索フィルターテンプレートが含まれます。

パラメーター	説明
エントリー DN	cn= <b>mapping_name</b> ,cn=mapping,cn=sasl,cn=config
有効な値	任意の文字列
デフォルト値	
構文	IA5String
例	nsSaslMapFilterTemplate: (cn=\1)

### 3.1.13.3. nsSaslMapPriority

Directory Server を使用すると、複数の簡易認証およびセキュリティー層 (SASL) マッピングを設定できます。SASL フォールバックが **nsslapd-sasl-mapping-fallback** パラメーターによって有効になっている場合には、**nsSaslMapPriority** 属性を設定して個別の SASL マッピングの優先順位を付けることができます。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn= <b>mapping_name</b> ,cn=mapping,cn=sasl,cn=config
有効な値	1(最も高い優先度)- 100(最も低い優先度)
デフォルト値	100
構文	整数
例	nsSaslMapPriority: 100

### 3.1.13.4. nsSaslMapRegexString

この属性には、SASL アイデンティティー文字列をマッピングするために使用される正規表現が含まれます。

パラメーター	説明
エントリー DN	cn= <b>mapping_name</b> ,cn=mapping,cn=sasl,cn=config
有効な値	有効な正規表現
デフォルト値	
構文	IA5String
例	nsSaslMapRegexString: \(.*\)

## 3.1.14. cn=SNMP

SNMP 設定属性は **cn=SNMP,cn=config** に保存されます。**cn=SNMP** エントリーは、**nsSNMP** オブジェクトクラスのインスタンスです。

### 3.1.14.1. nssnmpenabled

この属性は、SNMP を有効にするかどうかを設定します。

パラメーター	説明
エントリー DN	cn=SNMP,cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nssnmpenabled: off

### 3.1.14.2. nssnmporganization

この属性は、Directory Server が属する組織を設定します。

パラメーター	説明
エントリー DN	cn=SNMP,cn=config
有効な値	組織名
デフォルト値	
構文	DirectoryString
例	nssnmporganization: Red Hat, Inc.

### 3.1.14.3. nssnmplocation

この属性は、Directory Server が置かれている企業または組織内の場所を設定します。

パラメーター	説明
エントリー DN	cn=SNMP,cn=config
有効な値	ロケーション
デフォルト値	
構文	DirectoryString
例	nssnmplocation: B14

### 3.1.14.4. nssnmpcontact

この属性は、Directory Server を管理するユーザーのメールアドレスを設定します。

パラメーター	説明
エントリー DN	cn=SNMP,cn=config
有効な値	メールアドレスへのお問い合わせ
デフォルト値	
構文	DirectoryString
例	nssnmpcontact: <a href="mailto:jerome@example.com">jerome@example.com</a>

### 3.1.14.5. nssnmpdescription

Directory Server インスタンスの一意の説明を指定します。

パラメーター	説明
エントリー DN	cn=SNMP,cn=config
有効な値	説明
デフォルト値	
構文	DirectoryString
例	nssnmpdescription: Employee directory instance

### 3.1.14.6. nssnmpmasterhost

**nssnmpmasterhost** は非推奨になりました。この属性は、**net-snmp** が導入され非推奨となりました。属性は、引き続き **dse.ldif** で表示されますが、デフォルト値はありません。

パラメーター	説明
エントリー DN	cn=SNMP,cn=config
有効な値	マシンホスト名または localhost
デフォルト値	<blank>
構文	DirectoryString
例	nssnmpmasterhost: localhost

### 3.1.14.7. nssnmpmasterport

**nssnmpmasterport** 属性は **net-snmp** が導入され、非推奨となりました。属性は、引き続き **dse.ldif** で表示されますが、デフォルト値はありません。

パラメーター	説明
エン트리 DN	cn=SNMP,cn=config
有効な値	オペレーティングシステムに依存するポート番号。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。
デフォルト値	<blank>
構文	整数
例	nssnmpmasterport: 199

### 3.1.15. SNMP の静的属性

表3.7「SNMP の静的属性」LDAP および SNMP クライアントで利用可能な統計をリストする読み取り専用属性が含まれます。特に明記されていない限り、指定された属性の値は、サーバーが受信した要求の数、または起動後にサーバーが返した結果になります。これらの属性の一部は、Directory Server によって使用されないか、適用されませんが、SNMP クライアント向けに存在する必要があります。

**cn=config** の **nsslapd-counters** 属性が **on** (デフォルト設定) に設定されている場合は、32 ビットマシンや、32 ビットバージョンの Directory Server でも、Directory Server インスタンスにより保持されるすべてのカウンターは、64 ビットの整数を使用して増分します。SNMP 統計属性はすべて 64 ビット整数を使用します (設定されている場合)。



#### 注記

**nsslapd-counters** 属性は、これらの特定のデータベースおよびサーバーカウンターの 64 ビット整数を有効にします。64 ビットの整数を使用するカウンターは設定できません。64 ビットの整数は、許可されるすべてのカウンターに対して有効であるか、許可されているすべてのカウンターに対して無効にされます。

表3.7 SNMP の静的属性

属性	説明
AnonymousBinds	これは、匿名バインド要求の数を示しています。
UnAuthBinds	これは、認証されていない (匿名) バインドの数を示しています。
SimpleAuthBinds	これは、LDAP の単純なバインド要求 (DN およびパスワード) の数を示しています。

属性	説明
StrongAuthBinds	これは、すべての SASL メカニズムの LDAP SASL バインド要求の数を示しています。
BindSecurityErrors	これは、バインド要求に無効なパスワードが指定されている回数を示します。
InOps	これは、サーバーによって受信されるすべての要求総数を示します。
ReadOps	使用されていません。この値は、常に <b>0</b> です。
CompareOps	これは、LDAP 比較要求の数を示しています。
AddEntryOps	これは、LDAP 追加要求の数を示しています。
RemoveEntryOps	これは、LDAP 削除要求の数を示しています。
ModifyEntryOps	これは、LDAP 変更要求の数を示しています。
ModifyRDNops	これは、LDAP 変更 RDN (modrdn) 要求の数を示しています。
ListOps	使用されていません。この値は、常に <b>0</b> です。
SearchOps	LDAP 検索要求の数を示します。
OneLevelSearchOps	これは、1 レベルの検索操作の数を示しています。
WholeSubtreeSearchOps	これは、サブツリーレベルの検索操作の数を示しています。
Referrals	これは、返された LDAP 参照の数を示しています。
Chainings	使用されていません。この値は、常に <b>0</b> です。
SecurityErrors	これは、無効なパスワード、不明な認証方法、または強力な認証が必要など、セキュリティ関連のエラー数を示しています。
Errors	これは、返されたエラーの数を示しています。
Connections	現在開いている接続の数を示しています。



属性	説明
ConnectionSeq	これにより、現在閉じている接続も開いている接続も含めた、開放されている合計接続数が表示されます。
BytesRecv	受信したバイト数を表示します。
BytesSent	これは、送信されたバイト数を示します。
EntriesReturned	これは、検索結果として返されたエントリーの数を示しています。
ReferralsReturned	これは、検索結果として返された参照の情報を提供します (継続参照)。
MasterEntries	使用されていません。この値は、常に <b>0</b> です。
CopyEntries	使用されていません。この値は、常に <b>0</b> です。
CacheEntries <sup>[a]</sup>	サーバーにデータベースバックエンドが1つしかない場合は、エントリーキャッシュにキャッシュされたエントリーの数になります。サーバーに複数のデータベースバックエンドがある場合、この値は <b>0</b> になり、それぞれの監視エントリーで詳細情報が表示されます。
CacheHits	サーバーにデータベースバックエンドが1つしかない場合、これは検索結果に対してデータベースからではなく、エントリーキャッシュから返されるエントリーの数です。サーバーに複数のデータベースバックエンドがある場合、この値は <b>0</b> になり、それぞれの監視エントリーで詳細情報が表示されます。
SlaveHits	使用されていません。この値は、常に <b>0</b> です。
<p>[a] <b>CacheEntries</b> および <b>CacheHits</b> は 10 秒ごとに更新されます。Red Hat は、このデータベース情報およびその他のデータベース情報にデータベースバックエンド固有のモニターエントリーを使用することを強く推奨します。</p>	

### 3.1.16. cn=tasks

一部のコア Directory Server タスクは、LDAP ツールを使用してディレクトリーエントリーを編集することで開始できます。これらのタスクエントリーは、**cn=tasks** に含まれています。各タスクは、以下のようなエントリーを更新して呼び出すことができます。

```
dn: cn=task_id,cn=task_type,cn=tasks,cn=config
```

```
...
```

Directory Server 8.0 よりも前の Red Hat Directory Server デプロイメントでは、多くの Directory Server タスクが Administration Server によって管理されました。これらのタスクは、バージョン 8.0

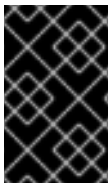
のコア Directory Server 設定に移動し、**cn=tasks** エントリーの下で Directory Server により呼び出され、管理されています。

**cn=tasks** エントリーで次のタスクが管理されます。

- 「cn=import」
- 「cn=export」
- 「cn=backup」
- 「cn=restore」
- 「cn=index」
- 「cn=schema リロードタスク」
- 「cn=memberof task」
- 「cn=fixup リンクされた属性」
- 「cn=syntax validate」
- 「cn=USN 破棄クリーンアップタスク」
- 「cn=cleanallruv」
- 「cn=abort cleanallruv」
- 「cn=automember rebuild membership」
- 「cn=automember export updates」
- 「cn=automember map updates」

これらのタスクの一般的な属性は、「cn=tasks の下にあるエントリーのタスク呼び出し属性」に記載されています。

**cn=tasks** エントリー自体には属性がなく、個別のタスクエントリーの親およびコンテナエントリーとして機能します。



### 重要

タスクエントリーは、永続的な設定エントリーではありません。このエントリーは、タスク操作が実行中であるか、**ttl** の有効期限が切れるまで、設定ファイルにだけ存在します。その後、エントリーは自動的にサーバーにより削除されます。

#### 3.1.16.1. cn=tasks の下にあるエントリーのタスク呼び出し属性

Directory Server インスタンスを管理する 5 つのタスクには、個々の操作を開始および特定する設定エントリーがあります。これらのタスクエントリーは、同じオブジェクトクラス **extensibleObject** のインスタンスで、このエントリーには Directory Server タスクの状態と動作を記述する特定の共通属性があります。タスクタイプは、インポート、エクスポート、バックアップ、復元、インデックス、スキーマのリロード、およびメンバーです。

**cn**

**cn** 属性は、開始する新しいタスク操作を特定します。**cn** 属性の値は、新しいタスクを定義する限りすべて使用できます。

パラメーター	説明
エントリー DN	cn=task_name,cn=task_type,cn=tasks,cn=config
有効な値	任意の文字列
デフォルト値	
構文	DirectoryString
例	cn: タスクエントリー名の例

### nsTaskStatus

この属性には、累積の統計や現在の出力メッセージなどのタスクのステータスの変更情報が含まれます。属性の全内容は、プロセスが実行されている限り定期的に更新できます。

この属性値はサーバーによって設定されるため、編集しないでください。

パラメーター	説明
エントリー DN	cn=task_name,cn=task_type,cn=tasks,cn=config
有効な値	任意の文字列
デフォルト値	
構文	大文字と小文字を区別する文字列
例	nsTaskStatus: エントリーの読み込みなど

### nsTaskLog

このエントリーには、警告メッセージおよび情報メッセージの両方など、タスクのすべてのログメッセージが含まれます。新しいメッセージはエントリー値の最後に追加されるため、この属性値は、デフォルトでは元の内容を消去しないので、数値が大きくなります。

**nsTaskExitCode** が **0** である正常なタスク操作は、**nsTaskLog** 属性にのみ記録されます。エラーを示すゼロ以外の応答は、エラーとしてエラーログに記録される場合がありますが、エラーメッセージは **nsTaskLog** 属性にのみ記録されます。このため、**nsTaskLog** 属性の情報を使用して、実際に発生したエラーが分かります。

この属性値はサーバーによって設定されるため、編集しないでください。

パラメーター	説明
エントリー DN	cn=task_name,cn=task_type,cn=tasks,cn=config
有効な値	任意の文字列
デフォルト値	
構文	大文字と小文字を区別する文字列
例	nsTaskLog: example...

### nsTaskExitCode

この属性には、タスクの終了コードが含まれます。この属性は、タスクの完了後にのみ存在し、値はタスクが完了した場合にのみ有効になります。結果コードは、「[LDAP の結果コード](#)」に記載されている LDAP 終了コードに指定できますが、**0** 値のみが成功に相当します。他の結果コードはエラーです。

この属性値はサーバーによって設定されるため、編集しないでください。

パラメーター	説明
エントリー DN	cn=task_name,cn=task_type,cn=tasks,cn=config
有効な値	0 (成功) から 97 <sup>[a]</sup>
デフォルト値	
構文	整数
例	nsTaskExitCode: 0
[a] <b>0</b> 以外の応答はエラーです。	

### nsTaskCurrentItem

この属性は、タスクがサブタスクに分割できると仮定して、タスク操作が完了したサブタスクの数を表示します。タスクが1つしかない場合は、タスクの実行中に **nsTaskCurrentItem** が **0**、タスクの完了時には **1** になります。このように、属性は進捗バーに似ています。**nsTaskCurrentItem** 属性に **nsTaskTotalItems** と同じ値がある場合は、タスクが完了します。

この属性値はサーバーによって設定されるため、編集しないでください。

パラメーター	説明
エントリー DN	cn=task_name,cn=task_type,cn=tasks,cn=config

パラメーター	説明
有効な値	0 から最大 32 ビットの整数値 (2147483647)
デフォルト値	
構文	整数
例	nsTaskCurrentItem: 148

### nsTaskTotalItems

この属性は、タスク操作で完了する必要があるサブタスクの合計数を示します。**nsTaskCurrentItem** 属性に **nsTaskTotalItems** と同じ値がある場合は、タスクが完了します。

この属性値はサーバーによって設定されるため、編集しないでください。

パラメーター	説明
エントリー DN	cn=task_name,cn=task_type,cn=tasks,cn=config
有効な値	0 から最大 32 ビットの整数値 (2147483647)
デフォルト値	
構文	整数
例	nsTaskTotalItems: 152

### nsTaskCancel

この属性を使用すると、進行中にタスクを中断できます。この属性は、ユーザーが変更できます。

パラメーター	説明
エントリー DN	cn=task_name,cn=task_type,cn=tasks,cn=config
有効な値	true   false
デフォルト値	
構文	大文字と小文字を区別しない文字列
例	nsTaskCancel: true

### ttl

この属性は、タスクが終了または中止した後に、タスクエントリーが DSE に留まる時間 (秒単位) を設定します。**ttl** 属性を設定すると、終了コードを失うことなく、タスクエントリーをポーリングして新しいステータス情報を取得できます。**ttl** 属性を **0** に設定すると、エントリーがキャッシュされません。

パラメーター	説明
エントリー DN	cn=task_name,cn=task_type,cn=tasks,cn=config
有効な値	0(キャッシュ不可) から最大 32 ビットの整数値 (2147483647)
デフォルト値	
構文	DirectoryString
例	ttl: 120

### 3.1.16.2. cn=import

LDIF ファイルまたは複数の LDIF ファイルは、タスクのパラメーターを定義してタスクを開始する特別なタスクエントリーを作成することで、コマンドラインでインポートできます。タスクが完了するとすぐに、タスクエントリーがディレクトリーから削除されます。

**cn=import** エントリーは、インポートタスク操作のコンテナエントリーです。**cn=import** エントリー自体には属性はありませんが、このエントリー内にあるタスクエントリーごとに (**cn=task\_ID**、**cn=import**、**cn=tasks**、**cn=config**) 以下の属性を使用してインポートタスクを定義します。

**cn=import** のインポートタスクエントリーには、インポートする LDIF ファイル (**nsFilename** 属性) と、ファイルをインポートするインスタンスの名前 (**nsInstance** 属性) が含まれている必要があります。さらに、タスクを識別するために一意の **cn** を含める必要があります。以下に例を示します。

```
dn: cn=example import,cn=import,cn=tasks,cn=config
objectclass: extensibleObject
cn: example import
nsFilename: /home/files/example.ldif
nsInstance: userRoot
```

インポート操作が実行されると、タスクエントリーには、「**cn=tasks** の下にあるエントリーのタスク呼び出し属性」に記載のサーバー生成タスク属性がすべて含まれます。

**ldif2db** と **ldif2db.pl** スクリプトのオプションと同様に、インポート操作の調整に使用できる任意の属性があります。

- **nsIncludeSuffix**: インポートする接尾辞を指定する **-s** オプションと類似しています。
- **nsExcludeSuffix**: インポートから除外する接尾辞またはサブツリーを指定する **-x** オプションと類似しています。
- **nsImportChunkSize**: インポート時新しいパスの開始を上書きして、チャンクをマージする **-c** オプションと類似しています。

- `nsImportIndexAttrs`: 属性インデックスをインポートするかどうかを設定します (スクリプトオプションに推論はありません)。
- `nsUniqueldGenerator`: エントリーの一意の ID 番号を生成する `-g` オプションと類似しています。
- `nsUniqueldGeneratorNamespace`: エントリーの一意の名前ベースの ID を生成する、`-G` オプションと類似しています。

## nsFilename

`nsFilename` 属性には、Directory Server インスタンスにインポートする LDIF ファイルのパスとファイル名が含まれます。複数のファイルをインポートするには、この属性のインスタンスを複数追加します。以下に例を示します。

```
nsFilename: file1.ldif
nsFilename: file2.ldif
```

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=import,cn=tasks,cn=config
有効な値	任意の文字列
デフォルト値	
構文	大文字と小文字を区別する文字列 (複数値)
例	nsFilename: /home/jsmith/example.ldif

## nsInstance

この属性は、`userRoot`、`slapd-example` などのファイルをインポートするデータベースインスタンスの名前を提供します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=import,cn=tasks,cn=config
有効な値	Directory Server インスタンスデータベースの名前 (任意の文字列)
デフォルト値	
構文	大文字と小文字を区別する文字列
例	nsInstance: userRoot

## nsIncludeSuffix

この属性は、LDIF ファイルからインポートする特定の接尾辞またはサブツリーを識別します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=import,cn=tasks,cn=config
有効な値	任意の DN
デフォルト値	
構文	DN、多値
例	nsIncludeSuffix: ou=people,dc=example,dc=com

### nsExcludeSuffix

この属性は、インポートから除外する LDIF ファイルの接尾辞またはサブツリーを識別します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=import,cn=tasks,cn=config
有効な値	任意の DN
デフォルト値	
構文	DN、多値
例	nsExcludeSuffix: ou=machines,dc=example,dc=com

### nsImportChunkSize

この属性は、インポート操作中に保持するチャンクの数を実数値として定義し、インポート中にサーバーが検出した、新規パスを開始してチャンクをマージするタイミングの内容を上書きします。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=import,cn=tasks,cn=config
有効な値	0 から最大 32 ビットの整数値 (2147483647)
デフォルト値	0
構文	整数
例	nsImportChunkSize: 10

### nsImportIndexAttrs



この属性は、データベースインスタンスにインポートされる属性をデプロイするかどうかを設定します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=import,cn=tasks,cn=config
有効な値	true   false
デフォルト値	true
構文	大文字と小文字を区別しない文字列
例	nsImportIndexAttrs: true

### nsUniqueldGenerator

これにより、インポートされたエントリーの一意の ID を生成するかどうかを設定されます。デフォルトでは、この属性は時間ベースの ID を生成します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=import,cn=tasks,cn=config
有効な値	none (一意 ID なし)   empty (タイムベースの ID)   deterministic <b>namespace</b> (名前ベース ID)
デフォルト値	空
構文	大文字と小文字を区別しない文字列
例	nsUniqueldGenerator:

### nsUniqueldGeneratorNamespace

この属性は、名前ベースの ID の生成方法を定義します。属性は、ID の生成に使用する名前空間を設定します。このオプションは、エントリーに同じ ID が必要な場合に同じ LDIF ファイルを 2 つの Directory Server インスタンスにインポートするのに便利です。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=import,cn=tasks,cn=config
有効な値	任意の文字列
デフォルト値	

パラメーター	説明
構文	大文字と小文字を区別しない文字列
例	nsUniquelGeneratorNamespace: example

### 3.1.16.3. cn=export

タスクのパラメーターを定義し、タスクを開始する特別なタスクエントリーを作成して、コマンドラインで1つまたは複数のデータベースをエクスポートできます。タスクが完了するとすぐに、タスクエントリーがディレクトリーから削除されます。

**cn=export,cn=tasks,cn=config** エントリーは、タスク操作をエクスポートするコンテナです。これらのタスクはこのコンテナ内に保存され、**cn=task\_name,cn=export,cn=tasks,cn=config** という名前が付けられます。

エクスポート操作の実行中に、タスクエントリーには、「[cn=tasks の下にあるエントリーのタスク呼び出し属性](#)」に記載されているサーバー生成タスク属性がすべて含まれます。

エクスポートタスクは手動で作成するか、**db2ldif.pl** コマンドを使用します。以下の表は、**db2ldif.pl** コマンドラインオプションとそれに対応する属性を示しています。

db2ldif.pl オプション	タスクの属性	説明
<b>-a</b>	<b>nsFilename</b>	エクスポートされた LDIF ファイルへのパスを設定します。
<b>-C</b>	<b>nsUsed2Entry</b>	有効な場合は、メインのデータベースファイルのみを使用します。
<b>-M</b>	<b>nsUseOneFile</b>	有効にすると、出力を複数のファイルに保存します。
<b>-n</b>	<b>nsInstance</b>	データベース名を設定します。
<b>-N</b>	<b>nsPrintKey</b>	シーケンス番号の出力を抑制できます。
<b>-r</b>	<b>nsExportReplica</b>	設定されている場合は、エクスポートにはレプリカを初期化する属性が含まれます。
<b>-s</b>	<b>nsIncludeSuffix</b>	エクスポートされたファイルに追加する接尾辞を設定します。
<b>-u</b>	<b>nsDumpUniqId</b>	一意の ID をエクスポートしないようにします。

db2ldif.pl オプション	タスクの属性	説明
<b>-U</b>	<b>nsNoWrap</b>	設定されている場合、長い行は折り返されません。
<b>-x</b>	<b>nsExcludeSuffix</b>	エクスポートされたファイルで除外する接尾辞を設定します。

### nsFilename

**nsFilename** 属性には、Directory Server インスタンスデータベースをエクスポートする LDIF ファイルのパスとファイル名が含まれます。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=export,cn=tasks,cn=config
有効な値	任意の文字列
デフォルト値	
構文	大文字と小文字を区別する文字列 (複数値)
例	nsFilename: /home/jsmith/example.ldif

### nsInstance

この属性は、**userRoot** や **userRoot** などのデータベースをエクスポートするデータベースインスタンスの名前を提供します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=export,cn=tasks,cn=config
有効な値	Directory Server インスタンスの名前 (任意の文字列)
デフォルト値	
構文	大文字と小文字を区別する文字列 (複数値)
例	nsInstance: userRoot

### nsIncludeSuffix

この属性は、LDIF ファイルにエクスポートする特定の接尾辞またはサブツリーを識別します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=export,cn=tasks,cn=config
有効な値	任意の DN
デフォルト値	
構文	DN、多値
例	nsIncludeSuffix: ou=people,dc=example,dc=com

### nsExcludeSuffix

この属性は、エクスポートした LDIF ファイルから除外するデータベースの接尾辞またはサブツリーを識別します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=export,cn=tasks,cn=config
有効な値	任意の DN
デフォルト値	
構文	DN、多値
例	nsExcludeSuffix: ou=machines,dc=example,dc=com

### nsUseOneFile

この属性は、すべての Directory Server インスタンスを単一の LDIF ファイルまたは個別の LDIF ファイルのどちらにエクスポートするかを設定します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=export,cn=tasks,cn=config
有効な値	true   false
デフォルト値	true
構文	大文字と小文字を区別しない文字列
例	nsUseOneFile: true

### nsExportReplica

この属性は、エクスポートされたデータベースがレプリケーションで使用されるかどうかを特定します。レプリカの場合に、レプリカを自動的に初期化するエントリーに、適切な属性と設定が含まれます。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=export,cn=tasks,cn=config
有効な値	true   false
デフォルト値	false
構文	大文字と小文字を区別しない文字列
例	nsExportReplica: true

### nsPrintKey

この属性は、エクスポートタスクでエントリーを処理する時にエントリー ID 番号を出力するかどうかを設定します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=export,cn=tasks,cn=config
有効な値	true   false
デフォルト値	true
構文	大文字と小文字を区別しない文字列
例	nsPrintKey: false

### nsUseld2Entry

**nsUseld2Entry** 属性は、メインのデータベースインデックス **id2entry** を使用してエクスポートされた LDIF エントリーを定義します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=export,cn=tasks,cn=config
有効な値	true   false
デフォルト値	false
構文	大文字と小文字を区別しない文字列

パラメーター	説明
例	nsUseId2Entry: true

### nsNoWrap

この属性は、LDIF ファイルで長い行を折り返すかどうかを設定します。

パラメーター	説明
エントリー DN	cn=task_name,cn=export,cn=tasks,cn=config
有効な値	true   false
デフォルト値	false
構文	大文字と小文字を区別しない文字列
例	nsNoWrap: false

### nsDumpUniqId

この属性は、エクスポートされたエントリーの一意の ID がエクスポートされないように設定します。

パラメーター	説明
エントリー DN	cn=task_name,cn=export,cn=tasks,cn=config
有効な値	true   false
デフォルト値	true
構文	大文字と小文字を区別しない文字列
例	nsDumpUniqId: true

#### 3.1.16.4. cn=backup

データベースをコマンドラインでバックアップするには、タスクのパラメーターを定義し、タスクを開始する特殊なタスクエントリーを作成します。タスクが完了するとすぐに、タスクエントリーがディレクトリーから削除されます。

**cn=backup** エントリーは、バックアップタスク操作のコンテナーエントリーです。**cn=backup** エントリー自体には属性はありませんが、**cn=task\_ID**、**cn=backup**、**cn=tasks**、**cn=config** など、このエントリー内にあるタスクエントリーごとに、以下の属性を使用してバックアップタスクを定義します。

**cn=backup** のバックアップタスクエントリーには、アーカイブコピーを取得するディレクトリーの場所 (**nsArchiveDir** 属性内) とバックアップするデータベースのタイプ (**nsDatabaseType** 属性) を含める必

必要があります。さらに、タスクを識別するために一意の **cn** を含める必要があります。以下に例を示します。

```
dn: cn=example backup,cn=backup,cn=tasks,cn=config
objectclass: extensibleObject
cn: example backup
nsArchiveDir: /export/backups/
nsDatabaseType: ldbm database
```

バックアップ操作が実行されると、タスクエントリーには、「**cn=tasks** の下にあるエントリーのタスク呼び出し属性」に記載のサーバー生成タスク属性がすべて含まれます。

### nsArchiveDir

この属性は、バックアップを書き込むディレクトリーの場所を指定します。

このバックアップディレクトリーは、通常 **nsslapd-bakdir** 属性で設定されたディレクトリーと同じである必要があります。

この属性が **cn=backup** タスクに含まれていないと、タスクは LDAP オブジェクトクラス違反エラー (65) で失敗します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=backup,cn=tasks,cn=config
有効な値	ローカルディレクトリーの場所
デフォルト値	
構文	大文字と小文字を区別する文字列
例	nsArchiveDir: /export/backups

### nsDatabaseType

この属性は、アーカイブされるデータベースの種類を指定します。データベースタイプの設定は、Directory Server がデータベースのアーカイブに使用するバックアッププラグインの種類を通知します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=backup,cn=tasks,cn=config
有効な値	ldbm データベース
デフォルト値	ldbm データベース
構文	大文字と小文字を区別する文字列

パラメーター	説明
例	nsDatabaseType: ldbm database

### 3.1.16.5. cn=restore

データベースをコマンドラインで復元するには、タスクのパラメーターを定義し、タスクを開始する特別なタスクエントリーを作成します。タスクが完了するとすぐに、タスクエントリーがディレクトリーから削除されます。

**cn=restore** エントリーは、データベースを復元するタスク操作のコンテナエントリーです。**cn=restore** エントリー自体には属性はありませんが、**cn=task\_ID**、**cn=restore**、**cn=tasks**、**cn=config** など、このエントリー内にあるタスクエントリーごとに、以下の属性を使用して復元タスクを定義します。

**cn=restore** の復元タスクエントリーには、アーカイブコピーを取得するディレクトリーの場所 (**nsArchiveDir** 属性内) と復元するデータベースのタイプ (**nsDatabaseType** 属性) を含める必要があります。さらに、タスクを識別するために一意の **cn** を含める必要があります。以下に例を示します。

```
dn: cn=example restore,cn=restore,cn=tasks,cn=config
objectclass: extensibleObject
cn: example restore
nsArchiveDir: /export/backups/
nsDatabaseType: ldbm database
```

復元操作が実行されると、タスクエントリーには、「**cn=tasks** の下にあるエントリーのタスク呼び出し属性」に記載のサーバー生成タスク属性がすべて含まれます。

### nsArchiveDir

この属性は、バックアップを書き込むディレクトリーの場所を指定します。

パラメーター	説明
エントリー DN	cn=task_name,cn=restore,cn=tasks,cn=config
有効な値	ローカルディレクトリーの場所
デフォルト値	
構文	大文字と小文字を区別する文字列
例	nsArchiveDir: /export/backups

### nsDatabaseType

この属性は、アーカイブされるデータベースの種類を指定します。データベースタイプの設定は、Directory Server がデータベースのアーカイブに使用するバックアッププラグインの種類を通知します。



パラメーター	説明
エントリー DN	cn=task_name,cn=restore,cn=tasks,cn=config
有効な値	ldbm データベース
デフォルト値	ldbm データベース
構文	大文字と小文字を区別する文字列
例	nsDatabaseType: ldbm database

### 3.1.16.6. cn=index

ディレクトリー属性は、タスクのパラメーターを定義してタスクを開始する特別なタスクエントリーを作成することで、コマンドラインでインデックスを作成できます。タスクが完了するとすぐに、タスクエントリーがディレクトリーから削除されます。

**cn=index** エントリーは、インデックスタスク操作のコンテナエントリーです。**cn=index** エントリー自体には属性はありませんが、**cn=task\_ID**、**cn=index**、**cn=tasks**、**cn=config** などのこのエントリー内にあるタスクエントリーごとに、以下の属性を使用してバックアップタスクを定義します。

**cn=index** の下のインデックスタスクエントリーは、**nsIndexAttribute** 属性に定義された、インデックスを作成する属性および作成するインデックスのタイプを識別することにより、標準のインデックスを作成できます。

または、インデックスタスクを使用して、**nsIndexVLVAttribute** 属性で、属性の仮想リストビュー (VLV) インデックスを生成できます。これは **vlvindex** スクリプトの実行と同じです。

以下に例を示します。

```
dn: cn=example presence index,cn=index,cn=tasks,cn=config
objectclass: top
objectclass: extensibleObject
cn: example presence index
nsInstance: userRoot
nsIndexAttribute: cn:pres

dn: cn=example VLV index,cn=index,cn=tasks,cn=config
objectclass: extensibleObject
cn: example VLV index
nsIndexVLVAttribute: "by MCC ou=people,dc=example,dc=com"
```

インデックス操作が実行されると、タスクエントリーには、「**cn=tasks** の下にあるエントリーのタスク呼び出し属性」に記載のサーバー生成タスク属性がすべて含まれます。

### nsIndexAttribute

この属性は、インデックスする属性の名前と、適用するインデックスのタイプを示します。属性値の形式は、属性名と、二重引用符で囲まれたインデックスタイプのコンマ区切りリストです。以下に例を示します。

```
nsIndexAttribute: attribute:index1,index2
```

パラメーター	説明
エントリー DN	cn=task_name,cn=index,cn=tasks,cn=config
有効な値	* すべての属性  * インデックスタイプ、 <b>pres</b> (presence)、 <b>eq</b> (equality)、 <b>approx</b> (approximate)、および <b>sub</b> (substring)
デフォルト値	
構文	大文字と小文字を区別しない文字列、多値
例	* nsIndexAttribute: cn:pres,eq  * nsIndexAttribute: description:sub

### nsIndexVLVAttribute

この属性は、VLV インデックスのターゲットエントリーの名前を指定します。仮想リストビューは、(Administration Guideで説明されているように) 参照インデックスエントリーに基づいており、仮想リストベース DN、スコープ、およびフィルターを定義します。**nsIndexVLVAttribute** 値は参照インデックスエントリーで、VLV 作成タスクは参照インデックスエントリーパラメーターに基づいて実行されます。

パラメーター	説明
エントリー DN	cn=task_name,cn=index,cn=tasks,cn=config
有効な値	VLV エントリー定義のサブエントリーの RDN
デフォルト値	
構文	DirectoryString
例	nsIndexVLVAttribute: "参照先のインデックスソート識別子"

### 3.1.16.7. cn=schema リロードタスク

ディレクトリースキーマは、ディレクトリーインスタンスの起動または再起動時に読み込まれます。カスタムスキーマ要素の追加を含むディレクトリースキーマへの変更は、サーバーが再起動するか、スキーマリロードタスクを開始するまで、自動的に読み込まれず、インスタンスで利用できます。

Directory Server インスタンスを再起動せずに、カスタムスキーマの変更を動的にリロードできます。これは、**cn=tasks** エントリーの下に新しいタスクエントリーを作成してスキーマの再読み込みタスクを開始することによって行われます。

カスタムスキーマファイルは任意のディレクトリーに配置できます。`schemadir` 属性で指定されていない場合には、サーバーはデフォルトの `/etc/dirsrv/slapped-instance/schema` ディレクトリーからスキーマを再読み込みします。



## 重要

別のディレクトリーからロードされたスキーマをスキーマディレクトリーにコピーする必要があります。そうしないと、サーバーでスキーマが失われます。

`schemd` リロードタスクは、タスクのパラメーターを定義してタスクを開始する特別なタスクエントリーを作成して、コマンドラインから開始されます。タスクが完了するとすぐに、タスクエントリーがディレクトリーから削除されます。以下に例を示します。

```
dn: cn=example schema reload,cn=schema reload task,cn=tasks,cn=config
objectclass: extensibleObject
cn:example schema reload
schemadir: /export/schema
```

`cn=schema reload task` エントリーは、スキーマリロード操作のコンテナエントリーです。`cn=schema` リロードタスク エントリー自体には属性はありませんが、`cn=task_ID`, `cn=schema reload task`, `cn=tasks`, `cn=config` など、このエントリー内にあるタスクエントリーごとに、スキーマリロード属性を使用して個別のリロードタスクを定義します。

## cn

`cn` 属性は、開始する新しいタスク操作を特定します。`cn` 属性の値は、新しいタスクを定義する限りすべて使用できます。

パラメーター	説明
エントリー DN	<code>cn=task_name,cn=schema reload task,cn=tasks,cn=config</code>
有効な値	任意の文字列
デフォルト値	
構文	DirectoryString
例	cn: リロードタスク ID の例

## schemadir

これには、カスタムスキーマファイルを含むディレクトリーへの完全パスが含まれます。

パラメーター	説明
エントリー DN	<code>cn=task_name,cn=schema reload task,cn=tasks,cn=config</code>
有効な値	任意のローカルディレクトリーパス。

パラメーター	説明
デフォルト値	/etc/dirsrv/schema
構文	DirectoryString
例	schemadir: /export/schema/

### 3.1.16.8. cn=memberof task

**memberof** 属性は、Directory Server で自動的に作成して管理され、メンバーのユーザーエントリーにグループメンバーシップを表示します。グループエントリーの **member** 属性を変更すると、すべてのメンバーに関連付けられたディレクトリーエントリーが、対応する **memberof** 属性で自動的に更新されます。

**cn=memberof task** (および関連する **fixup-memberof.pl** スクリプト) を使用して、ディレクトリー内のメンバーのユーザーエントリーに最初の **memberof** 属性を作成します。**memberof** 属性の作成後に、MemberOf プラグインは **memberof** 属性を自動的に管理します。

**memberof** 更新タスクには、更新タスクを実行するエントリーまたはサブツリーの DN (**basedn** 属性に設定) を指定する必要があります。必要に応じて、タスクにフィルターを追加して、更新するメンバーのユーザーエントリーを特定できます (**filter** 属性に設定)。以下に例を示します。

```
dn: cn=example memberOf,cn=memberof task,cn=tasks,cn=config
objectclass: extensibleObject
cn:example memberOf
basedn: ou=people,dc=example,dc=com
filter: (objectclass=groupofnames)
```

タスクが完了すると、タスクエントリーはディレクトリーから削除されます。

**cn=memberof task** エントリーは、**memberof** 更新操作のコンテナエントリーです。**cn=memberof task** エントリー自体には属性はありませんが、**cn=task\_ID**、**cn=memberof task**、**cn=tasks**、**cn=config** など、このエントリーのタスクエントリーごとに、その属性を使用して個別の更新タスクを定義します。

#### basedn

この属性は、**memberof** 属性を更新するユーザーエントリーの検索に使用するベース DN を指定します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=memberof task,cn=tasks,cn=config
有効な値	任意の DN
デフォルト値	
構文	DN

パラメーター	説明
例	basedn: ou=people,dc=example,dc=com

### filter

この属性は、**memberOf** 属性を更新するユーザーエントリーの選択に使用するオプションの LDAP フィルターを提供します。グループの各メンバーには、ディレクトリーに対応するユーザーエントリーがあります。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=memberof task,cn=tasks,cn=config
有効な値	任意の LDAP フィルター
デフォルト値	(objectclass=*)
構文	DirectoryString
例	filter: (!=Sunnyvale)

### 3.1.16.9. cn=fixup リンクされた属性

Directory Server にはリンク属性プラグインがあり、エントリーに設定されている属性で、自動的に別のエントリー内の別の属性を更新できます。どちらのエントリーにも、値の DN があります。最初のエントリーの DN 値は、更新するプラグインのエントリーを参照し、2 番目のエントリーの属性には、最初のエントリーへの DN バックポイントが含まれます。

これは、MemberOf プラグインがグループエントリーの **member** 属性を使用してユーザーエントリーの **memberOf** 属性を設定する方法と似ています。リンク属性を使用すると、すべての属性をリンクとして定義し、影響を受けるエントリーで別の属性が管理されます。

**cn=fixup linked attributes** (および関連する **fixup-linkedattrs.pl** スクリプト) は、リンクプラグインインスタンスが作成されると、データベースにすでに存在しているリンク属性に基づいて、管理属性を作成します。リンクおよび管理属性の設定後には、リンク属性プラグインは、ユーザーによりリンク属性が変更されると、マネージドの属性を動的に管理します。

リンク属性の更新タスクは、更新するリンク属性プラグインインスタンスはどれかを指定できます。これは **linkdn** 属性に設定されます (任意)。この属性がタスクエントリーに設定されていない場合は、設定されたリンク属性がすべて更新されます。

```
dn: cn=example,cn=fixup linked attributes,cn=tasks,cn=config
objectclass: extensibleObject
cn:example
linkdn: cn=Example Link,cn=Linked Attributes,cn=plugins,cn=config
```

タスクが完了すると、タスクエントリーはディレクトリーから削除されます。

**cn=fixup linked attributes** エントリーは、リンク属性更新操作のコンテナエントリーで

す。**cn=fixup** リンク属性エントリー自体には個別のタスクに関連する属性はありませんが、**cn=task\_ID**、**cn=fixup linked attributes**、**cn=tasks**、**cn=config** など、このエントリーの配下にあるタスクエントリーごとに属性を使用して個別の更新タスクを定義します。

## linkdn

リンク属性と管理属性のペアは、リンク属性プラグインインスタンスで設定されます。**linkdn** 属性は、プラグインインスタンス DN を指定して、エントリーの更新に使用される特定のリンク属性プラグインを設定します。以下に例を示します。

```
linkdn: cn=Manager Attributes,cn=Linked Attributes,cn=plugins,cn=config
```

プラグインインスタンスを指定しないと、リンクされているすべての属性が更新されます。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=fixup linked attributes,cn=tasks,cn=config
有効な値	DN(リンク属性プラグインのインスタンス)
デフォルト値	なし
構文	DN
例	linkdn: cn=Manager Links,cn=Linked Attributes,cn=plugins,cn=config

### 3.1.16.10. cn=syntax validate

構文の検証では、属性への変更をすべてチェックし、新しい値に、その属性タイプに必要な構文が含まれていることを確認します。属性構文は [RFC 4514](#) の定義に対して検証されます。

構文検証はデフォルトで有効になっています。ただし、構文の検証では、属性の追加や変更時など、属性値への変更のみが監査されます。**既存** の属性値の構文は検証されません。

既存の構文の検証は、構文検証タスクを使用して実行できます。このタスクは、(**basedn** 属性内の) 指定のサブツリーでエントリーをチェックし、任意で、指定されたフィルターに一致するエントリー (**filter** 属性) のみを確認します。

```
dn: cn=example,cn=syntax validate,cn=tasks,cn=config
objectclass: extensibleObject
cn:example
basedn: ou=people,dc=example,dc=com
filter: "(objectclass=inetorgperson)"
```

タスクが完了すると、タスクエントリーはディレクトリーから削除されます。

構文検証が無効であるか、サーバーを移行する場合は、属性構文の要件に準拠しないサーバーにデータが存在する可能性があります。構文検証タスクを実行して、構文の検証を有効にする前に既存の属性値を評価できます。

**cn=syntax validate** エントリーは、構文検証操作のコンテナエントリーです。**cn=syntax validate**

エントリー自体には、タスクに固有の属性はありません。**cn=task\_ID**, **cn=syntax validate**, **cn=tasks**, **cn=config** など、このエントリーにあるタスクエントリーごとに、その属性を使用して個別の更新タスクを定義します。

### basedn

構文検証タスクを実行するサブツリーを指定します。以下に例を示します。

```
basedn: ou=people,dc=example,dc=com
```

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=syntax validate,cn=tasks,cn=config
有効な値	任意の DN
デフォルト値	なし
構文	DN
例	basedn: dc=example,dc=com

### filter

構文検証タスクを実行する指定の **basedn** の下の特定のエントリーを識別するために使用できるオプションの LDAP フィルターが含まれます。この属性がタスクに設定されていない場合は、**basedn** 内のすべてのエントリーが監査されます。以下に例を示します。

```
filter: "(objectclass=person)"
```

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=syntax validate,cn=tasks,cn=config
有効な値	任意の LDAP フィルター
デフォルト値	"(objectclass=*)"
構文	DirectoryString
例	filter: "(objectclass=*)"

#### 3.1.16.11. cn=USN 破棄クリーンアップタスク

USN プラグインが有効な場合は、追加または変更などのディレクトリーの書き込み操作がそのエントリーで発生するたびに、すべてのエントリーで **シーケンス番号 (USN)** が設定されます。これは **entryUSN** 操作属性に反映されます。この USN は、エントリーが削除され、tombstone エントリーは

Directory Server インスタンスによって管理されます。

**cn=USN tombstone cleanup task** (および関連の **usn-tombstone-cleanup.pl** スクリプト) は、バックエンドデータベース (**バックエンド** 属性) または接尾辞 (**接尾辞** 属性) に従って、インスタンスが維持する tombstone エントリーを削除します。必要に応じて、削除する USN の最大値 (**max\_usn\_to\_delete** 属性) を指定して tombstone エントリーのサブセットのみを削除できるので、最新の tombstone エントリーを保持します。

```
dn: cn=example,cn=USN tombstone cleanup task,cn=tasks,cn=config
objectclass: extensibleObject
cn:example
backend: userroot
max_usn_to_delete: 500
```

## 重要

このタスクは、レプリケーションが有効で **ない** 場合にのみ起動できます。レプリケーションは独自の tombstone ストアを維持し、これらの tombstone エントリーは USN プラグインで削除できないので、レプリケーションプロセスで維持する必要があります。したがって、Directory Server は、レプリケートされたデータベースのクリーンアップタスクを実行できないようにします。

レプリケートされたバックエンドに対して、このタスクエントリーを作成しようとすると、以下のエラーがコマンドラインで返されます。

```
ldap_add: DSA is unwilling to perform
```

エラーログには、接尾辞にレプリケートされているため tombstone を削除できないという明示的なメッセージが追加されます。

```
[...] usn-plugin - Suffix dc=example,dc=com is replicated. Unwilling to perform cleaning up tombstones.
```

タスクが完了すると、タスクエントリーはディレクトリーから削除されます。

**cn=USN tombstone cleanup task** エントリーは、すべての USN tombstone delete 操作のコンテナエントリーです。**cn=USN tombstone cleanup** タスク エントリー自体には個別のタスクに関連する属性はありませんが、**cn=task\_ID**、**cn=USN tombstone cleanup task**、**cn=tasks**、**cn=config** など、このエントリーの配下にあるタスクごとに、属性を使用して個別の更新タスクを定義します。

## バックエンド

これにより、Directory Server インスタンスのバックエンドまたはデータベースにクリーンアップ操作を実行できます。バックエンドが指定されていない場合は、接尾辞を指定する必要があります。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=USN tombstone cleanup task,cn=tasks,cn=config
有効な値	データベース名



パラメーター	説明
デフォルト値	なし
構文	DirectoryString
例	backend: userroot

### max\_usn\_to\_delete

これにより、tombstone エントリーを削除する際に削除される USN の最大値が指定されます。この数字を含む tombstone エントリーはすべて削除されます。USN 値が大きい tombstone エントリー (つまり、最新のエントリー) は削除されません。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=USN tombstone cleanup task,cn=tasks,cn=config
有効な値	任意の整数
デフォルト値	なし
構文	整数
例	max_usn_to_delete: 500

### 接尾辞

これにより、Directory Server の接尾辞またはサブツリーに、クリーンアップ操作を実行できます。接尾辞が指定されていない場合は、バックエンドを指定する必要があります。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=USN tombstone cleanup task,cn=tasks,cn=config
有効な値	サブツリー DN
デフォルト値	なし
構文	DN
例	suffix: dc=example,dc=com

#### 3.1.16.12. cn=cleanallruv

レプリケーショントポロジに関する情報、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で加えた最新の変更状態番号、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、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エラーをログに記録します。

```
[09/Sep/2020:09:03:43 -0600] NSMMReplicationPlugin - ruv_compare_ruv: RUV [changelog max
RUV] does not
contain element [{"replica 55 ldap://server.example.com:389} 4e6a27ca000000370000
4e6a27e8000000370000]
which is present in RUV [database RUV]
.....
[09/Sep/2020:09:03:43 -0600] 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 CLEANRUV
task. If they
are not obsolete, you should check their status to see why there are no changes from those servers
in the changelog.
```

サプライヤーがトポロジから永久に削除されると、そのサプ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が、

**cn=cleanallruv** タスクは、レプリケーショントポロジ内のすべてのサーバーを介して伝播し、欠落しているサプライヤーや、古くなったサプライヤーに関連付けられた、指定の RUV エントリーを削除します。

タスクが完了すると、タスクエントリーはディレクトリーから削除されます。

**cn=cleanallruv** エントリーは、すべてのクリーン RUV 操作のコンテナーエントリーです。**cn=cleanallruv** エントリー自体には個別のタスクに関連する属性はありませんが、**cn=task\_ID**、**cn=cleanallruv**、**cn=tasks**、**cn=config** など、このエントリーの配下にあるタスクごとに、属性を使用して個別の更新タスクを定義します。

各クリーン RUV タスクは、削除するレプリカ RUV エントリーのレプリカ ID 番号、レプリケートされたデータベースのベース DN、および RUV データを削除する前に、欠落しているサプライヤーから残りの更新を適用する必要があるかどうかを指定する必要があります。

```
dn: cn=clean 55,cn=cleanallruv,cn=tasks,cn=config
objectclass: extensibleObject
replica-base-dn: dc=example,dc=com
replica-id: 55
replica-force-cleaning: no
cn: clean 55
```

### replica-base-dn

これにより、複製されたデータベースに関連付けられた Directory Server ベース DN が提供されます。これは、複製された接尾辞のベース DN です。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=cleanallruv,cn=tasks,cn=config
有効な値	ディレクトリーの接尾辞 DN
デフォルト値	なし
構文	DirectoryString
例	replica-base-dn: dc=example,dc=com

### replica-id

これにより、レプリカトポロジーから **削除される** レプリカのレプリカ ID(レプリカ設定エントリーの **nsDS5ReplicaId** 属性で定義されている) が指定されます。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=cleanallruv,cn=tasks,cn=config
有効な値	0 から 65534
デフォルト値	なし
構文	整数
例	replica-id: 55

### replica-force-cleaning

これにより、削除するレプリカから未処理の更新を適用する (**no**) か、clean RUV 操作を強制的に実行し、残りの更新を破棄する (**yes**) かどうかを設定します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=cleanallruv,cn=tasks,cn=config
有効な値	no   yes
デフォルト値	なし
構文	DirectoryString
例	replica-force-cleaning: no

### 3.1.16.13. cn=abort cleanallruv

「`cn=cleanallruv`」タスクの所要時間として、このタスクですべての更新を初めて処理する場合には、レプリケーショントポロジー内のすべてのサーバー間で伝播するのに数分かかる場合があります。パフォーマンスやその他のメンテナンスに関する考慮事項は、clean RUV タスクを終了し、その終了もレプリケーショントポロジー内のすべてのサーバーに伝播されます。

終了タスクは、`cn=abort cleanallruv` エントリーのインスタンスです。

タスクが完了すると、タスクエントリーはディレクトリーから削除されます。

`cn=abort cleanallruv` エントリーは、すべてのクリーン RUV 操作のコンテナエントリーです。`cn=abort cleanallruv` エントリー自体には個別のタスクに関連する属性はありませんが、`cn=task_ID`、`cn=abort cleanallruv`、`cn=tasks`、`cn=config` など、このエントリーの配下にあるタスクごとに、属性を使用して個別の更新タスクを定義します。

各クリーン RUV タスクは、**現在削除されている** へのレプリカ RUV エントリーのレプリカ ID 番号、レプリケートされたデータベースのベース DN、およびターミネイトタスクがトポロジー内のすべてのサーバーで完了したときに完了するか、ローカルのみで完了するかを指定する必要があります。

```
dn: cn=abort 55,cn=abort cleanallruv,cn=tasks,cn=config
objectclass: extensibleObject
replica-base-dn: dc=example,dc=com
replica-id: 55
replica-certify-all: yes
cn: abort 55
```

#### replica-base-dn

これにより、複製されたデータベースに関連付けられた Directory Server ベース DN が提供されます。これは、複製された接尾辞のベース DN です。

パラメーター	説明
エントリー DN	<code>cn=task_name,cn=abort cleanallruv,cn=tasks,cn=config</code>
有効な値	ディレクトリーの接尾辞 DN
デフォルト値	なし
構文	DirectoryString
例	<code>replica-base-dn: dc=example,dc=com</code>

#### replica-id

これにより、レプリカトポロジーから **削除中の** レプリカのレプリカ ID (レプリカ設定エントリーの `nsDS5ReplicaId` 属性で定義されている) が指定されます。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=abort cleanallruv,cn=tasks,cn=config
有効な値	0 から 65534
デフォルト値	なし
構文	整数
例	replica-id: 55

### replica-certify-all

これにより、タスクをローカルで完了する前にレプリケーショントポロジ内の**すべての**サーバーでタスクを正常に完了するか (**yes**)、ローカルで完了するとすぐにタスクを完了として表示するか (**no**) を設定します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=abort cleanallruv,cn=tasks,cn=config
有効な値	no   yes
デフォルト値	なし
構文	DirectoryString
例	replica-certify-all: yes

#### 3.1.16.14. cn=automember rebuild membership

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た場合にのみ実行されます。このプラグインは、automembership ルールに一致するように編集された既存のエントリーを無視します。

**cn=automember rebuild membership** タスクは、**既存** のエントリーに対して現在の automembership ルールを実行して、グループメンバーシップを更新または再構築します。設定された automembership ルールはすべて、特定されたエントリーに対して実行されます (すべてのルールが特定のエントリーに適用されるわけではありません)。

#### basedn

これにより、ユーザーエントリーの検索に使用する Directory Server ベース DN が提供されます。その後、指定した DN のエントリーが automembership ルールに従って更新されます。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=automember rebuild membership,cn=tasks,cn=config
有効な値	ディレクトリーの接尾辞 DN
デフォルト値	なし
構文	DirectoryString
例	basedn: dc=example,dc=com

### filter

この属性は、設定された automembership ルールに従って更新するユーザーエントリーの特定に使用する LDAP フィルターを指定します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=automember rebuild membership,cn=tasks,cn=config
有効な値	任意の LDAP フィルター
デフォルト値	なし
構文	DirectoryString
例	filter: (uid=*)

### scope

この属性は、指定のベース DN の検索時に使用する LDAP 検索範囲を指定します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=automember rebuild membership,cn=tasks,cn=config
有効な値	sub   base   one
デフォルト値	なし
構文	DirectoryString
例	scope: sub

### 3.1.16.15. cn=automember export updates

このタスクが、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。

automembership 関連の変更は実行 **されません**。提案された変更は、指定された LDIF ファイルに書き込まれます。

#### basedn

これにより、ユーザーエントリーの検索に使用する Directory Server ベース DN が提供されます。automembership ルールのテスト実行は、特定されたエントリーに対して実行されます。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=automember export updates,cn=tasks,cn=config
有効な値	ディレクトリーの接尾辞 DN
デフォルト値	なし
構文	DirectoryString
例	basedn: dc=example,dc=com

#### filter

この属性は、automembership ルールをテストするユーザーエントリーの識別に使用する LDAP フィルターを指定します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=automember export updates,cn=tasks,cn=config
有効な値	任意の LDAP フィルター
デフォルト値	なし
構文	DirectoryString
例	filter: (uid=*)

#### scope

この属性は、指定のベース DN の検索時に使用する LDAP 検索範囲を指定します。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=automember export updates,cn=tasks,cn=config
有効な値	sub   base   one
デフォルト値	なし
構文	DirectoryString
例	scope: sub

## ldif

この属性は、automembership ルールの test-run から提案された変更を書き込む LDIF ファイルの完全パスおよびファイル名を設定します。このファイルは、タスクが開始するシステムのローカルである必要があります。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=automember export updates,cn=tasks,cn=config
有効な値	ローカルパスおよびファイル名
デフォルト値	なし
構文	DirectoryString
例	ldif: /tmp/automember-results.ldif

### 3.1.16.16. cn=automember map updates

このタスクは、LDIF ファイル内 (新しいエントリー、または場合によってはテストエントリー内) でエントリーに対して実行され、提案された変更を LDIF ファイルに書き込みます。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。

automembership 関連の変更は実行 **されません**。提案された変更は、指定された LDIF ファイルに書き込まれます。

## ldif\_in

この属性は、設定された automembership ルールでテストするエントリーのインポート元の LDIF ファイルの完全パスおよびファイル名を設定します。これらのエントリーはディレクトリーにインポートされず、変更は実行されません。エントリーは、テスト実行でのみ読み込まれ、使用されます。

このファイルは、タスクが開始するシステムのローカルである必要があります。



パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=automember map updates,cn=tasks,cn=config
有効な値	ローカルパスおよびファイル名
デフォルト値	なし
構文	DirectoryString
例	ldif_in: /tmp/automember-test-users.ldif

### ldif\_out

この属性は、automembership ルールの test-run から提案された変更を書き込む LDIF ファイルの完全パスおよびファイル名を設定します。このファイルは、タスクが開始するシステムのローカルである必要があります。

パラメーター	説明
エントリー DN	cn= <b>task_name</b> ,cn=automember map updates,cn=tasks,cn=config
有効な値	ローカルパスおよびファイル名
デフォルト値	なし
構文	DirectoryString
例	ldif_out: /tmp/automember-results.ldif

### 3.1.16.17. cn=des2aes

このタスクは、古い **DES** 暗号を使用してエンコードされた、指定のユーザーデータベースにある可逆パスワードエントリーをすべて検索し、それらをより安全な **AES** 暗号に変換します。

以前は、このタスクは、Directory Server の起動中にすべての接尾辞に対して自動的に実行されていました。ただし、DES パスワードの検索は通常、インデックスが作成されていないため、大量のエントリーを含む接尾辞に対して実行するのに非常に時間がかかる可能性があり、その結果、Directory Server がタイムアウトして起動に失敗しました。このため、検索は **cn=config** でのみ実行されますが、他のデータベースでパスワードを変換するには、このタスクを手動で実行する必要があります。

#### 接尾辞

この複数値属性は、DES パスワードを確認し、AES に変換する接尾辞を指定します。この属性を省略すると、すべてのバックエンド/接尾辞がチェックされます。

パラメーター	説明
エントリー DN	cn=task_name,cn=des2aes,cn=tasks,cn=config
有効な値	ディレクトリーの接尾辞 DN
デフォルト値	なし
構文	DirectoryString
例	suffix: dc=example,dc=com

### 3.1.17. cn=uniqueid ジェネレーター

一意の ID ジェネレーター設定属性は、**cn=uniqueid generator,cn=config** の下に保管されます。**cn=uniqueid generator** エントリーは、**extensibleObject** オブジェクトクラスのインスタンスです。

#### nsstate

この属性は、サーバーの再起動後も一意の ID ジェネレーターの状態を保存します。この属性はサーバーによって維持されます。これは編集しないでください。

パラメーター	説明
エントリー DN	cn=uniqueid generator,cn=config
有効な値	
デフォルト値	
構文	DirectoryString
例	nsstate: Abld0c3oMIDUntiLCyYNGgAAAAAAAAAA

### 3.1.18. Root DSE 設定パラメーター

#### 3.1.18.1. nsslapd-return-default-opattr

Directory Server では、Root DSE 検索で操作属性は表示されません。たとえば、**-s base -b ""** パラメーターを指定して **ldapsearch** ユーティリティを実行している場合には、ユーザー属性のみが表示されます。Root DSE 検索出力で動作属性を想定しているクライアントの場合は、この動作を有効にして後方互換性を確保できます。

1. Directory Server インスタンスを停止します。
2. `/etc/dirsrv/slapped-instance_name/dse.ldif` ファイルを編集し、以下のパラメーターを **dn:** セクションに追加します。

```

nsslapd-return-default-opattr: supportedsaslm mechanisms
nsslapd-return-default-opattr: nsBackendSuffix
nsslapd-return-default-opattr: subschemasubentry
nsslapd-return-default-opattr: supportedldapversion
nsslapd-return-default-opattr: supportedcontrol
nsslapd-return-default-opattr: ref
nsslapd-return-default-opattr: vendorname
nsslapd-return-default-opattr: vendorVersion
nsslapd-return-default-opattr: supportedextension
nsslapd-return-default-opattr: namingcontexts

```

3. Directory Server インスタンスを開始します。

パラメーター	説明
エントリー DN	Root DSE
有効な値	supportedsaslm mechanisms   nsBackendSuffix   subschemasubentry   supportedldapversion   supportedcontrol   ref   vendorname   vendorVersion
デフォルト値	
構文	DirectoryString
例	nsslapd-return-default-opattr: supportedsaslm mechanisms

## 3.2. 設定オブジェクトクラス

多くの設定エントリーは **extensibleObject** オブジェクトクラスのみを使用しますが、設定エントリーで他のオブジェクトクラスが必要になるものもあります。これらの設定オブジェクトクラスがここにリスト表示されます。

### 3.2.1. changeLogEntry (オブジェクトクラス)

このオブジェクトクラスは、Directory Server エントリーへの変更を保存するエントリーに使用されません。

Directory Server 4.1x に実装された changelog との互換性を維持するように Directory Server を設定するには、Retro Changelog プラグインを有効にします。changelog の各エントリーには、**changeLogEntry** オブジェクトクラスがあります。

このオブジェクトクラスは、Changelog インターネットドラフトで定義されています。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.1

表3.8 必要な属性

objectClass	エントリーのオブジェクトクラスを定義します。
「changeNumber」	changelog に任意に割り当てられた数字が含まれません。
「changeTime」	変更が行われた時刻。
「changeType」	エントリーに対して実行される変更のタイプ。
「targetDn」	サプライヤーサーバーで追加、変更、または削除されているエントリーの識別名。

表3.9 使用できる属性

「changes」	Directory Server に対する変更。
「deleteOldRdn」	エントリーの古い Relative Distinguished Name (RDN) をエントリーの識別属性として保持するか、削除するかを定義するフラグ。
「newRdn」	modRDN または modDN 操作の対象となるエントリーの新しい RDN。
「newSuperior」	modDN 操作の処理時に既存のエントリーの1つ上の階層となるエントリーの名前。

### 3.2.2. directoryServerFeature (オブジェクトクラス)

このオブジェクトクラスは、ディレクトリーサービスの機能を識別するエントリー専用として使用されます。このオブジェクトクラスは Directory Server で定義されます。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.40

表3.10 必要な属性

属性	定義
objectClass	エントリーに割り当てられたオブジェクトクラスを指定します。

表3.11 使用できる属性

属性	定義
cn	エントリーの一般的な名前を指定します。
multiLineDescription	エントリーのテキスト説明を入力します。
oid	機能の OID を指定します。

### 3.2.3. nsBackendInstance (オブジェクトクラス)

このオブジェクトクラスは、Directory Server バックエンドまたはデータベース、インスタンスエントリーに使用されます。このオブジェクトクラスは Directory Server に定義されています。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.109

表3.12 必要な属性

属性	定義
objectClass	エントリーのオブジェクトクラスを定義します。
cn	エントリーの共通名を指定します。

### 3.2.4. nsChangelog4Config (オブジェクトクラス)

Directory Server 11.3 を Directory Server 4.x サーバー間で複製するには、Directory Server 11.3 インスタンスに特別な changelog を設定する必要があります。このオブジェクトクラスは retro changelog の設定を定義します。

このオブジェクトは Directory Server に対して定義されます。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.82

表3.13 使用できる属性

属性	定義
cn (共通名)	エントリーの共通名を指定します。

### 3.2.5. nsDS5Replica (オブジェクトクラス)

このオブジェクトクラスは、データベースレプリケーションにレプリカを定義するエントリー用です。これらの属性の多くはバックエンド内で設定され、変更することはできません。

このオブジェクトクラスの属性に関する情報は、**Directory & Server 設定、コマンド、およびファイルリファレンス** の 2 章のコア設定属性と共にリスト表示されます。

このオブジェクトクラスは Directory Server に定義されています。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.108

表3.14 必要な属性

objectClass	エントリーのオブジェクトクラスを定義します。
nsDS5ReplicaId	レプリケーション環境内のサプライヤーの一意の ID を指定します。
nsDS5ReplicaRoot	複製された領域のルートで接尾辞 DN を指定します。

表3.15 使用できる属性

cn	レプリカの名前を指定します。
nsDS5Flags	フラグで以前に設定された情報を指定します。
nsDS5ReplicaAutoReferral	サーバーが Directory Server データベースに設定された参照に従うかどうかを設定します。
nsDS5ReplicaBindDN	サプライヤーサーバーがコンシューマーにバインドする時に使用する DN を指定します。
nsDS5ReplicaChangeCount	changelog の合計エントリー数と、複製されたかどうかを示します。
nsDS5ReplicaLegacyConsumer	レプリカがレガシーコンシューマーであるかどうかを指定します。
nsDS5ReplicaName	内部操作のレプリカの一意 ID を指定します。
nsDS5ReplicaPurgeDelay	changelog がページされるまでの秒数を指定します。
nsDS5ReplicaReferral	ユーザー定義の参照の URL を指定します。

nsDS5ReplicaReleaseTimeout	更新の送信が完了したかどうかに関係なく、サプライヤーがレプリカをリリースするまでのタイムアウトを指定します。
nsDS5ReplicaTombstonePurgeInterval	ページ操作サイクルの間隔 (秒単位) を指定します。
nsDS5ReplicaType	読み取り専用コンシューマーなどのレプリカのタイプを定義します。
nsDS5Task	データベースのコンテンツを LDIF にダンプするなどのレプリケーションタスクを起動します。これは、Directory Server サプライヤーにより内部で使用されます。
nsState	適切な変更シーケンス番号が生成されるように、クロックに関する情報を格納します。

### 3.2.6. nsDS5ReplicationAgreement (オブジェクトクラス)

オブジェクトクラスが **nsDS5ReplicationAgreement** のエントリーは、設定された情報をレプリカ合意に保存します。このオブジェクトクラスの属性に関する情報は、**Directory Server 設定、コマンド、およびファイルリファレンス**の2章を参照してください。

このオブジェクトクラスは Directory Server に定義されています。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.103

表3.16 必要な属性

objectClass	エントリーのオブジェクトクラスを定義します。
cn	レプリカ合意の命名に使用されます。

表3.17 使用できる属性

description	フリーテキストでのレプリカ合意に関する説明が含まれます。
nsDS5BeginReplicaRefresh	レプリカを手動で初期化します。
nsds5debugreplicatimeout	レプリケーションがデバッグロギングで実行される場合に使用する別のタイムアウト期間を指定します。

nsDS5ReplicaBindDN	サプライヤーサーバーがコンシューマーにバインドする時に使用する DN を指定します。
nsDS5ReplicaBindMethod	バインディングに使用するメソッド (SSL または簡易認証) を指定します。
nsDS5ReplicaBusyWaitTime	コンシューマーがビジー応答を送信してから、コンシューマーがアクセスの取得を試みるまで、サプライヤーが待機する時間を秒単位で設定します。
nsDS5ReplicaChangesSentSinceStartup	サーバーが起動してからこのレプリカに送信された変更の数。
nsDS5ReplicaCredentials	バインド DN のパスワードを指定します。
nsDS5ReplicaHost	コンシューマーレプリカのホスト名を指定します。
nsDS5ReplicaLastInitEnd	コンシューマーレプリカの初期化がいつ終了したかを示します。
nsDS5ReplicaLastInitStart	コンシューマーレプリカの初期化を開始するタイミングを示します。
nsDS5ReplicaLastInitStatus	コンシューマーの初期化のステータス。
nsDS5ReplicaLastUpdateEnd	最新のレプリケーションスケジュールの更新がいつ終了したかを示します。
nsDS5ReplicaLastUpdateStart	直近のレプリケーションスケジュールの更新が開始されるタイミングを示します。
nsDS5ReplicaLastUpdateStatus	最新のレプリケーションスケジュール更新のステータスを示します。
nsDS5ReplicaPort	リモートレプリカのポート番号を指定します。
nsDS5ReplicaRoot	複製された領域のルートで接尾辞 DN を指定します。
nsDS5ReplicaSessionPauseTime	次の更新セッションまでの間に、サプライヤーが待機する時間を秒単位で指定します。
nsDS5ReplicatedAttributeList	コンシューマーサーバーにレプリケートしない属性を指定します。
nsDS5ReplicaTimeout	タイムアウトや失敗するまでの間に、アウトバウンド LDAP 操作がリモートレプリカからの応答を待機する時間を秒単位で指定します。



nsDS5ReplicaTransportInfo	レプリカと間のデータ転送に使用されるポートのタイプを指定します。
nsDS5ReplicaUpdateInProgress	レプリケーションスケジュールの更新が進行中であるかどうかを示します。
nsDS5ReplicaUpdateSchedule	レプリケーションスケジュールを指定します。
nsDS50ruv	レプリケーション更新ベクトルを使用してレプリカの内部状態を管理します。
nsruvReplicaLastModified	レプリカのエントリーが変更され、changelog が更新された最新の時間が含まれます。
nsds5ReplicaStripAttrs	一部のレプリケーションでは、除外した属性の更新も引き続きレプリケーションイベントをトリガーしますが、そのレプリケーションイベントは空となります。この属性は、レプリケーション更新から削除する属性を設定します。これにより、 <b>internalModifyTimestamp</b> など属性への変更で、空のレプリケーションの更新をトリガーできなくなります。

### 3.2.7. nsDSWindowsReplicationAgreement (オブジェクトクラス)

同期合意に関連する同期属性を保存します。このオブジェクトクラスの属性に関する情報は、**Red Hat Directory Server 設定、コマンド、およびファイルリファレンス**の2章を参照してください。

このオブジェクトクラスは Directory Server に定義されています。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.503

表3.18 必要な属性

objectClass	エントリーのオブジェクトクラスを定義します。
cn	同期合意の名前を指定します。

表3.19 使用できる属性

description	同期合意のテキストの説明が含まれます。
nsDS5BeginReplicaRefresh	手動同期を開始します。

nsds5debugreplicatimeout	同期がデバッグロギングで実行される場合に使用する別のタイムアウト期間を指定します。
nsDS5ReplicaBindDN	Directory Server が Windows サーバーへのバインドに使用する DN を指定します。
nsDS5ReplicaBindMethod	バインディングに使用するメソッド (SSL または簡易認証) を指定します。
nsDS5ReplicaBusyWaitTime	Windows サーバーがビジー応答を送信してから、別のアクセスの取得を試みるまでに、Directory Server が待機する時間を秒単位で指定します。
nsDS5ReplicaChangesSentSinceStartup	Directory Server の起動後に送信された変更の数を示します。
nsDS5ReplicaCredentials	バインド DN の認証情報を指定します。
nsDS5ReplicaHost	同期する Windows サーバーの Windows ドメインコントローラーのホスト名を指定します。
nsDS5ReplicaLastInitEnd	Windows サーバーの最後の完全な更新 (再同期) がいつ終了したかを示します。
nsDS5ReplicaLastInitStart	Windows サーバーの最後の完全な更新 (再同期) がいつ開始したかを示します。
nsDS5ReplicaLastInitStatus	Windows サーバーの完全な更新 (再同期) のステータス。
nsDS5ReplicaLastUpdateEnd	最新の更新がいつ終了したかを示します。
nsDS5ReplicaLastUpdateStart	最新の更新が開始された日時を示します。
nsDS5ReplicaLastUpdateStatus	最新の更新ステータスを示します。
nsDS5ReplicaPort	Windows サーバーのポート番号を指定します。
nsDS5ReplicaRoot	Directory Server のルート接尾辞 DN を指定します。
nsDS5ReplicaSessionPauseTime	次の更新セッションまでの間に、Directory Server が待機する時間を秒単位で指定します。
nsDS5ReplicaTimeout	タイムアウトや失敗するまでの間に、アウトバウンド LDAP 操作が Windows サーバーからの応答を待機する時間を秒単位で指定します。

nsDS5ReplicaTransportInfo	Windows サーバーとの間のデータ転送に使用されるトランスポートのタイプを指定します。
nsDS5ReplicaUpdateInProgress	更新が進行中であるかどうかを示します。
nsDS5ReplicaUpdateSchedule	同期スケジュールの適用
nsDS50ruv	レプリケーション更新ベクトル (RUV) を使用して、Directory Server 同期ピアの内部状態を管理します。
nsds7DirectoryReplicaSubtree	同期する Directory Server の接尾辞 (root または sub) を指定します。
nsds7DirsyncCookie	RUV として機能する同期サービスで設定される Cookie が含まれます。
nsds7NewWinGroupSyncEnabled	Directory Server で新しい Windows グループアカウントが自動的に作成されるかどうかを指定します。
nsds7NewWinUserSyncEnabled	Directory Server で新しい Windows ユーザーアカウントを自動的に作成するかどうかを指定します。
nsds7WindowsDomain	同期している Windows ドメインを特定します。レプリカ合意の <b>nsDS5ReplicaHost</b> と類似していません。
nsds7WindowsReplicaSubtree	同期する Windows サーバーの接尾辞 (root または sub) を指定します。
nsruvReplicaLastModified	Directory Server 同期ピアのエントリーが変更され、changelog が更新された最新の時間が含まれます。
winSyncInterval	Directory Server が Windows サーバーをポーリングして更新を行う頻度を秒単位で設定します。これが設定されていない場合に、デフォルトは <b>300</b> (300 秒または 5 分) になります。
winSyncMoveAction	同期されたサブツリー以外の Active Directory で検出された、対応のエントリーを同期プラグインが処理する方法を設定します。同期プロセスは、これらのエントリーを無視するか (none、デフォルト)、エントリーが意図的に移動されて同期から削除されたと思なし、対応する Directory Server エントリーまたは Synchronization 属性を削除するか (delete)、エントリーの同期を中止します (unsync)。

### 3.2.8. nsEncryptionConfig

**nsEncryptionConfig** オブジェクトクラスは、プロトコルや暗号スイートなど、使用可能な暗号化オプションの設定情報を保存します。これは、Administrative Services で定義されます。

## 上級クラス

top

## OID

nsEncryptionConfig-oid

表3.20 必要な属性

属性	定義
objectClass	エントリーのオブジェクトクラスを定義します。
cn (commonName)	デバイスの一般名を指定します。

表3.21 使用できる属性

属性	定義
nsSSL3SessionTimeout	SSLv3 暗号セッションのタイムアウト期間を設定します。
nsSSLClientAuth	サーバーがクライアント認証を処理する方法を設定します。allow、disallowed、または require の3つの値を使用できます。
nsSSLSessionTimeout	暗号セッションのタイムアウト期間を設定します。
nsSSLSupportedCiphers	サーバーへのセキュアな接続で使用可能な暗号のリストがすべて含まれます。
nsTLS1	サーバーで TLS バージョン1を有効にするかどうかを設定します。

### 3.2.9. nsEncryptionModule

**nsEncryptionModule** オブジェクトクラスは、暗号化モジュール情報を保存します。これは、Administrative Services で定義されます。

## 上級クラス

top

## OID

nsEncryptionModule-oid

表3.22 必要な属性

属性	定義
objectClass	エントリーのオブジェクトクラスを定義します。
cn (commonName)	デバイスの一般名を指定します。

表3.23 使用できる属性

属性	定義
nsSSLActivation	暗号ファミリーを有効にするかどうかを設定します。
nsSSLPersonalitySSL	SSL のサーバーが使用する証明書の名前が含まれます。
nsSSLToken	サーバーが使用するセキュリティトークンを特定します。

### 3.2.10. nsMappingTree(オブジェクトクラス)

マッピングツリーは、接尾辞をバックエンドにマップします。各マッピングツリーエントリーは **nsMappingTree** オブジェクトクラスを使用します。このオブジェクトクラスは Directory Server に定義されています。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.110

表3.24 必要な属性

属性	定義
objectClass	エントリーに割り当てられたオブジェクトクラスを指定します。
cn	エントリーの共通名を指定します。

### 3.2.11. nsSaslMapping (オブジェクトクラス)

このオブジェクトクラスは、SASL 属性を DirectoryServer 属性にマッピングするための ID マッピング設定を含むエントリーに使用されます。

このオブジェクトクラスは Directory Server に定義されています。

#### 上級クラス

top

**OID**

2.16.840.1.113730.3.2.317

表3.25 必要な属性

objectClass	エントリーのオブジェクトクラスを定義します。
cn	SASL マッピングエントリーの名前を指定します。
「nsSaslMapBaseDNTemplate」	検索ベースの DN テンプレートが含まれます。
「nsSaslMapFilterTemplate」	検索フィルターテンプレートが含まれます。
「nsSaslMapRegexString」	SASL ID 文字列に一致する正規表現が含まれます。

**3.2.12. nsslapdConfig (オブジェクトクラス)**

**nsslapdConfig** オブジェクトクラスは、Directory Server インスタンスの設定オブジェクト **cn=config** を定義します。

このオブジェクトクラスは Directory Server に定義されています。

**上級クラス**

top

**OID**

2.16.840.1.113730.3.2.39

表3.26 必要な属性

属性	定義
objectClass	エントリーに割り当てられたオブジェクトクラスを指定します。

表3.27 使用できる属性

属性	定義
cn	エントリーの共通名を指定します。

**3.2.13. passwordPolicy (オブジェクトクラス)**

local および global パスワードポリシーはいずれも **passwordPolicy** オブジェクトクラスを取ります。このオブジェクトクラスは Directory Server に定義されています。

**上級クラス**

top

## OID

2.16.840.1.113730.3.2.13

表3.28 必要な属性

属性	定義
objectClass	エントリーに割り当てられたオブジェクトクラスを指定します。

表3.29 使用できる属性

属性	定義
「passwordMaxAge (パスワードの最大年齢)」	ユーザーパスワードの有効期限が切れるまでの秒数を設定します。
「passwordExp (パスワードの有効期限)」	<b>passwordMaxAge</b> 属性で指定された間隔の後に、ユーザーのパスワードの有効期限が切れるかどうかを指定します。
「passwordMinLength (パスワードの最小長)」	パスワードで使用する必要がある文字の最小数を設定します。
「passwordInHistory (覚えておくべきパスワードの数)」	ディレクトリーストアが履歴に、パスワード数を設定します。
「passwordChange (パスワード変更)」	ユーザーが自身のパスワードを変更できるかどうかを指定します。
「passwordWarning (警告の送信)」	パスワードの有効期限が近づいているユーザーに警告メッセージが送信されるまでの秒数を設定します。
「passwordLockout (アカウントロックアウト)」	バインドの試行が一定回数失敗した後、ユーザーがディレクトリーからロックアウトされているかを示します。
「passwordMaxFailure (最大パスワードの失敗回数)」	バインドの試行を何回失敗するとユーザーがディレクトリーからロックアウトされるかを設定します。
「passwordUnlock (アカウントのロック解除)」	管理者がパスワードをリセットするまでユーザーをロックアウトするかどうか、特定のロックアウト期間後にユーザーが再度ログインできるかどうかを識別します。デフォルトでは、ロックアウト期間後にユーザーが再度ログインできるようになっています。

属性	定義
「passwordLockoutDuration (Lockout Duration)」	ユーザーがディレクトリーからロックされる時間を秒単位で設定します。
「passwordCheckSyntax (パスワード構文チェック)」	パスワードを保存する前に、サーバーによってパスワード構文をチェックするかどうかを指定します。
「passwordMustChange (パスワードを変更する必要があります)」	ディレクトリーへの初回ログイン時、または Directory Manager でパスワードのリセット後にパスワードを変更するかどうかを指定します。
「passwordStorageScheme (パスワードの保存スキーム)」	Directory Server のパスワード保存に使用する暗号化のタイプを設定します。
「passwordMinAge (パスワードの最短有効期限)」	ユーザーが次にパスワードを変更するまでに待機する必要がある秒数を設定します。
「passwordResetFailureCount (パスワードの失敗回数のリセット)」	パスワード障害カウンターをリセットするまでの時間を秒単位で設定します。無効なパスワードがユーザーのアカウントから送信されるたびに、パスワードの失敗カウンターがインクリメントされます。
「passwordGraceLimit (パスワードの有効期限)」	ユーザーのパスワードの有効期限が切れたときに猶予として許可されるログインの数を設定します。
「PasswordMinDigits (パスワード構文)」	パスワードで使用する必要がある数値の最小数 (0 から 9) を設定します。
「passwordMinAlphas (パスワード構文)」	パスワードで使用する必要があるアルファベット文字の最小数を設定します。
「PasswordMinUppers (パスワード構文)」	パスワードで使用する必要がある大文字の英字 (A から Z) の最小数を設定します。
「PasswordMinLowers (パスワード構文)」	パスワードで使用する必要がある小文字の英字 (a から z) の最小数を設定します。
「PasswordMinSpecials (パスワード構文)」	パスワードで使用する必要がある !@#\$. などの特殊 ASCII 文字の最小数を設定します。
「passwordMin8Bit (パスワード構文)」	パスワードで使用される 8 ビット文字の最小数を設定します。
「passwordMaxRepeats (パスワード構文)」	同じ文字を連続して使用できる最大回数を設定します。
「passwordMinCategories (パスワード構文)」	パスワードで使用する必要があるカテゴリーの最小数を設定します。



属性	定義
「PasswordMinTokenLength (パスワード構文)」	普通の言葉をチェックする長さを設定します。
「passwordTPRDelayValidFrom」	一時パスワードが有効になるまでの遅延を設定します。
「passwordTPRDelayExpireAt」	一時パスワードが有効な期間を秒数で設定します。
「passwordTPRMaxUse: 5」 一時パスワードを使用できる最大試行回数を設定します	

### 3.3. ROOT DSE 属性

このセクションの属性を使用して、サーバーインスタンスのルートディレクトリーサーバーエントリー (DSE) を定義します。DSE で定義された情報は、そのサーバーソフトウェアのそのバージョンでサポートされる制御、メカニズム、機能など、サーバーインスタンスの実際の設定に関連します。また、インスタンスに固有の情報 (ビルド番号やインストール日など) も含まれます。

DSE は通常の DIT 以外の特別なエントリーで、null 検索ベースで検索して返すことができます。以下に例を示します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -s base -b ""
"objectclass=*"

```

#### 3.3.1. dataversion

この属性には、ディレクトリー内のデータの最新の編集時刻を示すタイムスタンプが含まれます。

```
dataversion: 020090923175302020090923175302
```

OID	
構文	GeneralizedTime
多値または単一値	単一値
定義される場所	Directory Server

#### 3.3.2. defaultNamingContext

クライアントがデフォルトで使用する必要のあるすべての設定済みの命名コンテキストの中から特定の命名コンテキストに対応します。

OID	
-----	--

構文	DN
多値または単一値	単一値
定義される場所	Directory Server

### 3.3.3. lastusn

USN プラグインは、書き込み操作 (add、change、delete、および modrdn) がそのエントリーに対して実行されるたびに、すべてのエントリーにシーケンス番号を割り当てます。USN はエントリーの USN 操作属性に割り当てられます。

USN プラグインには、ローカルおよびグローバルの 2 つのモードがあります。

ローカルモードでは、サーバーインスタンス用に維持される各データベースには、バックエンドデータベースごとに個別の USN カウンターが割り当てられた USN プラグインの独自のインスタンスがあります。データベースのエントリーに割り当てられた最新の USN は **lastusn** 属性に表示されます。USN プラグインがローカルモードに設定されていると、**lastUSN** 属性は USN と USN が割り当てられたデータベースの両方を表示します。

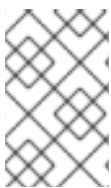
```
lastusn;database_name:USN
```

以下に例を示します。

```
lastusn;example1: 213
lastusn;example2: 207
```

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には、**lastUSN** 値では、データベースで割り当てられた最新の USN を表示します。

```
lastusn: 420
```



#### 注記

この属性は、内部のサーバー操作をカウントしません。バックエンドデータベースでの通常の実行操作があつた場合のみ (add、change、delete、および modrdn) が USN 数が 1 つ増えます。

構文	整数
多値または単一値	複数值
定義される場所	Directory Server

### 3.3.4. namingContexts

サーバーが制御またはシャドウする命名コンテキストに対応します。Directory Server で情報を制御しない場合は (パブリック X.500 ディレクトリーへの LDAP ゲートウェイである場合など)、この属性は

ありません。Directory Server がディレクトリー全体を含むと判断した場合は、この属性の値は1つで、その値は空の文字列 (root の Null DN を示す) です。この属性を使用し、サーバーに接続しているクライアントは、検索に適したベースオブジェクトを選択できます。

OID	1.3.6.1.4.1.1466.101.120.5
構文	DN
多値または単一値	複数値
定義される場所	<a href="#">RFC 2252</a>

### 3.3.5. netscapemdsuffix

この属性には、サーバーで保持されるマシンデータのディレクトリーツリーの最上位接尾辞の DN が含まれます。DN 自体は LDAP URL を参照します。以下に例を示します。

```
cn=ldap://dc=server_name,dc=example,dc=com:389
```

OID	2.16.840.1.113730.3.1.212
構文	DN
多値または単一値	単一値
定義される場所	Directory Server

### 3.3.6. supportedControl

この属性の値は、サーバーでサポートされる制御を識別するオブジェクト識別子 (OID) です。サーバーが制御に対応していない場合には、この属性は存在しません。

OID	1.3.6.1.4.1.1466.101.120.13
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2252</a>

### 3.3.7. supportedExtension

この属性の値は、サーバーでサポートされる拡張操作を識別するオブジェクト識別子 (OID) です。サーバーが拡張操作に対応していない場合には、この属性は存在しません。

OID	1.3.6.1.4.1.1466.101.120.7
-----	----------------------------

構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2252</a>

### 3.3.8. supportedFeatures

この属性には、Red Hat Directory Server の現行バージョンでサポートされている機能が含まれます。

OID	1.3.6.1.4.1.4203.1.3.5
構文	OID
多値または単一値	複数値
定義される場所	<a href="#">RFC 3674</a>

### 3.3.9. supportedLDAPVersion

この属性は、サーバーで実装された LDAP プロトコルのバージョンを識別します。

OID	1.3.6.1.4.1.1466.101.120.15
構文	整数
多値または単一値	複数値
定義される場所	<a href="#">RFC 2252</a>

### 3.3.10. supportedSASLMechanisms

この属性は、サーバーでサポートされる SASL メカニズムの名前を識別します。サーバーが SASL 属性に対応していない場合には、この属性は存在しません。

OID	1.3.6.1.4.1.1466.101.120.14
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2252</a>

### 3.3.11. vendorName

この属性には、サーバーベンダーの名前が含まれます。

OID	1.3.6.1.1.4
構文	DirectoryString
多値または単一値	単一値
定義される場所	<a href="#">RFC 3045</a>

### 3.3.12. vendorVersion

この属性では、サーバーのベンダーのバージョン番号を示します。

OID	1.3.6.1.1.5
構文	DirectoryString
多値または単一値	単一値
定義される場所	<a href="#">RFC 3045</a>

## 3.4. レガシーの属性

この属性は Directory Server 4.x 以前では標準でした。この属性は互換性を確保するためにスキーマに含まれますが、Directory Server の現行バージョンには含まれていません。

### 3.4.1. レガシーサーバーの属性

これらの属性は元は、Directory Server 4.x 以前のサーバーのサーバーインスタンスエントリー設定に使用されていました。

#### 3.4.1.1. LDAPServer (オブジェクトクラス)

このオブジェクトクラスは LDAP サーバー情報を識別します。Directory Server で定義されます。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.35

表3.30 必要な属性

属性	定義
----	----

属性	定義
objectClass	エントリーに割り当てられたオブジェクトクラスを指定します。
cn	エントリーの一般的な名前を指定します。

表3.31 使用できる属性

属性	定義
description	エントリーのテキスト説明を入力します。
l (localityName)	エントリーの市または地理的な場所を指定します。
ou (organizationalUnitName)	アカウントが属する組織単位または部門を指定します。
seeAlso	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
generation	サーバーの生成文字列を保存します。
changeLogMaximumAge	changelog の最大期間を指定します。
changeLogMaximumSize	changelog の最大サイズを指定します。

### 3.4.1.2. changeLogMaximumAge

これにより、サーバーで維持される changelog の最大期間が設定されます。

OID	2.16.840.1.113730.3.1.200
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 3.4.1.3. changeLogMaximumConcurrentWrites

この属性は、changelog に書き込むことのできる同時書き込みの最大数を設定します。

OID	2.16.840.1.113730.3.1.205
-----	---------------------------

構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

#### 3.4.1.4. changeLogMaximumSize

この属性は、changelog の最大サイズを設定します。

OID	2.16.840.1.113730.3.1.201
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

#### 3.4.1.5. generation

この属性には、その特定のサーバーとバージョンを一意に識別するバイトベクトルが含まれます。この数字で、レプリケーション中にサーバーを区別します。

OID	2.16.840.1.113730.3.1.612
構文	IA5String
多値または単一値	複数値
定義される場所	Directory Server

#### 3.4.1.6. nsSynchUniqueAttribute

この属性は Windows 同期に使用されます。

OID	2.16.840.1.113730.3.1.407
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

#### 3.4.1.7. nsSynchUserIDFormat

この属性は Windows 同期に使用されます。

OID	2.16.840.1.113730.3.1.406
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server



## 第4章 プラグイン実装サーバー機能リファレンス

本章では、Red Hat Directory Server プラグインの参考情報を紹介します。

Directory Server プラグイン機能の各設定には、独自のエントリーと、サブツリー **cn=plugins,cn=config** 下の属性セットがあります。

```
dn: cn=Telephone Syntax,cn=plugins,cn=config
objectclass: top
objectclass: nsSlapdPlugin
objectclass: extensibleObject
cn: Telephone Syntax
nsslapd-pluginPath: libsyntax-plugin
nsslapd-pluginInitfunc: tel_init
nsslapd-pluginType: syntax
nsslapd-pluginEnabled: on
```

これらの属性の一部はすべてのプラグインに共通するものであり、特定のプラグインに固有の場合もあります。**cn=config** サブツリーで **ldapsearch** を実行して、特定のプラグインで現在使用されている属性を確認します。

すべてのプラグインは **nsSlapdPlugin** オブジェクトクラスのインスタンスで、**extensibleObject** オブジェクトクラスから継承されます。サーバーによって考慮されるプラグイン設定属性については、以下の例のように、これらのオブジェクトクラス (**最上位** のオブジェクトクラスに加え) の両方がエントリーに存在する必要があります。

```
dn:cn=ACL Plugin,cn=plugins,cn=config
objectclass:top
objectclass:nsSlapdPlugin
objectclass:extensibleObject
```

### 4.1. サーバープラグインの機能リファレンス

以下の表は、Directory Server で提供されるプラグインの概要と、設定可能な引数、設定可能な引数、デフォルト設定、依存関係、一般的なパフォーマンス関連情報、および詳細な情報を示しています。これらの表は、プラグインのパフォーマンスの向上とコストを比較して、デプロイメントに最適な設定を選択するのに役立ちます。**詳細情報** セクションでは、該当箇所は、関連情報について相互参照しています。

#### 4.1.1. 7-bit Check プラグイン

プラグインパラメーター	説明
プラグイン ID	NS7bitAtt
設定エントリーの DN	cn=7-bit check,cn=plugins,cn=config
説明	特定の属性が7ビットクリーニングであることを確認します。
タイプ	preoperation

プラグインパラメーター	説明
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
属性のリスト ( <b>uid mail userpassword</b> ) の後に","が続き、チェックが実行される接尾辞が付きます。	依存関係
データベース	パフォーマンス関連の情報
なし	追加情報

#### 4.1.2. ACL プラグイン

プラグインパラメーター	説明
プラグイン ID	acl
設定エントリーの DN	cn=ACL Plugin,cn=plugins,cn=config
説明	ACL アクセスチェックプラグイン
型	accesscontrol
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
データベース	パフォーマンス関連の情報
アクセス制御により、パフォーマンスが最小限に抑えられます。サーバーのアクセス制御の主な方法であるため、このプラグインを有効にしたままにしておきます。	追加情報

#### 4.1.3. ACL Preoperation プラグイン

プラグインパラメーター	説明
プラグイン ID	acl
設定エントリーの DN	cn=ACL preoperation,cn=plugins,cn=config
説明	ACL アクセスチェックプラグイン
型	preoperation
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
データベース	パフォーマンス関連の情報
アクセス制御により、パフォーマンスが最小限に抑えられます。サーバーのアクセス制御の主な方法であるため、このプラグインを有効にしたままにしておきます。	追加情報

#### 4.1.4. Account Policy プラグイン

プラグインパラメーター	説明
プラグイン ID	none
設定エントリーの DN	cn=Account Policy Plugin,cn=plugins,cn=config
説明	特定の有効期限またはアクティブではない期間を経過後にユーザーアカウントをロックするポリシーを定義します。
型	object
設定可能な引数	on
off	デフォルト設定
off	設定可能な引数
グローバルアカウントポリシー設定を含む設定エントリーへのポインター。	依存関係

プラグインパラメーター	説明
データベース	パフォーマンス関連の情報
なし	追加情報

#### 4.1.5. Account Usability プラグイン

プラグインパラメーター	説明
プラグイン ID	acctusability
設定エントリーの DN	cn=Account Usability Plugin,cn=plugins,cn=config
説明	指定したユーザーとして実際に認証することなく、アカウントの認証ステータスまたはユーザービリティを確認します。
型	preoperation
設定可能な引数	on
off	デフォルト設定
on	依存関係
データベース	パフォーマンス関連の情報

#### 4.1.6. AD DN プラグイン

プラグインパラメーター	説明
プラグイン ID	addn
設定エントリーの DN	cn=addn,cn=plugins,cn=config
説明	バインド操作で、 <b>user_name</b> and <b>user_name@domain</b> などの Active Directory 形式のユーザー名を使用できるようにします。
型	preoperation
設定可能な引数	on
off	デフォルト設定

プラグインパラメーター	説明
off	設定可能な引数
<b>addn_default_domain</b> : ドメインなしでユーザー名に自動的に追加されるデフォルトドメインを設定します。	依存関係
None	パフォーマンス関連の情報

#### 4.1.7. Attribute Uniqueness プラグイン

プラグインパラメーター	説明
プラグイン ID	NSUniqueAttr
設定エントリーの DN	cn=Attribute Uniqueness,cn=plugins,cn=config
説明	エントリーに変更が発生するたびに、指定された属性の値が一意であることを確認します。たとえば、ほとんどのサイトでは、ユーザー ID とメールアドレスは一意でなければなりません。
型	preoperation
設定可能な引数	on
off	デフォルト設定
off	設定可能な引数
リスト表示されているすべてのサブツリーで UID 属性が一意であることを確認するには、 <b>uid "DN" "DN"....</b> を入力します。ただし、 <b>requiredObjectClass</b> でエントリーの追加または更新時に UID 属性の一意性を確認するには、 <b>attribute="uid" MarkerObjectclass = "ObjectClassName"</b> と入力し、必要に応じて <b>requiredObjectClass = "ObjectClassName"</b> と入力します。これは、 <b>MarkerObjectClass</b> 属性で定義される <b>ObjectClass</b> を含む親エントリーから必要なオブジェクトクラスのチェックを開始します。	依存関係
データベース	パフォーマンス関連の情報

プラグインパラメーター	説明
<p>Directory Server は、デフォルトで UID Uniqueness プラグインを提供します。他の属性の値を一意にするには、これらの属性の Attribute Uniqueness プラグインのインスタンスを作成します。属性一意性プラグインに関する詳細は、<b>Red Hat Directory Server 管理ガイド</b>の属性の一意性プラグインの使用セクションを参照してください。</p> <p>マルチサプライヤーレプリケーション環境でプラグインを有効にする前に対処する必要がある操作制限があるので、UID Uniqueness プラグインはデフォルトでオフになっています。プラグインを有効すると、Directory Server のパフォーマンスが遅くなる可能性があります。</p>	追加情報

#### 4.1.8. Auto Membership プラグイン

プラグインパラメーター	説明
プラグイン ID	自動メンバーシップ
設定エントリーの DN	cn=Auto Membership,cn=plugins,cn=config
説明	自動メンバー定義のコンテナ項目。Automember 定義は新規エントリーを検索して定義した LDAP 検索フィルターおよび正規表現条件と一致する場合は、指定されたグループにエントリーを自動的に追加します。
型	preoperation
設定可能な引数	on
off	デフォルト設定
off	設定可能な引数
メインプラグインエントリーの場合はなしです。定義エントリーでは、LDAP スコープ、LDAP フィルター、デフォルトグループ、およびメンバー属性形式を指定する必要があります。任意の正規表現の子エントリーは、包含式と排他式、異なるターゲットグループを指定できます。	依存関係
データベース	パフォーマンス関連の情報
なし。	追加情報

## 4.1.9. Binary Syntax プラグイン

**警告**

バイナリー構文は非推奨です。代わりに Octet String 構文を使用します。

プラグインパラメーター	説明
プラグイン ID	bin-syntax
設定エントリーの DN	cn=Binary Syntax,cn=plugins,cn=config
説明	バイナリーデータを処理する構文。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

## 4.1.10. Bit String Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	bitstring-syntax
設定エントリーの DN	cn=Bit String Syntax,cn=plugins,cn=config
説明	ビット文字列構文の値および関連するマッチングルールをサポートします ( <a href="#">RFC 4517</a> )。
型	構文

プラグインパラメーター	説明
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に行を実行することを推奨します。	追加情報

#### 4.1.11. Bitwise プラグイン

プラグインパラメーター	説明
プラグイン ID	bitwise
設定エントリーの DN	cn=Bitwise Plugin,cn=plugins,cn=config
説明	LDAP サーバーに対してビット単位の操作を実行するマッチングルール
型	matchingrule
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に行を実行することを推奨します。	追加情報

#### 4.1.12. Boolean Syntax プラグイン



プラグインパラメーター	説明
プラグイン ID	boolean-syntax
設定エントリーの DN	cn=Boolean Syntax,cn=plugins,cn=config
説明	ブール値構文値 (TRUE または FALSE) および関連のマッチングルールをサポートします ( <a href="#">RFC 4517</a> )。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に行うことを推奨します。	追加情報

#### 4.1.13. Case Exact String Syntax 構文プラグイン

プラグインパラメーター	説明
プラグイン ID	ces-syntax
設定エントリーの DN	cn=Case Exact String Syntax,cn=plugins,cn=config
説明	大文字と小文字が区別する照合または Directory String、IA5 String、および関連する構文をサポートします。これは大文字と小文字を区別する構文ではありません。このプラグインは、異なる文字列構文に対して大文字と小文字を区別する照合ルールを提供します。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数

プラグインパラメーター	説明
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

#### 4.1.14. Case Ignore String Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	directorystring-syntax
設定エントリーの DN	cn=Case Ignore String Syntax,cn=plugins,cn=config
説明	Directory String、IA5 String、および関連する構文の大文字と小文字を区別しない照合ルールをサポートします。これは大文字と小文字を区別しない構文ではありません。このプラグインは異なる文字列構文に対して大文字と小文字を区別する照合ルールを提供します。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

#### 4.1.15. Chaining Database プラグイン

プラグインパラメーター	説明
プラグイン ID	chaining database

プラグインパラメーター	説明
設定エントリーの DN	cn=Chaining database,cn=plugins,cn=config
説明	バックエンドデータベースをリンクできるようにします
型	database
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
チェーンデータベースに関連するパフォーマンス関連のチューニングパラメーターは多数あります。Red Hat Directory Server 管理ガイドのデータベースリンクの維持セクションを参照してください。	追加情報

#### 4.1.16. Class of Service プラグイン

プラグインパラメーター	説明
プラグイン ID	cos
設定エントリーの DN	cn=Class of Service,cn=plugins,cn=config
説明	エントリー間で属性を共有できます。
型	object
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係

プラグインパラメーター	説明
<ul style="list-style-type: none"> <li>* <b>Type:</b> データベース</li> <li>* <b>名前付き:</b> 状態変更プラグイン</li> <li>* <b>名前付き:</b> ビュープラグイン</li> </ul>	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。このプラグインは常に実行したままにします。	追加情報

#### 4.1.17. Content Synchronization プラグイン

プラグインパラメーター	説明
プラグイン ID	content-sync-plugin
設定エントリーの DN	cn=Content Synchronization,cn=plugins,cn=config
説明	<a href="#">RFC 4533</a> に従って、Directory Server の <b>SyncRepl</b> プロトコルのサポートを有効にします。
型	object
設定可能な引数	on
off	デフォルト設定
off	設定可能な引数
None	依存関係
Retro Changelog プラグイン	パフォーマンス関連の情報
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、それに応じて <b>Retro Changelog</b> プラグインのスコープを制限します。	追加情報

#### 4.1.18. Country String Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	countrystring-syntax
設定エントリーの DN	cn=Country String Syntax,cn=plugins,cn=config

プラグインパラメーター	説明
説明	国名構文の値および関連マッチングルールをサポートします (RFC 4517)。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

#### 4.1.19. Delivery Method Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	delivery-syntax
設定エントリーの DN	cn=Delivery Method Syntax,cn=plugins,cn=config
説明	推奨される配信メソッドおよび関連マッチングルールのリストである値をサポートします (RFC 4517)。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

## 4.1.20. deref プラグイン

プラグインパラメーター	説明
プラグイン ID	間接参照
設定エントリーの DN	cn=deref,cn=plugins,cn=config
説明	ディレクトリー検索における間接参照制御
型	preoperation
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
データベース	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

## 4.1.21. Distinguished Name Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	dn-syntax
設定エントリーの DN	cn=Distinguished Name Syntax,cn=plugins,cn=config
説明	DN 値の構文および関連マッチングルールをサポートします ( <a href="#">RFC 4517</a> )。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係

プラグインパラメーター	説明
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

#### 4.1.22. Distributed Numeric Assignment プラグイン

プラグイン情報	説明
プラグイン ID	分散数値割り当て
設定エントリー DN	cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
説明	Distributed Numeric Assignme プラグイン
型	preoperation
設定可能な引数	on
off	デフォルト設定
off	設定可能な引数
	依存関係
データベース	パフォーマンス関連の情報
なし	追加情報

#### 4.1.23. Enhanced Guide Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	enhancedguide-syntax
設定エントリーの DN	cn=Enhanced Guide Syntax,cn=plugins,cn=config
説明	検索を構築するために、属性およびフィルターに基づいて複雑な基準を作成するための構文および関連マッチングルールをサポートします (RFC 4517)。

プラグインパラメーター	説明
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

#### 4.1.24. Facsimile Telephone Number Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	facsimile-syntax
設定エントリーの DN	cn=Facsimile Telephone Number Syntax,cn=plugins,cn=config
説明	fax 番号の構文および関連マッチングルールをサポートします ( <a href="#">RFC 4517</a> )。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に実行することを推奨します。	追加情報



## 4.1.25. Fax Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	fax-syntax
設定エントリーの DN	cn=Fax Syntax,cn=plugins,cn=config
説明	ファックスされたオブジェクトのイメージを保存するための構文および関連マッチングルールをサポートします (RFC4517)。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

## 4.1.26. Generalized Time Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	time-syntax
設定エントリーの DN	cn=Generalized Time Syntax,cn=plugins,cn=config
説明	日付、時間、およびタイムゾーンを処理する構文および関連マッチングルールをサポートします (RFC 4517)。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数

プラグインパラメーター	説明
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

#### 4.1.27. Guide Syntax プラグイン



#### 警告

この構文は非推奨です。代わりに Enhanced Guide 構文を使用してください。

プラグインパラメーター	説明
プラグイン ID	guide-syntax
設定エントリーの DN	cn=Guide Syntax,cn=plugins,cn=config
説明	検索を構築するための、属性とフィルターに基づいた複雑な条件を作成する構文
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

#### 4.1.28. HTTP Client プラグイン

プラグインパラメーター	説明
プラグイン ID	http-client
設定エントリーの DN	cn=HTTP Client,cn=plugins,cn=config
説明	HTTP クライアントプラグイン
型	preoperation
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
データベース	パフォーマンス関連の情報
	追加情報

#### 4.1.29. Integer Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	int-syntax
設定エントリーの DN	cn=Integer Syntax,cn=plugins,cn=config
説明	整数構文および関連マッチングルールをサポートします ( <a href="#">RFC 4517</a> )。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報

プラグインパラメーター	説明
このプラグインの設定は変更しないでください。Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

#### 4.1.30. Internationalization プラグイン

プラグインパラメーター	説明
プラグイン ID	orderingrule
設定エントリーの DN	cn=Internationalization Plugin,cn=plugins,cn=config
説明	国際化された文字列をディレクトリーで並べ替えることができます
型	matchingrule
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
Internationalization プラグインには、 <b>/etc/dirsrv/config/slapd-collations.conf</b> ファイルの場所を指定する引数が1つあり、これは変更しないようにしてください。このファイルは、国際化プラグインが使用する照合順序とロケールを保存します。	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

#### 4.1.31. JPEG Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	jpeg-syntax

プラグインパラメーター	説明
設定エントリーの DN	cn=JPEG Syntax,cn=plugins,cn=config
説明	JPEG イメージデータの構文および関連のマッチングルールをサポートします (RFC 4517)。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

#### 4.1.32. ldbm database プラグイン

プラグインパラメーター	説明
プラグイン ID	ldbm-backend
設定エントリーの DN	cn=ldbm database,cn=plugins,cn=config
説明	ローカルデータベースの実装
型	database
設定可能な引数	
デフォルト設定	on
設定可能な引数	None
依存関係	* 構文 * matchingRule
パフォーマンス関連の情報	データベース設定の詳細は、「 <a href="#">データベースプラグインの属性</a> 」を参照してください。

プラグインパラメーター	説明
追加情報	Red Hat Directory Server 管理ガイドのディレクトリーデータベースの設定の章を参照してください。

#### 4.1.33. Linked Attributes プラグイン

プラグインパラメーター	説明
プラグイン ID	リンクされた属性
設定エントリーの DN	cn=Linked Attributes,cn=plugins,cn=config
説明	リンクされた管理属性設定エントリーのコンテナエントリー。コンテナの各設定エントリーは属性と属性をリンクして、あるエントリー (マネージャーエントリーなど) が更新されると、そのエントリーに関連付けられたエントリー (カスタムの <b>directReports</b> 属性など) がユーザー指定の対応する属性で自動的に更新されます。
型	preoperation
設定可能な引数	on
off	デフォルト設定
off	設定可能な引数
メインプラグインエントリーの場合はなしです。各プラグインインスタンスには、以下の3つの属性があります。  * linkType - プラグインが監視するプライマリー属性を設定します。  * managedType - linkType の属性が変更されるたびにプラグインによって動的に管理される属性を設定します。  * linkScope - ディレクトリーツリー内の特定のサブツリーにプラグインのアクティビティを制限します。	依存関係
データベース	パフォーマンス関連の情報
linkType に設定された属性は、DN 形式の値のみを許可する必要があります。managedType に設定された属性は複数值である必要があります。	追加情報

## 4.1.34. Managed Entries プラグイン

プラグイン情報	説明
プラグイン ID	マネージドエントリー
設定エントリー DN	cn=Managed Entries,cn=plugins,cn=config
説明	自動生成されるディレクトリーエントリーのコンテナエントリー。各設定エントリーは、ターゲットサブツリーとテンプレートエントリーを定義します。ターゲットサブツリーの一一致するエントリーが作成されると、プラグインはテンプレートに基づいて新規の関連エントリーを自動的に作成します。
型	preoperation
設定可能な引数	on
off	デフォルト設定
off	設定可能な引数
メインプラグインエントリーの場合はなしです。各プラグインインスタンスには、以下の4つの属性があります。  * originScope - 検索ベースを設定します。  * originFilter - 一致するエントリーの検索ベースを設定します。  * managedScope - 新しい管理エントリーを作成するサブツリーを設定します。  * managedTemplate - 管理エントリーの作成に使用されるテンプレートエントリー	依存関係
データベース	パフォーマンス関連の情報
なし	追加情報

## 4.1.35. MemberOf プラグイン

プラグイン情報	説明
プラグイン ID	memberOf
設定エントリー DN	cn=MemberOf Plugin,cn=plugins,cn=config

プラグイン情報	説明
説明	グループエントリーの <b>member</b> 属性に基づいて、ユーザーエントリーの <b>memberOf</b> 属性を管理します。
型	postoperation
設定可能な引数	on
off	デフォルト設定
off	設定可能な引数
* <b>memberOfAttr</b> は、ユーザーのエントリーで生成してグループメンバーシップを表示する属性を設定します。  * <b>memberOfGroupAttr</b> はグループメンバーの DN の識別に使用する属性を設定します。	依存関係
データベース	パフォーマンス関連の情報
なし	追加情報

#### 4.1.36. Multi-master Replication プラグイン

プラグインパラメーター	説明
プラグイン ID	replication-multimaster
設定エントリーの DN	cn=Multimaster Replication plugin,cn=plugins,cn=config
説明	2 つの現在の Directory Servers 間のレプリケーションを有効にします。
型	object
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係



プラグインパラメーター	説明
* <b>Named:</b> ldbm データベース	パフォーマンス関連の情報
* <b>Named:</b> DES	
* <b>Named:</b> サービスのクラス	
	追加情報

#### 4.1.37. Name and Optional UID Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	nameoptuid-syntax
設定エントリーの DN	cn=Name And Optional UID Syntax,cn=plugins,cn=config
説明	任意かつ一意の ID で DN を保存し、検索する構文および関連マッチングルールをサポートします ( <a href="#">RFC 4517</a> )。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に行うことを推奨します。	追加情報

#### 4.1.38. Numeric String Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	numstr-syntax
設定エントリーの DN	cn=Numeric String Syntax,cn=plugins,cn=config

プラグインパラメーター	説明
説明	数値とスペースの文字列の構文および関連マッチングルールをサポートします (RFC 4517)。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。Red Hat は、このプラグインを常に行うことを推奨します。	追加情報

#### 4.1.39. Octet String Syntax プラグイン



##### 注記

非推奨の Binary の代わりに Octet String 構文を使用してください。

プラグインパラメーター	説明
プラグイン ID	octetstring-syntax
設定エントリーの DN	cn=Octet String Syntax,cn=plugins,cn=config
説明	オクテット文字列構文および関連マッチングルールをサポートします (RFC 4517)。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係

プラグインパラメーター	説明
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に行を実行することを推奨します。	追加情報

#### 4.1.40. OID Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	oid-syntax
設定エントリーの DN	cn=OID Syntax,cn=plugins,cn=config
説明	オブジェクト識別子 (OID) 構文および関連マッチングルールをサポートします ( <a href="#">RFC 4517</a> )。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に行を実行することを推奨します。	追加情報

#### 4.1.41. PAM Pass Through Auth プラグイン

プラグインパラメーター	説明
プラグイン ID	pam_passthruauth
設定エントリーの DN	cn=PAM Pass Through Auth,cn=plugins,cn=config
説明	PAM のパススルー認証を有効にし、PAM サービスは Directory Server をユーザー認証ストアとして使用できます。

プラグインパラメーター	説明
型	preoperation
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
データベース	パフォーマンス関連の情報
	追加情報

#### 4.1.42. Pass Through Authentication プラグイン

プラグインパラメーター	説明
プラグイン ID	passthruauth
設定エントリーの DN	cn=Pass Through Authentication,cn=plugins,cn=config
説明	<b>パススルー認証</b> を有効にします。これにより、1つのディレクトリーが別のディレクトリーでバインド要求の認証を行うことができます。
型	preoperation
設定可能な引数	on
off	デフォルト設定
off	設定可能な引数
ldap://example.com:389/o=example	依存関係
データベース	パフォーマンス関連の情報
パススルー認証では、リモートサーバーにさらにホップを追加する必要があるため、バインド要求が若干遅くなります。 <b>Red Hat Directory Server 管理ガイド</b> のパススルー認証の使用の章を参照してください。	追加情報

#### 4.1.43. パスワードストレージスキーム

Directory Server は、パスワードストレージスキームをプラグインとして実装します。ただし、**cn=Password Storage Schemes,cn=plugins,cn=config** エントリー自体は単なるコンテナであり、プラグインエントリーではありません。パスワードストレージスキームプラグインはすべて、このコンテナのサブエントリーとして保存されます。

パスワードストレージスキームプラグインをすべて表示するには、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 \
  -b "cn=Password Storage Schemes,cn=plugins,cn=config" -s sub "(objectclass=*)" dn
```



#### 警告

Red Hat は、パスワードスキームプラグインを無効にしたり、予期しない認証動作を防ぐためにプラグインの設定を変更したりしないことを推奨します。

#### 強力なパスワードストレージスキーム

Red Hat は、以下の強力なパスワードストレージスキーム (最も強力なパスワードストレージスキーム) のみを使用することを推奨します。

- **PBKDF2\_SHA256** (デフォルト)

パスワードベースの鍵導出関数 2(PBKDF2) は、ブルートフォース攻撃に対応するリソースを当てるように設計されています。PBKDF2 は、ハッシュアルゴリズムを適用する可変回数の反復をサポートします。反復回数が高くなるとセキュリティが向上しますが、必要となるハードウェアリソースが増えます。Directory Server では、**PBKDF2\_SHA256** スキームは、30,000 回の反復を使用して実装し、SHA256 アルゴリズムを適用されます。この値はハードコーディングされ、管理者による操作を必要とせずに、Directory Server の今後のバージョンで増えます。



#### 注記

Red Hat Enterprise Linux 6 のネットワークセキュリティサービス (NSS) データベースは PBKDF2 をサポートしません。したがって、Directory Server 9 のレプリケーショントポロジーでは、このパスワードスキームを使用することはできません。

- **SSHA512**

SSHA(Salted Secure hashing algorithm) は、無作為に生成された salt を使用してハッシュ化されたパスワードのセキュリティを向上させる、セキュアなハッシュアルゴリズム (SHA) の強化バージョンを実装します。**SSHA512** は 512 ビットを使用してハッシュアルゴリズムを実装します。

#### 脆弱なパスワード保存スキーム

Directory Server は、推奨される強力なパスワードストレージスキームのほかに、後方互換性として、以下の強度の低いスキームをサポートします。

<b>AES</b>	<b>CLEAR</b>	<b>CRYPT</b>
------------	--------------	--------------

CRYPT-MD5	CRYPT-SHA256	CRYPT-SHA512
DES	MD5	NS-MTA-MD5 [a]
SHA [b]	SHA256	SHA384
SHA512	SMD5	SSHA
SSHA256	SSHA384	

[a] Directory Server は、このスキームを使用した認証のみをサポートします。パスワードの暗号化に使用できなくなりました。

[b] 160 ビット



### 重要

セキュリティリスクが高くなるので、強度の低いスキームは短期間のみ使用を継続するようにします。

#### 4.1.44. Posix Winsync API プラグイン

プラグインパラメーター	説明
プラグイン ID	posix-winsync-plugin
設定エントリーの DN	cn=Posix Winsync API,cn=plugins,cn=config
説明	Active Directory ユーザーおよびグループエントリーに設定された Posix 属性の Windows 同期を有効にして設定します。
型	preoperation
設定可能な引数	* on

プラグインパラメーター	説明
off  * memberUID マッピング (グループ)  * 小文字 (グループ) での memberUID 値の変換およびソート  * 同期操作による memberOf 修正タスク  * Windows 2003 Posix スキーマを使用	デフォルト設定
off	設定可能な引数
None	依存関係

#### 4.1.45. Postal Address String Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	postaladdress-syntax
設定エントリーの DN	cn=Postal Address Syntax,cn=plugins,cn=config
説明	住所構文および関連マッチングルールをサポートします ( <a href="#">RFC 4517</a> )。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

#### 4.1.46. Printable String Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	printablestring-syntax
設定エントリーの DN	cn=Printable String Syntax,cn=plugins,cn=config
説明	英数字および選択句読点文字列 (RFC 4517 で定義されている出力可能な文字列に準拠する文字列の場合) の構文およびマッチングルールをサポートします。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

#### 4.1.47. Referential Integrity Postoperation プラグイン

プラグインパラメーター	説明
プラグイン ID	referint
設定エントリーの DN	cn=Referential Integrity Postoperation,cn=plugins,cn=config
説明	サーバーが参照整合性を確保できるようにします。
型	postoperation
設定可能な引数	すべての設定および on
off	デフォルト設定
off	設定可能な引数



プラグインパラメーター	説明
有効にすると、操作後の Referential Integrity プラグインは、削除または名前変更操作の直後に、 <b>member</b> 、 <b>uniquemember</b> 、 <b>owner</b> 、および <b>seeAlso</b> で整合性の更新を実行します。プラグインは、他のすべての属性に対して整合性チェックを実行するように設定できます。詳細は、 <a href="#">Directory Server 管理ガイド</a> の該当するセクションを参照してください。	依存関係
データベース	パフォーマンス関連の情報
競合解決ループを回避するために、マルチサブレットレプリケーション環境の1つのサプライヤーでだけ Referential Integrity プラグインを有効にする必要があります。チェーンされたサーバーで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と時間のニーズ、整合性のニーズも分析してください。整合性チェックには、時間がかかり、メモリーと CPU への負荷が多くなる可能性があります。指定されたすべての属性は、存在と等価性の両方にインデックス化する必要があります。	追加情報

#### 4.1.48. Retro Changelog プラグイン

プラグインパラメーター	説明
プラグイン ID	retrocl
設定エントリーの DN	cn=Retro Changelog Plugin,cn=plugins,cn=config
説明	Directory Server 4.x バージョンとのアプリケーション互換性を維持するために LDAP クライアントによって使用されます。Directory Server で発生したすべての変更のログを維持します。retro changelog には、Directory Server の 4.x バージョンの changelog と同じ機能があります。このプラグインは <b>cn=changelog</b> 接尾辞をクライアントに公開するので、クライアントが単純な同期アプリケーションの永続的な検索の有無に関わらず、この接尾辞を使用できます。
型	object
設定可能な引数	on

プラグインパラメーター	説明
off	デフォルト設定
off	設定可能な引数
このプラグインの設定属性の詳細は、「 <a href="#">Retro Changelog プラグインの属性</a> 」を参照してください。	依存関係
* <b>Type:</b> データベース * <b>Named:</b> サービスのクラス	パフォーマンス関連の情報
Directory Server の更新パフォーマンスが低下する可能性があります。	追加情報

#### 4.1.49. Roles プラグイン

プラグインパラメーター	説明
プラグイン ID	roles
設定エントリーの DN	cn=Roles Plugin,cn=plugins,cn=config
説明	Directory Server でのロールの使用を有効にします。
型	object
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
* <b>Type:</b> データベース * <b>名前付き:</b> 状態変更プラグイン * <b>名前付き:</b> ビュープラグイン	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

## 4.1.50. RootDN Access Control プラグイン

プラグインパラメーター	説明
プラグイン ID	rootdn-access-control
設定エントリーの DN	cn=RootDN Access Control,cn=plugins,cn=config
説明	ルート DN エントリーに使用するアクセス制御を有効にして設定します。
型	internalpreoperation
設定可能な引数	on
off	デフォルト設定
off	設定可能な属性
* rootdn-open-time および rootdn-close-time(時間ベースのアクセス制御用) * rootdn-days-allowed (日ベースのアクセス制御用) * rootdn-allow-host、rootdn-deny-host、rootdn-allow-ip、および rootdn-deny-ip(ホストベースのアクセス制御用)	依存関係
None	追加情報

## 4.1.51. Schema Reload プラグイン

プラグイン情報	説明
プラグイン ID	schemareload
設定エントリー DN	cn=Schema Reload,cn=plugins,cn=config
説明	スキーマファイルを再ロードするタスクプラグイン
型	object
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数

プラグイン情報	説明
None	依存関係
None	パフォーマンス関連の情報
	追加情報

#### 4.1.52. Space Insensitive String Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	none
設定エントリーの DN	cn=Space Insensitive String Syntax,cn=plugins,cn=config
説明	スペースに依存しない値を処理するための構文
型	構文
設定可能な引数	on
off	デフォルト設定
off	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

#### 4.1.53. State Change プラグイン

プラグインパラメーター	説明
プラグイン ID	statechange
設定エントリーの DN	cn=State Change Plugin,cn=plugins,cn=config
説明	state-change-notification service サービスを有効にします。

プラグインパラメーター	説明
型	postoperation
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
	追加情報

#### 4.1.54. Syntax Validation Task プラグイン

プラグインパラメーター	説明
プラグイン ID	none
設定エントリーの DN	cn=Syntax Validation Task,cn=plugins,cn=config
説明	属性値の構文検証を有効にします。
型	object
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
	追加情報

#### 4.1.55. Telephone Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	tele-syntax

プラグインパラメーター	説明
設定エントリーの DN	cn=Telephone Syntax,cn=plugins,cn=config
説明	電話番号構文および関連マッチングルールをサポートします (RFC 4517)。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

#### 4.1.56. Teletex Terminal Identifier Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	teletextermid-syntax
設定エントリーの DN	cn=Teletex Terminal Identifier Syntax,cn=plugins,cn=config
説明	相互電話番号構文および関連マッチングルールをサポートします (RFC 4517)。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報

プラグインパラメーター	説明
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に行うことを推奨します。	追加情報

#### 4.1.57. Telex Number Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	telex-syntax
設定エントリーの DN	cn=Telex Number Syntax,cn=plugins,cn=config
説明	テレックス端末のテレックス番号、国コード、およびアンサーバックコードの構文および関連マッチングルールをサポートします (RFC 4517)。
型	構文
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に行うことを推奨します。	追加情報

#### 4.1.58. URI Syntax プラグイン

プラグインパラメーター	説明
プラグイン ID	none
設定エントリーの DN	cn=URI Syntax,cn=plugins,cn=config
説明	一意のリソースロケータ (URL) を含む、一意のリソース識別子 (URI) の構文と関連するマッチングルールをサポートします (RFC 4517)。

プラグインパラメーター	説明
型	構文
設定可能な引数	on
off	デフォルト設定
off	設定可能な引数
None	依存関係
None	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。有効な場合には、Red Hat は、このプラグインを常に実行することを推奨します。	追加情報

#### 4.1.59. USN プラグイン

プラグインパラメーター	説明
プラグイン ID	USN
設定エントリーの DN	cn=USN,cn=plugins,cn=config
説明	エントリーの追加および削除や属性値の変更など、変更が生じるたびに、ディレクトリー内のすべてのエントリーに更新シーケンス番号 (USN) を設定します。
型	object
設定可能な引数	on
off	デフォルト設定
off	設定可能な引数
None	依存関係
データベース	パフォーマンス関連の情報
レプリケーションでは、分数レプリケーションを使用して <b>entryUSN</b> 設定属性を除外することが推奨されます。	追加情報



## 4.1.60. Views プラグイン

プラグインパラメーター	説明
プラグイン ID	ビュー
設定エントリーの DN	cn=Views,cn=plugins,cn=config
説明	Directory Server データベースでのビューの使用を有効にします。
型	object
設定可能な引数	on
off	デフォルト設定
on	設定可能な引数
None	依存関係
* <b>Type:</b> データベース * <b>名前付き:</b> 状態変更プラグイン	パフォーマンス関連の情報
このプラグインの設定は変更しないでください。 Red Hat は、このプラグインを常に行うことを推奨します。	追加情報

## 4.2. すべてのプラグインに共通の属性のリスト

このリストには、簡単な属性の説明、エントリー DN、有効な範囲、デフォルト値、構文、および各属性の例が記載されています。

### 4.2.1. nsslapdPlugin (オブジェクトクラス)

各 Directory Server プラグインは **nsslapdPlugin** オブジェクトクラスに属します。

このオブジェクトクラスは Directory Server に定義されています。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.41

#### 表4.1 必要な属性

属性	定義
objectClass	エントリーに割り当てられたオブジェクトクラスを指定します。
cn	エントリーの共通名を指定します。
「nsslapd-pluginPath」	プラグインライブラリー名を特定します (ライブラリー接尾辞なし)。
「nsslapd-pluginInitfunc」	プラグインの初期化機能を特定します。
「nsslapd-pluginType」	プラグインのタイプを識別します。
「nsslapd-pluginId」	プラグイン ID を特定します。
「nsslapd-pluginVersion」	プラグインのバージョンを特定します。
「nsslapd-pluginVendor」	プラグインのベンダーを特定します。
「nsslapd-pluginDescription」	プラグインの説明を識別します。
「nsslapd-pluginEnabled」	プラグインを有効にするかどうかを特定します。
「nsslapd-pluginPrecedence」	実行順序でプラグインの優先度を設定します。

#### 4.2.2. nsslapd-logAccess

この属性を使用すると、プラグインにより実行される検索操作を、**cn=config** の **nsslapd-accesslog** パラメーターに設定したファイルに記録できます。

プラグインパラメーター	説明
エントリー DN	cn=plug-in name,cn=plugins,cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-logAccess: Off

#### 4.2.3. nsslapd-logAudit

この属性を使用すると、プラグインから送られるデータベースへの変更をログに記録して監査できます。

**nsslapd-auditlog-logging-enabled** パラメーターが **cn=config** で有効になっていると、正常な変更イベントが監査ログに記録されます。プラグインで失敗した変更データベース操作をログに記録するには、**cn=config** で **nsslapd-auditfaillog-logging-enabled** 属性を有効にします。

プラグインパラメーター	説明
エントリー DN	cn=plug-in name,cn=plugins,cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-logAudit: Off

#### 4.2.4. nsslapd-pluginDescription

この属性では、プラグインの説明を追加します。

プラグインパラメーター	説明
エントリー DN	cn=plug-in name,cn=plugins,cn=config
有効な値	
デフォルト値	なし
構文	DirectoryString
例	nsslapd-pluginDescription: acl access check plug-in

#### 4.2.5. nsslapd-pluginEnabled

この属性は、プラグインを有効にするかどうかを指定します。この属性はプロトコルで変更できますが、サーバーが次回再起動されたタイミングでのみ有効になります。

プラグインパラメーター	説明
エントリー DN	cn=plug-in name,cn=plugins,cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString

プラグインパラメーター	説明
例	nsslapd-pluginEnabled: on

#### 4.2.6. nsslapd-pluginId

この属性は、プラグイン ID を指定します。

プラグインパラメーター	説明
エントリー DN	cn=plug-in name,cn=plugins,cn=config
有効な値	有効なプラグイン ID
デフォルト値	なし
構文	DirectoryString
例	nsslapd-pluginId: chaining database

#### 4.2.7. nsslapd-pluginInitfunc

この属性は、起動するプラグイン関数を指定します。

プラグインパラメーター	説明
エントリー DN	cn=plug-in name,cn=plugins,cn=config
有効な値	有効なプラグイン機能
デフォルト値	なし
構文	DirectoryString
例	nsslapd-pluginInitfunc: NS7bitAttr_Init

#### 4.2.8. nsslapd-pluginPath

この属性は、プラグインへの完全パスを指定します。

プラグインパラメーター	説明
エントリー DN	cn=plug-in name,cn=plugins,cn=config
有効な値	有効なパス

プラグインパラメーター	説明
デフォルト値	なし
構文	DirectoryString
例	nsslapd-pluginPath: uid-plugin

#### 4.2.9. nsslapd-pluginPrecedence

この属性は、プラグインの実行順序の優先順位を設定します。優先順位は、プラグインの実行順序を定義し、プラグインの実行前に、プラグインの操作が完了するのを待機できるので、より複雑な環境や対話が可能になります。これは、事前操作および操作後のプラグインにはより重要です。

値が1のプラグインの優先度が最も高く、最初に実行され、値が99のプラグインは優先度が最も低くなります。デフォルトは50です。

プラグインパラメーター	説明
エントリー DN	cn=plug-in name,cn=plugins,cn=config
有効な値	1 から 99
デフォルト値	50
構文	整数
例	nsslapd-pluginPrecedence: 3

#### 4.2.10. nsslapd-pluginType

この属性は、プラグインのタイプを指定します。詳細は「[nsslapd-plugin-depends-on-type](#)」を参照してください。

プラグインパラメーター	説明
エントリー DN	cn=plug-in name,cn=plugins,cn=config
有効な値	有効なプラグインタイプ
デフォルト値	なし
構文	DirectoryString
例	nsslapd-pluginType: preoperation

### 4.2.11. nsslapd-pluginVendor

この属性は、プラグインのベンダーを指定します。

プラグインパラメーター	説明
エントリー DN	cn=plug-in name,cn=plugins,cn=config
有効な値	承認されたプラグインベンダー
デフォルト値	Red Hat, Inc.
構文	DirectoryString
例	nsslapd-pluginVendor: Red Hat, Inc.

### 4.2.12. nsslapd-pluginVersion

この属性は、プラグインのバージョンを指定します。

プラグインパラメーター	説明
エントリー DN	cn=plug-in name,cn=plugins,cn=config
有効な値	有効なプラグインバージョン
デフォルト値	製品バージョン番号
構文	DirectoryString
例	nsslapd-pluginVersion: 11.3

## 4.3. 特定のプラグインで使用できる属性

### 4.3.1. nsslapd-dynamic-plugins

Directory Server には、サーバーを再起動せずに有効にできる動的プラグインがあります。**nsslapd-dynamic-plugins** 属性は、サーバーが動的プラグインを許可するように設定されているかどうかを指定します。デフォルトでは、動的プラグインは無効になっています。



#### 警告

Directory Server は動的プラグインをサポートしません。これは、テストおよびデバッグの目的でのみ使用してください。

一部のプラグインは動的として設定できず、サーバーを再起動する必要があります。

プラグインパラメーター	説明
エントリー DN	cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-dynamic-plugins: on

### 4.3.2. nsslapd-pluginConfigArea

一部のプラグインエントリーはコンテナエントリーで、プラグインのインスタンスが複数、**cn=plugins,cn=config** のこのコンテナの下に作成されます。ただし、**cn=plugins,cn=config** は複製されないため、これらのコンテナエントリーの下にあるプラグイン設定は、すべての Directory Server インスタンスで手動で設定する必要があります。

**nsslapd-pluginConfigArea** 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領域にある別のコンテナエントリーを参照します。このコンテナエントリーは、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。

プラグインパラメーター	説明
エントリー DN	cn= <b>plug-in name</b> ,cn=plugins,cn=config
有効な値	任意の有効な DN
デフォルト値	
構文	DN
例	nsslapd-pluginConfigArea: cn=managed entries container,ou=containers,dc=example,dc=com

### 4.3.3. nsslapd-pluginLoadNow

この属性は、プラグインが使用するすべてのシンボル、これらのシンボルによるすべてのシンボル参照を即時に読み込む (**true**) か、初回時に使用されるシンボルを読み込む (**false**) かを指定します。

プラグインパラメーター	説明
エントリー DN	cn=plug-in name,cn=plugins,cn=config

プラグインパラメーター	説明
有効な値	true   false
デフォルト値	false
構文	DirectoryString
例	nsslapd-pluginLoadNow: false

#### 4.3.4. nsslapd-pluginLoadGlobal

この属性は、依存ライブラリーのシンボルがローカルで表示される (**false**) か、実行可能ファイルおよびすべての共有オブジェクトに (**true**) 表示されるかどうかを指定します。

プラグインパラメーター	説明
エントリー DN	cn=plug-in name,cn=plugins,cn=config
有効な値	true   false
デフォルト値	false
構文	DirectoryString
例	nsslapd-pluginLoadGlobal: false

#### 4.3.5. nsslapd-plugin-depends-on-type

プラグインが正しい順序でサーバーによって呼び出されるようにするために使用される多値属性。 **nsslapd-pluginType** 属性に含まれるプラグインのタイプ番号に対応する値を取ります。詳細は「[nsslapd-pluginType](#)」を参照してください。以下の有効な範囲内で、この値のいずれかに一致する type 値を持つプラグインはすべて、このプラグインの前にサーバーによって起動します。以下の postoperation Referential Integrity プラグインの例は、postoperation Referential Integrity プラグインの前にデータベースプラグインを起動することを示します。

プラグインパラメーター	説明
エントリー DN	cn=referential integrity postoperation,cn=plugins,cn=config
有効な値	database
デフォルト値	
構文	DirectoryString



プラグインパラメーター	説明
例	nsslapd-plugin-depends-on-type: database

#### 4.3.6. nsslapd-plugin-depends-on-named

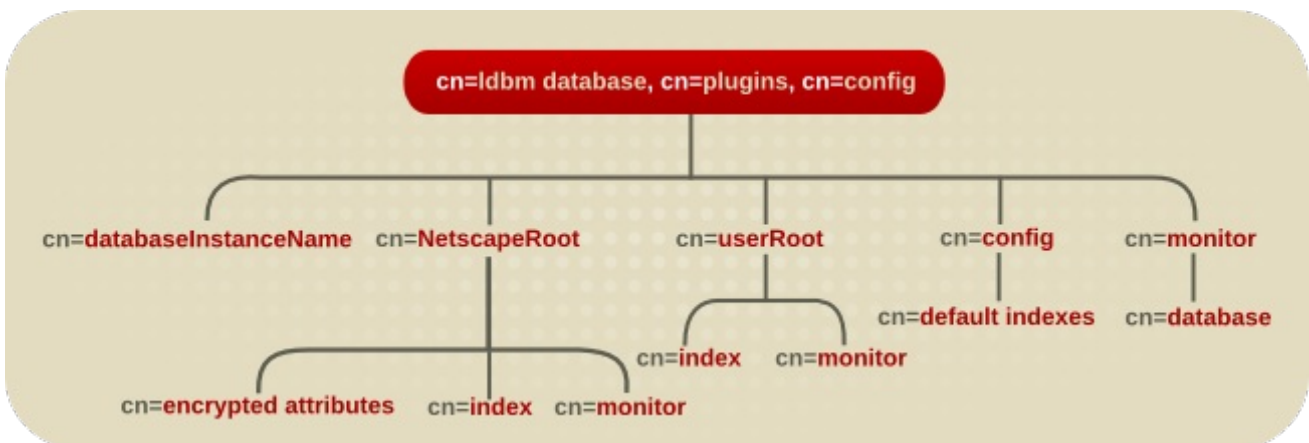
プラグインが正しい順序でサーバーによって呼び出されるようにするために使用される多値属性。プラグインの **cn** 値に対応する値を取ります。**cn** 値が、以下の値のいずれかに一致するプラグインは、このプラグインの前にサーバーにより起動されます。プラグインが存在しない場合は、サーバーが起動できません。以下の postoperation Referential Integrity プラグインの例は、Views プラグインが Roles の前に起動することを示しています。ビューが見つからない場合は、サーバーは起動しません。

プラグインパラメーター	説明
エントリー DN	cn=referential integrity postoperation,cn=plugins,cn=config
有効な値	サービスのクラス
デフォルト値	
構文	DirectoryString
例	* nsslapd-plugin-depends-on-named: Views  * nsslapd-pluginId: roles

#### 4.4. データベースプラグインの属性

また、データベースプラグインは 図4.1「データベースプラグイン」 に示されるように、情報ツリーにまとめられます。

図4.1データベースプラグイン



データベースインスタンスで使用されるすべてのプラグインテクノロジーは、**cn=ldbm database** プラグインノードに保存されます。このセクションでは、**cn=ldbm database,cn=plugins,cn=config** 情報ツリーの各ノードの追加の属性情報を太字で示します。

#### 4.4.1. cn=config,cn=ldbm database,cn=plugins,cn=config 下のデータベース属性

このセクションでは、すべてのインスタンスに共通するグローバル設定属性を **cn=config,cn=ldbm database,cn=plugins,cn=config** ツリーノードに格納します。

##### 4.4.1.1. nsslapd-backend-implement

**nsslapd-backend-implementations** パラメーターは、Directory Server が使用するデータベースバックエンドを定義します。



#### 重要

Directory Server は現在 Berkeley Database(BDB) のみをサポートしています。したがって、このパラメーターを別の値に設定できません。

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	bdb
デフォルト値	bdb
構文	ディレクトリー文字列
例	nsslapd-backend-implement: bdb

##### 4.4.1.2. nsslapd-backend-opt-level

このパラメーターは、実験的なコードをトリガーして書き込みパフォーマンスを向上できます。

値:

- **0:** パラメーターを無効にします。
- **1:** トランザクション中にレプリケーション更新ベクターがデータベースに書き込まれません
- **2:** バックエンドロックの取得順序を変更し、トランザクションを開始します。
- **4:** トランザクションからコードを移動します。

すべてのパラメーターを組み合わせたことができます。たとえば、**7** の場合は、すべての最適な機能を有効にします。



### 警告

このパラメーターは実験的なものです。Red Hat サポートから特に指示されない限り、値を変更しないでください。

パラメーター	説明
エントリー DN	cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	0   1   2   4
デフォルト値	0
構文	整数
例	nsslapd-backend-opt-level: 0

#### 4.4.1.3. nsslapd-directory

この属性は、データベースインスタンスへの絶対パスを指定します。データベースインスタンスを手動で作成する場合は、この属性を含める必要があります。これは、Directory Server コンソールでデフォルトで設定されています (変更可能)。データベースインスタンスを作成したら、このパスを変更しないでください。変更すると、サーバーがデータにアクセスできなくなるリスクがあります。

パラメーター	説明
エントリー DN	cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	データベースインスタンスへの有効な絶対パス
デフォルト値	
構文	DirectoryString
例	nsslapd-directory: /var/lib/dirsrv/slapd- <b>instance</b> /db

#### 4.4.1.4. nsslapd-exclude-from-export

この属性には、データベースのエクスポート時にエントリーから除外する属性の名前のスペース区切りのリストが含まれています。これは主に、サーバーインスタンス固有の設定および運用属性に使用されます。

サーバーのパフォーマンスに影響する可能性があるため、この属性のデフォルト値を削除しないでください。

パラメーター	説明
エントリー DN	cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	有効な属性
デフォルト値	entrydn entryid dncomp parentid numSubordinates entryusn
構文	DirectoryString
例	nsslapd-exclude-from-export: entrydn entryid dncomp parentid numSubordinates entryusn

#### 4.4.1.5. nsslapd-db-transaction-wait

**nsslapd-db-transaction-wait** パラメーターを有効にすると、Directory Server はトランザクションを開始せずに、ロックリソースが利用可能になるまで待機します。

パラメーター	説明
エントリー DN	cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-db-transaction-wait

#### 4.4.1.6. nsslapd-db-private-import-mem

**nsslapd-db-private-import-mem** パラメーターは、DirectoryServer がデータベースインポート用のリージョンとミュutexの割り当てにプライベートメモリーを使用するかどうかを管理します。

パラメーター	説明
エントリー DN	cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString

パラメーター	説明
例	nsslapd-db-private-import-mem: on

#### 4.4.1.7. nsslapd-db-deadlock-policy

**nsslapd-db-deadlock-policy** パラメーターは、**libdb** library-internal deadlock ポリシーを設定します。



#### 重要

このパラメーターは、Red Hat サポートから指示された場合にのみ変更します。

パラメーター	説明
エントリー DN	cn=config,cn=ldb database,cn=plugins,cn=config
有効な値	0-9
デフォルト値	0
構文	DirectoryString
例	nsslapd-db-deadlock-policy: 9

#### 4.4.1.8. nsslapd-idl-switch

**nsslapd-idl-switch** パラメーターは、Directory Server が使用する IDL 形式を設定します。Red Hat では、以前の IDL 形式に対応しなくなった点に注意してください。

パラメーター	説明
エントリー DN	cn=config,cn=ldb database,cn=plugins,cn=config
有効な値	new   old
デフォルト値	new
構文	ディレクトリー文字列
例	nsslapd-idl-switch: new

#### 4.4.1.9. nsslapd-idlistscanlimit

デフォルトでは、このパフォーマンス関連の属性は、検索操作中に特定されるエントリー ID の数を指定します。数値ではない場合、または 32 ビットの符号付き整数には大きすぎる値を設定しようとする、**LDAP\_UNWILLING\_TO\_PERFORM** エラーメッセージが返され、問題を説明する追加のエラー情

報が示されます。検索パフォーマンスを向上させるには、デフォルト値を保持することを推奨します。

詳細は、以下の該当するセクションを参照してください。

- [Directory Server パフォーマンスチューニングガイド](#)
- [Directory Server 管理ガイド](#)

このパラメーターはサーバーの実行中に変更でき、新しい値は後続の検索に影響します。

対応するユーザーレベルの属性は **nsIDListScanLimit** です。

パラメーター	説明
エントリー DN	cn=config,cn=ldb database,cn=plugins,cn=config
Valid Range	100 から最大 32 ビットの整数値 (2147483647) のエントリー ID
デフォルト値	4000
Syntax	整数
Example	nsslapd-idlistscanlimit: 4000

#### 4.4.1.10. nsslapd-lookthroughlimit

このパフォーマンス関連の属性は、検索要求に応答して候補エントリーを調べるときに DirectoryServer がチェックするエントリーの最大数を指定します。ただし、Directory Manager DN は、デフォルトでは無制限で、ここで指定したその他の設定を上書きします。この制限では、バインドベースのリソース制限が機能する点に注意する必要があります。つまり、ユーザーバインドするエントリーに操作属性 **nsLookThroughLimit** の値が存在する場合は、デフォルトの制限が上書きされます。数値ではない場合、または 32 ビットの符号付き整数には大きすぎる値を設定しようとする **LDAP\_UNWILLING\_TO\_PERFORM** エラーメッセージが返され、問題を説明する追加のエラー情報が示されます。

パラメーター	説明
エントリー DN	cn=config,cn=ldb database,cn=plugins,cn=config
Valid Range	エントリーの -1 から最大 32 ビット整数までです (-1 は無制限)。
デフォルト値	5000
構文	整数
例	nsslapd-lookthroughlimit: 5000

#### 4.4.1.11. nsslapd-mode

この属性は、新しく作成されたインデックスファイルに使用されるパーミッションを指定します。

パラメーター	説明
エントリー DN	cn=config,cn=ldb database,cn=plugins,cn=config
有効な値	4桁の8進数。ただし、モード <b>0600</b> が推奨されます。これにより、インデックスファイルの所有者 ( <b>ns-slapd</b> の実行ユーザー) の読み取り/書き込みアクセスが許可され、他のユーザーにはアクセスできなくなります。
デフォルト値	600
構文	整数
例	nsslapd-mode: 0600

#### 4.4.1.12. nsslapd-pagedidlistscanlimit

このパフォーマンス関連の属性は、簡単なページ結果制御を使用して検索操作で特定されるエントリー ID の数を指定します。

この属性は **nsslapd-idlistscanlimit** 属性と同じように機能しますが、単純なページ結果制御による検索にのみ適用される点が異なります。

この属性が存在しないか、ゼロに設定されている場合は、**nsslapd-idlistscanlimit** を使用してページングされた検索およびページ以外の検索を行います。

対応するユーザーレベルの属性は **nsPagedIDListScanLimit** です。

パラメーター	説明
エントリー DN	cn=config,cn=ldb database,cn=plugins,cn=config
Valid Range	エントリーの -1 から最大 32 ビット整数までです (-1 は無制限)。
デフォルト値	0
構文	整数
例	nsslapd-pagedidlistscanlimit: 5000

#### 4.4.1.13. nsslapd-pagedlookthroughlimit

このパフォーマンス関連の属性は、単純なページ結果制御を使用する検索の候補エントリーを調べる時に Directory Server がチェックするエントリーの最大数を指定します。

この属性は **nsslapd-lookthroughlimit** 属性と同じように機能しますが、単純なページ結果制御の検索にのみ適用される点が異なります。

この属性が存在しないか、ゼロに設定されている場合は、**nsslapd-lookthroughlimit** を使用して、ページングされた検索と、ページングされていない検索の両方を行います。

対応するユーザーレベルの属性は **nsPagedLookThroughLimit** です。

パラメーター	説明
エントリー DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	エントリーの -1 から最大 32 ビット整数までです (-1 は無制限)。
デフォルト値	0
構文	整数
例	nsslapd-pagedlookthroughlimit: 25000

#### 4.4.1.14. nsslapd-rangelookthroughlimit

このパフォーマンス関連の属性は、範囲検索リクエストへの応答として候補のエントリーを調べるときに Directory Server がチェックするエントリーの最大数を指定します。

範囲検索は演算子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ります。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。

**nsslapd-rangelookthroughlimit** 属性は、Directory Manager を含むすべてのユーザーに適用される個別の範囲のルックスルー制限を設定します。

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。



#### 注記

その他のリソース制限とは異なり、Directory Manager、通常ユーザー、およびその他の LDAP クライアントなどのユーザーによる検索に適用されます。



パラメーター	説明
エントリー DN	cn=config,cn=ldb database,cn=plugins,cn=config
Valid Range	エントリーの -1 から最大 32 ビット整数までです (-1 は無制限)。
デフォルト値	5000
構文	整数
例	nsslapd-rangelookthroughlimit: 5000

#### 4.4.1.15. nsslapd-search-bypass-filter-test

**nsslapd-search-bypass-filter-test** パラメーターを有効にすると、Directory Server は、検索時に候補リストをビルドするタイミングでフィルターチェックを回避します。パラメーターを **verify** に設定すると、Directory Server は検索候補エントリーに対してフィルターを評価します。

パラメーター	説明
エントリー DN	cn=config,cn=ldb database,cn=plugins,cn=config
有効な値	on   off   verify
デフォルト値	on
構文	ディレクトリー文字列
例	nsslapd-search-bypass-filter-test: on

#### 4.4.1.16. nsslapd-search-use-vlv-index

**nsslapd-search-use-vlv-index** は、仮想リストビュー (VLV) 検索を有効または無効にします。

パラメーター	説明
エントリー DN	cn=config,cn=ldb database,cn=plugins,cn=config
有効な値	on   off
デフォルト値	on
構文	ディレクトリー文字列
例	nsslapd-search-use-vlv-index: on

#### 4.4.1.17. nsslapd-subtree-rename-switch

すべてのディレクトリーエントリーは、エントリーインデックスファイルのキーとして保存されます。インデックスキーは、現在のエントリー DN をインデックスのメタエントリーにマッピングします。このマッピングは、エントリーの RDN またはエントリーの完全な DN で行います。

サブツリーエントリーの名前変更が許可されている場合 (つまり、子エントリーのあるエントリーは、サブツリー全体の名前を事実上変更) には、そのエントリーは **entryrdn.db** インデックスに格納され、DN ではなく割り当てられた ID によって親エントリーと子エントリーを関連付けます。サブツリーの名前変更操作が許可されていない場合は、**entryrdn.db** インデックスが無効になり、**entryrdn.db** インデックスが使用されます。これは、暗黙的な親子関係がある完全な DN を使用するだけです。

パラメーター	説明
エントリー DN	cn=config,cn=ldb database,cn=plugins,cn=config
有効な値	off   on
デフォルト値	on
構文	DirectoryString
例	nsslapd-subtree-rename-switch: on

#### 4.4.2. cn=bdb,cn=config,cn=ldb database,cn=plugins,cn=config 下のデータベース属性

このセクションでは、すべてのインスタンスに共通するグローバル設定属性を **cn=bdb,cn=config,cn=ldb database,cn=plugins,cn=config** ツリーノードに格納します。

##### 4.4.2.1. nsslapd-cache-autosize

このパフォーマンスチューニング関連の属性は、データベースおよびエントリーキャッシュの合計で使用される空きメモリーの割合を設定します。たとえば、値を **10** に設定する場合には、システムの空きメモリーの 10% が両方のキャッシュに使用されます。この値を **0** よりも大きい値に設定すると、データベースおよびエントリーキャッシュに対して自動サイズ設定が有効になります。

Red Hat は、パフォーマンスの最適化を図るため、自動サイジングを無効にしないことを推奨します。ただし、特定の状況では、自動サイジングを無効にする必要がある場合があります。この場合は **nsslapd-cache-autosize** 属性を **0** に設定し、手動で設定します。

- **nsslapd-dbcachesize** 属性のデータベースキャッシュ。
- **nsslapd-cachememsize** 属性のエントリーキャッシュ。

サイズ調整の詳細は、[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。



## 注記

**nsslapd-cache-autosize** および **nsslapd-cache-autosize-split** 属性が **100** などの高い値に設定されていると、Directory Server が起動に失敗します。この問題を修正するには、両方のパラメーターをより妥当な値に設定します。以下に例を示します。

```
nsslapd-cache-autosize: 10
nsslapd-cache-autosize-split: 40
```

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	0 から 1000 を設定すると、代わりにデフォルト値が使用されます。
デフォルト値	10
構文	整数
例	nsslapd-cache-autosize: 10

### 4.4.2.2. nsslapd-cache-autosize-split

このパフォーマンスチューニング関連の属性は、データベースキャッシュに使用されるメモリーの割合を設定します。残りのメモリーはエントリーキャッシュに使用されます。たとえば、値が **40** に設定されている場合には、データベースキャッシュは 40% を使用して、エントリーは、**nsslapd-cache-autosize** 属性で予約されている空きメモリーの残り 60% をキャッシュします。

サイズ調整の詳細は、[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。



## 注記

**nsslapd-cache-autosize** および **nsslapd-cache-autosize-split** 属性が **100** などの高い値に設定されていると、Directory Server が起動に失敗します。この問題を修正するには、両方のパラメーターをより妥当な値に設定します。以下に例を示します。

```
nsslapd-cache-autosize: 10
nsslapd-cache-autosize-split: 40
```

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config

パラメーター	説明
Valid Range	0 から 990 を設定すると、代わりにデフォルト値が使用されます。
デフォルト値	40
構文	整数
例	nsslapd-cache-autosize-split: 40

#### 4.4.2.3. nsslapd-db-checkpoint-interval

これは、Directory Server がチェックポイントエントリーをデータベーストランザクションログに送信するまでの時間を秒単位で設定します。データベーストランザクションログには、最近の全データベース操作の連続リストが含まれ、データベースリカバリーのみで使用されます。チェックポイントエントリーは、どのデータベース操作がディレクトリーデータベースに物理的に書き込まれたかを示します。チェックポイントエントリーは、データベーストランザクションログのどこでシステム障害後にリカバリーを開始するかを決定するために使用されます。**nsslapd-db-checkpoint-interval** 属性は **dse.ldif** に存在しません。チェックポイントの間隔を変更するには、属性を **dse.ldif** に追加します。この属性は **ldapmodify** を使用して動的に変更できます。この属性の変更に関する詳細は、**Red Hat Directory Server 管理ガイド**の Directory Server パフォーマンスの調整の章を参照してください。

この属性は、システムの変更/診断のためにのみ提供されており、Red Hat テクニカルサポートまたは Red Hat Consulting のガイダンスがある場合にのみ変更する必要があります。この属性およびその他の設定属性の設定に一貫性がないと、Directory Server が不安定になる可能性があります。

データベーストランザクションのロギングの詳細は、**Red Hat Directory Server 管理ガイド**のサーバーおよびデータベースアクティビティの監視の章を参照してください。

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	10 から 300 秒
デフォルト値	60
構文	整数
例	nsslapd-db-checkpoint-interval: 120

#### 4.4.2.4. nsslapd-db-circular-logging

この属性は、トランザクションログファイルの循環ロギングを指定します。この属性をオフにすると、以前のトランザクションログファイルが削除されず、以前のログトランザクションファイルとして名前が変更されたままになります。循環ロギングをオフにすると、サーバーのパフォーマンスが大幅に低下する可能性があるため、Red Hat テクニカルサポートまたはコンサルティングの指示がある場合以外は変更しないでください。

パラメーター	説明
エントリー DN	cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-db-circular-logging: on

#### 4.4.2.5. nsslapd-db-compactdb-interval

**nsslapd-db-compactdb-interval** 属性は、Directory Server がデータベースおよびレプリケーション変更ログを圧縮する際の間隔を秒単位で定義します。compact 操作は未使用のページをファイルシステムに戻し、データベースファイルサイズを縮小します。データベースの圧縮はリソースを大量に使用するため、頻繁に行うべきではない点に注意してください。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	0 (圧縮なし) から 2147483647 秒
デフォルト値	2592000 (30 日)
構文	整数
例	nsslapd-db-compactdb-interval: 2592000

#### 4.4.2.6. nsslapd-db-compactdb-time

**nsslapd-db-compactdb-time** 属性は、Directory Server がすべてのデータベースとそのレプリケーション変更ログを圧縮する時間を設定します。圧縮タスクは、圧縮間隔 (**nsslapd-db-compactdb-interval**) を超えた後に実行します。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config

パラメーター	説明
有効な値	HH:MM.時間は 24 時間形式で設定
デフォルト値	23:59
構文	DirectoryString
例	nsslapd-db-compactdb-time: 23:59

#### 4.4.2.7. nsslapd-db-debug

この属性は、追加のエラー情報を Directory Server に報告するかどうかを指定します。エラー情報を報告するには、パラメーターを **on** に設定します。このパラメーターはトラブルシューティングを目的としており、パラメーターを有効にすると Directory Server の速度が低下する可能性があります。

パラメーター	説明
エントリー DN	cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-db-debug: off

#### 4.4.2.8. nsslapd-db-durable-transactions

この属性は、データベースのトランザクションログエントリーをすぐにディスクに書き込むかどうかを設定します。データベーストランザクションログには、最近の全データベース操作の連続リストが含まれ、データベースリカバリーのみに使用されます。永続トランザクションを有効にすると、すべてのディレクトリーの変更は常にログファイルに物理的に記録されるため、システムに障害が発生した場合に復元できます。ただし、永続トランザクション機能は、Directory Server のパフォーマンスも低下させる可能性があります。永続トランザクションが無効の場合には、すべてのトランザクションはデータベーストランザクションログに論理的に書き込まれますが、すぐにディスクに物理的に書き込まれない可能性があります。ディレクトリーの変更をディスクに物理的に書き込む前にシステムに障害が発生した場合には、その変更は復元できません。 **nsslapd-db-durable-transactions** 属性は **dse.ldif** に存在しません。永続トランザクションを無効にするには、属性を **dse.ldif** に追加します。

この属性は、システムの変更/診断のためにのみ提供されており、Red Hat テクニカルサポートまたは Red Hat Consulting のガイダンスがある場合にのみ変更する必要があります。この属性およびその他の設定属性の設定に一貫性がないと、Directory Server が不安定になる可能性があります。

データベーストランザクションのロギングの詳細は、**Red Hat Directory Server 管理ガイド**のサーバーおよびデータベースアクティビティの監視の章を参照してください。

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-db-durable-transactions: on

#### 4.4.2.9. nsslapd-db-home-directory

パフォーマンス上の理由からデータベースを別の物理的な場所に移動するには、このパラメーターを使用してホームディレクトリーを指定します。

この状況は、データベースキャッシュサイズ、物理メモリーのサイズ、およびカーネルチューニング属性の特定の組み合わせでのみ発生します。特に、データベースキャッシュのサイズが100メガバイト未満の場合には、この状況は発生しません。

- ディスクは頻繁に使用される (1秒あたり1メガバイト以上のデータ転送)。
- サービス時間が長い (100ms 以上)。
- ほとんどが書き込みアクティビティーである。

これらがすべて該当する場合は、**nsslapd-db-home-directory** 属性を使用して **tempfs** タイプのファイルシステムのサブディレクトリーを指定します。

**nsslapd-db-home-directory** 属性で参照されるディレクトリーは、tempfs タイプのファイルシステムのサブディレクトリー (**/tmp** など) である必要があります。ただし、Directory Server では、この属性で参照されるサブディレクトリーは作成されません。このディレクトリーは、手動またはスクリプトを使用して作成する必要があります。**nsslapd-db-home-directory** 属性で参照されるディレクトリーの作成に失敗すると、Directory Server が起動できなくなります。

また、同じマシンに複数の Directory Server がある場合は、**nsslapd-db-home-directory** 属性を異なるディレクトリーで設定する必要があります。これを実行しないと、両方のディレクトリーのデータベースが破損します。

この属性を使用すると、内部の Directory Server データベースファイルが属性によって参照されるディレクトリーに移動します。十分なメモリーを割り当てることができないため、ファイルを移動後にサーバーが起動しないことがあります。これは、サーバーに設定されている大容量のデータベースキャッシュサイズが原因です。その場合は、サーバーが再起動する値まで、データベースキャッシュサイズを減らします。

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config

パラメーター	説明
有効な値	tempfs ファイルシステム内の有効なディレクトリー名 ( <b>/tmp</b> など)
デフォルト値	
構文	DirectoryString
例	nsslapd-db-home-directory: /tmp/slapd-phonebook

#### 4.4.2.10. nsslapd-db-idl-divisor

この属性は、データベースページごとのブロック数の観点から、インデックスブロックサイズを指定します。ブロックサイズは、データベースページサイズをこの属性の値で除算して計算します。値が **1** の場合は、ブロックサイズがページサイズとちょうど等しくなります。デフォルト値の **0** は、ブロックサイズをページサイズから内部データベースオーバーヘッドの推定許容値を引いたものに設定します。ほとんどのインストールでは、特定のチューニングが必要にならない限り、デフォルト値を変更しないでください。

この属性の値を変更する前に、**db2ldif** スクリプトを使用してすべてのデータベースをエクスポートします。変更が完了したら、**ldif2db** スクリプトを使用してデータベースを再読み込みします。



#### 警告

このパラメーターは、非常にスキルの高いユーザーのみが使用するようになっています。

パラメーター	説明
エントリー DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	0 から 8
デフォルト値	0
構文	整数
例	nsslapd-db-idl-divisor: 2

#### 4.4.2.11. nsslapd-db-locks

Directory Server のロックメカニズムは、Directory Server プロセスのコピーを同時に実行できる数を制御します。**nsslapd-db-locks** パラメーターは、ロックの最大数を設定します。



Directory Server がロックを使い果たして、**libdb: Lock table is out of available locks** のエラーメッセージがログに記録される場合にのみ、このパラメーターをより高い値に設定します。必要なしに高い値を設定すると、`/var/lib/dirsrv/slapd-instance_name/db__db.*` ファイルのサイズが増えるだけ、利点はありません。ログの監視および現実的な値の決定に関する詳細は、[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。

この属性への変更を反映するには、サービスを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	0 - 2147483647
デフォルト値	10000
構文	整数
例	nsslapd-db-locks: 10000

#### 4.4.2.12. nsslapd-db-locks-monitoring-enable

データベースロックが不足すると、データが破損する可能性があります。**nsslapd-db-locks-monitoring-enable** パラメーターを使用すると、データベースロックの監視を有効または無効にできます。パラメーターが有効になっている場合 (デフォルト)、アクティブなデータベースロックの数が [nsslapd-db-locks-monitoring-threshold](#) で設定されているパーセンテージのしきい値よりも大きい場合、DirectoryServer はすべての検索を中止します。問題が発生した場合には、管理者は [nsslapd-db-locks](#) パラメーターのデータベースロックの数を増やすことができます。

この属性への変更を有効にするには、サービスを再起動します。

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsslapd-db-locks-monitoring-enable: on

#### 4.4.2.13. nsslapd-db-locks-monitoring-pause

`nsslapd-db-locks-monitoring-enable` パラメーターでデータベースロックの監視が有効である場合には、`nsslapd-db-locks-monitoring-pause` は、次のチェックを行うまでに監視スレッドがスリープする間隔をミリ秒単位で定義します。

このパラメーターに設定する値が大きすぎると、監視チェックを行う前に、サーバーがデータベースロックを使い果たす可能性があります。ただし、値が低すぎると、サーバーの速度が遅くなる可能性があります。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	0 - 2147483647 (ミリ秒単位)
デフォルト値	500
構文	DirectoryString
例	nsslapd-db-locks-monitoring-pause: 500

#### 4.4.2.14. nsslapd-db-locks-monitoring-threshold

`nsslapd-db-locks-monitoring-enable` パラメーターでデータベースロックの監視が有効になっている場合には、`nsslapd-db-locks-monitoring-threshold` は、Directory Server が検索を終了する前にデータベースロックの最大使用率を設定し、ロックの枯渇を回避します。

この属性への変更を有効にするには、サービスを再起動します。

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	70 - 95
デフォルト値	90
構文	DirectoryString
例	nsslapd-db-locks-monitoring-threshold: 90

#### 4.4.2.15. nsslapd-db-logbuf-size

この属性は、ログ情報のバッファサイズを指定します。ログ情報は、バッファがいっぱいになるか、トランザクションコミットで、バッファがディスクに書き込まれるまでメモリーに保存されます。バッファサイズを大きくすると、トランザクションの実行時間が長い場合、同時アプリケーション

ンが多い場合、または大量のデータを生成するトランザクションが存在する場合に、スループットが大幅に向上します。ログ情報のバッファサイズは、トランザクションログのサイズを4で割ったものです。

**nsslapd-db-logbuf-size** 属性は、**nsslapd-db-durable-transactions** 属性が **on** に設定されている場合にのみ有効です。

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	32K から最大 32 ビット整数 (マシンで利用可能なメモリー容量に制限されます)
デフォルト値	32K
構文	整数
例	nsslapd-db-logbuf-size: 32K

#### 4.4.2.16. nsslapd-db-logdirectory

この属性は、データベーストランザクションログが含まれるディレクトリーへのパスを指定します。データベーストランザクションログには、最近のすべてのデータベース操作の連続リストが含まれます。Directory Server はこの情報を使用して、インスタンスが予期せずシャットダウンした後にデータベースを復元します。

デフォルトでは、データベーストランザクションログはディレクトリーデータベースと同じディレクトリーに保存されます。このパラメーターを更新するには、**/etc/dirsrv/slapd-instance\_name/dse.ldif** ファイルを手動で更新する必要があります。詳細は、[Red Hat Directory Server 管理ガイド](#) のトランザクションログディレクトリーの変更セクションを参照してください。

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	有効なパス
デフォルト値	
構文	DirectoryString
例	nsslapd-db-logdirectory: /var/lib/dirsrv/slapd-instance_name/db/

#### 4.4.2.17. nsslapd-db-logfile-size

この属性は、ログ内の単一ファイルの最大サイズをバイト単位で指定します。デフォルト、または値が **0** に設定されている場合には、最大 10 メガバイトが使用されます。最大サイズは符号なし 4 バイト値です。

パラメーター	説明
エントリー DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	0 から署名なしの 4 バイトの整数
デフォルト値	10MB
構文	整数
例	nsslapd-db-logfile-size: 10 MB

#### 4.4.2.18. nsslapd-db-page-size

この属性は、データベースのアイテムの保持に使用されるページのサイズをバイト単位で指定します。最小サイズは 512 バイトで、最大サイズは 64 キロバイトです。ページサイズが明示的に設定されていない場合には、Directory Server はデフォルトでページサイズ 8 キロバイトに設定されます。このデフォルト値を変更すると、パフォーマンスに大きな影響を及ぼす可能性があります。ページサイズが小さすぎると、ページの分割やコピーが大量に発生しますが、ページサイズが大きすぎると、ディスク領域が無駄になる可能性があります。

この属性の値を変更する前に、**db2ldif** スクリプトを使用してすべてのデータベースをエクスポートします。変更が完了したら、**ldif2db** スクリプトを使用してデータベースを再読み込みします。

パラメーター	説明
エントリー DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	512 バイトから 64 キロバイト
デフォルト値	8KB
構文	整数
例	nsslapd-db-page-size: 8KB

#### 4.4.2.19. nsslapd-db-spin-count

この属性は、test-and-set ミューテックスがブロックなしにスピンする回数を指定します。

**警告**

Berkeley DB の内部に精通しているか、Red Hat サポートにより具体的に依頼された場合を除き、この値は **変更しない** ください。

デフォルト値の **0** を指定すると、BDB は、利用可能な CPU コア数 (**nproc** ユーティリティーまたは **sysconf(\_SC\_NPROCESSORS\_ONLN)** 呼び出しで報告される) に **50** をかけて実際の値を計算します。たとえば、8 つの論理コアを備えたプロセッサでは、この属性を **0** に設定したままにすることは、**400** に設定することと同じです。スピンを完全にオフにすることはできません。test-and-set ミューテックスをブロックせずにスピンする回数を最小限に抑える場合は、この属性を **1** に設定します。

パラメーター	説明
エントリー DN	cn=config,cn=ldb database,cn=plugins,cn=config
Valid Range	0 から 2147483647 (2 <sup>31</sup> -1)
デフォルト値	0
構文	整数
例	nsslapd-db-spin-count: 0

#### 4.4.2.20. nsslapd-db-transaction-batch-max-wait

「[nsslapd-db-transaction-batch-val](#)」を設定すると、set batch 値に達したときに、トランザクションのフラッシュが別のスレッドによって行われます。ただし、更新が少ない場合は、このプロセスに時間がかかる場合があります。このパラメーターは、バッチ数とは関係なく、トランザクションを最新にフラッシュするタイミングを制御します。値はミリ秒単位で定義されます。

**警告**

このパラメーターは実験的なものです。Red Hat サポートから特に指示されない限り、値を変更 **しない** ください。

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldb database,cn=plugins,cn=config

パラメーター	説明
Valid Range	0 - 2147483647 (ミリ秒単位)
デフォルト値	50
構文	整数
例	nsslapd-db-transaction-batch-max-wait: 50

#### 4.4.2.21. nsslapd-db-transaction-batch-min-wait

「[nsslapd-db-transaction-batch-val](#)」を設定すると、set batch 値に達したときに、トランザクションのフラッシュが別のスレッドによって行われます。ただし、更新が少ない場合は、このプロセスに時間がかかる場合があります。このパラメーターは、トランザクションをバッチ数とは関係なく、最も早くフラッシュするタイミングを制御します。値はミリ秒単位で定義されます。



#### 警告

このパラメーターは実験的なものです。Red Hat サポートから特に指示されない限り、値を変更しないでください。

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	0 - 2147483647 (ミリ秒単位)
デフォルト値	50
構文	整数
例	nsslapd-db-transaction-batch-min-wait: 50

#### 4.4.2.22. nsslapd-db-transaction-batch-val

この属性は、コミット前にバッチ処理されるトランザクションの数を指定します。この属性は、完全なトランザクションの持続性が不要な場合には、更新のパフォーマンスを向上できます。この属性は **ldapmodify** を使用して動的に変更できます。この属性の変更に関する詳細は、**Red Hat Directory Server 管理ガイド**の Directory Server パフォーマンスの調整の章を参照してください。

**警告**

この値を設定すると、データの一貫性が低下し、データが失われる可能性があります。これは、サーバーがバッチ処理されたトランザクションをフラッシュする前に停電が発生した場合に、バッチ内のそれらのトランザクションが失われるためです。

Red Hat サポートから特に依頼されない限り、この値は設定しないでください。

この属性が定義されていないか、**0** に設定されている場合には、トランザクションバッチ処理はオフになり、LDAP を使用してこの属性にリモートで変更を加えることはできません。ただし、この属性を **0** より大きい値に設定すると、キューに置かれたトランザクションの数が属性値と同じになるまでトランザクションのコミットが遅延します。**0** より大きい値を指定すると、LDAP を使用してこの属性をリモートで変更できます。この属性の値が **1** の場合、LDAP を使用してリモートで属性設定を変更できませんが、バッチ処理は行われません。そのため、サーバーの起動時に **1** を指定すると、必要に応じて、リモートでトランザクションバッチのオンとオフを切り替えることができる一方で、通常の持続性を維持することができます。この属性の値では、**nsslapd-db-logbuf-size** 属性を変更して、バッチ処理されたトランザクションに対応するのに十分なログバッファサイズを確保しなければならない可能性がある点に注意してください。

**注記**

**nsslapd-db-transaction-batch-val** 属性は、**nsslapd-db-durable-transaction** 属性が **on** に設定されている場合にのみ有効です。

データベーストランザクションのロギングの詳細は、**Red Hat Directory Server 管理ガイド**のサーバーおよびデータベースアクティビティの監視の章を参照してください。

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	0 から 30
デフォルト値	0 (またはオフ)
構文	整数
例	nsslapd-db-transaction-batch-val: 5

**4.4.2.23. nsslapd-db-trickle-percentage**

この属性は、少なくとも共有メモリープールに指定したページの割合が、バッキングファイルにダーティーページを書き込むことで消去されるように設定します。これは、書き込みを待たずに、新しい情報の読み取りにページが常に利用できるようにするためです。

パラメーター	説明
エントリー DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	0 から 100
デフォルト値	40
構文	整数
例	nsslapd-db-trickle-percentage: 40

#### 4.4.2.24. nsslapd-db-verbose

この属性は、チェックポイントのログの検索、デッドロックの検出の実行、およびリカバリーの実行時に追加の情報およびデバッグメッセージを記録するかどうかを指定します。このパラメーターはトラブルシューティングを目的としており、パラメーターを有効にすると Directory Server の速度が低下する可能性があります。

パラメーター	説明
エントリー DN	cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-db-verbose: off

#### 4.4.2.25. nsslapd-import-cache-autosize

このパフォーマンスチューニング関連の属性は、LDIF ファイルのデータベースへのコマンドラインベースのインポートプロセス (**ldif2db** 操作) 中に使用されるインポートキャッシュ (**importCache**) のサイズを自動的に設定します。

Directory Server では、インポート操作はサーバタスクとして実行することも、コマンドラインでのみ実行できます。タスクモードでは、インポート操作は一般的な Directory Server 操作として実行されます。**nsslapd-import-cache-autosize** 属性を使用すると、インポート操作がコマンドラインで実行される場合に、インポートキャッシュを事前に決定したサイズに自動設定できます。また、この属性はタスクモードのインポート時に Directory Server で使用して、インポートキャッシュに指定した空きメモリーの割合を割り当てることができます。

デフォルトでは、**nsslapd-import-cache-autosize** 属性は有効で、値が **-1** に設定されます。この値により、**ldif2db** 操作のインポートキャッシュが自動的に設定され、インポートキャッシュの空き物理メモリーの 50% が自動的に割り当てられます。パーセンテージの値 (50%) はハードコーディングされており、変更はできません。



属性値を **50** (`nsslapd-import-cache-autosize: 50`) に設定すると、**ldif2db** 操作中のパフォーマンスにも同じ効果があります。ただし、このような設定は、インポート操作が Directory Server タスクとして実行するとパフォーマンスに影響を及ぼします。**-1** の値は、インポート、一般的な Directory Server タスクなど、**ldif2db** 操作に対してのみインポートキャッシュを自動的にサイズします。



### 注記

**-1** の設定の目的は、**ldif2db** 操作を有効にして空きの物理メモリーを活用できるようにすることですが、同時に、Directory Server の一般的な操作に使用されるエントリーキャッシュと価値のあるメモリーには競合しないようにします。

`nsslapd-import-cache-autosize` 属性の値を **0** に設定すると、インポートキャッシュの自動サイズ機能が無効になります。つまり、インポート操作のいずれかのモードでは自動調整は行われません。代わりに、Directory Server はインポートキャッシュサイズに `nsslapd-import-cachesize` 属性を使用し、デフォルト値は **20000000** です。

Directory Server のコンテキストには、データベースキャッシュ、エントリーキャッシュ、およびインポートキャッシュの3つのキャッシュがあります。インポートキャッシュは、インポート操作時にのみ使用されます。`nsslapd-cache-autosize` 属性。これはエントリーキャッシュとデータベースキャッシュの自動調整に使用されます。これは、Directory Server の操作時にのみ使用され、**ldif2db** コマンドの実行中は使用しません。属性の値は、エントリーキャッシュとデータベースキャッシュに割り当てられる空き物理メモリーの割合です。

自動サイズ属性である `nsslapd-cache-autosize` と `nsslapd-import-cache-autosize` の両方が有効になっている場合は、合計値が100未満であることを確認します。

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	-1、0 (インポートキャッシュの自動サイズ設定をオフにします) から 100
デフォルト値	-1 (ldif2db に対してのみインポートキャッシュの自動サイズ設定をオンにし、空き物理メモリーの 50% をインポートキャッシュに割り当てます)
構文	整数
例	nsslapd-import-cache-autosize: -1

#### 4.4.2.26. nsslapd-dbcachesize

このパフォーマンスチューニング関連の属性は、データベースインデックスキャッシュサイズをバイト単位で指定します。これは、Directory Server が使用する物理 RAM の量を制御するうえで最も重要な値の1つです。

これはエントリーキャッシュではありません。これは、Berkeley データベースバックエンドがインデックス (**.db** ファイル) およびその他のファイルをキャッシュするために使用するメモリー量です。この値は、Berkeley DB API 関数 `set_cachesize` に渡されます。自動キャッシュサイズ変更が有効になってい

ると、サーバーがサーバーの起動後の段階でこれらの値を推測した値に置き換えると、この属性が上書きされます。

この属性に関する技術的な情報

は、[https://docs.oracle.com/cd/E17076\\_04/html/programmer\\_reference/general\\_am\\_conf.html#am\\_cor](https://docs.oracle.com/cd/E17076_04/html/programmer_reference/general_am_conf.html#am_cor) の Berkeley DB リファレンスガイドのキャッシュサイズセクションを参照してください。

数値ではない場合、または 32 ビットの符号付き整数には大きすぎる値を設定しようとする、**LDAP\_UNWILLING\_TO\_PERFORM** エラーメッセージが返され、問題を説明する追加のエラー情報が示されます。



### 注記

データベースキャッシュのサイズは手動で設定しないでください。Red Hat は、パフォーマンスを最適化するためにデータベースキャッシュの自動サイジング機能を使用することを推奨します。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。

この属性の変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	32 ビットプラットフォームの場合は 500 キロバイトから 4 ギガバイト、64 ビットプラットフォームの場合 500 キロバイトから $2^{64}-1$
デフォルト値	
構文	整数
例	nsslapd-dbcachesize: 10000000

#### 4.4.2.27. nsslapd-dbcache

この属性は、LDBM キャッシュを、メモリーの個別の部分に均等に分割することができます。一部のアーキテクチャーで連続して割り当てることができないように、十分な大きさのキャッシュを指定することができます。たとえば、一部のシステムでは、プロセスによって連続して割り当てられる可能性のあるメモリーの量が制限されています。**nsslapd-dbcache** が **0** または **1** の場合、キャッシュはメモリーの連続して割り当てられます。**1** より大きい場合、キャッシュは **ncache** に分割され、メモリーの個別の部分と同等にサイズが設定されます。

4 ギガバイトを超える **dbcache** サイズを設定するには、**nsslapd-dbcache** 属性行と **nsslapd-logdirectory** 属性行の間の **cn = config,cn = ldbm database,cn=plugins,cn=config** に **nsslapd-dbcache** 属性を追加します。

この値を、ギガバイト単位のメモリー量の  $1/4(1/4)$  の整数に設定します。たとえば、12 ギガバイトシステムの場合は **nsslapd-dbcache** の値を **3** に設定します。8 ギガバイトシステムの場合は、**2** に設定します。

この属性は、システムの変更/診断のためにのみ提供されており、Red Hat テクニカルサポートまたは Red Hat プロフェッショナルサービスのガイダンスがある場合にのみ変更する必要があります。この属性およびその他の設定属性の設定に一貫性がないと、Directory Server が不安定になる可能性があります。

この属性の変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	1 から 4
デフォルト値	1
構文	整数
例	nsslapd-dbncache: 1

#### 4.4.2.28. nsslapd-search-bypass-filter-test

**nsslapd-search-bypass-filter-test** パラメーターを有効にすると、Directory Server は、検索時に候補リストをビルドするタイミングでフィルターチェックを回避します。パラメーターを **verify** に設定すると、Directory Server は検索候補エントリーに対してフィルターを評価します。

パラメーター	説明
エントリー DN	cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	on   off   verify
デフォルト値	on
構文	ディレクトリー文字列
例	nsslapd-search-bypass-filter-test: on

#### 4.4.3. cn=monitor,cn=ldbm database,cn=plugins,cn=config 下のデータベース属性

データベースのアクティビティをモニターするためのデータベース統計を含むグローバル読み取り専用属性は、**cn=monitor,cn=ldbm database,cn=plugins,cn=config** ツリーノードに保管されます。これらのエントリーの詳細は、Red Hat Directory Server 管理ガイドのサーバーおよびデータベースアクティビティの監視の章を参照してください。

##### dbcachehits

この属性は、データベースで見つかった要求されたページを表示します。

##### dbcachetries

この属性は、キャッシュルックアップ合計を表示します。

#### **dbcachehitratio**

この属性は、データベースキャッシュ (hits/tries) で見つかった要求されたページのパーセンテージを表示します。

#### **dbcachepagein**

この属性は、データベースキャッシュに読み込まれたページを表示します。

#### **dbcachepageout**

この属性は、データベースキャッシュからバッキングファイルに書き込まれたページを表示します。

#### **dbcacheroevict**

この属性は、キャッシュから強制されたクリーンページを表示します。

#### **dbcacherwevict**

この属性は、キャッシュから強制されたダーティーページを表示します。

#### **normalizedDNcachetries**

インスタンスが開始してからのキャッシュルックアップの合計数。

#### **normalizedDNcachehits**

キャッシュ内にある正規化された DN。

#### **normalizedDNcachemisses**

キャッシュ内に正規化された DN が見つかりません。

#### **normalizedDNcachehitratio**

キャッシュにある正規化された DN のパーセンテージ。

#### **currentNormalizedDNcachesize**

正規化された DN キャッシュの現在のサイズ (バイト単位)。

#### **maxNormalizedDNcachesize**

**nsslapd-ndn-cache-max-size** パラメーターの現在の値。この設定の更新方法は、[「nsslapd-ndn-cache-max-size」](#) を参照してください。

#### **currentNormalizedDNcachecount**

正規化されたキャッシュされた DN の数。

### 4.4.4. cunder cn=database\_name,cn=ldb database,cn=plugins,cn=config 下のデータベース属性

**cn=database\_name** サブツリーには、ユーザー定義データベースのすべての設定データが含まれています。

**cn=userRoot** サブツリーは、デフォルトで **userRoot** と呼ばれます。ただし、これはハードコーディングされず、複数のデータベースインスタンスが存在すると、この名前が変更され、新規データベースが追加されるとユーザーによって変更および定義されます。参照される **cn=userRoot** データベースは、

任意のユーザーデータベースにすることができます。

次の属性は、**cn=userRoot**などのデータベースに共通です。

#### 4.4.4.1. nsslapd-cachesize

この属性は非推奨になっています。エントリーキャッシュのサイズを変更するには、**nsslapd-cachememsize**を使用します。

このパフォーマンスチューニング関連の属性は、保持できるエントリー数に関してキャッシュサイズを指定します。ただし、「**nsslapd-cachememsize**」で説明されているように、この属性は非推奨になりました。**nsslapd-cachememsize**属性は、エントリーキャッシュサイズのRAMの絶対割り当てを設定します。

数値ではない場合、または32ビットの符号付き整数には大きすぎる値(32ビットシステムの場合)を設定しようとする、**LDAP\_UNWILLING\_TO\_PERFORM**エラーメッセージが返され、問題を説明する追加のエラー情報が示されます。

この属性の変更を反映するには、サーバーを再起動する必要があります。



#### 注記

この設定のパフォーマンスカウンターは、32ビットシステムでも最高の64ビット整数になりますが、システムメモリーをアドレス指定する方法のため、設定自体は32ビットシステムでは最高の32ビット整数に制限されます。

パラメーター	説明
エントリー DN	cn=database_name,cn=ldbm database,cn=plugins,cn=config
Valid Range	32ビットシステムの場合は1から $2^{32}-1$ 、または64ビットシステムの場合は $2^{63}-1$ 、もしくは-1(制限がないという意味)です。
デフォルト値	-1
構文	整数
例	nsslapd-cachesize: -1

#### 4.4.4.2. nsslapd-cachememsize

このパフォーマンスチューニング関連の属性は、エントリーキャッシュに使用可能なメモリースペースのサイズをバイト単位で指定します。最も簡単な方法として、メモリー関連のキャッシュサイズを制限しています。自動キャッシュサイズ変更をアクティブにすると、この属性が上書きされ、サーバー起動の後の段階でこれらの値が独自の推測値に置き換えられます。

数値ではない場合、または32ビットの符号付き整数には大きすぎる値(32ビットシステムの場合)を設定しようとする、**LDAP\_UNWILLING\_TO\_PERFORM**エラーメッセージが返され、問題を説明する追加のエラー情報が示されます。

この設定のパフォーマンスカウンターは、32 ビットシステムでも最高の 64 ビット整数になりますが、システムメモリーをアドレス指定する方法のため、設定自体は 32 ビットシステムでは最高の 32 ビット整数に制限されます。



## 注記

データベースキャッシュのサイズは手動で設定しないでください。Red Hat は、パフォーマンスを最適化するためにエントリーキャッシュの自動サイジング機能を使用することを推奨します。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。

パラメーター	説明
エントリー DN	cn= <b>database_name</b> ,cn=ldbm database,cn=plugins,cn=config
Valid Range	64 ビットシステムでは 500 キロバイトから $2^{64} - 1$
デフォルト値	209715200 (200 MiB)
構文	整数
例	nsslapd-cachememsize: 209715200

### 4.4.4.3. nsslapd-directory

この属性は、データベースインスタンスへのパスを指定します。相対パスの場合は、グローバルデータベースエントリー **cn=config,cn=ldbm database,cn=plugins,cn=config** の **nsslapd-directory** で指定されたパスから開始します。データベースインスタンスディレクトリーの名前はインスタンス名の後にあり、デフォルトではグローバルデータベースディレクトリーにあります。データベースインスタンスの作成後に、このパスを変更しないでください。変更すると、サーバーがデータにアクセスできなくなる可能性があります。

パラメーター	説明
エントリー DN	cn= <b>database_name</b> ,cn=ldbm database,cn=plugins,cn=config
有効な値	データベースインスタンスへの有効なパス
デフォルト値	
構文	DirectoryString
例	nsslapd-directory: /var/lib/dirsrv/slappd- <b>instance</b> /db/userRoot

### 4.4.4.4. nsslapd-dncachememsize

このパフォーマンスチューニング関連の属性は、DN キャッシュで利用可能なメモリ領域のサイズをバイト単位で指定します。DN キャッシュはデータベースのエントリーキャッシュと似ていますが、テーブルのみがエントリー ID とエントリー DN を保存します。これにより、名前変更および moddn 操作のルックアップが速くなります。

最も簡単な方法として、メモリ関連のキャッシュサイズを制限しています。

数値ではない場合、または 32 ビットの符号付き整数には大きすぎる値 (32 ビットシステムの場合) を設定しようとする、**LDAP\_UNWILLING\_TO\_PERFORM** エラーメッセージが返され、問題を説明する追加のエラー情報が示されます。



### 注記

この設定のパフォーマンスカウンターは、32 ビットシステムでも最高の 64 ビット整数になりますが、システムメモリーをアドレス指定する方法のため、設定自体は 32 ビットシステムでは最高の 32 ビット整数に制限されます。

パラメーター	説明
エントリー DN	cn= <b>database_name</b> ,cn=ldbm database,cn=plugins,cn=config
Valid Range	500 キロバイトから、32 ビットシステムの場合は $2^{32}-1$ 、64 ビットシステムの場合は $2^{64}-1$
デフォルト値	10485,760 (10 メガバイト)
構文	整数
例	nsslapd-dncachememsize: 10485760

#### 4.4.4.5. nsslapd-readonly

この属性は、1つのバックエンドインスタンスの読み取り専用モードを指定します。この属性の値が **off** である場合、ユーザーにはアクセスパーミッションで許可されるすべての読み取り、書き込み、および実行パーミッションが付与されます。

パラメーター	説明
エントリー DN	cn= <b>database_name</b> ,cn=ldbm database,cn=plugins,cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-readonly: off

#### 4.4.4.6. nsslapd-require-index

**on** に切り替えると、この属性はインデックスなしの検索を拒否することができます。このパフォーマンス関連の属性は、サーバーに誤った検索で満たされないようにします。

パラメーター	説明
エントリー DN	cn= <b>database_name</b> ,cn=ldbm database,cn=plugins,cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-require-index: off

#### 4.4.4.7. nsslapd-require-internalop-index

プラグインがデータを変更すると、データベースに書き込みロックがあります。大規模なデータベースでは、プラグインがインデックス化されていない検索を実行すると、プラグインはすべてのデータベースロックを使用し、データベースが破損したり、サーバーが応答しなくなることがあります。この問題を回避するには、**nsslapd-require-internalop-index** パラメーターを有効にして、インデックス化されていない内部検索を拒否することができるようになりました。

パラメーター	説明
エントリー DN	cn= <b>database_name</b> ,cn=ldbm database,cn=plugins,cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsslapd-require-internalop-index: off

#### 4.4.4.8. nsslapd-suffix

この属性は、データベースリンクの接尾辞を指定します。各データベースインスタンスには接尾辞が1つしかないため、この属性は1つの値の属性です。以前は、1つのデータベースインスタンスに複数の接尾辞を含めることができましたが、これは今後そうではなくなりました。その結果、この属性の値は、各データベースインスタンスに接尾辞エントリーを1つだけ持つことができるという事実を強制します。エントリーの作成後にこの属性に加えた変更は、データベースリンクを含むサーバーを再起動した後のみ反映されます。



パラメーター	説明
エントリー DN	cn= <b>database_name</b> ,cn=ldbm database,cn=plugins,cn=config
有効な値	任意の有効な DN
デフォルト値	
構文	DirectoryString
例	nsslapd-suffix: o=Example

#### 4.4.4.9. vlvBase

この属性は、参照または仮想リストビュー (VLV) インデックスが作成されるベース DN を設定します。

VLV インデックスの詳細は、[管理ガイド](#)のインデックスの章を参照してください。

パラメーター	説明
エントリー DN	cn= <b>index_name</b> ,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
有効な値	任意の有効な DN
デフォルト値	
構文	DirectoryString
例	vlvBase: ou=People,dc=example,dc=com

#### 4.4.4.10. vlvEnabled

**vlvEnabled** 属性は特定の VLV インデックスのステータス情報を提供し、Directory Server はランタイム時にこの属性を設定します。**vlvEnabled** が設定に表示されますが、この属性を変更することはできません。

VLV インデックスの詳細は、[管理ガイド](#)のインデックスの章を参照してください。

パラメーター	説明
エントリー DN	cn= <b>index_name</b> ,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
有効な値	0 (無効)   1 (有効)

パラメーター	説明
デフォルト値	1
構文	DirectoryString
例	vlvEnabled: 0

#### 4.4.4.11. vlvFilter

ブラウジングまたは仮想リストビュー (VLV) インデックスは、フィルターに従って検索を実行し、そのフィルターに一致するエントリーをインデックスに含めることによって作成されます。フィルターは **vlvFilter** 属性で指定されます。

VLV インデックスの詳細は、**管理ガイド**のインデックスの章を参照してください。

パラメーター	説明
エントリー DN	cn= <b>index_name</b> ,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
有効な値	有効な LDAP フィルター
デフォルト値	
構文	DirectoryString
例	vlvFilter: (

#### 4.4.4.12. vlvIndex (オブジェクトクラス)

**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、エントリーヘッダーの省略インデックスを動的に生成するため、大規模なインデックスを視覚的に参照する方がはるかに速くなります。VLV インデックス定義には、インデックスを定義する部分と、インデックスに追加するエントリーを識別するのに使用される検索を定義する 2つの部分があります。**vlvIndex** オブジェクトクラスは、インデックスエントリーを定義します。

このオブジェクトクラスは Directory Server に定義されています。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.42

#### 表 4.2 必要な属性

属性	定義
objectClass	エントリーのオブジェクトクラスを定義します。
cn	エントリーの共通名を指定します。
「vlvSort」	参照インデックス (仮想リストビューインデックス) がソートされている属性リストを識別します。

表4.3 使用できる属性

属性	定義
「vlvEnabled」	参照インデックスの可用性を保管します。
「vlvUses」	参照インデックスが使用されるカウントが含まれません。

#### 4.4.4.13. vlvScope

この属性は、参照または仮想リストビュー (VLV) インデックスのエントリー用に実行する検索の範囲を設定します。

VLV インデックスの詳細は、[管理ガイド](#)のインデックスの章を参照してください。

パラメーター	説明
エントリー DN	cn= <b>index_name</b> ,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
有効な値	* 1 (1 レベルまたは子の検索) * 2 (サブツリー検索)
デフォルト値	
構文	整数
例	vlvScope: 2

#### 4.4.4.14. vlvSearch (オブジェクトクラス)

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、エントリーヘッダーの省略インデックスを動的に生成するため、大規模なインデックスを視覚的に参照する方がはるかに速くなります。VLV インデックス定義には、インデックスを定義する部分と、インデックスに追加するエントリーを識別するのに使用される検索を定義する 2 つの部分があります。**vlvSearch** オブジェクトクラスは、検索フィルターエントリーを定義します。

このオブジェクトクラスは Directory Server に定義されています。

## 上級クラス

top

## OID

2.16.840.1.113730.3.2.38

表4.4 必要な属性

属性	定義
objectClass	エントリーのオブジェクトクラスを定義します。
[ <a href="#">vlvBase</a> ]	参照インデックスが作成されるベース DN を特定します。
[ <a href="#">vlvScope</a> ]	参照インデックスを定義するスコープを識別します。
[ <a href="#">vlvFilter</a> ]	参照インデックスを定義するフィルター文字列を識別します。

表4.5 使用できる属性

属性	定義
multiLineDescription	エントリーのテキスト説明を入力します。

### 4.4.4.15. vlvSort

この属性は、参照または仮想リストビュー (VLV) インデックスで返されるエントリーのソート順序を設定します。



#### 注記

この属性のエントリーは、**vlvSearch** エントリーの下にある **vlvIndex** エントリーです。

VLV インデックスの詳細は、**管理ガイド**のインデックスの章を参照してください。

パラメーター	説明
エントリー DN	cn= <b>index_name</b> ,cn= <b>index_name</b> ,cn=userRoot,cn=ldb database,cn=plugins,cn=config
有効な値	スペースで区切られたリスト内の任意の Directory Server 属性

パラメーター	説明
デフォルト値	
構文	DirectoryString
例	vlvSort: cn givenName o ou sn

#### 4.4.4.16. vlvUses

**vlvUses** 属性には参照インデックスが使用するカウントが含まれ、Directory Server はランタイム時にこの属性を設定します。**vlvUses** が設定に表示されますが、この属性を変更することはできません。

VLV インデックスの詳細は、[管理ガイド](#)のインデックスの章を参照してください。

パラメーター	説明
エントリー DN	cn=index_name,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
有効な値	該当なし
デフォルト値	
構文	DirectoryString
例	vlvUses: 800

#### 4.4.5. cn=database,cn=monitor,cn=ldbm database,cn=plugins,cn=config 下のデータベース属性

このツリーノードエントリーの属性はすべて読み取り専用であるデータベースパフォーマンスカウンターです。これらの属性のすべての値は、**entrycachehits** および **entrycachetriess** を除く 32 ビットの整数です。

**cn=config** の **nsslapd-counters** 属性が **on** に設定されている場合は、32 ビットマシンや 32 ビットバージョンの Directory Server の整数を使用して、64 ビットの整数を使用して Directory Server インスタンスにより保持されるカウンターの一部。データベースの監視には、**entrycachehits** カウンターおよび **entrycachetriess** カウンターは 64 ビットの整数を使用します。



#### 注記

**nsslapd-counters** 属性は、これらの特定のデータベースおよびサーバーカウンターの 64 ビットサポートを有効にします。64 ビットの整数を使用するカウンターは設定できません。64 ビットの整数は、許可されるすべてのカウンターに対して有効であるか、許可されているすべてのカウンターに対して無効にされます。

**nsslapd-db-abort-rate**

この属性は、中止されたトランザクションの数を示します。

#### **nsslapd-db-active-txns**

この属性は、現在アクティブなトランザクションの数を表示します。

#### **nsslapd-db-cache-hit**

この属性は、キャッシュにある要求されたページを表示します。

#### **nsslapd-db-cache-try**

この属性は、キャッシュルックアップ合計を表示します。

#### **nsslapd-db-cache-region-wait-rate**

この属性は、リージョンロックを取得する前に、コントロールのスレッドが強制的に待機した回数を表示します。

#### **nsslapd-db-cache-size-bytes**

この属性は、キャッシュの合計サイズをバイト単位で表示します。

#### **nsslapd-db-clean-pages**

この属性は、現在のキャッシュにクリーンなページを表示します。

#### **nsslapd-db-commit-rate**

この属性は、コミットされたトランザクションの数を表示します。

#### **nsslapd-db-deadlock-rate**

この属性は、検出されたデッドロックの数を表示します。

#### **nsslapd-db-dirty-pages**

この属性は、現在のキャッシュにダーティーページを表示します。

#### **nsslapd-db-hash-buckets**

この属性は、バッファハッシュテーブルのハッシュバケットの数を表示します。

#### **nsslapd-db-hash-elements-examine-rate**

この属性は、ハッシュテーブルのルックアップ中に走査されたハッシュ要素の合計数を表示します。

#### **nsslapd-db-hash-search-rate**

この属性は、バッファハッシュテーブル検索の合計数を表示します。

#### **nsslapd-db-lock-conflicts**

この属性は、競合によりすぐに利用できないロックの合計数を表示します。

#### **nsslapd-db-lock-region-wait-rate**

この属性は、リージョンロックを取得する前に、コントロールのスレッドが強制的に待機した回数を表示します。

#### **nsslapd-db-lock-request-rate**

この属性は、要求されたロックの合計数を表示します。

**nsslapd-db-lockers**

この属性は、現在のロックの数を表示します。

**nsslapd-db-log-bytes-since-checkpoint**

この属性は、最後のチェックポイント以降にこのログに書き込まれたバイト数を表示します。

**nsslapd-db-log-region-wait-rate**

この属性は、リージョンロックを取得する前に、コントロールのスレッドが強制的に待機した回数を表示します。

**nsslapd-db-log-write-rate**

この属性は、このログに書き込まれたメガバイトおよびバイト数を表示します。

**nsslapd-db-longest-chain-length**

この属性は、バッファハッシュテーブル検索で最も長いチェーンを示しています。

**nsslapd-db-page-create-rate**

この属性は、キャッシュで作成されたページを表示します。

**nsslapd-db-page-read-rate**

この属性は、キャッシュに読み取れるページを表示します。

**nsslapd-db-page-ro-evict-rate**

この属性は、キャッシュから強制されたクリーンページを表示します。

**nsslapd-db-page-rw-evict-rate**

この属性は、キャッシュから強制されたダーティーページを表示します。

**nsslapd-db-page-trickle-rate**

この属性は、**memp\_trickle** インターフェイスを使用して書き込まれたダーティーページを表示します。

**nsslapd-db-page-write-rate**

この属性は、キャッシュに読み取れるページを表示します。

**nsslapd-db-pages-in-use**

この属性は、現在使用中のクリーンまたはダーティのすべてのページを表示します。

**nsslapd-db-txn-region-wait-rate**

この属性は、リージョンロックを取得する前に、コントロールのスレッドが強制的に待機した回数を表示します。

**currentdncachecount**

この属性は、DN キャッシュに現在存在している DN の数を表示します。

**currentdn cachesize**

この属性は、DN キャッシュに現在存在する DN の合計サイズをバイト単位で示します。

## maxdncachesize

この属性は、データベース DN キャッシュに保持できる DN の最大サイズをバイト単位で示します。

### 4.4.6. cn=monitor,cn=userRoot,cn=ldbm database,cn=plugins,cn=config 下のデータベース属性

このツリーノードエントリーの属性はすべて読み取り専用であるデータベースパフォーマンスカウンターです。

**cn=config** の **nsslapd-counters** 属性が **on** に設定されている場合は、32 ビットマシンや 32 ビットバージョンの Directory Server の整数を使用して、64 ビットの整数を使用して Directory Server インスタンスにより保持されるカウンターの一部。データベースの監視では、**entrycachehits** および **entrycachetrials** カウンターは 64 ビット整数を使用します。



#### 注記

**nsslapd-counters** 属性は、これらの特定のデータベースおよびサーバーカウンターの 64 ビットサポートを有効にします。64 ビットの整数を使用するカウンターは設定できません。64 ビットの整数は、許可されるすべてのカウンターに対して有効であるか、許可されているすべてのカウンターに対して無効にされます。

## dbfilename-number

この属性は、ファイルの名前を示し、ファイルの順次整数 ID (0 から始まる) を提供します。ファイルに関連するすべての統計には、この同じ数値 ID が割り当てられます。

## dbfilecachehit-number

この属性は、このファイルからデータを必要とする検索を実行し、データをキャッシュから正常に取得した回数を示します。この属性名の数字は、**dbfilename** にあるものに対応します。

## dbfilecachemiss-number

この属性は、このファイルからのデータを必要とする検索が実行され、データをキャッシュから取得できなかった回数を示します。この属性名の数字は、**dbfilename** にあるものに対応します。

## dbfilepagein-number

この属性は、このファイルからキャッシュに取りられたページ数を示します。この属性名の数字は、**dbfilename** にあるものに対応します。

## dbfilepageout-number

この属性は、キャッシュからディスクに書き込まれたこのファイルのページ数を示します。この属性名の数字は、**dbfilename** にあるものに対応します。

## currentDNcachecount

キャッシュされた DN の数。

## currentDNcachesize

DN キャッシュの現在のサイズ (バイト単位)。

## DNcachehitratio

キャッシュで見つかった DN のパーセンテージ。



## DNcachehits

キャッシュ内にある DNS。

## DNcachemisses

DNS がキャッシュ内に見つかりません。

## DNcachetries

インスタンスが開始してからのキャッシュルックアップの合計数。

## maxDNcachesize

`nsslapd-ndn-cache-max-size` パラメーターの現在の値。この設定の更新方法は、[「nsslapd-ndn-cache-max-size」](#) を参照してください。

### 4.4.7. cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config 下のデータベース属性

デフォルトのインデックスのセットはここに保存されます。ほとんどの設定シナリオの Directory Server 機能を最適化するために、デフォルトのインデックスはバックエンドごとに設定されます。システムに不可欠なものを除くすべてのインデックスは削除できますが、不要な中断が生じないように注意する必要があります。インデックスの詳細は、[Red Hat Directory Server 管理ガイド](#)のインデックスの管理の章を参照してください。

#### 4.4.7.1. cn

この属性は、インデックスする属性の名前です。

パラメーター	説明
エントリー DN	cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	有効なインデックス cn
デフォルト値	なし
構文	DirectoryString
例	cn: aci

#### 4.4.7.2. nsIndex

このオブジェクトクラスはバックエンドデータベースのインデックスを定義します。このオブジェクトは Directory Server で定義されています。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.44

表4.6 必要な属性

属性	定義
objectClass	エントリーのオブジェクトクラスを定義します。
cn	エントリーの共通名を指定します。
[nsSystemIndex]	インデックスがシステムで定義されるインデックスであるかどうかを特定します。

表4.7 使用できる属性

属性	定義
description	エントリーのテキスト説明を入力します。
[nsIndexType]	インデックスタイプを識別します。
[nsMatchingRule]	マッチングルールを識別します。

#### 4.4.7.3. nsIndexType

このオプションの複数値属性は、Directory Server 操作のインデックスのタイプを指定し、インデックス化される属性の値を取ります。必要なインデックスタイプは、それぞれ別の行に入力する必要があります。

パラメーター	説明
エントリー DN	cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	<ul style="list-style-type: none"> <li>* pres = 存在インデックス</li> <li>* eq = equality index</li> <li>* approx = approximate index</li> <li>* sub = 部分文字列インデックス</li> <li>* matching rule = 国際インデックス</li> <li>* index browse = 参照インデックス</li> </ul>
デフォルト値	
構文	DirectoryString

パラメーター	説明
例	nsIndexType: eq

#### 4.4.7.4. nsMatchingRule

このオプションの複数値属性は、値と一致し、属性のインデックスキーを生成するために使用される順序一致ルール名またはOIDを指定します。これは、英語(7ビットASCII)以外の言語で等式および範囲検索が正しく機能することを保証するために最も一般的に使用されます。

これは、スキーマ定義で順序一致ルールを指定しない整数構文属性に対して範囲検索が正しく機能するようにするためにも使用されます。**uidNumber**と**gidNumber**は、このカテゴリーに含まれる一般的に使用される2つの属性です。

たとえば、整数構文を使用する**uidNumber**の場合、ルール属性は**nsMatchingRule:integerOrderingMatch**のようになります。



#### 注記

この属性への変更は、変更が保存され、**db2index**を使用してインデックスが再構築されるまで有効になりません。詳細については、**Red Hat Directory Server 管理ガイド**のインデックスの管理の章を参照してください。

パラメーター	説明
エントリー DN	cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config
有効な値	有効な照合順序オブジェクト識別子 (OID)
デフォルト値	なし
構文	DirectoryString
例	nsMatchingRule: 2.16.840.1.113730.3.3.2.3.1 (ブルガリア語の場合)

#### 4.4.7.5. nsSystemIndex

この必須属性は、インデックスが**システムインデックス**であるかどうかを指定します。これは、Directory Server の操作に不可欠なインデックスです。この属性の値が**true**の場合は、システムに不可欠です。サーバー機能が深刻な影響を与えるため、システムインデックスは削除できません。

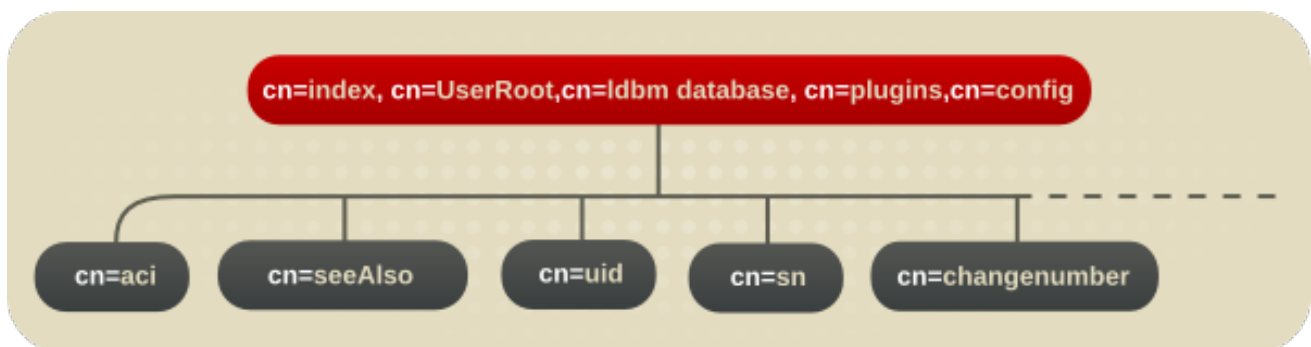
パラメーター	説明
エントリー DN	cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config

パラメーター	説明
有効な値	true   false
デフォルト値	
構文	DirectoryString
例	nssystemindex: true

#### 4.4.8. Database Attributes under cn=index,cn=database\_name,cn=ldb database,cn=plugins,cn=config

**cn=default indexes,cn=config,cn=ldb database,cn=plugins,cn=config** に保存されるデフォルトインデックスのセットのほかに、ユーザー定義のバックエンドインスタンス用にカスタムインデックスを作成できます。これらは **cn=index,cn=database\_name,cn=ldb database,cn=plugins,cn=config** に保存されます。次の図に示すように、インデックス付きの各属性は、**cn=config** 情報ツリーノードの下のサブエントリーを表します。

図4.2 サブエントリーを表すインデックス付き属性



たとえば、**o=UserRoot** の下にある **aci** 属性のインデックスファイルは、以下のように Directory Server に表示されます。

```
dn:cn=aci,cn=index,cn=UserRoot,cn=ldb database,cn=plugins,cn=config
objectclass:top
objectclass:nsIndex
cn:aci
nsSystemIndex:true
nsIndexType:pres
```

これらのエントリーは、「[cn=default indexes,cn=config,cn=ldb database,cn=plugins,cn=config](#) 下のデータベース属性」のデフォルトインデックスにリスト表示されるすべてのインデックス属性を共有します。インデックスの詳細は、[Red Hat Directory Server 管理ガイド](#)のインデックスの管理の章を参照してください。

##### 4.4.8.1. nsIndexIDListScanLimit

この複数値パラメーターは、特定のインデックスの検索制限や ID リストを使用しない場合は定義しません。詳細は、[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。

パラメーター	説明
エントリー DN	cn= <b>attribute_name</b> ,cn=index,cn= <b>database_name</b> ,cn=ldbm database,cn=plugins,cn=config
有効な値	<a href="#">Directory Server パフォーマンスチューニングガイド</a> の該当するセクションを参照してください。
デフォルト値	
構文	DirectoryString
例	nsIndexIDListScanLimit: limit=0 type=eq values=inetorgperson

#### 4.4.8.2. nsSubStrBegin

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。

この部分文字列の長さは、ワイルドカード文字の場所に基づいて編集できます。**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。以下に例を示します。

**abc\***

この属性の値が変更された場合は、**db2index** を使用してインデックスを再生成する必要があります。

パラメーター	説明
エントリー DN	cn= <b>attribute_name</b> ,cn=index,cn= <b>database_name</b> ,cn=ldbm database,cn=plugins,cn=config
有効な値	任意の整数
デフォルト値	3
構文	整数
例	nsSubStrBegin: 2

#### 4.4.8.3. nsSubStrEnd

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。

この部分文字列の長さは、ワイルドカード文字の場所に基づいて編集できます。**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

この属性の値が変更された場合は、**db2index** を使用してインデックスを再生成する必要があります。

パラメーター	説明
エントリー DN	cn= <b>attribute_name</b> ,cn=index,cn= <b>database_name</b> ,cn=ldbm database,cn=plugins,cn=config
有効な値	任意の整数
デフォルト値	3
構文	整数
例	nsSubStrEnd: 2

#### 4.4.8.4. nsSubStrMiddle

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。

この部分文字列の長さは、ワイルドカード文字の場所に基づいて編集できます。**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

この属性の値が変更された場合は、**db2index** を使用してインデックスを再生成する必要があります。

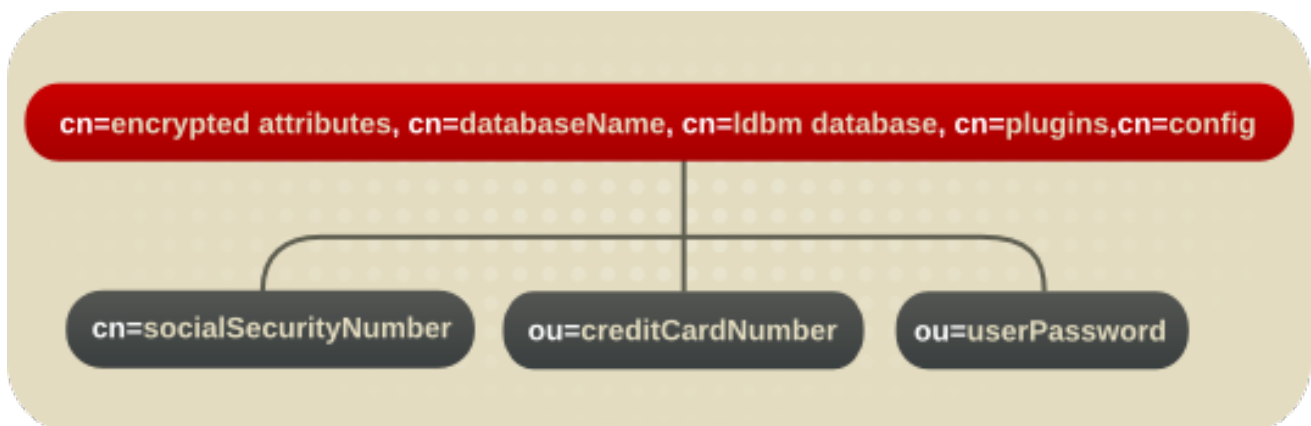
パラメーター	説明
エントリー DN	cn= <b>attribute_name</b> ,cn=index,cn= <b>database_name</b> ,cn=ldbm database,cn=plugins,cn=config
有効な値	任意の整数
デフォルト値	3

パラメーター	説明
構文	整数
例	nsSubStrMiddle: 3

#### 4.4.9. cn=attributeName,cn=encrypted attributes,cn=database\_name,cn=ldbm database,cn=plugins,cn=config 下のデータベース属性

**nsAttributeEncryption** オブジェクトクラスは、データベース内の属性の選択的な暗号化を可能にします。クレジットカード番号や政府の識別番号などの非常に機密性の高い情報は、日常的なアクセス制御手段では十分に保護されていない可能性があります。通常、これらの属性値はデータベース内の CLEAR に格納されます。保存時に暗号化すると、保護層がもう1つ追加されます。このオブジェクトクラスには、属性 **nsEncryptionAlgorithm** が1つあり、属性ごとに使用される暗号化暗号を設定します。次の図に示すように、暗号化された各属性は、上記の **cn=config** 情報ツリーノードの下のサブエントリを表します。

図4.3 cn=config ノードの下の暗号化された属性



たとえば、**o=UserRoot** の下にある **userPassword** 属性のデータベース暗号化ファイルは、以下のよう Directory Server に表示されます。

```
dn:cn=userPassword,cn=encrypted attributes,o=UserRoot,cn=ldbm database,
cn=plugins,cn=config
objectclass:top
objectclass:nsAttributeEncryption
cn:userPassword
nsEncryptionAlgorithm:AES
```

データベース暗号化を設定するには、**Red Hat Directory Server 管理ガイド**のデータベースの暗号化の章を参照してください。インデックスの詳細は、**Red Hat Directory Server 管理ガイド**のインデックスの管理の章を参照してください。

##### 4.4.9.1. nsAttributeEncryption (オブジェクトクラス)

このオブジェクトクラスは、DirectoryServer データベース内の選択された属性を識別および暗号化するコア設定エントリに使用されます。

このオブジェクトクラスは Directory Server に定義されています。

##### 上級クラス

top

**OID**

2.16.840.1.113730.3.2.316

表4.8 必要な属性

objectClass	エントリーのオブジェクトクラスを定義します。
cn	共通名を使用して暗号化される属性を指定します。
<code>nsEncryptionAlgorithm</code>	使用される暗号です。

**4.4.9.2. nsEncryptionAlgorithm**

`nsEncryptionAlgorithm` は、`nsAttributeEncryption` が使用する暗号を選択します。アルゴリズムは暗号化属性ごとに設定できます。

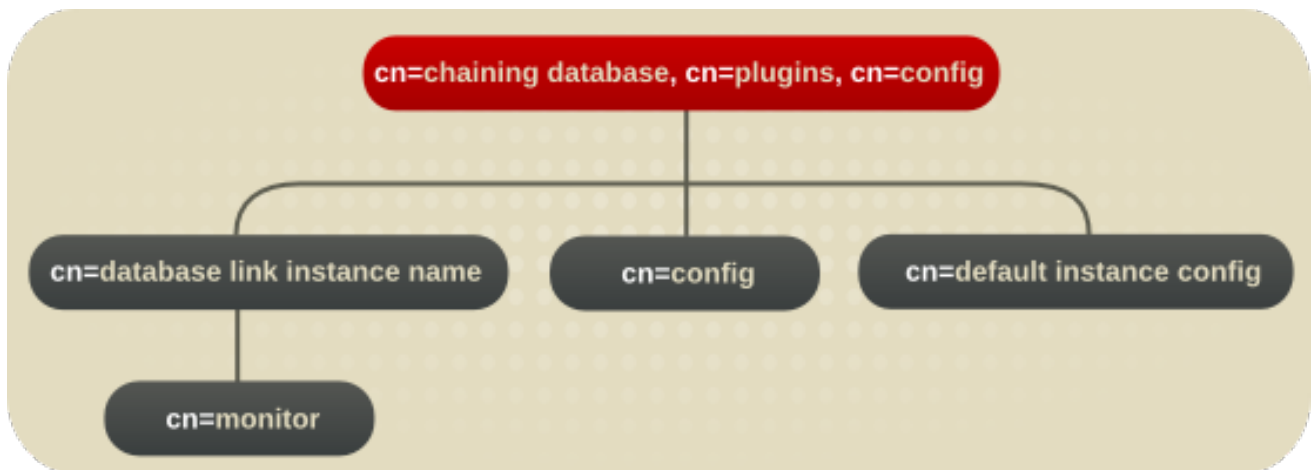
パラメーター	説明
エントリー DN	cn=attributeName,cn=encrypted attributes,cn= <b>database_name</b> ,cn=ldbm database,cn=plugins,cn=config
有効な値	<p>以下は、対応している暗号です。</p> <ul style="list-style-type: none"> <li>* Advanced Encryption Standard Block Cipher (AES)</li> <li>* Triple Data Encryption Standard Block Cipher (3DES)</li> </ul>
デフォルト値	
構文	DirectoryString
例	nsEncryptionAlgorithm: AES

**4.5. データベースリンクプラグイン属性 (チェーン属性)**

また、データベースリンクプラグイン属性は、以下の図に示すように情報ツリーに編成されています。



図4.4 データベースリンクプラグイン



データベースリンクインスタンスで使用されるすべてのプラグインテクノロジーは、**cn=chaining database** プラグインノードに格納されます。このセクションでは、図4.4「データベースリンクプラグイン」の **cn=chaining database,cn=plugins,cn=config** 情報ツリーで太字でマークされた3つのノードの追加の属性情報を示します。

#### 4.5.1. cn=config,cn=chaining database,cn=plugins,cn=config 下のデータベースリンク属性

このセクションでは、すべてのインスタンスに共通するグローバル設定属性を **cn=config,cn=chaining database,cn=plugins,cn=config** ツリーノードに格納します。

##### 4.5.1.1. nsActiveChainingComponents

この属性は、チェーンを使用してコンポーネントをリスト表示します。コンポーネントとは、サーバー内の機能単位です。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、**None**の値を使用します。この属性を使用すると、チェーンに使用されるコンポーネントを変更することもできます。デフォルトでは、チェーンが許可されないコンポーネントはありません。LDAP は空の属性が存在しないと見なすため、この属性が **cn=config,cn=chaining database,cn=config** 属性のリストに表示されなくなる理由を説明します。

パラメーター	説明
エントリー DN	cn=config,cn=chaining database,cn=plugins,cn=config
有効な値	有効なコンポーネントエントリー
デフォルト値	なし
構文	DirectoryString
例	nsActiveChainingComponents: cn=uid uniqueness,cn=plugins,cn=config

##### 4.5.1.2. nsMaxResponseDelay

このエラー検出のパフォーマンス関連属性は、エラーが疑われる前に、リモートサーバーがデータベースリンクによって行われた LDAP 操作要求に応答するのにかかる最大時間を指定します。この遅延期間が満たされると、データベースリンクはリモートサーバーとの接続をテストします。

パラメーター	説明
エントリー DN	cn=config,cn=chaining database,cn=plugins,cn=config
有効な値	有効な遅延期間 (秒単位)
デフォルト値	60 秒
構文	整数
例	nsMaxResponseDelay: 60

#### 4.5.1.3. nsMaxTestResponseDelay

このエラー検出のパフォーマンス関連の属性は、リモートサーバーが応答しているかどうかを確認するためのデータベースリンクが発行するテストの期間を指定します。この期間の経過前にリモートサーバーからの応答が返されなかった場合、データベースリンクはリモートサーバーが停止していることを想定し、後続の操作で接続は使用されません。

パラメーター	説明
エントリー DN	cn=config,cn=chaining database,cn=plugins,cn=config
有効な値	有効な遅延期間 (秒単位)
デフォルト値	15 秒
構文	整数
例	nsMaxTestResponseDelay: 15

#### 4.5.1.4. nsTransmittedControls

この属性は、グローバル (したがって動的) 設定またはインスタンス (つまり、**cn=データベースリンク** インスタンス、**cn=chaining database,cn=plugins,cn=config**) 設定属性の両方であり、データベースリンクを制御できます。変更されるように転送します。以下のコントロールは、デフォルトでデータベースリンクによって転送されます。

- 管理 DSA (OID: 2.16.840.1.113730.3.4.2)
- 仮想リストビュー (VLV) (OID: 2.16.840.1.113730.3.4.9)
- サーバー側のソート (OID: 1.2.840.113556.1.4.473)

- ループ検出 (OID: 1.3.6.1.4.1.1466.29539.12)

検索の逆参照や簡単なページ結果など、転送するコントロールリストにさらに制御を追加できます。

パラメーター	説明
エントリー DN	cn=config,cn=chaining database,cn=plugins,cn=config
有効な値	データベースリンクによって転送される有効な OID または上記のコントロール
デフォルト値	なし
構文	整数
例	nsTransmittedControls: 1.2.840.113556.1.4.473

#### 4.5.2. cn=default instance config,cn=chaining database,cn=plugins,cn=config 下のデータベースリンク属性

インスタンスのデフォルトのインスタンス設定属性は **cn=default instance config,cn=chaining database,cn=plugins,cn=config** ツリーノードに格納されます。

##### 4.5.2.1. nsAbandonedSearchCheckInterval

この属性は、サーバーが破棄操作をチェックするまで経過する秒数を表示します。

パラメーター	説明
エントリー DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	0 から最大 32 ビット 整数 (2147483647) 秒
デフォルト値	1
構文	整数
例	nsAbandonedSearchCheckInterval: 10

##### 4.5.2.2. nsBindConnectionsLimit

この属性は、データベースリンクがリモートサーバーで確立する TCP 接続の最大数を示します。

パラメーター	説明
エントリー DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	1 から 50 の接続
デフォルト値	3
構文	整数
例	nsBindConnectionsLimit: 3

#### 4.5.2.3. nsBindRetryLimit

名前が示すように、この属性は、データベースリンクがリモートサーバーとのバインドを再試行する回数ではなく、リモートサーバーとのバインドを**試行**する回数を指定します。ここでの値 **1** は、データベースリンクが一度だけバインドしようとするを示しています。



#### 注記

再試行は接続の失敗に対してのみ行われ、無効なバインド DN や不正なパスワードなどの他のタイプのエラーは発生しません。

パラメーター	説明
エントリー DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	0 から 5
デフォルト値	3
構文	整数
例	nsBindRetryLimit: 3

#### 4.5.2.4. nsBindTimeout

この属性は、バインドの試行がタイムアウトするまでの時間を表示します。妥当な範囲の制限を除き、この属性には実際の有効な範囲はありません。

パラメーター	説明
--------	----

パラメーター	説明
エントリー DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	0 から 60 秒
デフォルト値	15
構文	整数
例	nsBindTimeout: 15

#### 4.5.2.5. nsCheckLocalACI

高度な使用のために予約されます。この属性は、ACI がデータベースリンクおよび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。この属性への変更は、サーバーが再起動しないと反映されません。

パラメーター	説明
エントリー DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	nsCheckLocalACI: on

#### 4.5.2.6. nsConcurrentBindLimit

この属性は、TCP 接続ごとの同時バインド操作の最大数を表示します。

パラメーター	説明
エントリー DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	1 から 25 個のバインド
デフォルト値	10
構文	整数

パラメーター	説明
例	nsConcurrentBindLimit: 10

#### 4.5.2.7. nsConcurrentOperationsLimit

この属性は、許可される同時操作の最大数を指定します。

パラメーター	説明
エントリー DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	1 から 50 の操作
デフォルト値	2
構文	整数
例	nsConcurrentOperationsLimit: 5

#### 4.5.2.8. nsConnectionLife

この属性は、接続の有効期間を指定します。データベースリンクとリモートサーバー間の接続は、未指定の期間開いたままにしたり、一定期間後に閉じることができます。接続を開く方が高速ですが、より多くのリソースを使用します。値が **0** で、フェイルオーバーサーバーのリストが **nsFarmServerURL** 属性に提供されていると、メインサーバーは代替サーバーへのフェイルオーバー後に接続されません。

パラメーター	説明
エントリー DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	0 から無制限の秒 (0 は永遠を意味します)
デフォルト値	0
構文	整数
例	nsConnectionLife: 0

#### 4.5.2.9. nsOperationConnectionsLimit

この属性は、データベースリンクがリモートサーバーで確立する LDAP 接続の最大数を示します。

パラメーター	説明
エントリー DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	1 から n の接続
デフォルト値	20
構文	整数
例	nsOperationConnectionsLimit: 10

#### 4.5.2.10. nsProxiedAuthorization

高度な使用のために予約されます。プロキシ化された承認を無効にすると、チェーンされた操作のバインドは、**nsMultiplexorBindDn** 属性に設定されたユーザーとして実行されます。

パラメーター	説明
エントリー DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
有効な値	on   off
デフォルト値	on
構文	DirectoryString
例	nsProxiedAuthorization: on

#### 4.5.2.11. nsReferralOnScopedSearch

この属性は、参照がスコープ指定の検索によって返されるかどうかを制御します。スコープ指定の検索への応答で参照を返す方がより効率的であるため、この属性を使用してディレクトリーを最適化できます。設定されたすべてのファームサーバーへ参照が返されます。

パラメーター	説明
エントリー DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString

パラメーター	説明
例	nsReferralOnScopedSearch: off

#### 4.5.2.12. nsSizeLimit

この属性は、データベースリンクのデフォルトのサイズ制限をバイト単位で示します。

パラメーター	説明
エントリー DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	-1 (無制限) から 32 ビットの整数 (2147483647) エントリーまで
デフォルト値	2000
構文	整数
例	nsSizeLimit: 2000

#### 4.5.2.13. nsTimeLimit

この属性は、データベースリンクのデフォルトの検索時間制限を表示します。

パラメーター	説明
エントリー DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	-1 から最大 32 ビット 整数 (2147483647) 秒
デフォルト値	3600
構文	整数
例	nsTimeLimit: 3600

### 4.5.3. cn=database\_link\_name,cn=chaining database,cn=plugins,cn=config 下のデータベースリンク属性

この情報ノードは、データを含むサーバーに関する属性を保存します。ファームサーバーは、データベースのデータが含まれるサーバーです。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。チェーンをカスケードする場合、この URL は別のデータベースリンクを参照できます。



### 4.5.3.1. nsBindMechanism

この属性は、ファームサーバーがリモートサーバーに接続するバインドメカニズムを設定します。ファームサーバーは、1つ以上のデータベースにデータを含むサーバーです。この属性は、標準、TLS、または SASL のいずれかの接続タイプを設定します。

- **empty**.これは簡単な認証を実行し、バインド情報を提供する **nsMultiplexorBindDn** 属性および **nsMultiplexorCredentials** 属性が必要です。
- **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、リモートサーバーがファームサーバーの証明書をバインド ID にマップするように設定する必要があります。証明書マッピングについては、**管理ガイド**を参照してください。
- **DIGEST-MD5**.これは、DIGEST-MD5 暗号化を備えた SASL を使用します。単純な認証と同様に、これには、バインド情報を提供するために **nsMultiplexorBindDn** 属性と **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。DirectoryServer は SASL/GS-API over TLS をサポートしていないため、ファームサーバーは標準ポートを介して接続する必要があります。つまり、URL には **ldap** があります。  
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、**管理ガイド**を参照してください。

パラメーター	説明
エントリー DN	cn= <b>database_link_name</b> ,cn=chaining database,cn=plugins,cn=config
有効な値	* empty * EXTERNAL * DIGEST-MD5 * GSSAPI
デフォルト値	空
構文	DirectoryString
例	nsBindMechanism: GSSAPI

### 4.5.3.2. nsFarmServerURL

この属性は、リモートサーバーの LDAP URL を提供します。ファームサーバーは、1つ以上のデータベースにデータを含むサーバーです。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。カスケードの変更を使用する場合、この URL は別のデータベースリンクを参照できます。

パラメーター	説明
エントリー DN	cn= <b>database_link_name</b> ,cn=chaining database,cn=plugins,cn=config
有効な値	有効なリモートサーバーの LDAP URL
デフォルト値	
構文	DirectoryString
例	nsFarmServerURL: ldap://farm1.example.com farm2.example.com:389 farm3.example.com:1389/

#### 4.5.3.3. nsMultiplexorBindDN

この属性は、リモートサーバーとの通信に使用される管理エントリーの DN を指定します。**multiplexor** は、データベースリンクが含まれ、ファームサーバーと通信するサーバーです。このバインド DN をディレクトリーマネージャー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** としてバインドされます。

パラメーター	説明
エントリー DN	cn= <b>database_link_name</b> ,cn=chaining database,cn=plugins,cn=config
有効な値	
デフォルト値	マルチプレクサーの DN
構文	DirectoryString
例	nsMultiplexerBindDN: cn=proxy manager

#### 4.5.3.4. nsMultiplexorCredentials

プレーンテキストで指定された管理ユーザーのパスワード。パスワードが指定されていない場合は、ユーザーが **anonymous** 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。以下は、入力されている内容ではなく、表示される例です。

パラメーター	説明
エントリー DN	cn= <b>database_link_name</b> ,cn=chaining database,cn=plugins,cn=config

パラメーター	説明
有効な値	有効なパスワード。これは、DES 取り消し可能なパスワード暗号化スキーマを使用して暗号化されません。
デフォルト値	
構文	DirectoryString
例	nsMultiplexerCredentials: {DES} 9Ek069APCJfF

#### 4.5.3.5. nshoplimit

この属性は、データベースのチェーンが許可される最大回数を指定します。つまり、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リクエストを転送できる回数です。

パラメーター	説明
エントリー DN	cn= <b>database_link_name</b> ,cn=chaining database,cn=plugins,cn=config
Valid Range	1 からデプロイメントに適した上限
デフォルト値	10
構文	整数
例	nsHopLimit: 3

#### 4.5.3.6. nsUseStartTLS

この属性は、Start TLS を使用してセキュアでないポートでセキュアな暗号化された接続を開始するかどうかを設定します。この属性は、**nsBindMechanism** 属性が **EXTERNAL** に設定されているが、ファームサーバーの URL が標準 URL (**ldap**) に設定されている場合、または **nsBindMechanism** 属性が空のままの場合に使用できます。

パラメーター	説明
エントリー DN	cn= <b>database_link_name</b> ,cn=chaining database,cn=plugins,cn=config
有効な値	off   on
デフォルト値	off
構文	DirectoryString

パラメーター	説明
例	nsUseStartTLS: on

#### 4.5.4. cn=monitor,cn=database instance name,cn=chaining database,cn=plugins,cn=config 下のデータベースリンク属性

インスタンスでのアクティビティの監視に使用される属性は、**cn=monitor,cn=database instance name,cn=chaining database,cn=plugins,cn=config** 情報ツリーに保存されます。

##### nsAddCount

この属性は、受信される追加操作の数を示します。

##### nsDeleteCount

この属性は、受信される削除操作の数を示します。

##### nsModifyCount

この属性は、受信される変更操作の数を示します。

##### nsRenameCount

この属性は、受信される名前変更操作の数を示します。

##### nsSearchBaseCount

この属性は、受信されるベースレベルの検索の数を示します。

##### nsSearchOneLevelCount

この属性は、受信される1レベルの検索の数を示します。

##### nsSearchSubtreeCount

この属性は、受信されるサブツリー検索の数を指定します。

##### nsAbandonCount

この属性は、受信される破棄操作の数を提供します。

##### nsBindCount

この属性は、受信されるバインド要求の数を指定します。

##### nsUnbindCount

この属性は、受信されないバインドの数を指定します。

##### nsCompareCount

この属性は、受け取った比較操作の数を示します。

##### nsOperationConnectionCount

この属性は、通常の操作に対して開かれた接続の数を指定します。

##### nsOpenBindConnectionCount

この属性は、バインド操作に開いている接続の数を指定します。

## 4.6. PAM パススルー認証プラグイン属性

Unix システム上のローカルの PAM 設定は、LDAP ユーザーの外部認証ストアを利用できます。これはパススルー認証の形式で、Directory Server がディレクトリーアクセスに外部に保存されたユーザーの認証情報を使用できます。

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エントリーの下にある子エントリーで設定されます。PAM 認証用の可能な設定属性 (**60pam-plugin.ldif** スキーマファイルで定義) はすべて子エントリーで利用できます。子エントリーは PAM 設定オブジェクトクラスのインスタンスである必要があります。

### 例4.1 PAM パススルー認証設定エントリーの例

```
dn: cn=PAM Pass Through Auth,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
objectClass: pamConfig
cn: PAM Pass Through Auth
nsslapd-pluginPath: libpam-passthru-plugin
nsslapd-pluginInitfunc: pam_passthruauth_init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled: on
nsslapd-pluginLoadGlobal: true
nsslapd-plugin-depends-on-type: database
nsslapd-pluginId: pam_passthruauth
nsslapd-pluginVersion: 9.0.0
nsslapd-pluginVendor: Red Hat
nsslapd-pluginDescription: PAM pass through authentication plugin

dn: cn=Example PAM Config,cn=PAM Pass Through Auth,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
objectClass: pamConfig
cn: Example PAM Config
pamMissingSuffix: ALLOW
pamExcludeSuffix: cn=config
pamIDMapMethod: RDN ou=people,dc=example,dc=com
pamIDMapMethod: ENTRY ou=engineering,dc=example,dc=com
pamIDAttr: customPamUid
pamFilter: (manager=uid=bjensen,ou=people,dc=example,dc=com)
pamFallback: FALSE
pamSecure: TRUE
pamService: ldapserver
```

PAM 設定は、少なくとも Directory Server エントリーから PAM ユーザー ID を識別するためのマッピング方法、使用する PAM サーバー、サービスへのセキュアな接続を使用するかどうかを特定する必要があります。

```
pamIDMapMethod: RDN
pamSecure: FALSE
pamService: ldapserver
```

サブツリーを除外または具体的に含める、特定の属性値を PAM ユーザー ID にマップするなど、特別な設定のために設定を拡張できます。

#### 4.6.1. pamConfig (オブジェクトクラス)

このオブジェクトクラスは、ディレクトリーサービスと対話するための PAM 設定を定義するために使用されます。このオブジェクトクラスは Directory Server に定義されています。

##### 上級クラス

top

##### OID

2.16.840.1.113730.3.2.318

##### 使用できる属性

- 「[pamExcludeSuffix](#)」
- 「[pamIncludeSuffix](#)」
- 「[pamMissingSuffix](#)」
- 「[pamFilter](#)」
- 「[pamIDAttr](#)」
- 「[pamIDMapMethod](#)」
- 「[pamFallback](#)」
- 「[pamSecure](#)」
- 「[pamService](#)」
- **nsslapd-pluginConfigArea**

#### 4.6.2. pamExcludeSuffix

この属性は、PAM 認証から除外する接尾辞を指定します。

OID	2.16.840.1.113730.3.1.2068
構文	DN
多値または単一値	複数値
定義される場所	Directory Server

### 4.6.3. pamFallback

PAM 認証が失敗した場合に通常の LDAP 認証にフォールバックするかどうかを設定します。

OID	2.16.840.1.113730.3.1.2072
構文	Boolean
多値または単一値	単一値
定義される場所	Directory Server

### 4.6.4. pamFilter

PAM パススルー認証を使用するために含まれている接尾辞内の特定のエントリーを識別するために使用する LDAP フィルターを設定します。設定されていない場合、接尾辞内のすべてのエントリーが設定エントリーの対象になります。

OID	2.16.840.1.113730.3.1.2131
構文	Boolean
多値または単一値	単一値
定義される場所	Directory Server

### 4.6.5. pamIDAttr

この属性には、PAM ユーザー ID を保持するために使用される属性名が含まれています。

OID	2.16.840.1.113730.3.1.2071
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 4.6.6. pamIDMapMethod

LDAP バインド DN を PAMID にマップするために使用するメソッドを提供します。



#### 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。

OID	2.16.840.1.113730.3.1.2070
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

#### 4.6.7. pamIncludeSuffix

この属性は、PAM 認証に含める接尾辞を設定します。

OID	2.16.840.1.113730.3.1.2067
構文	DN
多値または単一値	複数値
定義される場所	Directory Server

#### 4.6.8. pamMissingSuffix

欠落している包含または除外接尾辞を処理する方法を識別します。オプションは次の通りです。ERROR は、バインド操作を失敗させます。ALLOW は、エラーをログに記録しますが、操作を続行できます。IGNORE は、操作を許可し、エラーをログに記録しません。

OID	2.16.840.1.113730.3.1.2069
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

#### 4.6.9. pamSecure

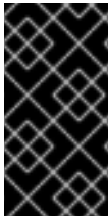
PAM 認証にはセキュアな TLS 接続が必要です。

OID	2.16.840.1.113730.3.1.2073
構文	Boolean
多値または単一値	単一値
定義される場所	Directory Server



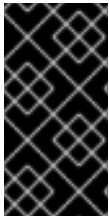
## 4.6.10. pamService

PAM に渡すサービス名が含まれます。これは、指定されたサービスに `/etc/pam.d/` ディレクトリーに設定ファイルがあることを前提としています。



### 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 **pam\_fprintd.so** 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。



### 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。

OID	2.16.840.1.113730.3.1.2074
構文	IA5String
多値または単一値	単一値
定義される場所	Directory Server

## 4.7. アカウントポリシープラグインの属性

アカウントポリシーは、一定期間が経過すると自動的にアカウントをロックするように設定できます。これは、事前設定された期間にのみ有効な一時的なアカウントを作成したり、一定期間非アクティブであったユーザーをロックしたりするために使用できます。

Account Policy プラグイン自体は、プラグイン設定エントリーを参照する引数のみを受け入れます。

```
dn: cn=Account Policy Plugin,cn=plugins,cn=config
...
nsslapd-pluginarg0: cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
```

アカウントポリシー設定エントリーは、サーバー全体で、アカウントポリシーに使用する属性を定義します。ほとんどの設定では、アカウントポリシーと有効期限の評価に使用する属性を定義しますが、設定はサブツリーレベルのアカウントポリシー定義を識別するために使用するオブジェクトクラスも定義します。

```
dn: cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
objectClass: top
objectClass: extensibleObject
cn: config
```

```
... attributes for evaluating accounts ...
alwaysRecordLogin: yes
```

```
stateattrname: lastLoginTime
altstateattrname: createTimeStamp
```

#### ... attributes for account policy entries ...

```
specattrname: acctPolicySubentry
limitattrname: accountInactivityLimit
```

プラグインの1つがグローバルで設定され、ユーザーのサブツリー内にアカウントポリシーエントリーを作成したり、これらのポリシーをサービスのクラスを介してユーザーおよびロールに適用できます。

### 例4.2 アカウントポリシー定義

```
dn: cn=AccountPolicy,dc=example,dc=com
objectClass: top
objectClass: ldapsubentry
objectClass: extensibleObject
objectClass: accountpolicy
# 86400 seconds per day * 30 days = 2592000 seconds
accountInactivityLimit: 2592000
cn: AccountPolicy
```

個々のユーザーとロールまたは CoS テンプレートの両方のエントリーには、アカウントポリシーのサブエントリーを指定できます。すべてのアカウントポリシーのサブエントリーには、有効期限ポリシーに対して追跡される作成およびログイン時間があります。

### 例4.3 アカウントポリシーを含むユーザーアカウント

```
dn: uid=scarter,ou=people,dc=example,dc=com
...
lastLoginTime: 20060527001051Z
acctPolicySubentry: cn=AccountPolicy,dc=example,dc=com
```

#### 4.7.1. altstateattrname

アカウントの有効期限ポリシーは、アカウントのいくつかの時間基準に基づいています。たとえば、非アクティブポリシーの場合、主要な基準は最終ログイン時刻 **lastLoginTime** です。ただし、アカウントにログインしたことがないユーザーなど、その属性がエントリーに存在しない場合があります。 **altstateattrname** 属性は、サーバーが有効期限を評価するために参照する backup 属性を提供します。

パラメーター	説明
エントリー DN	cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
Valid Range	時間ベースのエントリー属性
デフォルト値	なし

パラメーター	説明
構文	DirectoryString
例	altstateattrname: createTimeStamp

#### 4.7.2. alwaysRecordLogin

デフォルトでは、アカウントポリシーが直接適用されているエントリー (つまり、**acctPolicySubentry** 属性を持つエントリー) のみがログイン時間を追跡します。アカウントポリシーがサービスクラスまたはロールを通じて適用される場合、**acctPolicySubentry** 属性は、ユーザーエントリー自体ではなく、テンプレートまたはコンテナエントリーにあります。

**alwaysRecordLogin** 属性は、すべてのエントリーが最後のログイン時間を記録するように設定します。これにより、CoS およびロールを使用してアカウントポリシーを適用できます。

パラメーター	説明
エントリー DN	cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
Valid Range	yes   no
デフォルト値	いいえ
構文	DirectoryString
例	alwaysRecordLogin: no

#### 4.7.3. alwaysRecordLoginAttr

**Account Policy** プラグインは、**alwaysRecordLoginAttr** パラメーターに設定された属性名を使用して、ユーザーのディレクトリーエントリーに最後に成功したログインの時間を保存します。詳細は、[Directory Server 管理ガイド](#)の該当するセクションを参照してください。

パラメーター	説明
エントリー DN	cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
Valid Range	有効な属性名
デフォルト値	stateAttrName
構文	DirectoryString
例	alwaysRecordLoginAttr: lastLoginTime

#### 4.7.4. limitattrname

ユーザーディレクトリーのアカウントポリシーエントリーは、アカウントロックアウトポリシーの時間制限を定義します。この時間制限は任意のタイムベースの属性で設定でき、ポリシーエントリーには `ti` に複数の時間ベースの属性を含めることができます。アカウントの非アクティブ化制限に使用するポリシー内の属性は、Account Policy プラグインの **limitattrname** 属性で定義され、すべてのアカウントポリシーにグローバルに適用されます。

パラメーター	説明
エントリー DN	cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
Valid Range	時間ベースのエントリー属性
デフォルト値	なし
構文	DirectoryString
例	limitattrname: accountInactivityLimit

#### 4.7.5. specattrname

アカウントポリシーには、プラグイン設定エントリーのグローバル設定と、ユーザーディレクトリー内のエントリーの `user-` または `subtree-level` の設定という 2 つの設定エントリーがあります。アカウントポリシーはユーザーエントリーに直接設定することも、CoS またはロール設定の一部として設定することもできます。プラグインがアカウントポリシー設定エントリーであるエントリーを識別する方法は、アカウントポリシーとしてフラグを立てるエントリーの特定の属性を識別することです。プラグイン設定のこの属性は **specattrname** です。通常は **acctPolicySubentry** に設定されます。

パラメーター	説明
エントリー DN	cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
Valid Range	時間ベースのエントリー属性
デフォルト値	なし
構文	DirectoryString
例	specattrname: acctPolicySubentry

#### 4.7.6. stateattrname

アカウントの有効期限ポリシーは、アカウントのいくつかの時間基準に基づいています。たとえば、非アクティブポリシーの場合、主要な基準は最終ログイン時刻 **lastLoginTime** です。アカウントポリシーの評価に使用される主な時間属性は、**stateattrname** 属性に設定されます。

パラメーター	説明
エントリー DN	cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
Valid Range	時間ベースのエントリー属性
デフォルト値	なし
構文	DirectoryString
例	stateattrname: lastLoginTime

## 4.8. AD DN プラグインの属性

AD DN プラグインは、複数のドメイン設定をサポートします。ドメインごとに1つの設定エントリーを作成します。詳細は、[Red Hat Directory Server 管理ガイド](#)の該当するセクションを参照してください。

### 4.8.1. cn

設定エントリーのドメイン名を設定します。プラグインは、認証ユーザー名のドメイン名を使用して、対応する設定エントリーを選択します。

パラメーター	説明
エントリー DN	cn= <b>domain_name</b> ,cn=addn,cn=plugins,cn=config
有効なエントリー	任意の文字列
デフォルト値	なし
構文	DirectoryString
例	cn: example.com

### 4.8.2. addn\_base

Directory Server がユーザーの DN を検索するベース DN を設定します。

パラメーター	説明
エントリー DN	cn= <b>domain_name</b> ,cn=addn,cn=plugins,cn=config
有効なエントリー	任意の有効な DN

パラメーター	説明
デフォルト値	なし
構文	DirectoryString
例	addn_base: ou=People,dc=example,dc=com

### 4.8.3. addn\_filter

検索フィルターを設定します。Directory Server は、`%s` 変数を、認証ユーザーのドメイン以外の部分に自動的に置き換えます。たとえば、バインド内のユーザー名が `user_name@example.com` の場合、フィルターは対応する DN (`&(objectClass = account)(uid = user_name)`) を検索します。

パラメーター	説明
エントリー DN	cn=domain_name,cn=addn,cn=plugins,cn=config
有効なエントリー	任意の有効な DN
デフォルト値	なし
構文	DirectoryString
例	addn_filter: (&(objectClass=account)(uid=%s))

## 4.9. AUTO MEMBERSHIP プラグインの属性

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指定した静的グループに追加します。

Auto Membership プラグイン自体は、のコンテナエントリーです。各 automember 定義は、Auto Membership プラグインの子です。automember 定義は、LDAP 検索ベースと、エントリーを追加するデフォルトグループを特定するフィルターを定義します。

```
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,ou=groups,dc=example,dc=com
autoMemberGroupingAttr: member:dn
```

各 automember 定義には、グループにエントリーを割り当てる追加の条件を定義する独自の子エントリーを含めることができます。正規表現を使用すると、エントリーを包含または除外し、その条件に基づいて特定のグループに割り当てることができます。

```
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
```

エントリーがメイン定義に一致し、正規表現条件のいずれにも一致しない場合、メイン定義のグループを使用します。正規表現条件と一致する場合は、正規表現の条件グループに追加されます。

#### 4.9.1. autoMemberDefaultGroup

この属性は、エントリーをメンバーとして追加するデフォルトまたはフォールバックグループを設定します。定義エントリーのみを使用する場合は、一致するすべてのエントリーが追加されるグループになります。正規表現条件が使用される場合、LDAP 検索フィルターに一致するエントリーが正規表現にマッチしない場合、このグループはフォールバックとして使用されます。

パラメーター	説明
エントリー DN	cn=Auto Membership Plugin,cn=plugins,cn=config
Valid Range	既存の Directory Server グループ
デフォルト値	なし
単一値または複数値	単一値
構文	DirectoryString
例	autoMemberDefaultGroup: cn=hostgroups,ou=groups,dc=example,dc=com

#### 4.9.2. autoMemberDefinition (オブジェクトクラス)

この属性は、エントリーを automember 定義として識別します。このエントリーは、Auto Membership プラグイン (**cn=Auto Membership Plugin,cn=plugins,cn=config**) の子である必要があります。

##### 使用できる属性

- autoMemberScope
- autoMemberFilter
- autoMemberDefaultGroup
- autoMemberGroupingAttr

#### 4.9.3. autoMemberExclusiveRegex

この属性は、除外するエントリーの特定に使用する単一の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、[pcreyntax\(3\) の man ページ](#) を参照してください。



### 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。

パラメーター	説明
エントリー DN	cn=Auto Membership Plugin,cn=plugins,cn=config
Valid Range	正規表現
デフォルト値	なし
単一値または複数值	複数值
構文	DirectoryString
例	autoMemberExclusiveRegex: fqdn=^www\.web[0-9]+\\.example\.com

#### 4.9.4. autoMemberFilter

この属性は、一致するエントリーの検索に使用する標準の LDAP 検索フィルターを設定します。

パラメーター	説明
エントリー DN	cn=Auto Membership Plugin,cn=plugins,cn=config
Valid Range	有効な LDAP 検索フィルターです。
デフォルト値	なし
単一値または複数值	単一値
構文	DirectoryString
例	autoMemberFilter:objectclass=ntUser

#### 4.9.5. autoMemberGroupingAttr

この属性は、`group_member_attr:entry_attr` の形式で、グループエントリーのメンバー属性と、member 属性値を提供するオブジェクトエントリーの属性を指定します。



この構造では、グループの設定に応じて Automembership プラグインがグループにメンバーを追加する方法。たとえば、**groupOfUniqueNames** ユーザーグループの場合、各メンバーは **uniqueMember** 属性として追加されます。**uniqueMember** の値は、ユーザーエントリーの DN です。基本的に、各グループメンバーは **uniqueMember: user\_entry\_DN** の属性/値のペアで識別されます。メンバーエントリーのフォーマットは **uniqueMember:dn** です。

パラメーター	説明
エントリー DN	cn=Auto Membership Plugin,cn=plugins,cn=config
Valid Range	Directory Server の属性
デフォルト値	なし
単一値または複数値	単一値
構文	DirectoryString
例	autoMemberGroupingAttr: member:dn

#### 4.9.6. autoMemberInclusiveRegex

この属性は、追加するエントリーの特定に使用する単一の正規表現を設定します。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に一致する場合、そのエントリーはグループに含まれます (除外式と一致しない場合)。

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。

パラメーター	説明
エントリー DN	cn=Auto Membership Plugin,cn=plugins,cn=config
Valid Range	正規表現
デフォルト値	なし
単一値または複数値	複数値
構文	DirectoryString
例	autoMemberInclusiveRegex: fqdn=^www\.web[0-9]+\\.example\.com

#### 4.9.7. autoMemberProcessModifyOps

デフォルトでは、Directory Server は Automembership プラグインを呼び出して追加および変更操作を行います。この設定では、グループにグループエントリーを追加したり、ユーザーのグループエント

リーを変更したりする際に、プラグインがグループを変更します。**autoMemberProcessModifyOps** を **off** に設定すると、Directory Server はグループエントリーをユーザーに追加するときのみ Automembership プラグインを呼び出します。この場合、管理者がユーザーエントリーを変更し、そのエントリーがユーザーが属する Automembership グループに影響を与える場合、プラグインは古いグループからユーザーを削除せず、新規グループのみを追加します。古いグループを更新するには、修正タスクを手動で実行する必要があります。

パラメーター	説明
エントリー DN	cn=Auto Membership Plugin,cn=plugins,cn=config
有効な値	on   off
デフォルト値	on
単一値または複数値	単一値
構文	DirectoryString
例	autoMemberProcessModifyOps: on

#### 4.9.8. autoMemberRegexRule (オブジェクトクラス)

この属性は、エントリーを正規表現ルールとして識別します。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。

##### 使用できる属性

- `autoMemberInclusiveRegex`
- `autoMemberExclusiveRegex`
- `autoMemberTargetGroup`

#### 4.9.9. autoMemberScope

この属性は、エントリーを検索するサブツリー DN を設定します。これは検索ベースです。

パラメーター	説明
エントリー DN	cn=Auto Membership Plugin,cn=plugins,cn=config
Valid Range	Directory Server のサブツリー
デフォルト値	なし
単一値または複数値	単一値
構文	DirectoryString

パラメーター	説明
例	autoMemberScope: dc=example,dc=com

#### 4.9.10. autoMemberTargetGroup

この属性は、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。

パラメーター	説明
エントリー DN	cn=Auto Membership Plugin,cn=plugins,cn=config
Valid Range	Directory Server グループ
デフォルト値	なし
単一値または複数值	単一値
構文	DirectoryString
例	autoMemberTargetGroup: cn=websevers,cn=hostgroups,ou=groups,dc=example,dc=com

### 4.10. DISTRIBUTED NUMERIC ASSIGNMENT プラグインの属性

Distributed Numeric Assignment プラグインは、数値の範囲を管理し、その範囲内の一意の番号をエントリーに割り当てます。番号の割り当てを範囲に分割することで、Distributed Numeric Assignment プラグインは、競合なしに複数のサーバーが数値を割り当てることができます。プラグインは、サーバーに割り当てられた範囲も管理します。そのため、1つのインスタンスがその範囲で迅速に実行される場合は、他のサーバーから追加の範囲を要求できます。

分散数値割り当ては、単一の属性型または複数の属性タイプを使用するように設定でき、サブツリー内の特定の接尾辞および特定のエントリーにのみ適用されます。

分散数値割り当ては属性ごとに処理され、サブツリー内の特定の接尾辞と特定のエントリーにのみ適用されます。

#### 4.10.1. dnaPluginConfig (オブジェクトクラス)

このオブジェクトクラスは、エントリーに割り当てる DNA プラグインおよび数値範囲を設定するエントリーに使用されます。

このオブジェクトクラスは Directory Server に定義されています。

#### 上級クラス

top

## OID

2.16.840.1.113730.3.2.324

### 使用できる属性

- dnaType
- dnaPrefix
- dnaNextValue
- dnaMaxValue: 1000
- dnaInterval
- dnaMagicRegen
- dnaFilter
- dnaScope
- dnaSharedCfgDN
- dnaThreshold
- dnaNextRange
- dnaRangeRequestTimeout
- cn

### 4.10.2. dnaFilter

この属性は、分散数値割り当て範囲を適用するエントリーを検索および識別するために使用する LDAP フィルターを設定します。

**dnaFilter** 属性は、属性の分散数値割り当てを設定するために必要です。

パラメーター	説明
エントリー DN	cn= <b>DNA_config_entry</b> ,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
Valid Range	有効な LDAP フィルター
デフォルト値	なし
構文	DirectoryString
例	dnaFilter: (objectclass=person)

### 4.10.3. dnaInterval

この属性は、範囲内の数値をインクリメントするために使用する間隔を設定します。基本的に、これは事前定義されたレートで数値をスキップします。間隔が **3** で、範囲内の最初の数字が **1** の場合、範囲内で使用される次の数字は **4**、**7**、**10** と、新しい数字を割り当てるたびに3ずつ増加していきます。

レプリケーション環境では、**dnalInterval** により、複数のサーバーが同じ範囲を共有できます。ただし、同じ範囲を共有する異なるサーバーを設定する場合は、それに応じて **dnalInterval** パラメーターと **dnaNextVal** パラメーターを設定し、異なるサーバーが同じ値を生成しないようにします。レプリケーショントポロジーに新しいサーバーを追加する場合も、これを考慮する必要があります。

パラメーター	説明
エントリー DN	cn= <b>DNA_config_entry</b> ,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
Valid Range	任意の整数
デフォルト値	1
構文	整数
例	dnalInterval: 1

#### 4.10.4. dnaMagicRegen

この属性は、エントリーに新しい値を割り当てるようにプラグインに指示するユーザー定義の値を設定します。マジック値は、既存のエントリーに新しい一意の番号を割り当てるため、または新しいエントリーを追加するときの標準設定として使用できます。

マジックエントリーは、誤ってトリガーされないように、サーバーに対して定義された範囲外にある必要があります。DirectoryString または他の文字タイプで使用する場合は、この属性は数値である必要はないことに注意してください。ただし、ほとんどの場合、DNA プラグインは整数値のみを許可する属性で使用されます。この場合は、**dnamagicregen** 値も整数である必要があります。

パラメーター	説明
エントリー DN	cn= <b>DNA_config_entry</b> ,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
Valid Range	任意の文字列
デフォルト値	なし
構文	DirectoryString
例	dnaMagicRegen: -1

#### 4.10.5. dnaMaxValue: 1000

この属性は、範囲に割り当てることができる最大値を設定します。デフォルトは **-1** で、最大 64 ビット整数を設定するのと同じです。

パラメーター	説明
エントリー DN	cn= <b>DNA_config_entry</b> ,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
Valid Range	32 ビットシステムでは 1 から最大 32 ビット整数、64 ビットシステムでは最大 64 ビット整数。-1 は無制限。
デフォルト値	-1
構文	整数
例	dnaMaxValue: 1000

#### 4.10.6. dnaNextRange

この属性は、現在の範囲が使い切られるときに使用する次の範囲を定義します。この値は、範囲がサーバー間で転送される際に自動的に設定されますが、範囲要求を使用しない場合は、手動で範囲をサーバーに追加するように設定することもできます。

**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** が明示的に設定されているサーバーがその設定された **dnaMaxValue** に達した場合、次の値のセット (**dnaNextRange** の一部) がこのデッキから割り当てられます。

**dnaNextRange** の割り当ては、DNA 設定で設定された **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に内部で処理される場合。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。

この属性は、範囲を `lower_range-upper_range` の形式で設定します。

パラメーター	説明
エントリー DN	cn= <b>DNA_config_entry</b> ,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
Valid Range	32 ビットシステムでは 1 から最大 32 ビット整数、64 ビットシステムでは下限と上限の h 範囲に最大 64 ビット整数

パラメーター	説明
デフォルト値	なし
構文	DirectoryString
例	dnaNextRange: 100-500

#### 4.10.7. dnaNextValue

この属性は、次に割り当て可能な番号を提供します。設定エントリーで最初に設定された後、この属性は分散数値割り当てプラグインによって管理されます。

**dnaNextValue** 属性は、属性の分散数値割り当てを設定するために必要です。

パラメーター	説明
エントリー DN	cn= <b>DNA_config_entry</b> ,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
Valid Range	32 ビットシステムでは1から最大 32 ビット整数、64 ビットシステムでは最大 64 ビット整数
デフォルト値	-1
構文	整数
例	dnaNextValue: 1

#### 4.10.8. dnaPrefix

この属性は、属性に生成された番号の値の前に付けることができる接頭辞を定義します。たとえば、**user1000** などのユーザー ID を生成するには、**dnaPrefix** 設定は **user** になります。

**dnaPrefix** は、あらゆる種類の文字列を保持できます。ただし、**dnaType** の一部の可能な値 (**uidNumber** や **gidNumber** など) には整数値のみが必要です。接頭辞文字列を使用するには、文字列を許可する **dnaType** のカスタム属性の使用を検討してください。

パラメーター	説明
エントリー DN	cn= <b>DNA_config_entry</b> ,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
Valid Range	任意の文字列
デフォルト値	None

パラメーター	説明
例	dnaPrefix: id

#### 4.10.9. dnaRangeRequestTimeout

Distributed Numeric Assignment プラグインを使用すると考えられる状況の1つは、1台のサーバーが割り当てる番号が不足し始めていることです。**dnaThreshold** 属性は、範囲内で使用可能な番号のしきい値を設定します。これにより、サーバーは、番号の割り当てを実行できなくなる前に、他のサーバーに追加の範囲を要求できます。

**dnaRangeRequestTimeout** 属性は、範囲要求のタイムアウト期間を秒単位で設定します。これにより、サーバーは1つのサーバーからの新しい範囲の待機を停止せず、新しいサーバーからの範囲を要求できます。

範囲要求を実行するには、**dnaSharedCfgrDN** 属性を設定する必要があります。

パラメーター	説明
エントリー DN	cn= <b>DNA_config_entry</b> ,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
Valid Range	32 ビットシステムでは1から最大 32 ビット整数、64 ビットシステムでは最大 64 ビット整数
デフォルト値	10
構文	整数
例	dnaRangeRequestTimeout: 15

#### 4.10.10. dnaScope

この属性は、分散数値割り当てを適用するエントリーを検索するためのベース DN を設定します。これは **ldapsearch** のベース DN と似ています。

パラメーター	説明
エントリー DN	cn= <b>DNA_config_entry</b> ,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
Valid Range	Directory Server エントリー
デフォルト値	なし
構文	DirectoryString
例	dnaScope: ou=people,dc=example,dc=com



### 4.10.11. dnaSharedCfgDN

この属性は、サーバーが範囲を別の転送に使用できる共有 ID を定義します。このエントリーはサーバー間で複製され、他のサーバーが利用可能な範囲を認識できるようにプラグインによって管理されます。範囲転送を有効にするには、この属性を設定する必要があります。



#### 注記

共有設定エントリーは、エントリーをサーバーに複製できるように、レプリケートされたサブツリーで設定する必要があります。たとえば、**ou=People,dc=example,dc=com** サブツリーが複製されると、設定エントリーは **ou=UID Number Ranges, ou=People,dc=example,dc=com** などのサブツリーに存在する必要があります。

この設定で識別されるエントリーは、管理者が手動で作成する必要があります。サーバーには、範囲を転送するためにその下にサブエントリーが自動的に含まれます。

パラメーター	説明
エントリー DN	cn= <b>DNA_config_entry</b> ,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
Valid Range	任意の DN
デフォルト値	なし
構文	DN
例	dnaSharedCfgDN: cn=range transfer user,cn=config

### 4.10.12. dnaThreshold

Distributed Numeric Assignment プラグインを使用すると考えられる状況の1つは、1台のサーバーが割り当てる番号が不足し始め、問題が発生する可能性があることです。分散数値割り当てプラグインを使用すると、サーバーは他のサーバーで使用可能な範囲から新しい範囲を要求できます。

サーバーは割り当てられた範囲の終了に到達すると認識できるため、**dnaThreshold** 属性は範囲内で利用可能な残りの数字のしきい値を設定します。サーバーがしきい値に達すると、新しい範囲の要求を送信します。

範囲要求を実行するには、**dnaSharedCfgDN** 属性を設定する必要があります。

パラメーター	説明
エントリー DN	cn= <b>DNA_config_entry</b> ,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
Valid Range	32 ビットシステムでは1から最大 32 ビット整数、 64 ビットシステムでは最大 64 ビット整数

パラメーター	説明
デフォルト値	100
構文	整数
例	dnaThreshold: 100

#### 4.10.13. dnaType

この属性は、一意の数字が生成される属性を設定します。この場合、マジック番号を持つエントリーに属性が追加されるたびに、割り当てられた値が自動的に指定されます。

この属性は、属性に分散した数値割り当てを設定するには必要になります。

**dnaPrefix** 属性が設定されている場合、接頭辞の値は **dnaType** によって生成される値の前に付けられます。**dnaPrefix** 値には任意の文字列を指定できますが、**dnaType** (**uidNumber** や **gidNumber** など) の妥当な値には整数値のみが必要になります。接頭辞文字列を使用するには、文字列を許可する **dnaType** のカスタム属性の使用を検討してください。

パラメーター	説明
エントリー DN	cn= <b>DNA_config_entry</b> ,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
Valid Range	Directory Server の属性
デフォルト値	None
例	dnaType: uidNumber

#### 4.10.14. dnaSharedConfig (オブジェクトクラス)

このオブジェクトクラスは、数値割り当てのために同じ DNA プラグイン設定を使用するサプライヤー間でレプリケートされる共有設定エントリーを設定するために使用されます。

このオブジェクトクラスは Directory Server に定義されています。

##### 上級クラス

top

##### OID

2.16.840.1.113730.3.2.325

##### 使用できる属性

- dnaHostname
- dnaPortNum

- dnaSecurePortNum
- dnaRemainingValues

#### 4.10.15. dnaHostname

この属性は、複数サ upplier レプリケーションの特定ホストの DNA 範囲設定の一部として、共有範囲内のサーバーのホスト名を特定します。利用可能な範囲はホストによって追跡され、範囲情報はすべてのサプライヤー間で複製されるため、サプライヤーが利用可能な数が少ない場合に、ホスト情報を使用して別のサプライヤーに連絡し、新しい範囲を要求できます。

パラメーター	説明
エントリー DN	cn= <b>DNA_config_entry</b> ,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
構文	DirectoryString
Valid Range	有効なホスト名
デフォルト値	None
例	dnahostname: ldap1.example.com

#### 4.10.16. dnaPortNum

この属性は、**dnaHostname** で特定されたホストへの接続に使用する標準のポート番号を指定します。

パラメーター	説明
エントリー DN	cn= <b>DNA_config_entry</b> ,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
構文	整数
Valid Range	0 から 65535
デフォルト値	389
例	dnaPortNum: 389

#### 4.10.17. dnaRemainingValues

この属性には、残りの値と、エントリーに割り当てるサーバーで使用できる値の数が含まれます。

パラメーター	説明
エントリー DN	dnaHostname= <b>host_name</b> +dnaPortNum= <b>port_number</b> ,ou=ranges,dc=example,dc=com
構文	整数
Valid Range	任意の整数
デフォルト値	None
例	dnaRemainingValues: 1000

#### 4.10.18. dnaRemoteBindCred

Replication Manager のパスワードを指定します。認証が必要な **dnaRemoteBindMethod** 属性にバインドメソッドを設定する場合は、**cn=config** エントリーの下のプラグイン設定エントリーのレプリケーションデプロイメント内のすべてのサーバーに **dnaRemoteBindDN** および **dnaRemoteBindCred** パラメーターを追加で設定します。

パラメーターをプレーンテキストで設定します。値は、保存される前に自動的に AES 暗号化されません。

変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	cn= <b>DNA_config_entry</b> ,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
構文	DirectoryString {AES} encrypted_password
有効な値	有効な AES 暗号化パスワード。
デフォルト値	
例	dnaRemoteBindCred: {AES-TUhNRONTcUdTSWizRFFFRkRUQm1NRVVHQ1NxR1NJYjNEUUVGRERBNEJDUmxObUk0WXpjM1I5MHdaVE5rTXpZNA0KtnkxaE9XSmhORGRoT0MwMk1ESmpNV014TUFBQ0FRSUNBU0F3Q2dZSUtvWklodmNOQWdjd0hRWUpZSVpJQVdVRA0KQkFFcUJCQk5KbUFDUWFOMHIITWdsUVp3QjBJOQ==}bBR3On6cBmw0DdhcRx826g==

#### 4.10.19. dnaRemoteBindDN

Replication Manager DN を指定します。認証が必要な **dnaRemoteBindMethod** 属性にバインドメソッ

ドを設定する場合は、**cn=config** エントリーの下でのプラグイン設定のレプリケーションデプロイメント内のすべてのサーバーに **dnaRemoteBindDN** および **dnaRemoteBindCred** パラメーターを追加で設定します。

変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	cn= <b>DNA_config_entry</b> ,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
構文	DirectoryString
有効な値	有効な Replication Manager DN。
デフォルト値	
例	dnaRemoteBindDN: cn=replication manager,cn=config

#### 4.10.20. dnaRemoteBindMethod

リモートバインドメソッドを指定します。認証が必要な この属性にバインドメソッドを設定する場合は、**cn=config** エントリーの下でのプラグイン設定エントリーのレプリケーションデプロイメント内のすべてのサーバーに **dnaRemoteBindDN** および **dnaRemoteBindCred** パラメーターを追加で設定します。

変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	dnaHostname= <b>host_name</b> +dnaPortNum= <b>port_number</b> ,ou=ranges,dc=example,dc=com
構文	DirectoryString
有効な値	<b>SIMPLE</b>   <b>SSL</b>   <b>SASL/GSSAPI</b>   <b>SASL/DIGEST-MD5</b>
デフォルト値	
例	dnaRemoteBindMethod: SIMPLE

#### 4.10.21. dnaRemoteConnProtocol

リモート接続プロトコルを指定します。

変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	dnaHostname= <b>host_name</b> +dnaPortNum= <b>port_number</b> ,ou=ranges,dc=example,dc=com
構文	DirectoryString
有効な値	<b>LDAP</b> 、 <b>SSL</b> 、または <b>TLS</b>
デフォルト値	
例	dnaRemoteConnProtocol: LDAP

#### 4.10.22. dnaSecurePortNum

この属性は、**dnaHostname** で特定されたホストへの接続に使用するセキュアな (TLS) ポート番号を指定します。

パラメーター	説明
エントリー DN	dnaHostname= <b>host_name</b> +dnaPortNum= <b>port_number</b> ,ou=ranges,dc=example,dc=com
構文	整数
Valid Range	0 から 65535
デフォルト値	636
例	dnaSecurePortNum: 636

### 4.11. LINKED ATTRIBUTES 属性プラグインの属性

多くの場合、エントリーには相互に固有の関係があります (マネージャーと従業員、ドキュメントエントリーとその作成者、または特別なグループとグループメンバーなど)。これらの関係を反映する属性は存在しますが、これらの属性は各エントリーで手動で追加および更新する必要があります。そのため、これらのエントリーの関係が不明確、古くなっている、または欠落している、気まぐれに一貫性のないディレクトリーデータのセットが発生する可能性があります。

リンク属性プラグインでは、エントリーに設定されている属性で、自動的に別のエントリー内の別の属性を更新できます。最初の属性には、更新するエントリーを参照する DN 値があります。2 番目のエントリー属性には、1 番目のエントリーへのバックポイントである DN 値もあります。ユーザーによって設定されるリンク属性と、影響を受けるエントリーの動的に更新されるマネージド属性は、どちらもリンク属性プラグインインスタンスの管理者によって定義されます。

概念として、これは、MemberOf プラグインがグループエントリーの **member** 属性を使用してユーザーエントリーの **memberOf** 属性を設定する方法と似ています。リンク属性プラグインの場合のみ、すべてのリンク/管理属性はユーザー定義であり、プラグインの複数のインスタンスが存在する可能性があります。それぞれが異なるリンク管理関係を反映します。

属性リンクには、以下の2つの注意点があります。

- link 属性とマネージド属性の両方に DN を値として指定する必要があります。link 属性の DN は、管理属性を追加するエントリーを参照します。管理属性には、リンクしたエントリー DN が値として含まれています。
- 管理属性は多値である必要があります。それ以外の場合は、複数のリンク属性が同じ管理エントリーを参照すると、管理属性値は正確に更新されません。

#### 4.11.1. linkScope

これにより、プラグインのスコープが制限されるため、特定のサブツリーまたは接尾辞でのみ動作します。範囲が指定されていない場合、プラグインはディレクトリーツリーの一部を更新します。

パラメーター	説明
エントリー DN	cn= <b>plugin_instance</b> ,cn=Linked Attributes,cn=plugins,cn=config
Valid Range	任意の DN
デフォルト値	なし
構文	DN
例	linkScope: ou=People,dc=example,dc=com

#### 4.11.2. linkType

これにより、ユーザー管理属性が設定されます。この属性はユーザーが変更および維持し、この属性値が変更すると、リンクされた属性がターゲットエントリーで自動的に更新されます。

パラメーター	説明
エントリー DN	cn= <b>plugin_instance</b> ,cn=Linked Attributes,cn=plugins,cn=config
Valid Range	Directory Server の属性
デフォルト値	なし
構文	DirectoryString
例	linkType: directReport

#### 4.11.3. managedType

これにより、マネージドまたはプラグインの管理属性が設定されます。この属性は、リンク属性プラグインインスタンスによって動的に管理されます。マネージドの属性に変更が加えられるたびに、プラグインは対象のエントリーにあるリンク属性をすべて更新します。

パラメーター	説明
エントリー DN	cn= <b>plugin_instance</b> ,cn=Linked Attributes,cn=plugins,cn=config
Valid Range	Directory Server の属性
デフォルト値	なし
構文	DN
例	managedType: manager

## 4.12. MANAGED ENTRIES プラグインの属性

一意の状況では、別のエントリーの作成時にエントリーが自動的に作成される場合に便利です。たとえば、新規ユーザーの作成時に特定のグループエントリーを作成して、POSIX 統合の一部にすることができます。Managed Entries プラグインの各インスタンスは、2つの領域を特定します。

- プラグインのスコープ。対応するマネージドエントリーを必要とするエントリーを識別するために使用するサブツリーと検索フィルターを意味します。
- 管理エントリーがどのようになるかを定義するテンプレートエントリー

### 4.12.1. managedBase

この属性は、管理エントリーを作成するサブツリーを設定します。これは、ディレクトリーツリーの任意のエントリーにすることができます。

パラメーター	説明
エントリー DN	cn= <b>instance_name</b> ,cn=Managed Entries Plugin,cn=plugins,cn=config
有効な値	Directory Server のサブツリー
デフォルト値	なし
構文	DirectoryString
例	managedBase: ou=groups,dc=example,dc=com

### 4.12.2. managedTemplate

この属性は、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、ディレクトリツリーのどこにでも配置できます。ただし、レプリケーション内のすべてのサプライヤーとコンシューマーが同じテンプレートを使用するように、このエントリーをレプリケートされた接尾辞に含めることを推奨します。

管理対象エントリーテンプレートの作成に使用される属性は、[Red Hat Directory Server の設定、コマンド、およびファイルリファレンス](#) で説明されています。

パラメーター	説明
エントリー DN	cn= <b>instance_name</b> ,cn=Managed Entries Plugin,cn=plugins,cn=config
有効な値	<b>mepTemplateEntry</b> オブジェクトクラスの Directory Server エントリー
デフォルト値	なし
構文	DirectoryString
例	managedTemplate: cn=My Template,ou=Templates,dc=example,dc=com

### 4.12.3. originFilter

この属性は、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。フィルターを使用すると、マネージドエントリーの動作を特定のタイプやエントリーのサブセットに限定できます。構文は、通常の検索フィルターと同じです。

パラメーター	説明
エントリー DN	cn= <b>instance_name</b> ,cn=Managed Entries Plugin,cn=plugins,cn=config
有効な値	有効な LDAP フィルター
デフォルト値	なし
構文	DirectoryString
例	originFilter: objectclass=posixAccount

### 4.12.4. originScope

この属性は、プラグインモニターのエントリーを確認するために使用する検索の範囲を設定します。スコープのサブツリー内に新規エントリーが作成されると、Managed Entries プラグインはこれに対応する新しい管理エントリーを作成します。

パラメーター	説明
エントリー DN	cn= <b>instance_name</b> ,cn=Managed Entries Plugin,cn=plugins,cn=config
有効な値	Directory Server のサブツリー
デフォルト値	なし
構文	DirectoryString
例	originScope: ou=people,dc=example,dc=com

## 4.13. MEMBEROF プラグイン属性

グループメンバーシップは、**member** などの属性を使用してグループエントリー内で定義されます。**member** 属性を検索すると、グループのすべてのメンバーを簡単にリスト表示できます。ただし、グループメンバーシップはメンバーのユーザーエントリーに反映されないため、ユーザーのエントリーを参照してユーザーが所属するグループに指示することはできません。

MemberOf プラグインは、グループメンバーのグループメンバーシップをメンバーの個別のディレクトリーエントリーと同期します。これは、グループエントリー内の特定のメンバー属性 (**member** など) に変更を識別し、メンバーのユーザーエントリーの特定の属性にメンバーシップの変更を書き込むことです。

### 4.13.1. cn

プラグインインスタンスの名前を設定します。

パラメーター	説明
エントリー DN	cn=MemberOf Plugin,cn=plugins,cn=config
有効な値	有効な文字列
デフォルト値	
構文	DirectoryString
例	cn: MemberOf プラグインインスタンスの例

### 4.13.2. memberOfAllBackends

この属性は、ユーザーエントリーまたは利用可能なすべての接尾辞のローカル接尾辞を検索するかどうかを指定します。これは、複数のデータベースにユーザーを分散させるディレクトリーツリーで適し、グループメンバーシップを包括的かつ一貫して評価できるようにします。

パラメーター	説明
エントリー DN	cn=MemberOf Plugin,cn=plugins,cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	memberOfAllBackends: on

### 4.13.3. memberOfAttr

この属性は、グループメンバーシップを反映するために Directory Server が管理するユーザーエントリーの属性を指定します。MemberOf プラグインは、メンバーのディレクトリーエントリーで指定された属性の値を生成します。ユーザーが属するグループごとに個別の属性があります。

パラメーター	説明
エントリー DN	cn=MemberOf Plugin,cn=plugins,cn=config
Valid Range	Directory Server の属性
デフォルト値	memberOf
構文	DirectoryString
例	memberOfAttr: memberOf

### 4.13.4. memberOfAutoAddOC

**memberOf** プラグインで **memberOf** 属性をユーザーに追加できるようにするには、ユーザーオブジェクトにこの属性を許可するオブジェクトクラスが含まれている必要があります。エントリーに **memberOf** 属性を許可するオブジェクトクラスがない場合、**memberOf** プラグインは **memberOfAutoAddOC** パラメーターにリスト表示されているオブジェクトクラスを自動的に追加します。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=MemberOf Plugin,cn=plugins,cn=config
有効な値	Directory Server のオブジェクトクラス
デフォルト値	nsMemberOf

パラメーター	説明
構文	DirectoryString
例	memberOfAutoAddOC: nsMemberOf

#### 4.13.5. memberOfEntryScope

複数のバックエンドまたは複数のネストされた接尾辞を設定した場合、複数值の **memberOfEntryScope** パラメーターを使用すると、**MemberOf** プラグインが機能する接尾辞を設定できます。パラメーターが設定されていない場合、プラグインはすべての接尾辞で機能します。**memberOfEntryScopeExcludeSubtree** パラメーターに設定した値は、**memberOfEntryScope** に設定された値よりも優先順位が高くなります。

詳細は、[Directory Server 管理ガイド](#)の該当するセクションを参照してください。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=MemberOf Plugin,cn=plugins,cn=config
Valid Range	Directory Server エントリー DN。
デフォルト値	
構文	DirectoryString
例	memberOfEntryScope: ou=people,dc=example,dc=com

#### 4.13.6. memberOfEntryScopeExcludeSubtree

複数のバックエンドまたは複数のネストされた接尾辞を設定した場合、複数值の **memberOfEntryScopeExcludeSubtree** パラメーターを使用すると、**MemberOf** プラグインが除外する接尾辞を設定できます。**memberOfEntryScopeExcludeSubtree** パラメーターに設定した値は、**memberOfEntryScope** に設定された値よりも優先順位が高くなります。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。

詳細は、[Directory Server 管理ガイド](#)の該当するセクションを参照してください。

この設定を有効にするためにサーバーを再起動する必要はありません。

パラメーター	説明
エントリー DN	cn=MemberOf Plugin,cn=plugins,cn=config
Valid Range	Directory Server エントリー DN。

パラメーター	説明
デフォルト値	
構文	DirectoryString
例	memberOfEntryScopeExcludeSubtree: ou=sample,dc=example,dc=com

### 4.13.7. memberOfGroupAttr

この属性は、グループメンバーの DN 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指定します。デフォルトでは、これは **member** 属性ですが、**uniquemember** や **member** など、DN 値を含む任意のメンバーシップ関連属性にすることができます。



#### 注記

**memberOfGroupAttr** 値には任意の属性を使用できますが、MemberOf プラグインは、ターゲット属性の値にメンバーエントリーの DN が含まれる場合にのみ機能します。たとえば、**member** 属性にはメンバーのユーザーエントリーの DN が含まれます。

```
member: uid=jsmith,ou=People,dc=example,dc=com
```

一部のメンバー関連の属性には、**memberURL** 属性などの DN が含まれていないものもあります。この属性は **memberOfGroupAttr** の値として機能しません。**memberURL** 値は URL で、DN 以外の値は MemberOf プラグインでは機能しません。

パラメーター	説明
エントリー DN	cn=MemberOf Plugin,cn=plugins,cn=config
Valid Range	Directory Server の属性
デフォルト値	member
構文	DirectoryString
例	memberOfGroupAttr: member

## 4.14. ATTRIBUTE UNIQUENESS プラグインの属性

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。

### 4.14.1. cn

**Attribute Uniqueness** プラグイン設定レコードの名前を設定します。任意の文字列を使用できますが、RedHat では設定レコードに **attribute\_name** 属性の一意性 という名前を付けることを推奨します。

パラメーター	説明
エントリー DN	cn=attribute_uniqueness_configuration_record_name,cn=plugins,cn=config
有効な値	有効な文字列
デフォルト値	なし
構文	DirectoryString
例	cn: mail Attribute Uniqueness

#### 4.14.2. uniqueness-attribute-name

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。

パラメーター	説明
エントリー DN	cn=attribute_uniqueness_configuration_record_name,cn=plugins,cn=config
有効な値	有効な属性名
デフォルト値	なし
構文	DirectoryString
例	uniqueness-attribute-name: mail

#### 4.14.3. uniqueness-subtrees

プラグインが属性の値を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。

パラメーター	説明
エントリー DN	cn=attribute_uniqueness_configuration_record_name,cn=plugins,cn=config
有効な値	有効なサブツリー DN
デフォルト値	なし

パラメーター	説明
構文	DirectoryString
例	uniqueness-subtrees: ou=Sales,dc=example,dc=com

#### 4.14.4. uniqueness-across-all-subtrees

有効 (**on**) の場合、プラグインは属性がすべてのサブツリーセットで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。

パラメーター	説明
エントリー DN	cn=attribute_uniqueness_configuration_record_name,cn=plugins,cn=config
有効な値	on   off
デフォルト値	off
構文	DirectoryString
例	uniqueness-across-all-subtrees: off

#### 4.14.5. uniqueness-top-entry-oc

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。見つからない場合は、ディレクトリーツリーのルートまで、次に高いレベルのエントリーで検索が続行されます。オブジェクトクラスが見つかった場合、Directory Server はこのサブツリーで **uniqueness-attribute-name** に設定された属性の値が一意であることを確認します。

パラメーター	説明
エントリー DN	cn=attribute_uniqueness_configuration_record_name,cn=plugins,cn=config
有効な値	有効なオブジェクトクラス
デフォルト値	なし
構文	DirectoryString
例	uniqueness-top-entry-oc: nsContainer

#### 4.14.6. uniqueness-subtree-entries-oc

任意で、**uniqueness-top-entry-oc** パラメーターを使用する場合、**Attribute Uniqueness** プラグインは、このパラメーターに設定されているオブジェクトクラスが含まれている場合に属性が一意かどうかのみを検証するように設定できます。

パラメーター	説明
エントリー DN	cn=attribute_uniqueness_configuration_record_name,cn=plugins,cn=config
有効な値	有効なオブジェクトクラス
デフォルト値	なし
構文	DirectoryString
例	uniqueness-subtree-entries-oc: inetOrgPerson

## 4.15. POSIX WINSYNC API プラグインの属性

デフォルトでは、POSIX 関連の属性は Active Directory と Red Hat Directory Server の間で同期されません。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**、**homeDirectory** など) は、ActiveDirectory と DirectoryServer の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。

このプラグインはデフォルトで無効になっており、Posix 属性を ActiveDirectory エントリーから DirectoryServer エントリーに同期する前に有効にする必要があります。

### 4.15.1. posixWinsyncCreateMemberOfTask

この属性は、同期されたユーザーのグループメンバーシップを更新するために、同期実行後すぐに memberOf fix-up タスクを実行するかどうかを設定します。memberOf 修正タスクではリソースを集中的に使用し、頻繁に実行する場合はパフォーマンスの問題が発生する可能性があるため、デフォルトでは無効になっています。



パラメーター	説明
エントリー DN	cn=Posix Winsync API Plugin,cn=plugins,cn=config
Valid Range	true   false
デフォルト値	false
例	posixWinsyncCreateMemberOfTask: false

#### 4.15.2. posixWinsyncLowerCaseUID

この属性は、UID 値を **memberUID** 属性に小文字で格納する (および必要に応じて変換する) かどうかを設定します。

パラメーター	説明
エントリー DN	cn=Posix Winsync API Plugin,cn=plugins,cn=config
Valid Range	true   false
デフォルト値	false
例	posixWinsyncLowerCaseUID: false

#### 4.15.3. posixWinsyncMapMemberUID

この属性は、Active Directory グループの **memberUID** 属性を Directory Server グループの **uniqueMember** 属性にマッピングするかどうかを設定します。

パラメーター	説明
エントリー DN	cn=Posix Winsync API Plugin,cn=plugins,cn=config
Valid Range	true   false
デフォルト値	true
例	posixWinsyncMapMemberUID: false

#### 4.15.4. posixWinsyncMapNestedGrouping

**posixWinsyncMapNestedGrouping** パラメーターは、Active Directory POSIX グループの **memberUID** 属性が変更されるときにネスト化されたグループが更新されるかどうかを管理します。ネスト化されたグループの更新は、5つのレベルの深さに対応します。

パラメーター	説明
エントリー DN	cn=Posix Winsync API Plugin,cn=plugins,cn=config
Valid Range	true   false
デフォルト値	false
例	posixWinsyncMapNestedGrouping: false

#### 4.15.5. posixWinsyncMsSFUSchema

この属性は、Active Directory から Posix 属性を同期するときに Unix 3.0 (msSFU30) スキーマの古い Microsoft System Services にするかどうかを設定します。デフォルトでは、POSIX Winsync API プラグインは最新の Active Directory サーバーに Posix スキーマを使用します (2005、2008、およびそれ以降)。最新の Active Directory Posix スキーマと、Windows Server 2003 以前の Windows サーバーで使用される Posix スキーマには若干の違いがあります。Active Directory ドメインが古いスタイルスキーマを使用している場合は、代わりに古いスタイルスキーマを使用できます。

パラメーター	説明
エントリー DN	cn=Posix Winsync API Plugin,cn=plugins,cn=config
Valid Range	true   false
デフォルト値	false
例	posixWinsyncMsSFUSchema: true

### 4.16. RETRO CHANGELOG プラグインの属性

Directory Server では、2 種類の changelogs が維持されます。**changelog** と呼ばれる最初のタイプはマルチサ upplier レプリケーションに使用され、2 つ目の changelog は **retro changelog** と呼ばれるプラグインで、Directory Server 4.x バージョンとのアプリケーション互換性を維持するために LDAP クライアントが使用することを目的としています。

この Retro Changelog プラグインは、サプライヤーサーバーに加えられた変更を記録するために使用されます。サプライヤーサーバーのディレクトリーが変更されると、エントリーは、以下の両方が含まれる Retro Changelog に書き込まれます。

- 変更を一意に識別する数字。この数は、changelog の他のエントリーに関連して順次行われず。
- 変更アクション。つまりディレクトリーの変更内容を正確に行う必要があります。

**cn=changelog** 接尾辞への検索を使用して Directory Server に加えられた変更がアクセスされることが Retro Changelog プラグインを介して実行されます。

#### 4.16.1. isReplicated

このオプション属性は、そのサーバーで新たに変更が加えられているかどうか、別のサーバーから複製されたかどうかに関わらず、changelog の変更を示すフラグを設定します。

パラメーター	説明
OID	2.16.840.1.113730.3.1.2085
エントリー DN	cn=Retro Changelog Plugin,cn=plugins,cn=config
有効な値	true   false
デフォルト値	なし
構文	Boolean
例	isReplicated: true

#### 4.16.2. nsslapd-attribute

この属性は、retro changelog エントリーに含める必要のある別の Directory Server 属性を明示的に指定します。

通常、操作属性およびその他のタイプの属性は retro changelog から除外されますが、サードパーティーアプリケーションで changelog データを使用するにはこれらの属性が存在する必要がある場合があります。これは、**nsslapd-attribute** パラメーターを使用して retro changelog プラグイン設定に属性をリスト表示することで行います。

**nsslapd-attribute** 値内で指定した属性に任意のエイリアスを指定することもできます。

**nsslapd-attribute: attribute:alias**

属性のエイリアスを使用すると、retro changelog レコードを使用する外部サーバーまたはアプリケーションの他の属性との競合を避けることができます。



#### 注記

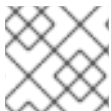
**nsslapd-attribute** 属性の値を **isReplicated** に設定することは、変更がローカルサーバーで行われていたか (つまり変更が元の変更かどうか) か、変更がサーバーに複製されたかを示す方法です。

パラメーター	説明
エントリー DN	cn=Retro Changelog Plugin,cn=plugins,cn=config
有効な値	有効なディレクトリー属性 (標準またはカスタム)
デフォルト値	なし

パラメーター	説明
構文	DirectoryString
例	nsslapd-attribute: nsUniqueid: uniqueID

### 4.16.3. nsslapd-changelogdir

この属性は、プラグインの初回実行時に changelog データベースが作成されるディレクトリーの名前を指定します。デフォルトでは、データベースは `/var/lib/dirsrv/slapd-instance/changelogdb` 下の他のすべてのデータベースに保存されます。



#### 注記

パフォーマンス上の理由から、このデータベースを異なる物理ディスクに保存します。

この属性の変更を反映するには、サーバーを再起動する必要があります。

パラメーター	説明
エントリー DN	cn=Retro Changelog Plugin,cn=plugins,cn=config
有効な値	ディレクトリーへの有効なパス
デフォルト値	なし
構文	DirectoryString
例	nsslapd-changelogdir: /var/lib/dirsrv/slapd-instance/changelogdb

### 4.16.4. nsslapd-changelogmaxage (Changelog 最大経過時間)

この属性は、changelog のエントリーの最大期間を指定します。changelog には各ディレクトリーの変更に対するレコードが含まれ、コンシューマーサーバーの同期時に使用されます。各レコードにはタイムスタンプが含まれます。この属性に指定した値よりも古いタイムスタンプを持つレコードはすべて削除されます。**nsslapd-changelogmaxage** 属性が存在しない場合に、変更ログレコードに有効期限はありません。



#### 注記

最大期間よりも長い合意がある場合は、期限切れの changelog レコードは削除されません。

パラメーター	説明
エントリー DN	cn=Retro Changelog Plugin,cn=plugins,cn=config

パラメーター	説明
Valid Range	0 (エントリーは経過時間に応じて削除されない) から最大 32 ビットの整数値 (2147483647)
デフォルト値	7d
構文	DirectoryString Integer AgeID  AgeID では、 <b>s (S)</b> は秒、 <b>m (M)</b> は分、 <b>h (H)</b> は時間、 <b>d (D)</b> は日、 <b>w (W)</b> は週を示します。
例	nsslapd-changelogmaxage: 30d

#### 4.16.5. nsslapd-exclude-attrs

**nsslapd-exclude-attrs** パラメーターは、レトロ変更ログデータベースから除外する属性名を保管します。複数の属性を除外するには、除外する属性ごとに1つの **nsslapd-exclude-attrs** パラメーターを追加します。

パラメーター	説明
エントリー DN	cn=Retro Changelog Plugin,cn=plugins,cn=config
有効な値	有効な属性名
デフォルト値	なし
構文	DirectoryString
例	nsslapd-exclude-attrs: <b>example</b>

#### 4.16.6. nsslapd-exclude-suffix

**nsslapd-exclude-suffix** パラメーターは、レトロ変更ログデータベースから除外する接尾辞を保管します。パラメーターを複数回追加して、複数の接尾辞を除外できます。

パラメーター	説明
エントリー DN	cn=Retro Changelog Plugin,cn=plugins,cn=config
有効な値	有効な属性名
デフォルト値	なし
構文	DirectoryString

パラメーター	説明
例	nsslapd-exclude-suffix: <b>ou=demo,dc=example,dc=com</b>

## 4.17. ROOTDN アクセス制御プラグインの属性

ルート DN である cn=Directory Manager は、通常のユーザーデータベースの外部で定義される特別なユーザーエントリーです。通常のアクセス制御ルールは root DN には適用されませんが、root ユーザーの強力な性質により、何らかのアクセス制御ルールを root ユーザーに適用することが有益です。

RootDN アクセス制御プラグインは、root ユーザーの通常のアクセス制御 (ホストおよび IP アドレスの制限、時刻の制限、および曜日の制限) を設定します。

このプラグインはデフォルトで無効になっています。

### 4.17.1. rootdn-allow-host

これにより、root ユーザーが Directory Server にアクセスするのに使用できるホストを完全修飾ドメイン名で設定します。リストされていないホストは暗黙的に拒否されます。

ワイルドカードは許可されています。

この属性は複数回使用して、複数のホスト、ドメイン、またはサブネットを指定できます。

パラメーター	説明
エントリー DN	cn=RootDN Access Control Plugin,cn=plugins,cn=config
Valid Range	ワイルドカードのアスタリスク (*) を含む有効なホスト名またはドメイン
デフォルト値	なし
構文	DirectoryString
例	rootdn-allow-host: *.example.com

### 4.17.2. rootdn-allow-ip

これにより、root ユーザーが Directory Server へのアクセスに使用できるマシンの IPv4 または IPv6 のいずれかの IP アドレスが設定されます。リストされていない IP アドレスは暗黙的に拒否されます。

ワイルドカードは許可されています。

この属性は複数回使用して、複数のアドレス、ドメイン、またはサブネットを指定できます。

パラメーター	説明
エントリー DN	cn=RootDN Access Control Plugin,cn=plugins,cn=config
Valid Range	ワイルドカードのアスタリスク (*) を含む、有効な IPv4 アドレスまたは IPv6 アドレス。
デフォルト値	なし
構文	DirectoryString
例	rootdn-allow-ip: 192.168..

### 4.17.3. rootdn-close-time

これは、root ユーザーが Directory Server にアクセスできる期間または範囲の一部を設定します。これは、root ユーザーが Directory Server へのアクセスが許可されなくなった場合に、時間ベースのアクセスの終了時に設定されます。

これは、**rootdn-open-time** 属性と組み合わせて使用されます。

パラメーター	説明
エントリー DN	cn=RootDN Access Control Plugin,cn=plugins,cn=config
Valid Range	24 時間形式で有効な時間。
デフォルト値	なし
構文	整数
例	rootdn-close-time: 1700

### 4.17.4. rootdn-days-allowed

これにより、root ユーザーが Directory Server にアクセスするのに使用できる日数のコンマ区切りリストが提供されます。リストされている日は暗黙的に拒否されます。これは、**rootdn-close-time** および **rootdn-open-time** とともに使用して、時間ベースのアクセスと曜日を組み合わせることができます。または、単独で使用することもできます (許可された日にすべての時間が許可されます)。

パラメーター	説明
エントリー DN	cn=RootDN Access Control Plugin,cn=plugins,cn=config

パラメーター	説明
有効な値	* Sun * Mon * Tue * Wed * Thu * Fri * Sat
デフォルト値	なし
構文	DirectoryString
例	rootdn-days-allowed: Mon, Tue, Wed, Thu, Fri

#### 4.17.5. rootdn-deny-ip

これにより、root ユーザーが Directory Server にアクセスすることができない マシンの IPv4 または IPv6 のいずれかの IP アドレスが設定されます。リストされていない IP アドレスは暗黙的に拒否されます。



#### 注記

deny ルールは allow ルールよりも優先されるため、IP アドレスが **rootdn-allow-ip** および **rootdn-deny-ip** 属性の両方にリスト表示されている場合、アクセスは拒否されます。

ワイルドカードは許可されています。

この属性は複数回使用して、複数のアドレス、ドメイン、またはサブネットを指定できます。

パラメーター	説明
エントリー DN	cn=RootDN Access Control Plugin,cn=plugins,cn=config
Valid Range	ワイルドカードのアスタリスク (*) を含む、有効な IPv4 アドレスまたは IPv6 アドレス。
デフォルト値	なし
構文	DirectoryString
例	rootdn-deny-ip: 192.168.0.0



### 4.17.6. rootdn-open-time

これは、root ユーザーが Directory Server にアクセスできる期間または範囲の一部を設定します。これは、時間ベースのアクセスがいつ **開始** するかを設定します。

これは、**rootdn-close-time** 属性と組み合わせて使用されます。

パラメーター	説明
エントリー DN	cn=RootDN Access Control Plugin,cn=plugins,cn=config
Valid Range	24 時間形式で有効な時間。
デフォルト値	なし
構文	整数
例	rootdn-open-time: 0800

## 4.18. REFERENTIAL INTEGRITY プラグインの属性

Referential Integrity により、ディレクトリー内のエントリーに対して更新または削除操作を実行すると、削除または更新されたエントリーを参照するエントリーの情報もサーバーによって更新されます。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、Referential Integrity 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。

### 4.18.1. nsslapd-pluginAllowReplUpdates

Referential Integrity は、非常にリソースを必要とする手順になる可能性があります。そのため、マルチサプライヤーレプリケーションを設定した場合に、Referential Integrity プラグインはデフォルトでレプリケートされた更新を無視します。ただし、Referential Integrity プラグインを有効にできない場合や、プラグインを使用できない場合があります。

たとえば、複製トポロジーのサプライヤーの1つが Active Directory ですが、Active Directory は Referential Integrity をサポートしていません (詳細は、[Windows の同期](#) の章を参照)。このような場合、**nsslapd-pluginAllowReplUpdates** 属性を使用して、別のサプライヤーの Referential Integrity プラグインが複製された更新を処理できるようにすることができます。



#### 重要

マルチサプライヤーレプリケーショントポロジーでは、1つのサプライヤーのみが **nsslapd-pluginAllowReplUpdates** 属性値を **on** にする必要があります。そうしないと、レプリケーションエラーが発生する可能性があります。問題を解決するには完全な初期化が必要になります。一方、Referential Integrity プラグインは、可能な限りすべてのサプライで有効にしておく必要があります。

パラメーター	説明
--------	----

パラメーター	説明
エントリー DN	cn=referential integrity postoperation,cn=plugins,cn=config
有効な値	on/off
デフォルト値	off
構文	Boolean
例	nsslapd-pluginAllowReplUpdates: on

## 第5章 ディレクトリーエントリースキーマのリファレンス

### 5.1. DIRECTORY SERVER スキーマについて

本章では、ディレクトリースキーマの基本概念の概要を説明し、スキーマを記述するファイルをリスト表示します。オブジェクトクラス、属性、オブジェクト識別子 (OID) を記述し、サーバースキーマとスキーマチェックの拡張を簡単に説明します。

#### 5.1.1. スキーマ定義

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設定されます。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。

基本的に、スキーマファイルは、作成できるエントリーの種類 (オブジェクトクラス) と、それらのエントリーを記述する方法 (属性) のリストです。スキーマは、オブジェクトクラスおよび属性を **定義** します。スキーマは、属性値に含まれる形式 (属性の **構文**) と、その属性のインスタンスが1つだけであるかどうかも定義します。

追加のスキーマファイルを DirectoryServer 設定に追加してサーバーにロードできるため、スキーマはカスタマイズ可能であり、必要に応じて拡張できます。

オブジェクトクラス、属性、および Directory Server がスキーマの使用方法についての詳細は、**デプロイメントガイド**を参照してください。



#### 警告

スキーマ定義に文字数が多すぎると、Directory Server は起動に失敗します。これらの場所では、LDAP 標準で、NAME キーワードと属性タイプの名前など、ゼロまたは多数のスペースを使用できるようにするスペースを1つだけ使用します。

#### 5.1.1.1. 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 (**groupOfUniqueNames**)、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)、および必須属性のリスト (**MUST**) および許可される属性のリスト (**MAY**) が続きます。

これは、[例5.1「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5.1 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 2256' )
```

#### 5.1.1.1.1. 必須および許可される属性

すべてのオブジェクトクラスは、多数の必須属性と許可される属性を定義します。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5.1「個人のオブジェクトクラススキーマエントリー」**と同様に、**person** オブジェクトクラスには **cn**、**sn**、および **objectClass** 属性が必要で、**description** (**seeAlso**、**TelephoneNumber**、および **userPassword** 属性) を許可します。



#### 注記

すべてのエントリーには、エントリーに割り当てられたオブジェクトクラスをリスト表示する **objectClass** 属性が必要です。

#### 5.1.1.1.2. オブジェクトクラスの継承

エントリーには、複数のオブジェクトクラスを含めることができます。たとえば、個人のエントリーは **person** オブジェクトクラスで定義されますが、同じユーザーが **inetOrgPerson** および **organizationalPerson** オブジェクトクラスの属性で記述することもできます。

さらに、オブジェクトクラスは階層的に実行できます。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** オブジェクトクラスです。

サーバーのオブジェクトクラス構造は、特定のエントリーに必要な属性と許可される属性のリストを決定します。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
```

**inetOrgPerson** オブジェクトクラスがエントリーに割り当てられている場合、エントリーは上位オブジェクトクラスから必要な属性および許可される属性を自動的に継承します。

#### 5.1.1.2. 属性

ディレクトリーエントリーは、属性とその値で設定されます。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、属性は **attributetypes** 行で識別され、次に OID、名前、説明、構文 (値に使用できる形式)、任意で属性が単一の値または複数の値であるかどうか、および属性が定義されます。

これは、例5.2「説明属性スキーマエントリー」に示されています。

### 例5.2 説明属性スキーマエントリー

```
attributetypes: ( 2.5.4.13 NAME 'description' DESC 'Standard LDAP attribute type' SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN 'RFC 2256' )
```

一部の属性は省略できます。これらの略語は、属性定義の一部としてリストされています。

```
attributetypes: ( 2.5.4.3 NAME ( 'cn' 'commonName' ) ...
```

#### 5.1.1.2.1. 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 Console では、構文は分かりやすい名前参照されます。

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。

表5.1 サポートされる LDAP 属性構文

Name	OID	定義
Binary	1.3.6.1.4.1.1466.115.121.1.5	非推奨:代わりに Octet 文字列を使用してください。
Bit String	1.3.6.1.4.1.1466.115.121.1.6	'0101111101'B などのビットリング値。
Boolean	1.3.6.1.4.1.1466.115.121.1.7	許可される値が TRUE または FALSE の 2 つしかない属性の場合。
国文字列	1.3.6.1.4.1.1466.115.121.1.11	正確に 2 つの印刷可能な文字列文字に制限されている値の場合。たとえば、米国の場合は米国です。
DN	1.3.6.1.4.1.1466.115.121.1.12	識別名 (DN) である値の場合。

Name	OID	定義
配信方法	1.3.6.1.4.1.1466.115.121.1.14	情報の配信やエンティティへの問い合わせの推奨方法が含まれる値の場合。異なる値はドル記号 (\$) で区切ります。以下に例を示します。  [literal,subs="+quotes,verbatim"] ... telephone \$ physical ...
ディレクトリー文字列	1.3.6.1.4.1.1466.115.121.1.15	有効な UTF-8 文字列である値の場合。これらの値は、大文字と小文字を区別する場合があります。Directory String および関連する構文では、大文字と小文字の区別のないマッチングルールの両方を使用できます。
拡張されたガイド	1.3.6.1.4.1.1466.115.121.1.21	属性およびフィルターに基づく、複雑な検索パラメーターが含まれる値の場合。
Facsimile	1.3.6.1.4.1.1466.115.121.1.22	ファックス番号を含む値の場合。
Fax	1.3.6.1.4.1.1466.115.121.1.23	送信されるファックスのイメージを含む値の場合。
一般化時間	1.3.6.1.4.1.1466.115.121.1.24	印刷可能な文字列としてエンコードされる値の場合。タイムゾーンを指定する必要があります。GMT 時間を使用することを強く推奨します。
ガイド	1.3.6.1.4.1.1466.115.121.1.25	<b>廃止</b> 属性およびフィルターに基づく、複雑な検索パラメーターが含まれる値の場合。
IA5 String	1.3.6.1.4.1.1466.115.121.1.26	有効な文字列である値の場合。これらの値は、大文字と小文字を区別する場合があります。IA5 String および関連する構文では、大文字と小文字の区別のないマッチングルールの両方を使用できます。
整数	1.3.6.1.4.1.1466.115.121.1.27	整数の値。

Name	OID	定義
JPEG	1.3.6.1.4.1.1466.115.121.1.28	イメージデータが含まれる値の場合。
名前および任意の UID	1.3.6.1.4.1.1466.115.121.1.34	DN と (オプションの) 一意の ID の組み合わせ値を含む値の場合。
数値文字列	1.3.6.1.4.1.1466.115.121.1.36	数値とスペースの両方の文字列を含む値の場合。
OctetString	1.3.6.1.4.1.1466.115.121.1.40	バイナリーの値の場合は、バイナリー構文が置き換えられます。
Object Class Description	1.3.6.1.4.1.1466.115.121.1.37	オブジェクトクラス定義を含む値の場合。
OID	1.3.6.1.4.1.1466.115.121.1.38	OID 定義を含む値の場合。
住所	1.3.6.1.4.1.1466.115.121.1.41	<p><b>postal-address =dstring*</b>  <b>("\$dstring")</b> の形式でエンコードされた値の場合。以下に例を示します。</p> <p>[literal,subs="+quotes,verbatim"]  ...1234 Main St.\$Raleigh, NC  12345\$USA ...</p> <p>各 <b>dstring</b> コンポーネントは DirectoryString の値としてエンコードされます。バックスラッシュとドル文字は引用符で囲まれるため、行の区切り文字では間違いはなくなりました。多くのサーバーは、ポストアドレスを最大 30 文字までの 6 行に制限します。</p>
出力可能な文字列	1.3.6.1.4.1.1466.115.121.1.44	印刷可能な文字列を含む値の場合。
Space-Insensitive String	2.16.840.1.113730.3.7.1	スペースの区別のない文字列を含む値の場合:
TelephoneNumber	1.3.6.1.4.1.1466.115.121.1.50	電話番号の形式にある値国際形式で電話番号を使用することが推奨されます。
Teletex Terminal Identifier	1.3.6.1.4.1.1466.115.121.1.51	国際電話番号が含まれる値の場合。

Name	OID	定義
Telex Number	1.3.6.1.4.1.1466.115.121.1.52	テレックス端末のテレックス番号、国コード、および回答コードを含む値の場合。
URI		<b>http://</b> 、 <b>https://</b> 、 <b>ftp://</b> 、 <b>ldap://</b> 、 <b>ldaps://</b> などの文字列によって導入された URL 形式の値の場合。URI の動作は IA5 文字列と同じです。この構文についての詳細は、 <a href="#">RFC 4517</a> を参照してください。

### 5.1.1.2.2. 単一値および複数値の属性

デフォルトでは、ほとんどの属性は複数値です。つまり、エントリーに同じ属性を複数回追加できます。以下に例を示します。

```
dn: uid=jsmith,ou=marketing,ou=people,dc=example,dc=com
ou: marketing
ou: people
```

**cn** 属性、**tel** 属性、および **objectclass** 属性は、すべて複数の値を持つことができます。単一値である属性 (属性の1つのインスタンスのみを指定可能) は、単一の値のみを許可するようにスキーマに指定されます。たとえば、**uidNumber** は使用可能な値は1つしかないため、スキーマエントリーには **SINGLE-VALUE** という用語があります。属性が複数値の場合、値式はありません。

## 5.1.2. デフォルトの Directory Server スキーマファイル

Directory Server のテンプレートスキーマ定義は **/etc/dirsrv/schema** ディレクトリーに保存されます。これらのデフォルトのスキーマファイルは、新しい Directory Server インスタンスのスキーマファイルを生成します。各サーバーインスタンスには、**/etc/dirsrv/slaped- instance/schema** に独自のインスタンス固有のスキーマディレクトリーがあります。インスタンスディレクトリーのスキーマファイルは、そのインスタンスによってのみ使用されます。

ディレクトリースキーマを変更するには、インスタンス固有のスキーマディレクトリーに新しい属性と新しいオブジェクトクラスを作成します。デフォルトのスキーマは新規インスタンスの作成に使用され、各インスタンスには独自のスキーマファイルがあるため、各インスタンスの使用は若干異なるため、各インスタンスの使用を照合することができます。

Directory Server コンソールまたは LDAP コマンドを使用して追加されるカスタム属性は、**99user.ldif** ファイルに保存されます。その他のカスタムスキーマファイルは、各インスタンスの **/etc/dirsrv/slaped- instance/schema** ディレクトリーに追加できます。Red Hat Directory Server に同梱されている標準ファイルには変更を加えないでください。

Directory Server の情報やプランニングディレクトリースキーマに関する提案についての詳細は、**デプロイメントガイド**を参照してください。

### 表5.2 スキーマファイル



スキーマファイル	目的
00core.ldif	X.500 および LDAP 標準 (RFC) から推奨されるコアスキーマ。このスキーマは、インスタンス設定の Directory Server 自体で使用され、サーバーインスタンスを起動します。
01core389.ldif	X.500 および LDAP 標準 (RFC) から推奨されるコアスキーマ。このスキーマは、インスタンス設定の Directory Server 自体で使用され、サーバーインスタンスを起動します。
02common.ldif	エントリーを設定するために使用される Directory Server で定義された RFC 2256、LDAPv3、および標準スキーマの標準関連のスキーマ。
05rfc2927.ldif	RFC 2927 からのスキーマ MIME Directory Profile for LDAP Schema。
05rfc4523.ldif	X.509 証明書のスキーマ定義。
05rfc4524.ldif	LDAP/X.500 スキーマをコーディングします。
06inetorgperson.ldif	RFC 2798、RFC 2079、および RFC 1274 の一部からの inetOrgPerson スキーマ要素。
10rfc2307.ldif	RFC 2307 からのスキーマ LDAP をネットワーク情報サービスとして使用するためのアプローチ。
20subscriber.ldif	Directory Server-Nortel サブスクライバーの相互運用性の一般的なスキーマ要素。
25java-object.ldif	RFC 2713 のスキーマ Schema for Representing Java Objects in an LDAP Directory
28pilot.ldif	パイロット RFC、特に RFC 1274 からのスキーマで、新しいデプロイメントでの使用は推奨されなくなりました。
30ns-common.ldif	共通スキーマ。
50ns-admin.ldif	管理サーバーで使用されるスキーマ。
50ns-certificate.ldif	Red Hat 証明書システムで使用されるスキーマ。
50ns-directory.ldif	レガシー Directory Server 4.x サーバーによって使用されるスキーマ。

スキーマファイル	目的
50ns-mail.ldif	メールサーバーのスキーマ。
50ns-value.ldif	Directory Server のバリューアイテムのスキーマ。
50ns-web.ldif	Web サーバーのスキーマ。
60autofs.ldif	自動マウント設定のオブジェクトクラス。これは、NIS サーバーに使用される複数のスキーマファイルの1つです。
60eduperson.ldif	企業関連の人や組織エントリーのスキーマ要素。
60mozilla.ldif	Mozilla 関連のユーザープロファイルのスキーマ要素。
60nss-ldap.ldif	GSS-API サービス名のスキーマ要素。
60pam-plugin.ldif	ディレクトリーサービスを PAM モジュールと統合するためのスキーマ要素。
60pureftpd.ldif	FTP ユーザーアカウントを定義するためのスキーマ要素。
60rfc2739.ldif	カレンダーおよび vCard プロパティのスキーマ要素。
60rfc3712.ldif	プリンターを設定するためのスキーマ要素。
60sabayon.ldif	sabayon ユーザーエントリーを定義するためのスキーマ要素。
60sudo.ldif	sudo ユーザーおよびロールを定義するためのスキーマ要素。
60trust.ldif	NSS または PAM の信頼関係を定義するためのスキーマ要素。
99user.ldif	Directory Server コンソールから追加されるカスタムスキーマ要素。

### 5.1.3. オブジェクト識別子 (OID)

すべてのスキーマ要素には、属性やオブジェクトクラスなど、オブジェクト識別子 (OID) が割り当てられます。OID は、通常はドットで区切られた文字列として記述される整数のシーケンスです。すべてのカスタム属性とクラスは、X.500 および LDAP 標準に準拠する必要があります。



### 警告

スキーマ要素に OID が指定されていない場合、Directory Server は **ObjectClass\_name-oid** および **attribute\_name-oid** を自動的に使用します。ただし、数値 OID の代わりにテキスト OID を使用すると、クライアント、サーバーの相互運用性、およびサーバーの動作に問題が発生する可能性があるため、数値 OID を割り当てることを強く推奨します。

OID をビルドできます。ベース OID は、組織のすべてのスキーマ要素に使用されるルート番号で、そこからスキーマ要素を増やすことができます。たとえば、ベース OID は **1** になります。その後、属性に **1.1** を使用するため、新しい属性の OID は **1.1.x** です。オブジェクトクラスに **1.2** を使用するため、新しいオブジェクトクラスの OID は **1.2.x** です。

Directory Server 定義のスキーマ要素では、ベース OID は以下のようにになります。

- Netscape のベース OID は **2.16.840.1.113730** です。
- DirectoryServer のベース OID は **2.16.840.1.113730.3** です。
- Netscape で定義されたすべての属性には、ベース **OID2.16.840.1.113730.3.1** があります。
- Netscape で定義されたすべてのオブジェクトクラスには、基本 **OID2.16.840.1.113730.3.2** があります。

OID に関する詳細や接頭辞を要求する場合は、Internet Assigned Number Authority (IANA) Web サイト (<http://www.iana.org/>) を参照してください。

#### 5.1.4. スキーマの拡張

Directory Server スキーマには、ほとんどのディレクトリー要件を満たすために使用できる数百のオブジェクトクラスと属性が含まれています。このスキーマは、カスタムスキーマファイルを作成して、企業内のディレクトリーサービスの進化要件を満たす新しいオブジェクトクラスおよび属性で拡張できます。

スキーマに新しい属性を追加する場合、新しいオブジェクトクラスを作成してスキーマを含める必要があります。既存のオブジェクトクラスに新しい属性を追加すると、標準の LDAP スキーマに依存する既存の LDAP クライアントとの互換性が Directory Server の互換性が危険にさらされ、サーバーのアップグレード時に問題が発生する可能性があります。

サーバースキーマの拡張に関する詳細は、**デプロイメントガイド**を参照してください。

#### 5.1.5. スキーマチェック

**スキーマチェック**とは、Directory Server が LDIF を使用してインポートされたデータベースで作成、変更、またはデータベースですべてのエントリーをチェックし、スキーマファイルのスキーマ定義に準拠することを確認します。スキーマチェックでは、次の3つのことを確認します。

- エントリーで使用され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。

スキーマチェックが有効になっている Directory Server を実行する必要があります。スキーマチェックを有効にする方法は、[管理ガイド](#)を参照してください。

### 5.1.6. 構文の検証

**構文の検証**とは、Directory Server が属性の値が、その属性に必要な構文と一致することを確認することを意味します。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。

基本設定では、構文検証 (スキーマチェックなど) により、ディレクトリーの変更がチェックされ、属性値が必要な構文と一致することが確認され、構文に違反する変更が拒否されます。オプションで、構文違反に関する警告メッセージをログに記録し、変更を拒否するか、変更プロセスを成功できるように構文の検証を設定できます。

すべての構文は、DN を除く [RFC 4514](#) に対して検証されます。デフォルトでは、DN は [RFC 1779](#) または [RFC 2253](#) に対して検証されますが、[RFC 4514](#) よりは厳格ではありません。DN の厳密な検証を明示的に設定する必要があります。

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、[表5.1「サポートされる LDAP 属性構文」](#)に記載されるすべての属性構文を確認します。**未検証**の構文は以下のとおりです。

- Fax (バイナリー)
- OctetString (binary)
- JPEG (バイナリー)
- バイナリー (標準以外)
- スペースに依存しない文字列 (非標準)
- URI (標準以外)

構文検証が有効になっている場合、属性がエントリーに追加または変更されるたびに、**新しい**属性値がチェックされます。(構文はサプライヤーサーバーでチェックされているため、これには**レプリケーション**の変更は含まれません。) **syntax-validation.pl** スクリプトを実行して、**既存**の属性値で構文違反の有無を確認することもできます。

構文の検証に関する詳細は、[管理ガイド](#)を参照してください。

## 5.2. エントリー属性の参照

この参照にリストされている属性は、手動で割り当てたり、ディレクトリーエントリーで利用したりできます。属性は、定義、構文、および OID を使用してアルファベット順にリスト表示されます。

### 5.2.1. abstract

**abstract** 属性には、ドキュメントエントリーの抽象が含まれます。

OID	0.9.2342.19200300.102.1.9
-----	---------------------------

構文	DirectoryString
多値または単一値	複数値
定義される場所	Internet White Pages Pilot

### 5.2.2. accessTo

この属性は、ユーザーがアクセスできる特定のホストまたはサーバーを定義します。

OID	5.3.6.1.1.1.1.1
構文	IA5String
多値または単一値	複数値
定義される場所	nss_ldap/pam_ldap

### 5.2.3. accountInactivityLimit

**accountInactivityLimit** 属性は、アカウントの最後のログイン時刻から、そのアカウントが非アクティブであるためにロックされるまでの期間を秒単位で設定します。

OID	1.3.6.1.4.1.11.1.3.2.1.3
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

### 5.2.4. acctPolicySubentry

**acctPolicySubentry** 属性は、アカウントポリシー (具体的には、アカウントロックアウトポリシー) に属するすべてのエントリーを識別します。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を参照します。

これは、個別のユーザーエントリーまたは CoS テンプレートエントリーまたはロールエントリーに設定できます。

OID	1.3.6.1.4.1.11.1.3.2.1.2
構文	DN
多値または単一値	単一値

定義される場所	Directory Server
---------	------------------

### 5.2.5. administratorContactInfo

この属性には、LDAP またはサーバー管理者の連絡先情報が含まれます。

OID	2.16.840.1.113730.3.1.74
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.6. adminRole

この属性には、エントリーで特定されたユーザーに割り当てられたロールが含まれます。

OID	2.16.840.1.113730.3.1.601
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape 管理サービス

### 5.2.7. adminUrl

この属性には、管理サーバーの URL が含まれます。

OID	2.16.840.1.113730.3.1.75
構文	IA5String
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.8. aliasedObjectName

**aliasedObjectName** 属性は、エイリアスエントリーを識別するために Directory Server によって使用されます。この属性には、このエントリーがエイリアスであるエントリーの DN (識別名) が含まれます。以下に例を示します。

```
aliasedObjectName: uid=jdoe,ou=people,dc=example,dc=com
```

OID	2.5.4.1
構文	DN
多値または単一値	単一値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.9. associatedDomain

**associatedDomain** 属性には、ディレクトリーツリーのエントリーに関連付けられた DNS ドメインが含まれます。たとえば、識別名 **c=US,o=Example Corporation** のエントリーは、**EC.US** の関連するドメインを持ちます。これらのドメインは RFC 822 の順序で表す必要があります。

associatedDomain:US

OID	0.9.2342.19200300.100.1.37
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.10. associatedName

**associatedName** は、DNS ドメインに関連付けられた組織ディレクトリーツリーエントリーを識別します。以下に例を示します。

associatedName: c=us

OID	0.9.2342.19200300.100.1.38
構文	DN
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.11. attributeTypes

この属性は、subschema 内で定義される属性を特定するためにスキーマファイルで使用されます。

OID	2.5.21.5
-----	----------

構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2252</a>

### 5.2.12. audio

**audio** 属性には、バイナリー形式を使用したサウンドファイルが含まれます。この属性は、**u-law** でエンコードされたサウンドデータを使用します。以下に例を示します。

```
audio:: AAAAAA==
```

OID	0.9.2342.19200300.100.155
構文	Binary
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.13. authorCn

**authorCn** 属性には、ドキュメントの作成者の共通名が含まれます。以下に例を示します。

```
authorCn: John Smith
```

OID	0.9.2342.19200300.102.1.11
構文	DirectoryString
多値または単一値	複数値
定義される場所	Internet White Pages Pilot

### 5.2.14. authorityRevocationList

**authorityRevocationList** 属性には、失効した CA 証明書のリストが含まれます。この属性は要求され、**authorityRevocationList;binary** などのバイナリー形式で保存する必要があります。以下に例を示します。

```
authorityrevocationlist;binary:: AAAAAA==
```

OID	2.5.4.38
-----	----------



構文	Binary
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.15. authorSn

**authorSn** 属性には、ドキュメントエントリーの作成者の名前またはファミリー名が含まれます。以下に例を示します。

authorSn: Smith

OID	0.9.2342.19200300.102.1.12
構文	DirectoryString
多値または単一値	複数値
定義される場所	Internet White Pages Pilot

### 5.2.16. automountInformation

この属性には、autofs 自動マウント機能が使用する情報が含まれます。



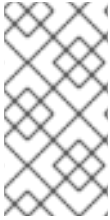
#### 注記

**automountInformation** 属性は、Directory Server の **60autofs.ldif** で定義されています。更新された RFC 2307 スキーマを使用するには、**60autofs.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.33
構文	DirectoryString
多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.17. bootFile

この属性には、ブートイメージのファイル名が含まれます。



### 注記

**bootFile** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.24
構文	IA5String
多値または単一値	複数値
定義される場所	<a href="#">RFC 2307</a>

## 5.2.18. bootParameter

この属性には、**rpc.bootparamd** の値が含まれます。



### 注記

**bootParameter** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.23
構文	IA5String
多値または単一値	複数値
定義される場所	<a href="#">RFC 2307</a>

## 5.2.19. buildingName

**buildingName** 属性には、エントリーに関連付けられたビルド名が含まれます。以下に例を示します。

```
buildingName: 14
```

OID	0.9.2342.19200300.100.1.48
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.20. businessCategory

**businessCategory** 属性は、エントリーが関与するビジネスのタイプを特定します。属性の値は、企業部門レベルなどの幅広い一般化である必要があります。以下に例を示します。

businessCategory: Engineering

OID	2.5.4.15
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.21. c (countryName)

**countryName** または **c** の属性には、国名を表す 2 文字の国コードが含まれます。国コードは ISO によって定義されています。以下に例を示します。

countryName: GB  
c: US

OID	2.5.4.6
構文	DirectoryString
多値または単一値	単一値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.22. cACertificate

**cACertificate** 属性には CA 証明書が含まれます。属性は要求され、**cACertificate;binary** などのバイナリ形式を保存する必要があります。以下に例を示します。

cACertificate;binary:: AAAAAA==

OID	2.5.4.37
構文	Binary
多値または単一値	複数値

定義される場所	<a href="#">RFC 2256</a>
---------	--------------------------

### 5.2.23. carLicense

**carLicense** 属性には、エントリーの automobile ライセンス plate 番号が含まれます。以下に例を示します。

```
carLicense: 6ABC246
```

OID	2.16.840.1.113730.3.1.1
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2798</a>

### 5.2.24. certificateRevocationList

**certificateRevocationList** 属性には、失効したユーザー証明書の一覧が含まれます。属性値は、**certificateACertificate;binary** として要求され、バイナリー形式で保存されます。以下に例を示します。

```
certificateRevocationList;binary:: AAAAAA==
```

OID	2.5.4.39
構文	Binary
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.25. cn (commonName)

**commonName** 属性にはエントリーの名前が含まれます。ユーザーエントリーの場合、**cn** 属性は通常ユーザーのフルネームです。以下に例を示します。

```
commonName: John Smith
cn: Bill Anderson
```

**LDAPReplica** または **LDAPServerobject** オブジェクトクラスを使用すると、**cn** 属性の値の形式は以下ようになります。

```
cn: replicater.example.com:17430/dc%3Dexample%2Cdc%3com
```

OID	2.5.4.3
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.26. co (friendlyCountryName)

**friendlyCountryName** 属性には国名が含まれます。これには任意の文字列を使用できます。多くの場合、**country** は ISO デザインされた 2 文字の国コードと共に使用され、**co** 属性には読み取り可能な国名が含まれます。以下に例を示します。

```
friendlyCountryName: Ireland
co: Ireland
```

OID	0.9.2342.19200300.100.1.43
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.27. cosAttribute

**cosAttribute** には、CoS の値を生成する属性の名前が含まれます。複数の **cosAttribute** 値を指定できます。この属性は、すべてのタイプの CoS 定義エントリーによって使用されます。

OID	2.16.840.1.113730.3.1.550
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 5.2.28. cosIndirectSpecifier

**cosIndirectSpecifier** は、テンプレートエントリーを識別するために間接 CoS によって使用される属性値を指定します。

OID	2.16.840.1.113730.3.1.577
-----	---------------------------

構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

### 5.2.29. cosPriority

**cosPriority** 属性は、CoS テンプレートが属性値を提供するために競合するときに属性値を提供するテンプレートを指定します。この属性は、テンプレートのグローバルの優先度を表します。0 の優先度が最も優先されます。

OID	2.16.840.1.113730.3.1.569
構文	整数
多値または単一値	単一値
定義される場所	Directory Server

### 5.2.30. cosSpecifier

**cosSpecifier** 属性には、従来の CoS で使用される属性値が含まれており、テンプレートエントリーの DN とテンプレートエントリーを特定します。

OID	2.16.840.1.113730.3.1.551
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

### 5.2.31. cosTargetTree

**cosTargetTree** 属性は、CoS スキーマが適用されるサブツリーを定義します。スキーマと複数の CoS スキーマのこの属性の値は、任意にターゲットツリーと重複する可能性があります。

OID	2.16.840.1.113730.3.1.552
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

### 5.2.32. cosTemplateDn

**cosTemplateDn** 属性には、共有属性値のリストが含まれるテンプレートエントリーの DN 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。

OID	2.16.840.1.113730.3.1.553
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

### 5.2.33. crossCertificatePair

**crossCertificatePair** 属性の値は要求され、**certificateCertificateRepair;binary** などのバイナリー形式で保存する必要があります。以下に例を示します。

```
crossCertificatePair;binary:: AAAAAA==
```

OID	2.5.4.40
構文	Binary
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.34. dc (domainComponent)

**dc** 属性には、ドメイン名の1つのコンポーネントが含まれます。以下に例を示します。

```
dc: example
domainComponent: example
```

OID	0.9.2342.19200300.100.1.25
構文	DirectoryString
多値または単一値	単一値
定義される場所	<a href="#">RFC 2247</a>

### 5.2.35. deltaRevocationList

**deltaRevocationList** 属性には、証明書失効リスト (CRL) が含まれます。属性の値を要求し、**deltaRevocationList;binary** などのバイナリー形式で保存されます。

OID	2.5.4.53
構文	Binary
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.36. departmentNumber

**departmentNumber** 属性には、エントリーの部門番号が含まれます。以下に例を示します。

```
departmentNumber: 2604
```

OID	2.16.840.1.113730.3.1.2
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2798</a>

### 5.2.37. description

**description** 属性は、人間が判読可能なエントリーの説明を提供します。**person** または **organization** のオブジェクトクラスの場合は、エントリーのロールや作業割り当てに使用することができます。以下に例を示します。

```
description: Quality control inspector for the ME2873 product line.
```

OID	2.5.4.13
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.38. destinationIndicator

**destinationIndicator** 属性には、エントリーに関連付けられた city および country が含まれます。この属性は、パブリックの telegram サービスを提供するために必要なばかりで、通常 **registeredAddress** 属性とともに使用されます。以下に例を示します。



destinationIndicator: Stow, Ohio, USA

OID	2.5.4.27
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.39. displayName

**displayName** 属性には、そのユーザーのエントリーを表示するときに使用するユーザーの優先名が含まれます。これは、1行のサマリーリストにエントリーの推奨名を表示する場合に特に便利です。**cn**などの他の属性タイプは多値であるため、優先される名前を表示するために使用できません。以下に例を示します。

displayName: John Smith

OID	2.16.840.1.113730.3.1.241
構文	DirectoryString
多値または単一値	単一値
定義される場所	<a href="#">RFC 2798</a>

### 5.2.40. dITRedirect

**dITRedirect** 属性は、1つのエントリーによって記述されたオブジェクトにディレクトリーツリー内の新しいエントリーがあることを示します。この属性は、個別の作業の場所が変更され、個人が新しい組織DNを取得する場合に使用できます。

dITRedirect: cn=jsmith,dc=example,dc=com

OID	0.9.2342.19200300.100.1.54
構文	DN
定義される場所	<a href="#">RFC 1274</a>

### 5.2.41. dmdName

**dmdName** 属性値は、Directory Server を操作する管理機関であるディレクトリー管理ドメイン (DMD) を指定します。

OID	2.5.4.54
構文	DirectoryString
多値または単一値	単一値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.42. dn (distinguishedName)

**dn** 属性には、エントリーの識別名が含まれます。以下に例を示します。

```
dn: uid=Barbara Jensen,ou=Quality Control,dc=example,dc=com
```

OID	2.5.4.49
構文	DN
定義される場所	<a href="#">RFC 2256</a>

### 5.2.43. dNSRecord

**dNSRecord** 属性には、A(Address)、タイプ A (Mail Exchange)、タイプ NS(Name Server)、タイプ SOA (Start of Authority) リソースレコードなどの DNS リソースレコードが含まれます。以下に例を示します。

```
dNSRecord: IN NS ns.uu.net
```

OID	0.9.2342.19200300.100.1.26
構文	IA5String
多値または単一値	複数値
定義される場所	インターネットディレクトリーパイロット

### 5.2.44. documentAuthor

**documentAuthor** 属性には、ドキュメントエントリーの作成者の DN が含まれます。以下に例を示します。

```
documentAuthor: uid=Barbara Jensen,ou=People,dc=example,dc=com
```

OID	0.9.2342.19200300.100.1.14
-----	----------------------------

構文	DN
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.45. documentIdentifier

**documentIdentifier** 属性には、ドキュメントの一意の識別子が含まれます。以下に例を示します。

documentIdentifier: L3204REV1

OID	0.9.2342.19200300.100.111
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.46. documentLocation

**documentLocation** 属性には、ドキュメントの元のバージョンの場所が含まれます。以下に例を示します。

documentLocation: Department Library

OID	0.9.2342.19200300.100.115
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.47. documentPublisher

**documentPublisher** 属性には、ドキュメントを公開した個人または組織が含まれます。以下に例を示します。

documentPublisher: Southeastern Publishing

OID	0.9.2342.19200300.100.156
-----	---------------------------

構文	DirectoryString
多値または単一値	単一値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.48. documentStore

**documentStore** 属性には、ドキュメントが保存される場所に関する情報が含まれます。

OID	0.9.2342.19200300.102.1.10
構文	DirectoryString
多値または単一値	複数値
定義される場所	Internet White Pages Pilot

### 5.2.49. documentTitle

**documentTitle** 属性には、ドキュメントのタイトルが含まれます。以下に例を示します。

`documentTitle: Red Hat Directory Server Administrator Guide`

OID	0.9.2342.19200300.100.1.12
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.50. documentVersion

**documentVersion** 属性には、ドキュメントの現在のバージョン番号が含まれます。以下に例を示します。

`documentVersion: 1.1`

OID	0.9.2342.19200300.100.1.13
構文	DirectoryString
多値または単一値	複数値

定義される場所	<a href="#">RFC 1274</a>
---------	--------------------------

### 5.2.51. drink (favouriteDrink)

**favouriteDrink** 属性には、人物のお気に入りの方が含まれます。これは、**drink** に短縮できます。以下に例を示します。

```
favouriteDrink: iced tea
drink: cranberry juice
```

OID	0.9.2342.19200300.100.1.5
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.52. dSAQuality

**dSAQuality** 属性には、ディレクトリーシステムエージェント (DSA) の品質の評価が含まれます。この属性により、DSA マネージャーは DSA の予想される可用性レベルを示します。以下に例を示します。

```
dSAQuality: high
```

OID	0.9.2342.19200300.100.1.49
構文	Directory-String
多値または単一値	単一値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.53. employeeNumber

**employeeNumber** 属性には、対象の個人の従業員番号が含まれます。以下に例を示します。

```
employeeNumber: 3441
```

OID	2.16.840.1.113730.3.1.3
構文	Directory-String
多値または単一値	単一値

定義される場所	<a href="#">RFC 2798</a>
---------	--------------------------

### 5.2.54. employeeType

**employeeType** 属性には、個人の勤務タイプが含まれます。以下に例を示します。

employeeType: Full time

OID	2.16.840.1.113730.3.1.4
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2798</a>

### 5.2.55. enhancedSearchGuide

**enhancedSearchGuide** 属性には、検索フィルターを構築する X.500 クライアントによって使用される情報が含まれます。以下に例を示します。

enhancedSearchGuide: (uid=bjensen)

OID	2.5.4.47
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2798</a>

### 5.2.56. fax (facsimileTelephoneNumber)

**facsimileTelephoneNumber** 属性にはエントリーの facsimile 番号が含まれます。この属性は **fax** として省略できます。以下に例を示します。

facsimileTelephoneNumber: +1 415 555 1212  
fax: +1 415 555 1212

OID	2.5.4.23
構文	TelephoneNumber
多値または単一値	複数値

定義される場所	<a href="#">RFC 2256</a>
---------	--------------------------

### 5.2.57. gecoc

**gecoc** 属性は、ユーザーの GECOS フィールドを判別するために使用されます。これは **cn** 属性と類似していますが、**gecoc** 属性を使用すると、共通名とは別の GECOS フィールドに追加情報を埋め込むことができます。また、このフィールドは、ディレクトリーに保存されている共通名がユーザーのフルネームではない場合にも便利です。

gecoc: John Smith



#### 注記

**gecoc** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.2
構文	DirectoryString
多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.58. generationQualifier

**generationQualifier** 属性には、人の名前の生成修飾子が含まれます。これは通常、名前に接尾辞として追加されます。以下に例を示します。

generationQualifier:III

OID	2.5.4.44
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.59. gidNumber

**gidNumber** 属性には、グループエントリーの一意の数値識別子が含まれるか、ユーザーエントリーのグループを特定します。これは Unix のグループ番号に似ています。

gidNumber: 100



## 注記

**gidNumber** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.1
構文	整数
多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

## 5.2.60. givenName

**givenName** 属性には、エントリーの指定された名前 (通常は名) が含まれます。以下に例を示します。

givenName: Rachel

OID	2.5.4.42
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

## 5.2.61. homeDirectory

**homeDirectory** 属性には、ユーザーのホームディレクトリーへのパスが含まれます。

homeDirectory: /home/jsmith



## 注記

**homeDirectory** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.3
構文	IA5String



多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.62. homePhone

**homePhone** 属性には、エントリーの常駐電話番号が含まれます。以下に例を示します。

```
homePhone: 415-555-1234
```



#### 注記

RFC 1274 は **homeTelephoneNumber** と **homePhone** の両方を residential phone number 属性の名前として定義しますが、Directory Server は **homePhone** 名のみを実装します。

OID	0.9.2342.19200300.100.120
構文	TelephoneNumber
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.63. homePostalAddress

**homePostalAddress** 属性には、エントリーのホームメーリングリストが含まれます。この属性は通常複数行にまたがるため、各行破損はドル記号 (\$) で表す必要があります。属性値の実際のドル記号 (\$) またはバックslash (\) を表すには、エスケープされた 16 進値 `\24` と `\5c` を使用します。以下に例を示します。

```
homePostalAddress: 1234 Ridgeway Drive$Santa Clara, CA$99555
```

以下の文字列を表すには、以下を実行します。

```
The dollar ($) value can be found
in the c:\cost file.
```

エントリーの値は以下のようになります。

```
The dollar (\24) value can be found$in the c:\c5cost file.
```

OID	0.9.2342.19200300.100.139
構文	DirectoryString

多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.64. host

**host** にはコンピューターのホスト名が含まれます。以下に例を示します。

```
host: labcontroller01
```

OID	0.9.2342.19200300.100.1.9
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.65. houseIdentifier

**houseIdentifier** には、ロケーションに特定のビルディング用の識別子が含まれています。以下に例を示します。

```
houseIdentifier: B105
```

OID	2.5.4.51
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.66. inetDomainBaseDN

この属性は、DNS ドメインのユーザーサブツリーのベース DN を識別します。

OID	2.16.840.1.113730.3.1.690
構文	DN
多値または単一値	単一値
定義される場所	サブスクライバーの相互運用性

### 5.2.67. inetDomainStatus

この属性は、ドメインの現在の状態を表示します。ドメインのステータスは **active**、**inactive**、または **deleted** になります。

OID	2.16.840.1.113730.3.1.691
構文	DirectoryString
多値または単一値	単一値
定義される場所	サブスクライバーの相互運用性

### 5.2.68. inetSubscriberAccountId

この属性には、サブスクライバーのユーザーエントリーを請求システムにリンクするために使用される一意の属性が含まれます。

OID	2.16.840.1.113730.3.1.694
構文	DirectoryString
多値または単一値	複数値
定義される場所	サブスクライバーの相互運用性

### 5.2.69. inetSubscriberChallenge

**inetSubscriberChallenge** 属性には、**subscriberIdentity** 属性でユーザーの ID を確認するために使用される、ある種の質問またはプロンプト、チャレンジフレーズが含まれています。この属性は、チャレンジへの応答が含まれる **inetSubscriberResponse** 属性とともに使用されます。

OID	2.16.840.1.113730.3.1.695
構文	IA5String
多値または単一値	単一値
定義される場所	サブスクライバーの相互運用性

### 5.2.70. inetSubscriberResponse

**inetSubscriberResponse** 属性には、**SubscriberChallenge** 属性のチャレンジ質問への回答が含まれており、**inetSubscriberChallenge** 属性のユーザーを確認します。

OID	2.16.840.1.113730.3.1.696
-----	---------------------------

構文	IA5String
多値または単一値	複数値
定義される場所	サブスクライバーの相互運用性

### 5.2.71. inetUserHttpURL

この属性には、ユーザーに関連付けられた Web アドレスが含まれます。

OID	2.16.840.1.113730.3.1.693
構文	IA5String
多値または単一値	複数値
定義される場所	サブスクライバーの相互運用性

### 5.2.72. inetUserStatus

この属性は、ユーザーの現在の状態 (subscriber) を表示します。ユーザーは、**active**、**inactive**、または **deleted** のステータスがあります。

OID	2.16.840.1.113730.3.1.692
構文	DirectoryString
多値または単一値	単一値
定義される場所	サブスクライバーの相互運用性

### 5.2.73. info

**info** 属性には、オブジェクトに関する一般情報が含まれます。特定の情報にこの属性を使用せず、代わりに特定のカスタム属性タイプに依存します。以下に例を示します。

info: not valid

OID	0.9.2342.19200300.100.1.4
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.74. initials

**initials** にはユーザーの最初のもが含まれます。これにはエントリーのラスト名は含まれません。以下に例を示します。

```
initials: BAJ
```

Directory Server と Active Directory は **initials** 属性を異なる方法で処理します。Directory Server では、実際には文字数が無制限になりますが、Active Directory では文字数が 6 文字に制限されます。エントリーが Windows ピアと同期され、**initial** 属性の値が 6 文字より長い場合、同期時に値は自動的に 6 文字に切り捨てられます。エラーログに書き込まれ、同期によって属性値が変更されたことを示す情報はありせん。

OID	2.5.4.43
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.75. installationTimeStamp

これには、サーバーインスタンスがインストールされた時間が含まれます。

OID	2.16.840.1.113730.3.1.73
構文	DirectoryString
多値または単一値	多値
定義される場所	Netscape 管理サービス

### 5.2.76. internationalISDNNumber

**internationalISDNNumber** 属性には、ドキュメントエントリーの ISDN 番号が含まれます。この属性は、CCITT Rec で指定される ISDN アドレスに国際化された形式を使用します。E.164.

OID	2.5.4.25
構文	IA5String
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.77. ipHostNumber

これには、サーバーの IP アドレスが含まれます。



#### 注記

**ipHostNumber** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.19
構文	DirectoryString
多値または単一値	多値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.78. ipNetmaskNumber

これには、サーバーの IP ネットマスクが含まれます。



#### 注記

**ipHostNumber** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	2.16.840.1.113730.3.1.73
構文	DirectoryString
多値または単一値	多値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.79. ipNetworkNumber

これにより、IP ネットワークが識別されます。



#### 注記

**ipNetworkNumber** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.20
構文	DirectoryString
多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.80. ipProtocolNumber

この属性は、IP プロトコルのバージョン番号を識別します。



#### 注記

**ipProtocolNumber** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.17
構文	整数
多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.81. ipServicePort

この属性は、IP サービスによって使用されるポートを提供します。



#### 注記

**ipServicePort** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.15
構文	整数
多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

## 5.2.82. ipServiceProtocol

これは、IP サービスによって使用されるプロトコルを特定します。



### 注記

**ipServiceProtocol** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.16
構文	DirectoryString
多値または単一値	多値
定義される場所	<a href="#">RFC 2307</a>

## 5.2.83. janetMailbox

**janetMailbox** には JANET メールアドレスが含まれます。通常、RFC 822 メールアドレスを使用しない米国の Kingdom にあるユーザー向けです。この属性が含まれるエントリーには、**rfc822Mailbox** 属性が含まれる必要もあります。

OID	0.9.2342.19200300.100.1.46
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

## 5.2.84. jpegPhoto

**jpegPhoto** 属性には、バイナリー値である JPEG 写真が含まれます。以下に例を示します。

```
jpegPhoto:: AAAAAA==
```

OID	0.9.2342.19200300.100.1.60
構文	Binary
多値または単一値	複数値
定義される場所	<a href="#">RFC 2798</a>



### 5.2.85. keyWords

**keyWord** 属性には、エントリーに関連付けられたキーワードが含まれます。以下に例を示します。

keyWords: directory LDAP X.500

OID	0.9.2342.19200300.102.1.7
構文	DirectoryString
多値または単一値	複数値
定義される場所	Internet White Pages Pilot

### 5.2.86. knowledgeInformation

この属性は使用されなくなりました。

OID	2.5.4.2
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.87. l (localityName)

**localityName** または **l** の属性には、エントリーに関連する county、city、またはその他の地理的な指定が含まれます。以下に例を示します。

localityName: Santa Clara  
l: Santa Clara

OID	2.5.4.7
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.88. labeledURI

**labeledURI** には、エントリーに関連する URI (Uniform Resource Identifier) が含まれます。属性に配置される値は、URI (現在、URL のみがサポートされています) で設定され、オプションで1つ以上のスペース文字とラベルが続く必要があります。

labeledURI: http://home.example.com  
 labeledURI: http://home.example.com Example website

OID	1.3.6.1.4.1.250.157
構文	IA5String
多値または単一値	複数値
定義される場所	<a href="#">RFC 2709</a>

## 5.2.89. loginShell

**loginShell** 属性には、ユーザーがドメインにログインする際に自動的に起動するスクリプトへのパスが含まれます。

loginShell: c:\scripts\jsmith.bat



### 注記

**loginShell** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.4
構文	IA5String
多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

## 5.2.90. macAddress

この属性は、サーバーまたは機器の MAC アドレスを提供します。



### 注記

**macAddress** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.22
-----	------------------

構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.91. mail

**mail** 属性には、ユーザーのプライマリーメールアドレスが含まれます。この属性値は、ホワイトページアプリケーションによって取得され、表示されます。以下に例を示します。

```
mail: jsmith@example.com
```

OID	0.9.2342.19200300.100.1.3
構文	DirectoryString
多値または単一値	単一値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.92. mailAccessDomain

この属性は、ユーザーがメッセージングサーバーにアクセスするために使用できるドメインをリスト表示します。

OID	2.16.840.1.113730.3.1.12
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.93. mailAlternateAddress

**mailAlternateAddress** 属性には、ユーザーの追加のメールアドレスが含まれます。この属性は、デフォルトまたはプライマリーメールアドレスを反映しません。メールアドレスは **mail** 属性で設定されます。

以下に例を示します。

```
mailAlternateAddress: jsmith@example.com
mailAlternateAddress: smith1701@alt.com
```

OID	2.16.840.1.113730.3.1.13
-----	--------------------------

構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.94. mailAutoReplyMode

この属性は、メッセージングサーバーに対して自動応答を有効にするかどうかを設定します。

OID	2.16.840.1.113730.3.1.14
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.95. mailAutoReplyText

この属性は、自動リプライメールで使用されるテキストを保存します。

OID	2.16.840.1.113730.3.1.15
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.96. mailDeliveryOption

この属性は、メールユーザーに使用するメール配信メカニズムを定義します。

OID	2.16.840.1.113730.3.1.16
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.97. mailEnhancedUniqueMember

この属性には、メールグループの一意のメンバーの DN が含まれます。

OID	2.16.840.1.113730.3.1.31
構文	DN
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.98. mailForwardingAddress

この属性には、ユーザーのメールを転送するメールアドレスが含まれます。

OID	2.16.840.1.113730.3.1.17
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.99. mailHost

**mailHost** 属性には、メールサーバーのホスト名が含まれます。以下に例を示します。

```
mailHost: mail.example.com
```

OID	2.16.840.1.113730.3.1.18
構文	DirectyString
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.100. mailMessageStore

これは、ユーザーのメールボックスの場所を特定します。

OID	2.16.840.1.113730.3.1.19
構文	IA5String
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.101. mailPreferenceOption

**mailPreferenceOption** は、電子と物理的なメーリングリストにユーザーを含めるかどうかを定義します。3つのオプションがあります。

0	メーリングリストには表示されません。
1	メーリングリストに追加します。
2	ユーザーの興味のあるプロバイダービューをメーリングリストにのみ追加。

属性が存在しない場合、デフォルトでは、ユーザーはどのメーリングリストにも含まれていないと見なされます。この属性は、ディレクトリーを使用してメーリングリストを取得し、その値を引き継ぐことで解釈する必要があります。以下に例を示します。

```
mailPreferenceOption: 0
```

OID	0.9.2342.19200300.100.1.47
構文	整数
多値または単一値	単一値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.102. mailProgramDeliveryInfo

この属性には、プログラムメール配信に使用するコマンドが含まれます。

OID	2.16.840.1.113730.3.1.20
構文	IA5String
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.103. mailQuota

この属性は、ユーザーのメールボックスに許可されるディスク領域を設定します。

OID	2.16.840.1.113730.3.1.21
構文	DirectoryString

多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.104. mailRoutingAddress

この属性には、ユーザーが受信した電子メールを別のメッセージングサーバーに転送するときに使用するルーティングアドレスが含まれます。

OID	2.16.840.1.113730.3.1.24
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.105. manager

**manager** には、その人のマネージャーの識別名 (DN) が含まれています。以下に例を示します。

```
manager: cn=Bill Andersen,ou=Quality Control,dc=example,dc=com
```

OID	0.9.2342.19200300.100.1.10
構文	DN
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.106. member

**member** 属性には、グループの各メンバーの識別名 (DN) が含まれます。以下に例を示します。

```
member: cn=John Smith,dc=example,dc=com
```

OID	2.5.4.31
構文	DN
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

## 5.2.107. memberCertificateDescription

この属性は多値属性で、各値は、証明書のサブジェクト DN に一致する説明、パターン、または TLS クライアント認証に使用される証明書に一致するフィルターになります。

**memberCertificateDescription** は、説明と同じ属性値アサーション (AVA) を持つサブジェクト DN を含むすべての証明書と一致します。説明は、複数の **ou** の AVA を含めることができます。一致する DN には、同じ **ou** AVA が同じ順序で含まれている必要がありますが、他の **ouAVA** を含む他の AVA が散在している場合があります。他の属性タイプ (**ou** ではない) の場合、説明にはそのタイプの AVA が最大で 1 つ含まれている必要があります。複数の場合、最後のものはすべて無視されます。

一致する DN には、同じ AVA が含まれている必要がありますが、ルートの近くにある同じタイプの他の AVA は含まれていません (後で構文的に)。

AVA に同じ属性の説明 (大文字と小文字を区別しない比較) と同じ属性値 (大文字と小文字を区別しない比較、先頭と末尾の空白は無視され、連続する空白文字は単一のスペースとして扱われる) が含まれている場合、AVA は同じと見なされます。

次の **memberCertificateDescription** 値を持つグループのメンバーと見なされるには、証明書に **ou=x**、**ou=A**、および **dc=example** が含まれている必要がありますが、**dc=company** は含まれていません。

```
memberCertificateDescription: {ou=x,ou=A,dc=company,dc=example}
```

グループの要件と一致させるには、証明書のサブジェクト DN には **memberCertificateDescription** 属性で定義された順序で同じ **ou** 属性タイプが含まれている必要があります。

OID	2.16.840.1.113730.3.1.199
構文	IA5String
多値または単一値	複数値
定義される場所	Directory Server

## 5.2.108. memberNisNetgroup

この属性は、マージ netgroup の名前をリスト表示することで、別の netgroup の属性値を現在の値にマージします。



### 注記

**memberNisNetgroup** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.13
構文	IA5String



多値または単一値	複数値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.109. memberOf

この属性には、ユーザーがメンバーであるグループ名が含まれます。

**memberOf** は、グループメンバーのユーザーエントリーの MemberOf プラグインによって生成されたデフォルトの属性です。この属性は、グループエントリーに記載されている **member** 属性に自動的に同期されるため、エントリーのグループメンバーシップを表示することは Directory Server によって管理されます。



#### 注記

この属性は、MemberOf プラグインが有効で、この属性を使用するように設定されている場合、グループエントリーと対応するメンバーのユーザーエントリー間で同期されません。

OID	1.2.840.113556.1.2.102
構文	DN
多値または単一値	複数値
定義される場所	Netscape 委譲管理者

### 5.2.110. memberUid

**memberUid** 属性には、グループのメンバーのログイン名が含まれます。これは、**member** 属性で特定された DN とは異なる場合があります。

memberUID: jsmith



#### 注記

**memberUID** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.12
構文	IA5String
多値または単一値	単一値

定義される場所	<a href="#">RFC 2307</a>
---------	--------------------------

### 5.2.111. memberURL

この属性は、グループの各メンバーに関連付けられた URL を識別します。ラベル付きの URL の任意のタイプを使用できます。

```
memberURL: ldap://cn=jsmith,ou=people,dc=example,dc=com
```

OID	2.16.840.1.113730.3.1.198
構文	IA5String
多値または単一値	複数値
定義される場所	Directory Server

### 5.2.112. mepManagedBy

この属性には、送信元エントリーの DN を参照する自動生成されるエントリーのポインターが含まれます。この属性はマネージドエントリープラグインによって設定され、手動で変更することはできません。

OID	2.16.840.1.113730.3.1.2086
構文	DN
多値または単一値	単一値
定義される場所	Directory Server

### 5.2.113. mepManagedEntry

この属性には、現在のエントリーに対応する自動生成されたエントリーへのポインターが含まれます。この属性はマネージドエントリープラグインによって設定され、手動で変更することはできません。

OID	2.16.840.1.113730.3.1.2087
構文	DN
多値または単一値	単一値
定義される場所	Directory Server

### 5.2.114. mepMappedAttr

この属性は、生成されたエントリーに存在する必要がある Managed Entries テンプレートエントリーに属性を設定します。マッピングは、元のエントリーの値の一部が指定の属性を指定するために使用されます。これらの属性の値は、`attribute: $attr` のトークンになります。以下に例を示します。

```
mepMappedAttr: gidNumber: $gidNumber
```

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
```

OID	2.16.840.1.113730.3.1.2089
構文	OctetString
多値または単一値	複数値
定義される場所	Directory Server

### 5.2.115. mepRDNAttr

この属性は、マネージドエントリープラグインによって作成された自動生成エントリーの命名属性として使用する属性を設定します。命名属性で指定されている属性型はすべて、マネージドエントリーのテンプレートエントリーに **mepMappedAttr** として存在する必要があります。

OID	2.16.840.1.113730.3.1.2090
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

### 5.2.116. mepStaticAttr

この属性は、Managed Entries プラグインによって管理される自動生成されたエントリーに追加する必要がある定義された値を持つ属性を設定します。この値は、Managed Entries プラグインのそのインスタンスが生成するすべてのエントリーに使用されます。

```
mepStaticAttr: posixGroup
```

OID	2.16.840.1.113730.3.1.2088
構文	OctetString

多値または単一値	複数値
定義される場所	Directory Server

### 5.2.117. mgrpAddHeader

この属性には、メッセージのヘッダーに関する情報が含まれます。

OID	2.16.840.1.113730.3.1.781
構文	IA5String
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.118. mgrpAllowedBroadcaster

この属性は、ユーザーがブロードキャストメッセージを送信できるようにするかどうかを設定します。

OID	2.16.840.1.113730.3.1.22
構文	IA5String
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.119. mgrpAllowedDomain

この属性は、メールグループのドメインを設定します。

OID	2.16.840.1.113730.3.1.23
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.120. mgrpApprovePassword

この属性は、メールへのアクセスに使用されるパスワードを承認する必要があるかどうかを設定します。

OID	mgrpApprovePassword-oid
構文	IA5String
多値または単一値	単一値
定義される場所	Netscape Messaging Server

### 5.2.121. mgrpBroadcasterPolicy

この属性は、メールをブロードキャストするポリシーを定義します。

OID	2.16.840.1.113730.3.1.788
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.122. mgrpDeliverTo

この属性には、電子メールの配信先に関する情報が含まれます。

OID	2.16.840.1.113730.3.1.25
構文	IA5String
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.123. mgrpErrorsTo

この属性には、メッセージングサーバーのエラーメッセージを配信する場所に関する情報が含まれません。

OID	2.16.840.1.113730.3.1.26
構文	IA5String
多値または単一値	単一値
定義される場所	Netscape Messaging Server

### 5.2.124. mgrpModerator

この属性には、メーリングリストの連絡先名が含まれます。

OID	2.16.840.1.113730.3.1.33
構文	IA5String
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.125. mgrpMsgMaxSize

この属性は、電子メールメッセージに許可される最大サイズを設定します。

OID	2.16.840.1.113730.3.1.32
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape Messaging Server

### 5.2.126. mgrpMsgRejectAction

この属性は、メッセージングサーバーが拒否されたメッセージに対して実行するアクションを定義します。

OID	2.16.840.1.113730.3.1.28
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.127. mgrpMsgRejectText

この属性は、拒否通知に使用するテキストを設定します。

OID	2.16.840.1.113730.3.1.29
構文	IA5String
多値または単一値	複数値

定義される場所	Netscape Messaging Server
---------	---------------------------

### 5.2.128. mgrpNoDuplicateChecks

この属性は、メッセージングサーバーが重複メールをチェックするかどうかを定義します。

OID	2.16.840.1.113730.3.1.789
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape Messaging Server

### 5.2.129. mgrpRemoveHeader

この属性は、応答メッセージでヘッダーが削除されるかどうかを設定します。

OID	2.16.840.1.113730.3.1.801
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.130. mgrpRFC822MailMember

この属性は、メールグループのメンバーを特定します。

OID	2.16.840.1.113730.3.1.130
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.131. mobile

**mobile**、または **mobileTelephoneNumber** には、エントリーのモバイルまたはセル形式の電話番号が含まれます。以下に例を示します。

**mobileTelephoneNumber**: 415-555-4321

OID	0.9.2342.19200300.100.1.41
構文	TelephoneNumber
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.132. mozillaCustom1

この属性は、共有アドレスの書籍を管理するために Mozilla Thunderbird によって使用されます。

OID	1.3.6.1.4.1.13769.4.1
構文	DirectoryString
多値または単一値	単一値
定義される場所	Mozilla Address Book

### 5.2.133. mozillaCustom2

この属性は、共有アドレスの書籍を管理するために Mozilla Thunderbird によって使用されます。

OID	1.3.6.1.4.1.13769.4.2
構文	DirectoryString
多値または単一値	単一値
定義される場所	Mozilla Address Book

### 5.2.134. mozillaCustom3

この属性は、共有アドレスの書籍を管理するために Mozilla Thunderbird によって使用されます。

OID	1.3.6.1.4.1.13769.4.3
構文	DirectoryString
多値または単一値	単一値
定義される場所	Mozilla Address Book



### 5.2.135. mozillaCustom4

この属性は、共有アドレスの書籍を管理するために Mozilla Thunderbird によって使用されます。

OID	1.3.6.1.4.1.13769.4.4
構文	DirectoryString
多値または単一値	単一値
定義される場所	Mozilla Address Book

### 5.2.136. mozillaHomeCountryName

この属性は、共有アドレスガイドの Mozilla gitops が使用する国を設定します。

OID	1.3.6.1.4.1.13769.3.6
構文	DirectoryString
多値または単一値	単一値
定義される場所	Mozilla Address Book

### 5.2.137. mozillaHomeLocalityName

この属性は、共有アドレスガイドの Mozilla pid で使用される都市を設定します。

OID	1.3.6.1.4.1.13769.3.3
構文	DirectoryString
多値または単一値	単一値
定義される場所	Mozilla Address Book

### 5.2.138. mozillaHomePostalCode

この属性は、共有アドレスガイドの Mozilla Warehouse で使用されるポストコードを設定します。

OID	1.3.6.1.4.1.13769.3.5
構文	DirectoryString
多値または単一値	単一値

定義される場所	Mozilla Address Book
---------	----------------------

### 5.2.139. mozillaHomeState

この属性は、共有アドレスガイドの Mozilla TEMPLATES で使用される状態またはプロイエンスを設定します。

OID	1.3.6.1.4.1.13769.3.4
構文	DirectoryString
多値または単一値	単一値
定義される場所	Mozilla Address Book

### 5.2.140. mozillaHomeStreet

この属性は、共有アドレスガイドで使用する Mozilla street アドレスを設定します。

OID	1.3.6.1.4.1.13769.3.1
構文	DirectoryString
多値または単一値	単一値
定義される場所	Mozilla Address Book

### 5.2.141. mozillaHomeStreet2

この属性には、共有電話帳の Mozilla TEMPLATES で使用されるアドレス帳の 2 行目が含まれます。

OID	1.3.6.1.4.1.13769.3.2
構文	DirectoryString
多値または単一値	単一値
定義される場所	Mozilla Address Book

### 5.2.142. mozillaHomeUrl

この属性には、共有アドレス帳ガイドの Mozilla Warehouse が使用する URL が含まれます。

OID	1.3.6.1.4.1.13769.3.7
-----	-----------------------

構文	DirectoryString
多値または単一値	単一値
定義される場所	Mozilla Address Book

### 5.2.143. mozillaNickname (xmozillanickname)

この属性には、MozillaThunderbird が使用する共有アドレス帳のニックネームが含まれています。

OID	1.3.6.1.4.1.13769.2.1
構文	DirectoryString
多値または単一値	複数値
定義される場所	Mozilla Address Book

### 5.2.144. mozillaSecondEmail (xmozillasecondemail)

この属性には、Mozilla pid の共有アドレス帳ガイドのエントリーの代替またはセカンダリーメールアドレスが含まれます。

OID	1.3.6.1.4.1.13769.2.2
構文	IA5String
多値または単一値	単一値
定義される場所	Mozilla Address Book

### 5.2.145. mozillaUseHtmlMail (xmozillausehtmlmail)

この属性は、Mozilla pid の共有アドレス帳ガイドのエントリーのメールタイプの優先度を設定します。

OID	1.3.6.1.4.1.13769.2.3
構文	Boolean
多値または単一値	単一値
定義される場所	Mozilla Address Book

### 5.2.146. mozillaWorkStreet2

この属性には、Mozilla 336 の共有アドレス帳ガイドのエントリーのワークスペースまたはオフィス用のストリートアドレスが含まれます。

OID	1.3.6.1.4.1.13769.3.8
構文	DirectoryString
多値または単一値	単一値
定義される場所	Mozilla Address Book

### 5.2.147. mozillaWorkUrl

この属性には、Mozilla pid の共有アドレス帳ガイドのエントリーにワークサイトの URL が含まれません。

OID	1.3.6.1.4.1.13769.3.9
構文	DirectoryString
多値または単一値	単一値
定義される場所	Mozilla Address Book

### 5.2.148. multiLineDescription

この属性には、LDIF ファイルの複数の行にまたがるエントリーの説明が含まれます。

OID	1.3.6.1.4.1.250.1.2
構文	DirectoryString
多値または単一値	複数値
定義される場所	Internet White Pages Pilot

### 5.2.149. name

**name** 属性は、命名用の文字列属性型を形成するために使用できる属性 supertype を識別します。

このタイプの値はエントリーで発生するわけではありません。属性サブタイプをサポートしない LDAP サーバー実装は、リクエストでこの属性を認識する必要はありません。クライアントの実装は、LDAP サーバーが属性のサブクラスを実行できることを想定すべきではありません。

OID	2.5.4.41
-----	----------

構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.150. netscapeReversiblePassword

この属性には、HTTP Digest/MD5 認証のパスワードが含まれます。

OID	2.16.840.1.113730.3.1.812
構文	OctetString
多値または単一値	複数値
定義される場所	Netscape Web Server

### 5.2.151. NisMapEntry

この属性には、ネットワーク情報サービスが使用する NIS マップの情報が含まれます。



#### 注記

この属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.27
構文	IA5String
多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.152. nisMapName

この属性には、NIS サーバーで使用されるマッピングの名前が含まれます。

OID	1.3.6.1.1.1.1.26
構文	DirectoryString

多値または単一値	複数値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.153. nisNetgroupTriple

この属性には、NIS サーバーが使用する netgroup に関する情報が含まれます。



#### 注記

この属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.14
構文	IA5String
多値または単一値	複数値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.154. nsAccessLog

このエントリーは、サーバーによって使用されるアクセスログを特定します。

OID	nsAccessLog-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.155. nsAdminAccessAddresses

この属性には、インスタンスによって使用される管理サーバーの IP アドレスが含まれます。

OID	nsAdminAccessAddresses-oid
構文	DirectoryString
多値または単一値	複数値

定義される場所	Netscape 管理サービス
---------	-----------------

### 5.2.156. nsAdminAccessHosts

この属性には、管理サーバーのホスト名が含まれます。

OID	nsAdminAccessHosts-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.157. nsAdminAccountInfo

この属性には、管理サーバーアカウントに関するその他の情報が含まれます。

OID	nsAdminAccountInfo-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.158. nsAdminCacheLifetime

これにより、Directory Server が使用するキャッシュを保存する時間が設定されます。

OID	nsAdminCacheLifetime-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.159. nsAdminCgiWaitPid

この属性は、管理サーバー CGI プロセス ID の待機時間を定義します。

OID	nsAdminCgiWaitPid-oid
-----	-----------------------

構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.160. nsAdminDomainName

この属性には、Directory Server インスタンスを含む管理ドメインの名前が含まれます。

OID	nsAdminDomainName-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.161. nsAdminEnableEnduser

この属性は、エンドユーザーが admin サービスへのアクセスを許可するかどうかを設定します。

OID	nsAdminEnableEnduser-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.162. nsAdminEndUserHTMLIndex

この属性は、エンドユーザーが admin サービスの HTML インデックスにアクセスできるようにするかどうかを設定します。

OID	nsAdminEndUserHTMLIndex-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.163. nsAdminGroupName



この属性は、管理者ガイドの名前を指定します。

OID	nsAdminGroupName-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

#### 5.2.164. nsAdminOneACLDDir

この属性は、管理サーバーのアクセス制御リストを含むディレクトリーパスを提供します。

OID	nsAdminOneACLDDir-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

#### 5.2.165. nsAdminSIEDN

この属性には、管理サーバーの SIE (saserer インスタンスエントリー) の DN が含まれます。

OID	nsAdminSIEDN-oid
構文	DN
多値または単一値	複数値
定義される場所	Netscape 管理サービス

#### 5.2.166. nsAdminUsers

この属性は、管理サーバーの管理ユーザーの情報が含まれるファイルのパスと名前を指定します。

OID	nsAdminUsers-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.167. nsAIMid

この属性には、ユーザーの AOL インスタントメッセージングユーザー ID が含まれます。

OID	2.16.840.1.113730.3.2.300
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 5.2.168. nsBaseDN

これには、Directory Server のサーバーインスタンスの定義エントリで使用されるベース DN が含まれます。

OID	nsBaseDN-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 5.2.169. nsBindDN

この属性には、Directory Server SIE で定義されたバインド DN が含まれます。

OID	nsBindDN-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 5.2.170. nsBindPassword

この属性には、**nsBindDN** で定義されたバインド DN によって使用されるパスワードが含まれます。

OID	nsBindPassword-oid
構文	DirectoryString
多値または単一値	複数値

定義される場所	Directory Server
---------	------------------

### 5.2.171. nsBuildNumber

これは、Directory Server SIE で、サーバーインスタンスのビルド番号を定義します。

OID	nsBuildNumber-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.172. nsBuildSecurity

これにより、Directory Server SIE でビルドのセキュリティーレベルが定義されます。

OID	nsBuildSecurity-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.173. nsCertConfig

この属性は、Red Hat Certificate System の設定を定義します。

OID	nsCertConfig-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Certificate System

### 5.2.174. nsClassname

OID	nsClassname-oid
構文	DirectoryString

多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.175. nsConfigRoot

この属性には、設定ディレクトリーのルート DN が含まれます。

OID	nsConfigRoot-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.176. nscpAIMScreenname

この属性は、ユーザーの AIM スクリーン名を示します。

OID	1.3.6.1.4.1.13769.2.4
構文	TelephoneString
多値または単一値	複数値
定義される場所	Mozilla Address Book

### 5.2.177. nsDefaultAcceptLanguage

この属性には、HTML クライアントで受け入れられる言語コードが含まれています。

OID	nsDefaultAcceptLanguage-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.178. nsDefaultObjectClass

この属性は、オブジェクトクラス情報をコンテナエントリーに格納します。

OID	nsDefaultObjectClass-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.179. nsDeleteclassname

OID	nsDeleteclassname-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.180. nsDirectoryFailoverList

この属性には、フェイルオーバーに使用する Directory Server のリストが含まれています。

OID	nsDirectoryFailoverList-oid
構文	IA5String
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.181. nsDirectoryInfoRef

この属性は、サーバーに関する情報を含むエントリーの DN を参照します。

OID	nsDirectoryInfoRef-oid
構文	DN
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.182. nsDirectoryURL

この属性には、Directory Server の URL が含まれます。

OID	nsDirectoryURL-oid
構文	IA5String
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.183. nsDisplayName

この属性には表示名が含まれています。

OID	nsDisplayName-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.184. nsErrorLog

この属性では、サーバーが使用するエラーログを特定します。

OID	nsErrorLog-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.185. nsExecRef

この属性には、サーバタスクの実行に使用できる実行可能ファイルのパスまたは場所が含まれます。

OID	nsExecRef-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.186. nsExpirationDate

この属性には、アプリケーションの有効期限が含まれます。

OID	nsExpirationDate-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.187. nsGroupRDNComponent

この属性は、グループエントリーの RDN に使用する属性を定義します。

OID	nsGroupRDNComponent-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.188. nsHardwarePlatform

この属性は、サーバーが実行されているハードウェアを指定します。この属性の値は、**uname -m** からの出力と同じです。以下に例を示します。

```
nsHardwarePlatform:i686
```

OID	nsHardwarePlatform-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.189. nsHelpRef

この属性には、オンラインヘルプファイルへの参照が含まれています。

OID	nsHelpRef-oid
構文	DirectoryString

多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.190. nsHostLocation

この属性には、サーバーホストに関する情報が含まれています。

OID	nsHostLocation-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.191. nsICQid

この属性には、ユーザーの ICQID が含まれています。

OID	2.16.840.1.113730.3.1.2014
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 5.2.192. nsInstalledLocation

この属性には、バージョン 7.1 以前の Directory Server のインストールディレクトリーが含まれます。

OID	nsInstalledLocation-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.193. nsJarfilename

この属性は、コンソールで使用される jar ファイル名を指定します。



OID	nsJarfilename-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.194. nsLdapSchemaVersion

これにより、LDAP ディレクトリースキーマのバージョン番号がわかります。

OID	nsLdapSchemaVersion-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.195. nsLicensedFor

**nsLicensedFor** 属性は、ユーザーが使用を許可されているサーバーを識別します。管理サーバーは、各 **nsLicenseUser** エントリーにこの属性のインスタンスが 0 個以上含まれていることを想定しています。この属性の有効なキーワードは次のとおりです。

- **slapd**: ライセンスされた Directory Server クライアントの場合
- ライセンスされたメールサーバクライアントの **mail**。
- ライセンスされたニュースサーバクライアントのための **news**。
- **cal**: ライセンスされたカレンダーサーバクライアントの場合

以下に例を示します。

```
nsLicensedFor: slapd
```

OID	2.16.840.1.113730.3.1.36
構文	DirectoryString
多値または単一値	複数値
定義される場所	管理サーバー

### 5.2.196. nsLicenseEndTime

将来の使用のために予約されています。

OID	2.16.840.1.113730.3.1.38
構文	DirectoryString
多値または単一値	複数値
定義される場所	管理サーバー

### 5.2.197. nsLicenseStartTime

将来の使用のために予約されています。

OID	2.16.840.1.113730.3.1.37
構文	DirectoryString
多値または単一値	複数値
定義される場所	管理サーバー

### 5.2.198. nsLogSuppress

この属性は、サーバーのロギングを抑制するかどうかを設定します。

OID	nsLogSuppress-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape

### 5.2.199. nsmsgDisallowAccess

この属性は、メッセージングサーバーへのアクセスを定義します。

OID	nsmsgDisallowAccess-oid
構文	IA5String
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.200. nsmsgNumMsgQuota

この属性は、メッセージングサーバーによって保持されるメッセージ数のクォータを設定します。

OID	nsmsgNumMsgQuota-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.201. nsMSNid

この属性には、ユーザーの MSN インスタントメッセージング ID が含まれます。

OID	2.16.840.1.113730.3.1.2016
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 5.2.202. nsNickName

この属性は、アプリケーションのニックネームを示します。

OID	nsNickName-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape

### 5.2.203. nsNYR

OID	nsNYR-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	管理サービス

### 5.2.204. nsOsVersion

この属性には、サーバーが実行されているホストのオペレーティングシステムのバージョン番号が含まれます。

OID	nsOsVersion-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape

### 5.2.205. nsPidLog

OID	nsPidLog-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape

### 5.2.206. nsPreference

この属性は、コンソール設定を格納します。

OID	nsPreference-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.207. nsProductName

これには、Red Hat Directory Server や管理サーバーなどの製品の名前が含まれます。

OID	nsProductName-oid
構文	DirectoryString
多値または単一値	複数値

定義される場所	Netscape
---------	----------

### 5.2.208. nsProductVersion

これには、Directory Server または管理サーバーのバージョン番号が含まれます。

OID	nsProductVersion-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape

### 5.2.209. nsRevisionNumber

この属性には、Directory Server または管理サーバーのリビジョン番号が含まれます。

OID	nsRevisionNumber-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape

### 5.2.210. nsSecureServerPort

この属性には、Directory Server の TLS ポートが含まれています。



#### 注記

この属性では、Directory Server の TLS ポートは **設定** されません。これは、Directory Server の **dse.ldif** ファイルの **nsslapd-secureport** 設定属性で設定されます。設定属性については、**設定、コマンド、およびファイルリファレンス** で説明しています。

OID	nsSecureServerPort-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 5.2.211. nsSerialNumber

この属性には、Red Hat Directory Server や管理サーバーなどの特定のサーバーアプリケーションに割り当てられたシリアル番号または追跡番号が含まれます。

OID	nsSerialNumber-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape

### 5.2.212. nsServerAddress

この属性には、Directory Server が実行されているサーバーホストの IP アドレスが含まれます。

OID	nsServerAddress-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape

### 5.2.213. nsServerCreationClassname

この属性は、サーバーの作成時に使用するクラス名を指定します。

OID	nsServerCreationClassname-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape

### 5.2.214. nsServerID

これには、サーバーのインスタンス名が含まれます。以下に例を示します。

nsServerID: slapd-example

OID	nsServerID-oid
構文	DirectoryString

多値または単一値	複数値
定義される場所	Netscape

### 5.2.215. nsServerMigrationClassname

この属性には、サーバーの移行時に使用するクラスの名前が含まれています。

OID	nsServerMigrationClassname-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape

### 5.2.216. nsServerPort

この属性には、Directory Server の標準 LDAP ポートが含まれています。



#### 注記

この属性では、Directory Server の標準ポートは **設定** されません。これは、Directory Server の **dse.ldif** ファイルの **nsslapd-port** 設定属性で設定されます。設定属性については、**設定**、**コマンド**、および**ファイルリファレンス** で説明しています。

OID	nsServerPort-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape

### 5.2.217. nsServerSecurity

これは、Directory Server が安全な TLS または SSL 接続を必要とするかどうかを示します。

OID	nsServerSecurity-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape

### 5.2.218. nsSNMPContact

この属性には、SNMP によって提供される連絡先情報が含まれます。

OID	2.16.840.1.113730.3.1.235
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 5.2.219. nsSNMPDescription

これには、SNMP サービスの説明が含まれています。

OID	2.16.840.1.113730.3.1.236
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 5.2.220. nsSNMPEnabled

この属性は、サーバーで SNMP が有効になっているかどうかを指定します。

OID	2.16.840.1.113730.3.1.232
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 5.2.221. nsSNMPLocation

この属性は、SNMP サービスによって提供される場所を指定します。

OID	2.16.840.1.113730.3.1.234
構文	DirectoryString
多値または単一値	複数値



定義される場所	Directory Server
---------	------------------

### 5.2.222. nsSNMPMasterHost

この属性は、SNMP マスターエージェントのホスト名を指定します。

OID	2.16.840.1.113730.3.1.237
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 5.2.223. nsSNMPMasterPort

この属性は、SNMP サブエージェントのポート番号を指定します。

OID	2.16.840.1.113730.3.1.238
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 5.2.224. nsSNMPOrganization

この属性には、SNMP によって提供される組織情報が含まれます。

OID	2.16.840.1.113730.3.1.233
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 5.2.225. nsSuiteSpotUser

この属性は非推奨になりました。

この属性は、サーバーをインストールした Unix ユーザーを識別します。

OID	nsSuiteSpotUser-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape

### 5.2.226. nsTaskLabel

OID	nsTaskLabel-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape

### 5.2.227. nsUniqueAttribute

これにより、サーバー設定に一意の属性が設定されます。

OID	nsUniqueAttribute-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.228. nsUserIDFormat

この属性は、**givenname** 属性と **sn** 属性から **uid** 属性を生成するために使用する形式を設定します。

OID	nsUserIDFormat-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.229. nsUserRDNComponent

この属性は、ユーザーエントリーの RDN を設定するための属性タイプを設定します。

OID	nsUserRDNComponent-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.230. nsValueBin

OID	2.16.840.1.113730.3.1.247
構文	Binary
多値または単一値	複数値
定義される場所	Netscape サーバー - 値項目

### 5.2.231. nsValueCES

OID	2.16.840.1.113730.3.1.244
構文	IA5String
多値または単一値	複数値
定義される場所	Netscape サーバー - 値項目

### 5.2.232. nsValueCIS

OID	2.16.840.1.113730.3.1.243
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape サーバー - 値項目

### 5.2.233. nsValueDefault

OID	2.16.840.1.113730.3.1.250
構文	DirectoryString

多値または単一値	複数値
定義される場所	Netscape サーバー - 値項目

### 5.2.234. nsValueDescription

OID	2.16.840.1.113730.3.1.252
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape サーバー - 値項目

### 5.2.235. nsValueDN

OID	2.16.840.1.113730.3.1.248
構文	DN
多値または単一値	複数値
定義される場所	Netscape サーバー - 値項目

### 5.2.236. nsValueFlags

OID	2.16.840.1.113730.3.1.251
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape サーバー - 値項目

### 5.2.237. nsValueHelpURL

OID	2.16.840.1.113730.3.1.254
構文	IA5String
多値または単一値	複数値
定義される場所	Netscape サーバー - 値項目

**5.2.238. nsValueInt**

OID	2.16.840.1.113730.3.1.246
構文	整数
多値または単一値	複数値
定義される場所	Netscape サーバー - 値項目

**5.2.239. nsValueSyntax**

OID	2.16.840.1.113730.3.1.253
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape サーバー - 値項目

**5.2.240. nsValueTel**

OID	2.16.840.1.113730.3.1.245
構文	TelephoneString
多値または単一値	複数値
定義される場所	Netscape サーバー - 値項目

**5.2.241. nsValueType**

OID	2.16.840.1.113730.3.1.249
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape サーバー - 値項目

**5.2.242. nsVendor**

これには、サーバーベンダーの名前が含まれます。

OID	nsVendor-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape

### 5.2.243. nsViewConfiguration

この属性は、コンソールで使用されるビュー設定を格納します。

OID	nsViewConfiguration-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.244. nsViewFilter

この属性は、ビューに属するエントリーを識別するために使用される属性と値のペアを設定します。

OID	2.16.840.1.113730.3.1.3023
構文	IA5String
多値または単一値	複数値
定義される場所	Directory Server

### 5.2.245. nsWellKnownJarfiles

OID	nsWellKnownJarfiles-oid
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.246. nswmExtendedUserPrefs

この属性は、メッセージングサーバーのアカウントのユーザー設定を保存するために使用されます。

OID	2.16.840.1.113730.3.1.520
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.247. nsYIMid

この属性には、ユーザーの Yahoo インスタントメッセージングユーザー名が含まれます。

OID	2.16.840.1.113730.3.1.2015
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 5.2.248. ntGroupAttributes

この属性は、グループに関する情報を含むバイナリーファイルを指定します。以下に例を示します。

```
ntGroupAttributes:: lyEvYmluL2tzaAoKIwojIGRIZmF1bHQgdmFsdWUKIwpIPSJgaG9zdG5hb
```

OID	2.16.840.1.113730.3.1.536
構文	Binary
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.249. ntGroupCreateNewGroup

**ntGroupCreateNewGroup** 属性は、Windows Sync によって使用され、Windows Server で新しいグループが作成されたときに Directory Server が新しいグループエントリーを作成するかどうかを決定します。**true** は新しいエントリーを作成します。**false** は Windows エントリーを無視します。

OID	2.16.840.1.113730.3.1.45
構文	DirectoryString

多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.250. ntGroupDeleteGroup

**ntGroupDeleteGroup** 属性は、Windows 同期によって使用され、Windows 同期ピアサーバーでグループが削除されたときに Directory Server がグループエントリを削除する必要があるかどうかを決定します。**true** は、アカウントを削除し、**false** は削除を無視します。

OID	2.16.840.1.113730.3.1.46
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.251. ntGroupDomainId

**ntGroupDomainId** 属性には、グループのドメイン ID 文字列が含まれます。

`ntGroupDomainId: DS HR Group`

OID	2.16.840.1.113730.3.1.44
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.252. ntGroupId

**ntGroupId** 属性は、グループを識別するバイナリーファイルを指します。以下に例を示します。

`ntGroupId: IOUnHNjjRgghghREgfvltrGHYuTYhjIOhTYtyHJuSDwOopKLhJGbnGFtr`

OID	2.16.840.1.113730.3.1.110
構文	Binary
多値または単一値	単一値
定義される場所	Netscape NT 同期



### 5.2.253. ntGroupType

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。

**ntGroupType** 属性は、Windows グループのタイプを識別します。有効な値は次のとおりです。

- グローバル/セキュリティーの場合は **-21483646**
- ドメインローカル/セキュリティーの場合は **-21483644**
- 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**

この値は、Windows グループの同期時に自動的に設定されます。グループのタイプを判別するには、グループの作成時に手動で設定する必要があります。デフォルトでは、Directory Server グループにはこの属性がなく、グローバル/セキュリティーグループとして同期されます。

**ntGroupType:** -21483646

OID	2.16.840.1.113730.3.1.47
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.254. ntUniqueld

**ntUniqueld** 属性には、生成された番号が含まれており、内部サーバーの識別と操作に使用されます。以下に例を示します。

**ntUniqueld:** 352562404224a44ab040df02e4ef500b

OID	2.16.840.1.113730.3.1.111
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.255. ntUserAcctExpires

この属性は、エントリーの Windows アカウントがいつ期限切れになるかを示します。この値は、GMT 形式の文字列として保存されます。以下に例を示します。

```
ntUserAcctExpires: 20081015203415
```

OID	2.16.840.1.113730.3.1.528
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.256. ntUserAuthFlags

この属性には、Windows アカウントに設定された認可フラグが含まれています。

OID	2.16.840.1.113730.3.1.60
構文	Binary
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.257. ntUserBadPwCount

この属性は、アカウントがロックされるまでに許容される、不正なパスワード入力の失敗回数を設定します。

OID	2.16.840.1.113730.3.1.531
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.258. ntUserCodePage

**ntUserCodePage** 属性には、選択したユーザーの言語のコードページが含まれています。以下に例を示します。

```
ntUserCodePage: AAAAAA==
```

OID	2.16.840.1.113730.3.1.533
構文	Binary
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.259. ntUserComment

この属性には、ユーザーエントリーに関するテキストの説明またはメモが含まれます。

OID	2.16.840.1.113730.3.1.522
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.260. ntUserCountryCode

この属性には、ユーザーの居住国の2文字の国コードが含まれています。

OID	2.16.840.1.113730.3.1.532
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.261. ntUserCreateNewAccount

**ntUserCreateNewAccount** 属性は、Windows Sync によって使用され、Windows Server で新しいユーザーが作成されたときに Directory Server が新しいユーザーエントリーを作成するかどうかを決定します。**true** は新しいエントリーを作成します。**false** は Windows エントリーを無視します。

OID	2.16.840.1.113730.3.1.42
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.262. ntUserDeleteAccount

**ntUserDeleteAccount** 属性は WindowsSync によって使用され、ユーザーが Windows 同期ピアサーバーから削除されたときに Directory Server エントリーが自動的に削除されるかどうかを決定します。**true** は、ユーザーエントリーを削除し、**false** は削除を無視します。

OID	2.16.840.1.113730.3.1.43
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.263. ntUserDomainId

**ntUserDomainId** 属性には、Windows ドメインのログイン ID が含まれています。以下に例を示します。

```
ntUserDomainId: jsmith
```

OID	2.16.840.1.113730.3.1.41
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.264. ntUserFlags

この属性には、Windows アカウントに設定された追加のフラグが含まれています。

OID	2.16.840.1.113730.3.1.523
構文	Binary
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.265. ntUserHomeDir

**ntUserHomeDir** 属性には、Windows ユーザーのホームディレクトリーを表す ASCII 文字列が含まれています。この属性は null にすることができます。以下に例を示します。

```
ntUserHomeDir: c:\jsmith
```

OID	2.16.840.1.113730.3.1.521
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.266. ntUserHomeDirDrive

この属性には、ユーザーのホームディレクトリーが保存されているドライブに関する情報が含まれています。

OID	2.16.840.1.113730.3.1.535
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.267. ntUserLastLogoff

**ntUserLastLogoff** 属性には、最後のログオフの時刻が含まれます。この値は、GMT 形式の文字列として保存されます。

セキュリティーロギングが有効な場合は、ユーザーエントリーの他の要素が変更になった場合にのみこの属性が同期時に更新されます。

ntUserLastLogoff: 20201015203415Z

OID	2.16.840.1.113730.3.1.527
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.268. ntUserLastLogon

**ntUserLastLogon** 属性には、ユーザーが最後に Windows ドメインにログインした時刻が含まれます。この値は、GMT 形式の文字列として保存されます。セキュリティーロギングが有効な場合は、ユーザーエントリーの他の要素が変更になった場合にのみこの属性が同期時に更新されます。

ntUserLastLogon: 20201015203415Z

OID	2.16.840.1.113730.3.1.526
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.269. ntUserLogonHours

**ntUserLogonHours** 属性には、ユーザーが Active Directory ドメインにログオンできる期間が含まれています。この属性は、Active Directory の **logonHours** 属性に対応します。

OID	2.16.840.1.113730.3.1.530
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.270. ntUserLogonServer

**ntUserLogonServer** 属性は、ユーザーのログオン要求の転送先となる Active Directory サーバーを定義します。

OID	2.16.840.1.113730.3.1.65
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.271. ntUserMaxStorage

**ntUserMaxStorage** 属性には、ユーザーが使用できる最大ディスク容量が含まれています。

**ntUserMaxStorage:** 4294967295

OID	2.16.840.1.113730.3.1.529
構文	Binary
多値または単一値	単一値

定義される場所	Netscape NT 同期
---------	----------------

### 5.2.272. ntUserNumLogons

この属性は、対象ユーザーの Active Directory ドメインへの正常なログオンの数を示します。

OID	2.16.840.1.113730.3.1.64
構文	Binary
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.273. ntUserParms

**ntUserParms** 属性には、アプリケーションで使用するために予約されている Unicode 文字列が含まれています。

OID	2.16.840.1.113730.3.1.62
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.274. ntUserPasswordExpired

この属性は、Active Directory アカウントのパスワードの有効期限が切れているかどうかを示します。

OID	2.16.840.1.113730.3.1.68
構文	Binary
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.275. ntUserPrimaryGroupld

**ntUserPrimaryGroupld** 属性には、ユーザーが属するプライマリーグループのグループ ID が含まれます。

OID	2.16.840.1.113730.3.1.534
-----	---------------------------

構文	Binary
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.276. ntUserPriv

この属性は、ユーザーに許可されている特権のタイプを指定します。

OID	2.16.840.1.113730.3.1.59
構文	Binary
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.277. ntUserProfile

**ntUserProfile** 属性には、ユーザーのプロファイルへのパスが含まれています。以下に例を示します。

```
ntUserProfile: c:\jsmith\profile.txt
```

OID	2.16.840.1.113730.3.1.67
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.278. ntUserScriptPath

**ntUserScriptPath** 属性には、ユーザーがドメインにログインするために使用する ASCII スクリプトへのパスが含まれています。

```
ntUserScriptPath: c:\jstorm\script.bat
```

OID	2.16.840.1.113730.3.1.524
構文	Binary



多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.279. ntUserUniqueld

**ntUserUniqueld** 属性には、Windows ユーザーの一意的な数値 ID が含まれています。

OID	2.16.840.1.113730.3.1.66
構文	Binary
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.280. ntUserUnitsPerWeek

**ntUserUnitsPerWeek** 属性には、ユーザーが Active Directory ドメインにログインしていた合計時間が含まれます。

OID	2.16.840.1.113730.3.1.63
構文	Binary
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.281. ntUserUsrComment

**ntUserUsrComment** 属性には、ユーザーに関する追加のコメントが含まれています。

OID	2.16.840.1.113730.3.1.61
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.282. ntUserWorkstations

**ntUserWorkstations** 属性には、ユーザーがログインできるワークステーションの名前のリストが ASCII 文字列で含まれています。最大 8 つのワークステーションをコンマで区切ってリストすることができます。ユーザーが任意のワークステーションからログオンできるようにするには、**null** を指定しま

す。以下に例を示します。

`ntUserWorkstations: firefly`

OID	2.16.840.1.113730.3.1.525
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape NT 同期

### 5.2.283. o (organizationName)

`organizationName` または `o` 属性には組織名が含まれます。以下に例を示します。

`organizationName: Example Corporation`  
`o: Example Corporation`

OID	2.5.4.10
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.284. objectClass

`objectClass` 属性は、エントリーに使用されるオブジェクトクラスを識別します。以下に例を示します。

`objectClass: person`

OID	2.5.4.0
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.285. objectClasses

この属性は、サブスキーマ定義で許可されているオブジェクトクラスを識別するためにスキーマファイルで使用されます。

OID	2.5.21.6
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2252</a>

### 5.2.286. `obsoletedByDocument`

`obsoletedByDocument` 属性には、ドキュメントの識別名が含まれます。これにより、現在のドキュメントエントリーが非推奨になります。

OID	0.9.2342.19200300.102.1.4
構文	DN
多値または単一値	複数値
定義される場所	Internet White Pages Pilot

### 5.2.287. `obsoletesDocument`

`obsoletesDocument` 属性にはドキュメントの識別名が含まれます。これにより、現在のドキュメントエントリーが非推奨になります。

OID	0.9.2342.19200300.102.1.3
構文	DN
多値または単一値	複数値
定義される場所	Internet White Pages Pilot

### 5.2.288. `oncRpcNumber`

`oncRpcNumber` 属性には、RPC マップの一部が含まれ、UNIXRPC の RPC 番号が格納されます。



#### 注記

`oncRpcNumber` 属性は、Directory Server の `10rfc2307.ldif` で定義されます。更新された RFC 2307 スキーマを使用するには、`10rfc2307.ldif` ファイルを削除し、`/usr/share/dirsrv/data` ディレクトリーの `10rfc2307bis.ldif` ファイルを `/etc/dirsrv/slapd-instance/schema` ディレクトリーにコピーします。

OID	1.3.6.1.1.1.18
-----	----------------

構文	整数
多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.289. organizationalStatus

**organizationalStatus** は、組織内の個人のカテゴリを識別します。

```
organizationalStatus: researcher
```

OID	0.9.2342.19200300.100.1.45
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.290. otherMailbox

**otherMailbox** 属性には、X.400 および RFC822 以外の電子メールタイプの値が含まれています。

```
otherMailbox: internet $ jsmith@example.com
```

OID	0.9.2342.19200300.100.1.22
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.291. ou (organizationalUnitName)

**OrganizationalUnitName**、または **ou** には、ディレクトリー階層内の組織部門またはサブツリーの名前が含まれます。

```
organizationalUnitName: Marketing
ou: Marketing
```

OID	2.5.4.11
構文	DirectoryString

多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.292. owner

**所有者** 属性には、エントリーの責任者の DN が含まれます。以下に例を示します。

```
owner: cn=John Smith,ou=people,dc=example,dc=com
```

OID	2.5.4.32
構文	DN
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.293. pager

**pagerTelephoneNumber** または **pager** 属性には、個人のポケットベルの電話番号が含まれます。

```
pagerTelephoneNumber: 415-555-6789
pager: 415-555-6789
```

OID	0.9.2342.19200300.100.1.42
構文	TelephoneNumber
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.294. parentOrganization

**parentOrganization** 属性は、組織または組織単位の親組織を識別します。

OID	1.3.6.1.4.1.1466.101.120.41
構文	DN
多値または単一値	単一値
定義される場所	Netscape

### 5.2.295. personalSignature

**personalSignature** 属性には、エントリーの署名ファイルがバイナリー形式で含まれています。

```
personalSignature:: AAAAAA==
```

OID	0.9.2342.19200300.100.1.53
構文	Binary
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.296. personalTitle

**personalTitle** 属性には、**Ms.**、**Dr.**、**Prof.**、および **Rev.** などの敬称が含まれます。

```
personalTitle: Mr.
```

OID	0.9.2342.19200300.100.1.40
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.297. photo

**photo** 属性には、バイナリー形式の写真ファイルが含まれています。

```
photo:: AAAAAA==
```

OID	0.9.2342.19200300.100.1.7
構文	Binary
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.298. physicalDeliveryOfficeName

**physicalDeliveryOffice** には、実際に郵便配達オフィスが配置されている市または町が含まれていません。

physicalDeliveryOfficeName: Raleigh

OID	2.5.4.19
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.299. postalAddress

**postalAddress** 属性は、エントリーの郵送先住所を識別します。このフィールドは、複数行を含めることが想定されています。LDIF 形式で表す場合には、各行はドル記号 (\$) で区切る必要があります。

エントリーテキスト内で実際のドル記号 (\$) または円記号 (\) を表すには、エスケープされた 16 進値 `\24` および `\5c` をそれぞれ使用します。たとえば、文字列を表すには、次のようにします。

The dollar (\$) value can be found  
in the c:\cost file.

文字列を指定します。

The dollar (\24) value can be found\$in the c:\5ccost file.

OID	2.5.4.16
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.300. postalCode

**postalCode** には、米国内にあるエントリーの郵便番号が含まれます。

postalCode: 44224

OID	2.5.4.17
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.301. postOfficeBox

**postOfficeBox** 属性には、エントリーの実際の郵送先住所の番地または私書箱番号が含まれます。

```
postOfficeBox: 1234
```

OID	2.5.4.18
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.302. preferredDeliveryMethod

**PreferredDeliveryMethod** には、エントリーの希望の連絡先または配信方法が含まれています。以下に例を示します。

```
preferredDeliveryMethod: telephone
```

OID	2.5.4.28
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.303. preferredLanguage

**preferredLanguage** 属性には、ユーザーが希望する使用言語が含まれています。値は、HTTP Accept-Language ヘッダー値の構文に準拠している必要があります。

OID	2.16.840.1.113730.3.1.39
構文	DirectoryString
多値または単一値	単一値
定義される場所	<a href="#">RFC 2798</a>

### 5.2.304. preferredLocale

**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などが含まれます。Directory Server は、アメリカ英語、日本語、およびドイツ語の3つのロケールを



サポートしています。

**PreferredLocale** 属性は、ユーザーが優先するロケールを設定します。

OID	1.3.6.1.4.1.1466.101.120.42
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape

### 5.2.305. preferredTimeZone

**PreferredTimeZone** 属性は、ユーザーエントリーに使用するタイムゾーンを設定します。

OID	1.3.6.1.4.1.1466.101.120.43
構文	DirectoryString
多値または単一値	単一値
定義される場所	Netscape

### 5.2.306. presentationAddress

**presentaddress** 属性には、エントリーの OSI のプレゼンテーションアドレスが含まれます。この属性には、OSI ネットワークアドレスと最大3つのセクターが含まれます。各セクターは、トランスポート、セッション、およびプレゼンテーションエンティティーで使用されます。以下に例を示します。

```
presentationAddress: TELEX+00726322+RFC-1006+02+130.59.2.1
```

OID	2.5.4.29
構文	IA5String
多値または単一値	単一値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.307. protocolInformation

**protocolInformation** 属性は、**presentationAddress** 属性と併用され、OSO ネットワークサービスに関する追加情報を提供します。

OID	2.5.4.48
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.308. pwdReset

管理者がユーザーのパスワードを変更すると、Directory Server は、ユーザーのエントリーの **pwdReset** 操作属性を **true** に設定します。アプリケーションはこの属性を使用して、管理者がユーザーのパスワードをリセットしたかどうかを識別できます。



#### 注記

**pwdReset** 属性は操作属性であるため、ユーザーは編集できません。

OID	1.3.6.1.4.1.1466.115.121.1.7
構文	Boolean
多値または単一値	単一値
定義される場所	<a href="#">RFC draft-behera-ldap-password-policy</a>

### 5.2.309. ref

**ref** 属性は、LDAPv3 スマート参照のサポートに使用されます。この属性の値は LDAPURL です。

ldap: **host\_name:port\_number/subtree\_dn**

ポート番号はオプションです。

以下に例を示します。

ref: ldap://server.example.com:389/ou=People,dc=example,dc=com

OID	2.16.840.1.113730.3.1.34
構文	IA5String
多値または単一値	複数値
定義される場所	LDAPv3 参照インターネットドラフト

### 5.2.310. registeredAddress

この属性には、電報または速達文書を受け取る住所が含まれます。通常、配達時に受取人の署名が必要です。

OID	2.5.4.26
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.311. roleOccupant

この属性には、**organizationalRole** エントリーで定義されたロールが割り当てられたユーザーの識別名が含まれます。

```
roleOccupant: uid=bjensen,dc=example,dc=com
```

OID	2.5.4.33
構文	DN
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.312. roomNumber

この属性は、オブジェクトの部屋番号を指定します。**cn** 属性は、部屋オブジェクトの命名に使用する必要があります。

```
roomNumber: 230
```

OID	0.9.2342.19200300.100.1.6
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.313. searchGuide

**searchGuide** 属性は、検索操作のディレクトリーツリーでエントリーをベースオブジェクトとして使用する場合に、推奨される検索条件の情報を指定します。検索フィルターの作成時には、代わりに **enhancedSearchGuide** 属性を使用してください。

OID	2.5.4.14
構文	IA5String
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.314. secretary

**secretary** 属性は、エントリーの秘書または管理アシスタントを識別します。

```
secretary: cn=John Smith,dc=example,dc=com
```

OID	0.9.2342.19200300.100.1.21
構文	DN
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.315. seeAlso

**seeAlso** 属性は、このエントリーに関連する情報を含む可能性のある別の Directory Server エントリーを識別します。

```
seeAlso: cn=Quality Control Inspectors,ou=manufacturing,dc=example,dc=com
```

OID	2.5.4.34
構文	DN
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.316. serialNumber

**serialNumber** 属性には、デバイスのシリアル番号が含まれています。

```
serialNumber: 555-1234-AZ
```

OID	2.5.4.5
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.317. serverHostName

**serverHostName** 属性には、Directory Server が実行されているサーバーのホスト名が含まれます。

OID	2.16.840.1.113730.3.1.76
構文	DirectoryString
多値または単一値	複数値
定義される場所	Red Hat 管理サービス

### 5.2.318. serverProductName

**serverProductName** 属性には、サーバー製品の名前が含まれています。

OID	2.16.840.1.113730.3.1.71
構文	DirectoryString
多値または単一値	複数値
定義される場所	Red Hat 管理サービス

### 5.2.319. serverRoot

この属性は廃止されました。

この属性は、Directory Server バージョン 7.1 以前のインストールディレクトリー (サーバールート) を示します。

OID	2.16.840.1.113730.3.1.70
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape 管理サービス

### 5.2.320. serverVersionNumber

**serverVersionNumber** 属性には、サーバーのバージョン番号が含まれます。

OID	2.16.840.1.113730.3.1.72
構文	DirectoryString
多値または単一値	複数値
定義される場所	Red Hat 管理サービス

### 5.2.321. shadowExpire

**shadowExpire** 属性には、シャドウアカウントの有効期限が切れる日付が含まれます。日付の形式は、UTC での EPOCH からの日数です。システムでこれを計算するには、現在の日付に **-d** を使用し、UTC に **-u** を使用して、次のようなコマンドを実行します。

```
$ echo date -u -d 20100108 +%s /24/60/60 |bc
```

```
14617
```

結果 (例では 14617) は、**shadowExpire** の値になります。

```
shadowExpire: 14617
```



#### 注記

**shadowExpire** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slaped-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.10
構文	整数
多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.322. shadowFlag

**shadowFlag** 属性は、シャドウマップのどの領域にフラグ値を格納するかを識別します。

shadowFlag: 150



## 注記

**shadowFlag** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.11
構文	整数
多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

## 5.2.323. shadowInactive

**shadowInactive** 属性は、シャドウアカウントを非アクティブにできる期間を日数で設定します。

shadowInactive: 15



## 注記

**shadowInactive** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.9
構文	整数
多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

## 5.2.324. shadowLastChange

**shadowLastChange** 属性には、1970 年 1 月 1 日からユーザーパスワードの最終設定日までの日数が含まれます。たとえば、アカウントのパスワードが 2016 年 11 月 4 日に最後に設定された場合は、**shadowLastChange** 属性は **0** に設定しておきます。

次の例外があります。

- **cn=config** エントリーで **passwordMustChange** パラメーターが有効になっている場合に、新しいアカウントでは、**shadowLastChange** 属性に **0** が設定されます。

- パスワードなしでアカウントを作成すると、**shadowLastChange** 属性は追加されません。

**shadowLastChange** 属性は、Active Directory から同期されたアカウントに対して自動的に更新されません。



#### 注記

**shadowLastChange** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.5
構文	整数
多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.325. shadowMax

**shadowMax** 属性は、シャドウパスワードが有効である最大日数を設定します。

```
shadowMax: 10
```



#### 注記

**shadowMax** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.7
構文	整数
多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.326. shadowMin

**shadowMin** 属性は、シャドウパスワードを変更するまでに最小限経過する必要がある日数を設定します。

```
shadowMin: 3
```





### 注記

**shadowMin** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.6
構文	整数
多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.327. shadowWarning

**shadowWarning** 属性は、パスワードの有効期限が切れる何日前にユーザーに警告を送信するかを設定します。

```
shadowWarning: 2
```



### 注記

**shadowWarning** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.8
構文	整数
多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.328. singleLevelQuality

**singleLevelQuality** は、ディレクトリーツリーのすぐ下のレベルでのデータ品質を指定します。

OID	0.9.2342.19200300.100.1.50
構文	DirectoryString
多値または単一値	単一値

定義される場所	<a href="#">RFC 1274</a>
---------	--------------------------

### 5.2.329. sn (surname)

**surname** または **sn** 属性には、エントリーの **姓** (名字など) が含まれます。

```

surname: Jensen
sn: Jensen

```

OID	2.5.4.4
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.330. st (stateOrProvinceName)

**stateOrProvinceName** または **st** 属性には、エントリーの州が含まれます。

```

stateOrProvinceName: California
st: California

```

OID	2.5.4.8
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.331. street

**streetAddress** または **street** 属性には、エントリーの番地と住所が含まれます。

```

streetAddress: 1234 Ridgeway Drive
street: 1234 Ridgeway Drive

```

OID	2.5.4.9
構文	DirectoryString
多値または単一値	複数値

定義される場所	<a href="#">RFC 2256</a>
---------	--------------------------

### 5.2.332. サブジェクト (subject)

**subject** 属性には、ドキュメントエントリーのサブジェクトに関する情報が含まれています。

subject: employee option grants

OID	0.9.2342.19200300.102.1.8
構文	DirectoryString
多値または単一値	複数値
定義される場所	Internet White Pages Pilot

### 5.2.333. subtreeMaximumQuality

**subtreeMaximumQuality** 属性は、ディレクトリーサブツリーの最大データ品質を指定します。

OID	0.9.2342.19200300.100.1.52
構文	DirectoryString
多値または単一値	単一値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.334. subtreeMinimumQuality

**subtreeMinimumQuality** は、ディレクトリーサブツリーの最小データ品質を指定します。

OID	0.9.2342.19200300.100.1.51
構文	DirectoryString
多値または単一値	単一値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.335. supportedAlgorithms

**supportedAlgorithms** 属性には、**supportedAlgorithms;binary** などのバイナリー形式で要求および保存されるアルゴリズムが含まれています。

```
supportedAlgorithms:: AAAAAA==
```

OID	2.5.4.52
構文	Binary
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.336. supportedApplicationContext

この属性には、OSI アプリケーションコンテキストの識別子が含まれています。

OID	2.5.4.30
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.337. telephoneNumber

**TelephoneNumber** には、エントリーの電話番号が含まれています。以下に例を示します。

```
telephoneNumber: 415-555-2233
```

OID	2.5.4.20
構文	TelephoneNumber
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.338. teletexTerminalIdentifier

**teletexTerminalIdentifier** 属性には、エントリーのテレテックス端末識別子が含まれています。この例で最初に出力できる文字列は、エンコードするテレテックス端末識別子の最初の部分で、次にくる 0 個以上のオクテット文字列が、テレテックス端末識別子の続きの部分となります。

```
teletex-id = ttx-term 0*("$" ttx-param)
ttx-term = printablestring
ttx-param = ttx-key ":" ttx-value
```

```
ttx-key = "graphic" / "control" / "misc" / "page" / "private"
ttx-value = octetstring
```

OID	2.5.4.22
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.339. telexNumber

この属性は、エントリーのテレックス番号を定義します。テレックス番号の形式は次のとおりです。

```
actual-number "$" country "$" answerback
```

- **actual-number** は、エンコードされているテレックス番号の番号部分の構文表現です。
- **country** は TELEX の国コードです。
- **answerback** は、TELEX 端末のアンサーバックコードです。

OID	2.5.4.21
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.340. title

**title** 属性には、組織内の個人の役職が含まれます。

```
title: Senior QC Inspector
```

OID	2.5.4.12
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.341. ttl (TimeToLive)

**TimeToLive** または **ttl** 属性には、キャッシュされているエントリーの情報が有効とみなされる時間 (秒) が含まれます。指定された時間が経過すると、情報は古くなったと見なされます。ゼロ (**0**) の値は、エントリーをキャッシュしてはならないことを指定します。

```
TimeToLive: 120
ttl: 120
```

OID	1.3.6.1.4.250.1.60
構文	DirectoryString
多値または単一値	複数値
定義される場所	LDAP キャッシングインターネットドラフト

### 5.2.342. uid (userID)

**userID** (より一般的には **uid**) 属性には、エントリーの一意的ユーザー名が含まれます。

```
userID: jsmith
uid: jsmith
```

OID	0.9.2342.19200300.100.1.1
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.343. uidNumber

**uidNumber** 属性には、ユーザーエントリーの一意的数値識別子が含まれています。これは、Unix のユーザー番号に似ています。

```
uidNumber: 120
```



#### 注記

**uidNumber** 属性は、Directory Server の **10rfc2307.ldif** で定義されます。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

OID	1.3.6.1.1.1.1.0
-----	-----------------

構文	整数
多値または単一値	単一値
定義される場所	<a href="#">RFC 2307</a>

### 5.2.344. uniqueIdentifier

この属性は、識別名が再利用されたときに2つのエントリーを区別するために使用される特定の項目を識別します。この属性は、削除された識別名への参照のインスタンスを検出することを目的としています。この属性はサーバーによって割り当てられます。

uniqueIdentifier:: AAAAAA==

OID	0.9.2342.19200300.100.1.44
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.345. uniqueMember

**uniqueMember** 属性は、エントリーに関連付けられた名前のグループを識別します。名前ごとに、一意性を確保するために **uniqueIdentifier** が割り当てられています。**uniqueMember** 属性の値は、DN の後に **uniqueIdentifier** が続きます。

OID	2.5.4.50
構文	DN
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.346. updatedByDocument

**updatedByDocument** 属性には、ドキュメントエントリーの更新バージョンであるドキュメントの識別名が含まれています。

OID	0.9.2342.19200300.102.1.6
構文	DN
多値または単一値	複数値

定義される場所	Internet White Pages Pilot
---------	----------------------------

### 5.2.347. updatesDocument

**updatesDocument** 属性には、このドキュメントの更新版であるドキュメントの識別名が含まれます。

OID	0.9.2342.19200300.102.1.5
構文	DN
多値または単一値	複数値
定義される場所	Internet White Pages Pilot

### 5.2.348. userCertificate

この属性は、**userCertificate;binary** として、バイナリー形式で保存および要求されます。

```
userCertificate;binary:: AAAAAA==
```

OID	2.5.4.36
構文	Binary
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.349. userClass

この属性は、コンピューターユーザーのカテゴリを指定します。この属性のセマンティクスは任意です。**OrganizationalStatus** 属性は、コンピューターユーザーと他のタイプのユーザーを区別しません。ユーザーのほうがより適切な場合があります。

```
userClass: intern
```

OID	0.9.2342.19200300.100.1.8
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 5.2.350. userPassword



この属性は、エントリーのパスワードと暗号化方式を `{encryption method}encrypted password` の形式で識別します。以下に例を示します。

```
userPassword: {sha}FTSLQhxXpA05
```

基盤となるトランスポートサービスが機密性を保証できない場合に、クリアテキストのパスワードを転送しないようにすることを強く推奨します。クリアテキストで転送すると、許可されていない第三者にパスワードが開示される可能性があります。

OID	2.5.4.35
構文	Binary
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.351. userPKCS12

この属性は、個人の ID 情報を交換するための形式を提供します。属性は、**userPKCS12;binary** として、バイナリー形式で保存および要求されます。属性値は、バイナリーデータとして保存された PFXPDU です。

OID	2.16.840.1.113730.3.1.216
構文	Binary
多値または単一値	複数値
定義される場所	<a href="#">RFC 2798</a>

### 5.2.352. userSMIMECertificate

**userSMIMECertificate** 属性には、メールクライアントが S/MIME に使用できる証明書が含まれています。この属性は、データをバイナリー形式で要求して保存します。以下に例を示します。

```
userSMIMECertificate;binary:: AAAAAA==
```

OID	2.16.840.1.113730.3.1.40
構文	Binary
多値または単一値	複数値
定義される場所	<a href="#">RFC 2798</a>

### 5.2.353. vacationEndDate

この属性は、ユーザーの休暇期間の終了日を指定します。

OID	2.16.840.1.113730.3.1.708
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.354. vacationStartDate

この属性は、ユーザーの休暇期間の開始日を指定します。

OID	2.16.840.1.113730.3.1.707
構文	DirectoryString
多値または単一値	複数値
定義される場所	Netscape Messaging Server

### 5.2.355. x121Address

**x121Address** 属性には、ユーザーの X.121 アドレスが含まれます。

OID	2.5.4.24
構文	IA5String
多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.2.356. x500UniqueIdentifier

将来の使用のために予約されています。X.500 識別子は、識別名が再利用されたときにオブジェクトを区別するのに役立つバイナリーの識別方法です。

x500UniqueIdentifier:: AAAAAA==

OID	2.5.4.45
構文	Binary

多値または単一値	複数値
定義される場所	<a href="#">RFC 2256</a>

### 5.3. エントリーオブジェクトクラスの参照

この参照は、デフォルトのスキーマで受け入れられるオブジェクトクラスのアルファベット順のリストです。各オブジェクトクラスの定義を示し、その必須属性と使用できる属性をリスト表示します。リストされているオブジェクトクラスは、エントリー情報をサポートするために使用できます。

オブジェクトクラスにリストされている必須属性は、そのオブジェクトクラスをディレクトリーの **ldif** ファイルに追加する時にエントリーに存在している必要があります。オブジェクトクラスに、親のオブジェクトクラスがある場合は、必要なすべての属性を持つこれらのオブジェクトクラスの両方がエントリーに存在する必要があります。必要な属性が **ldif** ファイルにリストされていない場合には、サーバーは再起動しません。



#### 注記

LDAP RFC および X.500 標準では、オブジェクトクラスに複数の親のオブジェクトクラスを含めることができます。この動作は現在、Directory Server ではサポートされていません。

#### 5.3.1. アカウント

アカウント オブジェクトクラスは、コンピューターアカウントのエントリーを定義します。このオブジェクトクラスは [RFC 1274](#) に定義されています。

#### 上級クラス

top

#### OID

0.9.2342.19200300.100.4.5

表5.3 必要な属性

属性	定義
<a href="#">「objectClass」</a>	エントリーのオブジェクトクラスを指定します。
<a href="#">「uid (userID)」</a>	定義されたアカウントのユーザー ID を指定します。

表5.4 使用できる属性

属性	定義
<a href="#">「description」</a>	エントリーのテキスト説明を入力します。
<a href="#">「host」</a>	アカウントが存在するマシンのホスト名を指定します。

属性	定義
<code>l (localityName)</code>	エントリーの市または地理的な場所を指定します。
<code>o (organizationName)</code>	アカウントが属する組織を指定します。
<code>ou (organizationalUnitName)</code>	アカウントが属する組織単位または部門を指定します。
<code>seeAlso</code>	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。

### 5.3.2. accountpolicy

**accountpolicy** オブジェクトクラスは、アカウントの非アクティブ化または有効期限ポリシーのエントリーを定義します。これは、アカウントポリシープラグイン設定と連携して機能するユーザーディレクトリー設定のエントリーに使用されます。

#### 上級クラス

top

#### OID

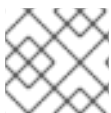
1.3.6.1.4.1.11.1.3.2.2.1

表5.5 使用できる属性

属性	定義
<code>accountInactivityLimit</code>	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。

### 5.3.3. alias

**alias** オブジェクトクラスは、他のディレクトリーエントリーを参照します。このオブジェクトクラスは [RFC 2256](#) に定義されています。



#### 注記

エントリーのエイリアス作成は Red Hat Directory Server ではサポートされていません。

#### 上級クラス

top

#### OID

2.5.6.1

表5.6 必要な属性

属性	定義
「objectClass」	エントリーのオブジェクトクラスを定義します。
「aliasedObjectName」	エントリーがエイリアスの場合にエントリーの識別名を指定します。

### 5.3.4. bootableDevice

**bootableDevice** オブジェクトクラスは、boot パラメーターが割り当てられたデバイスを指定します。このオブジェクトクラスは RFC 2307 に定義されています。



#### 注記

このオブジェクトクラスは、Directory Server の **10rfc2307.ldif** で定義されています。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

#### 上級クラス

top

#### OID

1.3.6.1.1.1.2.12

表5.7 必要な属性

属性	定義
「objectClass」	エントリーのオブジェクトクラスを定義します。
「cn (commonName)」	デバイスの一般名を指定します。

表5.8 使用できる属性

属性	定義
「bootFile」	ブートイメージファイルを指定します。
「bootParameter」	デバイスの起動プロセスで使用されるパラメーターを指定します。
「description」	エントリーのテキスト説明を入力します。
「l (localityName)」	エントリーの市または地理的な場所を指定します。
「o (organizationName)」	デバイスが属する組織を指定します。

属性	定義
<code>「ou (organizationalUnitName)」</code>	デバイスが属する組織単位または部門を指定します。
<code>「owner」</code>	デバイスに対応するユーザーの DN (識別名) を指定します。
<code>「seeAlso」</code>	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
<code>「serialNumber」</code>	デバイスのシリアル番号が含まれます。

### 5.3.5. cacheObject

**cacheObject** は、存続時間 (**ttl**) 属性タイプを含むオブジェクトです。このオブジェクトクラスは、LDAP キャッシングインターネットドラフトで定義されています。

#### 上級クラス

top

#### OID

1.3.6.1.4.1.250.3.18

表5.9 必要な属性

属性	定義
<code>「objectClass」</code>	エントリーのオブジェクトクラスを定義します。

表5.10 使用できる属性

属性	定義
<code>「ttl (TimeToLive)」</code>	オブジェクトがキャッシュに残っている (存続している) 時間。

### 5.3.6. cosClassicDefinition

**cosClassicDefinition** オブジェクトクラスは、`「cosTemplateDn」` 属性で指定されたエントリーの DN (識別名) と、`「cosSpecifier」` 属性で指定されたターゲット属性の1つの値を使用して、サービスクラステンプレートエントリーを定義します。

このオブジェクトクラスは [RFC 1274](#) に定義されています。

#### 上級クラス

cosSuperDefinition

## OID

2.16.840.1.113730.3.2.100

表5.11 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cosAttribute」	CoS が値を生成する属性の名前を指定します。複数の <b>cosAttribute</b> 値を指定できます。

表5.12 使用できる属性

属性	定義
「cn (commonName)」	エントリーの共通名を指定します。
「cosSpecifier」	従来の CoS が使用する属性値を指定します。これは、テンプレートエントリーの DN とともに、テンプレートエントリーを識別します。
「cosTemplateDn」	CoS 定義に関連付けられたテンプレートエントリーの DN を提供します。
「description」	エントリーのテキスト説明を入力します。

## 5.3.7. cosDefinition

**cosDefinition** オブジェクトクラスは、使用されているサービスクラスを定義します。このオブジェクトクラスで、DS4.1CoS プラグインとの互換性を確保します。

このオブジェクトクラスは [RFC 1274](#) に定義されています。

## 上級クラス

top

## OID

2.16.840.1.113730.3.2.84

表5.13 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。

表5.14 使用できる属性

属性	定義
「aci」	Directory Server がクライアントから LDAP 要求を受信したときに付与または拒否される権限を評価します。
「cn (commonName)」	エントリーの共通名を指定します。
「cosAttribute」	CoS が値を生成する属性の名前を指定します。複数の <b>cosAttribute</b> 値を指定できます。
「cosSpecifier」	従来 CoS が使用する属性値を指定します。これは、テンプレートエントリーの DN とともに、テンプレートエントリーを識別します。
「cosTargetTree」	CoS スキーマが適用されるディレクトリー内のサブツリーを定義します。
「cosTemplateDn」	CoS 定義に関連付けられたテンプレートエントリーの DN を提供します。
「uid (userID)」	エントリーのユーザー ID を指定します。

### 5.3.8. cosIndirectDefinition

**cosIndirectDefinition** は、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定義します。ターゲットエントリーの属性は、「[cosIndirectSpecifier](#)」属性で指定されます。

このオブジェクトクラスは Directory Server で定義されます。

#### 上級クラス

cosSuperDefinition

#### OID

2.16.840.1.113730.3.2.102

表5.15 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cosAttribute」	CoS が値を生成する属性の名前を指定します。複数の <b>cosAttribute</b> 値を指定できます。

表5.16 使用できる属性



属性	定義
「cn (commonName)」	エントリーの共通名を指定します。
「cosIndirectSpecifier」	テンプレートエントリーを識別するために間接 CoS が使用する属性値を指定します。
「description」	エントリーのテキスト説明を入力します。

### 5.3.9. cosPointerDefinition

このオブジェクトクラスは、テンプレートエントリーの DN 値を使用して、CoS 定義に関連付けられたテンプレートエントリーを識別します。テンプレートエントリーの DN は、「cosIndirectSpecifier」属性で指定されます。

このオブジェクトクラスは Directory Server で定義されます。

#### 上級クラス

cosSuperDefinition

#### OID

2.16.840.1.113730.3.2.101

表5.17 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cosAttribute」	CoS が値を生成する属性の名前を指定します。複数の <b>cosAttribute</b> 値を指定できます。

表5.18 使用できる属性

属性	定義
「cn (commonName)」	エントリーの共通名を指定します。
「cosTemplateDn」	CoS 定義に関連付けられたテンプレートエントリーの DN を提供します。
「description」	エントリーのテキスト説明を入力します。

### 5.3.10. cosSuperDefinition

すべての CoS 定義オブジェクトクラスは、**cosSuperDefinition** オブジェクトクラスを継承します。

このオブジェクトクラスは Directory Server で定義されます。

## 上級クラス

LDAPsubentry

## OID

2.16.840.1.113730.3.2.99

表5.19 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cosAttribute」	CoS が値を生成する属性の名前を指定します。複数の <b>cosAttribute</b> 値を指定できます。

表5.20 使用できる属性

属性	定義
「cn (commonName)」	エントリーの共通名を指定します。
「description」	エントリーのテキスト説明を入力します。

### 5.3.11. cosTemplate

**cosTemplate** オブジェクトクラスには、CoS の共有属性値のリストが含まれています。

このオブジェクトクラスは Directory Server で定義されます。

## 上級クラス

top

## OID

2.16.840.1.113730.3.2.128

表5.21 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。

表5.22 使用できる属性

属性	定義
「cn (commonName)」	エントリーの共通名を指定します。
「cosPriority」	属性値の指定時に CoS テンプレートが競合する場合に、どのテンプレートがその属性値を提供するかを指定します。

### 5.3.12. country

**country** オブジェクトクラスは、国を表すエントリーを定義します。このオブジェクトクラスは [RFC 2256](#) に定義されています。

#### 上級クラス

top

#### OID

2.5.6.2

表5.23 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「c (countryName)」	ディレクトリー内の ISO で定義されている国名を表す 2 文字のコードが含まれます。

表5.24 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。
「searchGuide」	エントリーを検索用にディレクトリーツリーのベースオブジェクトとして使用する場合に、提案した検索条件の情報を指定します。

### 5.3.13. dcObject

**dcObject** オブジェクトクラスを使用すると、エントリーに対してドメインコンポーネントを定義できます。このオブジェクトクラスは、一般的に **o (organization)**、**ou (organizationalUnit)**、**l (locality)** などの別のオブジェクトクラスと併用されるため、補助として定義されます。

以下に例を示します。

```
dn: dc=example,dc=com
objectClass: top
```

```
objectClass: organizationalUnit
objectClass: dcObject
dc: example
ou: Example Corporation
```

このオブジェクトクラスは [RFC 2247](#) に定義されています。

## 上級クラス

top

## OID

1.3.6.1.4.1.1466.344

表5.25 必要な属性

属性	定義
<a href="#">「objectClass」</a>	エントリーに割り当てられたオブジェクトクラスを指定します。
<a href="#">「dc (domainComponent)」</a>	ドメイン名の1つのコンポーネントが含まれます。

### 5.3.14. device

**device** オブジェクトクラスは、プリンターなどのネットワークデバイスに関する情報をディレクトリーに格納します。このオブジェクトクラスは [RFC 2247](#) に定義されています。

## 上級クラス

top

## OID

2.5.6.14

表5.26 必要な属性

属性	定義
<a href="#">「objectClass」</a>	デバイスに割り当てられたオブジェクトクラスを指定します。
<a href="#">「cn (commonName)」</a>	デバイスの一般名を指定します。

表5.27 使用できる属性

属性	定義
<a href="#">「description」</a>	エントリーのテキスト説明を入力します。
<a href="#">「l (localityName)」</a>	エントリーの市または地理的な場所を指定します。

属性	定義
「o (organizationName)」	デバイスが属する組織を指定します。
「ou (organizationalUnitName)」	デバイスが属する組織単位または部門を指定します。
「owner」	デバイスに対応するユーザーの DN (識別名) を指定します。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「serialNumber」	デバイスのシリアル番号が含まれます。

### 5.3.15. document

**document** オブジェクトクラスは、ドキュメントを表すディレクトリーエントリーを定義します。RFC 1247

#### 上級クラス

top

#### OID

0.9.2342.19200300.100.4.6

表5.28 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「documentIdentifier」	ドキュメントの一意の ID を提供します。

表5.29 使用できる属性

属性	定義
「abstract」	ドキュメントの要約が含まれています。
「audio」	サウンドファイルをバイナリー形式で格納します。
「authorCn」	著者の一般名または通称を付けます。

属性	定義
「authorSn」	作成者の名前を指定します。
「cn (commonName)」	エントリーの共通名を指定します。
「description」	エントリーのテキスト説明を入力します。
「dITRedirect」	ドキュメントエントリーのリダイレクトとして使用するエントリーの DN (識別名) が含まれます。
「documentAuthor」	作成者の DN (識別名) が含まれます。
「documentLocation」	元のドキュメントの場所を示します。
「documentPublisher」	ドキュメントを公開した個人または組織を識別します。
「documentStore」	
「documentTitle」	ドキュメントのタイトルが含まれています。
「documentVersion」	ドキュメントのバージョン番号を示します。
「info」	ドキュメントに関する情報が含まれています。
「jpegPhoto」	JPG イメージを保存します。
「keyWords」	ドキュメントに関連するキーワードが含まれています。
「l (localityName)」	エントリーの市または地理的な場所を指定します。
「lastModifiedBy」	ドキュメントエントリーを修正した最終ユーザーの DN (識別名) を指定します。
「lastModifiedTime」	最後の変更の時刻を示します。
「manager」	エントリーマネージャーの DN (識別名) を指定します。
「o (organizationName)」	ドキュメントが属する組織を指定します。
「obsoletedByDocument」	対象のドキュメントを置き換える、別のドキュメントエントリーの DN (識別名) を指定します。
「obsoletesDocument」	このドキュメントで置き換える、別のドキュメントエントリーの DN (識別名) を指定します。

属性	定義
「ou (organizationalUnitName)」	ドキュメントが属する組織単位または部門を指定します。
「photo」	ドキュメントの写真をバイナリー形式で保存します。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「サブジェクト (subject)」	ドキュメントの主題について説明します。
「uniqueIdentifier」	識別名を再利用する場合は、2つのエントリーを区別します。
「updatedByDocument」	対象のドキュメントを <b>更新する</b> 別のドキュメントエントリーの DN (識別名) を指定します。
「updatesDocument」	対象のドキュメントで <b>更新される</b> 別のドキュメントエントリーの DN (識別名) を指定します。

### 5.3.16. documentSeries

**documentSeries** オブジェクトクラスは、一連のドキュメントを表すエントリーを定義します。このオブジェクトクラスは [RFC 1274](#) に定義されています。

#### 上級クラス

top

#### OID

0.9.2342.19200300.100.4.9

表5.30 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.31 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。

属性	定義
「l (localityName)」	ドキュメントシリーズが物理的に配置されている場所を指定します。
「o (organizationName)」	ドキュメントシリーズが属する組織を指定します。
「ou (organizationalUnitName)」	シリーズが属する組織単位または部門を指定します。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「telephoneNumber」	一連のドキュメントの担当者の電話番号を指定します。

### 5.3.17. domain

**domain** オブジェクトクラスは、DNS ドメインを表すディレクトリーエントリーを定義します。「dc (domainComponent)」属性を使用して、このオブジェクトクラスのエントリーに名前を付けます。

このオブジェクトクラスは、**example.com** などのインターネットドメイン名にも使用されます。

**domain** オブジェクトクラスは、組織、組織単位、またはオブジェクトクラスが定義されているその他のオブジェクトに対応しないディレクトリーエントリーにのみ使用できます。

このオブジェクトクラスは [RFC 2252](#) で定義されます。

#### 上級クラス

top

#### OID

0.9.2342.19200300.100.4.13

表5.32 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「dc (domainComponent)」	ドメイン名の1つのコンポーネントが含まれます。

表5.33 使用できる属性

属性	定義
「associatedName」	DNS ドメインに関連付けられた組織ディレクトリーツリー内のエントリー名を指定します。



属性	定義
「businessCategory」	このドメインが従事しているビジネスの種類を指定します。
「description」	エントリーのテキスト説明を入力します。
「destinationIndicator」	エントリーに関連付けられている国と都市を指定します。これはかつて、公共の電報サービスを提供するのに必要でした。
「fax (facsimileTelephoneNumber)」	ドメインの FAX 番号を指定します。
「internationalISDNNumber」	ドメインの ISDN 番号を示します。
「l (localityName)」	エントリーの市または地理的な場所を指定します。
「o (organizationName)」	エントリーが属する組織を指定します。
「physicalDeliveryOfficeName」	物理的な配送が可能な場所を提供します。
「postOfficeBox」	ドメインの私書箱番号を示します。
「postalAddress」	ドメインのメールアドレスが含まれます。
「postalCode」	米国の郵便番号など、ドメインの郵便番号を示します。
「preferredDeliveryMethod」	ユーザーの連絡方法またはメッセージ配信方法を示します。
「registeredAddress」	受信者が配信を確認する必要がある場合に、編集したドキュメントを受け取るのに使用される住所を指定します。
「searchGuide」	エントリーを検索用にディレクトリーツリーのベースオブジェクトとして使用する場合に、提案した検索条件の情報を指定します。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「st (stateOrProvinceName)」	ドメインがある州を指定します。
「street」	ドメインの物理的な場所の番地と住所を示します。
「telephoneNumber」	ドメインの電話番号を示します。

属性	定義
<a href="#">「teletexTerminalIdentifier」</a>	ドメインのテレテックス端末の ID を提供します。
<a href="#">「telexNumber」</a>	ドメインのテレックス番号を示します。
<a href="#">「userPassword」</a>	エントリーがディレクトリーにバインドできるパスワードを保存します。
<a href="#">「x121Address」</a>	ドメインの X.121 アドレスを指定します。

### 5.3.18. domainRelatedObject

**domainRelatedObject** オブジェクトクラスは、組織や組織単位などの X.500 ドメインと同等の DNS ドメインまたは NRS ドメインを表すエントリーを定義します。

このオブジェクトクラスは [RFC 1274](#) に定義されています。

#### 上級クラス

top

#### OID

0.9.2342.19200300.100.4.17

表5.34 必要な属性

属性	定義
<a href="#">「objectClass」</a>	エントリーに割り当てられたオブジェクトクラスを指定します。
<a href="#">「associatedDomain」</a>	ディレクトリーツリー内のオブジェクトに関連付けられている DNS ドメインを指定します。

### 5.3.19. dSA

**dSA** オブジェクトクラスは、DSA を表すエントリーを定義します。

このオブジェクトクラスは [RFC 1274](#) に定義されています。

#### 上級クラス

top

#### OID

2.5.6.13

表5.35 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。
「presentationAddress」	エントリーの OSI プレゼンテーションアドレスが含まれます。

表5.36 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。
「knowledgeInformation」	
「l (localityName)」	エントリーの市または地理的な場所を指定します。
「o (organizationName)」	エントリーが属する組織を指定します。
「ou (organizationalUnitName)」	エントリーが属する組織単位または部門を指定します。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「supportedApplicationContext」	OSI アプリケーションコンテキストの識別子が含まれています。

### 5.3.20. extensibleObject

エントリーに存在する場合には、**extensibleObject** は、エントリーがオプションで任意の属性を保持できるようにします。このクラスで使用できる属性リストは、暗黙的にはサーバーが認識している属性セットすべてです。

このオブジェクトクラスは [RFC 2252](#) で定義されます。

#### 上級クラス

top

#### OID

1.3.6.1.4.1.1466.101.120.111

表5.37 必要な属性

属性	定義
<a href="#">「objectClass」</a>	エントリーに割り当てられたオブジェクトクラスを指定します。

### 使用できる属性

サーバーに認識されているすべての属性。

#### 5.3.21. friendlyCountry

**friendlyCountry** オブジェクトクラスは、ディレクトリー内の国のエントリーを定義します。このオブジェクトクラスでは、**country** オブジェクトクラスよりもわかりやすい名前を使用できます。

このオブジェクトクラスは [RFC 1274](#) に定義されています。

### 上級クラス

top

### OID

0.9.2342.19200300.100.4.18

表5.38 必要な属性

属性	定義
<a href="#">「objectClass」</a>	エントリーに割り当てられたオブジェクトクラスを指定します。
<a href="#">「co (friendlyCountryName)」</a>	人間が読める国の名前を格納します。
<a href="#">「c (countryName)」</a>	ディレクトリー内の ISO で定義されている国名を表す 2 文字のコードが含まれます。

表5.39 使用できる属性

属性	定義
<a href="#">「description」</a>	エントリーのテキスト説明を入力します。
<a href="#">「searchGuide」</a>	エントリーを検索用にディレクトリーツリーのベースオブジェクトとして使用する場合に、提案した検索条件の情報を指定します。

#### 5.3.22. groupOfCertificates

**groupOfCertificates** オブジェクトクラスは、一連の X.509 証明書を記述します。「[memberCertificateDescription](#)」値のいずれかに一致する証明書はすべて、グループのメンバーと見なされます。

## 上級クラス

top

### OID

2.16.840.1.113730.3.2.31

表5.40 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.41 使用できる属性

属性	定義
「businessCategory」	グループの参加先となるビジネス種別を指定します。
「description」	エントリーのテキスト説明を入力します。
「memberCertificateDescription」	特定の証明書がこのグループのメンバーであるかどうかを判別するために使用される値が含まれています。
「o (organizationName)」	エントリーが属する組織を指定します。
「ou (organizationalUnitName)」	エントリーが属する組織単位または部門を指定します。
「owner」	グループの対象者の DN (識別名) が含まれます。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。

### 5.3.23. groupOfMailEnhancedUniqueNames

**groupOfMailEnhancedUniqueNames** オブジェクトクラスは、メールグループに使用されます。このグループのメンバーは一意でなければなりません。このオブジェクトクラスは Netscape Messaging Server に定義されます。

## 上級クラス

top

### OID

2.16.840.1.113730.3.2.5

表5.42 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.43 使用できる属性

属性	定義
「businessCategory」	グループの参加先となるビジネス種別を指定します。
「description」	エントリーのテキスト説明を入力します。
「mailEnhancedUniqueMember」	メールグループのメンバーを識別するための一意の DN 値が含まれます。
「o (organizationName)」	エントリーが属する組織を指定します。
「ou (organizationalUnitName)」	エントリーが属する組織単位または部門を指定します。
「owner」	グループの対象者の DN (識別名) が含まれます。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。

### 5.3.24. groupOfNames

**groupOfNames** オブジェクトクラスには、名前のグループのエントリーが含まれています。このオブジェクトクラスは [RFC 2256](#) に定義されています。



#### 注記

Directory Server でのこのオブジェクトクラスの定義は、標準の定義とは異なります。標準の定義では、「member」は必須属性ですが、Directory Server では使用可能な属性です。したがって、Directory Server では、グループにメンバーがないのを許可します。

#### 上級クラス

top

OID

2.5.6.9

表5.44 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.45 使用できる属性

属性	定義
「businessCategory」	エントリーが従事しているビジネスの種類を示します。
「description」	エントリーのテキスト説明を入力します。
「member」	グループメンバーの DN (識別名) が含まれます。
「o (organizationName)」	エントリーが属する組織を指定します。
「ou (organizationalUnitName)」	エントリーが属する組織単位または部門を指定します。
「owner」	グループの対象者の DN (識別名) が含まれます。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。

### 5.3.25. groupOfUniqueNames

**groupOfUniqueNames** オブジェクトクラスは、一意の名前を含むグループを定義します。



#### 注記

Directory Server でのこのオブジェクトクラスの定義は、標準の定義とは異なります。標準の定義では、「uniqueMember」は必須属性ですが、Directory Server では使用可能な属性です。したがって、Directory Server では、グループにメンバーがないのを許可します。

このオブジェクトクラスは [RFC 2256](#) に定義されています。

#### 上級クラス

top

OID

2.5.6.17

表5.46 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.47 使用できる属性

属性	定義
「businessCategory」	エントリーが従事しているビジネスの種類を示します。
「description」	エントリーのテキスト説明を入力します。
「o (organizationName)」	エントリーが属する組織を指定します。
「ou (organizationalUnitName)」	エントリーが属する組織単位または部門を指定します。
「owner」	グループの対象者の DN (識別名) が含まれます。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「uniqueMember」	グループのメンバーの DN (識別名) が含まれます。この DN は一意である必要があります。

### 5.3.26. groupOfURLs

**groupOfURLs** オブジェクトクラスは、**groupOfUniqueNames** および **groupOfNames** オブジェクトクラスの補助オブジェクトクラスです。このグループは、ラベル付けされた URL のリストで設定されます。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.33

表5.48 必要な属性

属性	定義
----	----



属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.49 使用できる属性

属性	定義
「businessCategory」	グループの参加先となるビジネス種別を指定します。
「description」	エントリーのテキスト説明を入力します。
「memberURL」	グループの各メンバーに関連付けられた URL が含まれます。
「o (organizationName)」	エントリーが属する組織を指定します。
「ou (organizationalUnitName)」	エントリーが属する組織単位または部門を指定します。
「owner」	グループの対象者の DN (識別名) が含まれます。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。

### 5.3.27. ieee802Device

**ieee802Device** オブジェクトクラスは、MAC アドレスのあるデバイスを指します。このオブジェクトクラスは RFC 2307 に定義されています。



#### 注記

このオブジェクトクラスは、Directory Server の **10rfc2307.ldif** で定義されています。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

#### 上級クラス

top

#### OID

1.3.6.1.1.1.2.11

表5.50 必要な属性

属性	定義
<code>objectClass</code>	エントリーのオブジェクトクラスを定義します。
<code>cn (commonName)</code>	デバイスの一般名を指定します。

表5.51 使用できる属性

属性	定義
<code>description</code>	エントリーのテキスト説明を入力します。
<code>l (localityName)</code>	エントリーの市または地理的な場所を指定します。
<code>macAddress</code>	デバイスの MAC アドレスを指定します。
<code>o (organizationName)</code>	デバイスが属する組織を指定します。
<code>ou (organizationalUnitName)</code>	デバイスが属する組織単位または部門を指定します。
<code>owner</code>	デバイスに対応するユーザーの DN (識別名) を指定します。
<code>seeAlso</code>	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
<code>serialNumber</code>	デバイスのシリアル番号が含まれます。

### 5.3.28. inetAdmin

**inetAdmin** オブジェクトクラスは、管理グループまたはユーザーのマーカーです。このオブジェクトクラスは Netscape Delegated Administrator に対して定義されます。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.112

表5.52 必要な属性

属性	定義
<code>objectClass</code>	エントリーに割り当てられたオブジェクトクラスを指定します。

表5.53 使用できる属性

属性	定義
「adminRole」	管理ユーザーが属するロールを識別します。
「memberOf」	管理ユーザーが属するグループ名が含まれます。これは MemberOf プラグインによって動的に管理されます。

### 5.3.29. inetDomain

**inetDomain** オブジェクトクラスは、仮想ドメインノードの補助クラスです。このオブジェクトクラスは Netscape Delegated Administrator に対して定義されます。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.129

表5.54 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。

表5.55 使用できる属性

属性	定義
「inetDomainBaseDN」	DNS ドメインのユーザーサブツリーのベース DN を定義します。
「inetDomainStatus」	ドメインのステータスを示します。ステータスは active、inactive、または deleted のいずれかです。

### 5.3.30. inetOrgPerson

**inetOrgPerson** オブジェクトクラスは、組織のエンタープライズネットワーク内のユーザーを表すエントリーを定義します。このオブジェクトクラスは、**person** オブジェクトクラスから 「cn (commonName)」 属性および 「sn (surname)」 属性を継承します。

このオブジェクトクラスは RFC 2798 で定義されています。

#### 上級クラス

person

## OID

2.16.840.1.113730.3.2.2

表5.56 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。
「sn (surname)」	ユーザーの姓を指定します。

表5.57 使用できる属性

属性	定義
「audio」	サウンドファイルをバイナリー形式で格納します。
「businessCategory」	エントリーが従事しているビジネスの種類を示します。
「carLicense」	ユーザーの自動車登録番号を指定します。
「departmentNumber」	ユーザーが所属する部門を示します。
「description」	エントリーのテキスト説明を入力します。
「destinationIndicator」	エントリーに関連付けられている国と都市を指定します。これはかつて、公共の電報サービスを提供するのに必要でした。
「displayName」	エントリーを表示するときに使用する推奨のユーザー名を表示します。
「employeeNumber」	その人の従業員番号が含まれます。
「employeeType」	ユーザーの雇用の種類を指定します (たとえば、フルタイム)。
「fax (facsimileTelephoneNumber)」	ユーザーの FAX 番号が含まれています。
「givenName」	そのユーザーの名前が含まれます。
「homePhone」	そのユーザーの自宅の電話番号を示します。
「homePostalAddress」	そのユーザーの自宅の郵送先住所を指定します。

属性	定義
「initials」	そのユーザーのイニシャルを指定します。
「internationalISDNNumber」	エントリーの ISDN 番号を指定します。
「jpegPhoto」	JPG イメージを保存します。
「l (localityName)」	エントリーの市または地理的な場所を指定します。
「labeledURI」	エントリーに関連する URL が含まれています。
「mail」	そのユーザーのメールアドレスが含まれます。
「manager」	person エントリーの直接スーパーバイザーの DN (識別名) が含まれます。
「mobile」	そのユーザーの携帯電話番号を指定します。
「o (organizationName)」	エントリーが属する組織を指定します。
「ou (organizationalUnitName)」	エントリーが属する組織単位または部門を指定します。
「pager」	そのユーザーのポケットベル番号を指定します。
「photo」	人物の写真をバイナリー形式で保存します。
「physicalDeliveryOfficeName」	物理的な配送が可能な場所を提供します。
「postOfficeBox」	エントリーの私書箱番号を示します。
「postalAddress」	エントリーのメールアドレスが含まれます。
「postalCode」	米国の郵便番号など、エントリーの郵便番号を示します。
「preferredDeliveryMethod」	ユーザーの連絡方法またはメッセージ配信方法を示します。
「preferredLanguage」	その人が希望する書き言葉または話し言葉を指定します。
「registeredAddress」	受信者が配信を確認する必要がある場合に、編集したドキュメントを受け取るのに使用される住所を指定します。
「roomNumber」	そのユーザーが存在する部屋番号を指定します。

属性	定義
「secretary」	そのユーザーの秘密や管理アシスタントの DN (識別名) が含まれます。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「st (stateOrProvinceName)」	そのユーザーの所在する州を指定します。
「street」	そのユーザーの物理的な場所の住所を示します。
「telephoneNumber」	エントリーの電話番号を指定します。
「teletexTerminalIdentifier」	そのユーザーのテレテックス端末の識別子を提供します。
「telexNumber」	エントリーに関連付けられているテレックス番号を示します。
「title」	そのユーザーの役職を表示します。
「uid (userID)」	そのユーザーのユーザー ID (通常はログイン ID) が含まれます。
「userCertificate」	ユーザーの証明書をクリアテキストで保存します (使用されていません)。
「userPassword」	エントリーがディレクトリーにバインドできるパスワードを保存します。
「userSMIMECertificate」	S/MIME クライアントで使用できるように、個人の証明書をバイナリー形式で保存します。
「x121Address」	その人の X.121 アドレスを提供します。
「x500UniqueIdentifier」	将来の使用のために予約されています。

### 5.3.31. inetSubscriber

**inetSubscriber** オブジェクトクラスは、一般的なユーザーアカウント管理に使用されます。このオブジェクトクラスは Netscape サブスクリバラーの相互運用性に対して定義されます。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.134

表5.58 必要な属性

属性	定義
<code>ObjectClass</code>	エントリーに割り当てられたオブジェクトクラスを指定します。

表5.59 使用できる属性

属性	定義
<code>inetSubscriberAccountId</code>	加入者を課金システムにリンクする一意の属性が含まれています。
<code>inetSubscriberChallenge</code>	ユーザーの身元を確認するために使用される、ある種の質問またはプロンプト、チャレンジフレーズが含まれています。
<code>inetSubscriberResponse</code>	チャレンジフレーズの質問への回答が含まれていません。

### 5.3.32. inetUser

**inetUser** オブジェクトクラスは、加入者サービスを提供するためにエントリーに存在している必要がある補助クラスです。このオブジェクトクラスは Netscape サブスクリバラーの相互運用性に対して定義されます。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.130

表5.60 必要な属性

属性	定義
<code>ObjectClass</code>	エントリーに割り当てられたオブジェクトクラスを指定します。

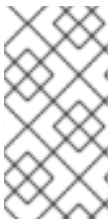
表5.61 使用できる属性

属性	定義
<code>inetUserHttpURL</code>	ユーザーに関連付けられた Web アドレスが含まれます。

属性	定義
<code>[inetUserStatus]</code>	ユーザーのステータスを指定します。ステータスは active、inactive、または deleted のいずれかです。
<code>[memberOf]</code>	ユーザーが属するグループ名が含まれます。これは MemberOf プラグインによって動的に管理されません。
<code>[uid (userID)]</code>	そのユーザーのユーザー ID (通常はログイン ID) が含まれます。
<code>[userPassword]</code>	ユーザーがユーザーアカウントへのアクセスに使用できるパスワードを保存します。

### 5.3.33. ipHost

**ipHost** オブジェクトクラスは、ホストに関する IP 情報を格納します。このオブジェクトクラスは [RFC 2307](#) に定義されています。



#### 注記

このオブジェクトクラスは、Directory Server の **10rfc2307.ldif** で定義されています。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、`/usr/share/dirsrv/data` ディレクトリーの **10rfc2307bis.ldif** ファイルを `/etc/dirsrv/slapd-instance/schema` ディレクトリーにコピーします。

#### 上級クラス

top

#### OID

1.3.6.1.1.2.6

表5.62 必要な属性

属性	定義
<code>[objectClass]</code>	エントリーのオブジェクトクラスを定義します。
<code>[cn (commonName)]</code>	デバイスの一般名を指定します。
<code>[ipHostNumber]</code>	デバイスまたはホストの IP アドレスが含まれます。

表5.63 使用できる属性

属性	定義
<code>[description]</code>	エントリーのテキスト説明を入力します。



属性	定義
「l (localityName)」	エントリーの市または地理的な場所を指定します。
「manager」	エントリーのメンテナーまたはスーパーバイザーの DN (識別名) が含まれます。
「o (organizationName)」	デバイスが属する組織を指定します。
「ou (organizationalUnitName)」	デバイスが属する組織単位または部門を指定します。
「owner」	デバイスに対応するユーザーの DN (識別名) を指定します。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「serialNumber」	デバイスのシリアル番号が含まれます。

### 5.3.34. ipNetwork

**ipNetwork** オブジェクトクラスは、ネットワークに関する IP 情報を格納します。このオブジェクトクラスは [RFC 2307](#) に定義されています。



#### 注記

このオブジェクトクラスは、Directory Server の **10rfc2307.ldif** で定義されています。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

#### 上級クラス

top

#### OID

1.3.6.1.1.1.2.7

表5.64 必要な属性

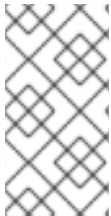
属性	定義
「objectClass」	エントリーのオブジェクトクラスを定義します。
「cn (commonName)」	デバイスの一般名を指定します。
「ipNetworkNumber」	ネットワークの IP 番号が含まれます。

表5.65 使用できる属性

属性	定義
<code>「description」</code>	エントリーのテキスト説明を入力します。
<code>「l (localityName)」</code>	エントリーの市または地理的な場所を指定します。
<code>「manager」</code>	エントリーのメンテナーまたはスーパーバイザーの DN (識別名) が含まれます。
<code>「ipNetmaskNumber」</code>	ネットワークの IP ネットマスクが含まれます。

### 5.3.35. ipProtocol

`ipProtocol` オブジェクトクラスは、IP プロトコルのバージョンを指定します。このオブジェクトクラスは [RFC 2307](#) に定義されています。



#### 注記

このオブジェクトクラスは、Directory Server の `10rfc2307.ldif` で定義されています。更新された RFC 2307 スキーマを使用するには、`10rfc2307.ldif` ファイルを削除し、`/usr/share/dirsrv/data` ディレクトリーの `10rfc2307bis.ldif` ファイルを `/etc/dirsrv/slapd-instance/schema` ディレクトリーにコピーします。

#### 上級クラス

top

#### OID

1.3.6.1.1.1.2.4

表5.66 必要な属性

属性	定義
<code>「objectClass」</code>	エントリーのオブジェクトクラスを定義します。
<code>「cn (commonName)」</code>	デバイスの一般名を指定します。
<code>「ipProtocolNumber」</code>	ネットワークの IP プロトコル番号が含まれます。

表5.67 使用できる属性

属性	定義
<code>「description」</code>	エントリーのテキスト説明を入力します。

### 5.3.36. ipService

**ipService** オブジェクトクラスは、IP サービスに関する情報を格納します。このオブジェクトクラスは [RFC 2307](#) に定義されています。



#### 注記

このオブジェクトクラスは、Directory Server の **10rfc2307.ldif** で定義されています。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slaped-instance/schema** ディレクトリーにコピーします。

#### 上級クラス

top

#### OID

1.3.6.1.1.1.2.3

表5.68 必要な属性

属性	定義
「objectClass」	エントリーのオブジェクトクラスを定義します。
「cn (commonName)」	デバイスの一般名を指定します。
「ipServicePort」	IP サービスで使用されるポート番号を示します。
「ipServiceProtocol」	サービスの IP プロトコル番号が含まれます。

表5.69 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。

### 5.3.37. labeledURIObject

このオブジェクトクラスを既存のディレクトリーオブジェクトに追加して、URI 値を含めることができます。このオブジェクトクラスを使用しても、必要に応じて、「labeledURI」属性タイプを他のオブジェクトクラスに直接含めることができます。

このオブジェクトクラスは [RFC 2079](#) に定義されています。

#### 上級クラス

top

#### OID

1.3.6.1.4.1.250.3.15

表5.70 必要な属性

属性	定義
<code>ObjectClass</code>	エントリーに割り当てられたオブジェクトクラスを指定します。

表5.71 使用できる属性

属性	定義
<code>labeledURI</code>	エントリーのオブジェクトに関連する URI を指定します。

### 5.3.38. locality

**locality** オブジェクトクラスは、地域または地理的領域を表すエントリーを定義します。

このオブジェクトクラスは [RFC 2256](#) に定義されています。

#### 上級クラス

top

#### OID

2.5.6.3

表5.72 必要な属性

属性	定義
<code>ObjectClass</code>	エントリーに割り当てられたオブジェクトクラスを指定します。

表5.73 使用できる属性

属性	定義
<code>description</code>	エントリーのテキスト説明を入力します。
<code>l (localityName)</code>	エントリーの市または地理的な場所を指定します。
<code>searchGuide</code>	エントリーを検索用にディレクトリーツリーのベースオブジェクトとして使用する場合に、提案した検索条件の情報を指定します。
<code>seeAlso</code>	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。

属性	定義
「 <a href="#">st (stateOrProvinceName)</a> 」	地域に関連付けられている州または県を指定します。
「 <a href="#">street</a> 」	地域に関連付けられている通りと番号を示します。

### 5.3.39. mailGroup

**mailGroup** オブジェクトクラスは、グループのメール属性を定義します。このオブジェクトは、Netscape Messaging Server のスキーマで定義されます。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.4

表5.74 必要な属性

属性	定義
「 <a href="#">objectClass</a> 」	エントリーに割り当てられたオブジェクトクラスを指定します。

表5.75 使用できる属性

属性	定義
「 <a href="#">cn (commonName)</a> 」	エントリーの共通名を指定します。
「 <a href="#">mail</a> 」	グループのメールアドレスを保存します。
「 <a href="#">mailAlternateAddress</a> 」	グループのセカンダリーメールアドレスが含まれます。
「 <a href="#">mailHost</a> 」	メールサーバーのホスト名が含まれます。
「 <a href="#">owner</a> 」	グループの対象者の DN (識別名) が含まれます。

### 5.3.40. mailRecipient

**mailRecipient** オブジェクトクラスは、ユーザーのメールアカウントを定義します。このオブジェクトは、Netscape Messaging Server のスキーマで定義されます。

#### 上級クラス

top

**OID**

2.16.840.1.113730.3.2.3

表5.76 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。

表5.77 使用できる属性

属性	定義
「cn (commonName)」	エントリーの共通名を指定します。
「mail」	グループのメールアドレスを保存します。
「mailAccessDomain」	ユーザーがメッセージングサーバーにアクセスできるドメインが含まれます。
「mailAlternateAddress」	グループのセカンダリーメールアドレスが含まれます。
「mailAutoReplyMode」	アカウントの自動リプライモードが有効であるかどうかを指定します。
「mailAutoReplyText」	自動返信メールに使用するテキストが含まれます。
「mailDeliveryOption」	メールユーザーに使用するメール配信メカニズムを指定します。
「mailForwardingAddress」	メールユーザーに使用するメール配信メカニズムを指定します。
「mailHost」	メールサーバーのホスト名が含まれます。
「mailMessageStore」	ユーザーのメールボックスの場所を指定します。
「mailProgramDeliveryInfo」	プログラムしたメール配信に使用されるコマンドを指定します。
「mailQuota」	ユーザーのメールボックスに許可されるディスク容量を指定します。

属性	定義
「mailRoutingAddress」	このエントリーのアカウントから別のメッセージングサーバーにメールを転送するときに使用するルーティングアドレスが含まれています。
「multiLineDescription」	複数行にまたがるエントリーのテキスト説明が含まれています。
「uid (userID)」	定義されたアカウントのユーザー ID を指定します。
「userPassword」	エントリーがアカウントにアクセスするために使用するパスワードを保存します。

### 5.3.41. mepManagedEntry

**mepManagedEntry** オブジェクトクラスは、マネージドエントリープラグインのインスタンスによって生成されたエントリーを識別します。このオブジェクトクラスは Directory Server に定義されています。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.319

表5.78 使用できる属性

属性	定義
「mepManagedBy」	マネージドエントリーに対応する元のエントリーの DN を指定します。

### 5.3.42. mepOriginEntry

**mepOriginEntry** オブジェクトクラスは、マネージドエントリープラグインのインスタンスによって監視されるサブツリー内にあり、さらにプラグインによって作成されたマネージドエントリーを含むエントリーを識別します。これが元のエントリーになります。このオブジェクトクラスは Directory Server に定義されています。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.320

表5.79 使用できる属性

属性	定義
「 <a href="#">mepManagedEntry</a> 」	マネージドエントリープラグインインスタンスによって作成され、この元のエントリーに対応するマネージドエントリーの DN を指定します。

### 5.3.43. mepTemplateEntry

**mepTemplateEntry** オブジェクトクラスは、マネージドエントリーを作成するためにマネージドエントリープラグインのインスタンスによってテンプレートとして使用されるエントリーを識別します。このオブジェクトクラスは Directory Server に定義されています。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.321

表5.80 使用できる属性

属性	定義
「 <a href="#">cn (commonName)</a> 」	エントリーの共通名を指定します。
「 <a href="#">mepMappedAttr</a> 」	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。
「 <a href="#">mepRDNAAttr</a> 」	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。
「 <a href="#">mepStaticAttr</a> 」	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。

### 5.3.44. netscapeCertificateServer

**netscapeCertificateServer** オブジェクトクラスは、Netscape 証明書サーバーに関する情報を格納します。このオブジェクトは、Netscape 証明書管理システムのスキーマで定義されています。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.18

表5.81 必要な属性



属性	定義
<a href="#">「objectClass」</a>	エントリーに割り当てられたオブジェクトクラスを指定します。

### 5.3.45. netscapeDirectoryServer

**netscapeDirectoryServer** オブジェクトクラスは、Directory Server インスタンスに関する情報を格納します。このオブジェクトは Netscape Directory Server のスキーマで定義されます。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.23

表5.82 必要な属性

属性	定義
<a href="#">「objectClass」</a>	エントリーに割り当てられたオブジェクトクラスを指定します。

### 5.3.46. NetscapeLinkedOrganization

**NetscapeLinkedOrganization** は、補助オブジェクトクラスです。このオブジェクトは、Netscape サーバースイートのスキーマで定義されています。

#### 上級クラス

top

#### OID

1.3.6.1.4.1.1466.101.120.141

表5.83 使用できる属性

属性	定義
<a href="#">「parentOrganization」</a>	サーバースイート用に定義されたリンクされた組織の親組織を識別します。

### 5.3.47. netscapeMachineData

**netscapeMachineData** オブジェクトクラスは、マシンデータとマシン以外のデータを区別します。このオブジェクトは Netscape Directory Server のスキーマで定義されます。

#### 上級クラス

top

## OID

2.16.840.1.113730.3.2.32

### 5.3.48. NetscapePreferences

**NetscapePreferences** は、ユーザー設定を格納する補助オブジェクトクラスです。このオブジェクトは Netscape によって定義されています。

#### 上級クラス

top

## OID

1.3.6.1.4.1.1466.101.120.142

表5.84 必要な属性

属性	定義
<code>preferredLanguage</code>	その人が希望する書き言葉または話し言葉を指定します。
<code>preferredLocale</code>	ユーザーの希望のロケールを指定します。ロケール設定は、日付形式や通貨などの文化または国の設定を定義します。
<code>preferredTimeZone</code>	その人の好みのタイムゾーンを示します。

### 5.3.49. netscapeReversiblePasswordObject

**netscapeReversiblePasswordObject** は、パスワードを格納するための補助オブジェクトクラスです。このオブジェクトは、NetscapeWeb サーバーのスキーマで定義されています。

#### 上級クラス

top

## OID

2.16.840.1.113730.3.2.154

表5.85 使用できる属性

属性	定義
<code>netscapeReversiblePassword</code>	HTTP ダイジェスト/MD5 認証に使用されるパスワードが含まれています。

### 5.3.50. netscapeServer

**netscapeServer** オブジェクトクラスには、Netscape サーバーとそのインストールに関するインスタンス固有の情報が含まれています。

## 上級クラス

top

### OID

2.16.840.1.113730.3.2.10

表5.86 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.87 使用できる属性

属性	定義
「administratorContactInfo」	サーバー管理者の連絡先情報が含まれています。
「adminUrl」	インスタンスが使用する管理サーバーの URL が含まれます。
「description」	エントリーのテキスト説明を入力します。
「installationTimeStamp」	サーバーインスタンスがインストールされた時間が含まれます。
「serverHostName」	Directory Server インスタンスが実行しているサーバーのホスト名が含まれます。
「serverProductName」	サーバータイプの製品名が含まれます。
「serverRoot」	サーバー製品がインストールされている最上位ディレクトリーを指定します。
「serverVersionNumber」	製品のバージョン番号が含まれています。
「userPassword」	エントリーがディレクトリーにバインドできるパスワードを保存します。

### 5.3.51. netscapeWebServer

**netscapeWebServer** オブジェクトクラスは、インストールされている Netscape Web サーバーを識別します。

## 上級クラス

top

**OID**

2.16.840.1.113730.3.2.29

表5.88 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。
「nsServerID」	サーバーの名前または ID が含まれます。

表5.89 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。
「nsServerPort」	サーバーのポート番号が含まれます。

**5.3.52. newPilotPerson**

**newPilotPerson** オブジェクトクラスは、**person** のサブクラスであり、**person** オブジェクトクラスのエントリーに追加の属性を割り当てることができます。このオブジェクトクラスは、**person** オブジェクトクラスから 「cn (commonName)」 属性および 「sn (surname)」 属性を継承します。

このオブジェクトクラスは、Internet White Pages Pilot で定義されています。

**上級クラス**

person

**OID**

0.9.2342.19200300.100.4.4

表5.90 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。
「sn (surname)」	ユーザーの姓を指定します。

表5.91 使用できる属性

属性	定義
「businessCategory」	エントリーが従事しているビジネスの種類を示します。
「description」	エントリーのテキスト説明を入力します。
「drink (favouriteDrink)」	ユーザーの好きな飲み物を指定します。
「homePhone」	そのユーザーの自宅の電話番号を示します。
「homePostalAddress」	そのユーザーの自宅の郵送先住所を指定します。
「janetMailbox」	ユーザーのメールアドレスを指定します。これは主に、英国または RFC822 メールアドレスを使用しない組織で使用するためのものです。
「mail」	そのユーザーのメールアドレスが含まれます。
「mailPreferenceOption」	メーリングリスト (電子的または物理的) に自分の名前を追加するかどうか、ユーザーの希望を指定します。
「mobile」	そのユーザーの携帯電話番号を指定します。
「organizationalStatus」	ユーザーの職務に共通する職種を示します。
「otherMailbox」	X.400 および RFC822 以外の電子メールボックスタイプの値が含まれています。
「pager」	そのユーザーのポケットベル番号を指定します。
「personalSignature」	個人の署名ファイルが含まれています。
「personalTitle」	ユーザーの敬称を指定します。
「preferredDeliveryMethod」	ユーザーの連絡方法またはメッセージ配信方法を示します。
「roomNumber」	そのユーザーが存在する部屋番号を指定します。
「secretary」	そのユーザーの秘密や管理アシスタントの DN (識別名) が含まれます。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。

属性	定義
「telephoneNumber」	エントリーの電話番号を指定します。
「uid (userID)」	そのユーザーのユーザー ID (通常はログイン ID) が含まれます。
「userClass」	このエントリーのコンピューターユーザーのタイプを説明します。
「userPassword」	エントリーがディレクトリーにバインドできるパスワードを保存します。

### 5.3.53. nisMap

このオブジェクトクラスは NIS マップを指します。

このオブジェクトクラスは [RFC 2307](#) で定義され、LDAP をネットワーク情報サービスとして使用するオブジェクトクラスおよび属性を定義します。



#### 注記

このオブジェクトクラスは、Directory Server の **10rfc2307.ldif** で定義されています。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

#### 上級クラス

top

#### OID

1.3.6.1.1.1.2.13

表5.92 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「nisMapName」	NIS マップ名が含まれます。

表5.93 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。

### 5.3.54. nisNetgroup

このオブジェクトクラスには、NIS ドメイン内で使用されるネットグループが含まれています。このオブジェクトクラスを追加すると、管理者はネットグループを使用して NIS でのログインとサービス認証を制御できます。

このオブジェクトクラスは RFC 2307 で定義され、LDAP をネットワーク情報サービスとして使用するオブジェクトクラスおよび属性を定義します。



#### 注記

このオブジェクトクラスは、Directory Server の **10rfc2307.ldif** で定義されています。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

#### 上級クラス

top

#### OID

1.3.6.1.1.1.2.8

表5.94 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.95 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。
「memberNisNetgroup」	マージ netgroup の名前をリスト表示することで、別の netgroup の属性値を現在の値にマージします。
「nisNetgroupTriple」	ユーザー名 ( <b>,bobby,example.com</b> ) またはマシン名 ( <b>,shellserver1,,example.com</b> ) が含まれます。

### 5.3.55. nisObject

このオブジェクトクラスには、NIS ドメイン内のオブジェクトに関する情報が含まれています。

このオブジェクトクラスは RFC 2307 で定義され、LDAP をネットワーク情報サービスとして使用するオブジェクトクラスおよび属性を定義します。



## 注記

このオブジェクトクラスは、Directory Server の **10rfc2307.ldif** で定義されています。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

## 上級クラス

top

## OID

1.3.6.1.1.1.2.10

表5.96 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。
「NisMapEntry」	NIS マップエントリーを識別します。
「nisMapName」	NIS マップの名前が含まれています。

表5.97 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。

## 5.3.56. nsAdminConfig

このオブジェクトクラスは管理サーバーの設定パラメーターを保存します。このオブジェクトは管理サービスに対して定義されます。

## 上級クラス

nsConfig

## OID

nsAdminConfig-oid

表5.98 使用できる属性

属性	定義
「nsAdminAccessAddresses」	管理サーバーの IP アドレスを識別します。



属性	定義
「nsAdminAccessHosts」	管理サーバーのホスト名またはホスト名のリストが含まれます。
「nsAdminCacheLifetime」	キャッシュタイムアウト期間の長さに注意してください。
「nsAdminCgiWaitPid」	サーバーが待機している CGI プロセスの PID が含まれます。
「nsAdminEnableEnduser」	管理サーバーの Web サービスページへのエンドユーザーアクセスを許可するか禁止するかを設定します。
「nsAdminOneACLDir」	管理サーバーのローカル ACL ディレクトリーのパスが含まれます。
「nsAdminUsers」	管理者ユーザー情報を含むファイルを指します。

### 5.3.57. nsAdminConsoleUser

このオブジェクトクラスは管理サーバーの設定パラメーターを保存します。このオブジェクトは管理サービスに対して定義されます。

#### 上級クラス

top

#### OID

nsAdminConsoleUser-oid

表5.99 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.100 使用できる属性

属性	定義
「nsPreference」	コンソール設定の設定情報を格納します。

### 5.3.58. nsAdminDomain

このオブジェクトクラスは、管理コンソールにアクセスするためのユーザー情報を格納します。このオブジェクトは管理サービスに対して定義されます。

### 上級クラス

organizationalUnit

### OID

nsAdminDomain-oid

表5.101 使用できる属性

属性	定義
<code>「nsAdminDomainName」</code>	サーバーの管理ドメインを識別します。

### 5.3.59. nsAdminGlobalParameters

このオブジェクトクラスは管理サーバーの設定パラメーターを保存します。このオブジェクトは管理サービスに対して定義されます。

### 上級クラス

top

### OID

nsAdminGlobalParameters-oid

表5.102 必要な属性

属性	定義
<code>「objectClass」</code>	エントリーに割り当てられたオブジェクトクラスを指定します。
<code>「cn (commonName)」</code>	エントリーの共通名を指定します。

表5.103 使用できる属性

属性	定義
<code>「nsAdminEndUserHTMLIndex」</code>	HTML インデックスページへのエンドユーザーアクセスを許可するか禁止するかを設定します。
<code>「nsNickName」</code>	アプリケーションのニックネームを指定します。

### 5.3.60. nsAdminGroup

このオブジェクトクラスは、管理者ユーザーのグループ情報を管理サーバーに格納します。このオブジェクトは管理サービスに対して定義されます。

## 上級クラス

top

## OID

nsAdminGroup-oid

表5.104 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.105 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。
「nsAdminGroupName」	管理者グループの名前が含まれます。
「nsAdminSIEDN」	管理サーバーインスタンスのサーバーインスタンスエントリー (SIE) の DN を表示します。
「nsConfigRoot」	管理サーバーインスタンスの設定ディレクトリーへの完全パスを指定します。

### 5.3.61. nsAdminObject

このオブジェクトクラスには、タスクなど、管理サーバーによって使用されるオブジェクトに関する情報が含まれています。このオブジェクトは管理サービスに対して定義されます。

## 上級クラス

top

## OID

nsAdminObject-oid

表5.106 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.107 使用できる属性

属性	定義
「nsClassname」	管理サーバーのタスクまたはリソースエディターに関連付けられているクラス名が含まれます。
「nsJarfilename」	管理サーバーコンソールがオブジェクトへのアクセスに使用する JAR ファイルの名前を指定します。

### 5.3.62. nsAdminResourceEditorExtension

このオブジェクトクラスには、コンソールリソースエディターで使用される拡張機能が含まれています。このオブジェクトは管理サービスに対して定義されます。

#### 上級クラス

nsAdminObject

#### OID

nsAdminResourceEditorExtension-oid

表5.108 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.109 使用できる属性

属性	定義
「nsAdminAccountInfo」	管理サーバーアカウントに関する情報が含まれています。
「nsDeleteclassname」	削除するクラスの名前が含まれます。

### 5.3.63. nsAdminServer

このオブジェクトクラスは、管理サーバーインスタンスを定義します。このオブジェクトは管理サービスに対して定義されます。

#### 上級クラス

top

#### OID

nsAdminServer-oid

表5.110 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。
「nsServerID」	<b>slapd-example</b> などの Directory Server ID が含まれます。

表5.111 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。

### 5.3.64. nsAIMpresence

**nsAIMpresence** は、AOL インスタンスメッセージングアカウントのステータスを定義する補助オブジェクトクラスです。このオブジェクトは Directory Server に対して定義されます。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.300

表5.112 使用できる属性

属性	定義
「nsAIMid」	エントリーの AIM ユーザー ID が含まれています。
「nsAIMStatusGraphic」	AIM アカウントのステータスを示すグラフィックイメージへのポインターが含まれています。
「nsAIMStatusText」	AIM アカウントのステータスを示すテキストが含まれています。

### 5.3.65. nsApplication

**nsApplication** は、アプリケーションまたはサーバーのエントリーを定義します。これは Netscape によって定義されています。

#### 上級クラス

top

**OID**

nsApplication-oid

表5.113 必要な属性

属性	定義
「objectClass」	エントリーのオブジェクトクラスを定義します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.114 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。
「installationTimeStamp」	サーバーインスタンスがインストールされた時間が含まれます。
「nsBuildNumber」	サーバーインスタンスのビルド番号が含まれます。
「nsBuildSecurity」	ビルドに使用されるセキュリティのレベルが含まれます。
「nsExpirationDate」	アプリケーションのライセンスの有効期限が切れる日付が含まれます。
「nsInstalledLocation」	バージョン 7.1 以前のサーバーの場合に、サーバーのインストールディレクトリーを表示します。
「nsLdapSchemaVersion」	Directory Server が使用する LDAP スキーマファイルのバージョンを示します。
「nsNickName」	アプリケーションのニックネームを指定します。
「nsProductName」	サーバー製品の名前を示します。
「nsProductVersion」	サーバー製品のバージョン番号を表示します。
「nsRevisionNumber」	製品のリビジョン番号 (マイナーバージョン) が含まれています。
「nsSerialNumber」	サーバー製品に割り当てられたシリアル番号を指定します。

属性	定義
「nsServerMigrationClassname」	サーバーインスタンスの移行に使用するクラスを指定します。
「nsServerCreationClassname」	サーバーインスタンスの作成に使用するクラスを指定します。
「nsVendor」	サーバーを設計したベンダーの名前が含まれます。

### 5.3.66. nsCertificateServer

**nsCertificateServer** オブジェクトクラスは、Red Hat Certificate System インスタンスに関する情報を格納します。このオブジェクトは、証明書システムのスキーマで定義されています。

#### 上級クラス

top

#### OID

nsCertificateServer-oid

表5.115 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「nsServerID」	サーバーの名前または ID が含まれます。

表5.116 使用できる属性

属性	定義
「nsCertConfig」	Red Hat Certificate System インスタンスの設定が含まれています。
「nsServerPort」	サーバーのポート番号が含まれます。
「serverHostName」	Directory Server インスタンスが実行しているサーバーのホスト名が含まれます。

### 5.3.67. nsComplexRoleDefinition

定義では、単純なロール以外のロールは、複雑なロールとなります。

このオブジェクトクラスは Directory Server で定義されます。

## 上級クラス

nsRoleDefinition

### OID

2.16.840.1.113730.3.2.95

表5.117 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。

表5.118 使用できる属性

属性	定義
「cn (commonName)」	エントリーの共通名を指定します。
「description」	エントリーのテキスト説明を入力します。

## 5.3.68. nsContainer

一部のエントリーは特定のエンティティを定義しませんが、類似または関連する子エントリーの親エントリーとしてディレクトリツリー内に定義されたスペースを作成します。これらは **コンテナエントリー** であり、**nsContainer** オブジェクトクラスによって識別されます。

## 上級クラス

top

### OID

2.16.840.1.113730.3.2.104

表5.119 必要な属性

属性	定義
objectClass	エントリーのオブジェクトクラスを定義します。
cn	エントリーの共通名を指定します。

## 5.3.69. nsCustomView

**nsCustomView** オブジェクトクラスは、Directory Server コンソールでの Directory Server データのカスタムビューに関する情報を定義します。これは管理サービスに対して定義されます。

## 上級クラス

nsAdminObject



**OID**

nsCustomView-oid

表5.120 使用できる属性

属性	定義
「nsDisplayName」	カスタムビュー設定プロファイルの名前が含まれます。

**5.3.70. nsDefaultObjectClasses**

**nsDefaultObjectClasses** は、ディレクトリー内に特定のタイプのオブジェクトの新規作成時に使用するデフォルトのオブジェクトクラスを設定します。これは管理サービスに対して定義されます。

**上級クラス**

top

**OID**

nsDefaultObjectClasses-oid

表5.121 必要な属性

属性	定義
「objectClass」	エントリーのオブジェクトクラスを定義します。
「cn (commonName)」	デバイスの一般名を指定します。

表5.122 使用できる属性

属性	定義
「nsDefaultObjectClass」	デフォルトでオブジェクトタイプに割り当てるオブジェクトクラスが含まれます。

**5.3.71. nsDirectoryInfo**

**nsDirectoryInfo** には、ディレクトリーインスタンスに関する情報が含まれています。これは管理サービスに対して定義されます。

**上級クラス**

top

**OID**

nsDirectoryInfo-oid

表5.123 必要な属性

属性	定義
「objectClass」	エントリーのオブジェクトクラスを定義します。
「cn (commonName)」	デバイスの一般名を指定します。

表5.124 使用できる属性

属性	定義
「nsBindDN」	サーバーインスタンスエントリーのサーバーに対して定義されたバインド DN が含まれます。
「nsBindPassword」	SIE にバインドアイデンティティのパスワードが含まれます。
「nsDirectoryFailoverList」	<b>nsDirectoryURL</b> のインスタンスが使用できない場合にフェイルオーバーサポートに使用する他の Directory Server インスタンスの URL のリストが含まれています。
「nsDirectoryInfoRef」	ディレクトリー内の識別名 (DN) への参照が含まれています。
「nsDirectoryURL」	Directory Server インスタンスにアクセスするための URL が含まれています。

### 5.3.72. nsDirectoryServer

**nsDirectoryServer** は、Directory Server インスタンスの定義オブジェクトクラスです。これは、Directory Server に対して定義されています。

#### 上級クラス

top

#### OID

nsDirectoryServer-oid

表5.125 必要な属性

属性	定義
「objectClass」	エントリーのオブジェクトクラスを定義します。
「nsServerID」	サーバーの名前または ID が含まれます。

表5.126 使用できる属性

属性	定義
「nsBaseDN」	サーバーインスタンスのベース DN が含まれます。
「nsBindDN」	サーバーインスタンスエントリーのサーバーに対して定義されたバインド DN が含まれます。
「nsBindPassword」	SIE にバインドアイデンティティのパスワードが含まれます。
「nsSecureServerPort」	サーバーの TLS ポート番号が含まれます。
「nsServerPort」	サーバーのポート番号が含まれます。
「serverHostName」	Directory Server インスタンスが実行しているサーバーのホスト名が含まれます。

### 5.3.73. nsFilteredRoleDefinition

**nsFilteredRoleDefinition** オブジェクトクラスは、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る方法を定義します。

このオブジェクトクラスは Directory Server に定義されています。

#### 上級クラス

nsComplexRoleDefinition

#### OID

2.16.840.1.113730.3.2.97

表5.127 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「nsRoleFilter」	フィルターされたロールのエントリーを識別するために使用されるフィルターを指定します。

表5.128 使用できる属性

属性	定義
「cn (commonName)」	エントリーの共通名を指定します。
「description」	エントリーのテキスト説明を入力します。

### 5.3.74. nsGlobalParameters

**nsGlobalParameters** オブジェクトクラスには、グローバルプリファレンス設定が含まれています。

このオブジェクトクラスは Administrative Services で定義されます。

#### 上級クラス

top

#### OID

nsGlobalParameters-oid

表5.129 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.130 使用できる属性

属性	定義
「nsGroupRDNComponent」	グループエントリーの RDN で使用されるデフォルトの属性タイプを定義します。
「nsUniqueAttribute」	設定で一意的な属性を定義します。
「nsUserIDFormat」	<b>givenname</b> 属性と <b>sn</b> 属性からユーザー ID を生成するための形式を設定します。
「nsUserRDNComponent」	ユーザー DN のネーミングコンポーネントとして使用する属性タイプを設定します。
nsNYR	使用されていません。
nsWellKnownJarfiles	使用されていません。

### 5.3.75. nsHost

**nsHost** オブジェクトクラスは、サーバーホストに関する情報を格納します。

このオブジェクトクラスは Administrative Services で定義されます。

#### 上級クラス

top

## OID

nsHost-oid

表5.131 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.132 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。
「l (localityName)」	エントリーの市または地理的な場所を指定します。
「nsHardwarePlatform」	Directory Server インスタンスが実行されているホストのハードウェアプラットフォームを識別します。これは、 <b>uname -m</b> を実行するのと同じ情報です。
「nsHostLocation」	サーバーホストの場所を示します。
「nsOsVersion」	サーバーホストのオペレーティングシステムバージョンが含まれます。
「serverHostName」	Directory Server インスタンスが実行しているサーバーのホスト名が含まれます。

## 5.3.76. nsICQpresence

**nsICQpresence** は、ICQ メッセージングアカウントのステータスを定義する補助オブジェクトクラスです。このオブジェクトは Directory Server に対して定義されます。

## 上級クラス

top

## OID

2.16.840.1.113730.3.2.301

表5.133 使用できる属性

属性	定義
「nsICQid」	エントリーの ICQ ユーザー ID が含まれています。

属性	定義
<a href="#">「nsICQStatusGraphic」</a>	ICQ アカウントのステータスを示すグラフィックイメージへのポインターが含まれています。
<a href="#">「nsICQStatusText」</a>	ICQ アカウントのステータスを示すテキストが含まれています。

### 5.3.77. nsLicenseUser

**nsLicenseUser** オブジェクトクラスは、クライアントごとにライセンスが付与されているサーバーのライセンスを追跡します。**nsLicenseUser** は、**inetOrgPerson** オブジェクトクラスで使用することを目的としています。このオブジェクトクラスの内容は、管理サーバーの **ユーザー** と **グループ** 領域から管理できます。

このオブジェクトクラスは、管理サーバスキーマで定義されています。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.7

表5.134 必要な属性

属性	定義
<a href="#">「objectClass」</a>	エントリーに割り当てられたオブジェクトクラスを指定します。

表5.135 使用できる属性

属性	定義
<a href="#">「nsLicensedFor」</a>	ユーザーが使用を許可されているサーバーを識別します。
<a href="#">「nsLicenseEndTime」</a>	将来の使用のために予約されています。
<a href="#">「nsLicenseStartTime」</a>	将来の使用のために予約されています。

### 5.3.78. nsManagedRoleDefinition

**nsManagedRoleDefinition** オブジェクトクラスは、明示的に列挙されたメンバーリストへのロールのメンバー割り当てを指定します。

このオブジェクトクラスは Directory Server に定義されています。

#### 上級クラス

nsComplexRoleDefinition

**OID**

2.16.840.1.113730.3.2.96

表5.136 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。

表5.137 使用できる属性

属性	定義
「cn (commonName)」	エントリーの共通名を指定します。
「description」	エントリーのテキスト説明を入力します。

**5.3.79. nsMessagingServerUser**

**nsICQpresence** は、メッセージングサーバーユーザーを記述する補助オブジェクトクラスです。このオブジェクトクラスは Netscape Messaging Server に定義されます。

**上級クラス**

top

**OID**

2.16.840.113730.3.2.37

表5.138 必要な属性

属性	定義
「objectClass」	エントリーのオブジェクトクラスを指定します。

表5.139 使用できる属性

属性	定義
「cn (commonName)」	エントリーの共通名を指定します。
「mailAccessDomain」	ユーザーがメッセージングサーバーにアクセスできるドメインが含まれます。
「mailAlternateAddress」	グループのセカンダリーメールアドレスが含まれません。

属性	定義
「mailAutoReplyMode」	アカウントの自動リプライモードが有効であるかどうかを指定します。
「mailAutoReplyText」	自動返信メールに使用するテキストが含まれます。
「mailDeliveryOption」	メールユーザーに使用するメール配信メカニズムを指定します。
「mailForwardingAddress」	メールユーザーに使用するメール配信メカニズムを指定します。
「mailMessageStore」	ユーザーのメールボックスの場所を指定します。
「mailProgramDeliveryInfo」	プログラムしたメール配信に使用されるコマンドを指定します。
「mailQuota」	ユーザーのメールボックスに許可されるディスク容量を指定します。
「nsmgDisallowAccess」	ユーザーが使用できるメールプロトコルに制限を設定します。
「nsmgNumMsgQuota」	ユーザーのメールボックスに許可されるメッセージの数を指定します。
「nswmExtendedUserPrefs」	ユーザーの拡張設定を保存します。
「vacationEndDate」	休暇期間の終了日が含まれます。
「vacationStartDate」	休暇期間の開始日が含まれます。

### 5.3.80. nsMSNpresence

**nsMSNpresence** は、MSN インスタンスメッセージングアカウントのステータスを定義する補助オブジェクトクラスです。このオブジェクトは Directory Server に対して定義されます。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.303

#### 表5.140 使用できる属性



属性	定義
「nsMSNid」	エントリーの MSN ユーザー ID が含まれます。

### 5.3.81. nsNestedRoleDefinition

**nsNestedRoleDefinition** オブジェクトクラスは、任意のタイプの1つ以上のロールが、ロール内のメンバーとして含まれることを指定します。

このオブジェクトクラスは Directory Server に定義されています。

#### 上級クラス

nsComplexRoleDefinition

#### OID

2.16.840.1.113730.3.2.98

表5.141 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「nsRoleDn」	エントリーに割り当てられるロールを指定します。

表5.142 使用できる属性

属性	定義
「cn (commonName)」	エントリーの共通名を指定します。
「description」	エントリーのテキスト説明を入力します。

### 5.3.82. nsResourceRef

**nsNestedRoleDefinition** オブジェクトクラスは、リソース参照を設定します。

このオブジェクトクラスは、管理サービスで定義されています。

#### 上級クラス

top

#### OID

nsResourceRef-oid

表5.143 必要な属性

属性	定義
<a href="#">「objectClass」</a>	エントリーに割り当てられたオブジェクトクラスを指定します。
<a href="#">「cn (commonName)」</a>	エントリーの共通名を指定します。

表5.144 使用できる属性

属性	定義
<a href="#">「seeAlso」</a>	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。

### 5.3.83. nsRoleDefinition

すべてのロール定義オブジェクトクラスは、**nsRoleDefinition** オブジェクトクラスから継承されます。

このオブジェクトクラスは Directory Server で定義されます。

#### 上級クラス

LDAPsubentry

#### OID

2.16.840.1.113730.3.2.93

表5.145 必要な属性

属性	定義
<a href="#">「objectClass」</a>	エントリーに割り当てられたオブジェクトクラスを指定します。

表5.146 使用できる属性

属性	定義
<a href="#">「cn (commonName)」</a>	エントリーの共通名を指定します。
<a href="#">「description」</a>	エントリーのテキスト説明を入力します。

### 5.3.84. nsSimpleRoleDefinition

このオブジェクトクラスを含むロールは、柔軟性が意図的に制限されているため、単純なロールと呼ばれます。これにより、次のことが容易になります。

- ロールのメンバーを列挙する。

- 特定のエントリーに特定のロールが割り当てられているどうかを判断する。
- 特定のエントリーが持つすべてのロールを列挙する。
- 特定のエントリーに特定のロールを割り当てる。
- 特定のエントリーから特定のロールを削除する

このオブジェクトクラスは Directory Server で定義されます。

## 上級クラス

nsRoleDefinition

## OID

2.16.840.1.113730.3.2.94

表5.147 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。

表5.148 使用できる属性

属性	定義
「cn (commonName)」	エントリーの共通名を指定します。
「description」	エントリーのテキスト説明を入力します。

### 5.3.85. nsSNMP

このオブジェクトクラスは、Directory Server が使用する SNMP プラグインオブジェクトの設定を定義します。

このオブジェクトクラスは Directory Server に定義されています。

## 上級クラス

top

## OID

2.16.840.1.113730.3.2.41

表5.149 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。

属性	定義
「cn (commonName)」	エントリーの共通名を指定します。
「nsSNMPEnabled」	Directory Server インスタンスに対して SNMP を有効にするかどうかを設定します。

表5.150 使用できる属性

属性	定義
「nsSNMPContact」	SNMP エージェントにより渡される連絡先情報が含まれています。
「nsSNMPDescription」	SNMP 設定のテキスト説明が含まれています。
「nsSNMPLocation」	SNMP エージェントのロケーション情報または設定が含まれています。
「nsSNMPMasterHost」	SNMP マスターエージェントが配置されているサーバーのホスト名が含まれます。
「nsSNMPMasterPort」	SNMP サブエージェントにアクセスするためのポートが含まれています。
「nsSNMPOrganization」	SNMP サービスで渡される組織名または情報が含まれます。

### 5.3.86. nsTask

このオブジェクトクラスは、Directory Server によって実行されるタスクの設定を定義します。

このオブジェクトクラスは Administrative Services に対して定義されます。

#### 上級クラス

top

#### OID

nsTask-oid

表5.151 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.152 使用できる属性

属性	定義
「nsExecRef」	タスクを実行するプログラムへの参照が含まれます。
「nsHelpRef」	タスクウィンドウに関連付けられたオンライン (HTML) ヘルプファイルへの参照が含まれます。
「nsLogSuppress」	タスクのロギングを抑制するかどうかを設定します。
「nsTaskLabel」	コンソールのタスクに関連付けられたラベルが含まれます。

### 5.3.87. nsTaskGroup

このオブジェクトクラスは、コンソール内のタスクのグループの情報を定義します。

このオブジェクトクラスは Administrative Services に対して定義されます。

#### 上級クラス

top

#### OID

nsTaskGroup-oid

表5.153 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.154 使用できる属性

属性	定義
「nsTaskLabel」	コンソールのタスクに関連付けられたラベルが含まれます。

### 5.3.88. nsTopologyCustomView

このオブジェクトクラスは、コンソールのプロファイルに使用されるトポロジービューを設定します。

このオブジェクトクラスは Administrative Services に対して定義されます。

## 上級クラス

nsCustomView

## OID

nsTopologyCustomView-oid

表5.155 必要な属性

属性	定義
<code>[cn (commonName)]</code>	エントリーの共通名を指定します。

表5.156 使用できる属性

属性	定義
<code>[nsViewConfiguration]</code>	コンソールで使用するビュー設定が含まれていません。

### 5.3.89. nsTopologyPlugin

このオブジェクトクラスは、コンソールでのビュー設定に使用されるトポロジープラグインを設定します。

このオブジェクトクラスは Administrative Services に対して定義されます。

## 上級クラス

nsAdminObject

## OID

nsTopologyPlugin-oid

### 5.3.90. nsValueItem

このオブジェクトクラスは、値アイテムのオブジェクト設定を定義します。これは、エントリーの値タイプに依存する情報を指定するために使用されます。値項目は、バイナリー文字列や、大文字と小文字を区別する文字列など、エントリー属性で使用可能な属性値の構文に関連しています。

このオブジェクトクラスは、Netscape Servers-Value Item で定義されています。

## 上級クラス

top

## OID

2.16.840.1.113730.3.2.45

表5.157 必要な属性

属性	定義
「ObjectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.158 使用できる属性

属性	定義
「nsValueBin」	バイナリー値タイプに関連する情報または操作が含まれています。
「nsValueCES」	大文字と小文字を区別する文字列 (CES) の値の型に関連する情報または操作が含まれます。
「nsValueCIS」	大文字と小文字を区別しない (CIS) 値の型に関連する情報または操作が含まれます。
「nsValueDefault」	属性または設定パラメーターに使用するデフォルト値のタイプを設定します。
「nsValueDescription」	値項目設定のテキスト説明を指定します。
「nsValueDN」	DN 値タイプに関連する情報または操作が含まれています。
「nsValueFlags」	値アイテムオブジェクトのフラグを設定します。
「nsValueHelpURL」	値アイテムオブジェクトに関連付けられたオンライン (HTML) ヘルプファイルへの参照が含まれています。
「nsValueInt」	整数値型に関連する情報または操作が含まれています。
「nsValueSyntax」	値アイテムオブジェクトに使用する構文を定義します。
「nsValueTel」	電話の文字列値のタイプに関連する情報または操作が含まれています。
「nsValueType」	適用する値のタイプを設定します。

### 5.3.91. nsView

このオブジェクトクラスは、ディレクトリーツリーのビューエントリーに使用されます。

このオブジェクトクラスは Directory Server に定義されています。

## 上級クラス

top

## OID

2.16.840.1.113730.3.2.304

表5.159 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.160 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。
「nsViewFilter」	ビュープラグインで使用されるフィルターを識別します。

### 5.3.92. nsYIMpresence

**nsYIMpresence** は、Yahoo インスタンスメッセージングアカウントのステータスを定義する補助オブジェクトクラスです。このオブジェクトは Directory Server に対して定義されます。

## 上級クラス

top

## OID

2.16.840.1.113730.3.2.302

表5.161 使用できる属性

属性	定義
「nsYIMid」	エントリーの Yahoo ユーザー ID が含まれます。
「nsYIMStatusGraphic」	Yahoo アカウントのステータスを示すグラフィックイメージへのポインターが含まれています。
「nsYIMStatusText」	Yahoo アカウントのステータスを示すテキストが含まれています。



### 5.3.93. ntGroup

**ntGroup** オブジェクトクラスは、Active Directory サーバーに格納されているグループエントリーのデータを保持します。いくつかの Directory Server 属性は、Windows グループ属性に直接対応するか、一致するようにマップされます。Windows Server グループと同期する新しいグループを Directory Server に作成すると、Directory Server 属性が Windows エントリーに割り当てられます。これらの属性は、いずれかのディレクトリーサービスでエントリーに追加、変更、または削除できます。

このオブジェクトクラスは Netscape NT Synchronization で定義されます。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.9

表5.162 必要なオブジェクトクラス

オブジェクトクラス	定義
「mailGroup」	メール属性を Windows グループと Directory Server グループ間で同期できるようにします。

表5.163 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「ntUserDomainId」	グループアカウントの Windows ドメインログイン ID が含まれます。

表5.164 使用できる属性

属性	定義
「cn (commonName)」	エントリーの一般的な名前を指定します。これは Windows の <b>name</b> フィールドに対応します。
「description」	エントリーのテキストの説明を指定します。Windows の <b>comment</b> フィールドに対応します。
「l (localityName)」	エントリーの市または地理的な場所を指定します。
「member」	グループのメンバーを指定します。

属性	定義
<code>「ntGroupCreateNewGroup」</code>	Directory Server でエントリーを作成するときに Windows アカウントを作成するかどうかを指定します。
<code>「ntGroupDeleteGroup」</code>	Directory Server でエントリーが削除されたときに Windows アカウントを削除するかどうかを指定します。
<code>「ntGroupDomainId」</code>	グループのドメイン ID 文字列を提供します。
<code>「ntGroupType」</code>	エントリーがどの種類の Windows ドメイングループであるかを定義します。
<code>「ntUniqueId」</code>	サーバーが操作と識別に使用する、生成された ID 番号が含まれます。
<code>「ou (organizationalUnitName)」</code>	エントリーが属する組織単位または部門を指定します。
<code>「seeAlso」</code>	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。

### 5.3.94. ntUser

**ntUser** エントリーは、Active Directory サーバーに格納されているユーザーエントリーのデータを保持します。いくつかの Directory Server 属性は、Windows ユーザーアカウントフィールドに直接対応するか、一致するようにマップされます。Windows Server と同期する新しい個人エントリーを Directory Server に作成すると、Directory Server の属性が Windows ユーザーアカウントフィールドに割り当てられます。これらの属性は、いずれかのディレクトリーサービスでエントリーに追加、変更、または削除できます。

このオブジェクトクラスは Netscape NT Synchronization で定義されます。

#### 上級クラス

top

#### OID

2.16.840.1.113730.3.2.8

表5.165 必要な属性

属性	定義
<code>「objectClass」</code>	エントリーに割り当てられたオブジェクトクラスを指定します。

属性	定義
「cn (commonName)」	エントリーの一般的な名前を指定します。これは Windows の <b>name</b> フィールドに対応します。
「ntUserDomainId」	ユーザーアカウントの Windows ドメインログイン ID が含まれます。

表5.166 使用できる属性

属性	定義
「description」	エントリーのテキストの説明を指定します。Windows の <b>comment</b> フィールドに対応します。
「destinationIndicator」	エントリーに関連付けられている国と都市を指定します。これはかつて、公共の電報サービスを提供するのに必要でした。
「fax (facsimileTelephoneNumber)」	ユーザーの FAX 番号を指定します。
「givenName」	そのユーザーの名前が含まれます。
「homePhone」	そのユーザーの自宅の電話番号を示します。
「homePostalAddress」	そのユーザーの自宅の郵送先住所を指定します。
「initials」	そのユーザーのイニシャルを指定します。
「l (localityName)」	エントリーの市または地理的な場所を指定します。
「mail」	そのユーザーのメールアドレスが含まれます。
「manager」	person エントリーの直接スーパーバイザーの DN (識別名) が含まれます。
「mobile」	そのユーザーの携帯電話番号を指定します。
「ntUserAcctExpires」	ユーザーの Windows アカウントの有効期限を識別します。
「ntUserCodePage」	ユーザーのコードページに移動します。
「ntUserCreateNewAccount」	このエントリーを Directory Server で作成するときに、Windows アカウントを作成するかどうかを指定します。

属性	定義
<code>[ntUserDeleteAccount]</code>	Directory Server でこのエントリーを削除するときに、Windows アカウントを削除するかどうかを指定します。
<code>[ntUserHomeDir]</code>	ユーザーのホームディレクトリーへのパスを指定します。
<code>[ntUserLastLogoff]</code>	Windows Server からユーザーが最後にログオフした時間を示します。
<code>[ntUserLastLogon]</code>	ユーザーが最後に Windows Server にログオンした時刻を示します。
<code>[ntUserMaxStorage]</code>	Windows Server でユーザーが使用できる最大ディスク容量を表示します。
<code>[ntUserParms]</code>	アプリケーションで使用するために予約されている Unicode 文字列が含まれています。
<code>[ntUserProfile]</code>	ユーザーの Windows プロファイルへのパスが含まれます。
<code>[ntUserScriptPath]</code>	ユーザーの Windows ログインスクリプトへのパスが含まれます。
<code>[ntUserWorkstations]</code>	ユーザーが Windows ドメインにログインできる Windows ワークステーションのリストが含まれます。
<code>[o (organizationName)]</code>	エントリーが属する組織を指定します。
<code>[ou (organizationalUnitName)]</code>	エントリーが属する組織単位または部門を指定します。
<code>[pager]</code>	そのユーザーのポケットベル番号を指定します。
<code>[postalAddress]</code>	エントリーのメールアドレスが含まれます。
<code>[postalCode]</code>	米国の郵便番号など、エントリーの郵便番号を示します。
<code>[postOfficeBox]</code>	エントリーの私書箱番号を示します。
<code>[registeredAddress]</code>	受信者が配信を確認する必要がある場合に、編集したドキュメントを受け取るのに使用される住所を指定します。

属性	定義
「 <a href="#">seeAlso</a> 」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「 <a href="#">sn (surname)</a> 」	ユーザーの姓を指定します。
「 <a href="#">st (stateOrProvinceName)</a> 」	ユーザーのいる州を指定します。
「 <a href="#">street</a> 」	ユーザーの物理的な場所の番地と住所を示します。
「 <a href="#">telephoneNumber</a> 」	エントリーの電話番号を指定します。
「 <a href="#">teletexTerminalIdentifier</a> 」	そのユーザーのテレテックス端末の識別子を提供します。
「 <a href="#">telexNumber</a> 」	エントリーに関連付けられているテレックス番号を示します。
「 <a href="#">title</a> 」	そのユーザーの役職を表示します。
「 <a href="#">userCertificate</a> 」	ユーザーの証明書をクリアテキストで保存します (使用されていません)。
「 <a href="#">x121Address</a> 」	エントリーの X.121 アドレスを指定します。

### 5.3.95. oncRpc

**oncRpc** オブジェクトクラスは、Open Network Computing Remote Procedure Call (ONC RPC) の抽象化を定義します。このオブジェクトクラスは [RFC 2307](#) に定義されています。



#### 注記

このオブジェクトクラスは、Directory Server の **10rfc2307.ldif** で定義されています。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

#### 上級クラス

top

#### OID

1.3.6.1.1.2.5

#### 表5.167 必要な属性

属性	定義
「objectClass」	エントリーのオブジェクトクラスを定義します。
「cn (commonName)」	エントリーの共通名を指定します。
「oncRpcNumber」	RPC マップの一部を含み、UNIXRPC の RPC 番号を格納します。

表5.168 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。

### 5.3.96. 組織

**organization** 属性は、組織を表すエントリーを定義します。組織は通常、大企業またはエンタープライズ内の大規模で比較的静的なグループであると見なされます。

このオブジェクトクラスは [RFC 2256](#) に定義されています。

#### 上級クラス

top

#### OID

2.5.6.4

表5.169 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「o (organizationName)」	エントリーが属する組織を指定します。

表5.170 使用できる属性

属性	定義
「businessCategory」	エントリーが従事しているビジネスの種類を示します。
「description」	エントリーのテキスト説明を入力します。

属性	定義
「destinationIndicator」	エントリーに関連付けられている国と都市を指定します。これはかつて、公共の電報サービスを提供するのに必要でした。
「fax (facsimileTelephoneNumber)」	エントリーの fax 番号を指定します。
「internationalISDNNumber」	エントリーの ISDN 番号を指定します。
「l (localityName)」	エントリーの市または地理的な場所を指定します。
「physicalDeliveryOfficeName」	物理的な配送が可能な場所を提供します。
「postalAddress」	エントリーのメールアドレスが含まれます。
「postalCode」	米国の郵便番号など、エントリーの郵便番号を示します。
「postOfficeBox」	エントリーの私書箱番号を示します。
「preferredDeliveryMethod」	エントリーの希望の連絡方法およびメッセージ配信方法を示します。
「registeredAddress」	受信者が配信を確認する必要がある場合に、編集したドキュメントを受け取るのに使用される住所を指定します。
「searchGuide」	エントリーを検索用にディレクトリーツリーのベースオブジェクトとして使用する場合に、提案した検索条件の情報を指定します。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「st (stateOrProvinceName)」	ユーザーのいる州を指定します。
「street」	そのユーザーの物理的な場所の住所を示します。
「telephoneNumber」	組織の責任者の電話番号を示します。
「teletexTerminalIdentifier」	エントリーのテレテックス端末の ID を指定します。
「telexNumber」	エントリーに関連付けられているテレックス番号を示します。

属性	定義
「userPassword」	エントリーがディレクトリーにバインドできるパスワードを保存します。
「x121Address」	エントリーの X.121 アドレスを指定します。

### 5.3.97. organizationalPerson

**organizationalPerson** オブジェクトクラスは、組織に雇用されている、または所属しているユーザーのエントリーを定義します。このオブジェクトクラスは、**person** オブジェクトクラスから「cn (commonName)」属性および「sn (surname)」属性を継承します。

このオブジェクトクラスは [RFC 2256](#) に定義されています。

#### 上級クラス

person

#### OID

2.5.6.7

表5.171 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。
「sn (surname)」	ユーザーの姓を指定します。

表5.172 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。
「destinationIndicator」	エントリーに関連付けられている国と都市を指定します。これはかつて、公共の電報サービスを提供するのに必要でした。
「fax (facsimileTelephoneNumber)」	エントリーの fax 番号を指定します。
「internationalISDNNumber」	エントリーの ISDN 番号を指定します。
「l (localityName)」	エントリーの市または地理的な場所を指定します。



属性	定義
「ou (organizationalUnitName)」	エントリーが属する組織単位または部門を指定します。
「physicalDeliveryOfficeName」	物理的な配送が可能な場所を提供します。
「postalAddress」	エントリーのメールアドレスが含まれます。
「postalCode」	米国の郵便番号など、エントリーの郵便番号を示します。
「postOfficeBox」	エントリーの私書箱番号を示します。
「preferredDeliveryMethod」	ユーザーの連絡方法またはメッセージ配信方法を示します。
「registeredAddress」	受信者が配信を確認する必要がある場合に、編集したドキュメントを受け取るのに使用される住所を指定します。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「st (stateOrProvinceName)」	ユーザーのいる州を指定します。
「street」	そのユーザーの物理的な場所の住所を示します。
「telephoneNumber」	エントリーの電話番号を指定します。
「teletexTerminalIdentifier」	エントリーのテレテックス端末の ID を指定します。
「telexNumber」	エントリーに関連付けられているテレックス番号を示します。
「title」	そのユーザーの役職を表示します。
「userPassword」	エントリーがディレクトリーにバインドできるパスワードを保存します。
「x121Address」	エントリーの X.121 アドレスを指定します。

### 5.3.98. organizationalRole

**organizationalRole** オブジェクトクラスは、組織内のユーザーが保持するロールのエントリーの定義に使用されます。

このオブジェクトクラスは [RFC 2256](#) に定義されています。

## 上級クラス

top

## OID

2.5.6.8

表5.173 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.174 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。
「destinationIndicator」	エントリーに関連付けられている国と都市を指定します。これはかつて、公共の電報サービスを提供するのに必要でした。
「fax (facsimileTelephoneNumber)」	エントリーの fax 番号を指定します。
「internationalISDNNumber」	エントリーの ISDN 番号を指定します。
「l (localityName)」	エントリーの市または地理的な場所を指定します。
「ou (organizationalUnitName)」	エントリーが属する組織単位または部門を指定します。
「physicalDeliveryOfficeName」	物理的な配送が可能な場所を提供します。
「postalAddress」	エントリーのメールアドレスが含まれます。
「postalCode」	米国の郵便番号など、エントリーの郵便番号を示します。
「postOfficeBox」	エントリーの私書箱番号を示します。
「preferredDeliveryMethod」	ロールの連絡方法またはメッセージ配信方法を示します。

属性	定義
「registeredAddress」	受信者が配信を確認する必要がある場合に、編集したドキュメントを受け取るのに使用される住所を指定します。
「roleOccupant」	ロール内のユーザーの DN (識別名) が含まれます。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「st (stateOrProvinceName)」	そのユーザーの所在する州を指定します。
「street」	そのロールの物理的な場所の住所を示します。
「telephoneNumber」	エントリーの電話番号を指定します。
「teletexTerminalIdentifier」	エントリーのテレテックス端末の ID を指定します。
「telexNumber」	エントリーに関連付けられているテレックス番号を示します。
「x121Address」	エントリーの X.121 アドレスを指定します。

### 5.3.99. organizationalUnit

**organizationalUnit** オブジェクトクラスは、**組織の部門**を表すエントリーを定義します。このクラスは、一般的に、大規模な組織内の比較的静的なグループ分けであると認識されています。

このオブジェクトクラスは [RFC 2256](#) に定義されています。

#### 上級クラス

top

#### OID

2.5.6.5

表5.175 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「ou (organizationalUnitName)」	エントリーが属する組織単位または部門を指定します。

表5.176 使用できる属性

属性	定義
「businessCategory」	エントリーが従事しているビジネスの種類を示します。
「description」	エントリーのテキスト説明を入力します。
「destinationIndicator」	エントリーに関連付けられている国と都市を指定します。これはかつて、公共の電報サービスを提供するのに必要でした。
「fax (facsimileTelephoneNumber)」	エントリーの fax 番号を指定します。
「internationalISDNNumber」	エントリーの ISDN 番号を指定します。
「l (localityName)」	エントリーの市または地理的な場所を指定します。
「physicalDeliveryOfficeName」	物理的な配送が可能な場所を提供します。
「postalAddress」	エントリーのメールアドレスが含まれます。
「postalCode」	米国の郵便番号など、エントリーの郵便番号を示します。
「postOfficeBox」	エントリーの私書箱番号を示します。
「preferredDeliveryMethod」	希望する連絡方法を示します。
「registeredAddress」	受信者が配信を確認する必要がある場合に、編集したドキュメントを受け取るのに使用される住所を指定します。
「searchGuide」	エントリーを検索用にディレクトリーツリーのベースオブジェクトとして使用する場合に、提案した検索条件の情報を指定します。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「st (stateOrProvinceName)」	ユーザーのいる州を指定します。
「street」	そのロールの物理的な場所の住所を示します。
「telephoneNumber」	エントリーの電話番号を指定します。
「teletexTerminalIdentifier」	エントリーのテレテックス端末の ID を指定します。

属性	定義
「telexNumber」	エントリーに関連付けられているテレックス番号を示します。
「userPassword」	エントリーがディレクトリーにバインドできるパスワードを保存します。
「x121Address」	エントリーの X.121 アドレスを指定します。

### 5.3.100. person

**person** オブジェクトクラスは、一般的なユーザーのエントリーを表します。これは、**organizationalPerson** オブジェクトクラスの基本オブジェクトクラスです。

このオブジェクトクラスは [RFC 2256](#) に定義されています。

#### 上級クラス

top

#### OID

2.5.6.6

表5.177 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。
「sn (surname)」	ユーザーの姓を指定します。

表5.178 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「telephoneNumber」	エントリーの電話番号を指定します。
「userPassword」	エントリーがディレクトリーにバインドできるパスワードを保存します。

### 5.3.101. pilotObject

**pilotObject** は、他のすべてのオブジェクトクラスのエントリーに追加の属性を割り当てることができるようにするサブクラスです。

このオブジェクトクラスは [RFC 1274](#) に定義されています。

#### 上級クラス

top

#### OID

0.9.2342.19200300.100.4.3

表5.179 必要な属性

属性	定義
<code>「objectClass」</code>	エントリーに割り当てられたオブジェクトクラスを指定します。

表5.180 使用できる属性

属性	定義
<code>「audio」</code>	サウンドファイルをバイナリー形式で保存します。
<code>「dITRedirect」</code>	エントリーのリダイレクトとして使用するエントリーの DN (識別名) が含まれます。
<code>「info」</code>	エントリーに関する情報が含まれています。
<code>「jpegPhoto」</code>	JPG イメージを保存します。
<code>「lastModifiedBy」</code>	ドキュメントエントリーを修正した最終ユーザーの DN (識別名) を指定します。
<code>「lastModifiedTime」</code>	オブジェクトが最後に変更された時刻を指定します。
<code>「manager」</code>	エントリーマネージャーの DN (識別名) を指定します。
<code>「photo」</code>	ドキュメントの写真をバイナリー形式で保存します。
<code>「uniqueIdentifier」</code>	識別名を再利用する場合は、2つのエントリーを区別します。

### 5.3.102. pilotOrganization

**pilotOrganization** オブジェクトクラスは、**organization** および **organizationalUnit** オブジェクトクラスエントリーに属性を追加するために使用されるサブクラスです。

このオブジェクトクラスは [RFC 1274](#) に定義されています。

## 上級クラス

top

## OID

0.9.2342.19200300.100.4.20

表5.181 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「o (organizationName)」	エントリーが属する組織を指定します。
「ou (organizationalUnitName)」	エントリーが属する組織単位または部門を指定します。

表5.182 使用できる属性

属性	定義
「buildingName」	対象のエントリーが配置されている建物の名前を指定します。
「businessCategory」	エントリーが従事しているビジネスの種類を示します。
「description」	エントリーのテキスト説明を入力します。
「destinationIndicator」	エントリーに関連付けられている国と都市を指定します。これはかつて、公共の電報サービスを提供するのに必要でした。
「fax (facsimileTelephoneNumber)」	エントリーの fax 番号を指定します。
「internationalISDNNumber」	エントリーの ISDN 番号を指定します。
「l (localityName)」	エントリーの市または地理的な場所を指定します。
「physicalDeliveryOfficeName」	物理的な配送が可能な場所を提供します。
「postalAddress」	エントリーのメールアドレスが含まれます。

属性	定義
「postalCode」	米国の郵便番号など、エントリーの郵便番号を示します。
「postOfficeBox」	エントリーの私書箱番号を示します。
「preferredDeliveryMethod」	希望する連絡方法を示します。
「registeredAddress」	受信者が配信を確認する必要がある場合に、編集したドキュメントを受け取るのに使用される住所を指定します。
「searchGuide」	エントリーを検索用にディレクトリーツリーのベースオブジェクトとして使用する場合に、提案した検索条件の情報を指定します。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「st (stateOrProvinceName)」	ユーザーのいる州を指定します。
「street」	ユーザーの物理的な場所の番地と住所を示します。
「telephoneNumber」	エントリーの電話番号を指定します。
「teletexTerminalIdentifier」	エントリーのテレテックス端末の ID を指定します。
「telexNumber」	エントリーに関連付けられているテレックス番号を示します。
「userPassword」	エントリーがディレクトリーにバインドできるパスワードを保存します。
「x121Address」	エントリーの X.121 アドレスを指定します。

### 5.3.103. pkiCA

**pkiCA** 補助オブジェクトクラスには、認証局用に設定された必須または使用可能な証明書が含まれています。このオブジェクトクラスは [RFC 4523](#) に定義されています。これは、X.509 証明書および関連する証明書サービスの管理に使用する LDAP のオブジェクトクラスおよび属性を定義します。

#### 上級クラス

top

OID

2.5.6.22



表5.183 使用できる属性

属性	定義
<code>[authorityRevocationList]</code>	取り消された CA 証明書のリストが含まれています。
<code>[cACertificate]</code>	CA 証明書が含まれています。
<code>[certificateRevocationList]</code>	取り消された証明書のリストが含まれています。
<code>[crossCertificatePair]</code>	FBCA スタイルのブリッジ CA 設定で CA のペアを相互認証するために使用される証明書のペアが含まれています。

### 5.3.104. pkiUser

**pkiUser** 補助オブジェクトクラスには、公開鍵インフラストラクチャーの認証局または要素に接続するユーザーまたはクライアントに必要な証明書が含まれます。このオブジェクトクラスは [RFC 4523](#) に定義されています。これは、X.509 証明書および関連する証明書サービスの管理に使用する LDAP のオブジェクトクラスおよび属性を定義します。

#### 上級クラス

top

#### OID

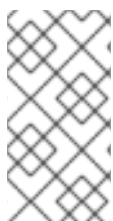
2.5.6.21

表5.184 使用できる属性

属性	定義
<code>[userCertificate]</code>	ユーザーの証明書 (通常はバイナリー形式) を保存します。

### 5.3.105. posixAccount

**posixAccount** オブジェクトクラスは、POSIX 属性を使用するネットワークアカウントを定義します。このオブジェクトクラスは [RFC 2307](#) で定義され、LDAP をネットワーク情報サービスとして使用するオブジェクトクラスおよび属性を定義します。



#### 注記

このオブジェクトクラスは、Directory Server の **10rfc2307.ldif** で定義されています。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、`/usr/share/dirsrv/data` ディレクトリーの **10rfc2307bis.ldif** ファイルを `/etc/dirsrv/slapd-instance/schema` ディレクトリーにコピーします。

#### 上級クラス

top

**OID**

1.3.6.1.1.1.2.0

表5.185 必要な属性

属性	定義
「cn (commonName)」	エントリーの共通名を指定します。
「gidNumber」	Unix のグループ番号に類似した、グループエントリーまたはユーザーエントリーのグループを識別するための一意の数値識別子が含まれます。
「homeDirectory」	ユーザーのホームディレクトリーへのパスを含めます。
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「uid (userID)」	定義されたアカウントのユーザー ID を指定します。
「uidNumber」	Unix のユーザー番号に類似した、ユーザーエントリーの一意の数値識別子が含まれます。

表5.186 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。
「gecos」	ユーザーの GECOS フィールドを決定するために使用されます。これはコモンネーム (CN) に基づいており、追加の情報が埋め込まれています。
「loginShell」	ユーザーがドメインにログインすると自動的に起動するスクリプトへのパスが含まれます。
「userPassword」	エントリーがディレクトリーにバインドできるパスワードを保存します。

**5.3.106. posixGroup**

**posixGroup** オブジェクトクラスは、POSIX 属性を使用するネットワークアカウントのグループを定義します。このオブジェクトクラスは [RFC 2307](#) で定義され、LDAP をネットワーク情報サービスとして使用するオブジェクトクラスおよび属性を定義します。

**上級クラス**

top

**OID**

1.3.6.1.1.1.2.2

表5.187 必要な属性

属性	定義
「gidNumber」	ユーザーがドメインにログインすると自動的に起動するスクリプトへのパスが含まれます。
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。

表5.188 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。
「memberUid」	グループメンバーのログイン名を指定します。これは、メンバーのDNと同じではない可能性があります。
「userPassword」	グループのメンバーのログイン名が含まれます。

**5.3.107. referral**

**referral** オブジェクトクラスは、LDAPv3 スマート参照をサポートするオブジェクトを定義します。このオブジェクトクラスは、LDAPv3 参照インターネットドラフトで定義されています。

**上級クラス**

top

**OID**

2.16.840.1.113730.3.2.6

表5.189 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。

表5.190 使用できる属性

属性	定義
<a href="#">「ref」</a>	LDAPv3 スマート参照の情報が含まれています。

### 5.3.108. residentialPerson

**residentialPerson** オブジェクトクラスでは、個人の居住情報を管理します。

このオブジェクトクラスは [RFC 2256](#) に定義されています。

#### 上級クラス

top

#### OID

2.5.6.10

表5.191 必要な属性

属性	定義
<a href="#">「objectClass」</a>	エントリーに割り当てられたオブジェクトクラスを指定します。
<a href="#">「cn (commonName)」</a>	エントリーの共通名を指定します。
<a href="#">「l (localityName)」</a>	エントリーの市または地理的な場所を指定します。
<a href="#">「sn (surname)」</a>	ユーザーの姓を指定します。

表5.192 使用できる属性

属性	定義
<a href="#">「businessCategory」</a>	エントリーが従事しているビジネスの種類を示します。
<a href="#">「description」</a>	エントリーのテキスト説明を入力します。
<a href="#">「destinationIndicator」</a>	エントリーに関連付けられている国と都市を指定します。これはかつて、公共の電報サービスを提供するのに必要でした。
<a href="#">「fax (facsimileTelephoneNumber)」</a>	エントリーの fax 番号を指定します。
<a href="#">「internationalISDNNumber」</a>	エントリーの ISDN 番号を指定します。
<a href="#">「physicalDeliveryOfficeName」</a>	物理的な配送が可能な場所を提供します。

属性	定義
「postalAddress」	エントリーのメールアドレスが含まれます。
「postalCode」	米国の郵便番号など、エントリーの郵便番号を示します。
「postOfficeBox」	エントリーの私書箱番号を示します。
「preferredDeliveryMethod」	ユーザーの連絡方法またはメッセージ配信方法を示します。
「registeredAddress」	受信者が配信を確認する必要がある場合に、編集したドキュメントを受け取るのに使用される住所を指定します。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「st (stateOrProvinceName)」	ユーザーのいる州を指定します。
「street」	ユーザーの物理的な場所の番地と住所を示します。
「telephoneNumber」	エントリーの電話番号を指定します。
「teletexTerminalIdentifier」	エントリーのテレテックス端末の ID を指定します。
「telexNumber」	エントリーに関連付けられているテレックス番号を示します。
「userPassword」	エントリーがディレクトリーにバインドできるパスワードを保存します。
「x121Address」	エントリーの X.121 アドレスを指定します。

### 5.3.109. RFC822LocalPart

**RFC822LocalPart** オブジェクトクラスは、RFC822 メールアドレスのローカル部分を表すエントリーを定義します。ディレクトリーは、RFC822 アドレスのこの部分をドメインとして扱います。

このオブジェクトクラスは、インターネットディレクトリーパイロットによって定義されます。

#### 上級クラス

domain

#### OID

0.9.2342.19200300.100.4.14

表5.193 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「dc (domainComponent)」	ドメイン名の1つのコンポーネントが含まれます。

表5.194 使用できる属性

属性	定義
「associatedName」	DNS ドメインに関連付けられた組織ディレクトリーツリー内のエントリー名を指定します。
「businessCategory」	エントリーが従事しているビジネスの種類を示します。
「cn (commonName)」	エントリーの共通名を指定します。
「description」	エントリーのテキスト説明を入力します。
「destinationIndicator」	エントリーに関連付けられている国と都市を指定します。これはかつて、公共の電報サービスを提供するのに必要でした。
「fax (facsimileTelephoneNumber)」	エントリーの fax 番号を指定します。
「internationalISDNNumber」	エントリーの ISDN 番号を指定します。
「l (localityName)」	エントリーの市または地理的な場所を指定します。
「o (organizationName)」	アカウントが属する組織を指定します。
「physicalDeliveryOfficeName」	物理的な配送が可能な場所を提供します。
「postalAddress」	エントリーのメールアドレスが含まれます。
「postalCode」	米国の郵便番号など、エントリーの郵便番号を示します。
「postOfficeBox」	エントリーの私書箱番号を示します。
「preferredDeliveryMethod」	ユーザーの連絡方法またはメッセージ配信方法を示します。

属性	定義
「registeredAddress」	受信者が配信を確認する必要がある場合に、編集したドキュメントを受け取るのに使用される住所を指定します。
「searchGuide」	エントリーを検索用にディレクトリーツリーのベースオブジェクトとして使用する場合に、提案した検索条件の情報を指定します。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「sn (surname)」	ユーザーの姓を指定します。
「st (stateOrProvinceName)」	ユーザーのいる州を指定します。
「street」	ユーザーの物理的な場所の番地と住所を示します。
「telephoneNumber」	エントリーの電話番号を指定します。
「teletexTerminalIdentifier」	そのユーザーのテレテックス端末の識別子を提供します。
「telexNumber」	エントリーに関連付けられているテレテックス番号を示します。
「userPassword」	エントリーがディレクトリーにバインドできるパスワードを保存します。
「x121Address」	エントリーの X.121 アドレスを指定します。

### 5.3.110. room

**room** オブジェクトクラスは、部屋に関する情報をディレクトリーに格納します。

#### 上級クラス

top

#### OID

0.9.2342.19200300.100.4.7

#### 表5.195 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「cn (commonName)」	エントリーの共通名を指定します。

表5.196 使用できる属性

属性	定義
「description」	ルームのテキスト説明を入力します。
「roomNumber」	部屋の番号が含まれます。
「seeAlso」	関連情報が含まれる別のエントリーまたはサイトへの URL が含まれます。
「telephoneNumber」	エントリーの電話番号を指定します。

### 5.3.111. shadowAccount

**shadowAccount** オブジェクトクラスを使用すると、LDAP ディレクトリーをシャドウパスワードサービスとして使用できます。シャドウパスワードサービスは、アクセスが厳密に制限されたシャドウファイルに、ホスト上のパスワードファイルを再配置します。

このオブジェクトクラスは [RFC 2307](#) で定義され、LDAP をネットワーク情報サービスとして使用するオブジェクトクラスおよび属性を定義します。



#### 注記

このオブジェクトクラスは、Directory Server の **10rfc2307.ldif** で定義されています。更新された RFC 2307 スキーマを使用するには、**10rfc2307.ldif** ファイルを削除し、**/usr/share/dirsrv/data** ディレクトリーの **10rfc2307bis.ldif** ファイルを **/etc/dirsrv/slapd-instance/schema** ディレクトリーにコピーします。

#### 上級クラス

top

#### OID

1.3.6.1.1.2.1

表5.197 必要な属性

属性	定義
----	----



属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「uid (userID)」	定義されたアカウントのユーザー ID を指定します。

表5.198 使用できる属性

属性	定義
「description」	エントリーのテキスト説明を入力します。
「shadowExpire」	シャドウアカウントの有効期限が切れる日付が含まれます。
「shadowFlag」	シャドウマップのどの領域にフラグ値が格納されているかを識別します。
「shadowInactive」	シャドウアカウントを非アクティブにできる期間を設定します。
「shadowLastChange」	シャドウアカウントへの最後の変更の日時が含まれます。
「shadowMax」	シャドウパスワードの最大有効日数を設定します。
「shadowMin」	シャドウパスワードを変更するまでに最低でも経過する必要がある日数を設定します。
「shadowWarning」	パスワードの有効期限が切れる何日前にユーザーに警告を送信するかを設定します。
「userPassword」	エントリーがディレクトリーにバインドできるパスワードを保存します。

### 5.3.112. simpleSecurityObject

**simpleSecurityObject** オブジェクトクラスでは、エントリーのプリンシパルオブジェクトクラスでパスワード属性が許可されていない場合に、エントリーに **userPassword** 属性を含めることができます。将来の使用のために予約されています。

このオブジェクトクラスは [RFC 1274](#) に定義されています。

#### 上級クラス

top

#### OID

0.9.2342.19200300.100.4.19

表5.199 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「userPassword」	エントリーがディレクトリーにバインドできるパスワードを保存します。

### 5.3.113. strongAuthenticationUser

**strongAuthenticationUser** オブジェクトクラスは、ユーザーの証明書をディレクトリーに格納します。

このオブジェクトクラスは [RFC 2256](#) に定義されています。

#### 上級クラス

top

#### OID

2.5.6.15

表5.200 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。
「userCertificate」	ユーザーの証明書 (通常はバイナリー形式) を保存します。

## 第6章 運用属性とオブジェクトクラス

操作属性は、ディレクトリー操作を実行するために使用される属性で、エントリーのオブジェクトクラスに定義されているかどうかに関係なく、ディレクトリー内のすべてのエントリーで使用できます。操作属性は、特に要求された場合にのみ **ldapsearch** 操作で返されます。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。

操作属性は、エントリーの作成または変更時刻や作成者の名前など、Directory Server がエントリーに対して作成して管理します。これらの属性は、エントリーの他の属性やオブジェクトクラスに関係なく、任意のエントリーに設定できます。

### 6.1. ACCOUNTUNLOCKTIME

**accountUnlockTime** 属性には、アカウントのロックが解除される日付と時刻が GMT 形式で含まれています。値 **0** は、管理者がアカウントのロックを解除する必要があることを意味します。

OID	2.16.840.1.113730.3.1.95
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

### 6.2. ACI

この属性は、クライアントから LDAP 要求を受信したときに、どの権限が許可または拒否されるかの評価に Directory Server が使用します。

OID	2.16.840.1.113730.3.1.55
構文	IA5String
多値または単一値	複数値
定義される場所	Directory Server

### 6.3. ALTSERVER

この属性値は、対象のサーバーが使用できなくなったときに接続する可能性のある他のサーバーの URL です。このサーバーが、使用可能な他のサーバーを認識していない場合には、この属性は空です。この情報は、優先 LDAP サーバーが後で使用できなくなった場合に備えてキャッシュできます。

OID	1.3.6.1.4.1.1466.101.120.6
構文	IA5String

多値または単一値	複数値
定義される場所	<a href="#">RFC 2252</a>

## 6.4. CREATETIMESTAMP

この属性には、エントリーが最初に作成された日時が含まれます。

OID	2.5.18.1
構文	GeneralizedTime
多値または単一値	単一値
定義される場所	<a href="#">RFC 1274</a>

## 6.5. CREATORSNAME

この属性には、エントリーを作成したユーザーの名前が含まれています。

OID	2.5.18.3
構文	DN
多値または単一値	単一値
定義される場所	<a href="#">RFC 1274</a>

## 6.6. DITCONTENTRULES

この属性は、サブスキーマ内で有効な DIT コンテンツルールを定義します。各値は、1つの DIT コンテンツルールを定義します。各値は、関連する構造オブジェクトクラスのオブジェクト識別子によってタグ付けされます。

OID	2.5.21.2
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2252</a>

## 6.7. DITSTRUCTURERULES

この属性は、サブスキーマ内で有効な DIT 構造ルールを定義します。各値は、1つの DIT 構造規則を定義します。

OID	2.5.21.1
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2252</a>

## 6.8. ENTRYUSN

USN プラグインが有効になっている場合に、サーバーは、書き込み操作 (追加、変更、modrdn、または削除) が実行されるたびに、**更新シーケンス番号** をエントリーに自動的に割り当てます。USN は、エントリーの **entryUSN** 操作属性に格納され、次に **entryUSN** は、任意のエントリーの最新の変更の番号を表示します。



### 注記

**entryUSN** 属性は、LDAP クライアントによって実行される操作があった場合にのみ増分します。内部操作はカウントされません。

デフォルトでは、**entryUSN** はバックエンドデータベースインスタンスごとに一意であるため、他のデータベースのエントリーは同じ USN を使用している必要があります。**nsslapd-entryusn-global** パラメーターは、USN の割り当てをローカルからグローバルに変更します。つまり、単一のデータベースでカウントされていたものが、トポロジー内のすべてのデータベースでカウントされるようになります。パラメーターはデフォルトでオフになっています。

対応するエントリー **lastusn** は、ルート DSE エントリーに保持され、最後に割り当てられた USN を示します。**local** モードでは、**lastusn** はバックエンドデータベースごとに最後に割り当てられた USN を表示します。**global** モードでは、**lastusn** はトポロジー全体に最後に割り当てられた USN を表示します。

OID	2.16.840.1.113730.3.1.606
構文	整数
多値または単一値	単一値
定義される場所	Directory Server

## 6.9. INTERNALCREATORSNAME

Directory Server ユーザーではなく、プラグインまたはサーバーで作成されたエントリーの場合には、この属性は、(プラグイン DN により) 内部ユーザーがどのようなエントリーを作成したかを記録します。

**internalCreatorsname** 属性は、常にプラグインを ID 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 により

変更が加えられると、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。

OID	2.16.840.1.113730.3.1.2114
構文	DN
多値または単一値	単一値
定義される場所	Directory Server

## 6.10. INTERNALMODIFIERSNAME

Directory Server ユーザーではなく、プラグインまたはサーバーでエントリーを編集した場合、この属性は、内部ユーザー (プラグイン DN を使用) がエントリーを修正した内容を記録します。

**internalModifiersname** 属性は、常にプラグインを ID 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 により変更が加えられると、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。

OID	2.16.840.1.113730.3.1.2113
構文	DN
多値または単一値	単一値
定義される場所	Directory Server

## 6.11. HASSUBORDINATES

この属性は、エントリーに従属エントリーがあるかどうかを示します。

OID	1.3.6.1.4.1.1466.115.121.1.7
構文	Boolean
多値または単一値	単一値
定義される場所	numSubordinates Internet Draft

## 6.12. LASTLOGINTIME

**lastLoginTime** 属性には、YYMMDDHHMMSSZ の形式で、指定されたアカウントがディレクトリーに対して最後に認証されたときのタイムスタンプが含まれます。以下に例を示します。

```
lastLoginTime: 20200527001051Z
```

これは、アカウントが使用されていない期間をもとに、アカウントのロックアウトポリシーを評価するために使用されます。

OID	2.16.840.1.113719.1.1.4.1.35
構文	GeneralizedTime
多値または単一値	単一値
定義される場所	Directory Server

### 6.13. LASTMODIFIEDBY

**lastModifiedBy** 属性には、エントリーを最後に編集したユーザーの識別名 (DN) が含まれます。以下に例を示します。

```
lastModifiedBy: cn=Barbara Jensen,ou=Engineering,dc=example,dc=com
```

OID	0.9.2342.19200300.100.1.24
構文	DN
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 6.14. LASTMODIFIEDTIME

**lastModifiedTime** 属性には、エントリーが最後に変更された時刻が UTC 形式で含まれています。以下に例を示します。

```
lastModifiedTime: Thursday, 22-Sep-93 14:15:00 GMT
```

OID	0.9.2342.19200300.100.1.23
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 1274</a>

### 6.15. LDAPSUBENTRY

これらのエントリーは、操作データを保持します。このオブジェクトクラスは、LDAP サブエントリーインターネットドラフト

## 上級クラス

top

### OID

2.16.840.1.113719.2.142.6.1.1

表6.1 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。

表6.2 使用できる属性

属性	定義
「cn (commonName)」	エントリーの一般的な名前を指定します。

## 6.16. LDAPSYNTAXES

この属性は、実装されている構文を識別し、各値は1つの構文に対応します。

OID	1.3.6.1.4.1.1466.101.120.16
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2252</a>

## 6.17. MATCHINGRULES

この属性は、サブスキーマ内で使用される照合ルールを定義します。各値は1つの一致ルールを定義します。

OID	2.5.21.4
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2252</a>

## 6.18. MATCHINGRULEUSE



この属性は、サブスキーマで照合ルールが適用される属性タイプを示します。

OID	2.5.21.8
構文	DirectoryString
多値または単一値	複数値
定義される場所	<a href="#">RFC 2252</a>

## 6.19. MODIFYTIMESTAMP

この属性には、エントリーが最後に変更された日時が含まれます。

OID	2.5.18.2
構文	GeneralizedTime
多値または単一値	単一値
定義される場所	<a href="#">RFC 1274</a>

## 6.20. MODIFIERSNAME

この属性には、エントリーを最後に変更したユーザーの名前が含まれます。

OID	2.5.18.4
構文	DN
多値または単一値	単一値
定義される場所	<a href="#">RFC 1274</a>

## 6.21. NAMEFORMS

この属性は、サブスキーマで使用される名前の形式を定義します。値ごとに名前形式が1つ定義されます。

OID	2.5.21.7
構文	DirectoryString
多値または単一値	複数値

定義される場所	<a href="#">RFC 2252</a>
---------	--------------------------

## 6.22. NSACCOUNTLOCK

この属性は、アカウントがアクティブか非アクティブかを示します。

OID	2.16.840.1.113730.3.1.610
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

## 6.23. NSAIMSTATUSGRAPHIC

この属性には、AIM のユーザーステータスを示すグラフィックを示すパスが含まれます。

OID	2.16.840.1.113730.3.1.2018
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

## 6.24. NSAIMSTATUSTEXT

この属性には、現在の AIM ユーザーステータスを示すテキストが含まれています。

OID	2.16.840.1.113730.3.1.2017
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

## 6.25. NSBACKENDSUFFIX

これには、バックエンドで使用される接尾辞が含まれます。

OID	2.16.840.1.113730.3.1.803
-----	---------------------------

構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

## 6.26. NSCPENTRYDN

この属性には、tombstone エントリー用の (以前の) エントリー DN が含まれます。

OID	2.16.840.1.113730.3.1.545
構文	DN
多値または単一値	単一値
定義される場所	Directory Server

## 6.27. NSDS5REPLCONFLICT

この属性は、同期またはレプリケーションプロセスで自動解決できない変更で競合があるエントリーに含まれます。**nsDS5ReplConflict** の値には、競合しているエントリーに関する情報が含まれています。通常は、現在のエントリーと tombstone エントリーの両方の **nsUniqueID** で参照されます。

OID	2.16.840.1.113730.3.1.973
構文	DirectoryString
多値または単一値	複数値
定義される場所	Directory Server

## 6.28. NSICQSTATUSGRAPHIC

この属性には、ICQ のユーザステータスを示すグラフィックを示すパスが含まれます。

OID	2.16.840.1.113730.3.1.2022
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

## 6.29. NSICQSTATUSTEXT

この属性には、現在の ICQ ユーザーステータスのテキストが含まれます。

OID	2.16.840.1.113730.3.1.2021
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

## 6.30. NSIDLETIMEOUT

この属性は、ユーザーベースの接続アイドルタイムアウト期間を秒単位で識別します。

OID	2.16.840.1.113730.3.1.573
構文	整数
多値または単一値	単一値
定義される場所	Directory Server

## 6.31. NSIDLISTSCANLIMIT

この属性は、検索操作中に検索されるエントリー ID の数を指定します。検索パフォーマンスを向上させるために、デフォルト値を保持します。検索パフォーマンスにおける ID リストの影響に関する詳細な説明は、[Red Hat Directory Server 管理ガイド](#)のインデックスの管理の章の検索アルゴリズムの概要セクションを参照してください。

OID	2.16.840.1.113730.3.1.2106
構文	整数
多値または単一値	単一値
定義される場所	Directory Server

## 6.32. NSLOOKTHROUGHLIMIT

この属性は、対象のユーザーに対して、サーバーが検索操作時に検索可能な最大エントリー数を設定します。この属性はサーバー自体で設定され、ユーザーが検索を開始するときにユーザーに適用されません。

OID	2.16.840.1.113730.3.1.570
-----	---------------------------

構文	整数
多値または単一値	単一値
定義される場所	Directory Server

### 6.33. NSPAGEDIDLISTSCANLIMIT

この属性は、簡単なページ結果制御を使用して検索操作で特定されるエントリー ID の数を指定します。この属性は **nsIDListScanLimit** 属性と同じように機能しますが、単純なページ結果制御による検索にのみ適用される点が異なります。

この属性が存在しないか、ゼロに設定されている場合は、**nsIDListScanLimit** を使用してページングされた検索およびページ以外の検索を行います。

OID	2.16.840.1.113730.3.1.2109
構文	整数
多値または単一値	単一値
定義される場所	Directory Server

### 6.34. NSPAGEDLOOKTHROUGHLIMIT

この属性は、単純なページ結果制御を使用する検索の候補エントリーを調べる時に Directory Server がチェックするエントリーの最大数を指定します。この属性は **nsLookThroughLimit** 属性と同じように機能しますが、単純なページ結果制御の検索にのみ適用される点が異なります。

この属性が存在しないか、ゼロに設定されている場合は、**nsLookThroughLimit** を使用して、ページングされた検索と、ページングされていない検索の両方を行います。

OID	2.16.840.1.113730.3.1.2108
構文	整数
多値または単一値	単一値
定義される場所	Directory Server

### 6.35. NSPAGEDSIZELIMIT

この属性は、簡単なページ結果制御を使用する検索操作から返すエントリーの最大数を設定します。これにより、ページ検索の **nsSizeLimit** 属性がオーバーライドされます。

この値がゼロに設定されている場合、**nsSizeLimit** 属性は、そのユーザーに対してページ化された検索だけでなく、ページ化されていない検索にも使用されるか、グローバル設定が使用されます。

OID	2.16.840.1.113730.3.1.2107
構文	整数
多値または単一値	単一値
定義される場所	Directory Server

## 6.36. NSPARENTUNIQUEID

レプリケーションに保存される tombstone (削除済み) エントリーの場合、**nsParentUniqueid** 属性には、元のエントリーの親の DN またはエントリー ID が含まれます。

OID	2.16.840.1.113730.3.1.544
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

## 6.37. NSROLE

この属性は、計算属性で、エントリー自体と一緒に保存されません。エントリーが属するロールを識別します。

OID	2.16.840.1.113730.3.1.574
構文	DN
多値または単一値	複数值
定義される場所	Directory Server

## 6.38. NSROLEDN

この属性には、エントリーに適用されるすべてのロールの識別名が含まれます。マネージドロールのメンバーシップは、ロールの DN をエントリーの **nsRoleDN** 属性に追加してエントリーが追加されるタイミングで付与されます。以下に例を示します。

```
dn: cn=staff,ou=employees,dc=example,dc=com
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsSimpleRoleDefinition
objectclass: nsManagedRoleDefinition

dn: cn=userA,ou=users,ou=employees,dc=example,dc=com
```

```

objectclass: top
objectclass: person
sn: uA
userpassword: secret
nsroledn: cn=staff,ou=employees,dc=example,dc=com

```

ネストされたロールは、対象のロールに任意のタイプのロールが1つまたは複数含まれていることを指定します。その場合、**nsRoleDN** は含まれるロールの DN を定義します。以下に例を示します。

```

dn: cn=everybody,ou=employees,dc=example,dc=com
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsNestedRoleDefinition
nsroledn: cn=manager,ou=employees,dc=example,dc=com
nsroledn: cn=staff,ou=employees,dc=example,dc=com

```

OID	2.16.840.1.113730.3.1.575
構文	DN
多値または単一値	複数値
定義される場所	Directory Server

## 6.39. NSROLEFILTER

この属性は、フィルターがロールに属するエントリーを識別するように設定します。

OID	2.16.840.1.113730.3.1.576
構文	IA5String
多値または単一値	単一値
定義される場所	<a href="#">RFC 2252</a>

## 6.40. NSSCHEMACSN

この属性は、サブスキーマ DSE 属性タイプの1つです。

OID	2.5.21.82.16.840.1.113730.3.1.804
構文	DirectoryString
多値または単一値	単一値

定義される場所	Directory Server
---------	------------------

## 6.41. NSSIZELIMIT

この属性は、データベースまたはデータベースリンクのデフォルトのサイズ制限をバイト単位で示します。

OID	2.16.840.1.113730.3.1.571
構文	整数
多値または単一値	単一値
定義される場所	Directory Server

## 6.42. NSTIMELIMIT

この属性は、データベースまたはデータベースリンクのデフォルトの検索時間制限を示します。

OID	2.16.840.1.113730.3.1.572
構文	整数
多値または単一値	単一値
定義される場所	Directory Server

## 6.43. NSTOMBSTONE (オブジェクトクラス)

Tombstone エントリーは、Directory Server から削除されたエントリーです。レプリケーションおよび復元操作の場合には、必要に応じて再生成および置き換えができるように、対象の削除済みのエントリーは保存されます。各 tombstone エントリーには、自動的に **nsTombstone** オブジェクトクラスが追加されます。

このオブジェクトクラスは Directory Server に定義されています。

### 上級クラス

top

### OID

2.16.840.1.113730.3.2.113

### 表6.3 必要な属性



属性	定義
「ObjectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。

表6.4 使用できる属性

属性	定義
「nsParentUniqueld」	元のエントリーの親エントリーの一意 ID を指定します。
「nscpEntryDN」	Tombstone エントリー内の元のエントリー DN を特定します。

## 6.44. NSUNIQUEID

この属性は、サーバーエントリーに一意 ID を指定するか、割当ます。

OID	2.16.840.1.113730.3.1.542
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

## 6.45. NSYIMSTATUSGRAPHIC

この属性には、Yahoo IM のユーザーステータスを示すグラフィックを示すパスが含まれます。

OID	2.16.840.1.113730.3.1.2020
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

## 6.46. NSYIMSTATUSTEXT

この属性には、現在の Yahoo IM ユーザーステータスのテキストが含まれています。

OID	2.16.840.1.113730.3.1.2019
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

## 6.47. NUMSUBORDINATES

この属性は、エントリーに含まれる多くの直属の部下を示します。たとえば、リーフエントリーの **numSubordinates=0** です。

OID	1.3.1.1.4.1.453.16.2.103
構文	整数
多値または単一値	単一値
定義される場所	numSubordinates インターネットドラフト

## 6.48. PASSWORDGRACEUSERTIME

この属性は、ユーザーが期限切れのパスワードを使用して試行した回数をカウントします。

OID	2.16.840.1.113730.3.1.998
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

## 6.49. PASSWORDRETRYCOUNT

この属性は、パスワードを正しく入力できなかった試行回数をカウントします。

OID	2.16.840.1.113730.3.1.93
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

## 6.50. PWDPOLICYSUBENTRY

この属性値は、新しいパスワードポリシーのエントリー DN を参照します。

OID	2.16.840.1.113730.3.1.997
構文	DirectoryString
多値または単一値	単一値
定義される場所	Directory Server

## 6.51. PWDUPDATETIME

この属性値は、アカウントのパスワードが最後に変更された時間を保存します。

OID	2.16.840.1.113730.3.1.2133
構文	GeneralizedTime
多値または単一値	単一値
定義される場所	Directory Server

## 6.52. SUBSCHEMASUBENTRY

この属性には、スキーマ情報を含むエントリーの DN が含まれます。以下に例を示します。

```
subschemaSubentry: cn=schema
```

OID	2.5.18.10
構文	DN
多値または単一値	単一値
定義される場所	<a href="#">RFC 2252</a>

## 6.53. GLUE(オブジェクトクラス)

**glue** オブジェクトクラスは、特別な状態のエントリー (レプリケーションの競合が原因で再生成された場合など) を定義します。

このオブジェクトクラスは Directory Server で定義されます。

上級クラス

top

**OID**

2.16.840.1.113730.3.2.30

表6.5 必要な属性

属性	定義
「objectClass」	エントリーに割り当てられたオブジェクトクラスを指定します。

**6.54. PASSWORDOBJECT (オブジェクトクラス)**

このオブジェクトクラスは、ユーザーのパスワード情報をディレクトリーに格納するエントリーに使用されます。

このオブジェクトクラスは Directory Server に定義されています。

**上級クラス**

top

**OID**

2.16.840.1.113730.3.2.12

表6.6 必要な属性

「objectClass」	エントリーのオブジェクトクラスを定義します。
---------------	------------------------

表6.7 使用できる属性

「accountUnlockTime」	アカウントのロックアウト後に、ユーザーがディレクトリーに再度バインドできるようになるまでに待機する必要がある時間を指します。
「passwordAllowChangeTime」	ユーザーがパスワードを変更できるようになるまでに待機する必要がある時間を指定します。
「passwordExpirationTime」	ユーザーのパスワードの有効期限が切れるまでに経過する時間の長さを指定します。
「passwordExpWarned」	パスワードの有効期限の警告がユーザーに送信されたことを示します。
「passwordGraceUserTime」	パスワードの有効期限が切れてから、ユーザーが試行できるログイン回数を指定します。
「passwordHistory (パスワード履歴)」	ユーザーの以前のパスワードの履歴が含まれます。

「passwordRetryCount」	パスワードを正しく入力できなかった試行回数をカウントします。
「pwdpolicysubentry」	新しいパスワードポリシーのエントリー DN を参照します。
「retryCountResetTime」	<b>passwordRetryCount</b> 属性がリセットされるまでに経過する時間を指定します。

## 6.55. SUBSCHEM (オブジェクトクラス)

これは、サブスキーマ管理領域のサブスキーマを管理する補助オブジェクトクラスサブエントリーを指定します。サブスキーマを表すポリシーパラメーターを表す操作属性を保持します。

このオブジェクトクラスは RFC 2252 に定義されています。

### 上級クラス

top

### OID

2.5.20.1

表6.8 必要な属性

「objectClass」	エントリーのオブジェクトクラスを定義します。
---------------	------------------------

表6.9 使用できる属性

「attributeTypes」	サブスキーマ内で使用される属性タイプ。
「dITContentRules」	サブスキーマ内で有効な DIT コンテンツルールを定義します。
「dITStructureRules」	サブスキーマ内で有効な DIT 構造ルールを定義します。
「matchingRuleUse」	サブスキーマで照合致ルールが適用される属性タイプを示します。
「matchingRules」	サブスキーマ内で使用される照合ルールを定義します。
「nameForms」	サブスキーマで使用される名前の形式を定義します。
「objectClasses」	サブスキーマで使用されるオブジェクトクラスを定義します。

## 第7章 ログファイルのリファレンス

Red Hat Directory Server (Directory Server) は、ディレクトリーのアクティビティを監視するのに役立つログを提供します。監視は、障害を素早く検出および修正し、プロアクティブに実施された場合に、障害やパフォーマンスが低下する前に潜在的な問題を予測して解決するのに役立ちます。ディレクトリーを効果的に監視するには、ログファイルの設定と内容を理解する必要があります。

本章では、ログメッセージの完全なリストは提供しません。ただし、本章で紹介している情報は、一般的な問題の開始点として機能し、アクセス、エラー、監査ログの情報をよりよく理解するのに役立ちます。

ログは Directory Server インスタンスごとに保持され、`/var/log/dirsrv/slaped-instance` ディレクトリーに置かれます。

### 7.1. アクセスログリファレンス

Directory Server のアクセスログには、ディレクトリーへのクライアント接続に関する詳細情報が含まれます。接続は、同じクライアントからのリクエストシーケンスで、以下の設定になっています。

- 接続レコード - 接続インデックスおよびクライアントの IP アドレスを提供します。
- バインドレコード
- バインド結果レコード
- レコードの操作要求と操作結果ペアのシーケンス、または接続、クローズ、および破棄レコードの場合の個別レコード
- バインド解除レコード
- クローズレコード

次に、アクセスログエントリーの例を示します。

```
[23/Jun/2020:16:30:27.388006333 -0400] conn=20 op=5 SRCH base="dc=example,dc=com"
scope=2 filter="(&(objectClass=top)(objectClass=ldapsubentry)(objectClass=passwordpolicy))"
attrs="distinguishedName"
```

個別に表示される接続、クローズ、および破棄レコードとは別に、すべてのレコードはペアで表示され、サービスの要求レコードとそれに続く **RESULT** レコードで設定されます。

```
[23/Jun/2020:16:30:27.390881301 -0400] conn=20 op=5 RESULT err=0 tag=101 nentries=0
wtime=0.000035342 optime=0.002877749 etime=0.002911121
```

**RESULT** メッセージには、以下のパフォーマンス関連のエントリーが含まれます。

- **wtime**: ワーカーレッドが操作をピックアップするまで、操作がワークキューで待機していた時間
- **optime**: 実際の操作がタスクを実行するのにかった時間
- **etime**: サーバーが操作を受け取ってから結果をクライアントに返すまでの経過時間



## 注記

**wtime** および **optime** の値により、サーバーがどのように負荷をおよび操作を処理するかに関する有用な情報を提供されます。Directory Server がこれらの統計値を収集するタイミングにより、**wtime** 値と **optime** 値の合計は **etime** の値よりも若干大きくなります。ただし、その違いは無視できるレベルです。

アクセスログには、**nsslapd-accesslog-level** 属性で設定されるさまざまなロギングのレベルがあります。以下のセクションでは、デフォルトのアクセスログの内容、ログレベル、およびさまざまなログレベルでログに記録されるコンテンツの概要を説明します。

- 「アクセスロギングレベル」
- 「デフォルトのアクセスログのコンテンツ」
- 「追加のアクセスロギングレベルのアクセスログコンテンツ」

アクセスログの形式を変更することはできません。

### 7.1.1. アクセスロギングレベル

アクセスロギングのレベルにより、さまざまな量の情報が生成され、さまざまな操作が記録されます。ログレベルは、インスタンスの「**nsslapd-accesslog-level(アクセスログレベル)**」設定属性で設定されます。デフォルトのロギングレベルはレベル 256 で、エントリーへのアクセスをログに記録しますが、4 つの異なるログレベルを利用できます。

- 0 = アクセスログなし
- 4 = 内部アクセス操作のロギング
- 256 = エントリーへのアクセスのログ。
- 512 = エントリーおよび参照情報にアクセスするためのロギング

このレベルは加算されるため、さまざまな種類のロギングを有効にするには、これらのレベルの値を一緒に加算します。たとえば、内部アクセス操作、エントリーアクセス、および参照をログに記録するには、**nsslapd-accesslog-level** の値を **516 (512+4)** に設定します。

### 7.1.2. デフォルトのアクセスログのコンテンツ

このセクションでは、以下に記載のデフォルトのアクセスロギングレベルの抽出に基づいて、アクセスログの内容の詳細を説明します。

#### 例7.1 アクセスログの例

```
[21/Apr/2020:11:39:51 -0700] conn=11 fd=608 slot=608 connection from 207.1.153.51 to
192.18.122.139
[21/Apr/2020:11:39:51 -0700] conn=11 op=0 BIND dn="cn=Directory Manager" method=128
version=3
[21/Apr/2020:11:39:51 -0700] conn=11 op=0 RESULT err=0 tag=97 nentries=0 etime=0
[21/Apr/2020:11:39:51 -0700] conn=11 op=1 SRCH base="dc=example,dc=com" scope=2 filter="
(mobile=+1 123 456-7890)"
[21/Apr/2020:11:39:51 -0700] conn=11 op=1 RESULT err=0 tag=101 nentries=1 etime=3 notes=U
[21/Apr/2020:11:39:51 -0700] conn=11 op=2 UNBIND
[21/Apr/2020:11:39:51 -0700] conn=11 op=2 fd=608 closed - U1
```

```
[21/Apr/2020:11:39:52 -0700] conn=12 fd=634 slot=634 connection from 207.1.153.51 to
192.18.122.139
[21/Apr/2020:11:39:52 -0700] conn=12 op=0 BIND dn="cn=Directory Manager" method=128
version=3
[21/Apr/2020:11:39:52 -0700] conn=12 op=0 RESULT err=0 tag=97 nentries=0 etime=0
[21/Apr/2020:11:39:52 -0700] conn=12 op=1 SRCH base="dc=example,dc=com" scope=2 filter="
(uid=bjensen)"
[21/Apr/2020:11:39:52 -0700] conn=12 op=2 ABANDON targetop=1 msgid=2 nentries=0 etime=0
[21/Apr/2020:11:39:52 -0700] conn=12 op=3 UNBIND
[21/Apr/2020:11:39:52 -0700] conn=12 op=3 fd=634 closed - U1
[21/Apr/2020:11:39:53 -0700] conn=13 fd=659 slot=659 connection from 207.1.153.51 to
192.18.122.139
[21/Apr/2020:11:39:53 -0700] conn=13 op=0 BIND dn="cn=Directory Manager" method=128
version=3
[21/Apr/2020:11:39:53 -0700] conn=13 op=0 RESULT err=0 tag=97 nentries=0 etime=0
[21/Apr/2020:11:39:53 -0700] conn=13 op=1 EXT oid="2.16.840.1.113730.3.5.3"
[21/Apr/2020:11:39:53 -0700] conn=13 op=1 RESULT err=0 tag=120 nentries=0 etime=0
[21/Apr/2020:11:39:53 -0700] conn=13 op=2 ADD dn="cn=Sat Apr 21 11:39:51 MET DST
2020,dc=example,dc=com"
[21/Apr/2020:11:39:53 -0700] conn=13 op=2 RESULT err=0 tag=105 nentries=0 etime=0
csn=3b4c8cfb000000030000
[21/Apr/2020:11:39:53 -0700] conn=13 op=3 EXT oid="2.16.840.1.113730.3.5.5"
[21/Apr/2020:11:39:53 -0700] conn=13 op=3 RESULT err=0 tag=120 nentries=0 etime=0
[21/Apr/2020:11:39:53 -0700] conn=13 op=4 UNBIND
[21/Apr/2020:11:39:53 -0700] conn=13 op=4 fd=659 closed - U1
[21/Apr/2020:11:39:55 -0700] conn=14 fd=700 slot=700 connection from 207.1.153.51 to
192.18.122.139
[21/Apr/2020:11:39:55 -0700] conn=14 op=0 BIND dn="" method=sasl version=3 mech=DIGEST-
MD5
[21/Apr/2020:11:39:55 -0700] conn=14 op=0 RESULT err=14 tag=97 nentries=0 etime=0, SASL
bind in progress
[21/Apr/2020:11:39:55 -0700] conn=14 op=1 BIND dn="uid=jdoe,dc=example,dc=com"
method=sasl version=3 mech=DIGEST-MD5
[21/Apr/2020:11:39:55 -0700] conn=14 op=1 RESULT err=0 tag=97 nentries=0 etime=0
dn="uid=jdoe,dc=example,dc=com"
[21/Apr/2020:11:39:55 -0700] conn=14 op=2 UNBIND
[21/Apr/2020:11:39:53 -0700] conn=14 op=2 fd=700 closed - U1
```

## 接続番号

すべての外部 LDAP 要求が、一連の接続番号 (ここでは **conn=11** で、サーバー起動直後の **conn=0** から始まります) と共にリスト表示されます。

```
[21/Apr/2020:11:39:51 -0700] conn=11 fd=608 slot=608 connection from 207.1.153.51 to
192.18.122.139
```

内部 LDAP 要求は、デフォルトではアクセスログに記録されません。内部アクセス操作のロギングを有効にするには、`「nsslapd-accesslog-level(アクセスログレベル)」` 設定属性でアクセスログレベル **4** を指定します。

## ファイル記述子

外部 LDAP クライアントから Directory Server へのすべての接続には、オペレーティングシステムからのファイル記述子またはソケット記述子 (この場合は **fd=608**) が必要です。**fd=608** は、使用された利用可能なファイル記述子の合計プールの中の、ファイル記述子番号 608 であったことを示しています。



```
[21/Apr/2020:11:39:51 -0700] conn=11 fd=608 slot=608 connection from 207.1.153.51 to 192.18.122.139
```

### スロット番号

スロット番号 (この場合は **slot=608**) はアクセスログのレガシー部分で、ファイル記述子と同じ意味を持ちます。アクセスログのこの部分は無視します。

```
[21/Apr/2020:11:39:51 -0700] conn=11 fd=608 slot=608 connection from 207.1.153.51 to 192.18.122.139
```

### 操作番号

特定の LDAP 要求を処理するには、Directory Server は必要な一連の操作を実行します。特定の接続では、すべての操作要求と操作結果のペアに **op=0** で始まる一連の操作番号が割り当てられ、実行される個別の操作を特定します。

```
[21/Apr/2020:11:39:51 -0700] conn=11 op=0 RESULT err=0 tag=97 nentries=0 etime=0
```

「[デフォルトのアクセスログのコンテンツ](#)」では、バインド操作の要求と結果のペアを **op=0** とし、次に LDAP 検索の要求と結果のペアを **op=1** のように行います。アクセスログのエントリ **op=-1** は、通常この接続の LDAP 要求が外部 LDAP クライアントによって発行されず、代わりに内部で開始されたことを意味します。

### メソッドのタイプ

メソッド番号 (この場合は **method=128**) は、クライアントによって使用された LDAPv3 バインドメソッドを示します。

```
[21/Apr/2020:11:39:51 -0700] conn=11 op=0 BIND dn="cn=Directory Manager" method=128 version=3
```

可能なバインドメソッドの値は次の3つです。

- 認証の場合は **0**
- ユーザーパスワードを使用した単純なバインドの場合は **128**
- 外部認証メカニズムを使用する SASL バインドの場合は **sasl**

### バージョン番号

バージョン番号 (この場合は **version=3**) は、LDAP クライアントが LDAP サーバーとの通信に使用した LDAP バージョン番号 (LDAPv2 または LDAPv3 のいずれか) を示します。

```
[21/Apr/2020:11:39:51 -0700] conn=11 op=0 BIND dn="cn=Directory Manager" method=128 version=3
```

### エラー番号

エラー番号 (この場合は **err=0**) は、実行された LDAP 操作から返された LDAP 結果コードを提供します。LDAP エラー番号 **0** は、操作が成功したことを意味します。LDAP 結果コードのより包括的なリストは、「[LDAP の結果コード](#)」を参照してください。

```
[21/Apr/2020:11:39:51 -0700] conn=11 op=0 RESULT err=0 tag=97 nentries=0 etime=0
```

## タグ番号

タグ番号 (この場合は **tag=97**) は、返される結果のタイプを示します。これは、ほとんどの場合、実行された操作のタイプを反映します。使用されるタグは LDAP プロトコルからの BER タグです。

```
[21/Apr/2020:11:39:51 -0700] conn=11 op=0 RESULT err=0 tag=97 nentries=0 etime=0
```

表7.1 一般的に使用されるタグ

タグ	説明
tag=97	クライアントのバインド操作の結果。
tag=100	検索されている実際のエントリー。
tag=101	検索操作からの結果。
tag=103	変更操作からの結果。
tag=105	追加操作からの結果。
tag=107	削除操作からの結果。
tag=109	moddn 操作からの結果。
tag=111	比較操作からの結果。
tag=115	検索を実行したエントリーが、必要なエントリーへの参照を保持する場合の検索参照。検索参照は参照に関して表現されます。
tag=120	拡張操作からの結果。
tag=121	中間操作の結果



### 注記

**tag=100** および **tag=115** は、それ自体が結果タグではないため、アクセスログに記録される可能性はほとんどありません。

## エントリー数

**nentries** は、LDAP クライアントの要求に一致することが検出されたエントリーの数を示します (この場合は **nentries=0**)。

```
[21/Apr/2020:11:39:51 -0700] conn=11 op=0 RESULT err=0 tag=97 nentries=0 etime=0
```

## 経過時間

**etime** は、経過時間 (この場合は **etime=3**)、または Directory Server が LDAP 操作を実行するのにかった時間 (秒単位) を示します。

```
[21/Apr/2020:11:39:51 -0700] conn=11 op=1 RESULT err=0 tag=101 nentries=1 etime=3 notes=U
```

**etime** 値が **0** の場合は、操作の実行に実際には 0 ナノ秒かかったことを意味します。

### LDAP 要求タイプ

LDAP 要求タイプは、LDAP クライアントによって発行されている LDAP 要求のタイプを示します。可能な値は次のとおりです。

- 検索の **SRCH**
- 変更の **MOD**
- 削除の **DEL**
- 追加の **ADD**
- moddn の **MODDN**
- 拡張操作の **EXT**
- 破棄操作の **ABANDON**

LDAP 要求によってエントリーがソートされた場合、メッセージ **SORT serialNo** がログに記録され、その後ソートされた候補エントリーの数が続きます。以下に例を示します。

```
[04/May/2020:15:51:46 -0700] conn=114 op=68 SORT serialNo (1)
```

丸かっこで囲まれた数字は、ソートされた候補エントリーの数を指定します。この場合は **1** です。

### LDAP 応答タイプ

LDAP 応答タイプは、LDAP クライアントによって発行されている LDAP 応答を示します。以下の3つの値を使用できます。

- **RESULT**
- **ENTRY**
- LDAP 参照または検索参照である **REFERRAL**

### 検索インジケータ

Directory Server は、ログエントリーの **notes** フィールドで検索に関する追加情報を提供します。以下に例を示します。

```
[21/Apr/2016:11:39:51 -0700] conn=11 op=1 RESULT err=0 tag=101 nentries=1 etime=3 notes=U
```

次の検索インジケータがあります。

#### ページ検索インジケータ: **notes=P**

リソースが制限された LDAP クライアントでは、LDAP サーバーが検索操作の結果を返す速度を制御できます。実行された検索で LDAP 制御拡張を使用して検索結果を単純にページングすると、DirectoryServer は **notes=P** ページ検索インジケータをログに記録します。このインジケータは

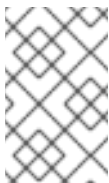
情報を提供するもので、それ以上のアクションは必要ありません。  
 詳細は、[RFC 2696](#) を参照してください。

## インデックス化されていない検索インジケータ:notes=A および notes=U

属性がインデックス化されていない場合、Directory Server はそれらを直接データベースで検索する必要があります。この手順では、インデックスファイルを検索する場合よりも多くのリソースが消費されます。

以下のインデックス化されていない検索インジケータをログに記録できます。

- **notes=A**  
 フィルターのすべての候補属性はインデックス化されておらず、完全なテーブルスキャンが必要でした。これは、**nsslapd-lookthroughlimit** パラメーターで設定した値を超える可能性があります。
- **notes=U**  
 この状態は以下の状況で設定されます。
  - 少なくとも1つの検索用語がインデックス化されていない。
  - 検索操作時に、**nsslapd-idlistscanlimit** パラメーターで設定された制限に達した。詳細は「[nsslapd-idlistscanlimit](#)」を参照してください。  
 インデックス化されていない検索は、以下のシナリオで発生します。
- 検索に使用されるインデックスファイル内で、**nsslapd-idlistscanlimit** パラメーターの値に達した。
- インデックスファイルが存在しない。
- インデックスファイルが検索に必要な方法で設定されなかった。  
 今後の検索を最適化するには、インデックスに頻繁に検索されるインデックス化されていない属性を追加します。詳細は、[Directory Server 管理ガイド](#)の該当するセクションを参照してください。



### 注記

通常、インデックス化されていない検索には時間がかかるため、インデックス化されていない検索インジケータには大きな **etime** 値が伴うことがあります。

単一の値の他に、**notes** フィールドには **notes=P,A** および **notes=U,P** の値の組み合わせを使用できます。

## VLV 関連エントリー

検索に仮想リストビュー (VLV) が含まれる場合、適切なエントリーがアクセスログファイルに記録されます。他のエントリーと同様に、VLV 固有のエントリーは、要求と応答の情報を並べて表示します。

### VLV RequestInformation ResponseInformation

RequestInformation の形式は次のとおりです。

### beforeCount:afterCount:index:contentCount

クライアントが position-by-value の VLV 要求 (最初の部分の形式) を使用する場合、要求情報は **beforeCount: afterCount: value** になります。

ResponseInformation 形式は次のとおりです。

### targetPosition:contentCount (resultCode)

以下の例では、VLV 固有のエントリーを強調表示しています。

```
[07/May/2020:11:43:29 -0700] conn=877 op=8530 SRCH base="(ou=People)" scope=2 filter="(uid=*)"
[07/May/2020:11:43:29 -0700] conn=877 op=8530 SORT uid
[07/May/2020:11:43:29 -0700] conn=877 op=8530 VLV 0:5:0210 10:5397 (0)
[07/May/2020:11:43:29 -0700] conn=877 op=8530 RESULT err=0 tag=101 nentries=1 etime=0
```

上記の例では、最初の部分 **0:5:0210** は VLV 要求情報です。

- beforeCount は **0** です。
- afterCount は **5** です。
- 値は **0210** です。

2 番目の部分 **10:5397(0)** は VLV 応答情報です。

- targetPosition は **10** です。
- contentCount は **5397** です。
- (resultCode) は **(0)** です。

### 検索範囲

エントリー **scope=n** は、実行する検索の範囲を定義します。n には、**0**、**1**、または **2** の値を指定できます。

- ベース検索の場合は **0**
- 1 レベル検索の場合は **1**
- サブツリー検索の場合は **2**

### 拡張操作 OID

例7.1「アクセスログの例」の **EXT oid="2.16.840.1.113730.3.5.3"** または **EXT oid="2.16.840.1.113730.3.5.5"** などの拡張操作 OID は、実行されている拡張操作の OID を提供します。表7.2「Directory Server でサポートされる LDAPv3 拡張操作」に、Directory Server でサポートされる LDAPv3 拡張操作とその OID の部分的なリストを示します。

表7.2 Directory Server でサポートされる LDAPv3 拡張操作

拡張操作名	説明	OID
-------	----	-----

拡張操作名	説明	OID
Directory Server レプリケーション要求の開始	レプリケーションセッションが必要であることを示すために、レプリケーションイニシエーターによって送信されます。	2.16.840.1.113730.3.5.3
Directory Server のレプリケーション応答	Start Replication Request Extended Operation または End Replication Request Extended Operation に対応して、レプリケーションレスポンスによって送信されます。	2.16.840.1.113730.3.5.4
Directory Server のレプリケーション終了要求	レプリケーションセッションが終了することを示すために送信されます。	2.16.840.1.113730.3.5.5
Directory Server レプリケーションエントリー要求	エントリーとその状態情報 ( <b>csn</b> および <b>UniquelIdentifier</b> ) を保持し、レプリカの初期化の実行に使用されます。	2.16.840.1.113730.3.5.6
ディレクトリーサーバーの一括インポートの開始	インポートされる接尾辞と共に一括でインポートすることを要求するためにクライアントにより送信され、一括インポートが開始されることを示すためにサーバーによって送信されます。	2.16.840.1.113730.3.5.7
ディレクトリーサーバーの一括インポートの完了	一括インポートの終了を通知するためにクライアントによって送信され、それを確認するためにサーバーによって送信されます。	2.16.840.1.113730.3.5.8

## シーケンス番号の変更

この場合の **csn=3b4c8cfb000000030000** の変更シーケンス番号は、この特定の命名コンテキストでレプリケーションが有効になっていることを示すレプリケーション変更シーケンス番号です。

## 破棄メッセージ

破棄メッセージは、操作が中断されていることを示します。

```
[21/Apr/2020:11:39:52 -0700] conn=12 op=2 ABANDON targetop=1 msgid=2 nentries=0 etime=0
```

**nentries=0** は、操作が中断される前に送信されたエントリーの数を示します。**etime=0** の値は経過した時間 (秒単位) を示し、**targetop=1** は以前に開始した操作 (アクセスログの以前に表示される) からの操作値に対応します。

中断する操作をメッセージ ID が探すことができたかどうかによって、2 種類のログ **ABANDON** メッセージが記録される可能性があります。メッセージ ID が操作 (**targetop**) を見つけるのに成功した場合

は、上記のようなログが記録されます。しかし、メッセージ ID が操作を見つけられなかった場合や、**ABANDON** 要求が送信される前に操作がすでに終了した場合は、以下のようなログが記録されません。

```
[21/Apr/2020:11:39:52 -0700] conn=12 op=2 ABANDON targetop=NOTFOUND msgid=2
```

**targetop=NOTFOUND** は、中断する操作が不明な操作であるか、すでに完了していることを示します。

## メッセージ ID

メッセージ ID(ここでは **msgid=2**) は、LDAP SDK クライアントによって生成された LDAP 操作識別子です。メッセージ ID は操作番号とは異なる値となる可能性があります。同じ操作を識別します。メッセージ ID は **ABANDON** 操作に使用され、破棄されるクライアント操作をユーザーに通知します。

```
[21/Apr/2020:11:39:52 -0700] conn=12 op=2 ABANDON targetop=NOTFOUND msgid=2
```



### 注記

Directory Server の操作番号は 0 から始まり、ほとんどの LDAP SDK/クライアント実装では、メッセージ ID 番号は 1 から始まります。通常、メッセージ ID が Directory Server の操作番号に 1 を加えた値と等しくなるのはこのためです。

## SASL マルチステージバインドログイン

Directory Server では、マルチステージバインドのログは明示的です。バインドプロセスの各段階がログに記録されます。これらの SASL 接続のエラーコードは、実際には戻りコードです。[例7.1「アクセスログの例」](#)では、SASL バインドは現在進行中であるため、戻りコードが **err=14** になります。つまり、接続は開いたままで、対応する進捗ステートメント **SASL bind in progress** があります。

```
[21/Apr/2020:11:39:55 -0700] conn=14 op=0 BIND dn="" method=sasl version=3 mech=DIGEST-MD5
```

```
[21/Apr/2020:11:39:55 -0700] conn=14 op=0 RESULT err=14 tag=97 nentries=0 etime=0, SASL bind in progress
```

SASL バインドのログインでは、**sasl** メソッドの後に LDAP [バージョン番号](#) および使用される SASL メカニズムが続きます。GSS-API メカニズムのケースを以下に示します。

```
[21/Apr/2020:12:57:14 -0700] conn=32 op=0 BIND dn="" method=sasl version=3 mech=GSSAPI
```



### 注記

以前は、バインド要求行ではなく、認証された DN (アクセス制御の決定に使用される DN) が BIND 結果行にログインされるようになりました。

```
[21/Apr/2020:11:39:55 -0700] conn=14 op=1 RESULT err=0 tag=97 nentries=0 etime=0 dn="uid=jdoe,dc=example,dc=com"
```

SASL バインドでは、バインド要求行に表示される DN 値はサーバーによって使用されず、したがって該当しません。ただし、認証された DN が監査目的で使用する必要がある DN である場合 (SASL バインド)、これを明確にログに記録することが重要です。この認証済み DN をバインド結果行にログとして記録すると、どの DN がどの DN であるかを明確にできます。

### 7.1.3. 追加のアクセスロギングレベルのアクセスログコンテンツ

本セクションでは、Directory Server のアクセスログで利用可能な追加のアクセスロギングレベルについて説明します。

例7.2 「内部アクセス操作レベルのアクセスログの抜粋 (レベル 4)」では、内部操作をログに記録するアクセスログレベル 4 が有効です。

#### 例7.2 内部アクセス操作レベルのアクセスログの抜粋 (レベル 4)

```
[12/Jul/2020:16:45:46 +0200] conn=Internal op=-1 SRCH
base="cn=\22dc=example,dc=com\22,cn=mapping tree,cn=config"scope=0
filter="objectclass=nsMappingTree"attrs="nsslapd-referral" options=persistent
[12/Jul/2020:16:45:46 +0200] conn=Internal op=-1 RESULT err=0 tag=48 nentries=1 etime=0
[12/Jul/2020:16:45:46 +0200] conn=Internal op=-1 SRCH
base="cn=\22dc=example,dc=com\22,cn=mapping tree,cn=config"scope=0
filter="objectclass=nsMappingTree" attrs="nsslapd-state"
[12/Jul/2020:16:45:46 +0200] conn=Internal op=-1 RESULT err=0 tag=48 nentries=1 etime=0
```

アクセスログレベル 4 では、内部操作のログが有効になります。これにより、実行される検索の詳細の他に、検索ベース、スコープ、フィルター、および要求された検索属性がログに記録されます。

以下の例では、アクセスロギングレベル **768** (512 + 256) が有効化され、エントリーおよび参照へのアクセスがログに記録されます。この抜粋では、最初の行に表示される検索リクエストへの応答として、6つのエントリーと1つの参照が返されています。

```
[12/Jul/2020:16:43:02 +0200] conn=306 fd=60 slot=60 connection from 127.0.0.1 to 127.0.0.1
[12/Jul/2020:16:43:02 +0200] conn=306 op=0 SRCH base="dc=example,dc=com" scope=2 filter="
(description=*)" attrs=ALL
[12/Jul/2020:16:43:02 +0200] conn=306 op=0 ENTRY dn="ou=Special
[12/Jul/2020:16:43:02 +0200] conn=306 op=0 ENTRY dn="cn=Accounting
Managers,ou=groups,dc=example,dc=com"
[12/Jul/2020:16:43:02 +0200] conn=306 op=0 ENTRY dn="cn=HR
Managers,ou=groups,dc=example,dc=com"
[12/Jul/2020:16:43:02 +0200] conn=306 op=0 ENTRY dn="cn=QA
Managers,ou=groups,dc=example,dc=com"
[12/Jul/2020:16:43:02 +0200] conn=306 op=0 ENTRY dn="cn=PD
Managers,ou=groups,dc=example,dc=com"
[12/Jul/2020:16:43:02 +0200] conn=306 op=0 ENTRY dn="ou=Red Hat
Servers,dc=example,dc=com"
[12/Jul/2020:16:43:02 +0200] conn=306 op=0 REFERRAL
```

#### コネクションの説明

接続の説明 (この場合は **conn=Internal**) は、接続が内部接続であることを示しています。操作番号 **op=-1** も、操作が内部で開始されたことを示しています。

```
[12/Jul/2020:16:45:46 +0200] conn=Internal op=-1 ENTRY
dn="cn=\22dc=example,dc=com\22,cn=mapping tree,cn=config"
```

#### Options Description



オプションの説明 (**options=persistent**) は、通常の検索操作とは異なる、永続的な検索が実行されることを示しています。永続的な検索は、監視形式として使用し、特定の設定に対する変更の発生時にその変更を返すように設定できます。

この例では、ログレベル **512** と **4** の両方が有効になっているため、内部アクセス操作およびエントリーと参照へのアクセスの両方がログに記録されます。

```
[12/Jul/2020:16:45:46 +0200] conn=Internal op=-1 SRCH
base="cn=\22dc=example,dc=com\22,cn=mapping tree,cn=config"scope=0
filter="objectclass=nsMappingTree"attrs="nsslapd-referral" options=persistent
```

#### 7.1.4. 一般的な接続コード

接続コードは、**closed** のログメッセージに追加されるコードで、接続終了に関する追加情報を提供します。

表7.3 一般的な接続コード

接続コード	説明
A1	クライアントが接続をアボートしました。
B1	破損した BER タグが見つかりました。BER タグ (ネットワーク経由で送信されているデータをカプセル化する) が受信時に破損している場合、B1 接続コードがアクセスログに記録されます。BER タグは、物理レイヤーのネットワークの問題や、すべてのリクエストの結果を受け取る前に LDAP クライアントがアボートした等、不適切な LDAP クライアント操作により破損する可能性があります。
B2	BER タグが <b>nsslapd-maxbersize</b> 属性値よりも長い。この設定属性の詳細は、「 <a href="#">nsslapd-maxbersize(最大メッセージサイズ)</a> 」を参照してください。
B3	破損した BER タグが見つかりました。
B4	サーバーがデータの応答をフラッシュしてクライアントに戻せなかった。
P2	接続の終了または破損が検出されている。
T1	クライアントが、指定した <b>idletimeout</b> 期間内に結果を受信しなかった。この設定属性の詳細は、「 <a href="#">nsslapd-idletimeout(デフォルトのアイドルタイムアウト)</a> 」を参照してください。
T2	<b>ioblocktimeout</b> 期間が経過した後に、サーバーは接続を終了します。この設定属性の詳細は、「 <a href="#">nsslapd-ioblocktimeout(I/O ブロックのタイムアウト)</a> 」を参照してください。

接続コード	説明
U1	クライアントがバインド解除要求を送信した後に、サーバーによって接続が終了された。バインド解除要求が送信されると、サーバーは常に接続を終了します。

## 7.2. エラーログ参照

Directory Server のエラーログは、Directory Server のトランザクションおよび操作のメッセージを記録します。これらは、失敗した操作のエラーメッセージである可能性がありますが、サーバーの起動メッセージ、ログイン、ディレクトリーの検索、接続情報など、Directory Server および LDAP タスクのプロセスに関する一般情報も含まれます。

### 7.2.1. エラーログのロギングレベル

エラーログは、操作のさまざまな量の **詳細** と、有効になっているエラーログの種類に応じてさまざまな **種類** の情報を記録することができます。

ロギングレベルは「[nsslapd-errorlog-level\(エラーログレベル\)](#)」設定属性で設定されます。デフォルトのログレベルは **16384** です。これには、LDAP の結果コードや起動メッセージなどの重要なエラーメッセージおよび標準のログメッセージが含まれます。アクセスロギングと同様に、エラーロギングレベルは加算式です。レプリケーションロギング (**8192**) とプラグインロギング (**65536**) の両方を有効にするには、ログレベルを **73728 (8192 + 65536)** に設定します。



#### 注記

高いレベルのデバッグロギングを有効にすると、サーバーのパフォーマンスが大幅に低下する可能性があります。レプリケーション (**8192**) などのデバッグログレベルは、通常の操作用ではなく、トラブルシューティングのためにのみ有効にする必要があります。

表7.4 エラーログレベル

設定	コンソール名	説明
1	関数呼び出しの追跡	サーバーに入る際にメッセージをログに記録し、関数を終了します。
2	パケット処理	サーバーが処理するパケットのデバッグ情報をログに記録します。
4	ヘビートレース出力	サーバーが関数を開始/終了する際に、追加のデバッグメッセージと共にログを記録します。
8	接続管理	SASL バインドに使用される接続方法を含む、現在の接続ステータスをログに記録します。

設定	コンソール名	説明
16	送信/受信パケット	サーバーが送受信したパケットの数を出力します。
32	検索フィルター処理	検索操作によって呼び出されるすべての関数をログに記録します。
64	設定ファイルの処理	サーバーの起動時に、サーバーで使用される <b>.conf</b> 設定ファイルを1行ごとに出力します。デフォルトでは、 <b>slapd-collations.conf</b> のみが利用でき、処理されます。
128	アクセス制御リスト処理	非常に詳細なアクセス制御リスト処理情報を提供します。
2048	ログエントリーの解析	デバッグ情報を解析するスキーマをログに記録します。
4096	Housekeeping	ハウスキーピングスレッドのデバッグ。
8192	レプリケーション	更新やエラーなど、レプリケーション関連のすべての操作に関する詳細情報をログに記録します。これはレプリケーションの問題のデバッグに重要です。
16384	デフォルト	重大なエラーや、常にエラーログに書き込まれるその他のメッセージに使用されるデフォルトのロギングレベル (例: サーバー起動メッセージ) このレベルのメッセージは、ログレベルの設定に関係なく、常にエラーログに含まれます。
32768	エントリーキャッシュ	データベースエントリーキャッシュのデバッグ。
65536	プラグイン	サーバープラグインが <b>slapi-log-error</b> を呼び出すときにログファイルにエントリーを書き込みます。そのため、これはサーバープラグインのデバッグに使用されます。

設定	コンソール名	説明
262144	アクセス制御の概要	サーバーへのアクセスに関する情報をまとめています。情報量はレベル <b>128</b> よりもはるかに少ないです。この値は、アクセス制御処理の概要が必要な場合に推奨されます。非常に詳細な処理メッセージには <b>128</b> を使用します。

## 7.2.2. エラーログコンテンツ

エラーログの形式は、アクセスログの形式とは異なります。

### サーバーが書き込むログエントリ

サーバーがファイルに書き込むエントリの形式は、以下のとおりです。

```
time_stamp - severity_level - function_name - message
```

以下に例を示します。

```
[24/Mar/2017:11:31:38.781466443 +0100] - ERR - no_diskspace - No enough space left on device (/var/lib/dirsrv/slapd-instance_name/db) (40009728 bytes); at least 145819238 bytes space is needed for db region files
```

### プラグインが書き込むログエントリ

プラグインがファイルに書き込むエントリの形式は、以下のとおりです。

```
time_stamp - severity_level - plug-in_name - function_name - message
```

以下に例を示します。

```
[24/Mar/2017:11:42:17.628363848 +0100] - ERR - NSMMReplicationPlugin - multimaster_extop_StartNSDS50ReplicationRequest - conn=19 op=3 repl="o=example.com": Excessive clock skew from supplier RUV
```

エラーログのエントリには以下の列が含まれます。

- **タイムスタンプ**: 形式はローカルの設定によって異なります。**cn=config** エントリの **nsslapd-logging-hr-timestamps-enabled** 属性で高解像度のタイムスタンプが有効になっている場合 (デフォルト)、タイムスタンプの精度はナノ秒レベルです。
- **重大度レベル**: 以下の重大度レベルが使用されます。
  - **EMERG**: サーバーが起動に失敗すると、このレベルがログに記録されます。
  - **ALERT**: サーバーに重大な問題があり、適切な対応を取る必要があります。
  - **CRIT**: 重大なエラー。
  - **ERR**: 一般エラー。

- **WARNING:** 警告メッセージ (必ずしもエラーとは言えない)。
- **NOTICE:** 正常だが、重大な状態が発生している。たとえば、これは予想される動作に対してログとして記録されます。
- **INFO:** 起動、シャットダウン、インポート、エクスポート、バックアップ、復元などの情報メッセージ。
- **DEBUG:** デバッグレベルのメッセージ。このレベルは、**Trace function calls (1)**、**Access control list processing (128)**、および **Replication (8192)** などの詳細ログレベルを使用する場合にも、デフォルトで使用されます。エラーログレベルのリストは、[表7.4「エラーログレベル」](#) を参照してください。  
重大度レベルを使用して、ログエントリーを絞り込むことができます。たとえば、重大度 **ERR** を使用するログエントリーのみを表示するには、次のコマンドを実行します。

```
# grep ERR /var/log/dirsrv/slapd-instance_name/errors
[24/Mar/2017:11:31:38.781466443 +0100] - ERR - no_diskspace - No enough space left
on device (/var/lib/dirsrv/slapd-instance_name/db) (40009728 bytes); at least
145819238 bytes space is needed for db region files
[24/Mar/2017:11:31:38.815623298 +0100] - ERR - ldbm_back_start - Failed to init
database, err=28 No space left on device
[24/Mar/2017:11:31:38.828591835 +0100] - ERR - plugin_dependency_startall - Failed
to start database plugin ldbm database
...
```

- プラグイン名: プラグインがエントリーをログに記録すると、この列にプラグインの名前が表示されます。サーバーがエントリーをログに記録した場合には、この列は表示されません。
- 関数名: 操作またはプラグインが呼び出した関数。
- メッセージ: 操作またはプラグインが返した出力。このメッセージには、LDAP のエラーコードや接続情報などの追加情報が含まれます。

### 7.2.3. 他のログレベルのエラーログコンテンツ

ログレベルが異なると、返される情報の詳細度レベルが異なるだけでなく、サーバー操作のタイプも異なります。これらの一部はここで要約されていますが、可能なロギングレベルの組み合わせは多数あります。

レプリケーションロギングは、実装する最も重要な診断レベルの1つです。このロギングレベルは、サプライヤー変更の処理および変更ログへの書き込み、更新の送信、ならびにレプリカ合意の変更など、レプリケーションおよび Windows 同期に関連するすべての操作を記録します。

レプリケーションの更新が準備または送信されるたびに、エラーログは指定されたレプリカ合意または同期合意、コンシューマーホストおよびポート、ならびに現在のレプリケーションタスクを特定します。

```
[timestamp] NSMMReplicationPlugin - agmt="name" (consumer_host:consumer_port):
current_task
```

以下に例を示します。

```
[09/Jan/2020:13:44:48 -0500] NSMMReplicationPlugin - agmt="cn=example2" (alt:13864):
{replicageneration} 4949df6e000000010000
```

**{replicageneration}** は、新しい情報が送信されることを意味します。**4949df6e000000010000** は、複製されるエントリーの変更シーケンス番号です。

例7.3 「レプリケーションエラーログエントリー」 は、changelog へのエントリーの追加からレプリケーションの完了後のコンシューマーの解放まで、単一エントリーをコンシューマーに送信する完全なプロセスを示しています。

### 例7.3 レプリケーションエラーログエントリー

```
[29/May/2017:14:15:30.539817639 +0200] - DEBUG - _csngen_adjust_local_time - gen state
before 592c103d0000:1496059964:0:1
[29/May/2017:14:15:30.562983285 +0200] - DEBUG - _csngen_adjust_local_time - gen state
after 592c10e20000:1496060129:0:1
[29/May/2017:14:15:30.578828393 +0200] - DEBUG - NSMMReplicationPlugin -
ruv_add_csn_inprogress - Successfully inserted csn 592c10e2000000020000 into pending list
[29/May/2017:14:15:30.589917123 +0200] - DEBUG - NSMMReplicationPlugin - changelog
program - _cl5GetDBFileByReplicaName - found DB object 0x558ddfe1f720 for database
/var/lib/dirsrv/slapd-supplier_2/changelogdb/d3de3e8d-446611e7-a89886da-
6a37442d_592c0e0b000000010000.db
[29/May/2017:14:15:30.600044236 +0200] - DEBUG - NSMMReplicationPlugin - changelog
program - cl5WriteOperationTxn - Successfully written entry with csn (592c10e2000000020000)
[29/May/2017:14:15:30.615923352 +0200] - DEBUG - NSMMReplicationPlugin - changelog
program - _cl5GetDBFileByReplicaName - found DB object 0x558ddfe1f720 for database
/var/lib/dirsrv/slapd-supplier_2/changelogdb/d3de3e8d-446611e7-a89886da-
6a37442d_592c0e0b000000010000.db
[29/May/2017:14:15:30.627443305 +0200] - DEBUG - NSMMReplicationPlugin -
csnplCommitALL: committing all csns for csn 592c10e2000000020000
[29/May/2017:14:15:30.632713657 +0200] - DEBUG - NSMMReplicationPlugin -
csnplCommitALL: processing data csn 592c10e2000000020000
[29/May/2017:14:15:30.652621188 +0200] - DEBUG - NSMMReplicationPlugin - ruv_update_ruv -
Successfully committed csn 592c10e2000000020000
[29/May/2017:14:15:30.669666453 +0200] - DEBUG - NSMMReplicationPlugin - repl5_inc_run -
agmt="cn=meTo_localhost:39001" (localhost:39001): State: wait_for_changes ->
wait_for_changes
[29/May/2017:14:15:30.685259483 +0200] - DEBUG - NSMMReplicationPlugin - repl5_inc_run -
agmt="cn=meTo_localhost:39001" (localhost:39001): State: wait_for_changes ->
ready_to_acquire_replica
[29/May/2017:14:15:30.689906327 +0200] - DEBUG - NSMMReplicationPlugin - conn_connect -
agmt="cn=meTo_localhost:39001" (localhost:39001) - Trying non-secure slapi_ldap_init_ext
[29/May/2017:14:15:30.700259799 +0200] - DEBUG - NSMMReplicationPlugin - conn_connect -
agmt="cn=meTo_localhost:39001" (localhost:39001) - binddn = cn=replrepl,cn=config, passwd =
{AES-
TUhNR0NTcUdTSWlZRFffFRkRUQm1NRVvHQ1NxR1NJYjNEUUVGRERBNEJDUmIZVFUzTnp
RMk55MDBaR1ZiTXpobQ0KTWkxaE9XTTRPREpoTIMwME1EaGpabVUxWmdBQ0FRSUNBU0F
3Q2dZSUtvWklodmNOQWdjd0hRWUpZSVpJQVdVRA0KQkFFcUJCRGhwMnNLcEZ2ZWE2RzEw
WG10OU41Tg==}+36owal7oTmvWhxRzUqX5w==
[29/May/2017:14:15:30.712287531 +0200] - DEBUG - NSMMReplicationPlugin -
conn_cancel_linger - agmt="cn=meTo_localhost:39001" (localhost:39001) - No linger to cancel on
the connection
[29/May/2017:14:15:30.736779494 +0200] - DEBUG - _csngen_adjust_local_time - gen state
before 592c10e20001:1496060129:0:1
[29/May/2017:14:15:30.741909244 +0200] - DEBUG - _csngen_adjust_local_time - gen state
after 592c10e30000:1496060130:0:1
[29/May/2017:14:15:30.880287041 +0200] - DEBUG - NSMMReplicationPlugin - acquire_replica -
agmt="cn=meTo_localhost:39001" (localhost:39001): Replica was successfully acquired.
[29/May/2017:14:15:30.897500049 +0200] - DEBUG - NSMMReplicationPlugin - repl5_inc_run -
```

```
agmt="cn=meTo_localhost:39001" (localhost:39001): State: ready_to_acquire_replica ->
sending_updates
[29/May/2017:14:15:30.914417773 +0200] - DEBUG - csngen_adjust_time - gen state before
592c10e30001:1496060130:0:1
[29/May/2017:14:15:30.926341721 +0200] - DEBUG - NSMMReplicationPlugin - changelog
program - _cl5GetDBFile - found DB object 0x558ddf1f720 for database /var/lib/dirsrv/slapd-
supplier_2/changelogdb/d3de3e8d-446611e7-a89886da-6a37442d_592c0e0b000000010000.db
[29/May/2017:14:15:30.943094471 +0200] - DEBUG - NSMMReplicationPlugin - changelog
program - _cl5PositionCursorForReplay - (agmt="cn=meTo_localhost:39001" (localhost:39001)):
Consumer RUV:
[29/May/2017:14:15:30.949395331 +0200] - DEBUG - NSMMReplicationPlugin -
agmt="cn=meTo_localhost:39001" (localhost:39001): {replicageneration}
592c0e0b000000010000
[29/May/2017:14:15:30.961118175 +0200] - DEBUG - NSMMReplicationPlugin -
agmt="cn=meTo_localhost:39001" (localhost:39001): {replica 1 ldap://localhost:39001}
592c0e17000000010000 592c0e1a000100010000 00000000
[29/May/2017:14:15:30.976680025 +0200] - DEBUG - NSMMReplicationPlugin -
agmt="cn=meTo_localhost:39001" (localhost:39001): {replica 2 ldap://localhost:39002}
592c103c000000020000 592c103c000000020000 00000000
[29/May/2017:14:15:30.990404183 +0200] - DEBUG - NSMMReplicationPlugin - changelog
program - _cl5PositionCursorForReplay - (agmt="cn=meTo_localhost:39001" (localhost:39001)):
Supplier RUV:
[29/May/2017:14:15:31.001242624 +0200] - DEBUG - NSMMReplicationPlugin -
agmt="cn=meTo_localhost:39001" (localhost:39001): {replicageneration}
592c0e0b000000010000
[29/May/2017:14:15:31.017406105 +0200] - DEBUG - NSMMReplicationPlugin -
agmt="cn=meTo_localhost:39001" (localhost:39001): {replica 2 ldap://localhost:39002}
592c103c000000020000 592c10e2000000020000 592c10e1
[29/May/2017:14:15:31.028803190 +0200] - DEBUG - NSMMReplicationPlugin -
agmt="cn=meTo_localhost:39001" (localhost:39001): {replica 1 ldap://localhost:39001}
592c0e1a000100010000 592c0e1a000100010000 00000000
[29/May/2017:14:15:31.040172464 +0200] - DEBUG - agmt="cn=meTo_localhost:39001"
(localhost:39001) - clcache_get_buffer - found thread private buffer cache 0x558ddf870f00
[29/May/2017:14:15:31.057495165 +0200] - DEBUG - agmt="cn=meTo_localhost:39001"
(localhost:39001) - clcache_get_buffer - _pool is 0x558ddf294d0 _pool->pl_busy_lists is
0x558ddf870f00 _pool->pl_busy_lists->bl_buffers is 0x558ddf870f00
[29/May/2017:14:15:31.063015498 +0200] - DEBUG - agmt="cn=meTo_localhost:39001"
(localhost:39001) - clcache_initial_anchorcsn - agmt="cn=meTo_localhost:39001"
(localhost:39001) - (cscb 0 - state 0) - csnPrevMax () csnMax (592c10e2000000020000) csnBuf
(592c103c000000020000) csnConsumerMax (592c103c000000020000)
[29/May/2017:14:15:31.073252305 +0200] - DEBUG - clcache_initial_anchorcsn - anchor is now:
592c103c000000020000
[29/May/2017:14:15:31.089915209 +0200] - DEBUG - NSMMReplicationPlugin - changelog
program - agmt="cn=meTo_localhost:39001" (localhost:39001): CSN 592c103c000000020000
found, position set for replay
[29/May/2017:14:15:31.095825439 +0200] - DEBUG - agmt="cn=meTo_localhost:39001"
(localhost:39001) - clcache_get_next_change - load=1 rec=1 csn=592c10e2000000020000
[29/May/2017:14:15:31.100123762 +0200] - DEBUG - NSMMReplicationPlugin -
repl5_inc_result_threadmain - Starting
[29/May/2017:14:15:31.115749709 +0200] - DEBUG - NSMMReplicationPlugin -
repl5_inc_result_threadmain - Read result for message_id 0
[29/May/2017:14:15:31.125866330 +0200] - DEBUG - NSMMReplicationPlugin - replay_update -
agmt="cn=meTo_localhost:39001" (localhost:39001): Sending add operation
(dn="cn=user,ou=People,dc=example,dc=com" csn=592c10e2000000020000)
[29/May/2017:14:15:31.142339398 +0200] - DEBUG - NSMMReplicationPlugin -
repl5_inc_result_threadmain - Read result for message_id 0
```

```
[29/May/2017:14:15:31.160456597 +0200] - DEBUG - NSMMReplicationPlugin - replay_update -
agmt="cn=meTo_localhost:39001" (localhost:39001): Consumer successfully sent operation with
csn 592c10e2000000020000
[29/May/2017:14:15:31.172399536 +0200] - DEBUG - NSMMReplicationPlugin -
repl5_inc_result_threadmain - Read result for message_id 0
[29/May/2017:14:15:31.188857336 +0200] - DEBUG - agmt="cn=meTo_localhost:39001"
(localhost:39001) - clcache_adjust_anchorcsn - agmt="cn=meTo_localhost:39001"
(localhost:39001) - (cscb 0 - state 1) - csnPrevMax (592c10e2000000020000) csnMax
(592c10e2000000020000) csnBuf (592c10e2000000020000) csnConsumerMax
(592c10e2000000020000)
[29/May/2017:14:15:31.199605024 +0200] - DEBUG - agmt="cn=meTo_localhost:39001"
(localhost:39001) - clcache_load_buffer - rc=-30988
[29/May/2017:14:15:31.210800816 +0200] - DEBUG - NSMMReplicationPlugin - send_updates -
agmt="cn=meTo_localhost:39001" (localhost:39001): No more updates to send
(cl5GetNextOperationToReplay)
[29/May/2017:14:15:31.236214134 +0200] - DEBUG - NSMMReplicationPlugin -
repl5_inc_waitfor_async_results - 0 5
[29/May/2017:14:15:31.246755544 +0200] - DEBUG - NSMMReplicationPlugin -
repl5_inc_result_threadmain - Read result for message_id 0
[29/May/2017:14:15:31.277705986 +0200] - DEBUG - NSMMReplicationPlugin -
repl5_inc_result_threadmain - Read result for message_id 0
[29/May/2017:14:15:31.303530336 +0200] - DEBUG - NSMMReplicationPlugin -
repl5_inc_result_threadmain - Read result for message_id 5
[29/May/2017:14:15:31.318259308 +0200] - DEBUG - NSMMReplicationPlugin -
repl5_inc_result_threadmain - Result 1, 0, 0, 5, (null)
[29/May/2017:14:15:31.335263462 +0200] - DEBUG - NSMMReplicationPlugin -
repl5_inc_result_threadmain - Read result for message_id 5
[29/May/2017:14:15:31.364551307 +0200] - DEBUG - NSMMReplicationPlugin -
repl5_inc_waitfor_async_results - 5 5
[29/May/2017:14:15:31.376301820 +0200] - DEBUG - NSMMReplicationPlugin -
repl5_inc_result_threadmain exiting
[29/May/2017:14:15:31.393707037 +0200] - DEBUG - agmt="cn=meTo_localhost:39001"
(localhost:39001) - clcache_return_buffer - session end: state=5 load=1 sent=1 skipped=0
skipped_new_rid=0 skipped_csn_gt_cons_maxcsn=0 skipped_up_to_date=0
skipped_csn_gt_ruv=0 skipped_csn_covered=0
[29/May/2017:14:15:31.398134114 +0200] - DEBUG - NSMMReplicationPlugin -
consumer_connection_extension_acquire_exclusive_access - conn=4 op=3 Acquired consumer
connection extension
[29/May/2017:14:15:31.423099625 +0200] - DEBUG - NSMMReplicationPlugin -
multimaster_extop_StartNSDS50ReplicationRequest - conn=4 op=3 repl="dc=example,dc=com":
Begin incremental protocol
[29/May/2017:14:15:31.438899389 +0200] - DEBUG - csngen_adjust_time - gen state before
592c10e30001:1496060130:0:1
[29/May/2017:14:15:31.443800884 +0200] - DEBUG - csngen_adjust_time - gen state after
592c10e40001:1496060130:1:1
[29/May/2017:14:15:31.454123488 +0200] - DEBUG - NSMMReplicationPlugin -
replica_get_exclusive_access - conn=4 op=3 repl="dc=example,dc=com": Acquired replica
[29/May/2017:14:15:31.469698781 +0200] - DEBUG - NSMMReplicationPlugin - release_replica -
agmt="cn=meTo_localhost:39001" (localhost:39001): Successfully released consumer
[29/May/2017:14:15:31.475096195 +0200] - DEBUG - NSMMReplicationPlugin -
conn_start_linger -agmt="cn=meTo_localhost:39001" (localhost:39001) - Beginning linger on the
connection
[29/May/2017:14:15:31.485281588 +0200] - DEBUG - NSMMReplicationPlugin - repl5_inc_run -
agmt="cn=meTo_localhost:39001" (localhost:39001): State: sending_updates ->
wait_for_changes
[29/May/2017:14:15:31.495865065 +0200] - DEBUG - NSMMReplicationPlugin -
```



```

multimaster_extop_StartNSDS50ReplicationRequest - conn=4 op=3 repl="dc=example,dc=com":
StartNSDS90ReplicationRequest: response=0 rc=0
[29/May/2017:14:15:31.501617765 +0200] - DEBUG - NSMMReplicationPlugin -
consumer_connection_extension_relinquish_exclusive_access - conn=4 op=3 Relinquishing
consumer connection extension
[29/May/2017:14:15:31.716627741 +0200] - DEBUG - NSMMReplicationPlugin -
consumer_connection_extension_acquire_exclusive_access - conn=4 op=4 Acquired consumer
connection extension
[29/May/2017:14:15:31.735431913 +0200] - DEBUG - NSMMReplicationPlugin -
replica_relinquish_exclusive_access - conn=4 op=4 repl="dc=example,dc=com": Released replica
held by locking_purl=conn=4 id=3
[29/May/2017:14:15:31.745841821 +0200] - DEBUG - NSMMReplicationPlugin -
consumer_connection_extension_relinquish_exclusive_access - conn=4 op=4 Relinquishing
consumer connection extension

```

プラグインのロギングは、プラグインの名前およびプラグインによって呼び出されるすべての関数を記録します。形式は単純です。

**[timestamp] Plugin\_name - message**

**[timestamp] - function - message**

すべてのステップが処理されるので、返される情報は数百行になる可能性があります。記録される正確な情報は、プラグイン自体により異なります。たとえば、ACL プラグインには、[例7.4「プラグインロギングを使用した ACL プラグインエラーログエントリーの例」](#)に示されるように接続および操作番号が含まれます。

#### 例7.4 プラグインロギングを使用した ACL プラグインエラーログエントリーの例

```

[29/May/2017:14:38:19.133878244 +0200] - DEBUG - get_filter_internal - ==>
[29/May/2017:14:38:19.153942547 +0200] - DEBUG - get_filter_internal - PRESENT
[29/May/2017:14:38:19.177908064 +0200] - DEBUG - get_filter_internal - <= 0
[29/May/2017:14:38:19.193547449 +0200] - DEBUG - slapi_vattr_filter_test_ext_internal - =>
[29/May/2017:14:38:19.198121765 +0200] - DEBUG - slapi_vattr_filter_test_ext_internal - <=
[29/May/2017:14:38:19.214342752 +0200] - DEBUG - slapi_vattr_filter_test_ext_internal -
PRESENT
[29/May/2017:14:38:19.219886104 +0200] - DEBUG - NSACLPlugin - acl_access_allowed -
conn=15 op=1 (main): Allow search on entry(cn=replication,cn=config): root user
[29/May/2017:14:38:19.230152526 +0200] - DEBUG - slapi_vattr_filter_test_ext_internal - <= 0
[29/May/2017:14:38:19.240971955 +0200] - DEBUG - NSACLPlugin -
acl_read_access_allowed_on_entry - Root access (read) allowed on
entry(cn=replication,cn=config)
[29/May/2017:14:38:19.246456160 +0200] - DEBUG - cos-plugin - cos_cache_vattr_types -
Failed to get class of service reference
[29/May/2017:14:38:19.257200851 +0200] - DEBUG - NSACLPlugin - Root access (read) allowed
on entry(cn=replication,cn=config)
[29/May/2017:14:38:19.273534025 +0200] - DEBUG - NSACLPlugin - Root access (read) allowed
on entry(cn=replication,cn=config)
[29/May/2017:14:38:19.289474926 +0200] - DEBUG - slapi_filter_free - type 0x87

```



## 注記

例7.4「プラグインロギングを使用した ACL プラグインエラーログエントリーの例」は、プラグインロギングと検索フィルター処理の両方を示しています (ログレベル 65696)。

他の多くのロギングで、プラグインロギングレベルと類似した情報を出力しますが、内部操作だけは異なります。ヘビートレース出力 (4)、アクセス制御リスト処理 (128)、スキーマ解析 (2048)、およびハウスキーピング (4096) は、すべて実行されるさまざまな操作によって呼び出された関数を記録します。この場合、異なるのは記録される内容の形式ではなく、記録対象の操作です。

サーバーが起動するたびに、すべての **.conf** 設定ファイルが処理され、すべての行を出力します。これは、サーバーの通常の設定以外のファイルの問題をデバッグするために使用できます。デフォルトでは、国際言語セットの設定が含まれる **slapd-collations.conf** ファイルのみが利用可能です。

### 例7.5 設定ファイル処理ログエントリー

```
[29/May/2017:15:26:48.897935879 +0200] - DEBUG - collation_read_config - Reading config file
/etc/dirsrv/slapd-supplier_1/slapd-collations.conf
[29/May/2017:15:26:48.902606586 +0200] - DEBUG - collation-plugin - collation_read_config -
line 16: collation "" "" "" 1 3 2.16.840.1.113730.3.3.2.0.1 default
[29/May/2017:15:26:48.918493657 +0200] - DEBUG - collation-plugin - collation_read_config -
line 17: collation ar "" "" 1 3 2.16.840.1.113730.3.3.2.1.1 ar
[29/May/2017:15:26:48.932550086 +0200] - DEBUG - collation-plugin - collation_read_config -
line 18: collation be "" "" 1 3 2.16.840.1.113730.3.3.2.2.1 be be-BY
...
```

ACI ロギングには、デバッグ情報用と概要用の 2 つのレベルがあります。これらの ACI ロギングレベルはどちらも、[接続番号](#) および [操作番号](#) 情報など、他のタイプのプラグインまたはエラーロギングには含まれない追加情報を記録します。プラグインの名前、ユーザーのバインド DN、実行した/試みた操作、および適用された ACI を表示します。デバッグレベルは、バインドや他の操作において呼び出される一連の関数も表示します。

例7.6「アクセス制御の概要ロギング」は、要約アクセス制御ログエントリーを示しています。

### 例7.6 アクセス制御の概要ロギング

```
[29/May/2017:15:34:52.742034888 +0200] - DEBUG - NSACLPlugin - acllist_init_scan - Failed to
find root for base: cn=features,cn=config
[29/May/2017:15:34:52.761702767 +0200] - DEBUG - NSACLPlugin - acllist_init_scan - Failed to
find root for base: cn=config
[29/May/2017:15:34:52.771907825 +0200] - DEBUG - NSACLPlugin - acl_access_allowed - #####
conn=6 op=1 binddn="cn=user,ou=people,dc=example,dc=com"
[29/May/2017:15:34:52.776327012 +0200] - DEBUG - NSACLPlugin - ***** RESOURCE
INFO STARTS *****
[29/May/2017:15:34:52.786397852 +0200] - DEBUG - NSACLPlugin - Client DN:
cn=user,ou=people,dc=example,dc=com
[29/May/2017:15:34:52.797004451 +0200] - DEBUG - NSACLPlugin - resource
type:256(search target_DN )
[29/May/2017:15:34:52.807135945 +0200] - DEBUG - NSACLPlugin - Slapi_Entry DN:
cn=features,cn=config
[29/May/2017:15:34:52.822877838 +0200] - DEBUG - NSACLPlugin - ATTR: objectClass
[29/May/2017:15:34:52.827250828 +0200] - DEBUG - NSACLPlugin - rights:search
[29/May/2017:15:34:52.831603634 +0200] - DEBUG - NSACLPlugin - ***** RESOURCE
```

```
INFO ENDS *****
```

```
[29/May/2017:15:34:52.847183276 +0200] - DEBUG - NSACLPlugin - acl__scan_for_acis - Num
of ALLOW Handles:0, DENY handles:0
[29/May/2017:15:34:52.857857195 +0200] - DEBUG - NSACLPlugin -
print_access_control_summary - conn=6 op=1 (main): Deny search on
entry(cn=features,cn=config).attr(objectClass) to cn=user,ou=people,dc=example,dc=com: no aci
matched the resource
```

### 7.3. 監査ログリファレンス

監査ログは、サーバーインスタンスに **加えられた変更** を記録します。エラーログやアクセスログとは異なり、監査ログはサーバーインスタンスへの **アクセス** を記録しないため、データベースに対する検索はログに記録されません。

監査ログの形式はアクセスログやエラーログとは異なり、タイムスタンプ付きの LDIF ファイルに似ています。監査ログに記録される操作は LDIF ステートメントとしてフォーマットされます。

```
timestamp: date
dn: modified_entry
changetype: action
action:attribute
attribute:new_value
-
replace: modifiersname
modifiersname: dn
-
replace: modifytimestamp
modifytimestamp: date
-
```

LDIF ファイルと形式の詳細は、**管理ガイド**の LDAP データ交換形式付録を参照してください。

例7.7「監査ログコンテンツ」には、さまざまな種類の監査エントリーが表示されています。

#### 例7.7 監査ログコンテンツ

```
... modifying an entry ...
time: 20200108181429
dn: uid=scarter,ou=people,dc=example,dc=com
changetype: modify
replace: userPassword
userPassword: {SHA}8EcJhJolgBgY/E5j8JiVoj6W3BLyj9Za/rCPOw==
-
replace: modifiersname
modifiersname: cn=Directory Manager
-
replace: modifytimestamp
modifytimestamp: 20200108231429Z
-

... sending a replication update ...
time: 20200109131811
dn: cn=example2,cn=replica,cn="dc=example,dc=com",cn=mapping tree,cn=config
```

```

changetype: modify
replace: nsds5BeginReplicaRefresh
nsds5BeginReplicaRefresh: start
-
replace: modifiersname
modifiersname: cn=Directory Manager
-
replace: modifytimestamp
modifytimestamp: 20200109181810Z
-

```

監査ログのフォーマットやログレベルの設定はできないことに注意してください。

## 7.4. LDAP の結果コード

Directory Server は以下の LDAP 結果コードを使用します。

表7.5 LDAP の結果コード

10 進数値	16 進値	定数
0	0x00	LDAP_SUCCESS
1	0x01	LDAP_OPERATIONS_ERROR
2	0x02	LDAP_PROTOCOL_ERROR
3	0x03	LDAP_TIMELIMIT_EXCEEDED
4	0x04	LDAP_SIZELIMIT_EXCEEDED
5	0x05	LDAP_COMPARE_FALSE
6	0x06	LDAP_COMPARE_TRUE
7	0x07	LDAP_AUTH_METHOD_NOT_SUPPORTED
LDAP_STRONG_AUTH_NOT_SUPPORTED		
8	0x08	LDAP_STRONG_AUTH_REQUIRED
9	0x09	LDAP_PARTIAL_RESULTS
10	0x0a	LDAP_REFERRAL [a]
11	0x0b	LDAP_ADMINLIMIT_EXCEEDED

10 進数値	16 進値	定数
12	0x0c	LDAP_UNAVAILABLE_CRITICAL_EXTENSION
13	0x0d	LDAP_CONFIDENTIALITY_REQUIRED
14	0x0e	LDAP_SASL_BIND_IN_PROGRESS
16	0x10	LDAP_NO_SUCH_ATTRIBUTE
17	0x11	LDAP_UNDEFINED_TYPE
18	0x12	LDAP_INAPPROPRIATE_MATCHING
19	0x13	LDAP_CONSTRAINT_VIOLATION
20	0x14	LDAP_TYPE_OR_VALUE_EXISTS
21	0x15	LDAP_INVALID_SYNTAX
32	0x20	LDAP_NO_SUCH_OBJECT
33	0x21	LDAP_ALIAS_PROBLEM
34	0x22	LDAP_INVALID_DN_SYNTAX
35	0x23	LDAP_IS_LEAF <sup>[b]</sup>
36	0x24	LDAP_ALIAS_DEREF_PROBLEM
48	0x30	LDAP_INAPPROPRIATE_AUTH
49	0x31	LDAP_INVALID_CREDENTIALS
50	0x32	LDAP_INSUFFICIENT_ACCESS
51	0x33	LDAP_BUSY
52	0x34	LDAP_UNAVAILABLE
53	0x35	LDAP_UNWILLING_TO_PERFORM

10 進数値	16 進値	定数
54	0x36	LDAP_LOOP_DETECT
60	0x3c	LDAP_SORT_CONTROL_MISSING
61	0x3d	LDAP_INDEX_RANGE_ERROR
64	0x40	LDAP_NAMING_VIOLATION
65	0x41	LDAP_OBJECT_CLASS_VIOLATION
66	0x42	LDAP_NOT_ALLOWED_ON_NONLEAF
67	0x43	LDAP_NOT_ALLOWED_ON_RDN
68	0x44	LDAP_ALREADY_EXISTS
69	0x45	LDAP_NO_OBJECT_CLASS_MODS
70	0x46	LDAP_RESULTS_TOO_LARGE [c]
71	0x47	LDAP_AFFECTS_MULTIPLE_DSAS
76	0x4C	LDAP_VIRTUAL_LIST_VIEW_ERROR
80	0x50	LDAP_OTHER
81	0x51	LDAP_SERVER_DOWN
82	0x52	LDAP_LOCAL_ERROR
83	0x53	LDAP_ENCODING_ERROR
84	0x54	LDAP_DECODING_ERROR
85	0x55	LDAP_TIMEOUT
86	0x56	LDAP_AUTH_UNKNOWN

10 進数値	16 進値	定数
87	0x57	LDAP_FILTER_ERROR
88	0x58	LDAP_USER_CANCELLED
89	0x59	LDAP_PARAM_ERROR
90	0x5A	LDAP_NO_MEMORY
91	0x5B	LDAP_CONNECT_ERROR
92	0x5C	LDAP_NOT_SUPPORTED
93	0x5D	LDAP_CONTROL_NOT_FOUND
94	0x5E	LDAP_MORE_RESULTS_TO_RETURN
95	0x5F	LDAP_MORE_RESULTS_TO_RETURN
96	0x60	LDAP_CLIENT_LOOP
97	0x61	LDAP_REFERRAL_LIMIT_EXCEEDED
118	0x76	LDAP_CANCELLED
<p>[a] LDAPv3</p> <p>[b] LDAPv3 では使用されない</p> <p>[c] CLDAP に予約済み</p>		

## 7.5. ログファイルの名前付きパイプへの置き換え

多くの管理者は、特定のイベントのみを記録するようにアクセスログを設定するなど、ロギングデータに関して特別な設定や操作を行う必要があります。標準の Directory Server ログファイル設定属性を使用してこれを行うことはできませんが、ログデータを名前付きパイプに送信し、別のスクリプトを使用してデータを処理することで可能になります。ログに名前付きパイプを使用すると、以下のような特別なタスクが簡素化されます。

- 特定のユーザーまたは IP アドレスからのバインドの試みや接続の失敗など、特定イベントをログに記録する
- 特定の正規表現パターンにマッチするエントリをログに記録する

- ログを特定の期間保持する (最後の行番号のみをログに記録する)
- イベント発生時にメールなどの通知を送信する

ログファイルをパイプに置き換えると、特に操作率が高いサーバーではパフォーマンスが向上します。

ログバッファでデータ処理方法により、名前付きパイプは、スクリプトを使用してログからのデータを抽出することとは異なります。

ログがバッファされると、サーバーのパフォーマンス的には良好ですが、重要なデータがイベントの発生直後にディスク (ログファイル) に書き込まれません。サーバーでクラッシュの問題がある場合は、データがディスクに書き込まれる前にクラッシュし、スクリプトが抽出するためのデータがないことがあります。

ログがバッファされない場合<sup>[1]</sup>、書き込みは各操作でディスクにフラッシュされるため、ディスク I/O とパフォーマンスが大幅に低下します。

ログディスクファイルをパイプに置き換えると、バッファに利点があります。これは、パイプから読み取るスクリプトがメモリー内に受信ログデータをバッファできるためです (単純なスクリプトでは不可能)。

スクリプトの使用方法およびオプションの詳細は、「[ds-logpipe.py](#)」で説明されています。基本的な形式は、`ds-logpipe.py/path/to/named_pipe--userpipe_user--maxlinesnumber--serverpidfilefile.pid--serverpidPID--servertimeoutseconds--plugin=/path/to/plugin.pypluginfile.arg=value` です。

### 7.5.1. ロギングへの名前付きパイプの使用

Directory Server インスタンスは、名前付きパイプログスクリプトを実行し、パイプの名前を指定するだけで、ロギングに名前付きパイプを使用できます。(サーバーがすでに実行中の場合は、ログを再度開く必要がありますが、それ以外の設定は必要ありません)。

```
# ds-logpipe.py /var/log/dirsrv/slapd-example/access
```

このように **ds-logpipe.py** を実行することには、実装が簡単であり、Directory Server 設定を変更する必要がない、という利点があります。これは、特に特定のイベントタイプを検索する場合に、高速のデバッグまたはモニタリングに役立ちます。

Directory Server インスタンスがロギング用に実際のファイルではなく名前付きパイプを頻繁に、または永続的に使用する場合には、(デフォルトでログファイルを使用するのと同じように) 名前付きパイプを作成してロギングに使用するようにインスタンスを再設定することができます。

インスタンスのログ設定には、3つの項目を設定する必要があります。

- 使用するログファイルをパイプに変更しなければなりません (**nsslapd-\*log**: ここで、\*は設定されるログタイプに応じて `access`、`error`、または `audit`<sup>[2]</sup>のいずれかです)。
- スクリプトがログエントリをバッファするため、バッファを無効にする必要があります (**nsslapd-\*log-logbuffering**)。
- サーバーが名前付きパイプのローテーションを試行しないように、ログローテーションを無効にする必要があります (**nsslapd-\*log-maxlogsperdir**、**nsslapd-\*log-logexpiriationtime**、および **nsslapd-\*log-logrotationtime**)。

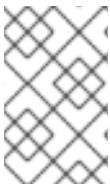
これらの設定変更は、Directory Server Console で行うか、**ldapmodify** を使用して実行できます。

たとえば、これによりアクセスログが **access.pipe** に切り替わります。



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

dn: cn=config
changetype: modify
replace: nsslapd-accesslog
nsslapd-accesslog: /var/log/dirsrv/slapd-instance/access.pipe
-
replace: nsslapd-accesslog-logbuffering
nsslapd-accesslog-logbuffering: off
-
replace: nsslapd-accesslog-maxlogsperdir
nsslapd-accesslog-maxlogsperdir: 1
-
replace: nsslapd-accesslog-logexpirationtime
nsslapd-accesslog-logexpirationtime: -1
-
replace: nsslapd-accesslog-logrotationtime
nsslapd-accesslog-logrotationtime: -1
```



### 注記

これらの変更を行うと、サーバーは現在のログファイルを閉じ、即座に名前付きパイプに切り替えます。これは、実行中のサーバーのデバッグや、特定のメッセージのログ出力を移動するのに非常に便利です。

## 7.5.2. サーバーで名前付きパイプを起動する

インスタンスの init スクリプト設定ファイルを編集して、名前付きパイプを Directory Server インスタンスと共に起動およびシャットダウンできます。



### 注記

名前付きパイプスクリプトは、サーバーの起動時に呼び出す前にインスタンスの **dse.ldif** ファイルで明示的に設定する必要があります。

1. サーバースステムのインスタンス設定ファイルを開きます。

```
/etc/sysconfig/dirsrv-instance_name
```



### 警告

**/etc/sysconfig/dirsrv** ファイルを編集 **しない** ください。

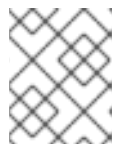
2. ファイルの末尾に、以下という行があります。

```
# Put custom instance specific settings below here.
```

その行の下に **ds-logpipe.py** コマンドを挿入し、サーバーの起動時に起動します。以下に例を示します。

```
# only keep the last 1000 lines of the error log
python /usr/bin/ds-logpipe.py /var/log/dirsrv/slapd-example/errors.pipe -m 1000 -u dirsrv -s
/var/run/dirsrv/slapd-example.pid > /var/log/dirsrv/slapd-example/errors &

# only log failed binds
python /usr/bin/ds-logpipe.py /var/log/dirsrv/slapd-example/access.pipe -u dirsrv -s
/var/run/dirsrv/slapd-example.pid --plugin=/usr/share/dirsrv/data/failedbinds.py
failedbinds.logfile=/var/log/dirsrv/slapd-example/access.failedbinds &
```



### 注記

**-s** オプションは、どちらも PID を書き込むサーバー用の .pid ファイルを指定し、サーバープロセスで起動および停止するようにスクリプトを設定します。

## 7.5.3. 名前付きパイプログでのプラグインの使用

プラグインは、名前付きパイプからログデータを読み取り、操作を実行するために呼び出すことができます。名前付きパイプログスクリプトによりプラグインを使用する場合、いくつかの考慮事項があります。

- プラグイン関数は、名前付きパイプから読み取られる各行に対して呼び出されます。
- プラグイン関数は Python スクリプトで、**.py** で終了する必要があります。
- プラグインの引数は、コマンドラインで名前付きパイプログスクリプトに渡されます。
- プラグインの読み込みタイミングに関して、事前操作関数を指定できます。
- スクリプトの終了タイミングに関して、事後操作関数を呼び出すことができます。

### 7.5.3.1. 名前付きパイプログスクリプトを使用したプラグインの読み込み

プラグインに使用する **ds-logpipe.py** には、以下の 2 つのオプションがあります。

- **--plugin** オプションはプラグインファイル (Python スクリプトで、**.py** で終了する必要があります) へのパスを提供します。
- **plugin.arg** オプションは、プラグイン引数を名前付きパイプログスクリプトに渡します。プラグインファイル名 (**.py** 拡張子なし) は **plugin** で、そのプラグインで許可される任意の引数は **arg** です。

以下に例を示します。

```
ds-logpipe.py /var/log/dirsrv/slapd-example/errors.pipe --plugin=/usr/share/dirsrv/data/example-funct.py example-funct.regex="warning" > warnings.txt
```

同じ引数に複数の値を渡すと、それらはプラグインディクショナリーの値のリストに変換されます。たとえば、このスクリプトでは、**arg1** に 2 つの値を指定します。

```
--plugin=/path/to/pluginname.py pluginname.arg1=foo pluginname.arg1=bar pluginname.arg2=baz
```

プラグインでは、これは次のように変換されます。

```
{'arg1': ['foo', 'bar'],
 'arg2': 'baz'}
```

これは、2つのキーを持つ Python **dict** オブジェクトです。最初のキーは文字列 **arg1** で、その値は文字列 **foo** と **bar** の2つの要素を持つ Python リストオブジェクトです。2つ目のキーは文字列 **arg2** で、その値は文字列 **baz** です。引数に値が1つしかない場合は、単純な文字列のままです。1つの引数名に対する複数の値が、文字列のリストに変換されます。

### 7.5.3.2. 名前付きパイプログスクリプトで使用するプラグインの作成

**ds-logpipe.py** コマンドは、どのプラグインでも最大3つの関数 (**plugin ()**)、**pre ()**、および **post ()** ) があることを想定します。

**ds-logpipe.py** コマンドで使用されるすべてのプラグインは、**plugin** 関数を指定する必要があります。

**plugin ()** 関数はログデータ内のすべての行に対して実行されますが、**pre ()** および **post ()** 関数は、それぞれスクリプトの開始および停止時に実行されます。

各関数に引数を定義でき、これらの引数は **plugin.arg** オプションを使用してスクリプトに渡すことができます。さらに、各関数には独自の戻り値とアクションを定義できます。

#### 例7.8 単純な名前付き Pipe ログプラグイン

```
def pre(myargs):
    retval = True
    myarg = myargs['argname']
    if isinstance(myarg, list): # handle list of values
    else: # handle single value
    if bad_problem:
        retval = False
    return retval

def plugin(line):
    retval = True
    # do something with line
    if something_is_bogus:
        retval = False
    return retval

def post(): # no arguments
    # do something
    # no return value
```

[1] アクセスログでログバッファリングが無効になっている場合、エラーログでログレベルが変更されている場合、または監査ログでログバッファリングが無効になっている場合、サーバーのパフォーマンスが低下します。

[2] 監査ログはデフォルトでは有効になっていないため、名前付きパイプを使用して置き換える前に、このログを有効にする必要があります。

## 第8章 設定ファイルのリファレンス

設定するほとんどのディレクトリーサーバー機能は、ディレクトリーの **cn=config** エントリーにあります。ただし、特定の機能については、Directory Server は設定ファイルから設定を読み取ります。本章では、これらのファイルとその設定について説明します。

### 8.1. CERTMAP.CONF

証明書ベースの認証を設定する場合は、`/etc/dirsrv/slapd-instance_name/certmap.conf` ファイルが Directory Server が証明書をユーザーエントリーに動的にマッピングする方法を管理します。

`/etc/dirsrv/slapd-instance_name/certmap.conf` ファイルは以下のフォーマットを使用します。

```
certmap alias_name      certificate_issuer_DN
alias_name:parameter_name  value
```

それぞれの証明書発行者識別名 (DN) に個別の設定を指定できます。別の設定を持たない発行者 DN の場合、**default** エントリーの設定が使用されます。以下は、**default** エントリーに必要な最小設定です。

```
certmap default  default
```

また、**default** エントリーに利用可能なすべてのパラメーターを設定することができます。Directory Server は、パラメーターが発行者 DN の個別設定で指定されていない場合、それらを使用します。

#### 例8.1 default エントリーおよび特定の発行者 DN の設定

以下の設定では、**o=Example Inc.,c=US** 発行者 DN が設定された証明書の個別設定を設定します。他の証明書は、**default** エントリーの設定を使用します。

```
certmap default      default
default:DNComps      dc
default:FilterComps  mail, cn
default:VerifyCert   on

certmap example      o=Example Inc.,c=US
example:DNComps
```

以下のパラメーターを設定できます。

#### DNComps

**DNComps** パラメーターは、Directory Server がディレクトリー内のユーザーを検索するために使用されるベース DN を生成する方法を決定します。

- 証明書の **subject** フィールドの属性がベース DN と一致する場合は、**DNComps** パラメーターをこれらの属性に設定します。複数の属性はコンマで区切ります。ただし、**DNComps** パラメーターの属性の順序は、証明書のサブジェクトの順番と一致している必要があります。たとえば、証明書の発行先が **e=user\_name@example.com,cn=user\_name,o=Example Inc.,c=US** で、ユーザーを検索する際のベース DN として Directory Server が **cn=user\_name,o=Example Inc.,c=US** を使用する場合は、**DNComps** パラメーターを **cn, o, c** に設定します。



## 重要

**DNComps** パラメーターに設定された属性の値は、データベース内で一意でなければなりません。

- 証明書の **subject** フィールドからベース DN を生成できない場合は、パラメーターを空の値に設定します。この場合、Directory Server は **FilterComps** パラメーターの設定から生成されたフィルターを使用してディレクトリー全体でユーザーを検索します。  
たとえば、証明書の発行先が **e=user\_name@example.com,cn=user\_name,o=Example Inc.,c=US** だが、Directory Server はデータを **dc=example,dc=com** エントリーに保存する場合、必要なコンポーネントが発行先の一部ではないため、Directory Server は証明書の発行先から有効なベース DN を生成することができません。この場合は、**DNComps** を空の文字列に設定し、ディレクトリー全体でユーザーを検索します。
- 証明書の **subject** フィールドのいずれかが Directory Server のユーザーの DN に完全に一致する場合や、**CmapLdapAttr** パラメーターの設定を使用する場合は、このパラメーターをコメントアウトするか、設定しないでください。  
または、ハードコードされたベース DN を使用するように、**cn=config** エントリーで **nsslapd-certmap-basedn** パラメーターを設定します。

## FilterComps

このパラメーターは、Directory Server が使用する証明書の **subject** フィールドからの属性を設定し、ユーザーの検索に使用する検索フィルターを生成します。

- このパラメーターを、証明書のサブジェクトで使用される属性のコンマ区切りリストに設定します。Directory Server はこれらの属性をフィルター内の **AND** 操作で使用します。



## 注記

証明書サブジェクトは、メールアドレスにデフォルトの Directory Server スキーマには存在しない **e** 属性を使用します。このため、Directory Server は自動的にこの属性を **mail** 属性にマッピングします。つまり、**FilterComps** パラメーターで **mail** 属性を使用すると、Directory Server は証明書のサブジェクトから **e** 属性の値を読み取ります。

たとえば、証明書の発行先が

**e=user\_name@example.com,cn=user\_name,dc=example,dc=com,o=Example Inc.,c=US** で、**(&(mail=username@domain)(cn=user\_name))** フィルターを動的に生成する場合は、**FilterComps** パラメーターを **mail,cn** に設定します。

- パラメーターがコメントアウトまたは空の値に設定されていると、**(objectclass=\*)** フィルターが使用されます。

## verifycert

Directory Server は、証明書が信頼できる認証局 (CA) によって発行されたかどうかを常に検証します。ただし、追加で **verifycert** パラメーターを **on** に設定すると、Directory Server は、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 DER (Distinguished Encoding Rules) 形式の証明書と一致することを確認します。

このパラメーターを設定しないと、**verifycert** は無効になります。

## CmapLdapAttr

ユーザーエントリーに、ユーザー証明書の発行先 DN を保存する属性が含まれている場合は、**CmapLdapAttr** をこの属性名に設定します。Directory Server はこの属性およびサブジェクト DN

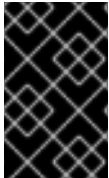
を使用してユーザーを検索します。この場合、**FilterComps** パラメーターの属性に基づいてフィルターが生成されません。

### library

共有ライブラリーまたは動的リンクライブラリー (DLL) ファイルへのパス名を設定します。この設定は、証明書 API を使用して独自のプロパティーを作成する場合にのみ使用します。このパラメーターは非推奨となっており、今後のリリースで削除されます。

### InitFn

カスタムライブラリーを使用する場合、**init** 関数の名前を設定します。この設定は、証明書 API を使用して独自のプロパティーを作成する場合にのみ使用します。このパラメーターは非推奨となっており、今後のリリースで削除されます。



### 重要

Directory Server がマッチするユーザーを検索する場合、その検索はエントリーを1つだけ返す必要があります。検索が複数のエントリーを返すと、Directory Server は **multiple matches** エラーをログに記録し、認証に失敗します。

詳細は、[Directory Server 管理ガイド](#)の該当するセクションを参照してください。

## 第9章 コマンドラインユーティリティ

本章では、Red Hat Directory Server (Directory Server) で使用されるコマンドラインユーティリティに関する参考情報を紹介します。これらのコマンドラインユーティリティを使用すると、Directory Server での管理タスクの実施が容易になります。

### 9.1. DS-REPLCHECK

**ds-replcheck** ユーティリティは、2つの Directory Server インスタンスまたは LDIF 形式のファイルと比較して、同期されているかどうかを特定します。詳細は、[Red Hat Directory Server 管理ガイド](#) の2つの Directory Server インスタンスの比較セクションを参照してください。

構文およびコマンドラインオプションの詳細は、**ds-replcheck(1)** の man ページを参照してください。

### 9.2. LDIF

**ldif** は LDIF ファイルを自動的にフォーマットし、base-64 でエンコードされた属性値を作成します。base-64 エンコーディングでは、LDIF で JPEG イメージなどのバイナリーデータを表現できます。base-64 でエンコードされたデータは、二重コロンの (::) 記号を使用して表されます。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他の値は、以下を含む他のシンボルで特定できます。

- スペースで始まる値。
- 1つのコロン (:) で始まる値。
- ASCII 以外のデータを含む値 (新しい行を含む)。

**ldif** コマンドラインユーティリティは入力を取得し、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** ユーティリティは、入力に base-64 エンコーディングが必要かどうかを評価します。

構文およびコマンドラインオプションの詳細は、**ldif(5)** の man ページを参照してください。

### 9.3. DBSCAN

**dbscan** ツールは、Directory Server データベースファイルから情報を分析して抽出します。**dbscan** でスキャンできるデータベースファイルは4種類あります。

- **id2entry.db** (ユーザーデータベースの主なデータベースファイル)
- **entryrdn.db** (ユーザーデータベース)
- **cn.db** などのユーザーデータベースのセカンダリーインデックスファイル
- **numeric\_string.db** (`/var/lib/dirsrv/slaped-instance/changelogdb` の変更ログ)

データベースファイルの詳細は、「[データベースファイル](#)」を参照してください。

データベースファイルは、Directory Server のバージョンに応じて、ファイル名の拡張子 **.db2**、**.db3**、**.db4**、および **.db** を使用します。

構文およびコマンドラインオプションの詳細は、**dbscan(1)** の man ページを参照してください。

## 例

以下は、**dbscan** を使用して Directory Server データベースを調べるさまざまな状況のコマンドラインの例です。

### 例9.1 エントリーファイルのダンプ

```
dbscan -f /var/lib/dirsrv/slapd-instance/db/userRoot/id2entry.db
```

### 例9.2 cn.db のインデックスキーの表示

```
dbscan -f /var/lib/dirsrv/slapd-instance/db/userRoot/cn.db
```

### 例9.3 mail.db のインデックスキーおよびキーを持つエントリーの数の表示

```
# dbscan -r -f /var/lib/dirsrv/slapd-instance/db/userRoot/mail.db
```

### 例9.4 sn.db のインデックスキーおよび 20 を超える ID を持つすべての ID の表示

```
# dbscan -r -G 20 -f /var/lib/dirsrv/slapd-instance/db/userRoot/sn.db
```

### 例9.5 objectclass.db の概要の表示

```
# dbscan -s -f /var/lib/dirsrv/slapd-instance/db/userRoot/objectclass.db
```

### 例9.6 VLV インデックスファイルの内容の表示

```
# dbscan -r -f /var/lib/dirsrv/slapd-instance/db/userRoot/vlv#bymccoupeoledcpeopleddccom.db
```

### 例9.7 変更ログファイルの内容の表示

```
# dbscan -f /var/lib/dirsrv/slapd-instance/changelogdb/c1a2fc02-1d11b2-8018afa7-  
fdce000_424c8a000f00.db
```

### 例9.8 Raw モードでのインデックスファイル uid.db のダンプ

```
# dbscan -R -f /var/lib/dirsrv/slapd-instance/db/userRoot/uid.db
```



**例9.9 共通名キー=hr manager を使用した entryID の表示**

この例では、共通名キーは **=hr managers** で、等号 (=) はキーが等価インデックスであることを示します。

```
# dbscan -k "=hr managers" -r -f /var/lib/dirsrv/slapd-instance/db/userRoot/cn.db
=hr%20managers 7
```

**例9.10 エントリー ID が 7 のエントリーの表示**

```
# dbscan -K 7 -f /var/lib/dirsrv/slapd-instance/db/userRoot/id2entry.db

id 7 dn: cn=HR Managers,ou=groups,dc=example,dc=com
objectClass: top
objectClass: groupOfUniqueNames
cn: HR Manager
ou: groups
description: People who can manage HR entries
creatorsName: cn=Directory Manager
modifiersName: cn=Directory Manager
createTimestamp: 20050408230424Z
modifyTimestamp: 20050408230424Z
nsUniqueId: 8b465f73-1dd211b2-807fd340-d7f40000 parentid: 3
entryid: 7
entrydn: cn=hr managers,ou=groups,dc=example,dc=com
```

**例9.11 entryrdn インデックスの内容の表示**

```
# dbscan -f /var/lib/dirsrv/slapd-instance/db/userRoot/entryrdn.db -k "dc=example,dc=com"

dc=example,dc=com
ID: 1; RDN: "dc=example,dc=com"; NRDN: "dc=example,dc=com"
C1:dc=example,dc=com
ID: 2; RDN: "cn=Directory Administrators"; NRDN: "cn=directory administrators"
2:cn=directory administrators
ID: 2; RDN: "cn=Directory Administrators"; NRDN: "cn=directory administrators"
P2:cn=directory administrators
ID: 1; RDN: "dc=example,dc=com"; NRDN: "dc=example,dc=com"
C1:dc=example,dc=com
ID: 3; RDN: "ou=Groups"; NRDN: "ou=groups"
3:ou=groups
ID: 3; RDN: "ou=Groups"; NRDN: "ou=groups"
[...]
```

**9.4. DS-LOGPIPE.PY**

名前付きパイプログスクリプトは、Directory Server の任意のログファイル (access、errors、および audit) を名前付きパイプに置き換えることができます。このパイプは別のスクリプトに加えることがで

きます。このパイプは、特定のパターンに一致する行や特定のイベントタイプにのみ記述するなど、出力に送信する前にログデータを処理できます。

名前付きパイプスクリプトを使用すると、柔軟性が得られます。

- エラーログレベルを問題の診断用に非常に高く設定すると、パフォーマンスへの影響を抑えて最後の数 100 または数 1000 のログメッセージだけのログを作成できます。
- メッセージをフィルターして、特定のイベントのみを保持することができます。たとえば、名前付きパイプスクリプトはアクセスログで失敗した BIND 試行のみを記録し、他のイベントは破棄されます。
- このスクリプトは、ユーザーエントリーの追加や削除などのイベントが発生した時、または特定のエラーが発生した時などに、通知を送信するために使用できます。

構文およびコマンドラインオプションの詳細は、`ds-logpipe.py(1)` の man ページを参照してください。

## 例

名前付きパイプロギング用にサーバーを設定する手順は、「[ログファイルの名前付きパイプへの置き換え](#)」で説明されています。

名前付きパイプログスクリプトの最も基本的な使用方法は、名前付きパイプのみを参照するものです。

### 例9.12 基本的な名前付きパイプログスクリプト

```
# ds-logpipe.py /var/log/dirsrv/slapd-example/errors.pipe
```



#### 注記

スクリプトが終了すると (処理が完了したか、SIGTERM または Ctrl+C で終了したかのいずれか)、スクリプトにより、エラーログの最後の 1000 行が標準出力にダンプされません。

スクリプトはバックグラウンドで実行でき、出力を対話的に監視できます。この場合、コマンド `kill -1 %1` を使用して、バッファの最後の 1000 行を標準出力にダンプし、バックグラウンドで実行を継続するようにスクリプトに指示できます。

### 例9.13 バックグラウンドでの名前付きパイプログスクリプトの実行

```
# ds-logpipe.py /var/log/dirsrv/slapd-example/errors.pipe &
```

スクリプトの終了 (または強制終了もしくは中断) 時に最後の 1000 行をダンプし、出力をファイルに自動的に保存するには、スクリプト出力をユーザー定義のファイルにリダイレクトします。

### 例9.14 名前付きパイプログスクリプトからの出力の保存

```
# ds-logpipe.py /var/log/dirsrv/slapd-example/errors.pipe > /etc/dirsrv/myerrors.log 2>&1
```

名前付きパイプスクリプトは、Directory Server プロセスと連動して自動的に開始および停止するよう

に設定できます。これには、**-s** 引数を使用して、スクリプトの実行中にスクリプトの PID を書き込むサーバーの PID ファイルの名前が必要になります。サーバーの PID は、サーバーの PID ファイルを参照するか、実際のプロセス ID 番号を指定して参照できます (サーバープロセスがすでに実行している場合)。

### 例9.15 サーバー PID の指定

```
# ds-logpipe.py /var/log/dirsrv/slapd-example/errors.pipe --serverpidfile /var/run/dirsrv/slapd-example.pid
```

プラグインは、名前付きパイプからログデータを読み取り、操作を実行するために呼び出すことができます。

### 例9.16 関連するプラグインを使用した名前付きパイプログスクリプト

```
# ds-logpipe.py /var/log/dirsrv/slapd-example/errors.pipe --
plugin=/usr/share/dirsrv/data/logregex.py logregex.regex="warning"
```

例9.16 「関連するプラグインを使用した名前付きパイプログスクリプト」では、文字列 **warning** を含むログ行のみが内部バッファに保存され、スクリプトの終了時に出力されます。

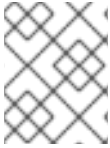
スクリプト引数で渡されるプラグインがない場合、スクリプトは 1000 のログ行をバッファに格納 (デフォルト) し、終了時に出力します。スクリプトには 2 つのプラグインがあります。

- **logregex.py** は、所定の正規表現に一致するログ行のみを保持します。プラグイン引数には、使用する文字列または正規表現を指定する **logregex.regex=pattern** の形式があります。複数の **logregex.regex** 引数があり、すべては AND ステートメントとして処理されます。エラーログの行は、指定したすべての引数と一致する必要があります。一致するログ行をレコード (OR) にするには、文字列または式の間パイプ (|) を付けて単一の **logregex.regex** 引数を使用します。正規表現とその構文についての詳細は、**pcre** または **Python** の正規表現のドキュメントを参照してください。
- **failedbinds.py** ログは BIND の試行に失敗したため、このプラグインはアクセスログにのみ使用されます。これは、実際のログメッセージが書き込まれるファイルである **failedbinds.logfile=/path/to/access.log** オプションを取ります。このプラグインは、かなりの量の処理を行う複雑なプラグインの例で、他のタイプのアクセスログ処理を行うための参照先として最適です。

## 9.5. DN2RDN

9.0 よりも古い Directory Server のバージョンは、**entrydn** インデックスを使用して **id2entry.db4** データベースのエントリー ID をエントリーの完全な DN にマッピングするのに役立てました。ただし、これにより、親 DN が変更した場合にエントリーの子を識別して DN を更新する方法がなかったため、**modrdn** 操作はリーフエントリーでしか実行できなくなります。サブツリーレベルの名前変更が許可されている場合、ID からエントリーへのマッピングは、**id2entry.db** データベースで **entryrdn** インデックスを使用して行われます。

アップグレード後も、Directory Server のインスタンスは **entrydn** インデックスを引き続き使用している可能性があります。**dn2rdn** ツールには目的が 1 つあります (**entrydn** インデックスを **entryrdn** に変換して、DN ベースのフォーマットから RDN ベースの形式に変換する)。



## 注記

**dn2rdn** ツールは、常にローカルの Directory Server インスタンス上で実行されるため、ツールは `/usr/sbin/` ディレクトリーにあります。

## 9.6. PWDHASH

**pwdhash** ユーティリティーは、指定したプレーンテキストのパスワードを暗号化します。ユーザーまたは Directory Manager がログインできない場合は、**pwdhash** を使用して、暗号化されたパスワードを比較します。生成されたハッシュを使用して、Directory Manager のパスワードを手動でリセットすることもできます。

**pwdhash** ユーティリティーは、以下のストレージスキームを使用してパスワードを暗号化します。

- **-s storage\_scheme** パラメーターを **pwdhash** に渡すと、指定されたスキームが使用されません。
- **-D config\_directory** パラメーターを **pwdhash** に渡すと、**nsslapd-rootpwstoragescheme** 属性に設定されたスキームが使用されます。
- 有効な Directory Server 設定ディレクトリーへのパスを指定しない場合や、スキームを **pwdhash** に渡さない場合は、ユーティリティーでは Directory Server のデフォルトストレージスキームが使用されます。

ストレージスキームの詳細、サポートされている値のリスト、およびデフォルト設定の詳細は、「[パスワードストレージスキーム](#)」を参照してください。

構文およびコマンドラインオプションの詳細は、**pwdhash(1)** の man ページを参照してください。

## 付録A DIRECTORY SERVER で利用可能なスクリプトのテスト

Red Hat Directory Server には、ストレスや負荷の異なる条件で Directory Server のパフォーマンスをテストするために使用できるスクリプトが2つ用意されています。テストスクリプトは各種環境をシミュレートします。これにより、管理者は設定やマシンの変更を実稼働環境に配置する前に評価できます。

**ldclt** および **rsearch** はどちらも **/usr/bin** ディレクトリーにあります。

### A.1. LDCLT (負荷ストレステスト)

LDAP クライアントスクリプト (**ldclt**) は、Directory Server のロードテストを行うために、ユーザー定義のシナリオで、サーバーへの複数のクライアント接続を確立します。クライアント操作には、ディレクトリーの追加、検索、変更、modRDN、削除、LDIF ファイルの生成などの設定操作が含まれます。操作は、ディレクトリーのより現実的な使用環境をシミュレートするために、ランダムなユーザーとしてバインディングとバインディング解除を行い、ランダムなタスクを実行して、操作をランダムに行うことができます。

**ldclt** ツールは、Directory Server のパフォーマンスを測定するために、継続的に繰り返し実行される操作の完了時間を測定します。複数のスレッドを使用すると、負荷の高い状態でパフォーマンスをテストできます。各テストは、同じタイプの LDAP 操作を実行しますが、異なる設定 (ユーザーの認証情報が異なる、属性タイプやサイズが異なる、ターゲットサブツリーが異なる等) を使用します。

LDAP 操作変数の定義に加えて、管理者はサーバーに特定の負荷を設定するために、スレッドのパフォーマンスを制御できます。

**ldclt** ツールは、特に自動テストに使用することが意図されているため、複雑なテスト操作に対しても、オプションは広範囲で柔軟で簡単にスクリプト化されます。



#### 注記

**ldclt** は負荷テストであるため、大量のシステムリソースを使用することに注意してください。このツールは、最低 8MB のメモリーを使用します。スレッドの数、操作のタイプ、およびその他の設定によっては、より多くのメモリーを使用する場合があります。

操作のタイプと、それらの操作に使用されるディレクトリーデータに応じて、**ldclt** は独自のリソース制限を設定する可能性があります。システムリソース制限の管理の詳細は、**ulimit** および **getrlimit** の man ページを参照してください。

**ldclt** ユーティリティーは、**/usr/bin** ディレクトリーにあります。

#### A.1.1. 構文

```
ldclt-q-Q-v-V-Emax_errors-bbase_DN-hhost-pport-ttimeout-Dbind_DN-wpassword-
oSASL_options-eexecution_params-amax_pending-nnumber_of_threads-iinactivity_times-
Nnumber_of_samples-lerror_code-Ttotal_number_of_operations-rlow_range-Rhigh_range-ffilter-
sscope-Sconsumer-Psupplier_port-Wwait_time-Zcertificate_file
```

#### A.1.2. ldclt オプション

表A.1 ldclt オプション

オプション	説明
<code>-a max_pending_ops</code>	定義された最大保留操作数で、ツールを非同期モードで実行します。
<code>-b base_dn</code>	LDAP 操作テストの実行に使用するベース DN を指定します。指定しないと、デフォルト値は <b>dc=example,dc=com</b> になります。
<code>-D bind_dn</code>	サーバーへの接続に使用する <b>ldclt</b> ユーティリティのバインド DN を指定します。
<code>-E max_errors</code>	ツールの終了までに、テスト LDAP 操作で許容されるエラー発生数の最大数を設定します。デフォルトは 1000 です。
<code>-e execution_params</code>	テストに使用する操作のタイプおよび他のテスト環境パラメーターを指定します。 <b>-e</b> の可能な値は、表 A.2 「実行パラメーター」 にリスト表示されています。このオプションには、コンマ区切りリストで複数の値を指定できます。
<code>-f filter</code>	検索テストに使用する LDAP 検索フィルターを指定します。
<code>-h</code>	テストを実行する Directory Server のホスト名または IP アドレスを指定します。ホストが指定されていない場合、 <b>ldclt</b> はローカルホストを使用します。
<code>-l error_code</code>	特定のレスポンスコードにマッチするエラーが発生した場合に無視するように <b>ldclt</b> に指示します。たとえば、 <b>-l 89</b> は、エラーコード 89 を無視するようにサーバーに指示します。
<code>-i inactivity_times</code>	終了するまでにツールが非アクティブでいられる間隔を設定します。デフォルトでは、この設定は 3 で、これは 30 秒 (各操作の間隔が 10 秒) と解釈されます。
<code>-N number_of_samples</code>	実行する反復回数を設定します (10 秒間のテストを実行する回数)。デフォルトでは、この設定は無制限で、ツールは手動で停止した場合にのみ終了します。
<code>-n number_of_threads</code>	操作に対して同時に実行するスレッドの数を設定します。デフォルト値は 10 です。

オプション	説明
-o SASL_option	<p>SASL を使用してサーバーに接続するようにツールに指示し、使用する SASL メカニズムを指定します。形式は <b>-o saslOption=value</b> です。saslOption は、次の 6 つの値のいずれかになります。</p> <ul style="list-style-type: none"> <li>* mech: SASL 認証メカニズム</li> <li>* authid: サーバーにバインドしているユーザー (Kerberos プリンシパル)</li> <li>* authzid: プロキシ認可 (プロキシ認可はサポートされていないため、サーバーによって無視されます)</li> <li>* secProp: セキュリティープロパティ</li> <li>* realm: Kerberos レalm</li> <li>* flags</li> </ul> <p>想定される値は、サポートされるメカニズムによって異なります。-o を複数回使用して、メカニズムに必要なすべての SASL 情報を渡すことができます。以下に例を示します。</p> <pre>[literal,subs="+quotes,verbatim"].... -o "mech=DIGEST-MD5" -o "authzid=test_user" -o "authid=test_user" ....</pre>
-P supplier_port	<p>レプリケーションテストのためにサプライヤーサーバーに接続するのに使用するポートを指定します。指定されていない場合のデフォルトは 16000 です。</p>
-p port	<p>テストしている Directory Server インスタンスのサーバーポート番号を指定します。</p>
-Q	<p>ツールをスーパー quiet モードで実行します。これは、<b>ldclt</b> が実行する操作で発生したエラーを無視します。</p>
-q	<p>ツールを quiet モードで実行します。</p>
-R number	<p>範囲の上限の数値を設定します。</p>
-r number	<p>範囲の下限の数値を設定します。</p>
-S consumer_name	<p>レプリケーションテストを実行するために接続するコンシューマーサーバーのホスト名を指定します。</p>
-s scope	<p>検索条件を指定します。<b>ldapsearch</b> と同様に、値は subtree、one、または base に設定することができます。</p>

オプション	説明
<code>-T ops_per_thread</code>	1 スレッドで許容される最大操作数を設定します。
<code>-t timeout</code>	LDAP 操作のタイムアウト期間を設定します。デフォルトは 30 秒です。
<code>-V</code>	ツールを高冗長モードで実行します。
<code>-v</code>	ツールを冗長モードで実行します。
<code>-W wait_time</code>	1 つの操作の終了後に次の操作を開始するまで <b>ldclt</b> ツールが待機する時間を秒単位で設定します。デフォルトは 0 で、待機時間がないことを意味します。
<code>-w password</code>	テストのために Directory Server にバインドする際に、 <b>-D</b> アイデンティティで使用するパスワードを指定します。
<code>-Z /path/to/cert.db</code>	テスト接続用に TLS を有効にし、証明書データベースとして使用するファイルを参照します。

**-e** オプションは、**ldclt** テスト操作の実行パラメーターを設定します。複数のパラメーターをコンマ区切りリストで設定できます。以下に例を示します。

```
-e add,bindeach,genldif=/var/lib/dirsrv/slaped-instance/ldif/generated.ldif,inetOrgPerson
```

表A.2 実行パラメーター

パラメーター	説明
abandon	非同期検索リクエストについて破棄操作を開始します。
add	ディレクトリーにエントリーを追加します ( <b>ldapadd</b> )。
append	<b>genldif</b> オプションで生成された LDIF ファイルの最後にエントリーを追加します。
ascii	ASCII 7 ビットの文字列を生成します。
attreplace=name:mask	既存のエントリーの属性 ( <b>name</b> ) を置き換える変更操作を実行します。
attrlist=name:name:name	検索操作で返す属性のリストを指定します。



パラメーター	説明
attrsonly=#	検索操作で使用して、属性値を読み取るかどうかを設定します。設定できる値は 0 (値を読み取る) または 1 (値を読み取らない) です。
bindeach	<b>ldclt</b> ツールに対し、試行する各操作とバインドするように指示します。
bindonly	<b>ldclt</b> ツールに対し、バインド/バインド解除操作のみを実行するように指示します。他の操作は実行されません。
close	バインド解除操作を実行するのではなく、接続を終了するようにツールに指示します。
cltcertname=name	TLS 接続に使用する TLS クライアント証明書の名前を指定します。
commoncounter	<b>ldclt</b> ツールにより開いたすべてのスレッドが同じカウンターを共有します。
counteach	ツールに対し、成功した操作だけでなく、各操作をカウントするように指示します。
delete	削除操作を開始します。
deref	検索操作 ( <b>esearch</b> ) に間接参照制御を追加します。adds で、これは <b>ldclt</b> に、新しいエントリーに <b>secretary</b> 属性を追加するよう指示し、間接参照を許可します。
dontsleeponserverdown	サーバーが停止すると、ツールのループが非常に高速になります。
emailPerson	これにより、 <b>emailPerson</b> オブジェクトクラスが生成されたエントリーに追加されます。これは add 操作でのみ有効です ( <b>-e add</b> )。
esearch	完全一致の検索を実行します。
genldif=filename	操作で使用する LDIF ファイルを生成します。
imagesdir=path	テストで使用するイメージの場所を指定します。
incr	増分値を有効にします。

パラメーター	説明
inetOrgPerson	これにより、 <b>inetOrgPerson</b> オブジェクトクラスが生成されたエントリーに追加されます。これは add 操作でのみ有効です ( <b>-e add</b> )。
keydbfile= <b>file</b>	TLS 接続で使用するキーデータベースのパスおよびファイル名を含めます。
keydbpin= <b>password</b>	キーデータベースにアクセスするためのトークンパスワードを含めます。
noglobalstats	定期的なグローバル統計値を <b>出力しない</b> ようにツールに指示します。
noloop	増分数をループしません。
object= <b>filename</b>	入力ファイルからエントリーオブジェクトを作成します。
person	これにより、 <b>person</b> オブジェクトクラスが生成されたエントリーに追加されます。これは add 操作でのみ有効です ( <b>-e add</b> )。
random	<b>Idclt</b> ユーティリティーに、ランダムなフィルターやベース DNS などすべてのランダムな要素を使用するように指示します。
randomattrlist= <b>name:name:name</b>	<b>Idclt</b> ユーティリティーに対して、指定のリストからランダムな属性を選択するように指示します。
randombase	<b>Idclt</b> ユーティリティーに、ディレクトリーからランダムなベース DN を選択するように指示します。
randombaselow= <b>value</b>	乱数ジェネレーターの低い値を設定します。
randombasehigh= <b>value</b>	乱数ジェネレーターの高い値を設定します。
randombinddn	<b>Idclt</b> ユーティリティーに、ランダムなバインド DN を使用するように指示を出します。
randombinddnfromfile= <b>file</b>	<b>Idclt</b> ユーティリティーに対して、ファイルから選択したランダムなバインド DN を使用するように指示します。ファイルの各エントリーには、適切な DN/パスワードのペアが必要です。
randombinddnlow= <b>value</b>	乱数ジェネレーターの低い値を設定します。
randombinddnhigh= <b>value</b>	乱数ジェネレーターの高い値を設定します。

パラメーター	説明
rdn=attrname:value	検索フィルターとして使用する RDN を提供します。これは、 <b>-f</b> フィルターの代わりに使用されます。
referral=value	操作の参照動作を設定します。on (参照を許可する)、off (参照を許可しない)、または rebind(再度接続を試みる) の 3 つのオプションを選択できます。
smoothshutdown	<b>ldclt</b> ユーティリティーに、ワーカースレッドが終了するまでメインスレッドをシャットダウンしないように指示します。
string	<b>ldclt</b> ユーティリティーに、ランダムな数値ではなくランダムな文字列を作成するように指示します。
v2	テスト操作に LDAPv2 を使用するように <b>ldclt</b> ユーティリティーに指示します。
withnewparent	modRDN 操作を実行し、エントリーの名前を引数として設定した newparent に変更します。
randomauthid	ランダムな SASL 認証 ID を使用します。
randomauthidlow=value	ランダムな SASL 認証 ID の下限値を設定します。
randomauthidhigh=value	ランダムな SASL 認証 ID の上限値を設定します。

### A.1.3. ldclt の結果

**ldclt** は、指定された数のスレッドで、指定したあらゆる操作を継続的に実行します。デフォルトでは、10 秒間隔でパフォーマンスの統計値を画面に出力します。

結果には、1 スレッドあたりおよび 1 秒あたりの平均操作数と、次にその 10 秒間のウィンドウで実行された操作の合計数が示されます。

```
ldclt[process_id] Average rate: number_of_ops/thr (number_of_ops/sec), total:
total_number_of_ops
```

以下に例を示します。

```
ldclt[22774]: Average rate: 10298.20/thr (15447.30/sec), total: 154473
```

**ldclt** は、15 分ごとおよびツールの終了時に、累積の平均と合計を出力します。

```
ldclt[22774]: Global average rate: 821203.00/thr (16424.06/sec), total: 12318045
ldclt[22774]: Global number times "no activity" reports: never
ldclt[22774]: Global no error occurs during this session.
```

```
Catch SIGINT - exit...
ldclt[22774]: Ending at Wed Feb 24 18:39:38 2010
ldclt[22774]: Exit status 0 - No problem during execution.
```

一部の操作 (adds など) や、**-v** や **-V** などの詳細出力オプションを使用して、追加のデータを画面に出力します。情報の種類は操作のタイプによって異なりますが、通常、操作を実行するスレッドと、操作によって呼び出されるプラグインが表示されます。以下に例を示します。

```
ldclt -b ou=people,dc=example,dc=com -D "cn=Directory Manager" -w secret12 -e
add,person,incr,noloop,commoncounter -r90000 -R99999 -f "cn=testXXXXX" -V

...
ldclt[11176]: T002: After ldap_simple_bind_s (cn=Directory Manager, secret12)
ldclt[11176]: T002: incremental mode:filter="cn=test00009"
ldclt[11176]: T002: ttctx->bufFilter="cn=test00009"
ldclt[11176]: T002: attrs[0]=("objectclass" , "person")
ldclt[11176]: T002: attrs[1]=("cn" , "test00009")
ldclt[11176]: T002: attrs[2]=("sn" , "toto sn")

...
ldclt[11176]: Average rate: 195.00/thr ( 195.00/sec), total: 1950
ldclt[10627]: Global average rate: 238.80/thr (238.80/sec), total: 2388
ldclt[10627]: Global number times "no activity" reports: never
ldclt[10627]: Global no error occurs during this session.
Catch SIGINT - exit...
ldclt[10627]: Ending at Tue Feb 23 11:46:04 2010
ldclt[10627]: Exit status 0 - No problem during execution.
```

ほとんどのエラーは、テストを中断せずに **ldclt** により処理されます。発生した致命的なエラーはツールの終了ステータスでリスト表示され、累積合計で返されます。

```
Global no error occurs during this session.
```

発生した LDAP 操作エラーはスレッド内で処理されます。接続エラーは、テスト全体に影響を与えることなくスレッドを強制終了します。**ldclt** ユーティリティーは、各 LDAP エラーが発生する回数をカウントします。記録されるエラーの合計数が 1000(デフォルト) を超える場合、スクリプト自体はエラーになります。

**ldclt** が LDAP エラーに応答する方法を設定できます。**-E** オプションを使用して、LDAP エラーの発生後にスクリプトがエラーになる際のしきい値に異なる値を設定します。**-I** オプションを使用して、全スレッドで指定された LDAP エラーコードを無視するようにスクリプトに指示します。エラーによる終了の上限値を変更し、特定のエラーコードを無視することで、テストスクリプトまたはテスト設定を調整したり、改善したりできます。

#### A.1.4. ldclt の終了および ldclt 終了コード

**ldclt** コマンドは無期限に実行されます。このスクリプトは、致命的なランタイムや初期化エラーが発生した、LDAP エラーの上限値に達した、すべてのスレッドの機能が停止した、または操作や時間の制限に達したなど、さまざまな状況で自身を停止できます。

実行の統計値は、スクリプトが終了するか、ユーザーがスクリプトを終了するのいずれかによってコマンドが完了するまで表示されません。**ldclt** スクリプトを中断する方法は 2 つあります。

- Ctrl+バックスラッシュ (kbd:[^\]) キーを押すか、**kill -3** コマンドを実行すると、スクリプトを終了せずに現在の統計値を表示します。

- Ctrl+C (^C) キーを押すか、**kill -2** コマンドを実行すると、スクリプトを終了してグローバル統計値を出力します。

**ldclt** スクリプトが終了するか、中断されると、統計値およびエラー情報と共に終了コードが返されません。

表A.3 ldclt 終了コード

終了コード	説明
0	成功 (エラーなし)。
1	操作で深刻な致命的なエラーが発生した。
2	ツールで渡されたパラメーターにエラーがあった。
3	ツールが LDAP エラー数の上限に達した。
4	このツールが Directory Server インスタンスにバインドできなかった。
5	このツールが TLS 経由で接続するための TLS ライブラリーを読み込むことができなかった。
6	マルチスレッド (mutex) エラーが発生しました。
7	初期化に問題があった。
8	このツールが、メモリー割り当てエラーなどのリソース制限に達した。
99	スクリプトで不明なエラーが発生した。

### A.1.5. 使用方法

ここでは、**ldclt** を使用して Directory Server をテストする一般的な例を説明します。より複雑な例を含むテストスクリプトは、**ldclt** ソースファイルにあります。このファイルは、389 Directory Server プロジェクト (<https://github.com/389ds/389-ds-base/tree/master/ldap/servers/slapd/tools/ldclt/examples>) からダウンロードできます。

すべての **ldclt** コマンドには、(テストのタイプによって異なる) 実行パラメーターと (すべての操作タイプで同じ) 接続パラメーターのセットが必要です。以下に例を示します。

```
# ldclt -e execution_parameters -h localhost -p 389 -D "cn=Directory Manager" -w secret -b "ou=people,dc=example,dc=com"
```

**ldclt** を実行すると、まずそのテスト用に設定されたすべてのパラメーターを出力します。

```
Process ID      = 1464
Host to connect = localhost
Port number     = 389
```

```

Bind DN          = cn=Directory Manager
Passwd          = secret
Referral        = on
Base DN         = ou=people,dc=example,dc=com
Filter          = "cn=MrXXX"
Max times inactive = 3
Max allowed errors = 1000
Number of samples = -1
Number of threads = 10
Total op. req.   = -1
Running mode     = 0xa0000009
Running mode     = quiet verbose random exact_search
LDAP oper. timeout = 30 sec
Sampling interval = 10 sec
Scope           = subtree
Attrsonly       = 0
Values range     = [0 , 1000000]
Filter's head    = "cn=Mr"
Filter's tail    = ""

```

### A.1.5.1. LDIF の生成

**ldclt** ツール自体を使用して、テストに使用できる LDIF ファイルを生成できます。



#### 注記

LDIF ファイルを生成する際に、**ldclt** ツールはサーバーへの接続や操作の実行を試みません。

LDIF ファイルの生成には、エントリーの作成にツールが使用する基本的なテンプレートファイル (**-e object**) と、指定した出力ファイル (**-e genldif**) が必要です。

テンプレートファイルでは、エントリー属性に明示的な値を指定することや、変数を使用することができます。エントリー属性に一意の値を簡単に指定する場合は、**/usr/share/dirsrv/data** ディレクトリーに、姓、名、および組織単位を生成する 3 つのデータファイルが含まれます。これらの値のリストは、テストユーザーおよびディレクトリーツリー (それぞれ **dbgen-FamilyNames**、**dbgen-GivenNames**、および **dbgen-OrgUnits**) を作成するために使用できます。これらのファイルは、**rndfromfile** オプション、**incrfromfile** オプション、または **incrfromfilenoloop** オプションで使用できます。

テンプレートファイルの基本的な形式は次のとおりです。

```
# comment
```

```
attribute: string | variable=keyword(value)
```

変数は A から H までの任意の文字にすることができます。使用できるキーワードは、[表A.4 「ldclt テンプレート LDIF ファイルのキーワード」](#) にリスト表示されています。

一部の変数およびキーワードは、**-e object** オプションおよびその他の利用可能なパラメーター (**rdn** など) で渡すことができます。

```
-e object=inet.txt,rdn='uid:[A=INCRNNOLOOP(0;99999;5)]'
```

表A.4 ldclt テンプレート LDIF ファイルのキーワード

キーワード	説明	形式
RNDN	指定された範囲 (low - high) 内で指定の長さの、ランダムな値を生成します。	RNDN(low;high;length)
RNDFROMFILE	指定されたファイルで利用可能な値からランダムな値を取得します。	RNDFROMFILE(filename)
INCRN	指定された範囲 (low - high) 内で指定の長さの、連続した値を生成します。	INCRN(low;high;length)
INCRNOLOOP	(インクリメント範囲をループすることなく) 指定された範囲内 (low - high) で、指定された長さの連続した値を作成します。	INCRNOLOOP(low;high;length)
INCRFROMFILE	指定のファイルの値全体をインクリメントして値を作成します。	INCRFROMFILE(filename)
INCRFROMFILENOLOOP	値をループバックすることなく、ファイルの値全体をインクリメントして値を作成します。	INCRFROMFILENOLOOP(filename)
RNDS	指定された長さのランダムな値を生成します。	RNDS(length)

たとえば、このテンプレートファイルは `/usr/share/dirsrv/data` のサンプルファイルから名前を取得し、他の属性を動的にビルドします。

#### 例A.1 テンプレートファイルの例

```
objectclass: inetOrgPerson
sn: [B=RNDFROMFILE(/usr/share/dirsrv/data/dbgen-FamilyNames)]
cn: [C=RNDFROMFILE(/usr/share/dirsrv/data/dbgen-GivenNames)] [B]
password: test[A]
description: user id [A]
mail: [C].[B]@example.com
telephonenumber: (555) [RNDN(0;999;3)]-[RNDN(0;9999;4)]
```

続いて、`ldclt` コマンドは、そのテンプレートを使用して 100,000 エントリーを持つ LDIF ファイルをビルドします。

```
# ldclt -b "ou=people,dc=csb" -e object=inet.txt,rdn='uid:[A=INCRNOLOOP(0;99999;5)]' -e genldif=100Kinet.ldif,commoncounter
```

#### A.1.5.2. エントリーの追加

**ldclt** ツールは、2つのテンプレートのいずれかにマッチするエントリーを追加できます。

- person
- inetorgperson

**-f** フィルターは、ユーザーエントリーの命名属性の形式を設定します。たとえば、**-f "cn=MrXXXXX"** は、**-f "cn=Mr01234"** ような名前を作成します。**-f** で **person** または **inetorgperson** パラメーターを使用すると、基本的なエントリーが作成されます。

```
objectclass: person
sn: ex sn
cn: Mr01234
```

**rdn** パラメーターと **object** ファイルを使用して、さらに複雑なエントリー (検索とテストの変更に適切) を作成できます。エントリーの全オプションについては、「[LDIF の生成](#)」で説明されています。**rdn** パラメーターおよび **object** パラメーターは、ディレクトリーに追加または変更するエントリーの形式です。**rdn** 実行パラメーターはキーワードパターン ([表A.4 「ldclt テンプレート LDIF ファイルのキーワード」](#) に記載) を取り、テキストファイルに記載されているエントリーからエントリープールを取得します。

```
-e rdn='uid:[A=INCRNNOLOOP(0;99999;5)]',object=inet.txt
```

**ldclt** ツールは、数値順にエントリーを作成します。つまり、これらのエントリーを追加する方法およびシーケンスをカウントする方法も定義する必要があります。このためのいくつかの可能なオプションは次のとおりです。

- **-r** および **-R**: エントリーの数値の範囲を設定します
- **incr** または **random**: 番号の割り当て方法を設定します (**-f** でのみ使用されます)
- **-r** および **-R**: エントリーの数値の範囲を設定します
- **noloop**: 範囲の最後に到達したら、ループバックするのではなく add 操作を停止します

## 例A.2 エントリーの追加

```
# ldclt -b ou=people,dc=example,dc=com -D "cn=Directory Manager" -w secret -e
add,person,incr,noloop,commoncounter -r0 -R99999 -f "cn=MrXXXXX" -v -q
```

**add** 操作は、より複雑なテスト用にディレクトリーツリーをビルドするのにも使用できます。存在しないブランチに属するディレクトリーにエントリーが追加されるたびに、**ldclt** ツールはそのブランチエントリーを自動的に作成します。



### 注記

存在しないブランチの子であるエントリーを初めて追加すると、ブランチエントリーがディレクトリーに追加されます。ただし、エントリー自体は追加されません。これ以降のエントリーは新しいブランチに追加されます。

ブランチエントリーを自動的に追加するには、その命名属性を **cn**、**o**、または **ou** にする必要があります。



### 例A.3 ディレクトリーツリーの作成

```
# ldclt -b ou=DeptXXX,dc=example,dc=com -D "cn=Directory Manager" -w secret -e
add,person,incr,noloop,commoncounter -r0 -R999999 -f "cn=MrXXXXXX" -v -q
```

#### A.1.5.3. 検索操作

最も基本的な **ldclt** 検索テストは、指定されたベース DN 内のすべてのエントリーを検索します。これには、**esearch** と **random** の 2 つの実行パラメーターが使用されます。

### 例A.4 基本的な検索操作

```
# ldclt -h localhost -p 389 -D "cn=Directory Manager" -w secret -b
"ou=people,dc=example,dc=com" -f uid=testXXXXXX -e esearch,random -r0 -R999999 -l 32
```



#### 重要

すべてのエントリーを返す検索は、スレッドごとに 1GB 程度の大量のメモリーを使用する可能性があります。**ldclt** は、1つのエントリーを返す検索を実行するように設計されています。

検索結果を拡張して、エントリーに含まれる属性を返すことができます (「LDIF の生成」に、複数の属性が含まれるエントリーを生成するための情報が記載されています)。エントリーの属性の特定リストを返すには、**attrlist** 実行パラメーターと、属性のコロン区切りリストを使用します。

### 例A.5 属性リストの検索

```
# ldclt -h localhost -p 389 -b "ou=people,dc=example,dc=com" -f uid=XXXXXX -e esearch,random -
r0 -R999999 -l 32 -e attrlist=cn:mail
```

また、**ldclt** 検索操作は、検索リストからランダムに選択した属性の属性値を返すことができます。このリストは、属性のコロン区切りリストの **randomattrlist** 実行パラメーターで指定します。

### 例A.6 ランダム属性のリストの検索

```
# ldclt -h localhost -p 389 -b "ou=people,dc=example,dc=com" -f uid=XXXXXX -e esearch,random -
r0 -R999999 -l 32 -e randomattrlist=cn:sn:ou:uid:mail:mobile:description
```

エントリーを照合するのに使用されるフィルターは、命名属性だけでなく、他のエントリー属性を対象とすることができます。生成された LDIF の属性により異なります。

### 例A.7 代替のフィルターを使用した検索

```
# ldclt -h localhost -p 389 -b "ou=people,dc=example,dc=com" -f mail=XXXXXX@example.com -
e esearch,random -r0 -R999999 -l 32 -e randomattrlist=cn:sn:ou:uid:mail:mobile:description
```

検索操作では、RDN 形式のフィルターを使用してエントリーを検索することもできます。**rdn** パラメーターおよび **object** 実行パラメーターは、ディレクトリーに追加または変更するエントリーの形式です。**rdn** 実行パラメーターはキーワードパターン (表A.4 「[Idclt テンプレート LDIF ファイルのキーワード](#)」に記載) を取り、テキストファイルに記載されているエントリーからエントリープールを取得します。

#### 例A.8 RDN フィルターを使用した検索

```
# Idclt -h localhost -p 389 -b "ou=people,dc=example,dc=com" -e rdn='mail:
[RNDN(0;99999;5)]@example.com',object="inet.txt" -e attrlist=cn:telephonenumber
```

#### A.1.5.4. 操作の変更

**attreplace** 実行パラメーターは、エントリーの特定属性を置き換えます。

変更操作では、RDN フィルターを使用して更新するエントリーを検索します。**rdn** パラメーターおよび **object** パラメーターは、ディレクトリーに追加または変更するエントリーの形式です。**rdn** 実行パラメーターはキーワードパターン (表A.4 「[Idclt テンプレート LDIF ファイルのキーワード](#)」に記載) を取り、テキストファイルに記載されているエントリーからエントリープールを取得します。

#### 例A.9 変更操作

```
# Idclt -h localhost -p 389 -D "cn=Directory Manager" -w secret -b
"ou=people,dc=example,dc=com" -e rdn='uid:[RNDN(0;99999;5)]' -l 32 -e
attreplace='description: random modify XXXXX'
```

#### A.1.5.5. modrdn 操作

**Idclt** コマンドは、2 種類の **modrdn** 操作をサポートします。

- エントリーの名称変更
- エントリーの新しい親への移動

**Idclt** ユーティリティーは、無作為に選択された DN から新しいエントリー名または親を作成します。

基本的な名称変更操作には、3 つの実行パラメーターが必要です。

- **rename**
- **rdn='pattern'**
- **object=file**

**rdn** パラメーターおよび **object** パラメーターは、ディレクトリーに追加または変更するエントリーの形式です。**rdn** 実行パラメーターはキーワードパターン (表A.4 「[Idclt テンプレート LDIF ファイルのキーワード](#)」に記載) を取り、テキストファイルに記載されているエントリーからエントリープールを取得します。

#### 例A.10 単純な名前操作

```
# ldclt -h localhost -p 389 -D "cn=Directory Manager" -w secret -b
"ou=people,dc=example,dc=com" -l 32 -l 68 -e rename,rdn='uid:[RNDN(0;999;5)]',object="inet.txt"
```

**withnewparent** 実行パラメーターを使用すると、エントリーの名前が変更され、新しい親エントリーの下に移動します。親エントリーが存在しない場合は、**ldclt** ツールがこれを作成します。[3]

#### 例A.11 エントリーの名前を変更して新しい親に移動する

```
# ldclt -h localhost -p 389 -D "cn=Directory Manager" -w secret12 -b
"ou=DeptXXX,dc=example,dc-com" -l 32 -l 68 -e
rename,withnewparent,rdn='uid:Mr[RNDN(0;99999;5)]',object="inet.txt"
```

### A.1.5.6. 操作の削除

**ldclt** の削除操作は、追加操作のまったく逆になります。追加と同様に、削除操作は複数の方法でエントリーを削除できます。

- ランダムに (**-e delete,random**)
- RDN-ranges (**-e delete,rdn=[pattern]**)
- 順次 (**-e delete,incr**)

ランダム削除は、指定されたエントリーの範囲内で実行されるように設定されます。これには以下のオプションが必要です。

- **-e delete,random**
- バインドされた範囲の **-r** および **-R**
- **-f** (フィルターがエントリーと一致する)

#### 例A.12 ランダム削除操作

```
# ldclt -b "ou=people,dc=example,dc=com" -D "cn=Directory Manager" -w secret -e delete,random
-r0 -R99999 -f "uid=XXXXXX" -l 32 -v -q
```

RDN ベースの削除は、キーワード (表A.4 「**ldclt** テンプレート LDIF ファイルのキーワード

 にリスト表示されている) と共に **rdn** 実行パラメーターを使用して、テキストファイルにリスト表示されているエントリーからエントリープールを取得します。この形式には3つの実行パラメーターが必要です。

- **-e delete**
- **-e rdn='pattern'**
- **-e object='file'**

#### 例A.13 RDN ベースの削除操作

```
# ldclt -b "ou=people,dc=example,dc=com" -D "cn=Directory Manager" -w secret -e
delete,rdn='uid:[INCRNNOLOOP(0;99999;5)]',object="inet.txt" -l 32 -v -q
```

最後の削除操作形式はランダム削除形式と似ていますが、ランダムではなく、指定した範囲を順番に処理するだけです。

- `-e delete,incr`
- バインドされた範囲の `-r` および `-R`
- `-f` (フィルターがエントリーと一致する)

#### 例A.14 順次削除操作

```
# ldclt -b "ou=people,dc=example,dc=com" -D "cn=Directory Manager" -w secret -e delete,incr -r0
-R99999 -f "uid=XXXXXX" -l 32 -v -q
```

#### A.1.5.7. バインド操作

デフォルトでは、各 `ldclt` スレッドはサーバーに一度バインドし、そのすべての操作を1つのセッションで実行します。`-e bindeach` は他の操作と併用して、操作ごとにバインドしてから、次の操作を開始する前にバインドを解除することを `ldclt` ツールに指示します。

```
-e add,bindeach ...
```

バインドおよびバインド解除操作のみをテストするには、`-e bindeach,bindonly` 実行パラメーターを使用し、他の操作情報は使用しません。以下に例を示します。

```
# ldclt -h localhost -p 389 -b "ou=people,dc=example,dc=com" -e bindeach,bindonly -e bind_info
```

バインド操作は、接続パラメーターで `-D` および `-w` ユーザー名/パスワードペアを使用して、テストに使用する単一のユーザーを指定できます。



#### 注記

`-e close` オプションをバインドパラメーターと共に使用して、完全にバインドを解除するのではなく、接続を破棄することが Directory Server に及ぼす影響をテストします。

#### 例A.15 バインドのみおよびテスト終了

```
# ldclt -h localhost -p 389 -D "cn=Directory Manager" -w secret -e bindeach,bindonly,close
```

指定のファイル (`randombinddnfromfile`) からランダムなバインドアイデンティティを選択するために使用できる、または範囲内からランダムに選択された DN (`-e randombinddn,randombinddnhigh=Y`) を使用する実行パラメーターもあります。

#### 例A.16 ファイル内の ID からのランダムバインド

```
# ldclt -h localhost -p 389 -e bindeach,bindonly -e randombinddnfromfile=/tmp/testbind.txt
```

ランダムなアイデンティティとのバインディングは、生成された LDIF からアイデンティティが追加されている場合や、**-e add** を使用してアカウントが範囲に追加された場合に便利です。**ldclt** ツールは、X を変数として使用し、指定された範囲でインクリメントして値を自動生成できます。

#### 例A.17 ランダムベース DN からのランダムバインド

```
# ldclt -h localhost -p 389 -e bindeach,bindonly -D "uid=XXXXX,dc=example,dc=com" -w
testXXXXX -e randombinddn,randombinddnlow=0,randombinddnhigh=99999
```

#### A.1.5.8. ランダムベース DN での操作の実行

ランダムに選択されたベース DN に対して、任意の操作を実行できます。3 つの **randombase** パラメーターは、選択元となる組織単位の範囲を設定します。**-b** ベースエントリの変数はベース DN の形式を設定します。

```
-b "ou=DeptXXX,dc=example,dc=com" -e randombase,randombaselow=0,randombasehigh=999 ...
```

#### A.1.5.9. TLS 認証

セキュアな認証およびセキュアな接続のパフォーマンスをテストするために、すべての操作を TLS 経由で実行できます。TLS 認証には 2 つのパラメーターが必要です。

- Directory Server のセキュリティーデータベースへのパスを指定する接続パラメーター **-Z**
- TLS データベースにアクセスするためにサーバーが要求する情報が含まれる実行パラメーター **cltcertname**、**keydbfile**、および **keydbpin**

たとえば、これは TLS を介してバインドテストを実行します。

```
# ldclt -h host -p port -e bindeach,bindonly -Z certPath -e
cltcertname=certName,keydbfile=filename,keydbpin=password
```

#### A.1.5.10. 破棄操作

**-e abandon** パラメーターを開いてから、サーバーでの操作をキャンセルします。この操作はそれ自体で、または他のタイプの操作 (**-e add**、**-e esearch** など) と共に実行できます。

```
# ldclt -e abandon -h localhost -p 389 -D "cn=Directory Manager" -w secret -v -q -b
"ou=people,dc=example,dc=com"
```

## A.2. RSEARCH(検索ストレステスト)

**rsearch** ユーティリティーは、コマンドで設定されたパラメーターに従って、指定した Directory Server インスタンスに対するループで、同じ操作を迅速かつ繰り返し実行する複数のスレッドを開きます。

最も単純な **rsearch** は、検索操作のための複数のクライアント接続をエミュレートします。追加オプションを使用すると、**rsearch** を拡張して、検索操作と共に比較、変更、削除、バインド/バインド解除操作を実行できます。

このツールは操作のパフォーマンスも追跡し、平均結果の実行ストリームを出力します。



### 注記

**rsearch** テストの結果は、Directory Server とそのホストマシンのパフォーマンスによって自然に異なります。**Red Hat Directory Server Performance Tuning Guide**にあるように、最初にパフォーマンスのチューニングにより Directory Server およびマシンの設定を最適化します。

**rsearch** ユーティリティーは、**/usr/bin** ディレクトリーにあります。

## A.2.1. 構文

```
rsearch-Dbind_dn-wpassword-ssuffix-ffilter-hhost-pport-Sscope-b-u-L-N-v-y-q-l-m-M-d-c-
ifile_for_filters-BDN_or_uid_file-Aattributes-afile_of_attributes-n-osearch_time_limits-
jsample_interval-tthreads-Ttimelimit-V-Cnumber_of_samples-Rreconnect_interval-x-Wpassword-
Utext-\ or -H</pre

```

## A.2.2. オプション

表A.5 rsearch オプション

オプション	説明
-A attributes	検索要求と使用する属性のリストが含まれます。これは <b>-a</b> と併用できません。
-a file_of_attributes	検索要求で使用される属性のリストを含むファイルを参照します。各属性は、ファイルの個別の行に指定する必要があります。以下に例を示します。  [literal,subs="+quotes,verbatim"] .... attr1 attr2 ... ....  これは <b>-A</b> と併用できません。
-B DN_or_uid_file	サーバーにバインドするために使用される DN または UID のリストが含まれます。DN の場合、各エントリーには 2 つの行があります。1 つは DN と UID 用で (デフォルトパスワードとして使用されます)。  [literal,subs="+quotes,verbatim"] ....DN: dn UID: uid ... ....  単純な UID ファイルには、1 行に 1 つの UID があります。  [literal,subs="+quotes,verbatim"] ....UID: uid1 UID: uid2 ... ..

オプション	説明
-b	すべての操作の前にバインドするようにユーティリティーに指示します。
-C sample_numbers	ユーティリティーを取得して終了するサンプルの数を指定します。
-c	比較操作を指定します。これを使用する場合は、 <b>-B</b> オプションを使用する必要があります。
-D bind_dn	サーバーへの接続に使用する <b>rsearch</b> ユーティリティーのバインド DN を提供します。DN ファイル ( <b>-B -x</b> ) で他の ID が指定されていない場合、これはテストの実行に使用される ID です。
-d	削除操作を指定します。これを使用する場合は、 <b>-B</b> オプションを使用する必要があります。
-f filter	検索操作で使用する検索フィルターが含まれます。
-h host	接続する LDAP サーバーのホスト名を指定します。指定されていない場合、デフォルトは localhost です。
-i file	<p><b>-f</b> オプションで渡される検索フィルターに追加される名前が含まれるファイルを参照します。name ファイルはリストであり、各名前は別々の行にあります。以下に例を示します。</p> <p>[literal,subs="+quotes,verbatim"] .... joe jane ....</p> <p>このファイルで使用できるフィルターオプションは、たとえば、<b>-f "uid =%s"</b> です。これにより、<b>"uid=joe"</b> と <b>"uid=jane"</b> の両方のフィルターがランダムに使用されます。</p>
-j sample_interval	サンプルを収集する前に待機する間隔を秒単位で指定します。
-L	linger への接続を設定します。ユーティリティーが閉じられると接続が破棄されます。
-l	ユーティリティーの出力をログに記録します。
-M	インデックス付き属性 ( <b>telephonenumber</b> ) の変更操作を指定します。これには、 <b>-B</b> オプションが必要です。

オプション	説明
-m	インデックス付けされていない属性の変更操作を指定します ( <b>description</b> )。これには、 <b>-B</b> オプションが必要です。
-N	ツールは、他の操作を実行せずにサーバーにのみバインドすることを指定します。
-n	将来の使用のために予約されています。
-o search_time_limit	検索操作に使用する時間制限を秒単位で指定します。
-p port	Directory Server インスタンスへの接続に使用するポートを指定します。これを使用しない場合、デフォルトは 389 です。
-q	ツールを警告なしで実行します。
-R reconnect_interval	サーバーへの接続を切断し、指定された回数の検索後に再接続するようにユーティリティーに指示します ( <b>reconnect_interval</b> )。
-S scope	検索範囲を設定します。許可される値は、それぞれ 1 レベル、ベース、およびサブツリーに対応する 0、1、および 2 です。デフォルトは 2 です。
-s suffix	すべてのテストを実行する Directory Server の接尾辞を指定します。
-T timelimit	<b>rsearch</b> テストの合計時間制限を設定します。ユーティリティーがその制限に達すると、このツールは閉じます。
-t threads	ユーティリティーが開くスレッドの数を設定します。デフォルトでは 1 回です。
-U	バインドファイルで使用するフィルターを渡します。 <b>-x</b> を使用しない場合、このオプションは無視されます。デフォルト値は ' <b>uid=%s</b> ' です。
-u	ユーティリティーに、サーバーからバインドを解除しないように指示しますが、単に接続を閉じるように指示します。
-V	<b>rsearch</b> の結果の実行平均値を表示します。
-v	詳細モードでコマンドを実行します。



オプション	説明
-W	<b>-B</b> ファイルのアイデンティティーにバインドするために使用するパスワードを指定します。これが指定されていない場合、デフォルトは UID 値になります。
-x	ユーティリティーに対し、バインディングに <b>-B</b> ファイルの内容を使用するように指示します。これを使用しない場合は、 <b>-B</b> オプションよりも無視されません。
-y	テスト間の遅延なしでコマンドを実行します。
-\? or -H	ツールの使用方法を出力します。

### A.2.3. 使用方法

**rsearch** ユーティリティーは、LDAP 操作のパフォーマンスを測定するために使用できます。以下の例は、さまざまな一般的なテストシナリオで **rsearch** を使用方法を示しています。



#### 注記

**rsearch** にはフィルターやスコープなどの検索パラメーターの引数が必要ですが、これらの引数は空のままにして、他の種類の LDAP 操作のテストを実行できます。以下に例を示します。

```
# rsearch -D "cn=Directory Manager" -w secret -s "" -f ""
```

#### A.2.3.1. 許可される設定ファイル

多くの場合、**rsearch** ツールはコマンドラインに渡される情報を使用してサーバーに接続します。**rsearch** ツールは、渡された引数の代わりに 2 つの異なる設定ファイルを受け入れることができます。

- UID のリスト、または DN と UID の両方を含む **DN または UID** ファイル。DN/UID ファイルは、**rsearch** が複数のランダムに選択されたバインド ID を使用して接続できるようにします。操作テストは、バインド/バインド解除テストと組み合わせることができます。



#### 警告

コマンドは、ディレクトリーからすでに削除されている DN/UID ファイル内の ID でバインドを試みる可能性があるため、ランダムバインド ID を削除テストで使用しないでください。

DN/UID ファイルは、**-B** オプションとともに使用され、ファイルを渡し、次に操作オプション (**-c**、**-d**、**-m**、または **-x**) を渡します。

- **name** ファイル。指定の LDAP フィルターの一部として使用する名前のリストが含まれます。ファイル内のフィルターは、**-f** オプションで指定されたフィルターよりも複雑になります。フィルターファイルは、さまざまな検索テストの実行に使用できます。たとえば、フィルターが少ししかない、ツールがキャッシュから結果の取得を開始しますが、無効なフィルターを使用すると検索の失敗をテストすることができます。また、完全一致、複雑なフィルター、属性検索など、フィルターのパフォーマンスをテストすることもできます。フィルターファイルを使用する場合は、プレースホルダーの値で **-f** オプションを指定する必要があります。プレースホルダーは **cn=%s** などの属性値のみを置き換えるために使用できません。これは、フィルターファイルから属性値変数を取得するようにコマンドに指示します。プレースホルダーは、フィルター自体 (**-f "%s"**) を置き換えて、ファイルからランダムに選択されたフィルターを提供することもできます。

**-i** オプションは、検索フィルターに使用する名前ファイルを渡します。ファイル内のすべての行は、**-f** オプションで指定したフィルターに追加されます。これらの 2 つのオプションを同時に使用する方法は複数あります。

- 最も簡単なシナリオでは、**-f** オプションを空のままにするため、プレースホルダーだけになります。この場合、フィルターは **-i** オプションで渡されたファイルから直接提供されません。
- または、ファイルのエントリは単に名前のリストとなり、**-f** オプションにパーシャルフィルターを指定することもできます。たとえば、名前ファイルには UID のリスト (jsmith、bjensen、amorrow) を含めることができ、**-f** フィルターは **uid=** にすることができます。**rsearch** は、検索フィルターを完了する名前を自動的に追加します。

### A.2.3.2. rsearch の結果

定期的に (デフォルトでは 10 秒ごと)、**rsearch** は、スクリプトによって実行された操作の現在の移動平均を返します。

結果は最初に、**その間隔内に** 実行された操作の数を示します。括弧内の 2 つの比率は、1 秒あたりの合計操作数と、各操作に費やされた時間 (ミリ秒単位) を示しています (1 秒を操作の合計数で割った値に 1000 を掛けたもの)。

```
date timestamp - Rate: num_ops/thr (ops/sec = num ms/op), total: ops (number thr)
```

以下に例を示します。

```
# rsearch -D "cn=Directory Manager" -w password -s "ou=people,dc=example,dc=com" -f
"objectclass=%s" -i /home/filter.txt
rsearch: 1 threads launched.
```

```
20100209 20:20:40 - Rate: 65961.00/thr (6596.10/sec = 0.1516ms/op), total: 65961 (1 thr)
```

### A.2.3.3. 検索テスト

**rsearch** の主な用途は、検索テストです。検索パフォーマンスの測定は、任意の引数なしで、**rsearch** で必要な引数のみを使用して実行できます。

```
# rsearch -D bind_dn -w password -s suffix -f filter
```

オプションは、特定のパフォーマンスを測定したり、特定の環境を使用したりするために使用できます。

検索フィルター (コマンドラインまたは **-i** ファイルを含むファイル) は、さまざまな種類のインデックス付き属性をテストできます。

- ワイルドカードのないフィルターでは、完全一致するパフォーマンスが表示されます。
- ワイルドカードのあるフィルターにより、サブ文字列インデックスのパフォーマンスが提供されます。
- 演算子 (=、>=、<=、~=) を使用したフィルターは、近似インデックスのパフォーマンスを示します

#### 例A.18 基本検索

```
# rsearch -D "cn=test user,cn=config" -w secret -s "dc=example,dc=com" -f "sn=smith"
```

フィルターが1つしかなく、複数の検索操作があるため、キャッシュを行う基本的な検索では以下の引数を使用します。

- **-D**: バインドアイデンティティを提供します。
- **-w**: バインドパスワードを指定します。
- **-s**: 検索ターゲット (スコープ) を指定します。
- **-f**: 検索フィルターを提供します。

#### 例A.19 特定の属性の検索

```
# rsearch -D "cn=test user,cn=config" -w secret -s "dc=example,dc=com" -f "sn=%s" -i /home/filter.txt -A givenname,mail,uid
```

このコマンドは、必要な引数に加えて、**-A** オプションを使用して、エントリー内の3つの特定の属性を検索します。

**-f** フィルター オプションで **%s** 変数を使用する場合は、**-i filter\_file** オプションが必要です。

### A.2.3.4. 認証テスト

**rsearch** ユーティリティーは、(必須の) **-D** および **-w** 引数でユーザー DN とパスワードを使用して、サーバーにバインドします。認証のパフォーマンスをテストするために、これらの認証情報を空白のままにするには、ランダムに選択した認証情報のリストを渡すか、Directory Manager などの特別なユーザーに設定します。

#### 例A.20 匿名バインド

```
# rsearch -D "" -w "" -s "dc=example,dc=com" -f "sn=%s" -i /home/filter.txt
```

**-D** 引数および **-w** 引数には empty の値があるため、このツールにはサーバーへの接続に使用するバインド認証情報がありません。これにより、匿名バインドが開始します。

## 例A.21 ランダムなユーザー認証

```
# rsearch -D "" -w "" -s "dc=example,dc=com" -f "sn=%s" -i /home/filter.txt -B /home/uids.txt -x
```

**-D** および **-w** 引数で認証情報を使用する代わりに、**rsearch** ツールに対して、指定の UID または DN のリストからランダムバインド ID をプルするように指示できます。これには、以下の 2 つのオプションが必要です。

- **-B** はバインド ID のリストを含むファイルを参照します。UID ファイルの場合、これは UID のリストであり、1行に1つずつです。

```
UID: uid1
UID: uid2
...
```

DN の場合、各エントリーには 2 つの行があります。1つは DN と UID 用で (デフォルトパスワードとして使用されます)。

```
DN: dn
UID: uid
...
```

- **-x** は、ツールによる **-B** 引数のファイルの使用を強制します。

DN の場合、ツールは DN に DN、UID 行の DN 行をパスワードとして使用します。**-U** オプションは、エントリーの名前付け属性として UID 以外の属性を使用するようにツールに指示し、**-W** は別のパスワード (デフォルトでは UID) を渡します。

```
# rsearch -D "" -w "" -s "dc=example,dc=com" -f "sn=%s" -i /home/filter.txt -B /home/uids.txt -x -U "(cn=*)" -W newpassword
```

### A.2.3.5. 操作テストの変更

**rsearch** を使用すると、インデックス付きおよびインデックス化されていない 2 種類の属性で変更操作のパフォーマンスを測定できます。変更操作は、**-M** または **-m** オプションを使用して通知されます。変更操作を実行するエントリーのリストは、**-B** オプションを使用して渡されます。

#### 注記

変更操作を実行するには、以下の形式の DN ファイルが必要です。

```
DN: dn1
UID: uid1

DN: dn2
UID: uid2
...
```

**-b** オプションを使用すると、bind-modify 操作の各セットのレートが測定されます。**-b** オプションを使用しない場合は、バインド操作が1つだけあり、テストでは実行されたすべての変更操作の平均が表示されます。

#### 例A.22 インデックス付けされていない属性の操作を変更する

```
# rsearch -D "cn=test user,cn=config" -w secret -s "" -f "" -m -B /home/dns.txt -v
```

**unindexed** 属性に対する変更操作は、**-m** オプションを使用して実行されます。このコマンドは、DN ファイルから選択した各エントリーの **description** 属性で変更操作を実行します。

テストは **description** 属性をインデックス化しても正常に実行されるので、テストを実行する前に属性がインデックス化されていないことを確認してください。

#### 例A.23 インデックス付き属性の操作を変更する

```
# rsearch -D "cn=test user,cn=config" -w secret -s "" -f "" -M -B /home/dns.txt -v
```

**indexed** 属性に対する変更操作は、**-M** オプションを使用して実行されます。このコマンドは、DN ファイルから選択された各エントリーの **telephoneNumber** 属性に対して変更操作を実行します。

**telephoneNumber** 属性がインデックス化されていない場合でもテストが正常に実行されるため、テストを実行する前に属性がインデックス化されていることを確認します。

### A.2.3.6. 操作テストの比較

**-c** オプションを渡すと、**rsearch** を使用して **ldapcompare** 操作をテストできます。このツールは、**-B** オプションで指定した UID のリストに基づいて、UID 属性に対して比較操作を実行します。



#### 注記

比較操作を実行するには、形式が含まれる DN ファイルが必要です。

```
DN: dn1
UID: uid1

DN: dn2
UID: uid2
...
```

#### 例A.24 比較処理

```
# rsearch -D "cn=test user,cn=config" -w secret -s "" -f "" -c -B /home/dns.txt -v
```

**-c** 引数は、比較操作を実行するようにコマンドに指示します。これは必須です。他の2つの引数は、比較操作のパフォーマンスを測定するのに便利です。

- **-B** (**-x** なし)。サーバーが比較操作を実行できるエントリーのリストを提供します。
- **-v**: **rsearch** を冗長モードで実行し、各バインド試行と比較操作の結果を出力します。

### A.2.3.7. 操作テストの削除

削除のパフォーマンステストには1つのオプションのみ必要です。-d は、削除操作を実行するコマンドに指示します。他の操作と同様に、-B 引数を使用して、ランダムに選択および削除されるエントリーのリストを含むファイルを渡すことができます。



#### 注記

コマンドは、すでに削除されている ID を使用してサーバーにバインドしようとする可能性があるため、削除操作で -B-x オプションのペアを使用しないでください。

#### 例A.25 操作の削除

```
# rsearch -D "cn=test user,cn=config" -w secret -s "" -f "" -d -B /home/dns.txt
```

-B 引数を使用して削除可能なエントリーのリストを提供する場合は、以下の形式の DN ファイルである必要があります。

```
DN: dn1
UID: uid1

DN: dn2
UID: uid2
...
```

### A.2.3.8. 時間制限の変更

パフォーマンステストが多数ある場合と同様に、**rsearch** には複数の時間ベースのメトリックがありません。

- 1ラウンドの統計を収集するために操作が実行される期間 (デフォルトでは 10 秒)
- ツールの実行時間 (デフォルトでは、無期限)
- ツールがサーバーへの接続を維持する期間 (デフォルトでは、無期限)

3つの制限時間はすべてリセットできます。

#### 例A.26 操作間隔の設定

```
# rsearch -D "cn=test user,cn=config" -w secret -s "dc=example,dc=com" -f "cn=%s" -i /home/filter.txt -b -j 20
```

**rsearch** ツールは、即時間隔で実行される操作の結果を出力します。デフォルトの間隔は 10 秒であるため、出力のすべての行は、前の 10 秒間に実行された操作の統計を表します。この間隔は、-j オプションを使用して変更できます。

これにより、テスト間隔が 20 秒にリセットされます。

### 例A.27 テスト時間制限の設定

```
# rsearch -D "cn=test user,cn=config" -w secret -s "dc=example,dc=com" -f "cn=%s" -i
/home/filter.txt -b -T 600

...

20100210 18:36:21 - Rate: 68561.00/thr (6856.10/sec = 0.1459ms/op), total: 68561 (1 thr)
20100210 18:36:31 - Rate: 78016.00/thr (7801.60/sec = 0.1282ms/op), total: 78016 (1 thr)
Final Average rate: 7328.85/sec = 0.1364msec/op, total: 78016
```

通常、コマンドは、コマンドが中断されるまで無期限に実行されます。**-T** オプションは、テストを実行して正常に終了するための時間制限 (秒単位) を設定します。ツールが終了すると、すべてのテスト実行間隔の平均の最終的な要約が出力されます。

### 例A.28 再接続間隔の設定

```
# rsearch -D "cn=test user,cn=config" -w secret -s "dc=example,dc=com" -f "cn=%s" -i
/home/filter.txt -b -R 30
```

このツールは、通常、サーバーへの接続を1つ開きます。reconnect オプションの **-R** は、ツールが Directory Server に再接続する間隔を設定します。

## A.2.3.9. 任意の操作によるテストのバインド

バインドおよびバインド解除のレートは、**rsearch** によって測定される任意の操作 (検索、変更、削除、比較) で確認できます。これには、すべての操作でサーバーにバインドするようにツールに指示する1つのオプション **-b** が必要です。

バインドテストでは、他に2つの属性を使用できます。**-L** (ツールを長持ちさせる) と **-N** (ツールに、他の操作を実行せずにバインドおよびバインド解除するように指示する) です。

### 例A.29 すべての操作でのバインドとバインド解除

```
# rsearch -D "cn=test user,cn=config" -w secret -s "dc=example,dc=com" -f "cn=%s" -i
/home/filter.txt -b -L
```

**rsearch** によって実行されるすべての操作に対して、バインド操作とバインド解除操作を開始するために2つのオプションが使用されます。

- **-b** (必須)
- **-L** (推奨)

**-f** フィルター オプションで **%s** 変数を使用する場合は、**-i filter\_file** オプションが必要です。

### 例A.30 匿名バインド操作のテスト

```
# rsearch -D "" -w "" -s "" -f "" -N -b -L
```

匿名バインドレートをテストするには、**-b** オプションを使用して、**-D** オプションおよび **-w** オプションの値を空のままにしてください。**-N** オプションは、コマンドがバインドおよびバインド解除操作のみを試みるようにします。

#### 例A.31 ランダムバインド操作のテスト

```
# rsearch -D "" -w "" -s "" -f "" -B /home/uids.txt -x -N -b -L
```

匿名バインド操作と同様に、必要な引数を空白のままにすることができます。**-N** オプションは、コマンドがバインドおよびバインド解除操作のみを試行することを保証し、**-B** および **-x** オプションは、コマンドが選択するランダムなバインド認証情報のリストを提供します。

#### 例A.32 バインド操作でフィルターを使用したテスト

```
# rsearch -D "" -w "" -s "" -f "" -B /home/uids.txt -x -U "(uid=*son)" -N -b -L
```

通常、バインドファイル (UID または DN) に含まれる任意の ID をバインドテストに使用できます。デフォルトのフィルターは **(uid=%s)** で、これはすべての ID エントリーにあります。ファイル内の ID のサブセットのみを使用するには、**-U** オプションを使用して代替フィルターを渡すことができます。

### A.2.3.10. 複数スレッドテストの実行

#### 例A.33 複数のスレッド

```
# rsearch -D "cn=test user,cn=config" -w secret -s "dc=example,dc=com" -f "sn=%s" -i /home/filter.txt -t 5
```

デフォルトでは、**rsearch** は操作のために1つのスレッドを開きます。**-t** オプションを使用すると、複数のスレッドを開くことができます。

[3] 追加操作と同様に、親がツールで最初に参照される際に親エントリーが作成されますが、追加操作を求めたエントリーは **作成されません**。



## 付録B レプリカ合意のステータス

各レプリカ合意の読み取り専用 **nsds5replicaLastUpdateStatus** 属性に、Directory Server は、契約の最新ステータスを表示します。以下は、可能なステータスのリストです。

### 無効な合意

レプリカ合意が無効になると、**nsds5replicaLastUpdateStatus** パラメーターが更新されなくなり、以下のステータスを表示できます。

- サーバーの起動時に、レプリカ合意はすでに無効になっています。

Error (0) No replication sessions started since server startup

- 実行時に契約が無効になりました。

Error (0) Replica acquired successfully: agreement disabled

### 一般契約状況

{blank}

- レプリケーションアグリーメントが停止されました。

Error (0) Replica acquired successfully: Protocol stopped

- 増分更新が開始されました。

Error (0) Replica acquired successfully: Incremental update started

- 増分更新が成功しました。

Error (0) Replica acquired successfully: Incremental update succeeded

- レプリケーションは成功しましたが、コンシューマーはセッションを終了して、別のサプライヤーに取得できるようにしました。

Error (0) Replica acquired successfully: Incremental update succeeded and yielded

### ACQUIRING\_REPLICA 状態のエラーメッセージ

レプリケーションセッションの最初の部分で、サプライヤーはコンシューマーを取得し、接続を確立し、コンシューマーにバインドし、コンシューマーが別のサプライヤーによってまだ更新されていないことを確認し、追加のチェックを実行します。この状態では、次のエラーコードが表示される可能性があります。

- コンシューマーとの接続を確立する際の失敗:

Error (**result\_code**) Problem connecting to replica - LDAP error: **ldap\_error\_message**

Error (**result\_code**) Problem connecting to replica (SSL not enabled) - LDAP error: **ldap\_error\_message**

結果コードとエラーメッセージは、接続を確立できなかった理由を示しています。

- コンシューマーで内部エラーが発生しました。

**Error (8) :Failed to acquire replica: Internal error occurred on the remote replica**

このエラーは、コンシューマーの変更シーケンス番号 (CSN) ジェネレーターに関連する障害が原因で発生します。詳細は、コンシューマーログファイルを参照してください。

- コンシューマーへの認証に使用された ID は、有効なレプリケーションバインド識別名 (DN) でも、バインド DN グループのメンバーでもありませんでした。

**Error (3) :Unable to acquire replica: permission denied. The bind dn does not have permission to supply replication updates to the replica. Will retry later.**

- コンシューマーの接尾辞に有効なレプリカが定義されていません。

**Error (6) :Unable to acquire replica: there is no replicated area on the consumer server. Replication is aborting.**

- コンシューマーに送信されたレプリケーション制御のデコードエラー:

**Error (4) :Unable to acquire replica: the consumer was unable to decode the startReplicationRequest extended operation sent by the supplier. Replication is aborting.**

- レプリカは現在、別のサプライヤーによって更新されています。

**Error (1) :Unable to acquire replica: the replica is currently being updated by another supplier.**

- サプライヤーとコンシューマーは同じレプリカ ID を使用します。

**Error (11) :Unable to acquire replica: the replica has the same Replica ID as this one. Replication is aborting.**

サプライヤーまたはコンシューマーが正しく設定されていません。問題を修正するには、レプリケーション設定で一意的レプリカ ID を設定します。

- サプライヤーは **backoff** モードに設定されました。

**Error (14) :Unable to acquire replica: the replica instructed us to go into backoff mode. Will retry later.**

この状態は、カスタムレプリケーションフックが実装されている場合にのみ表示されます。

- コンシューマーから受信したレプリケーション制御のデコードエラー:

**Error (extop\_result) :Unable to acquire replica**

**Error (4) Unable to parse the response to the startReplication extended operation. Replication is aborting.**

Error (16) Unable to receive the response for a startReplication extended operation to consumer. Will retry later.

Error (0) Unable to obtain current CSN. "Replication is aborting.

### SENDING\_UPDATES 状態のエラーメッセージ

レプリカが正常に取得された後、セッションは更新の送信を開始します。この状態では、それぞれの手順で次のメッセージを表示できます。

#### a. レプリカ更新ベクトル (RUV) の調査:

- レプリカに更新ベクターが設定されていないか、コンシューマーでレプリケーションが有効にされていません。

Error (19) : Replica is not initialized

- コンシューマーは、サプライヤーと同じデータベース生成を使用して初期化されませんでした。

Error (19) : Replica has different database generation ID, remote replica may need to be initialized

問題を修正するには、サプライヤーまたはコンシューマーのいずれかを初期化します。

#### b. 状態変化番号 (CSN) ジェネレーターの更新:

- ローカルサーバーと削除サーバーの時間差が大きすぎます。

Error (2) : fatal error - too much time skew between replicas

- Directory Server は CSN ジェネレーターの更新に失敗しました。

Error (2) : fatal internal error updating the CSN generator

#### c. 初期変更ログの配置:

- changelog を処理できない場合の一般的なエラー:

Error (15) : Unexpected format encountered in changelog database

このエラーは、たとえば changelog ファイルへのパスが存在しない場合はログに記録されます。

- 変更ログのエントリーの解析に失敗しました。

Error (15) : Unexpected format encountered in changelog database

- 変更ログのデータベース層に関連するエラー:

Error (15) : Changelog database was in an incorrect state

Error (15) : Incorrect dbversion found in changelog database

Error (15) : Changelog database error was encountered

詳細は、`/var/log/dirsrv/slaped-instance_name/errors` ログファイルを参照してください。

- Directory Server はメモリーの割り当てに失敗しました。

Error (15) : changelog memory allocation error occurred

このエラーは、changelog バッファまたは changelog イテレーターがメモリーの割り当てに失敗した場合などにログに記録されます。

- サプライヤーはコンシューマーよりも先にあり、更新を送信することができますが、changelog で開始点を見つけることはできません。

Error (15) : Data required to update replica has been purged from the changelog. "The replica must be reinitialized.

Error (15) : Changelog data is missing

Directory Server はこれらのエラーを致命的として扱いますが、コンシューマーが別のサプライヤーから更新を受け取る場合は解決できます。この場合、一時的なものとして扱われます。

d. 次の更新を送信します。

- 結果スレッドの作成に失敗しました。

Error (**result\_code**) : Failed to create result thread

結果コードは、スレッドが作成されなかった理由を示します。

- changelog を処理できない場合の一般的なエラー:

Error (15) : Invalid parameter passed to cl5GetNextOperationToReplay

このエラーは、たとえば changelog ファイルへのパスが存在しない場合はログに記録されます。

- 変更ログの読み取り中にデータベースエラーが発生しました。

Error (15) : Database error occurred while getting the next operation to replay

このイベントは、Directory Server がロックされたデータベースページにアクセスする場合などに記録されます。

- Directory Server は作成を実行しました。

Error (15) : Memory allocation error occurred (cl5GetNextOperationToReplay)

e. サブエントリーの更新:

- **replica keep alive** エントリーが失敗しました。

Error (-1) : Agreement is corrupted: missing suffix

**SEND\_UPDATES** 状態の一般的なステータス:

- 変更ログの処理中に、ローカルサーバーで致命的でないエラーが発生しました。

Error (18) : Incremental update transient error. Backing off, will retry update later.

詳細は、`/var/log/dirsrv/slaped-instance_name/errors` ファイルを参照してください。

- レプリケーション接続は、接続が確立された後に切断されました。

Error (16) : Incremental update connection error. Backing off, will retry update later.

- 既存のレプリケーション接続でタイムアウトが表示されます。

Error (17) : Incremental update timeout error. Backing off, will retry update later.

レプリケーションは自動的に再開しようとします。

## 用語集

### アクセス制御命令 (ACI)

ディレクトリー内のエントリーへのアクセス許可を付与または拒否する命令。

### アクセス制御リスト (ACL)

ディレクトリーへのアクセスを制御するためのメカニズム。

### アクセス権

アクセス制御のコンテキストで、許可または拒否されるアクセスのレベルを指定します。アクセス権は、ディレクトリーで実行できる操作の種類に関連しています。次の権限を付与または拒否できます: read、write、add、delete、search、compare、selfwrite、proxy、および all

### アカウントの非アクティブ化

すべての認証の試行が自動的に拒否されるよう、ユーザーアカウント、アカウントグループ、またはドメイン全体を無効にします。

### すべての ID しきい値

サーバーによって管理されるすべてのインデックスキーにグローバルに適用されるサイズ制限。個々の ID リストのサイズがこの制限に達すると、サーバーはその ID リストをすべての ID トークンに置き換えます。

### すべての ID トークン

サーバーがすべてのディレクトリーエントリーがインデックスキーと一致すると想定するメカニズム。実際には、All ID トークンにより、サーバーは検索要求に使用できるインデックスがないかのように動作します。

### 匿名アクセス

付与されると、バインドの条件に関係なく、認証情報を提供せずに誰でもディレクトリー情報にアクセスできるようになります。

### 概算インデックス

効率的な近似またはのような検索を可能にします。

## 属性

エントリーに関する説明情報を保持します。属性にはラベルと値があります。各属性は、属性値として格納できる情報のタイプの標準構文にも準拠しています。

## 属性リスト

特定のエントリータイプまたはオブジェクトクラスの必須属性と任意の属性のリスト。

## ディレクトリーサーバーの認証

パススルー認証 (PTA) では、認証 Directory Server は、要求元のクライアントの認証認証情報を含む Directory Server です。PTA 対応のホストは、クライアントから受信した PTA 要求をホストに送信します。

## 認証証明書

譲渡および偽造が不可能であり、第三者によって発行されたデジタルファイル。認証証明書は、相手を検証および認証するために、サーバーからクライアントまたはクライアントからサーバーに送信されます。

## base DN

基本識別名。検索操作は、ベース DN、エントリーの DN、およびディレクトリーツリー内のその下にあるすべてのエントリーに対して実行されます。

## 基本識別名 (バインド DN)

操作の実行時に Directory Server への認証に使用される識別名。

## バインド識別名 (バインドルール)

アクセス制御のコンテキストでは、バインドルールは、ディレクトリー情報にアクセスするのに特定のユーザーまたはクライアントが満たさなければならない認証情報と条件を指定します。

## ブランチエントリー

ディレクトリー内のサブツリーの最上位を表すエントリー。

## ブラウザ

HTML ファイルとして保存された World Wide Web 資料を要求および表示するのに使用される Mozilla Firefox などのソフトウェア。ブラウザは HTTP プロトコルを使用してホストサーバーと通信します。

## 参照インデックス

Directory Server コンソールのエントリーの表示を高速化します。表示パフォーマンスを向上させるために、ディレクトリーツリーの任意の分岐点に参照インデックスを作成できます。

## 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、ハブサプライヤーと呼ばれることが多い1つのサーバーが、特定のレプリカのコンシューマーとサプライヤーの両方として機能します。読み取り専用のレプリカを保持し、変更ログを維持します。データのサプライヤ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信し、それらの更新をコンシューマーに提供します。

## certificate

ネットワークユーザーの公開鍵をディレクトリー内の DN に関連付けるデータのコレクション。証明書は、ユーザーオブジェクト属性としてディレクトリーに保存されます。

## 認証局

認証証明書を販売および発行する会社または組織。認証証明書は、信頼する認証局から購入できません。CA としても知られています。

## CGI

共通のゲートウェイインターフェイス外部プログラムが HTTP サーバーと通信するためのインターフェイス。CGI を使用するよう書かれたプログラムは CGI プログラムまたは CGI スクリプトと呼ばれ、多くの一般的なプログラミング言語で記述できます。CGI プログラムはフォームを処理したり、サーバー自体が実行していない出力解析を実行します。

## チェーン

別のサーバーにリクエストをリレーする方法です。リクエストの結果が収集され、コンパイルされてからクライアントに戻ります。

changelog: changelog は、レプリカで発生した変更を記述するレコードです。次に、サプライヤーサーバーは、マルチサプライヤーレプリケーションの場合には、レプリカ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。

## 文字タイプ

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、大文字から小文字へのマッピング。

## ciphertext

情報を復号化するための適切なキーがないと誰も読み取れない暗号化された情報。

## クラス定義

特定のオブジェクトのインスタンスを作成するために必要な情報を指定し、ディレクトリー内の他のオブジェクトとの関係でオブジェクトがどのように機能するかを決定します。

## serviceclassic CoS のクラス

従来の CoS は、DN とターゲットエントリーのいずれかの属性の値の両方で、テンプレートエントリーを特定します。

## clientcode ページ

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に関連付けるために使用する国際化プラグインのコンテキストでロケールによって使用される内部テーブル。

## 照合順序

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。この情報には、アルファベットの文字の順序や、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が含まれる場合があります。

## コンシューマー

サプライヤーサーバーから複製されたディレクトリーツリーまたはサブツリーを含むサーバー。

## コンシューマーサーバー

レプリケーションのコンテキストでは、別のサーバーからコピーされるレプリカを保持するサーバーは、そのレプリカのコンシューマーと呼ばれます。

## CoS

アプリケーションに表示される方法でエントリー間で属性を共有する方法。

## CoS 定義エントリー

使用している CoS のタイプを識別します。これは、影響を与えるブランチの下に LDAP サブエントリーとして保存されます。

## CoS テンプレートエントリー

共有属性値のリストが含まれます。

## daemon

特定のシステムタスクを行う Unix マシンのバックグラウンドプロセス。デーモンプロセスは、機能を継続するために人間の介入を必要としません。

## DAP

ディレクトリーアクセスプロトコル。ディレクトリーへのクライアントアクセスを提供する ISOX.500 標準プロトコル。

## データベースリンク

チェーンの実装。データベースリンクはデータベースのように動作しますが、永続ストレージはありません。代わりに、リモートに保存されているデータを指します。

## デフォルトインデックス

データベースインスタンスごとに作成されるデフォルトインデックスのセット。一部のプラグインはそれらに依存する可能性があるため、デフォルトのインデックスは削除する前に行う必要があります。

## ディレクトリーツリー

ディレクトリーに保存されている情報の論理表現。これは、ほとんどのファイルシステムで使用されるツリーモデルをミラーリングし、ツリーのルートポイントが階層の上部に表示されます。DITとも呼ばれます。

## Directory Manager

特権データベース管理者 (UNIX の root ユーザーと比較)。アクセス制御は Directory Manager には適用されません。

## ディレクトリーサービス

組織内の人やリソースに関する説明的な属性ベースの情報を管理するために設計されたデータベースアプリケーション。

## 識別名

LDAP ディレクトリーのエントリーの名前と場所の文字列表現。

## DNS

ドメイン名システム。ネットワーク上のマシンが標準 IP アドレス (198.93.93.10 など) をホスト名 (**www.example.com** など) に関連付けるために使用するシステム。マシンは通常、DNS サーバーからホスト名の IP アドレスを取得するか、システムで維持されているテーブルで検索します。

## DNS エイリアス

DNS エイリアスは、DNS サーバーが別のホストを指していることを認識しているホスト名です。マシンは常に1つの実際の名前を持ちますが、1つ以上のエイリアスを持つことができます。たとえば、**www.yourdomain.domain** などのエイリアスは、サーバーが現在存在する実際マシン (**realthing.yourdomain.domain**) を参照することがあります。

## エントリー

オブジェクトに関する情報が含まれる LDIF ファイルの行グループ。

## エントリー分布

多数のエントリーをサポートするようにスケーリングするために、ディレクトリーエントリーを複数のサーバーに分散する方法。

## エントリー ID リスト

ディレクトリーが使用する各インデックスは、インデックスキーと一致するエントリー ID リストの表で設定されます。エントリー ID リストは、クライアントアプリケーションの検索要求に一致する候補エントリーのリストをビルドするためにディレクトリーによって使用されます。

## 等価インデックス

特定の属性値を含むエントリーを効率的に検索できます。

## ファイル拡張子

通常、ファイルのタイプ (.GIF や .HTML など) を定義するピリオドまたはドット (.) の後のファイル名のセクション。ファイル名 **index.html** では、ファイル拡張子は **html** です。

## ファイルのタイプ

指定のファイルの形式。たとえば、グラフィックファイルは GIF 形式で保存されますが、テキストファイルは通常 ASCII テキスト形式として保存されます。ファイルタイプは通常、ファイル拡張子 (.GIF や .HTML など) で識別されます。

## filter

返される情報を制限するディレクトリークエリーに適用される制約。



フィルターされたロール: 各エントリーに含まれる属性に応じて、エントリーをロールに割り当てることができます。これを行うには、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、そのロールを持っていると言われます。

### 一般的なアクセス

付与されると、認証されたすべてのユーザーがディレクトリー情報にアクセスできることを示します。

### GSS-API

一般的なセキュリティーサービス。UNIX ベースのシステムが Kerberos サービスにアクセスして認証する方法のネイティブのアクセスプロトコルで、セッション暗号化にも対応しています。

### ホスト名

machine.domain.dom 形式のマシンの名前。これは IP アドレスに変換されます。たとえば、**www.example.com** は、サブドメイン **example** および **com** ドメインのマシン **www** です。

### HTML

ハイパーテキストマークアップ言語。WorldWideWeb 上のドキュメントに使用されるフォーマット言語。HTML ファイルは、Mozilla Firefox などのブラウザに、テキストの表示方法、グラフィックの配置方法、およびフォームアイテムの表示方法、および他のページへのリンクの表示方法を指示するフォーマットコードを含むプレーンテキストファイルです。

### HTTP

Hypertext Transfer Protocol。HTTP サーバーとクライアント間で情報を交換する方法。

### HTTPD

HTTP デモンまたはサービスの省略形。HTTP プロトコルを使用して情報を提供するプログラムです。デモンまたはサービスは、しばしば httpd と呼ばれます。

### HTTPS

Secure Sockets Layer (SSL) を使用して実装された HTTP の安全なバージョン。

### hub

レプリケーションのコンテキストでは、別のサーバーからコピーされたレプリカを保持し、次にそれを 3 番目のサーバーにレプリケートするサーバー。

### ID リストのスキャン制限

インデックス化された検索操作にグローバルに適用されるサイズ制限。個々の ID リストのサイズがこの制限に達すると、サーバーはその ID リストをすべての ID トークンに置き換えます。

### インデックスキー

ディレクトリーが使用する各インデックスは、インデックスキーと一致するエントリー ID リストの表で設定されます。

### 間接的な CoS

間接的な 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。

### 国際インデックス

国際ディレクトリーでの情報の検索を高速化します。

### IP アドレス

インターネット上のマシンの実際の場所を指定する、ドットで区切られた一連の数字 (たとえば、198.93.93.10)。

### ISO

国際標準化機構。

### ナレッジ参照

異なるデータベースに保存されているディレクトリー情報へのポインター。

## LDAP

Lightweight Directory Access Protocol。TCP/IP および複数のプラットフォームで実行するためのディレクトリーサービスプロトコル。

## LDAPv3

Directory Server がスキーマ形式をベースとする LDAP プロトコルのバージョン 3。

## LDAP クライアント

LDAP Directory Server から LDAP エントリーを要求および表示するために使用されるソフトウェア。

## LDAP URL

DNS を使用して Directory Server を見つけ、LDAP を使用してクエリーを完了する手段を提供します。LDAP URL の例は `ldap://ldap.example.com` です。

## LDBM データベース

割り当てられた全データが含まれる大容量ファイルのセットで設定される高パフォーマンスのディスクベースのデータベース。Directory Server のプライマリーデータストア。

## LDIF

LDAP データ交換形式 Directory Server エントリーをテキスト形式で表現するために使用される形式。

## Leaf エントリー

他のエントリーがないエントリー。leaf のエントリーは、ディレクトリーツリー内のブランチポイントにすることはできません。

## locale

特定の地域、文化、および習慣のユーザーにデータを提示するために使用される照合順序、文字タイプ、通貨形式、および日時形式を識別します。これには、特定の言語のデータがどのように解釈、保存、または照合されるかに関する情報が含まれます。ロケールは、特定の言語を表すために使用するコードページも示します。

## 管理オブジェクト

SNMP エージェントがアクセスして NMS に送信する標準値。各管理オブジェクトは、公式の名前とドットのアノテーションで表現される数値の識別子で識別されます。

## マネージドロール

明示的に列挙されたメンバーのリストの作成を許可します。

## マッピングツリー

接尾辞 (サブツリー) の名前をデータベースに関連付けるデータ構造。

## マッチングルール

検索操作時にサーバーが文字列を比較する方法についてのガイドラインを提供します。国際検索では、マッチングルールによって、使用する照合順序と演算子がサーバーに指示します。

## MD5

メッセージ RSA Data Security, Inc. によるメッセージダイジェストアルゴリズム。これは、高い確率で一意で、生成が非常に困難であるデータの短いダイジェストを生成するために使用できます。

## MD5 署名

MD5 アルゴリズムにより生成されるメッセージダイジェスト。

## MIB

管理情報ベース。SNMP ネットワークに関連付けられているすべてのデータまたはその一部。MIB は、すべての SNMP マネージドオブジェクトの定義を含むデータベースと考えることができます。MIB にはツリーのような階層があり、最上位レベルにはネットワークに関する最も一般的な情報が含まれ、下位レベルには特定の個別のネットワーク領域が処理されます。

## MIB 名前空間

管理情報ベースの名前空間。ディレクトリーデータに名前を付けて参照するための手段。ディレクトリーツリーとも呼ばれます。

## 通貨形式

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。

## マルチサプライヤーのレプリケーション

2 台のサーバーがそれぞれ同じ読み取り/書き込みレプリカのコピーを保持する高度なレプリケーションシナリオ。各サーバーは、レプリカの変更ログを維持します。一方のサーバーで行われた変更は、もう一方のサーバーに自動的に複製されます。競合が発生した場合は、タイムスタンプを使用して、最新バージョンを保持しているサーバーを判別します。

## multiplexor

リモートサーバーと通信するデータベースリンクを含むサーバー。

## n+1ディレクトリーの問題

異なるディレクトリーにある同じ情報の複数のインスタンスを管理する問題。その結果、ハードウェアと人件費が増加します。

## 名前の競合

同じ識別名を持つ複数のエントリー。

## ネストされたロール

他のロールを含むロールの作成を許可します。

## ネットワーク管理アプリケーション

SNMP 管理デバイスの情報をグラフィカルに表示する Network Management Station コンポーネントは、どのデバイスがダウンしているか、受信したエラーメッセージの数などを表示します。

## NIS

ネットワーク情報サービス Unix マシンが、コンピューターのネットワーク全体でマシン、ユーザー、ファイルシステム、およびネットワークパラメーターに関する特定の情報を収集、照合、および共有するために使用するプログラムとデータファイルのシステム。

## NMS

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。

## ns-slapd

Directory Server のすべてのアクションを担当する Red Hat の LDAP DirectoryServer デーモンまたは Directory Server。

## オブジェクトクラス

エントリーに含まれる属性を定義して、ディレクトリー内のエントリータイプを定義します。

## オブジェクト識別子

オブジェクト指向システムで、オブジェクトクラスや属性などのスキーマ要素を一意に識別する文字列 (通常は 10 進数)。オブジェクト識別子は、ANSI、IETF、または同様の組織によって割り当てられます。

## 運用上の属性

変更とサブツリーのプロパティを追跡するためにディレクトリーによって内部的に使用される情報が含まれています。明示的に要求されない限り、検索に回答して操作属性は返されません。

## 親アクセス

付与されると、バインド DN がターゲットエントリーの親である場合、ユーザーがディレクトリーツリー内の自分より下のエントリーにアクセスできることを示します。

## パススルーサブツリー

パススルー認証では、このサブツリーに含まれるすべてのクライアントから PTA Directory Server が認証する Directory Server にバインド要求を渡します。

### パスワードファイル

Unix ユーザーのログイン名、パスワード、およびユーザー ID 番号を格納する Unix マシン上のファイル。保持される場所から、`/etc/passwd` とも呼ばれます。

### パスワードポリシー

特定のディレクトリーでのパスワードの使用方法を管理する一連のルール。

### permission

アクセス制御のコンテキストでは、アクセス許可は、ディレクトリー情報へのアクセスが許可または拒否されるかどうか、および許可または拒否されるアクセスのレベルを示します。

### PDU

SNMP デバイス間のデータ交換の基礎を形成するエンコードされたメッセージ。

### 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。

### 存在インデックス

特定のインデックス付き属性を含むエントリーの検索を許可します。

### protocol

ネットワーク上のデバイスが情報を交換する方法を説明する一連のルール。

### プロトコルデータユニットプロキシ認証

ディレクトリーへのアクセスを要求するユーザーが自身の DN ではなくプロキシ DN にバインドする特殊な形式の認証。

### プロキシ DN

プロキシ認証で使用されます。プロキシ DN は、クライアントアプリケーションが操作を実行しようとしているターゲットへのアクセス許可を持つエントリーの DN です。

### PTA

あるディレクトリーサーバーが別の Directory Server に問い合わせるバインド認証情報を確認するメカニズム。

### PTA ディレクトリーサーバー

パススルー認証 (PTA) では、PTA Directory Server は、認証する Directory Server に受信するバインド要求を送信するサーバーです。

### PTA LDAP URL

パススルー認証において、認証 Directory Server、パススルーサブツリー、およびオプションのパラメーターを定義する URL です。

### RAM

ランダムアクセスメモリー。コンピューターの物理的な半導体ベースのメモリー。コンピューターをシャットダウンすると、RAM に保存されている情報は失われます。

### rc.local

マシンの起動時に実行されるプログラムを記述した Unix マシン上のファイル。その場所から `/etc/rc.local` とも呼ばれます。

### RDN

エントリーの祖先が文字列に追加されて完全な識別名が形成される前の、実際のエントリー自体の名前。

### 参照の整合性

関連するエントリー間の関係がディレクトリー内で維持されることを保証するメカニズム。

## 読み取り専用レプリカ

すべての更新操作を読み取り/書き込みレプリカに参照するレプリカ。サーバーは、読み取り専用のレプリカをいくつでも保持できます。

## 読み取り/書き込みレプリカ

ディレクトリー情報の書き込み可能なコピーを含み、更新可能なレプリカ。サーバーは、任意の数の読み取り/書き込みレプリカを保持できます。

## replica

レプリケーションに参加するデータベース。

## レプリケーション

ディレクトリーツリーまたはサブツリーをサプライヤーサーバーからレプリカサーバーにコピーする行為。

## レプリカ合意

サプライヤーサーバーに格納され、レプリケートするデータベース、データのプッシュ先のレプリカサーバー、レプリケーションが発生する可能性のある時間、サプライヤーがコンシューマーにバインドするためにサプライヤーが使用する DN と認証情報を識別する設定パラメーターのセット。接続がどのように保護されているか。

## RFC

コメントの要求インターネットコミュニティに送信された手順または標準ドキュメント。標準が受け入れられる前に、テクノロジーにコメントを送信できます。

## role

エントリーのグループ化メカニズム。各ロールには、ロールを持つエントリーである **メンバー** があります。

## ロールベースの属性

この属性は、エントリーが特定のロールを持ち、関連する CoS テンプレートを持っているために表示されます。

## root

Unix マシンで最も特権のあるユーザーが利用できます。root ユーザーには、マシン上のすべてのファイルに対する完全なアクセス権限が付与されます。

## root 接尾辞

1つ以上のサブ接尾辞の親。ディレクトリーツリーには、複数のルート接尾辞を含めることができます。

## SASL

ディレクトリーへのバインドを試みるクライアントの認証フレームワーク。

## schema

ディレクトリーにエントリーとして保存できる情報の種類を説明する定義。スキーマと一致しない情報がディレクトリーに保存されている場合、ディレクトリーにアクセスしようとするクライアントは適切な結果を表示できない可能性があります。

## スキーマチェック

ディレクトリーに追加または変更されたエントリーが、定義されたスキーマに準拠していることを確認します。スキーマチェックはデフォルトでオンになっており、スキーマに準拠していないエントリーを保存しようとする、ユーザーはエラーを受け取ります。

## Secure Sockets Layer self アクセス

付与されると、バインド DN がターゲットエントリーと一致する場合、ユーザーが自分のエントリーにアクセスできることを示します。

## サーバーデーモン

サーバーデーモンは、実行後に、クライアントからの要求をリッスンし、許可するプロセスです。

## サーバーサービス

Windows のプロセスは、実行後に、クライアントからの要求をリッスンして受け入れるプロセスです。Windows NT の SMB サーバーです。

## サーバーセレクター

ブラウザを使用してサーバーを選択および設定できるインターフェイス。

## service

特定のシステムタスクを行う Windows マシンのバックグラウンドプロセス。サービスプロセスは、機能を継続するために人間の介入を必要としません。

## SIE

サーバーインスタンスエントリー。インストール時に Directory Server のインスタンスに割り当てられた ID。

## 単一サプライヤーレプリケーション

複数のサーバーが最大 4 台までの最も基本的なレプリケーションシナリオでは、それぞれが同じ読み書きレプリカのコピーをレプリカサーバーに保持します。単一サプライヤーのレプリケーションシナリオでは、サプライヤーサーバーは変更ログを維持します。

## SIRslapd

LDAP Directory Server デーモンまたはレプリケーションを除くディレクトリーのほとんどの機能を担当するサービス。

## SNMP

ネットワークアクティビティーに関するデータを交換して、サーバーで実行しているアプリケーションプロセスを監視および管理するために使用されます。

## SNMP マスターエージェント

さまざまなサブエージェントと NMS との間で情報を交換するソフトウェアです。

## SNMP サブエージェント

マネージド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すソフトウェア。サブエージェントとも呼ばれます。

## SSL

HTTP のセキュリティー保護されたバージョンである HTTPS を実装するために使用される、2 つのシステム (クライアントとサーバー) 間の安全な接続を確立するためのソフトウェアライブラリー。Secure Sockets Layer と呼ばれます。

## 標準インデックス

デフォルトで維持されるインデックス。

## サブ接尾辞

root 接尾辞の下にあるブランチ。

## 部分文字列インデックス

エントリー内の部分文字列を効率的に検索できます。部分文字列のインデックスは、各エントリーの最小 2 文字に制限されます。

## 接尾辞

ディレクトリーツリーの最上部にあるエントリーの名前。その下にデータが保存されます。同じディレクトリー内で複数の接尾辞を使用できます。各データベースには 1 つの接尾辞しかありません。

## スーパーユーザー

Unix マシンで最も特権のあるユーザーが利用できます。root ユーザーには、マシン上のすべてのファイルに対する完全なアクセス権限が付与されます。これは root と呼ばれます。

## サプライヤー

レプリカサーバーに複製されるディレクトリーツリーまたはサブツリーの書き込み可能なコピーを含むサーバー。

### サプライヤーサーバー

レプリケーションのコンテキストでは、別のサーバーにコピーされた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。

### サプライヤーが開始したレプリケーション

レプリケーション設定。サプライヤーサーバーはディレクトリーデータを任意のレプリカサーバーに複製します。

### 対象暗号化 (symmetric encryption)

暗号化と復号化の両方に同じキーを使用する暗号化。DES は、対称暗号化アルゴリズムの一例です。

### システムインデックス

Directory Server の操作に不可欠であるため、削除または変更できません。

### target

アクセス制御のコンテキストでは、ターゲットは特定の ACI が適用されるディレクトリー情報を識別します。

### ターゲットエントリー

CoS のスコープ内のエントリー。

### TCP/IP

送信制御プロトコル/インターネットプロトコル。インターネットおよびエンタープライズ (企業) ネットワークのメインネットワークプロトコル。

### 時刻/日付の形式

特定のリージョン内の時刻と日付のカスタムフォーマットを示します。

### TLS

安全なソケット層の新しい標準。公開鍵ベースのプロトコルです。さらにトランスポート層セキュリティ。

### topology

ディレクトリーツリーが物理サーバー間で分割される方法、およびこれらのサーバーが相互にリンクする方法。

### uid

Unix システムの各ユーザーに関連付けられている一意の番号。

### URL

統一されたリソースロケーター。ドキュメントを要求するためにサーバーとクライアントが使用するアドレス指定システム。多くの場合、これは場所と呼ばれます。URL の形式は **protocol://machine:port/document** です。ポート番号は選択したサーバーでのみ必要であり、多くの場合サーバーによって割り当てられるため、ユーザーはポート番号を URL に配置する必要がありません。

### 仮想リストビューのインデックス

Directory Server コンソールのエントリーの表示を高速化します。表示パフォーマンスを向上させるために、ディレクトリーツリーの任意の分岐点に、仮想リストビューのインデックスを作成できます。

### X.500 標準

Directory Server の実装で使用される推奨情報モデル、オブジェクトクラス、および属性の概要を示す ISO/ITU-T ドキュメントのセット。

## 付録C 更新履歴

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。

### 11.5-1

2022 年 5 月 10 日 (火) Marc Muehlfeld ([mmuehlfeld@redhat.com](mailto:mmuehlfeld@redhat.com))

- Red Hat Directory Server 11.5 版のガイドをリリース

### 11.4-1

2021 年 11 月 9 日 (火) Marc Muehlfeld ([mmuehlfeld@redhat.com](mailto:mmuehlfeld@redhat.com))

- Red Hat Directory Server 11.4 版のガイドをリリース

### 11.3-1

2021 年 5 月 11 日 (火) Marc Muehlfeld ([mmuehlfeld@redhat.com](mailto:mmuehlfeld@redhat.com))

- Red Hat Directory Server 11.3 版のガイドをリリース

### 11.2-1

2020 年 11 月 03 日 (火) Marc Muehlfeld ([mmuehlfeld@redhat.com](mailto:mmuehlfeld@redhat.com))

- Red Hat Directory Server 11.2 版のガイドをリリース

### 11.1-1

2020 年 4 月 28 日 (火) Marc Muehlfeld ([mmuehlfeld@redhat.com](mailto:mmuehlfeld@redhat.com))

- Red Hat Directory Server 11.1 版のガイドをリリース

### 11.0-1

2019 年 11 月 5 日 (火) Marc Muehlfeld ([mmuehlfeld@redhat.com](mailto:mmuehlfeld@redhat.com))

- Red Hat Directory Server 11.0 版のガイドをリリース