opentext™

OpenText™ eDirectory CE 24.4 (v9.3) 管理ガイド

2025 年 1 月

保証と著作権

Copyright 2009 - 2024 Open Text All rights reserved.

Open Text、関連会社、およびライセンサ (「Open Text」) の製品およびサービスに対する保証は、当該製品およびサービスに付属する保証書に明示的に規定されたものに限られます。本書のいかなる内容も、当該保証に新たに保証を追加するものではありません。Open Text は、本書に技術的または編集上の誤りまたは不備があっても責任を負わないものとします。本書の内容は、将来予告なしに変更されることがあります。

目次

本書およびライブラリについて 19

1 OpenText eDirectory について 21

OpenText Identity Console による容易な管理 . . . 22
- 強力なツリー構造 . . . 22
- Web ベースの管理ユーティリティ . . . 24
- シングルログインと認証 . . . 25

オブジェクトクラスとプロパティ . . . 26
- オブジェクトのリスト . . . 26
- コンテナオブジェクトクラス . . . 28
- リーフオブジェクトクラス . . . 33

コンテキストと命名規則 . . . 47
- 識別名 . . . 48
- タイプ付きの名前 . . . 48
- ネームレゾリューション . . . 49
- 現在のワークステーションのコンテキスト . . . 49
- 先頭ピリオド . . . 49
- 相対命名 . . . 49
- 後続ピリオド . . . 50
- Linux でのコンテキストと名前付け . . . 50

スキーマ . . . 51
- スキーマ管理 . . . 51
- スキーマクラス、属性、および構文 . . . 51
- 必須属性およびオプション属性について . . . 58
- スキーマのサンプル . . . 59
- スキーマを設計する . . . 59

パーティション . . . 59
- パーティション . . . 60
- パフォーマンス向上のためにレプリカを分散する . . . 60
- パーティションと WAN リンク . . . 61

レプリカ . . . 62
- レプリカのタイプ . . . 63
- フィルタ済みレプリカ . . . 66

レプリカリングでのサーバの同期 . . . 68

リソースへのアクセス . . . 69

OpenText eDirectory での権利 . . . 69
- トラスティ割り当ておよびターゲット . . . 70
- OpenText eDirectory の権利の概念 . . . 70
- 新規サーバのデフォルト権 . . . 76
- 管理の委託 . . . 77
- 権利の管理 . . . 78

2 OpenText eDirectory ネットワークの設計 85

OpenText eDirectory の設計の基本 . . . 85
- ネットワークのレイアウト . . . 85
- 組織の構造 . . . 86

OpenText eDirectory 設計の準備をする . . . 86
OpenText eDirectory ツリーの設計 . . . 86
命名標準ドキュメントを作成する . . . 86
Tree の上位層を設計する . . . 89
ツリーの下位層を設計する . . . 92
ツリーのパーティション化のガイドライン . . . 93
ツリーの上位層におけるパーティションを決定する . . . 94
ツリーの下位層におけるパーティションを決定する . . . 94
パーティションサイズを決定する . . . 94
ネットワーク変数について . . . 95
ツリーのレプリカ作成に関するガイドライン . . . 95
ワークグループのニーズ . . . 96
障害対策 . . . 96
レプリカ数を決定する . . . 97
Tree パーティションのレプリカを作成する . . . 98
管理用にレプリカを作成する . . . 98
WAN トラフィックを管理する . . . 98
ユーザ環境についてのプランニング . . . 98
ユーザのニーズを確認する . . . 99
アクセスに関するガイドラインを作成する . . . 99
e ビジネスに対応する OpenText eDirectory の設計 . . . 99
OpenText Certificate Server について . . . 100
OpenText Certificate Server でタスクを実行するのに必要な権利 . . . 101
Linux コンピュータ上の OpenText eDirectory 操作をセキュリティで保護する . . . 102
ネットワーク時刻の同期 . . . 105
Linux コンピュータで時刻を同期する . . . 106
時刻同期を確認する . . . 106

3 オブジェクトの管理 107

オブジェクトに関連する一般的なタスク . . . 107
OpenText eDirectory ツリーをブラウズする . . . 108
オブジェクトを作成する . . . 110
オブジェクトのプロパティの変更 . . . 110
オブジェクトをコピーする . . . 111
オブジェクトを移動する . . . 111
オブジェクトの削除 . . . 111
オブジェクトをリネームする . . . 112
ユーザアカウントの管理 . . . 112
ユーザアカウントを作成および変更する . . . 113
オプションのアカウント機能を設定する . . . 114
ログイン時間更新間隔の無効化 . . . 116
ログインスクリプトを設定する . . . 117
リモートユーザのログイン時間制限 . . . 118
ユーザアカウントを削除する . . . 119
役割とアクセス制御の設定 . . . 119
RAC 役割の定義 . . . 122
カスタムの RAC タスクの定義 . . . 124

4 バックグラウンドプロセスの管理 127

同期 . . . 127
同期の特徴 . . . 128
通常同期またはレプリカ同期 . . . 130

優先度同期 . . . 132
ポリシーベースのレプリケーション . . . 139
同期スレッドの手動設定 . . . 141
非同期アウトバウンド同期の設定 . . . 142
バックグラウンドプロセスの設定 . . . 142
ハード制限ポリシー . . . 142
CPU ベースの動的ポリシー . . . 143
バックグラウンドプロセスの間隔 . . . 143

5 スキーマの管理 145

スキーマの拡張 . . . 146
クラスを作成する . . . 146
クラスを削除する . . . 147
属性を作成する . . . 147
クラスへオプション属性を追加する . . . 147
属性を削除する . . . 148
補助クラスを作成する . . . 149
補助クラスのプロパティでオブジェクトを拡張する . . . 149
オブジェクトの補助プロパティを変更する . . . 150
オブジェクトから補助プロパティを削除する . . . 150
スキーマの表示 . . . 150
クラス情報を参照する . . . 151
属性情報を表示する . . . 151
手動でスキーマを拡張する . . . 151
Windows でスキーマを拡張する . . . 151
Linux でスキーマを拡張する . . . 152
バージョン 8.7 以降に追加されたスキーマフラグ . . . 154
クライアントを使用してスキーマ操作を実行する . . . 155
DSSchema eMTool を使用する . . . 155
DSSchema eMTool オプション . . . 156

6 パーティションおよびレプリカの管理 159

パーティションの作成 . . . 160
パーティションのマージ . . . 161
パーティションの移動 . . . 162
パーティションの作成操作またはマージ操作のキャンセル . . . 164
レプリカの管理 . . . 164
レプリカを追加する . . . 164
レプリカを削除する . . . 165
レプリカタイプを変更する . . . 166
フィルタ済みレプリカを設定し管理する . . . 168
フィルタ処理済レプリカウィザードを使用する . . . 168
パーティションスコープを定義する . . . 169
サーバフィルタを設定する . . . 170
パーティションおよびレプリカを表示する . . . 171
サーバのパーティションを表示する . . . 171
パーティションレプリカを表示する . . . 171
パーティションに関する情報を表示する . . . 172
パーティションの階層を表示する . . . 172
レプリカに関する情報を表示する . . . 172

7 OpenText eDirectory 管理ユーティリティ 175

OpenText インポート / エクスポート変換ユーティリティ . . . 175
コマンドラインインタフェースを使用する . . . 176
変換ルール . . . 197
LBURP (LDAP Bulk Update/Replication Protocol) . . . 207
LDIF のインポートを高速化する . . . 208
インデックスマネージャ . . . 209
インデックスを作成する . . . 210
インデックスを削除する . . . 211
インデックスをオフラインにする . . . 211
他のサーバ上でインデックスを管理する . . . 212
OpenText eDirectory Service Manager . . . 212
クライアントのサービスマネージャ eMTool を使用する . . . 212
オフラインのバルクロードユーティリティ . . . 213
eDirectory のパフォーマンスの改善 . . . 214
バルクロードに ldif2dib を使用する . . . 219
複数のインスタンス . . . 219
ldif2dib を調整する . . . 220
制限 . . . 221
注意事項 . . . 223
LDIF ファイル . . . 224
LDIF について . . . 224
LDIF ファイルのデバッグ . . . 233
LDIF を使用してスキーマを拡張する . . . 236
ldif2dib の制限 . . . 240

8 OpenText の監視 243

OpenText iMonitor の使用 . . . 243
システム要件 . . . 245
OpenText iMonitor へのアクセス . . . 245
OpenText iMonitor のアーキテクチャ . . . 246
OpenText iMonitor の機能 . . . 252
OpenText iMonitor の操作のセキュリティ保護 . . . 277
HTTP サーバオブジェクトの設定 . . . 278
ndsconfig を使用する HTTP スタックパラメータの設定 . . . 279
cn=monitor を使用した監視 . . . 280
監視統計情報の表示 . . . 281
DSTrace の使用 . . . 292
基本機能 . . . 293
デバッギングメッセージ . . . 294
バックグラウンド処理 . . . 296
DSTrace メッセージ . . . 301
Linux . . . 301
Windows . . . 303
OpenText iMonitor のメッセージフィルタリング . . . 305
SAL メッセージのフィルタ . . . 305
重大度レベルの設定 . . . 305
ログファイルパスを設定する . . . 306

9 OpenText eDirectory サーバの SecretStore 環境設定 307

Linux . . . 307

Windows 308

10 OpenText eDirectory ツリーのマージ 309

OpenText eDirectory ツリーの統合 310
前提条件 310
ターゲットツリーの要件 310
スキーマの要件 311
ソースツリーをターゲットツリーへマージする 311
パーティションの変化 311
ソースツリーとターゲットツリーを準備する 312
マージする前の時刻の同期 313
2 つのツリーのマージ 314
マージ後の作業 315
サーバツリーの結合 316
コンテキスト名の変更について 318
ソースツリーとターゲットツリーを準備する 319
結合操作における包含要件 321
ソースツリーとターゲットツリーを結合する 321
ツリー名の変更 321
クライアントを使用したツリーのマージ 322
DSMerge eMTool を使用する 322
DSMerge eMTool オプション 323

11 OpenText eDirectory のデータを暗号化する 325

暗号化属性 325
暗号化方式を使用する 327
暗号化属性ポリシーを管理する 328
暗号化属性にアクセスする 332
暗号化属性を表示する 334
バックアップデータを暗号化 / 復号化する 335
暗号化属性を含む DIB ファイルセットのクローンを作成する 335
レプリカリングに OpenText eDirectory サーバを追加する 335
後方互換性 335
暗号化属性に移行する 335
暗号化属性のレプリカを作成する 336
暗号化レプリケーション 336
暗号化複製の必要性 337
暗号化複製を有効にする 337
新しいレプリカをレプリカリングに追加する 341
同期と暗号化複製 342
暗号化複製ステータスを表示する 343
データを暗号化するときにデータの完全な安全性を確保する 343
完全に新しい設定でデータを暗号化する 344
既存の設定でデータを暗号化する 345
結論 346

12 OpenText eDirectory データベースの修復 347

基本修復操作の実行 348
標準修復を実行する 349
ローカルデータベースの修復の実行 350
外部参照のチェック 351

単一オブジェクトの修復 . . . 351
不明なリーフオブジェクトの削除 . . . 351
修復ログファイルの表示と設定 . . . 352
ログファイルを開く . . . 352
ログファイルオプションを設定する . . . 352
OpenText iMonitor での修復の実行 . . . 352
レプリカの修復 . . . 353
すべてのレプリカを修復する . . . 353
選択したレプリカを修復する . . . 353
タイムスタンプを修復する . . . 353
このサーバを新しいマスタレプリカに設定する . . . 355
選択したレプリカを削除する . . . 355
レプリカリングを修復する . . . 355
すべてのレプリカリングを修復する . . . 356
選択したレプリカリングを修復する . . . 356
リング内のすべてのサーバにすべてのオブジェクトを送信する . . . 356
マスタから選択したレプリカへすべてのオブジェクトを受信する . . . 357
レプリカリングからこのサーバを削除する . . . 357
スキーマの保守 . . . 358
ツリーからスキーマを要求する . . . 358
ローカルスキーマをリセットする . . . 359
オプションスキーマ拡張機能を実行する . . . 359
リモートスキーマをインポートする . . . 359
新規スキーマエポックを宣言する . . . 360
サーバのネットワークアドレスの修復 . . . 360
すべてのネットワークアドレスを修復する . . . 361
サーバのネットワークアドレスの修復 . . . 361
同期化操作の実行 . . . 362
選択したレプリカをこのサーバで同期する . . . 362
このサーバの同期ステータスをレポートする . . . 362
すべてのサーバの同期ステータスをレポートする . . . 363
時刻同期を実行する . . . 363
即時同期をスケジュールする . . . 364
DSRepair オプション . . . 364
OpenText eDirectory サーバ上で DSRepair を実行する . . . 364
DSRepair コマンドラインオプション . . . 366
DSRepair 詳細設定スイッチの使用 . . . 368
クライアントを使用したデータベースの修復 . . . 369
DSRepair eMTool を使用する . . . 369
DSRepair eMTool のオプション . . . 370
グラフィカルな DS 修復ユーティリティ . . . 372

13 LDAP Services for OpenText eDirectory について 373

LDAP サービスの主な用語 . . . 374
クライアントとサーバ . . . 374
オブジェクト . . . 374
参照 . . . 375
LDAP と OpenText eDirectory の連携について . . . 377
LDAP から OpenText eDirectory に接続する . . . 377
クラスと属性のマッピング . . . 381
構文の相違 . . . 384
サポートされる OpenText LDAP コントロールと拡張 . . . 386
Linux での LDAP ツールの使用 . . . 386

LDAP ツール . . . 387
拡張可能一致検索フィルタ . . . 399
LDAP トランザクション . . . 401
制限 . . . 402

14 LDAP Services for OpenText eDirectory の環境設定 403

LDAP Services for OpenText eDirectory のロードおよびアンロード . . . 403
LDAP サーバがロードされているか確認する . . . 404
LDAP サーバが実行されているか確認する . . . 405
シナリオ . . . 405
LDAP サーバが実行されているか確認する . . . 406
デバイスが受信待機していることを確認する . . . 406
SSLv3 無効化による POODLE 攻撃の防止 . . . 407
LDAP オブジェクトを環境設定する . . . 407
Linux 上で LDAP サーバオブジェクトおよび LDAP グループオブジェクトを環境設定する . 409
ldapSSLConfig 属性を使用してプロトコルと暗号を構成する . . . 419
LDAP サーバをリフレッシュする . . . 422
認証とセキュリティ . . . 423
パスワードとの単純バインドに TLS を要求する . . . 424
TLS を開始 / 停止する . . . 424
TLS のサーバを環境設定する . . . 425
TLS のクライアントを環境設定する . . . 427
ルート認証局をエクスポートする . . . 427
クライアント証明書で認証を受ける . . . 428
サードパーティプロバイダの証明書を使用する . . . 428
LDAP プロキシユーザを作成および使用する . . . 429
SASL を使用する . . . 430
NMAS ベースのログインを使用した LDAP 認証 . . . 432
LDAP サーバを使ってディレクトリを検索する . . . 433
検索制限を設定する . . . 433
参照を使用する . . . 434
上方参照を設定する . . . 442
シナリオ : 連結ツリーでの上方参照 . . . 442
信頼されていない領域を作成する . . . 443
参照データを指定する . . . 445
LDAP で参照情報を更新する . . . 446
影響を受ける操作 . . . 446
上方参照のサポートの有無を確認する . . . 447
持続的検索 : OpenText eDirectory イベントの設定 . . . 447
持続的検索の管理 . . . 448
イベントの監視拡張操作の使用を制御する . . . 449
LDAP サーバの情報を取得する . . . 449
汎用タイムサポートの設定 . . . 451
許容変更の設定 . . . 452
プロキシ承認コントロール . . . 452
LDAP ページ検索コントロール . . . 453
LDAP 拡張 DN コントロール . . . 453
LDAP イベントの監査 . . . 457

15 OpenText eDirectory のバックアップと復元 459

OpenText eDirectory のバックアップ処理に関する確認事項 . . . 461
バックアップサービスおよび復元サービスについて . . . 463
OpenText eDirectory バックアップツールについて . . . 464
DSBK のバックアップおよび復元と TSA for NDS バックアップの違い . . . 465
バックアップツールによる復元作業の概要 . . . 466
バックアップファイルのヘッダ書式 . . . 468
バックアップログファイルの書式 . . . 472
DSMASTER サーバによる災害対策 . . . 473
遷移ベクトルと復元後の検証処理 . . . 474
ロールフォワードログを使用する . . . 475
ロールフォワードログ機能を使用する上での注意事項 . . . 476
ロールフォワードログの保存先 . . . 477
ロールフォワードログのバックアップと削除 . . . 479
注意 : OpenText eDirectory を削除するとロールフォワードログも削除される問題 . . . 480
復元処理の準備 . . . 480
復元作業の前提条件 . . . 480
復元に必要なバックアップファイルの収集 . . . 481
DSBK の使用 . . . 483
前提条件 . . . 483
さまざまなプラットフォームで DSBK を使用する . . . 484
DSBK による手動バックアップ . . . 487
OpenText eDirectory のバックアップの自動化 . . . 488
DSBK によるロールフォワードログの設定 . . . 488
DSBK によるバックアップファイルの復元作業 . . . 489
バックアップ / 復元のコマンドラインオプション . . . 491
cron ジョブとしての DSBK の実行 . . . 501
NICI のバックアップと復元 . . . 501
NICI のバックアップ . . . 502
NICI の復元 . . . 502
復元後の検証処理に失敗した場合の対処方法 . . . 503
レプリカリングをクリーンアップする . . . 504
サーバの復旧とレプリカの再追加 . . . 506
バックアップ / 復元の運用例 . . . 507
シナリオ : 単一サーバ構成のネットワークで、OpenText eDirectory を格納しているハードディスクが故障した場合 . . . 508
シナリオ : 複数サーバ構成のネットワークで、OpenText eDirectory を格納しているハードディスクが故障した場合 . . . 509
シナリオ : 複数サーバ構成のネットワークで、1 台のサーバが完全に使えなくなった場合 . . . 511
シナリオ : 複数サーバ構成のネットワークで、数台のサーバが使えなくなった場合 . . . 512
シナリオ : 複数サーバ構成のネットワークで、すべてのサーバが使えなくなった場合 . . . 512
DSBK を使用した障害復旧計画 . . . 514
Linux 上での障害復旧計画 . . . 514
Windows 上での障害復旧計画 . . . 516
LDAP ベースのバックアップ . . . 517
LDAP ベースのバックアップの必要性 . . . 518
その他の情報 . . . 518
SMS による OpenText eDirectory のバックアップ . . . 518

16 Suite B モードでの OpenText eDirectory の設定 519

新規インストールでの Suite B の有効化 . . . 520
Certificate Server 上での Suite B の有効化 . . . 521
ECDSA 証明書と Suite B
Cipher を使用するための LDAP サービスと HTTP サービスの設定 . . . 522
AES 256 ビット SDI 鍵の作成 . . . 525
バックグラウンド認証の有効化 . . . 526
既存のサーバ上での Suite B の設定 . . . 526

17 Enhanced Background Authentication の有効化 529

EBA の有効化 . . . 531
OpenText eDirectory ツリー上での EBA の有効化 . . . 531
OpenText eDirectory サーバ上での EBA の有効化 . . . 532
OpenText eDirectory サーバ上での EBA の無効化 . . . 533
EBA に関する情報の表示 . . . 533
OpenText Identity Console を使用した EBA CA の管理 . . . 535
EBA が有効な場合の OpenText eDirectory 操作の制限 . . . 536
レプリカタイプの変更に関する制限 . . . 536
パーティションのマスタの変更に関する制限 . . . 536
パーティションのマージに関する制限 . . . 536
EBA が有効になっているサーバの再設定に関する制限 . . . 537
EBA 対応サーバのバックアップ . . . 537
新しいサーバへの EBA CA の役割の移動 . . . 537

18 OpenText eDirectory の SNMP サポート 539

SNMP に関する用語の定義 . . . 539
SNMP サービスについて . . . 540
OpenText eDirectory と SNMP . . . 542
OpenText eDirectory の管理に SNMP を使う利点 . . . 542
SNMP と OpenText eDirectory の連携について . . . 543
OpenText eDirectory の SNMP サービスのインストールと設定 . . . 545
SNMP サーバモジュールのロードとアンロード . . . 545
サブエージェントの設定 . . . 546
OpenText eDirectory の SNMP サービスの設定 . . . 549
SNMP による OpenText eDirectory の監視 . . . 552
トラップ . . . 552
トラップに関する設定 . . . 569
[Statistics (統計情報)] . . . 577
トラブルシューティング . . . 582

19 OpenText eDirectory のメンテナンス 583

詳細参照コスト . . . 583
サーバ間接続の向上 . . . 584
参照コストのメリット . . . 586
ARC の展開 . . . 587
詳細参照コストの有効化 . . . 588
詳細参照コストの調整 . . . 589
詳細参照コストの監視 . . . 590
OpenText eDirectory の正常動作の維持 . . . 593

ヘルスチェックを実行する時期 . . . 593
ヘルスチェックの概要 . . . 593
OpenText iMonitor を使用した OpenText eDirectory のヘルスチェック . . . 594
その他の情報 . . . 595
監視のためのリソース . . . 596
ハードウェアのアップグレードやサーバの交換 . . . 596
サーバを交換しないでハードウェアまたはストレージデバイスを計画的にアップグレードする . . . 596
サーバの計画的な交換 . . . 600
サーバの IP アドレスの変更 . . . 603
ハードウェア障害後の OpenText eDirectory の復元 . . . 604
サブツリー検索のパフォーマンスの向上 . . . 604
コンテナの準備状況 . . . 605

20 DHost OpenText iConsole Manager 607

DHost について . . . 608
DHost OpenText iConsole の実行 . . . 609
Windows で DHost OpenText iConsole を実行する . . . 609
Linux で DHost OpenText iConsole を実行する . . . 609
OpenText eDirectory モジュールの管理 . . . 610
Windows でモジュールをロードまたはアンロードする . . . 610
Linux でモジュールをロードまたはアンロードする . . . 611
DHost 情報の照会 . . . 611
環境設定パラメータを表示する . . . 611
プロトコル情報を表示する . . . 612
接続プロパティを表示する . . . 612
スレッドプールの統計情報を表示する . . . 612
プロセススタック . . . 613

21 sadmin パスワードの設定 615

22 OpenText eDirectory 管理ツールボックス 617

コマンドラインクライアントの使用 . . . 618
コマンドラインヘルプを表示する . . . 619
コマンドラインクライアントを対話式モードで実行する . . . 619
コマンドラインクライアントをバッチモードで実行する . . . 624
eMBox コマンドラインクライアントのオプション . . . 626
クライアントを使用してセキュア接続を確立する . . . 627
OpenText eDirectory のポート番号を確認する . . . 627
ログの記録の使用 . . . 628
ログの記録コマンドラインクライアントを使用する . . . 629
eMBox クライアントを使ったバックアップ / 復元作業 . . . 629
前提条件 . . . 630
eMBox クライアントによる手動バックアップ . . . 631
バッチファイルと eMBox クライアントによる無人バックアップ . . . 633
eMBox クライアントによるロールフォワードログの設定 . . . 634
eMBox クライアントによるバックアップファイルの復元作業 . . . 636

23 OpenText eDirectory イベントの監査 639

CEF による監査 . . . 639

CEF の設定 . . . 639
ジャーナルイベントのキャッシング . . . 662
LDAP 監査 . . . 664
LDAP 監査の必要性 . . . 664
LDAP 監査の利用 . . . 664
その他の情報 . . . 665

24 OpenText eDirectory の認証フレームワークについて 667

NMAS の機能 . . . 667
ユーザ識別フェーズ . . . 667
認証 (ログイン) フェーズ . . . 668
デバイス取り外し検出フェーズ . . . 670
ログインメソッドとポストログインメソッドとシーケンス . . . 670
セキュリティオブジェクトのキャッシュ . . . 671
NMAS ソフトウェア . . . 671
サーバソフトウェアとクライアントソフトウェアのインストール . . . 672
ログインメソッドソフトウェアとパートナー . . . 672
ユニバーサルパスワード . . . 673
OpenText Identity Console での管理 . . . 673
ログインとポストログインのメソッドとシーケンスの管理 . . . 674
ログインメソッドのインストール方法 . . . 674
ログインメソッドとポストログインメソッドの更新 . . . 676
ログインシーケンスの管理 . . . 676
ユーザに対するログインシーケンスの承認 . . . 679
デフォルトログインシーケンスの設定 . . . 680
ログインメソッドの削除 . . . 680
ログインシーケンスの削除 . . . 681
NMAS を使用したネットワークへのログイン . . . 682
パスワードフィールド . . . 682
カスタムログイン . . . 682
ワークステーションのロック解除 . . . 683
NMAS クライアントトレースの収集 . . . 683
NMAS クリアランスステータスの表示 . . . 683
OpenText パスワードの履歴 . . . 684
NMAS HOTP ベースのログイン . . . 685
概要 . . . 685
インストール . . . 686
カウンタの再同期 . . . 688
環境設定 . . . 689
当バージョンの注意事項 . . . 690
nmashotpconf ユーティリティは、ユーザ再同期ウィンドウを変更できません。 . . . 691
その他の管理タスク . . . 691
ポリシーリフレッシュレートコマンドの使用 . . . 691
LoginInfo コマンドの使用 . . . 692
LDAP に対する NMAS ベースのログインの無効化 . . . 695
NMAS コマンドの呼び出し . . . 695
失敗したログイン試行の遅延時間の設定 . . . 696
DSTrace の使用 . . . 696
NMAS クライアントの無効化とアンインストール . . . 696
OpenText Audit と外部証明書の使用 . . . 697
セキュリティ上の考慮事項 . . . 697
パートナーログインメソッド . . . 698
ログインポリシー . . . 698

NMASInst 699
ユニバーサルパスワード 699
SDI キー 701

25 Certificate Server について **703**

OpenText Certificate Server の機能 703
OpenText Certificate Server のコンポーネント 704
OpenText Certificate Server 704
NICI 711
OpenText Certificate Server の設定 711
使用する認証局のタイプの決定 712
組織の認証局オブジェクトを作成する 713
従属認証局 715
認証局オブジェクトの作成に関する制限事項 718
Suite B モードでの認証局の設定 718
サーバ証明書オブジェクトを作成する 719
暗号化対応アプリケーションの設定 720
追加コンポーネントの設定 720
OpenText Certificate Server の管理 722
認証局のタスク 724
サーバ証明書オブジェクトのタスク 733
ユーザ証明書のタスク 744
X.509 証明書の自己プロビジョニング 749
外部アプリケーションでの OpenText eDirectory 証明書の使用 753
ルート認証局オブジェクトのタスク 756
証明書取り消しリスト (CRL) のタスク 758
OpenText eDirectory のタスク 767
アプリケーションのタスク 768
PKI ヘルスチェック 769
公開鍵暗号化の基本 773
概要 773
セキュアなデータ転送 773
鍵ペア 773
信頼関係の確立 776
タスクを実行するために必要なエントリ権 779

26 パスワードを管理する **785**

ユニバーサルパスワードについて 785
ユニバーサルパスワードの安全性 785
ユニバーサルパスワード 787
パスワードポリシー 788
パスワード同期 788
非可逆パスワードストレージを理解する 788
非可逆パスワードストレージの有効化 789
パスワードポリシー 790
ユニバーサルパスワードの導入 790
ステップ 1: ユニバーサルパスワードの必要性を特定する 790
ステップ 2: セキュリティコンテナが使用可能であることを確認する 791
ステップ 3:
SDI ドメインキーサーバがユニバーサルパスワードに対応可能であることを検証する 791
ステップ 4: SDI 鍵の一貫性を確認する 793
ステップ 5: ユニバーサルパスワードを有効にする 793

後方互換性 . . . 794
パスワードの管理 . . . 794
注意を要する問題 . . . 795
パスワードポリシーを使用したパスワードの管理 . . . 796
パスワードポリシー機能の概要 . . . 796
パスワードポリシーの計画 . . . 797
パスワードポリシーを使用するために必要な事前タスク . . . 801
パスワードポリシーの作成 . . . 803
ユーザへのパスワードポリシーの割り当て . . . 820
ユーザに適用されているポリシーの識別 . . . 822
ユーザのパスワードの設定 . . . 822
ユニバーサルパスワードの診断ユーティリティ . . . 823
パスワードポリシーのトラブルシューティング . . . 825
セキュリティ上の考慮事項 . . . 826
eDirectory へのハッシュベースのパスワードのインポート . . . 828

27 REST サービス 829

セキュリティの推奨事項 . . . 830
OpenText eDirectory 向けの REST サービスのインストールを計画する . . . 830
OpenText eDirectory 用の REST サービスの設定 . . . 834
データ永続性の管理 . . . 836
REST サービスによる監査 . . . 837
REST イベントを理解する . . . 837
REST コンテナを使用して LDAP パスワードを変更する . . . 838
REST コンテナを使用してサーバ証明書を変更する . . . 838
OpenText eDirectory 用の REST サービスのアップグレード . . . 839
Rest API マニュアル . . . 841

A NMAS の注意事項 843

独立したパーティションとしてのセキュリティコンテナの設定 . . . 843
複数のセキュリティコンテナを持つツリーのマージ . . . 843
ツリーのマージ前に実行する製品固有の操作 . . . 844
ツリーのマージを実行する . . . 847
ツリーのマージ後に実行する製品固有の操作 . . . 847

B OpenText eDirectory 用の Linux コマンドとそれらの使用法 849

一般ユーティリティ . . . 849
LDAP 固有のコマンド . . . 854

C OpenSLP for OpenText eDirectory の設定 859

Service Location Protocol . . . 859
SLP の基本 . . . 859
OpenText Service Location Providers . . . 860
ユーザエージェント . . . 861
サービスエージェント . . . 861
環境設定パラメータ . . . 862

D OpenText eDirectory での DNS の使用方法 865

E OpenText eDirectory での GSSAPI の設定 867

概念 . . . 867
Kerberos について . . . 868
SASL について . . . 868
GSSAPI について . . . 868
OpenText eDirectory での GSSAPI の動作 . . . 868
GSSAPI を設定するための前提条件 . . . 869
ネットワークの特性に関する前提 . . . 870
Kerberos の LDAP 拡張の追加 . . . 871
ルート認証局証明書のエクスポート . . . 872
SASL-GSSAPI メソッドを使用して設定された OpenText eDirectory ツリーをマージする . . . 872
SASL-GSSAPI メソッドの管理 . . . 872
サービスプリンシパルの管理 . . . 873
ログインシーケンスの作成 . . . 874
LDAP での SASL-GSSAPI の使用方法 . . . 874
エラーメッセージ . . . 874
よく使用される用語 . . . 874

F セキュリティ上の考慮事項 877

LDAP バインド . . . 877
Nessus のスキャン結果 . . . 878

G OpenText eDirectory イベントと CEF イベントのマッピング 881

OpenText eDirectory イベントと CEF イベントのマッピング . . . 881
CEF イベント . . . 886
セキュリティイベント . . . 886
オブジェクトイベント . . . 893
属性イベント . . . 895
EBA イベント . . . 897

H トラブルシューティング 899

SNMP のトラブルシューティング . . . 899
OpenText iMonitor のトラブルシューティング . . . 904
破損通知のトラブルシューティング . . . 905
OpenText eDirectory への移行 . . . 909
スキーマのトラブルシューティング . . . 916
DSRepair のトラブルシューティング . . . 917
レプリケーションのトラブルシューティング . . . 917
クローン DIB に関する問題のトラブルシューティング . . . 918
OpenText 公開鍵インフラストラクチャサービスのトラブルシューティング . . . 918
Linux でのユーティリティのトラブルシューティング . . . 925
NMAS のトラブルシューティング . . . 926
ディレクトリサービスがロードされない場合の HTTPSTK へのアクセス . . . 929
データ暗号化のトラブルシューティング . . . 931
OpenText eDirectory 管理ツールボックス . . . 934

SASL-GSSAPI に関する問題のトラブルシューティング 936
OpenText eDirectory のエラーログを管理する 938
その他 941
IPV6 に関する問題のトラブルシューティング 950
EBA のトラブルシューティング. 950

本書およびライブラリについて

この *管理ガイド* では、OpenText™ eDirectory CE 24.4 (v9.3) 製品を管理および設定する方法について説明します。

本書の読者

このマニュアルはネットワーク管理者を対象としています。

ライブラリに含まれているその他の情報

ライブラリには次の情報リソースが含まれています。

インストールガイド

OpenText eDirectory をインストールする方法について説明します。ネットワーク管理者を対象としています。

Linux プラットフォーム用チューニングガイド

Linux プラットフォーム上の OpenText eDirectory を分析し、すべての展開で優れたパフォーマンスが実現されるように調整する方法について説明します。

これらのガイドは、OpenText eDirectory 9.3 マニュアルの Web サイト (https://www.netiq.com/documentation/edirectory-93/) で入手できます。

OpenText eDirectory 管理ユーティリティの詳細については、『OpenText™ Identity Console 管理ガイド』を参照してください。

1 OpenText eDirectory について

簡単に言うと、OpenText eDirectory とは、ネットワークユーザ、サーバ、プリンタ、プリントキュー、アプリケーションなどのネットワークリソースを表す一連のオブジェクトです。OpenText eDirectory は、スケーラブルで高性能、安全性の高いディレクトリサービスです。ユーザ、アプリケーション、ネットワークデバイス、およびデータなど、多量のオブジェクトを格納および管理できます。OpenText eDirectory が提供するセキュア識別情報管理ソリューションは、複数のプラットフォーム間で実行され、インターネットスケーラブルであり、拡張可能です。

また、OpenText eDirectory は、中央型識別情報管理、インフラストラクチャ、社内ネットワーク全体をカバーするセキュリティ、およびスケーラビリティを、ファイアウォールの内外で実行するすべてのアプリケーションに提供します。OpenText eDirectory には、Web ベースのワイヤレス管理機能が含まれており、Web ブラウザや各種ハンドヘルドデバイスから、ディレクトリやユーザ、アクセス権、およびネットワークリソースにアクセスしたり、それらを管理したりすることができます。

OpenText eDirectory では、ディレクトリ標準 LDAP (Lightweight Directory Access Protocol) バージョン 3 がネイティブでサポートされ、OpenSSL ソースコードに基づいた TLS/SSL サービスへのサポートが提供されています。

OpenText eDirectory エンジンの詳細については、「OpenText eDirectory Process Requests (http://support.novell.com/techcenter/articles/anp20020801.html)」を参照してください。

「図 1-1」は、「OpenText™ Identity Console 管理ガイド」管理ユーティリティに表示されるオブジェクトの一部を示しています。

図 1-1 *OpenText Identity Console の OpenText eDirectory オブジェクト*



OpenText eDirectory サーバに設定された実際のスキーマや OpenText eDirectory を実行するオペレーティングシステムによっては、使用できないオブジェクトクラスもあります。

オブジェクトの詳細については、26 ページの 「オブジェクトクラスとプロパティ」を参照してください。

ネットワークに複数の OpenText eDirectory サーバがある場合、ディレクトリは複数のサーバに複製できます。

このセクションでは、次の情報について説明します。

- 22 ページの 「OpenText Identity Console による容易な管理」
- 26 ページの 「オブジェクトクラスとプロパティ」
- 47 ページの 「コンテキストと命名規則」
- 51 ページの 「スキーマ」
- 59 ページの 「パーティション」
- 62 ページの 「レプリカ」
- 68 ページの 「レプリカリングでのサーバの同期」
- 69 ページの 「リソースへのアクセス」
- 69 ページの 「OpenText eDirectory での権利」

OpenText Identity Console による容易な管理

OpenText eDirectory では、ネットワークリソースを容易に、強力に、そしてフレキシブルに管理することができます。eDirectory は、グループウェアやその他アプリケーションのユーザ情報のリポジトリとしても機能します。これらのアプリケーションは、業界標準の LDAP(Lightweight Directory Access Protocol) を使用して、ディレクトリにアクセスします。

OpenText eDirectory の便利な管理機能には、強力なツリー構造、統合管理ユーティリティ、およびシングルログインと認証機能があります。

OpenText Identity Console を使用すると、Web ブラウザやさまざまなハンドヘルドデバイスから、ディレクトリやユーザ、ディレクトリ内のアクセス権やネットワークリソースを管理できます。OpenText Identity Console を使用すると、基本的なディレクトリ管理タスクにアクセスしたり、DSRepair、DSMerge、バックアップおよび復元など、以前は OpenText eDirectory サーバで実行する必要のあった OpenText eDirectory 管理ユーティリティにアクセスしたりできます。

詳細については、『*OpenText™ Identity Console 管理ガイド* (https://www.netiq.com/documentation/identity-console/identity_console-admin/data/bookinfo.html)』を参照してください。

強力なツリー構造

OpenText eDirectory ではオブジェクトがツリー構造に編成されます。最上位の Tree オブジェクトにはツリー名が付けられます。

OpenText eDirectory サーバで動作するオペレーティングシステムが Linux、Windows のどちらであっても、すべてのリソースを同じツリー内に保管できます。オブジェクトの作成や、権利の許可、パスワードの変更、アプリケーションの管理のために、それぞれのサーバやドメインに個別にアクセスする必要はありません。

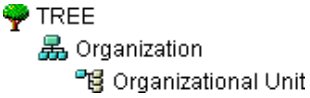
ツリーの階層構造によって、極めて柔軟で強力な管理が可能になっています。このような管理機能は、主に次の 2 つの機能で実現できます。

- 23 ページの 「コンテナオブジェクト」
- 24 ページの 「継承」

コンテナオブジェクト

コンテナオブジェクトを利用することによって、オブジェクトを個々に扱うのではなく、オブジェクトのまとまりとして扱うことができます。に示すように、コンテナオブジェクトには、3 つの共通クラスがあります。図 1-2

***図 1-2** コンテナオブジェクトの共通クラス*



ツリーオブジェクトは、ツリー内の最上位のコンテナオブジェクトです。通常、このコンテナには、その会社の組織オブジェクトが格納されます。

組織は通常、ツリーオブジェクトの直下に位置するコンテナクラスです。一般に、組織オブジェクトには会社名に基づく名前が付けられます。小規模な会社の場合は、管理を容易化するために、他のオブジェクトをすべて組織オブジェクトの直下に配置します。

組織の下に部門オブジェクトを作成し、異なった地区、ネットワークキャンパス、または個々の部署を表すことができます。部門の下にさらに部門を作成して、ツリーを細分化することもできます。

その他、コンテナオブジェクトには、カントリクラスと地域クラスがあります。これらのクラスは通常、複数の国にまたがるネットワークでのみ使用されます。

ドメインオブジェクトは、ツリーオブジェクトの下、または組織、部門、国、地域オブジェクトの下に作成できます。

コンテナ内のすべてのオブジェクトに対してなんらかの処理を行う場合、コンテナオブジェクトに処理を行えば 1 回の実行で済みます。Accounting コンテナ (Database アプリケーション、Bookkeepers グループ、LaserPrinter プリンタ、および Amy、Bill、Bob というユーザが含まれている) 内の全オブジェクトに対する完全な管理権限をユーザ Amy に与える必要があるとします。

これを行うには、OpenText Identity Console のホームページのツリービューに移動し、左側のペインで **Accounting** オブジェクトの親ツリーを選択します。右側のペインで［**Accounting**］を選択して、［**アクション**］>［**トラスティの変更**］をクリックします。［**トラスティの追加**］をクリックして、Amy をトラスティとして追加します。次に、［**割り当てられた権利**］をクリックして、Amy に与える権利を選択します。これで Amy には、Database アプリケーション、Bookkeepers グループ、LaserPrinter プリンタ、および自分自身とユーザ Bill、Bob を管理する権利が与えられました。

継承

OpenText eDirectory のもう 1 つの強力な機能は、権利の継承です。継承とは、ツリー内の上位層のコンテナの権利が、それぞれの下位層のすべてのコンテナに受け継がれることを意味します。この機能によって、権利の割り当て回数を少なくすることができます。たとえば、24 ページの 図 1-3 に示すオブジェクトに対する管理権を与えるとします。

図 1-3 *OpenText eDirectory オブジェクトのサンプル*



次の割り当てのいずれかを行うことができます。

- ユーザに Allentown に対する権利を与えると、そのユーザは Allentown コンテナ内のオブジェクトのみ管理できます。
- ユーザに East に対する権利を与えると、そのユーザは East、Allentown、および Yorktown コンテナ内のオブジェクトを管理できます。
- ユーザに YourCo に対する権利を与えると、そのユーザは図中のすべてのコンテナのすべてのオブジェクトを管理できます。

権利の割り当ての詳細については、69 ページの 「OpenText eDirectory での権利」 を参照してください。

Web ベースの管理ユーティリティ

OpenText Identity Console はブラウザベースのツールで、OpenText eDirectory オブジェクトを運用、管理、設定するために使用します。OpenText Identity Console を使用すると、ユーザに特定のタスクや責任を割り当てたり、それらのタスクを実行するために必要なツールおよびそれに伴う権利だけを付与したりすることができます。

OpenText Identity Console を実行するには、Microsoft Internet Explorer 6.0 SP1 以降 (推奨)、Mozilla 1.7 以降、または Mozilla Firefox 0.9.2 以降がインストールされているワークステーションが必要です。

重要 : これら以外の Web ブラウザでも OpenText Identity Console にアクセスできる場合がありますが、完全な機能は保証されません。

OpenText Identity Console を使用すると、次のようなスーパバイザの作業を実行できます。

- OpenText eDirectory への LDAP および XML ベースのアクセス設定
- ネットワークユーザ、デバイス、およびリソースを表すオブジェクトの作成
- 新規ユーザアカウント作成用のテンプレートの定義

- ネットワークオブジェクトの検索、変更、移動、および削除
- 管理権を委託する権利と職種の定義
- カスタムオブジェクトタイプとプロパティを作成するための OpenText eDirectory スキーマの拡張
- 複数のサーバでの OpenText eDirectory データベースのパーティション化とレプリカの作成
- DSRepair、DSMerge、バックアップおよび復元などの OpenText eDirectory 管理ユーティリティの実行

OpenText Identity Console を使用すると、他の管理機能も実行できます。OpenText Identity Console では、次の OpenText eDirectory タイルを使用できます。

- LDAP 環境設定
- ユニバーサルパスワードの強制
- 暗号化属性
- 認証管理
- PKI/Certificate
- フィルタ処理済レプリカ環境設定ウィザード
- ［SNMP］
- WAN Traffic Manager

Identity Console のインストール、設定、および実行の詳細については、『*OpenText™ Identity Console 管理ガイド* (https://www.netiq.com/documentation/identity-console/identity_console-admin/data/bookinfo.html)』を参照してください。

シングルログインと認証

OpenText eDirectory では、ユーザはグローバルディレクトリにログインするため、ユーザごとに複数のサーバやドメインのアカウントを管理する必要はありません。また、ドメイン間の信頼関係やパススルー認証を管理する必要もありません。

ディレクトリのセキュリティ機能の 1 つは、ユーザの認証です。ユーザがログインするためには、あらかじめユーザオブジェクトがディレクトリ内に作成されている必要があります。ユーザオブジェクトには、名前やパスワードといった、特定のプロパティがあります。

ユーザがログインすると、OpenText eDirectory は入力されたパスワードとディレクトリに格納されているそのユーザのパスワードを照合し、一致した場合にアクセスを許可します。

オブジェクトクラスとプロパティ

OpenText eDirectory オブジェクトの各タイプの定義を、オブジェクトクラスといいます。たとえば、「ユーザ」や「組織」は、オブジェクトクラスです。オブジェクトの各クラスには、それぞれ特定のプロパティがあります。たとえば、ユーザオブジェクトでは、名、姓、および他の多くのプロパティがあります。

スキーマでは、オブジェクトクラスとプロパティ、および包含ルール (どのコンテナにどのオブジェクトを保管するか) が定義されます。Openext eDirectory にはベーススキーマが付属しています。このベーススキーマは、ユーザまたはユーザが使用するアプリケーションによる拡張が可能です。スキーマの詳細については、51 ページの 「スキーマ」を参照してください。

コンテナオブジェクトは、他のオブジェジェクトを格納し、ツリーをさまざまな分岐に分割するために使用されます。一方、リーフオブジェクトはネットワークリソースを表します。

オブジェクトのリスト

次の表に、OpenText eDirectory のオブジェクトクラスを示します。サービスを追加した場合は、表内のオブジェクトクラス以外のオブジェクトクラスが OpenText eDirectory 内に新たに作成されることがあります。

OpenText eDirectory コンテナオブジェクトクラス

Identity Console アイコン	コンテナオブジェクト (略語)	説明
	ツリー	ツリーの開始点を表します。詳細については、29 ページの 「ツリー」を参照してください。
	カントリ (C)	ネットワークが存在する国を表します。その下には、国内の他のディレクトリオブジェクトが編成されます。詳細については、31 ページの 「国」を参照してください。
	ライセンスコンテナ (LC)	NLS (NetIQ Licensing Services) 技術を使用してライセンス許可証をインストールした場合やメータリング許可証を作成した場合に、自動的に作成されます。NLS 対応のアプリケーションをインストールすると、LC コンテナオブジェクトがツリーに追加され、ライセンス許可証リーフオブジェクトがそのコンテナに追加されます。
	組織 (O)	ディレクトリ内の他のオブジェクトの編成に使用されます。組織オブジェクトは、カントリオブジェクトの直下に配置されます (カントリオブジェクトを作成している場合)。詳細については、29 ページの「組織」を参照してください。

Identity Console アイコン	コンテナオブジェクト (略語)	説明
	部門 (OU)	ディレクトリ内の他のオブジェクトをさらに細かく編成するために使用されます。部門オブジェクトは、組織オブジェクトの直下に配置されます。詳細については、30 ページの 「部門」を参照してください。
	ドメイン (DC)	ディレクトリ内の他のオブジェクトをさらに細かく編成するために使用されます。ドメインオブジェクトは、Tree オブジェクトの下、または組織、部門、カントリ、および地域オブジェクトの下に作成されます。詳細については、32 ページの 「ドメイン」を参照してください。

OpenText eDirectory リーフオブジェクトクラス

Identity Console アイコン	Leaf Object (リーフオブジェクト)	説明
	AFP サーバ	OpenText eDirectory ネットワーク内のノードとして機能する、AppleTalk* ファイリングプロトコルサーバを表します。通常、複数の Macintosh* コンピュータに対するルータおよび AppleTalk サーバとしても機能します。
	別名	ディレクトリ内にあるオブジェクトの実際の位置を指します。別名を使用することによって、ディレクトリ内のディレクトリオブジェクトを、実際の場所とは異なる場所に存在するように表示できます。詳細については、44 ページの 「別名」を参照してください。
	Application (アプリケーション)	ネットワークアプリケーションを表します。アプリケーションオブジェクトによって、権利の割り当て、ログインスクリプトのカスタマイズ、およびアプリケーションの起動のような、管理作業を簡素化できます。
	コンピュータ	ネットワーク内のコンピュータを表します。
	ディレクトリマップ	ファイルシステム内のディレクトリを表します。詳細については、45 ページの 「ディレクトリマップ」を参照してください。
	グループ	ディレクトリ内のユーザオブジェクトのリストに名前を割り当てます。権利をそれぞれのユーザに割り当てる代わりに、グループに割り当てることができます。こうすると、グループ内の各ユーザに権利が与えられます。詳細については、36 ページの 「グループ」を参照してください。

Identity Console アイコン	Leaf Object (リーフオブジェクト)	説明
	ライセンス許可証	プロダクトライセンス許可証をデータベースのオブジェクトとしてインストールするために、NLS 技術とともに使用されます。NLS 対応アプリケーションをインストールすると、ライセンス許可証オブジェクトがライセンスプロダクトコンテナに追加されます。
	職種	組織内での地位や職種を定義します。
	プリントキュー	ネットワークのプリントキューを表します。
	Print Server	ネットワークのプリントサーバを表します。
	プリンタ	ネットワークのプリンタを表します。
	プロファイル	共通のログインスクリプトコマンドを共有するユーザグループが使用するログインスクリプトを表します。これらのユーザは同じコンテナに属する必要はありません。詳細については、46 ページの 「プロファイル」を参照してください。
	サーバ	任意のオペレーティングシステムが動作するサーバを表します。詳細については、33 ページの 「サーバ」を参照してください。
	テンプレート	新しいユーザオブジェクトに適用する標準のユーザオブジェクトプロパティを表します。
	不明	Identity Console にカスタムアイコンが存在しないオブジェクトを表します。
	User (ユーザ)	ネットワークを使用する人を表します。詳細については、34 ページの 「ユーザ」を参照してください。
	ボリューム	ネットワーク上の物理的なボリュームを表します。詳細については、34 ページの 「ボリューム」を参照してください。

コンテナオブジェクトクラス

- 29 ページの 「ツリー」
- 29 ページの 「組織」
- 30 ページの 「部門」
- 31 ページの 「国」
- 32 ページの 「ドメイン」

ツリー

ネットワーク内のサーバに OpenText eDirectory を初めてインストールすると、ツリーコンテナ (以前の［ルート］) が作成されます。最上位のコンテナである Tree コンテナには、通常、組織オブジェクト、カントリオブジェクト、または別名オブジェクトが格納されます。

Tree が表す内容

Tree コンテナはツリーの最上部を表します。

使用法

Tree は、包括的な権利の割り当てに使用します。Tree に対して行った権利の割り当ては、継承機能によって、ツリー内のすべてのオブジェクトに適用されます。69 ページの「OpenText eDirectory での権利」を参照してください。デフォルトで、トラスティ［Public］は Tree に対するブラウズ権を所有し、Admin は Tree に対するスーパバイザ権を所有します。

重要なプロパティ

- Tree オブジェクトには［名前］プロパティがあります。［名前］プロパティは、最初のサーバをインストールした時にユーザが指定したツリー名で、ツリービューに表示されます。
- ツリー名は 32 文字以下にする必要があります。

組織

ネットワーク内のサーバに OpenText eDirectory を初めてインストールすると、組織コンテナオブジェクトが作成されます。通常、組織コンテナは最上部の Tree コンテナの直下に作成され、コンテナ内には部門オブジェクトとリーフオブジェクトが格納されます。

デフォルトでは、最初の組織コンテナに、Admin という名のユーザオブジェクトが作成されます。

組織オブジェクトが表す内容

通常、組織オブジェクトは会社を表しますが、Tree の下に組織オブジェクトを追加作成することもできます。一般的に、組織オブジェクトの追加作成は、さまざまな地区で構成されるネットワークや、独立した複数の OpenText eDirectory ツリーがマージされているネットワークで行われます。

使用法

ツリーでの組織オブジェクトの運用方法は、ネットワークのサイズと構造により異なります。小規模のネットワークでは、1 つの組織オブジェクトの下にすべてのリーフオブジェクトを配置します。

大規模なネットワークでは、組織オブジェクトの下に部門オブジェクトを作成します。これにより、リソースの検索と管理を容易化できます。たとえば、社内の各部署や事業部ごとに部門オブジェクトを作成できます。

複数のサイトがあるネットワークでは、組織オブジェクトの下に各サイトを表す部門オブジェクトを作成します。ディレクトリを分割するためのサーバ数が十分にあれば (または設置の計画があれば)、このようにサイトの境界で論理的にパーティションを区切ることができます。

プリンタ、ボリューム、アプリケーションといった、社内全体で使用するリソースを共有しやすくするために、組織の直下に、対応するプリンタ、ボリューム、アプリケーションのオブジェクトを作成します。

重要なプロパティ

組織オブジェクトの最も有用なプロパティを次に示します。名前プロパティは必須です。すべてのプロパティの一覧を表示するには、OpenText Identity Console で組織オブジェクトを選択します。プロパティの各ページの説明を表示するには、[ヘルプ] をクリックします。

- 名前

  通常、名前プロパティは会社名と同じです。簡素化のために短くすることもできます。たとえば、会社名が Your Shoe Company の場合、YourCo とすることができます。

  組織名は、その下に作成されるすべてのオブジェクトのコンテキストの一部として使用されます。
- ログインスクリプト

  ログインスクリプトプロパティには、組織の直下にあるユーザオブジェクトが実行するコマンドが格納されます。これらのコマンドは、ユーザのログイン時に実行されます。
- 組織名の長さは 64 文字以下にする必要があります。

部門


ツリーを細分化するために部門 (OU) コンテナオブジェクトを作成することができます。部門は、OpenText Identity Console で、組織、国、または別の部門オブジェクトの下に作成されます。

部門には、ユーザオブジェクトやアプリケーションオブジェクトといった、他の部門やリーフオブジェクトを格納できます。

部門オブジェクトが表す内容

通常は、部門オブジェクトは 1 つの部署を表し、互いにアクセスする必要があるオブジェクトのセットを格納します。部門オブジェクトの主な格納内容として、ユーザのセット、およびユーザが使用するプリンタ、ボリューム、アプリケーションなどを挙げることができます。

部門オブジェクトの最上位レベルに配置された各部門は、WAN リンクごとに区切られたネットワークの各サイトを表します。

使用法

ツリーでの部門オブジェクトの運用方法は、ネットワークのサイズと構造により異なります。小規模のネットワークでは、部門オブジェクトを作成する必要がない場合もあります。

大規模なネットワークでは、組織オブジェクトの下に部門オブジェクトを作成します。これにより、リソースの検索と管理を容易化できます。たとえば、社内の各部署や事業部ごとに部門オブジェクトを作成できます。ユーザオブジェクトと、ユーザが頻繁に使用するリソースを一緒に部門オブジェクトに格納すると、管理が最も容易になります。

複数のサイトがあるネットワークでは、組織オブジェクトの下に各サイトを表わす部門オブジェクトを作成します。ディレクトリを分割するためのサーバ数が十分にあれば、このようにサイトの境界で論理的にパーティションを区切ることができます。

重要なプロパティ

部門オブジェクトの最も有用なプロパティを次に示します。名前プロパティは必須です。すべてのプロパティの一覧を表示するには、OpenText Identity Console で部門オブジェクトを選択します。プロパティの各ページの説明を表示するには、[ヘルプ] をクリックします。

- 名前

  通常、名前プロパティは部署名と同じです。簡素化のために短くすることもできます。たとえば、部署名が Accounts Payable の場合、省略して AP とすることができます。

  部門名は、その下に作成されるすべてのオブジェクトのコンテキストの一部として使用されます。
- ログインスクリプト

  ログインスクリプトプロパティには、部門の直下にあるユーザオブジェクトが実行するコマンドが格納されます。これらのコマンドは、ユーザのログイン時に実行されます。
- 部門名は 64 文字まで指定できます。

国

 OpenText Identity Console を使用して、国オブジェクトをツリーオブジェクトの直下に作成できます。国オブジェクトは、特定の X.500 グローバルディレクトリに接続する場合にのみ必要です。

国オブジェクトが表す内容

国オブジェクトは、ツリーの分岐の国名を表します。

使用法

ネットワークが複数の国に渡っている場合でも、管理者は通常、国オブジェクトを作成しません。これは、国オブジェクトがツリーに不要なレベルを追加するだけだからです。ネットワークが複数の国家で構成されている場合は、必要に応じて、Tree オブジェクトの下に 1 つ以上の国オブジェクトを作成できます。国オブジェクトには、組織オブジェクトのみ格納できます。

国オブジェクトを作成していない場合でも、後で必要になった時には、随時ツリーを変更して国オブジェクトを追加できます。

重要なプロパティ

- 国オブジェクトには、2 文字の名前プロパティがあります。国オブジェクト名には、US、UK、または DE といった、2 文字の標準コードが使用されます。
- カントリ名は 2 文字を超えることができません。

ドメイン


OpenText Identity Console を使用して、ドメインオブジェクトをツリーオブジェクトの直下に作成できます。また、組織、部門、国、および地域オブジェクトの下にも作成できます。

ドメインオブジェクトが表す内容

ドメインオブジェクトは、DNS ドメインのコンポーネントを表します。ドメインオブジェクトを使用すると、ドメインネームシステムによって示されるサービスリソースレコードの場所 (DNS SRV) に基づいて、ツリー内のサービスを検索できます。

ドメインオブジェクトを使用すると、ツリーは次のように表されます。

DS=Novell.DC=Provo.DC=USA

この例では、すべてのサブコンテナがドメインになっています。次のように、異なるツリーが混在する場合にもドメインオブジェクトを使用できます。

DC=Novell.O=Provo.C=USA

または

OU=Novell.DC=Provo.C=USA

通常、先頭のドメインは Tree 全体を表し、サブドメインはその Tree の下位の部分を表します。たとえば machine1.novell.com をツリーで表すと、DC=machine1.DC=novell.DC=com となります。ドメインは、OpenText eDirectory ツリーの設定で使用される一般的な方法です。コンテナおよびサブコンテナがすべて DC オブジェクトである場合は、オブジェクトを検索するときに、C、O、または OU を意識する必要はありません。

使用法

ドメイン名の長さは 64 文字以下にする必要があります。

リーフオブジェクトクラス

- 33 ページの 「サーバ」
- 34 ページの 「ボリューム」
- 34 ページの 「ユーザ」
- 36 ページの 「グループ」
- 39 ページの 「ネストされたグループ」
- 44 ページの 「別名」
- 45 ページの 「ディレクトリマップ」
- 46 ページの 「プロファイル」

サーバ

 サーバに OpenText eDirectory をインストールすると、そのサーバのオブジェクトがツリー内に自動作成されます。このオブジェクトクラスは、OpenText eDirectory が動作しているいずれかのサーバを表します。

サーバオブジェクトが表す内容

サーバオブジェクトは、OpenText eDirectory を実行しているサーバまたはバインダリベースのサーバを表します。

使用法

サーバオブジェクトはレプリケーション処理のリファレンスポイントの役目を果たします。バインダリベースのサーバを表すサーバオブジェクトでは、OpenText Identity Console でそのサーバのボリュームを管理できます。

重要なプロパティ

サーバオブジェクトの主なプロパティとして、ネットワークアドレスプロパティがあります。ネットワークアドレスプロパティには、そのサーバのプロトコルとアドレス番号が表示されます。これはパケットレベルでのトラブルシューティングに役立ちます。

すべてのプロパティの一覧を表示するには、OpenText Identity Console でサーバオブジェクトを選択します。プロパティの各ページの説明を表示するには、[ヘルプ] をクリックします。

ボリューム

サーバ上に物理ボリュームを作成すると、ツリー内にボリュームオブジェクトが自動作成されます。デフォルトでは、サーバ名にアンダースコアと物理ボリューム名を追加したものが、ボリュームオブジェクトの名前になります (たとえば YOSERVER_SYS)。

Linux ファイルシステムのパーティションは、ボリュームオブジェクトを使用して管理することはできません。ボリュームオブジェクトは OES Linux でのみサポートされます。

ボリュームオブジェクトが表す内容

ボリュームオブジェクトは、サーバ上の物理ボリューム (書き込み可能ディスクや CD などのストレージメディア) を表します。OpenText eDirectory 内のボリュームオブジェクトには、そのボリューム内のファイルやディレクトリに関する情報は含まれませんが、OpenText Identity Console を使用すれば、それらの情報にアクセスできます。ファイルおよびディレクトリに関する情報は、ファイルシステム自体に保存されます。

使用法

OpenText Identity Console で［**ボリューム**］アイコンをクリックすると、そのボリュームのファイルやディレクトリを管理できます。Identity Console により、ボリュームの空きディスク容量、ディレクトリエントリ領域、および圧縮の統計に関する情報が提供されます。

重要なプロパティ

必須の名前プロパティおよびホストサーバプロパティに加えて、ボリュームオブジェクトには他にも重要なプロパティがあります。

- 名前

  ツリー内のボリュームオブジェクトの名前です。デフォルトでは、この名前は物理ボリュームの名前に基づいて付けられますが、変更も可能です。

- Host Server

  ボリュームが存在するサーバの名前です。

- バージョン

  これは、ボリュームをホストしているサーバの OpenText eDirectory のバージョンです。

ユーザ

ユーザオブジェクトは、ログインで必要とされます。ツリーに最初のサーバがインストールされると、Admin というユーザオブジェクトが作成されます。初回ログイン時には、Admin としてログインします。

ユーザオブジェクトの作成またはインポートには、次の機能を使用できます。

- OpenText Identity Console

OpenText Identity Console の詳細については、『*OpenText™ Identity Console 管理ガイド* (https://www.netiq.com/documentation/identity-console/identity_console-admin/data/bookinfo.html)』を参照してください。

- データベースファイルからのバッチ処理

  バッチファイルの使用の詳細については、86 ページの 「OpenText eDirectory ツリーの設計」を参照してください。

ユーザオブジェクトが表す内容

ユーザオブジェクトはネットワークを使用するユーザを表します。

使用法

ネットワークを使用するユーザ全員に対して、ユーザオブジェクトを作成します。ユーザオブジェクトは個別に管理することもできますが、次のようにすると時間の節約になります。

- テンプレートオブジェクトを使用して、通常のユーザオブジェクトのデフォルトのプロパティを設定する。新しく作成するユーザに対して、自動的にテンプレートが適用されます (既存のユーザには適用されません)。
- ユーザのセットを一括管理するためのグループオブジェクトを作成する。
- コンテナオブジェクトをトラスティとして使用して、権利を割り当てる。これにより、権利の割り当てをコンテナ内のユーザオブジェクトすべてに適用できます。
- 複数のユーザオブジェクトは、<Shift> または <Ctrl> を押してクリックすると選択できます。これにより、選択したすべてのユーザオブジェクトのプロパティ値を一度に変更できます。

重要なプロパティ

ユーザオブジェクトには 80 を超えるプロパティがあります。すべてのプロパティの一覧を表示するには、OpenText Identity Console でユーザオブジェクトを選択します。プロパティの各ページの説明を表示するには、[ヘルプ] をクリックします。

ログイン名プロパティと姓プロパティは必須です。これら必須のプロパティおよび、他の有用なプロパティを次に示します。

- [Account Expiration Date] では、ユーザアカウントの有効期限を設定できます。有効期限を過ぎた後は、アカウントはロックされ、ユーザはログインできなくなります。
- [アカウント使用不可] には、アカウントがロックされ、ユーザがログインできない状態であることを示すシステム生成の値があります。ロックは、アカウントの有効期限が切れた場合や、ユーザが不正なパスワードを連続して何度も入力した場合などに発生します。
- [定期的なパスワード強制変更] によって、一定期間ごとにパスワードの変更をユーザに要求し、セキュリティの強化を図ることができます。
- [グループメンバーシップ] は、ユーザがメンバーとして含まれているすべてのグループオブジェクトを示します。
- [最終ログイン時刻] はシステムが生成するプロパティで、ユーザが最後にログインした日時を示します。

- ［姓］は必須ですが、OpenText eDirectory が直接使用することはありません。OpenText eDirectory ネームベースを利用するアプリケーションが、名、役職、位置、Fax 番号など、他の識別プロパティとともに、このプロパティを使用する場合があります。
- ［同時接続数の制限］では、ユーザがネットワークで開くことができるセッションの最大数を設定できます。
- ［ログイン名］は、OpenText Identity Console でユーザアイコンのそばに表示される名前です。これは、ログイン時にユーザが入力する名前でもあります。

  OpenText eDirectory では、各コンテナ内で固有のログイン名を使用する必要がありますが、ネットワーク内の別のコンテナ間では同じログイン名を使用できます。ただし、社内全体で固有のログイン名を使用した方が、管理を単純化できます。

  一般に、ログイン名には姓と名の組み合わせが使用されます。たとえば、Steve Jones の場合は、STEVEJ や SJONES のようになります。
- ［ログインスクリプト］では、個々のユーザオブジェクトの特定のログインコマンドを作成できます。ユーザのログイン時には、最初にコンテナログインスクリプトが実行されます。次に、ユーザオブジェクトがプロファイルオブジェクトのメンバーシップリストに登録されている場合は、プロファイルログインスクリプトが実行されます。最後に、ユーザのログインスクリプトが実行されます (スクリプトが存在する場合)。

  管理に要する時間を節約するため、ログインコマンドのほとんどの部分はコンテナのログインスクリプトに保管するようにします。ユーザログインスクリプトを編集すると、共通の必要条件に対する例外に対処できます。
- ［ログイン時間制限］では、ユーザがログインできる時刻と曜日を設定できます。
- ［ネットワークアドレス］には、ユーザのログイン元である IPX™ アドレスや IP アドレスをすべて示すシステム生成の値が格納されます。これらの値は、パケットレベルでのネットワークの問題のトラブルシューティングに役立ちます。
- ［パスワード要求］では、パスワードの入力をユーザに要求するかどうかを指定できます。他の関連プロパティでは、パスワード長などの一般的なパスワード制限を設定できます。
- ［Rights to Files and Directories］は、ユーザに割り当てられた、ファイルシステムに対するすべての権利を示します。OpenText Identity Console を使用して、ファイルおよびディレクトリに対するユーザの有効な権利 (他のオブジェクトから継承されたものを含む) を確認することもできます。

グループ

グループオブジェクトを作成すると、ユーザオブジェクトをセットで管理できます。

グループオブジェクトが表す内容

グループオブジェクトは、ユーザオブジェクトのセットを表します。

使用法

コンテナオブジェクトではコンテナ内のすべてのユーザオブジェクトを管理でき、グループオブジェクトでは 1 つまたは複数のコンテナ内のサブセットを管理できます。

グループオブジェクトは、次の 2 つの主な目的に対して使用されます。

- 多数のユーザオブジェクトに、一度に権利を与える。
- IF MEMBER OF 構文を使用して、ログインスクリプトコマンドを指定する。

スタティックグループ

スタティックグループでは、メンバーオブジェクトを明示的に指定します。各メンバーは、グループに明示的に割り当てられます。

これらのグループでは固定したメンバーのリストが示され、また、グループのメンバーリストと、オブジェクト上で属性を持つメンバーとの間の参照整合性を提供します。グループメンバーシップは、メンバーの属性によって明示的に管理されます。

ダイナミックグループ

ダイナミックグループでは、LDAP URL を使用して規則のセットを定義します。この規則に従って、OpenText eDirectory のユーザオブジェクトに一致したときに、グループのメンバーが定義されます。ダイナミックグループのメンバーは、URL に指定された検索フィルタによって定義される共通の属性を共有します。LDAP URL の形式に関する詳細は、RFC 2255 (http://www.ietf.org/rfc/rfc2255.txt) を参照してください。

ダイナミックグループを使用すると、グループのメンバーシップを評価する際に使用される条件を指定できます。グループの実際のメンバーは、OpenText eDirectory によって動的に評価されます。つまり、論理的にグループ化することでグループのメンバーを定義するため、OpenText eDirectory はグループのメンバーを自動的に追加または削除できます。この拡張性の高いソリューションによって、管理コストを低減でき、LDAP の通常のグループに高い柔軟性を補うことができます。

OpenText eDirectory では、任意の属性に基づいてユーザを自動的にグループ化したり、一致する識別名 (DN) を含む特定のグループに対して ACL を適用したりする場合に、ダイナミックグループを作成できます。たとえば、部署 = マーケティングという属性を持つすべての DN を自動的に含むグループを作成できます。部署 = マーケティングという検索フィルタを適用すると、部署 = マーケティングの属性を持つすべての DN を含むグループが検索結果として返されます。その後、このフィルタに基づく検索結果からダイナミックグループを定義できます。部署 = マーケティングという条件に一致するユーザがディレクトリに追加されると、このグループにも自動的に追加されます。部署が他の値に変更されたユーザ (またはディレクトリから削除されたユーザ) は、グループから自動的に削除されます。

OpenText eDirectory でダイナミックグループを作成するには、objectclass=dynamicGroup というタイプのオブジェクトを作成します。スタティックグループオブジェクトをダイナミックグループに変換するには、補助クラス dynamicGroupAux をグループオブジェクトに関連付けます。ダイナミックグループは、グループに関連付けられた memberQueryURL 属性を持ちます。

dgIdentity 属性は、ダイナミックグループオブジェクト上で、グループのダイナミックメンバーを拡張するために使用される証明書と権利を持つエントリの識別名に設定することができます。

グループは、memberQueryURL を使用して管理されます。基本的な memberQueryURL には、ベース DN、スコープ、フィルタ、およびオプション拡張があります。ベース DN は検索ベースを指定します。スコープはベース内の検索レベルを指定します。フィルタは、指定したスコープ内で選択されたエントリに基づく検索フィルタです。

注 : memberQueryURL によって作成されたリストに例外を設定するため、ダイナミックグループでもユーザを明示的に含めたり、除外したりできます。

ダイナミックグループは、OpenText Identity Console を使用して作成および管理できます。［**グループ管理**］タイルをクリックすると、グループ管理タスクにアクセスできます。

また、LDAP コマンドを使用してもグループを管理できます。ダイナミックグループに関連付けられた最も有用なプロパティは、dgIdentity および memberQueryURL です。

重要なプロパティ

グループオブジェクトの最も有用なプロパティは、メンバープロパティとファイル / ディレクトリへの権利プロパティです。すべてのプロパティの一覧を表示するには、OpenText Identity Console でグループオブジェクトを選択します。プロパティの各ページの説明を表示するには、［**ヘルプ**］をクリックします。

- グループの作成
- グループの削除
- グループの変更
- グループの追加または変更
- グループの検索

OpenText eDirectory 8.6.1 より前のデータベースのダイナミックグループを更新する

ダイナミックグループがローカルで作成されるか、同期の一部として取得されると、ダイナミックグループオブジェクトが作成されますが、ダイナミックグループの機能は、それらのオブジェクトに格納されるいくつかの内部値を必要とします。

古いサーバでもダイナミックグループを格納できますが、値を生成することはできません。ダイナミックグループは OpenText eDirectory 8.6.1 で導入されたためです。

OpenText eDirectory 8.6.2 では、OpenText eDirectory 8.6.1 データベースに適合させるために、8.6.1 以前のデータベースのダイナミックグループオブジェクトが自動で更新されました。

memberQueryURL の追加構文のサポート

memberQueryURL 属性は、ダイナミックグループのメンバーを計算するために OpenText eDirectory サーバが使用する検索フィルタを格納できます。

OpenText eDirectory 8.6.1 では、フィルタで使用される属性の構文は、次の基本的な文字列型のみに制限されていました。

- SYN_CE_STRING
- SYN_CI_STRING
- SYN_PR_STRING
- SYN_NU_STRING
- SYN_CLASS_NAME
- SYN_TEL_NUMBER
- SYN_INTEGER
- SYN_COUNTER
- SYN_TIME
- SYN_INTERVAL
- SYN_BOOLEAN
- SYN_DIST_NAME
- SYN_PO_ADDRESS
- SYN_CI_LIST
- SYN_FAX_NUMBER
- SYN_EMAIL_ADDRESS

OpenText eDirectory 8.7.3 以降では、次の属性構文が memberQueryURL の値として追加でサポートされています。

- SYN_PATH
- SYN_TIMESTAMP
- SYN_TYPED_NAME

OpenTexte OpenText eDirectory 8.6.1 および OpenText eDirectory 8.7.x の両方では、SYN_OCTET_STRING や SYN_NET_ADDRESS のようなバイナリ構文は、memberQueryURL 検索フィルタでサポートされません。

詳細については、「「How to Manage and Use Dynamic Groups in OpenText eDirectory (OpenText eDirectory でダイナミックグループを管理および使用する方法)」(http://support.novell.com/techcenter/articles/ana20020405.html)」を参照してください。

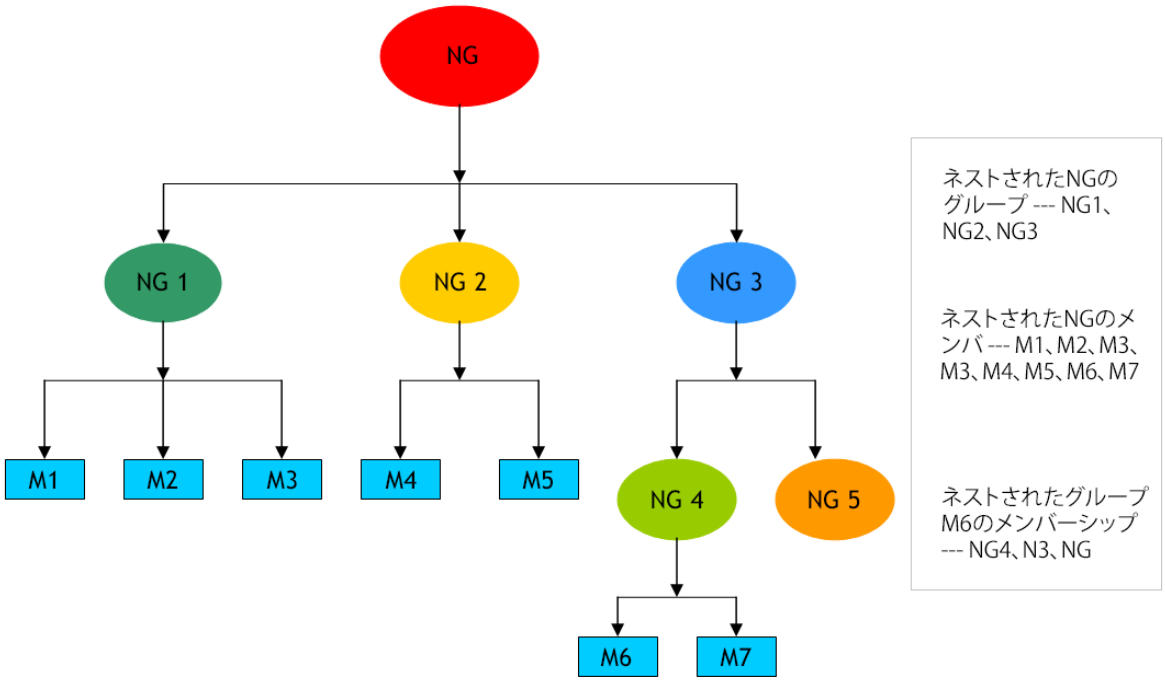
ネストされたグループ

ネストされたグループにより、グループをグループ化することができ、より構造化された形態のグループ化が可能になります。groupMember 属性はネストされたグループを指定し、そのメンバーは、それを含有するネストされたグループオブジェクトのネストされたメンバーになります。グループオブジェクトは groupMember 属性で静的に指定されます。

他のグループを含んでいるグループを「含有するグループ」といい、このグループに含まれているグループを「含有されるグループ」といいます。OpenText eDirectory は、スタティック (静的) グループとダイナミック (動的) グループの両方のネストをサポートしています。最大で 200 レベルまでのネストが可能です。

重要 : ネストは、ローカルサーバ内のみでサポートされます。含有されるグループがローカルサーバ上に見つからない場合、そのメンバーは、含有するグループのネストされたメンバーとしてリストされません。

図 1-4 *ネストされたグループ*



OpenText Identity Console または LDAP のツールを使用して、ネストされたグループを作成できます。

- 40 ページの 「LDAP ツールを使用してネストされたグループを作成する」
- 41 ページの 「OpenText Identity Console を使用してネストされたグループを作成する」

LDAP ツールを使用してネストされたグループを作成する

LDAP ツールを使用して、ネストされたグループを作成することができます。新しい補助クラス nestedGroupAux を構造化クラス Group と一緒に用いると、nestedGroupAux はネストされたグループを表します。この補助クラスを既存のスタティックグループオブジェクトに追加すると、それがネストされたグループに変換されます。

含有されるグループと含有するグループの両方が、ネストされたグループオブジェクトでなければなりません。含有されるグループがネストされたグループである場合にのみ、それに groupMembership 属性 (groupMembership 属性はスタティックグループには含まれませ

ん) を設定して、含有するグループを指定することができます。含有されるグループのタイプが含有するグループと同じでない場合は、含有されるグループのスタティックメンバーのみが、ネストされたメンバーとしてリストされます。

LDIF ファイルおよび LDAP ツールを使用して、そのようなグループを管理できます。ネストされたグループに関連する最も役立つプロパティは、groupMember および nestedConfig です。

OpenText Identity Console を使用してネストされたグループを作成する

OpenText Identity Console タイルを使用して、ネストされたグループを作成したり、スタティックグループをネストされたグループに変更して別のグループにそれを関連付けたりすることができます。

1. 管理者の資格情報を使って OpenText Identity Console にログインし、左側のパネルから［**グループの管理 (Groups Management)**］>［**グループの作成**］を選択して、スタティックグループを作成します。
2. 左側のパネルから［**グループ**］>［**グループの作成**］を選択し、［**ネストされたグループを有効にする**］チェックボックスを選択してネストされたグループを作成して、［**OK**］をクリックします。
3. ［**グループ**］>［**グループの変更 (Modify Group)**］を選択し、［**ネストされたグループ**］チェックボックスをオンにしてネストされたグループを変更し、［**保存**］をクリックして［**OK**］をクリックします。
4. ネストされたグループを変更するには、［**グループ**］>［**グループの変更 (Modify Group)**］を選択し、変更するグループをブラウズして選択します。
5. スタティックグループを関連付けるには、［**ネスト**］タブ >［**グループメンバー**］タブを選択し、スタティックグループをブラウズして選択します。
6. ［**保存**］をクリックして［**OK**］をクリックすると、スタティックグループがネストされたグループに変換されます。

   ネストされたグループに変換されたスタティックグループが、スタティックグループのメンバーになります。

スタティックグループのメンバーシップの詳細を確認するには、［**グループ**］>［**グループの変更 (Modify Group)**］を選択し、スタティックグループを選択します。［**ネスト**］>［**グループメンバーシップ**］タブを選択して、スタティックグループのメンバーシップ情報を確認します。メンバーをグループに追加するには［ ＋ ］をクリックし、グループに割り当てられているメンバーを除外または削除するには［ 🗑 ］をクリックします。

ネストされたグループのプロパティ

- member

  ネストされたグループのメンバー属性は、ネストされたグループのすべての直接メンバーを一覧表示し、ネストされたメンバーグループをそのメンバーに展開します。

注 : ネストされたグループメンバーシップの列挙は、メンバー属性にアサーションがある場合にのみ実行されます。したがって、多数のユーザを含むネストされたメンバーグループが多数ある場合、グループメンバーシップの合計を列挙すると、OpenText eDirectory に負荷が発生する可能性があります。

- groupMember

  ネストされたグループの groupMember 属性は、ネストされたメンバーグループをそのメンバーに展開し、ネストされたグループ名のみを表示します。

- グループメンバーシップ

  groupMembership は、このオブジェクト (通常はユーザオブジェクト) が属するグループを指定します。この属性は nestedGroupAux クラスに関連付けられ、このグループがグループメンバーとなっているネストされたグループの DN を保持します。グループオブジェクトに関連付けられている場合、この属性は、そのグループがメンバー (正確には groupMember) になっているネストされたグループを示します。member および groupMember と似た点として、groupMembership のリストには、ネスト関係を通してそのグループが groupMembership を持っているすべてのネストされたグループが含まれます。さらに、groupMembership 属性には、nestedConfig も適用されます。グループメンバーオブジェクト以外に対しては、個々のグループの nestedConfig が使用されます。

ネストされたグループの操作

1. groupMember 属性を使用して、あるグループを別のグループのメンバーにすることができます。含有されるグループと含有するグループのどちらのグループオブジェクトにも、ネストされたグループ補助クラスが関連付けられている必要があります。

   ```
   dn: cn=finance,o=nov

   objectclass: group

   objectclass: nestedGroupAux

   groupMember: cn=accounts,o=nov

   member: cn=jim,o=nov


   dn: cn=accounts,o=nov

   objectclass: group

   objectclass: nestedGroupAux

   member: cn=allen,o=nov

   member: cn=ESui,o=nov

   member: cn=YLi,o=nov
   ```

2. 含有されるグループと含有するグループの両方がサーバ上にローカルに存在する場合は、ネストされたグループの member 属性を読み込むと、含有されるグループのメンバーも返されます。

```
dn: cn=finance,o=nov
member: cn=jim,o=nov
member: cn=allen,o=nov
member: cn=ESui,o=nov
member: cn=YLi,o=nov
```

groupMember 属性に関してもこれが当てはまります。

3. groupMembership は、member 属性の相互性属性です。つまり、ユーザオブジェクト cn=allen,o=nov は、cn=accounts,o=nov というグループ DN が設定された groupMembership 属性を持つ必要があります。cn=accounts,o=nov グループの groupMembership には、cn=finance,o=nov が設定されている必要があります。cn=allen,o=nov ユーザオブジェクトの groupMembership 属性を読み込むと、両方のグループが返されます。

```
dn: cn=allen,o=nov
groupMembership: cn=accounts,o=nov
groupMembership: cn=finance,o=nov
```

4. ネストされたグループに ACL を割り当てることができます。その場合、ネストされたグループのメンバーであるすべてのオブジェクトが権利を獲得します。［割り当てられた権利］フィールドでは、割り当てられている権利に加えて、ネストされた ACL のビット (0x80000000) をさらに設定する必要があります。

```
dn: cn=finance,o=nov
groupMember: cn=accounts,o=nov


dn: cn=accounts,o=nov
member: cn=allen,o=nov


dn: ou=MyCo,o=nov
objectclass: Organizational Unit
ACL: 2147483650#entry#cn=finance,o=nov#[All Attributes Rights]
```

権利の値「– 2147483650 (0x80000002)」には、ネストされた ACL (0x80000000) と読み取り権限ビット (0x00000002) が設定されています。したがって、ユーザオブジェクト cn=allen,o=nov には、ネストされたグループ cn=finance,o=nov を介して、MyCo オブジェクトのすべての属性に対する読み取り権限が付与されます。

5. アプリケーションでは、member、groupMember、および groupMembership 属性に対するフィルタアサーションを使用できます。上記の例では、member=cn=allen,o=nov のアサーションによって以下が返されます。

```
dn: cn=accounts,o=nov
dn: cn=finance,o=nov
```

groupMembership=cn=finance,o=nov のアサーションによって次のオブジェクトが返されます。

```
dn: cn=allen,o=nov
dn: cn=jim,o=nov
dn: cn=ESui,o=nov
dn: cn=YLi,o=nov
dn: cn=accounts,o=nov
```

注 : 上記の例ではいずれも、ネストのレベルに制限はありません。上記のいずれの属性を読み込むときにも、ネストされたグループでループ検出が実行されます。

制限

- ネストされた関係の範囲がローカルサーバを超えることはありません。関係するオブジェクト、ユーザ、グループはサーバ上にローカルに存在する必要があります。
- メンバーシップリストでは、重複排除は行われません。
- 古い OpenText eDirectory サーバ (バージョン 8.8 SP1 またはそれ以前) では、ネストされた ACL もネストセマンティックもサポートされません。

別名


ツリー内の別のオブジェクトをポイントする別名オブジェクトを作成できます。別名オブジェクトによって、ユーザは自分の属するコンテナの外部にあるオブジェクトに、ローカル名を付けることができます。

コンテナの名前を変更するときには、必要に応じて、元のコンテナの位置に新しい名前をポイントする別名を作成できます。これにより、コンテナ内のオブジェクトを参照するログインスクリプトコマンドやワークステーションは、コンテナ名が更新されていなくても対象のオブジェクトにアクセスできます。

別名オブジェクトが表す内容

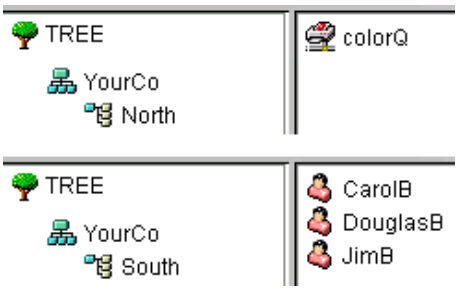
別名オブジェクトは、コンテナやユーザなど、ツリー内の別のオブジェクトを表します。別名オブジェクト自体にはトラスティ権はありません。別名オブジェクトに与えられたトラスティ権は、その別名オブジェクトが示している実際のオブジェクトに適用されます。ただし、別名をトラスティ割り当ての対象とすることもできます。

使用法

別名オブジェクトを作成すると、名前の解決が容易になります。オブジェクトの命名規則では、現在のコンテキストのオブジェクトに対する命名が最も簡単なため、現在のコンテキストに、現在のコンテキストの外部のリソースをポイントする別名オブジェクトを作成します。

たとえば、図 1-5 に示すように、ユーザが South コンテナにログインし、現在のコンテキストを確立する場合に、North コンテナの ColorQ というプリントキューオブジェクトにアクセスする必要があるとします。

図 1-5 コンテナの例



に示されるように、South コンテナに別名オブジェクトを作成できます。図 1-6

図 1-6 OpenText eDirectory Container の別名オブジェクト



別名オブジェクトによって、元の ColorQ オブジェクトがポイントされるため、South コンテナでは ColorQ をローカルオブジェクトとして印刷設定できます。

重要なプロパティ

別名オブジェクトには別名元オブジェクトプロパティがあり、このプロパティによって、別名オブジェクトと元のオブジェクトとが関連付けられます。

ディレクトリマップ

ディレクトリマップオブジェクトは、サーバのファイルシステム内のパスへのポインタです。これにより、ディレクトリをより簡単に参照できます。

ネットワークにボリュームがない場合は、ディレクトリマップオブジェクトを作成することはできません。

ディレクトリマップオブジェクトが表す内容

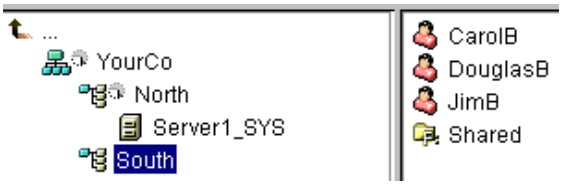
ディレクトリマップオブジェクトは、ボリューム上のディレクトリを表しますそれに対し、別名オブジェクトはオブジェクトを表します。

使用法

ディレクトリマップオブジェクトは、ログインスクリプトでのドライブマッピングを単純化するために作成します。ディレクトリマップオブジェクトを使用すると、複雑なファイルシステムパスを簡単な名前にすることができます。

また、ファイルの場所を変更した場合でも、新しい場所を参照するように、ログインスクリプトやバッチファイルを変更する必要がありません。ディレクトリマップオブジェクトを編集するだけです。たとえば、図 1-7 に示すように、South コンテナのログインスクリプトを編集するとします。

***図 1-7** OpenText eDirectory コンテナの例*



ドライブをボリューム sys: 上の Shared ディレクトリにマッピングするコマンドは、次のようになります。

```
MAP N:=sys.North.:Shared
```

共有ディレクトリマップオブジェクトを作成した場合、マップコマンドは、次のようにさらに簡単になります。

```
MAP N:=Shared
```

重要なプロパティ

ディレクトリマップオブジェクトには、次のようなプロパティがあります。

- 名前

  ディレクトリ内のオブジェクト (たとえば、Shared) を指定します。名前プロパティは MAP コマンドで使用されます。

- ボリューム

  Sys.North.YourCo のような、ディレクトリマップオブジェクトが参照するボリュームオブジェクトの名前が格納されます。

- パス

  ボリュームのルートからのパスでディレクトリを指定します (例 : public\winnt\nls\english)。

プロファイル

プロファイルオブジェクトは、ログインスクリプトの管理に役立ちます。

プロファイルオブジェクトが表す内容

プロファイルオブジェクトは、コンテナログインスクリプトの後、およびユーザログインスクリプトの前に実行されるログインスクリプトを表します。

使用法

特定のユーザのみを対象にログインスクリプトコマンドを実行したい場合は、プロファイルオブジェクトを作成します。対象のユーザには、同一コンテナ内のユーザだけでなく、異なるコンテナ内のユーザも指定できます。プロファイルオブジェクトを作成した後は、プロファイルのログインスクリプトプロパティにコマンドを指定します。続いて、該当のユーザオブジェクトをプロファイルオブジェクトのトラスティに指定し、それらのユーザオブジェクトのプロファイルメンバーシッププロパティにそのプロファイルオブジェクトを追加します。

重要なプロパティ

プロファイルオブジェクトには、次の 2 つの重要なプロパティがあります。

- ログインスクリプト

  そのプロファイルのユーザに対して実行するコマンドを格納します。
- ファイル / ディレクトリへの権利

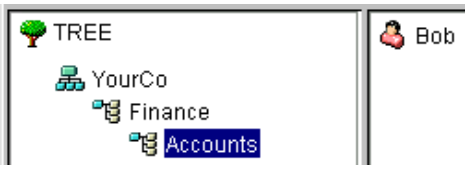
  ログインスクリプトに INCLUDE ステートメントを使用した場合は、プロファイルオブジェクトに、ファイル / ディレクトリへの権利プロパティに指定されているファイルに対する権利を与える必要があります。

コンテキストと命名規則

オブジェクトのコンテキストは、ツリー内でのそのオブジェクトの位置を表します。コンテキストは、DNS ドメインとほぼ同じです。

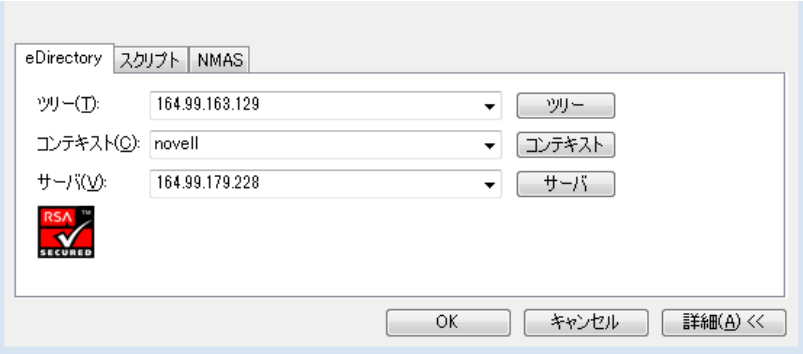
図 1-8 は、ユーザ Bob が、YourCo という組織の Finance 部門の中の Accounts 部門に所属していることを示しています。

図 1-8 *OpenText eDirectory コンテナの例*



ただし、オブジェクトのコンテキストを OpenText eDirectory ユーティリティで表現する必要がある場合もあります。たとえば、図 1-9 に示すように、Bob のワークステーションを設定するときには、名前のコンテキストを指定する必要があります。

図 1-9 Novell Client の NDS ページ



コンテキストは、対象のオブジェクトとツリーの最上部との間に存在する各コンテナをピリオドで区切ったリストとして指定します。

識別名

オブジェクトの識別名とは、オブジェクト名にコンテキストを付けたものです。たとえば、ユーザオブジェクト Bob の識別名は、Bob.Accounts.Finance.YourCo です。

タイプ付きの名前

タイプ付きの名前が、OpenText eDirectory ユーティリティに表示されることがあります。タイプつきの名前には、次の表に示されたようなオブジェクトタイプの略語があります。

オブジェクトクラス	タイプ	省略形
すべてのリーフオブジェクトクラス	Common Name	CN
組織	組織	O
部門	部門	OU
国	国	C
地域	地域または都道府県	L または S

OpenText eDirectory では、タイプ付きの名前の作成に、タイプの略語、等号、およびオブジェクト名を使用します。たとえば、Bob のタイプ付きの名前は、CN=Bob となります。Bob のタイプ付きの完全な名前は、CN=Bob.OU=Accounts.OU=Finance.O=YourCo となります。OpenText eDirectory ユーティリティにおいて、タイプ付きの名前は、タイプなしの名前と互換性があります。

ネームレゾリューション

ディレクトリツリー内のオブジェクトの位置を見つけるために OpenText eDirectory が使用するプロセスのことを、ネームレゾリューション (名前解決) といいます。OpenText eDirectory ユーティリティでオブジェクト名を使用すると、現在のコンテキストまたはツリーの最上部を基準として名前の解決を行います。

現在のワークステーションのコンテキスト

ネットワークソフトウェアの実行時には、ワークステーションにはコンテキストが設定されます。このコンテキストによって、ネットワーク内のワークステーションの位置が相対的に指定されます。たとえば、Bob のワークステーションでは、現在のコンテキストが次のように設定されます。

```
Accounts.Finance.YourCo
```

次のセクションで説明するように、現在のコンテキストは、先頭のピリオド、相対命名、および後続ピリオドの使用法を理解するうえで重要です。

先頭ピリオド

現在のコンテキストがどこに設定されているかに関わりなく、ツリーの最上部から名前を解決するには、先頭ピリオドを使用します。次の例では、先頭ピリオドが CX(コンテキストの変更) ユーティリティに、ツリーの最上部を基準として名前を解決するよう指定しています。

```
CX .Finance.YourCo
```

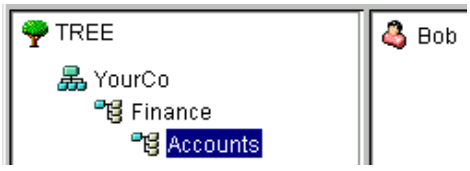
OpenText eDirectory はこのコマンドを、「ツリーの最上部から名前の解決を行い、YourCo コンテナにある Finance コンテナにコンテキストを変更する」と解釈します。

相対命名

相対命名とは、ツリーの最上部ではなく、ワークステーションの現在のコンテキストを基準に、名前を解決することを意味します。先頭ピリオドはツリーの最上部からの名前解決を表すため、相対命名の場合は先頭ピリオドを使用しません。

たとえば、ワークステーションの現在のコンテキストが Finance に設定されているとします。詳細については、図 1-10 を参照してください。

***図 1-10** OpenText eDirectory コンテナの例*



Bob の相対オブジェクト名は、次のようになります。

```
Bob.Accounts
```

OpenText eDirectory は、この名前を「現在のコンテキスト Finance から解決される Accounts に属する Bob」と解釈します。

後続ピリオド

後続ピリオドは、相対命名でのみ使用されます。したがって、先頭ピリオドと後続ピリオドの両方を使用することはできません。後続ピリオドは、OpenText eDirectory が名前解決を開始するコンテナを変更します。

後続ピリオド 1 つにつき、解決地点がツリーの上部に向かって 1 コンテナずつ引き上げられます。たとえば、図 1-11 の例は、ワークステーションの現在のコンテキストを Timmins から Allentown に変更する場合を示しています。

図 1-11 *OpenText eDirectory コンテナの例*



この場合の適切な CX コマンドでは、次に示すように、後続ピリオドを付けた相対命名を使用します。

```
CX Allentown.East..
```

OpenText eDirectory はこのコマンドを、「現在のコンテキストの 2 つ上にあるコンテナから名前の解決を行い、East にある Allentown にコンテキストを変更する」と解釈します。

同様に、Bob が Allentown コンテナに属し、ワークステーションの現在のコンテキストが Timmins の場合、Bob の相対名は次のようになります。

```
Bob.Allentown.East..
```

Linux でのコンテキストと名前付け

Linux のユーザアカウントを OpenText eDirectory に移行する場合、ユーザの名前付けに OpenText eDirectory のコンテキストは使用されません。

スキーマ

スキーマは、ツリー内で作成できるオブジェクトのタイプ (ユーザ、プリンタ、グループなど)、およびオブジェクトの作成時に使用する必須情報とオプション情報を定義します。各オブジェクトには、そのオブジェクトタイプのスキーマクラスが定義されています。

製品に元から付属しているスキーマは、ベーススキーマといいます。新しいクラスや新しい属性の追加など、なんらかの形でベーススキーマが変更されると、そのスキーマは拡張スキーマと見なされます。

スキーマを必ずしも拡張する必要はありませんが、拡張することも可能です。OpenText Identity Console のスキーマ役割を使用すれば、組織のニーズに合わせてスキーマを拡張できます。たとえば、組織の従業員たちに特殊な履き物が必要で、従業員の靴のサイズを把握しておく必要があるとします。この場合は、Shoe Size という新しい属性を作成し、この属性を補助クラスに追加することができます。その後、その補助クラスを使用して、ユーザオブジェクトを必要に応じて拡張できます。補助クラスの作成の詳細については、149 ページの 「補助クラスを作成する」を参照してください。

OpenText eDirectory スキーマの操作の詳細については、145 ページの第 5 章「スキーマの管理」を参照してください。

スキーマ管理

OpenText Identity Console のスキーマ役割によって、ツリーへのスーパバイザ権を持っているユーザは、そのツリーのスキーマをカスタマイズできます。スキーマ役割、およびその関連タスクは、OpenText Identity Console の［スキーマ管理］ページで使用できます。

スキーマ役割の使用目的

- スキーマ内のすべてのクラスおよび属性の一覧を表示する。
- 構文やフラグなどの属性についての情報を表示する。
- 既存のスキーマにクラスまたは属性を追加して、スキーマを拡張する。
- 名前を付けてから、属性、フラグ、追加できるコンテナ、および属性の継承元のペアレントクラスを指定して、クラスを作成する。
- 名前を付けてから、構文およびフラグを指定して、属性を作成する。
- 既存のクラスにオプションの属性を追加する。
- 使用されていない、または必要のなくなったクラスや属性を削除する。

スキーマクラス、属性、および構文

- 52 ページの 「Classes (クラス)」
- 52 ページの 「属性」
- 52 ページの 「構文」
- 56 ページの 「High-Valued 属性 (HVA)」

Classes (クラス)

クラスとは、ディレクトリオブジェクトのテンプレートのようなものです。ディレクトリオブジェクトとは、データを挿入されたクラスです。つまり、次のように表すことができます。

CLASS + DATA = DIRECTORY OBJECT

各クラスは、クラス名、継承クラス (そのクラスがクラス階層の最上位である場合を除く)、クラスフラグ、および属性のグループを持っています。クラスは、ディレクトリオブジェクト (ユーザ、プリンタ、キュー、サーバなど) と同じように命名されますが、単なる構造であり内容はありません。

継承クラスとは、他のオブジェクトクラスを定義するときの開始点となるクラスです。継承クラスの属性はすべて、クラス階層でそのクラスの下位に位置するクラスへ継承されます。

クラス階層は、あるクラスがどのようにペアレントクラスと関連付けられているかを示します。クラス階層により、類似したクラスが関連付けられ、属性の継承が可能になります。また、クラスを格納できる有効なコンテナのタイプも定義されます。

クラスの作成時には、クラス階層と追加属性を使用して各クラスをカスタマイズできます。継承クラスを指定することによって、階層内の上位のクラスからその属性とフラグのすべてを新しいクラスに継承できます。さらに、継承された属性クラスに追加する属性を 1 つまたは複数選択することによって、新しいクラスをカスタマイズできます。追加の属性は、必須属性、ネーミング属性、またはオプション属性として選択できます。

オプションの属性を追加して、既存のクラスを変更することもできます。

属性

属性とは、OpenText eDirectory データベース内のデータフィールドのことです。たとえば、クラスが記入用紙のようなものだとすれば、属性は記入用紙における 1 つの記入欄です。属性の作成時に、名前 (*姓*や*社員番号*など) と、構文タイプ (*文字列*や*数字*など) が指定されます。その後、その属性はスキーママネージャの属性リストで使用できるようになります。

注 : OpenText eDirectory では、レプリケーションの問題のため、ストリーム属性タイプ以外の属性に 60KB または 30,000 文字を超える値を含めることができません。ユーザやアプリケーションが、その制限を超える文字列またはバイナリ属性の値を設定すると、OpenText eDirectory は値が長すぎることを示す -649 エラーを返します。

構文

構文にはいくつかの選択可能なオプションがあります。これらの構文オプションは、各属性で入力するデータのタイプを指定するために使用されます。構文は属性の作成時にのみ指定できます。後から構文を変更することはできません。利用可能な構文は次のようなものです。

- Back Link (バックリンク)

オブジェクトを参照する他のサーバの追跡に使用されます。また、OpenText eDirectory の内部管理目的で使用されます。

- ブール

  TRUE(1) または FALSE(0) の値をとる属性に使用されます。この構文タイプには、単一の値のフラグが設定されます。

- Case Exact String (大 / 小文字一致文字列)

  比較演算で大文字 / 小文字が区別される Unicode 文字列を値としてとる属性に使用されます。2 つの「大文字小文字を区別する文字列」は、長さが等しく、対応する文字 (大文字 / 小文字の区別を含む) が同一の場合に一致とみなされます。

- Case Ignore List (大 / 小文字無視リスト)

  比較演算で大文字 / 小文字が区別されない Unicode 文字列の順序列を値としてとる属性に使用されます。2 つの「大文字小文字を無視するリスト」は、文字列の数が等しく、対応するすべての文字列が同じ場合 (つまり、長さと対応する文字が同一の場合) に一致とみなされます。

- Case Ignore String (大 / 小文字無視文字列)

  比較演算で大文字 / 小文字が区別されない Unicode 文字列を値としてとる属性に使用されます。2 つの「大文字小文字を無視する文字列」は、長さが等しく、対応する文字があらゆる面で (ただし、大文字 / 小文字の区別を除く) 同一の場合に一致とみなされます。

- クラス名

  オブジェクトクラスの名前を値としてとる属性に使用されます。2 つのクラス名は、長さが等しく、対応する文字があらゆる面で (ただし、大文字 / 小文字の区別を除く) 同一の場合に一致とみなされます。

- カウンタ

  増分変更された符号付き整数を値としてとる属性に使用されます。カウンタによって定義された属性は、単一の値の属性です。カウンタは、この構文の属性に値が追加されると足し算で合計に追加され、この構文の属性から値が削除されると引き算で合計から差し引かれるという点で、整数とは異なります。

- 識別名

  OpenText eDirectory ツリー内のオブジェクト名を値としてとる属性に使用されます。DN(識別名) では大文字 / 小文字が区別されませんが、ネーミング属性の 1 つでは大文字 / 小文字が区別されます。

- EMail Address (電子メールアドレス)

  バイナリ情報の文字列を値としてとる属性に使用されます。OpenText eDirectory は、この構文の内容の内部構造については想定しません。

- Fax 番号

  国際電話番号と、推奨 T.20 に従って形式設定されたオプションのビット文字列の格納を規定する E.123 標準に準拠した文字列を指定します。2 つの Fax 番号値は、長さが等しく、対応する文字列が同一 (ただし、比較処理で無視されるスペースとハイフンを除く) の場合に、一致とみなされます。

- ［保持］

符号付き整数の値を持つアカウント数量の属性に使用されます。この構文は、アカウント数量 (トランザクションが完了するまでの間、サブジェクトのクレジット限度に対して暫定的に保持される金額) を表します。保持量は、カウンタ構文と同じように扱われ、新しい値がベース合計に加算されるか、ベース合計から減算されます。計算された保持量が 0 になると、保持レコードは削除されます。

- 整数

  符号付き数値として表される属性に使用されます。2 つの整数値は、値が同一である場合に一致とみなされます。順序付けの比較では、符号付き整数ルールが使用されます。

- 整数 64

  64 ビット整数値として表される属性に使用されます。整数 64 属性は Microsoft Large Integer Syntax をサポートし、large-integer 値や 1970 年より前または 2038 年より後の日付を格納するために使用できます。

注 : OpenText eDirectory では、内部タイムスタンプ用に既存の構文と 32 ビット値を使用します。

- Interval

  符号付き整数を値としてとり、時間の間隔を表す属性に使用されます。間隔構文では、整数構文と同じ表現が使用されます。Interval (間隔) 値は時間間隔の秒数です。

- Net Address (ネットアドレス)

  サーバ環境でのネットワーク層アドレスを表します。このアドレスは、バイナリ形式です。2 つのネットアドレスは、各アドレスのタイプ、長さ、および値が一致する場合に、一致とみなされます。

- Numeric String (数値文字列)

  CCITT X.208 定義で数値文字列として定義されている数値整数を値としてとる属性に使用されます。2 つの数値文字列は、長さが等しく、対応する文字が同一の場合に一致とみなされます。数値文字列文字セットにおいて有効な文字は、数字 (0 ～ 9) およびスペースのみです。

- Object ACL (オブジェクト ACL)

  アクセス制御リスト (ACL) エントリを表す値をとる属性に使用されます。Object ACL (オブジェクト ACL) 値は、オブジェクトまたは属性のいずれかを保護できます。

- Octet List (オクテットリスト)

  バイナリ情報またはオクテット文字列の順序付き文字列シーケンスを表します。オクテットリストは、保存済みリストのサブセットである場合に、保存済みリストと一致するとみなされます。2 つのオクテットリストは、長さが等しく、対応するビット列 (オクテット) が同一の場合に一致とみなされます。

- Octet String (オクテット文字列)

  OpenText eDirectory によって解釈されないバイナリ情報の文字列を値としてとる属性に使用されます。これらのオクテット文字列は、非 Unicode 文字列です。2 つのオクテット文字列は、長さが等しく、対応するビット列 (オクテット) が同一の場合に一致とみなされます。

- パス

  ファイルシステムパスを表す属性であり、サーバ上でファイルを見つけるために必要なすべての情報が格納されます。2 つのパスは、長さが等しく、対応する文字 (大文字 / 小文字の区別を含む) が同一の場合に、一致とみなされます。

- Postal Address (住所)

  住所を表す Unicode 文字列を値としてとる属性に使用されます。通常、住所の属性値は、勧告 F.401 に基づいて MHS Unformatted Postal O/R Address Specification バージョン 1 から選択された属性で構成されます。この値は 6 行 (各行は 30 文字) に制限されます (郵送先国名を含む)。2 つの住所は、文字列の数が等しく、対応するすべての文字列が同じ場合 (つまり、長さと対応する文字が同一の場合) に一致とみなされます。

- 印刷可能文字列

  CCITT X.208 に定義されている印刷可能文字列を値としてとる属性に使用されます。印刷可能文字セットは、次のものから構成されます。

  - 英大文字および英小文字
  - 数字 (0 〜 9)
  - スペース
  - アポストロフィ (')
  - 左カッコおよび右カッコ ( )
  - プラス記号 (+)
  - カンマ (,)
  - ハイフン (-)
  - ピリオド (.)
  - スラッシュ (/)
  - コロン (:)
  - 等号 (=)
  - 疑問符 (?)

  2 つの印刷可能文字列は長さが等しく、対応する文字が同一の場合に一致とみなされます。大文字 / 小文字は区別されます。

- Replica Pointer (レプリカポインタ)

  パーティションレプリカを表す値をとる属性に使用されます。OpenText eDirectory ツリーのパーティションでは、複数のサーバにレプリカを置くことができます。この構文は、次の 6 つのコンポーネントで構成されます。

  - サーバ名
  - レプリカタイプ (マスタ、セカンダリ、読み込み専用、サブオーディネートリファレンス)
  - レプリカ番号
  - レプリカルート ID

- アドレスの数
- アドレスレコード

- Stream (ストリーム)

  任意のバイナリ情報を表します。ストリーム構文を使用すると、ファイルサーバ上のファイルから OpenText eDirectory 属性を作成できます。この構文は、ログインスクリプトやその他のストリーム属性で使用されます。ストリームファイルに格納されているデータは、いかなる種類の構文強制も受けません。この構文は単に任意のデータであり、個別のアプリケーションで作成、定義および使用されます。

- Telephone Number (電話番号)

  電話番号を値としてとる属性に使用されます。2 つの電話番号は、長さが等しく、対応する文字列が同一 (ただし、比較処理で無視されるスペースとハイフンを除く) の場合に、一致とみなされます。

- 時刻

  符号なし整数を値としてとり、秒単位の時間を表す属性に使用されます。

- Timestamp (タイムスタンプ)

  特定のイベントが発生した時刻を示す値をとる属性に使用されます。特定のイベントが発生すると、OpenText eDirectory サーバによって新しいタイムスタンプ値が生成され、その値がイベントに関連付けられます。各タイムスタンプ値は、OpenText eDirectory パーティション内で固有です。これにより、同一パーティションのレプリカを保持しているすべてのサーバで発生したイベントを 1 つにまとめて順序付けできます。

- Typed Name (タイプ付きの名前)

  オブジェクトに関連付けられたレベルおよび間隔を表す値をとる属性に使用されます。この構文は、OpenText eDirectory オブジェクトを指定し、次の 2 つの数値を当該オブジェクトにアタッチします。

  - 優先番号を示す属性レベル
  - イベント間の秒数、または参照の頻度を表す間隔

- 不明

  スキーマから削除されている属性定義をもつ属性に使用されます。この構文は、バイナリ情報の文字列を表します。

High-Valued 属性 (HVA)

HVAConfig 属性は設定可能な属性です。展開プロセスはシンプルで時間効率に優れています。デフォルトで、HVAConfig は Dir-XMLEntitlementResults および oidpInstanceData を監視します。HVAConfig にカスタム値を使用する場合は、これら 2 つの属性に適切な値を再設定する必要があります。カスタムの HVAConfig 属性は、ldif ファイルを使用して設定できます。HVAConfig では、STREAM を除くすべての属性タイプを監視できます。HVAConfig に指定される属性関連の値はすべて、JSON 形式で指定する必要があります。

HVA の設定で使用される 4 つの必須 JSON キーは以下のとおりです。

1. AttributeName - 監視対象の属性の名前。

2. Type - このキーは、監視対象の属性タイプを参照しますたとえば、1 はカウントベースの属性、2 はサイズベースの属性です。
3. Limit - 上限値の警告 / ログ記録が開始される上限。
4. Interval - このキーは、カウントベースの属性の場合に使用されます。属性が上限を超えると、間隔カウントごとにアラートが表示されます。

HVAconfig 属性は、次に示すいくつかの方法で追加または変更できます。

1 ldif ファイルを使用して、HVAconfig 属性を NCP サーバオブジェクトに追加する。

ldif ファイルのサンプルを次に示します。

```
version:1
# define attributes
dn: cn=servername,o=server
changetype: modify
add: HVAConfig
HVAConfig:
[{"AttributeName":"oidpInstanceData","Type":2,"Limit":1,"Interval":0}]
```

「dn:」は ncp サーバオブジェクト DN です。

ldif ファイルをアップロードするには、ldapmodify を使用します。

ldapmodify -h 10.xx.xx.xx -D cn=admin,o=novell -w password -f hvaPolicy.ldif

2 OpenText Identity Console から HVAConfig 属性を追加する。

1. ［役割とタスク (Roles and Tasks)］から［ディレクトリ管理］に移動し、［Modify Objects(オブジェクトの変更)］>［NCP サーバオブジェクト］の順に選択します。［OK］をクリックします。
2. 変更ページの［**値がない属性 (Unvalued Attributes)**］から［**その他**］タブを選択し、［HVAConfig］を選択します。
3. JSON 形式で制限を設定する属性を入力します。

   例 :[{"AttributeName":"oidpInstanceData","Type":2,"Limit":1,"Interval":0}]
4. ［**保存**］、［**適用**］の順にクリックします。

3 OpenText Identity Console から HVAConfig 属性を追加する。

1. ［**オブジェクト管理**］ページに移動し、［**NCP サーバオブジェクト**］を選択して変更します。
2. ［**値がある属性**］の下の［+］ボタンをクリックします。
3. 属性のリストから、［HVAConfig］を選択します。
4. 値を JSON 形式で入力し、保存します。

   例 :[{"AttributeName":"oidpInstanceData","Type":2,"Limit":1,"Interval":0}]

HVAconfig は次に示すように変更できます。

次のサンプル ldif ファイルは、HVAConfig 属性の変更方法を示しています。

```
Version:1
# define attributes
dn: cn=servername, o=server
changetype:modify
replace:HVAConfig
HVAConfig: [{AttributeName":"DirXML-
EntitlementResult","Type":1,"Limit":5010,"Interval":500}],
{"AttributeNAme":"oidpInstaceData","Type":2,"Limit":16588,"Interval":0}]
```

同様に、1 つの ldif ファイルで複数の属性を追加および変更できます。

サンプルの ldapmodify コマンド :

ldapmodify -h 10.71.128.217 -D cn=admin,o=novell -w n -f hvaPolicy.ldif

HVAConfig で指定された属性が最大値になると、同じ属性に関するログが hvAttr-alert.log の下に記録されます。このログは、通常の $LOG_DIR に作成されます。環境設定が変更されるたびに、Limber を imonitor からトリガして変更を有効にする必要があります。HVAConfig は NCPServer オブジェクトの一部なので、監視対象の属性はすべてサーバ固有になります。

必須属性およびオプション属性について

オブジェクトにはどれも、オブジェクトのタイプに合わせて定義されたスキーマクラスがあります。クラスとは、意味のある方法で組織された属性のグループを指します。これらの属性の一部は必須で、一部はオプションです。

必須属性

必須属性とは、オブジェクトの作成時に指定する必要がある属性です。たとえば、社員番号が必須属性であるユーザクラスで新しいユーザを作成する場合、社員番号を入力せずに新しいユーザオブジェクトを作成することはできません。

オプション属性

オプション属性とは、必要に応じて指定できる属性を指します。たとえば、ニックネームがオプション属性となっているユーザクラスで新しいユーザオブジェクトを作成するとします。この場合、この属性が入力されてもされなくてもユーザオブジェクトは作成できます。属性を入力するかどうかは、新しいユーザにニックネームが付けられているかどうかに左右されます。

例外的にオプション属性が命名に使用される場合がありますが、その場合は属性が必須になります。

スキーマのサンプル

図 1-12 にスキーマの一部の例を示します。これは、基本のスキーマに類似している場合があります。この図は、組織クラスについての情報を表しています。この画面に表示されているほとんどの情報は、クラスが作成されたときに指定されたものです。オプションの属性のいくつかは後に追加されました。

 このアイコンは、ベーススキーマの拡張であるすべてのクラスと属性に割り当てられます。

図 1-12 *OpenText Identity Console の［クラス情報］ページ*



スキーマを設計する

最初にスキーマの設計を行うと、長期的に見た場合、時間と労力を節約できます。ベーススキーマを表示して、それが実際の必要条件に見合うか、あるいは変更が必要かを判断できます。変更が必要な場合は、スキーママネージャを使用してスキーマを拡張します。詳細については、「146 ページの 「スキーマの拡張」」と「150 ページの 「スキーマの表示」」を参照してください。

パーティション

パーティションは、OpenText eDirectory データベースの論理区分です。各ディレクトリパーティションは、ディレクトリ情報を格納するツリー内の個別のデータユニットとなります。

パーティションを分割すると、ディレクトリの一部を 1 つのサーバから切り離し、別のサーバに置くことができます。

WAN リンクが遅い場合や信頼性に乏しい場合、またはディレクトリに多量のオブジェクトがあるためにサーバが処理しきれず、アクセスが遅くなる場合は、ディレクトリをパーティション分割することを検討してください。パーティションの詳細な説明については、159 ページの第 6 章「パーティションおよびレプリカの管理」を参照してください。

各ディレクトリパーティションは、コンテナオブジェクト、それに含まれるすべてのオブジェクト、およびそれらのオブジェクトに関するデータのセットから構成されます。OpenText eDirectory パーティションには、ファイルシステムに関する情報、またはパーティションに含まれるディレクトリやファイルに関する情報は格納されません。

パーティション設定は、OpenText Identity Console で行います。パーティションは、OpenText Identity Console 内の［**パーティション管理**］タイルで、パーティションアイコン ( ) によって識別されます。

OpenText eDirectory のデフォルトのパーティション設定では、ディレクトリ全体が 1 つのパーティションにまとめられます。Identity Console でサーバのレプリカビューを表示すると、そのサーバに保持されているすべてのレプリカが表示されます。詳細については、62 ページの 「レプリカ」および 169 ページの 「OpenText eDirectory サーバのレプリカを表示する」を参照してください。

パーティション

パーティションには、その最上位のコンテナの名前が付けられます。たとえば、Tree と Finance という名前の 2 つのパーティションがあるとします。パーティション Finance は Tree から分割されたため、このパーティションは Tree のチャイルドパーティションと呼ばれます。また Tree は、Finance のペアレントパーティションと呼ばれます。

ディレクトリに大量のオブジェクトがあるためにサーバが処理しきれず、OpenText eDirectory へのアクセスが遅くなる場合、このように新しいパーティションを作成すると便利です。新しいパーティションを作成すると、データベースを分割し、その分岐のオブジェクトを異なるサーバに渡すことができます。

OpenText Identity Console でパーティションのレプリカビューを表示すると、そのパーティションのレプリカを保持しているすべてのサーバが表示されます。この場合、Server1 には、Finance パーティションの読み書き可能レプリカが保持されています。詳細については、171 ページの 「パーティションレプリカを表示する」を参照してください。

パフォーマンス向上のためにレプリカを分散する

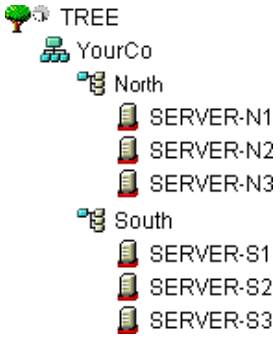
前の例で、Server1 サーバに、Tree パーティションと Finance パーティションの両方のレプリカが保管されているとします。この場合、パーティション分割後も Server1 はディレクトリ全体 (両パーティションのレプリカ) をまだ保持しているため、パーティション分割による OpenText eDirectory のパフォーマンスの向上は実現されません。

パフォーマンスを向上させるには、レプリカの 1 つを別のサーバに移動する必要があります。たとえば、Tree パーティションを Server2 へ移動すると、Tree および YourCo コンテナにあるすべてのオブジェクトが Server2 に移動します。Server1 は、Finance および Accounts コンテナのオブジェクトだけを格納することになります。Server1 と Server2 への負荷は、パーティション分割を行っていないときに比べて、いずれも軽くなります。

パーティションと WAN リンク

ネットワークが、WAN リンクで分割された North サイトと South サイトの 2 つのサイトに渡っている場合を考えてみます。各サイトには、それぞれ 3 つのサーバがあります。

図 1-13 *Opentext eDirectory コンテナの例*



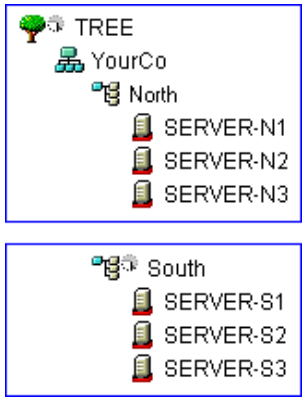
このケースでは、ディレクトリを 2 つのパーティションに分割すると、OpenText eDirectory はより高速になり、信頼性も向上します。

パーティションが 1 つしかないと、レプリカは 1 つのサイトに保存されるか、2 つのサイトに分散されます。この場合、次の 2 つの問題点があります。

- たとえば、North サイトにすべてのレプリカが保存されているとすると、South サイトでは、ログインやリソースへのアクセスに時間がかかります。また、リンクが停止した場合、South サイトのユーザは、ログインまたはリソースにアクセスすることができなくなります。
- レプリカが 2 つのサイトに分散されていると、ユーザはローカルでディレクトリにアクセスできます。ただし、サーバ間のレプリカの同期は WAN リンクを通して行われるので、リンクの信頼性が低い場合には、OpenText eDirectory エラーが発生する可能性があります。ディレクトリに加えられた変更を WAN 経由で伝えるには、時間がかかります。

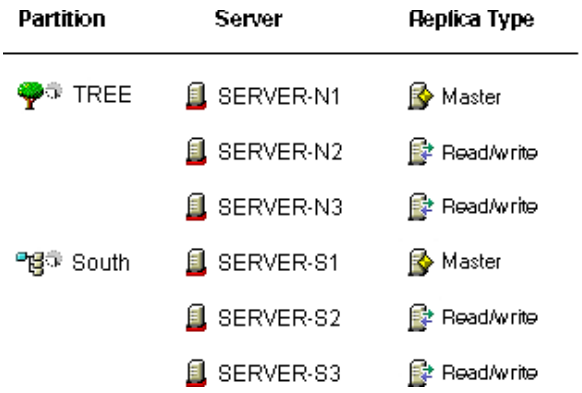
に示した 2 つのパーティションによるソリューションは、WAN リンクでのパフォーマンスと信頼性の問題を解決します。図 1-14

図 1-14 *パーティションの例*



Tree パーティションのレプリカは、North サイトのサーバに保存されています。South パーティションのレプリカは、図 1-15 に示すように、South サイトのサーバに保存されています。

***図 1-15** パーティション、サーバ、およびレプリカの例*



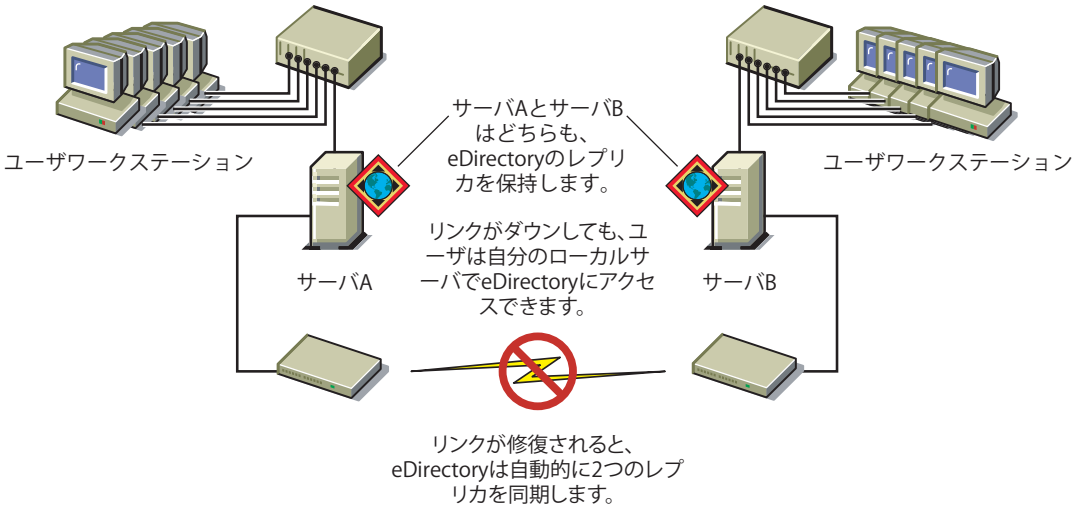
各サイトでは、ローカルリソースを表すオブジェクトは、ローカルで保存されています。サーバ間の同期トラフィックは、通信速度が遅く信頼性の低い WAN リンクではなく、LAN 上でローカルに行われます。

ただし、異なるサイトのオブジェクトにユーザや管理者がアクセスする場合は、WAN リンク上での OpenText eDirectory トラフィックが発生します。

レプリカ

レプリカは、OpenText eDirectory サーバに分配されたユーザ定義済みのパーティションのコピーまたはインスタンスです。ネットワークに複数の OpenText eDirectory サーバがある場合、ディレクトリのレプリカ (コピー) を複数保存しておくことができます。これにより、あるサーバやそのサーバに対するネットワークリンクが機能しなくなった場合でも、ユーザが残りのネットワークリソースにログインして使用できます (図 1-16 を参照)。

***図 1-16** OpenText eDirectory レプリカ*



各サーバには、65,000 を超える OpenText eDirectory レプリカを格納できます。ただし、同じユーザ定義パーティションのレプリカは、同じサーバ上には 1 つしか存在できません。レプリカの詳細な説明については、159 ページの第 6 章「パーティションおよびレプリカの管理」を参照してください。

OpenText eDirectory の耐障害性を高めるため、レプリカを 3 つ保存しておくことをお勧めします (レプリカを保存する OpenText eDirectory サーバが 3 箇所あると想定した場合)。単独のサーバに、複数のパーティションのレプリカを保存することもできます。

レプリカサーバは、OpenText eDirectory レプリカのみを格納する専用サーバです。このタイプのサーバは、DSMASTER サーバとも呼ばれます。この環境設定は、多くの単一サーバリモートオフィスを使用する企業などでよく利用されます。レプリカサーバを使用すると、リモートオフィスの場所のパーティション用として追加のレプリカを格納できます。

また、障害復旧計画にそれを含めることもできます (473 ページの 「DSMASTER サーバによる災害対策」の説明を参照)。

OpenText eDirectory のレプリケーションでは、サーバファイルシステムの耐障害性は提供されません。OpenText eDirectory オブジェクトに関する情報のみが、複製されます。TTS™ (トランザクショントラッキングシステム ™)、ディスクミラーリング / 二重化、RAID、または NRS (NetIQ Replication Services) を使用することにより、ファイルシステムの耐障害性を得ることができます。

バインダリサービスを提供するサーバでは、マスタレプリカ、または読み書き可能レプリカが必要になります。

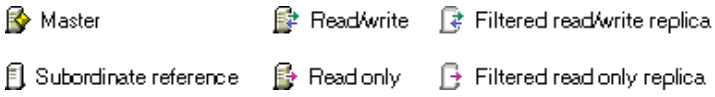
ユーザが WAN リンク経由で OpenText eDirectory 情報に定期的にアクセスしている場合は、必要な情報を含んだレプリカをサーバに配置し、ユーザがローカルでアクセスできるようにすることで、アクセス時間と WAN トラフィックを減らすことができます。

LAN においても、ある程度は同じことが当てはまります。ネットワーク上の複数のサーバにレプリカを分散すると、情報は、通常、最も近くにある使用可能なサーバから取得されます。

レプリカのタイプ

OpenText eDirectory では、次の図に示したレプリカのタイプがサポートされます。

図 1-17 *レプリカのタイプ*



- 64 ページの 「マスタレプリカ」
- 64 ページの 「読み書き可能レプリカ」
- 65 ページの 「読み込み専用レプリカ」
- 65 ページの 「フィルタ済み読み書き可能レプリカ」
- 65 ページの 「フィルタ済み読み込み専用レプリカ」
- 66 ページの 「サブオーディネートリファレンスレプリカ」

マスタレプリカ

 マスタレプリカはオブジェクトやパーティションへの変更を開始するために使用される、書き込み可能なレプリカタイプです。マスタレプリカは、次のタイプの OpenText eDirectory パーティション操作を管理します。

- レプリカのサーバへの追加
- レプリカのサーバからの削除
- OpenText eDirectory ツリーの新しいパーティションの作成
- OpenText eDirectory ツリーの既存のパーティションの削除
- OpenText eDirectory ツリーのパーティションの移動

マスタレプリカは、次のタイプの OpenText eDirectory オブジェクト操作にも使用されます。

- OpenText eDirectory ツリーへの新しいオブジェクトの追加
- OpenText eDirectory ツリーの既存のオブジェクトの削除、リネーム、または移動
- OpenText eDirectory ツリーへのオブジェクトの認証
- OpenText eDirectory ツリーへの新しいオブジェクト属性の追加
- 既存の属性の変更または削除

デフォルトでは、ネットワークの最初の OpenText eDirectory サーバが、マスタレプリカを保持します。マスタレプリカは、各パーティションに同時に 1 つだけ存在します。別のレプリカを作成すると、デフォルトで読み書き可能レプリカになります。

マスタレプリカを保持しているサーバを 1 ～ 2 日以上ダウンさせる場合は、読み書き可能レプリカのうち 1 つをマスタレプリカにすることができます。オリジナルのマスタレプリカは、自動的に読み書き可能レプリカになります。

OpenText eDirectory で新しいレプリカやパーティションの作成などの操作をするには、ネットワークでマスタレプリカが使用可能である必要があります。

読み書き可能レプリカ

 OpenText eDirectory が、読み書き可能レプリカおよびマスタレプリカのオブジェクト情報にアクセスして、変更できます。すべての変更内容は、すべてのレプリカに自動的に伝えられます。

WAN リンクの通信速度が遅い場合やルータがビジー状態の場合など、ネットワークインフラストラクチャでの遅延が原因で OpenText eDirectory のユーザへの反応が遅い場合は、OpenText eDirectory を必要とするユーザの近くに読み書き可能レプリカを作成できます。読み書き可能レプリカは、サーバにいくつでも作成できますが、レプリカが増えると、同期を取るためにトラフィックの量が増加します。

読み込み専用レプリカ

読み込み専用レプリカは、パーティションの境界内にあるすべてのオブジェクトに関する情報を読み込むのに使用される、読み込み可能なレプリカタイプです。読み込み専用レプリカは、マスタレプリカと読み書き可能レプリカからの同期更新は受け入れますが、クライアントからの直接の変更は受け入れません。ログイン更新が有効になっている場合、読み込み専用レプリカへのログインにより属性の更新が発生するため、ログインは失敗します。

このレプリカタイプは、バインダリエミュレーションを提供できませんが、OpenText eDirectory ツリーの耐障害性を提供します。マスタレプリカおよびすべての読み書き可能なレプリカが破壊または破損すると、読み込み専用レプリカを新しいマスタレプリカに昇格させることができます。

また、NDS オブジェクトの読み込みやパーティションの境界内のすべてのオブジェクトを含むフォールトトレランス、およびパーティションルートオブジェクトを含む NDS ディレクトリツリーへの接続も提供します。

クライアントは常に読み書き可能レプリカにアクセスし、変更を加え続けることができるため、ツリー内のセキュリティポリシーを確立してオブジェクトの変更を制限する目的のために読み込み専用レプリカを使用することは避けてください。権利継承フィルタの使用のように、この目的でディレクトリに存在するメカニズムは他にもあります。詳細については、76 ページの 「IRF(権利継承フィルタ)」を参照してください。

フィルタ済み読み書き可能レプリカ

フィルタ済み読み書き可能レプリカには、フィルタ済みオブジェクトセットまたはオブジェクトクラスが、それらのオブジェクトのフィルタ済み属性セットおよび値とともに格納されます。それに含まれる内容は、ホストサーバのレプリケーションフィルタに固有の OpenText eDirectory オブジェクトおよびプロパティのタイプに限定されます。ユーザはレプリカの内容の読み取ったり、変更したりでき、OpenText eDirectory は選択されたオブジェクト情報にアクセスしたり、変更したりできます。選択された変更内容は、すべてのレプリカに自動的に伝えられます。

フィルタ済みレプリカの場合、サーバごとにフィルタを 1 つだけ指定できます。つまり、あるサーバで定義されているフィルタは、そのサーバ上のすべてのフィルタ済みレプリカに適用されます。フィルタ済みレプリカは、サーバにいくつでも作成できますが、レプリカが増えると、同期を取るためのトラフィックの量が増加してしまいます。

詳細については、66 ページの 「フィルタ済みレプリカ」を参照してください。

フィルタ済み読み込み専用レプリカ

フィルタ済み読み込み専用レプリカには、フィルタ済みオブジェクトセットまたはオブジェクトクラスが、それらのオブジェクトのフィルタ済み属性セットおよび値とともに格納されます。フィルタ済み読み込み専用レプリカは、マスタレプリカと読み書き可能レプリカからの同期更新は受け入れますが、クライアントからの直接の変更は受け入れませ

ん。ユーザはこのレプリカの内容を読み込むことができますが、変更はできません。それに含まれる内容は、ホストサーバのレプリケーションフィルタに固有の OpenText eDirectory オブジェクトおよびプロパティのタイプに限定されます。

詳細については、66 ページの 「フィルタ済みレプリカ」を参照してください。

サブオーディネートリファレンスレプリカ

サブオーディネートリファレンスレプリカは、システム生成のレプリカで、マスタレプリカや読み書き可能レプリカのオブジェクトをすべて含んでいるわけではありません。したがって、サブオーディネートリファレンスレプリカは、障害対策を提供していません。サブオーディネートリファレンスレプリカは、OpenText eDirectory によるパーティションの境界を超えた名前の解決に必要な情報のみを格納するために生成される内部ポインタです。

サブオーディネートリファレンス (レプリカ) は削除できません。OpenText eDirectory では、不要になったサブオーディネートリファレンスレプリカは自動的に削除されます。サブオーディネートリファレンスレプリカは、ペアレントパーティションのレプリカを保持し、チャイルドパーティションのレプリカを保持していないサーバでのみ作成されます。

チャイルドパーティションのレプリカが、ペアレントのレプリカを保持するサーバにコピーされた場合、サブオーディネートリファレンスレプリカは自動的に削除されます。

フィルタ済みレプリカ

フィルタ済みレプリカには、オブジェクトまたはオブジェクトクラスのフィルタされたセットが、これらのオブジェクトの属性および値のフィルタされたセットと共に格納されます。たとえば、OpenText eDirectory ツリーのさまざまなパーティション内のユーザオブジェクトのみを格納するフィルタ済みレプリカのセットを、1 つのサーバ上に作成できます。また、ユーザオブジェクトのデータ (名、姓、電話番号など) のサブセットのみを格納するレプリカを作成することもできます。

フィルタ済みレプリカを使用して、1 つのサーバ上に OpenText eDirectory データのビューを構築できます。そのために、フィルタ済みレプリカでは、スコープとフィルタを作成できます。これにより、ツリー内の数多くのパーティションから 1 つの OpenText eDirectory サーバに、適切に定義されたデータセットを格納することが可能になります。

サーバのスコープとデータフィルタの説明は OpenText eDirectory に格納され、OpenText Identity Console のサーバオブジェクトを通して管理できます。

1 つまたは複数のフィルタ済みレプリカを格納するサーバは、レプリケーションフィルタを 1 つだけ保持します。そのため、そのサーバにあるすべてのフィルタ済みレプリカは、それぞれのパーティションからの同じ情報のサブセットを格納します。フィルタ済みレプリカのマスタパーティションレプリカは、OpenText eDirectory 8.5 以降が動作する OpenText eDirectory サーバでホストされていなければなりません。

フィルタ済みレプリカは、次の目的に使用できます。

- 他のサーバから複製する必要があるデータ量が減ることによる、サーバへの同期トラフィックの削減。
- OpenText Identity Manager によってフィルタリングする必要があるイベント数の削減。

OpenText Identity Manager の詳細については、『OpenText™ Identity Manager CE 24.4 (v4.10) Overview and Planning Guide』を参照してください。

- ディレクトリデータベースのサイズの縮小。

  レプリカの数が増えれば、その分データベースのサイズが増大します。完全なレプリカを作成するのではなく、特定のクラスのデータのみを格納するフィルタ済みレプリカを作成することによって、ローカルデータベースのサイズを小さくすることができます。

  たとえば、ツリーに 10,000 のオブジェクトが含まれていても、そのうち［ユーザ］オブジェクトの占める割合がわずかであれば、10,000 すべてのオブジェクトを格納する完全なレプリカではなく、［ユーザ］オブジェクトのみを含むフィルタ済みレプリカを作成できます。

ローカルデータベースに格納されているデータをフィルタリングできるという特徴を別にすれば、フィルタ済みレプリカは通常の OpenText eDirectory レプリカと同じであり、必要なときはいつでも完全なレプリカに戻すことができます。

注 : デフォルトでフィルタ済みのレプリカは、必須のフィルタとして組織および部門を持つことになります。

フィルタ済みレプリカの設定と管理の詳細については、168 ページの 「フィルタ済みレプリカを設定し管理する」を参照してください。

フィルタ済みレプリカへのローカルログインの許可

フィルタ済みレプリカへのローカルログインを許可するには、OpenText Identity Console で［**ローカルログインの有効化**］オプションを選択することに加えて、クラス ndsLoginProperties をフィルタに追加する必要もあります。

フィルタ済みレプリカへログインする前に、次の属性を設定する必要があります。

- 不正侵入者を検出する
- 不正ログイン回数のリセット間隔
- 最終ログイン時刻
- 不正侵入者によりロックされました
- 検出後のロックアウト
- ログイン許容時間マップ
- ログインが無効化されました
- ログインの有効期限
- ログイン猶予制限
- 残りのログイン猶予
- ログイン不正侵入者のアドレス
- ログイン不正侵入者の試行
- ログイン不正侵入者の制限
- ログイン不正侵入者のリセット時間

- 同時ログインの最大数
- ログイン時刻
- ネットワークアドレス
- ネットワークアドレスの制限
- パスワード期限間隔
- パスワード期限の時刻
- 秘密鍵
- 公開鍵
- nspmDoNotExpirePassword
- nspmPasswordKey
- nspmPasswordPolicyDN
- pwdAccountLockedTime
- pwdFailureTime
- sasLoginFailureDelay
- sasOTPCounter
- sasOTPDigits
- sasOTPEnabled
- sasOTPReSync
- sasUpdateLoginInfo
- sasUpdateLoginTimeInterval

注 : 上記の属性は、ユーザオブジェクト、ペアレントコンテナ、またはログインポリシーで設定できます。

レプリカリングでのサーバの同期

複数のサーバが、同じパーティションのレプリカを保持している場合、それらのサーバはレプリカリングとみなされます。同期は、あるレプリカから他のレプリカへのディレクトリ情報の伝播であるため、各パーティションの情報は他のパーティションの情報と整合性を保ちます。OpenText eDirectory は、それらのサーバの同期を自動的に維持します。詳細については、127 ページの 「同期」を参照してください。

OpenText eDirectory 同期のタイプは、次のとおりです。

- 通常同期またはレプリカ同期
- 優先度同期

リソースへのアクセス

OpenText eDirectory は、デフォルトの権利に従って基本レベルのネットワークアクセスセキュリティを構築します。アクセス制御をより強化するための方法を、次に示します。

- 権利の割り当て

  ユーザがネットワークリソースへのアクセスを試みると、システムはそのリソースに対するユーザの有効な権利を計算します。明示的なトラスティ割り当て、同等セキュリティの付与、および継承される権利に対するフィルタ処理を行うことによって、ユーザがリソースに対する適切で有効な権利を持つように設定できます。

  権利の割り当てを単純化するため、グループおよび職種オブジェクトを作成した後に、グループや職種にユーザを割り当てることができます。

- ログインセキュリティの追加

  ログインセキュリティは、デフォルトでは設定されていません。ログインセキュリティ方法の設定にはいくつかあります。ログインパスワード、ログインする場所および時間の制限、同時ログインセッションの制限、不正侵入者検出、およびログインの禁止があります。

- ロールベース管理の設定

  特定のオブジェクトプロパティに対して管理者を設定し、これらのプロパティに対してのみ、権利を許可します。これにより、特定の責任を担う管理者を作成できます。この責任はどのコンテナオブジェクトの下位のオブジェクトへも継承できます。ロールベースの管理者は、従業員の情報やパスワードに関連したプロパティのような、特定のプロパティに対して責任を持ちます。

  RAC 設定の設定方法については、『*OpenText™ Identity Console 管理ガイド* (https://www.netiq.com/documentation/identity-console/identity_console-admin/data/bookinfo.html)』の RAC 設定に関するセクションを参照してください。

  管理者がロールベースの管理アプリケーションを実行できるよう、特定のタスクに基づいて役割を定義することもできます。詳細については、役割とアクセス制御の設定を参照してください。

OpenText eDirectory での権利

ツリーを作成すると、デフォルトの権利割り当てによって、ネットワークへの汎用アクセスとセキュリティが与えられます。デフォルトの割り当てには、次のようなものがあります。

- ユーザ Admin はツリーの最上位に対してスーパバイザ権を持ちます。これにより、Admin はディレクトリ全体を完全に制御できます。Admin は、サーバオブジェクトに対してもスーパバイザ権を持ち、サーバ上のどのボリュームに対しても完全に制御できます。
- ［Public］はツリーの最上位のブラウズ権を持ちます。これにより、すべてのユーザが、パブリックアクセスを通して、ツリー内のすべてのオブジェクトを表示する権利を持つことになります。
- アップグレードプロセス、印刷アップグレード、Windows ユーザ移行を通して作成されたオブジェクトは、ほとんどの状況で適切なトラスティ割り当てを受け取ります。

トラスティ割り当ておよびターゲット

権利の割り当てには、トラスティとターゲットオブジェクトが関係します。トラスティとは、権限を付与される単一のユーザまたはユーザの集合です。ターゲットとは、ユーザが権利を持っているネットワークリソースです。

- 別名をトラスティにする場合、権利は別名が示すオブジェクトにのみ適用されます。ただし、別名オブジェクトを明示的にターゲットにすることは可能です。
- ファイルシステム権は OpenText eDirectory 内ではなく、ファイルシステム自体に保存されていますが、ファイルシステム内のファイルまたはディレクトリも、ターゲットにすることができます。

注 : [Public] トラスティは、オブジェクトではありません。すべてのネットワークユーザに対して与えられる、権利割り当てに関連するトラスティです。

[This] は特殊な種類のトラスティです。これは、その名前がアクセス対象のエントリと一致する場合に、認証されたオブジェクトになるよう定義されています。これにより管理者は、[This] をトラスティとして、ツリー最上位で 1 つの ACL を使用することで、簡単に権利 (「各ユーザが自分の電話番号を管理する」など) を指定できます。

OpenText eDirectory の権利の概念

OpenText eDirectory での権利についてより良く理解していただくために、次に各権利の概念について説明します。

- 70 ページの 「オブジェクト (エントリ) 権」
- 71 ページの 「プロパティ権」
- 72 ページの 「有効な権利」
- 72 ページの 「有効な権利を計算する方法」
- 75 ページの 「同等セキュリティ」
- 75 ページの 「ACL(アクセス制御リスト)」
- 76 ページの 「IRF(権利継承フィルタ)」

オブジェクト (エントリ) 権

トラスティ割り当てを作成するときには、オブジェクト権とプロパティ権を与えることができます。オブジェクト権はオブジェクト全体の操作に適用されますが、プロパティ権は一定のオブジェクトプロパティにのみ適用されます。オブジェクト権は、OpenText eDirectory データベース内でエントリが提供されるため、エントリ権と呼ばれます。

各権利について、次に説明します。

- **スーパバイザ** には、オブジェクトおよびそのオブジェクトのすべてのプロパティに対する、すべての権利が含まれます。
- **参照** とは、ツリー内のオブジェクトを参照する権利です。オブジェクトのプロパティを参照する権利は含まれていません。

- **作成** は、ターゲットオブジェクトがコンテナの場合にのみ適用されます。トラスティがコンテナの下に新規オブジェクトを作成する権利のことで、ブラウズ権も含まれます。
- **削除** とは、ディレクトリからターゲットを削除する権利です。
- **リネーム** とは、ターゲット名を変更する権利です。

プロパティ権

トラスティ割り当てを作成するときには、オブジェクト権とプロパティ権を与えることができます。オブジェクト権はオブジェクト全体の操作に適用されますが、プロパティ権は一定のオブジェクトプロパティにのみ適用されます。

OpenText Identity Console では、プロパティ権の管理用に次の 2 つのオプションが用意されています。

- ［**All Attributes Rights**］項目が選択されている場合は、すべてのプロパティを同時に管理できます。
- 特定のプロパティが選択されている場合は、選択された個々のプロパティを管理できます。

重要 : あるユーザの［All Attributes Rights］プロパティへの読み取りアクセスをトラスティに付与した場合、そのトラスティには、そのユーザの［Password Management］属性への読み取りアクセスが与えられます。トラスティは、パスワード関連の情報、パスワードの有効期限、猶予ログイン制限などのユーザ情報を読み込むことができます。

パスワードポリシーの作成と管理の詳細については、803 ページの 「パスワードポリシーの作成」を参照してください。

各権利について、次に説明します。

- **スーパバイザ** トラスティにプロパティの完全な制御権を与えます。
- **比較** トラスティは、プロパティの値と任意の値を比較できます。この権利によって検索が可能になります。検索では、TRUE または FALSE の結果のみ返されます。トラスティは、プロパティの値を実際に参照することはできません。
- **読み込み** トラスティはプロパティの値を参照することができます。この権利には、比較権が含まれます。
- **書き込み** トラスティはプロパティの値を作成、変更、および削除できます。
- **自己追加** トラスティ自身をプロパティの値として追加または削除する権利です。この権利は、メンバーシップリストまたは ACL(アクセス制御リスト) などのような、値としてオブジェクト名を持つプロパティに適用されます。

有効な権利

ユーザは、明示的なトラスティ割り当て、継承、および同等セキュリティなどのさまざまな方法で権利を受け取ることができます。権利は、権利継承フィルタにより制限され、下位レベルのトラスティ割り当てによって変更、または取り消すこともできます。これらのすべてのアクションによる最終結果、つまりユーザが行使できる権利は、*有効な権利*と呼ばれます。

ユーザがいずれかの操作を実行しようとするたびに、該当のオブジェクトに対するそのユーザの有効な権利が計算されます。

有効な権利を計算する方法

ユーザがネットワークリソースにアクセスを試みるたびに、OpenText eDirectory は次の手順でそのターゲットリソースに対するユーザの有効な権利を計算します。

1 OpenText eDirectory によって、計算で考慮される権利を持ったトラスティがリストされます。考慮されるトラスティには次のものがあります。

- ターゲットリソースにアクセスしようとしているユーザ
- ユーザが同等セキュリティとなっているオブジェクト

2 リスト内の各トラスティに対して、OpenText eDirectory は次のように有効な権利を決定します。

2a まず、ツリーの最上位に対してトラスティが継承可能な権利をチェックします。

OpenText eDirectory は Tree オブジェクトのオブジェクトトラスティ (ACL) プロパティをチェックして、該当のトラスティが登録された項目があるかどうか調べます。該当のトラスティが登録された項目が見つかり、その権利が継承可能な場合、OpenText eDirectory はそれらの項目に指定された権利を、該当のトラスティの有効な権利の初期セットとして使用します。

2b ツリー内の、ターゲットリソースが含まれているブランチを 1 レベル下に移動します。

2c このレベルでフィルタリングされるすべての権利を削除します。

OpenText eDirectory はこのレベルの ACL をチェックし、該当のトラスティの有効な権利のタイプ (オブジェクト、すべてのプロパティ、または特定のプロパティ) と一致する IRF(権利継承フィルタ) がないか調べます。検出された場合は、OpenText eDirectory は該当のトラスティの有効な権利から、これらの IRF によって継承を阻止されるすべての権利を削除します。

たとえば、上位のレベルで、トラスティの有効な権利にすべてのプロパティに対する書き込み権の割り当てが含まれていても、このレベルの IRF によってその継承を阻止された場合、すべてのプロパティに対する書き込み権はトラスティの有効な権利から削除されます。

2d このレベルで割り当てられた継承可能な権利があれば追加し、必要に応じてオーバーライドします。

OpenText eDirectory はこのレベルの ACL をチェックし、該当のトラスティが登録された項目があるかどうかチェックします。該当のトラスティが登録された項目が見つかり、その権利が継承可能な場合、それらの項目の権利がトラスティの有効な権利にコピーされ、必要に応じて他の割り当てが無効化されます。

たとえば、上位のレベルで、トラスティの有効な権利に作成および削除オブジェクト権が含まれ、プロパティ権についてはまったく含まれていないときに、このレベルの ACL に、該当のトラスティに対する 0 オブジェクト権の割り当てとすべてのプロパティの書き込み権の割り当てが含まれている場合、トラスティの既存のオブジェクト権 (作成および削除) は 0 権利と置き換えられ、新たにすべてのプロパティ権が追加されます。

2e OpenText eDirectory は、ツリーの各レベル (ターゲットリソースのレベルを含む) でフィルタリングと追加の手順 (上記のステップ 2c およびステップ 2d) を繰り返します。

2f ターゲットリソースで割り当てられた継承不可能な権利があれば追加し、必要に応じてオーバーライドします。

OpenText eDirectory は、上記のステップ 2d と同じ処理を行います。その結果作成される権利セットが、該当のトラスティの有効な権利になります。

3 リスト内のすべてのトラスティの有効な権利を次のように結合します。

3a OpenText eDirectory はリスト内のいずれかのトラスティが所有している権利をすべて含めます。リスト内のどのトラスティも所有していない権利のみ除外します。権利タイプは混在させません。たとえば、特定のプロパティに対する権利を、すべてのプロパティに対する権利に追加したり、その逆を行うことはありません。

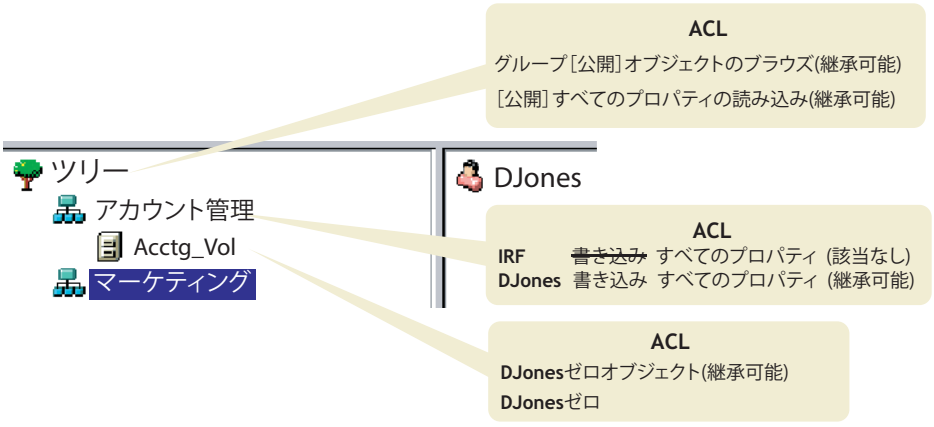
3b 現在有効な権利に暗黙で含まれる権利を追加します。

権利の設定により、ターゲットリソースに対するユーザの有効な権利が作られます。

例

ユーザ「DJones」が、ボリューム「Acctg_Vol」にアクセスしようとしています。図 1-18 を参照してください。

図 1-18 トラスティ権の例



次の手順は、OpenText eDirectory が Acctg_Vol に対する DJones の有効な権利を計算する方法を示したものです。

1. 計算で考慮される権利を持ったトラスティは、DJones、Marketing、Tree、および［Public］です。

ここでは、DJones がどのグループまたは役割にも所属せず、どの同等セキュリティにも明示的に割り当てられていないと仮定しています。

2. 各トラスティの有効な権利は、次のとおりです。
   - DJones: 0 オブジェクト、0 すべてのプロパティ

     Acctg Vol で 0 すべてのプロパティ権を割り当てられることで、Accounting でのすべてのプロパティに対する書き込み権の割り当ては無効化されます。
   - Marketing: 0 すべてのプロパティ

     ツリーの最上位にあるすべてのプロパティに対する書き込み権の割り当ては、Accounting での IRF によって除外されます。
   - Tree: 権利なし

     Tree には、関連するどのツリー分岐でも、権利はまったく割り当てられません。
   - [パブリック]: オブジェクトのブラウズ権、すべてのプロパティの読み込み権

     これらの権利はルートで割り当てられ、関連するツリー分岐のどの位置でも、フィルタリングも無効化もされません。
3. これらすべてのトラスティの権利を結合することによって、次のようになります。

   DJones: オブジェクトのブラウズ権、すべてのプロパティの読み込み権
4. すべてのプロパティの読み込み権に伴って暗黙で割り当てられるすべてのプロパティの比較権を追加することで、最終的に Acctg_Vol に対する DJones の有効な権利は次のようになります。

   DJones: オブジェクトのブラウズ権、すべてのプロパティの読み込み、比較権

有効な権利のブロック

有効な権利の計算方法では、IRF に頼らずに特定の権利を特定のユーザに対してブロックする方法は、必ずしも明確ではありません (IRF はすべてのユーザの権利をブロックします)。

特定の権利を IRF に頼らずにユーザに対してブロックするには、次のような方法があります。

- ターゲットリソース、およびツリー内のターゲットリソースよりも上位のレベルで、該当のユーザおよびそのユーザと同等セキュリティになるオブジェクトにそれらの権利を割り当てないようにする。
- 該当のユーザまたはそのユーザが同等セキュリティになるいずれかのオブジェクトにそれらの権利が実際に割り当てられている場合は、ツリーの下位レベルに、それらの権利の割り当てを阻止する何らかの割り当てをそのオブジェクトに対して与えるようにする。不適切な権利を持つすべての (ユーザと関連する) トラスティに対して、これを実行します。

同等セキュリティ

同等セキュリティとは、別のオブジェクトと同じ権利を持つことを意味します。あるオブジェクト (オブジェクト A) を別のオブジェクト (オブジェクト B) に対して同等セキュリティに設定すると、オブジェクト A の有効な権利の計算時にオブジェクト B の権利がオブジェクト A に追加されます。

たとえば、ユーザオブジェクト Joe を Admin オブジェクトと同等セキュリティにするとします。同等セキュリティを割り当てた後は、Joe は、ツリーおよびファイルシステムに対して、Admin が持つ権利と同じ権利を持つことになります。

同等セキュリティには、次の 3 つのタイプがあります。

- 明示的 : 割り当てによる
- 自動的 : グループまたは役割のメンバーシップによる
- 暗黙的 : すべてのペアレントコンテナおよび［パブリック］トラスティと同等

同等セキュリティは、1 ステップにかぎり有効です。たとえば、さらに別のユーザを上の例の Joe と同等セキュリティにした場合、このユーザは Joe の権利は受け取りますが、Admin の権利は受け取りません。

同等セキュリティは、該当のユーザオブジェクトの同等セキュリティプロパティの値として OpenText eDirectory に記録されます。

ユーザオブジェクトを職種オブジェクトにその職種の担当者として追加すると、そのユーザは自動的にその職種オブジェクトと同等セキュリティになります。ユーザをグループオブジェクトに追加した場合も同様です。

ACL(アクセス制御リスト)

ACL(アクセス制御リスト) は、オブジェクトトラスティプロパティとも呼ばれます。トラスティを割り当てると、そのトラスティは値としてターゲットのオブジェクトトラスティ (ACL) プロパティに追加されます。

このプロパティは、次の理由から、ネットワークのセキュリティに大きく影響します。

- オブジェクトのオブジェクトトラスティ (ACL) プロパティに対する書き込み権またはスーパバイザ権を持つユーザは、そのオブジェクトのトラスティが誰であるかを知ることができる。
- オブジェクトのオブジェクトトラスティ (ACL) プロパティに対して自己追加権を持つユーザは、そのオブジェクトに対する自分の権利を変更できる。たとえば、そのユーザは自分にスーパバイザ権を与えることができます。

このため、コンテナオブジェクトのすべてのプロパティに対して自己追加権を与える場合は、慎重に行う必要があります。自己追加権を割り当てられたトラスティは、該当のコンテナ、その中のすべてのオブジェクト、およびその下のコンテナ内のすべてのオブジェクトに対してスーパバイザとなることができます。

IRF(権利継承フィルタ)

権利継承フィルタにより、OpenText eDirectory ツリーの下位レベルへの権利の継承をブロックできます。このフィルタの設定の詳細については、82 ページの 「OpenText eDirectory オブジェクトまたはプロパティへの権利継承をブロックする」を参照してください。

新規サーバのデフォルト権

新規サーバオブジェクトをツリーにインストールすると、次のトラスティ割り当てが作成されます。

デフォルトトラスティ	デフォルトの権利
Admin (ツリー内の最初の OpenText eDirectory サーバ)	Tree オブジェクトに対するスーパバイザオブジェクト権。 Admin は、サーバオブジェクトに対してスーパバイザオブジェクト権を持ちます。これは、Admin がサーバ上に存在するあらゆるボリュームのファイルシステムのルートディレクトリに対してもスーパバイザ権を持つことを意味します。
[Public] (ツリー内の最初の OpenText eDirectory サーバ)	Tree オブジェクトに対するブラウズオブジェクト権。
ツリー	すべてのボリュームオブジェクトのホストサーバ名プロパティおよびホストリソースプロパティに対する Tree 読み込みプロパティ権。 これにより、すべてのオブジェクトが物理ボリューム名および物理サーバ名にアクセスできるようになります。
コンテナオブジェクト	sys:\public フォルダに対する読み込み権およびファイルスキャン権。これにより、コンテナの下のユーザオブジェクトは、\public 内のユーティリティにアクセスできるようになります。 **注 :** これらの権利は、OES Linux を実行しているサーバにのみ適用されます。
ユーザオブジェクト	ユーザ用にホームディレクトリが自動的に作成されると、ユーザにはそのディレクトリに対するスーパーバイザ権が与えられます。

管理の委託

OpenText eDirectory では、自分が管理するツリーのブランチに対する管理権を無効にして、そのブランチの管理を他の人物に委託できます。たとえば、特別なセキュリティ要件により、ツリーの分岐を完全に制御する管理者を個別に置かなければならないような場合には、管理権の委託というこの方法をとることが必要になります。

管理権を委託するには、次を実行します。

1 委託先のユーザに、該当のコンテナに対するスーパーバイザオブジェクト権を与えます。

   1a OpenText Identity Console のホームページで、[**権利管理**] をクリックします。

   1b [**トラスティの変更**] をクリックします。

   1c [オブジェクトの検索] () をクリックし、アクセスを制御するコンテナオブジェクトのコンテキストを入力して、[**OK**] をクリックします。

   1d [**Assigned Rights**] をクリックします。

   1e 該当するプロパティの [**スーパバイザ**] チェックボックスをオンにします。

   1f [**完了**] をクリックし、[**適用**] をクリックします。

2 継承を阻止したいスーパバイザ権や他の権利をフィルタリングするための IRF を、そのコンテナに対して作成します。

   2a OpenText Identity Console のホームページで、[**権利管理**] をクリックします。

   2b [**権利継承フィルタ**] をクリックします。

   2c 権利継承フィルタを変更する対象のオブジェクトの名前とコンテキストを指定して、[**OK**] をクリックします。

   2d 必要に応じて権利継承フィルタのリストを編集します。

   フィルタのリストを編集するには、オブジェクトの ACL プロパティに対するスーパバイザ権またはアクセス制御権を持っている必要があります。オブジェクトの継承された権利を全体的にブロックするフィルタは、オブジェクトのすべてのプロパティおよび個々のプロパティに対して設定できます。

   注 : このオブジェクトのトラスティに明示的に付与された権利は継承されないため、これらのフィルタでそのような権利を阻止することはできません。

   2e **OK** をクリックします。

重要 : ユーザオブジェクトに管理を委託した後に、そのオブジェクトが削除されると、その分岐を管理する権利を持つオブジェクトはなくなります。

パスワード管理など、特定の OpenText eDirectory プロパティの管理を委任するには、79 ページの 「同等セキュリティを付与する」を参照してください。

ロールベース管理アプリケーションの特定の機能の使用を委託するには、119 ページの「役割とアクセス制御の設定」を参照してください。

権利の管理

- 78 ページの 「権利を明示的に割り当てる」
- 79 ページの 「同等セキュリティを付与する」
- 82 ページの 「OpenText eDirectory オブジェクトまたはプロパティへの権利継承をブロックする」
- 82 ページの 「OpenText eDirectory オブジェクトまたはプロパティへの有効な権利を参照する」

権利を明示的に割り当てる

OpenText eDirectory ツリーでデフォルトで割り当てられた権利によって、ユーザが必要以上にリソースにアクセスできたり、アクセスが不十分であったりする場合は、権利を明示的に作成して割り当てたり、それを変更したりできます。権利の割り当てを作成または変更するには、まず最初にアクセスを制御しているリソースやトラスティ (権利を所有している、またはこれから所有する OpenText eDirectory オブジェクト) を選択します。

ヒント : ユーザの権利を個別にではなく、まとめて管理するには、グループ、役割、またはコンテナオブジェクトをトラスティにします。すべてのユーザについてリソースへのアクセスを全体的に制限するには、82 ページの 「OpenText eDirectory オブジェクトまたはプロパティへの権利継承をブロックする」を参照してください。

- 78 ページの 「リソースに基づいて OpenText eDirectory へのアクセスを制御する」
- 79 ページの 「トラスティに基づいて OpenText eDirectory へのアクセスを制御する」

リソースに基づいて OpenText eDirectory へのアクセスを制御する

1. OpenText Identity Console のホームページで、[**権利管理**] をクリックします。
2. [**トラスティの変更**] をクリックします。
3. 制御するアクセス権の対象となる OpenText eDirectory リソース (オブジェクト) の名前とコンテキストを指定して、[**OK**] をクリックします。

   コンテナの下のすべてのオブジェクトへのアクセスを制御するには、そのコンテナを選択します。
4. トラスティのリストおよび権利の割り当てを必要に応じて編集します。
   - **4a** トラスティの権利の割り当てを変更するには、トラスティを選択し、[**割り当てられた権利**] をクリックし、必要に応じて権利の割り当てを変更して、[**完了**] をクリックします。
   - **4b** オブジェクトをトラスティとして追加するには、[**トラスティの追加**] をクリックし、オブジェクトを選択し、[**OK**] をクリックします。次に、[**割り当てられた権利**] をクリックしてトラスティの権利を割り当て、[**完了**] をクリックします。

     権利の割り当てを作成または変更する場合、オブジェクト全体に対しても、オブジェクトのすべてのプロパティまたは個々のプロパティに対しても、アクセスを付与したり拒否したりすることができます。

4c トラスティとなっているオブジェクトを削除するには、そのトラスティを選択し、［**トラスティの削除**］をクリックします。

削除されたトラスティには、オブジェクトやプロパティに対する明示的な権利はすでにありませんが、継承や同等セキュリティによる有効な権利はまだ存在する可能性があります。

5 **OK** をクリックします。

トラスティに基づいて OpenText eDirectory へのアクセスを制御する

1 Identity Console のホームページで、［**権利管理**］タイル、［**トラスティ**］の順にクリックします。

2 ［**権利**］>［**オブジェクト名**］をクリックしてコンテキストを選択します。

3 権利を変更するトラスティ (権利を所有している、またはこれから所有するオブジェクト) の名前やコンテキストを入力します。

4 ［**検索範囲のコンテキスト**］フィールドで、トラスティが権利割り当てを持っている OpenText eDirectory オブジェクトを検索する際の検索範囲となる OpenText eDirectory ツリーの部分を指定します。

5 ［**OK**］をクリックします。

検索結果を示す画面が表示されます。

6 必要に応じてトラスティの OpenText eDirectory 権利割り当てを編集します。

6a 権利割り当てを追加するには、［**割り当てられた権利**］をクリックし、トラスティの権利を割り当てて、［**完了**］をクリックします。

6b 権利割り当てを変更するには、制御するアクセス権の対象となるオブジェクトを選択し、［**割り当てられた権利**］をクリックし、必要に応じてトラスティの権利の割り当てを変更してから、［**完了**］をクリックします。

権利の割り当てを作成または変更する場合、オブジェクト全体に対しても、オブジェクトのすべてのプロパティまたは個々のプロパティに対しても、アクセスを付与したり拒否したりすることができます。

6c 権利割り当てを削除するには、制御するアクセス権の対象となるオブジェクトを選択して、［**プロパティの削除**］をクリックします。

トラスティには、オブジェクトやプロパティに対する明示的な権利はすでにありませんが、継承や同等セキュリティによる有効な権利はまだ存在する可能性があります。

7 **適用**をクリックします。

同等セキュリティを付与する

別の OpenText eDirectory オブジェクトに対して同等セキュリティとなっているユーザは、事実上そのオブジェクトのすべての権利を持っています。ユーザは自動的に、所属するグループや役割に対して同等セキュリティになります。すべてのユーザは、［Public］トラスティ、および Tree オブジェクトなど、OpenText eDirectory ツリーのユーザオブジェクトの上にある個々のコンテナに対して、暗黙で同等セキュリティとなります。また、任意の OpenText eDirectory オブジェクトに対して同等セキュリティを明示的に付与することもできます。

注 : このセクションのタスクを実行すると、OpenText eDirectory 権利を通じて管理権限を委任することができます。ロールベースサービス (RBS) 役割を使用する管理アプリケーションがある場合、それらの役割にユーザメンバーシップを割り当てることで管理権限を委託することもできます。

- 80 ページの 「メンバーシップに基づいて同等セキュリティを付与する」
- 80 ページの 「明示的に同等セキュリティを付与する」
- 81 ページの「オブジェクト固有の OpenText eDirectory プロパティの管理者を設定する」

メンバーシップに基づいて同等セキュリティを付与する

1 同等セキュリティにするユーザのグループオブジェクトか役割オブジェクトを作成します (まだ作成していない場合)。

詳細については、110 ページの 「オブジェクトを作成する」を参照してください。

2 グループや役割に、ユーザに必要な OpenText eDirectory 権利を与えます。

詳細については、78 ページの 「権利を明示的に割り当てる」を参照してください。

3 グループや役割のメンバーシップを編集して、グループや役割の権利を必要とするユーザを追加します。

- グループオブジェクトの場合、[**グループの変更**] ウィンドウを使用します。

  OpenText Identity Console のホームページで、[**グループの管理 (Groups Management)**] タイルをクリックし、[オブジェクト] を選択します。[**オブジェクトの変更**] アイコンをクリックして、グループに追加するメンバーを指定し、[**保存**] をクリックします。

- 役割オブジェクトの場合、[**オブジェクトの変更**] ウィンドウを使用します。

  OpenText Identity Console のホームページで、[**オブジェクト管理**] をクリックし、[**タイプ**] ドロップダウンメニューから [組織の役割 (Organizational Role)] オブジェクトを選択し、[**検索**] をクリックします。[オブジェクト] を選択し、[**値がある属性**] をクリックして任意の属性を選択し、[**OK**] をクリックします。[**オブジェクトの変更**] ウィンドウで、役割に追加する選択した属性の詳細を指定し、[**保存**] をクリックします。

4 **OK** をクリックします。

明示的に同等セキュリティを付与する

1 OpenText Identity Console のホームページで、[**オブジェクト管理**] をクリックし、[**タイプ**] ドロップダウンメニューで、[ユーザ] を選択して、[**検索**] をクリックします。

2 検索したユーザをクリックし、[**セキュリティ**] タブをクリックします。

3 [セキュリティ] タブで、次のように同等セキュリティを付与します。

- ユーザを選択した場合は、[同等セキュリティ] をクリックし、そのユーザとセキュリティを同等にするオブジェクトの名前とコンテキストを選択またはブラウズして、[OK] をクリックします。
- ユーザを同等セキュリティにする対象のオブジェクトを選択した場合は、[同等セキュリティ保有者] をクリックし、そのオブジェクトとセキュリティの面で同等にするユーザの名前とコンテキストを選択またはブラウズして、[OK] をクリックします。

これらの 2 つのプロパティページのコンテキストは、システムによって同期されます。

4 [保存] をクリックします。

5 [OK] をクリックします。

オブジェクト固有の OpenText eDirectory プロパティの管理者を設定する

1 オブジェクト固有のプロパティのトラスティにするユーザ、グループ、役割、またはコンテナのオブジェクトを作成します (まだ作成していない場合)。

トラスティとしてコンテナを作成する場合、コンテナ内またはその下のすべてのオブジェクトに権利が付与されます。プロパティは継承可能なものにする必要があります。そうしないと、コンテナおよびそのメンバーは下位レベルへの権利を持たなくなります。

詳細については、110 ページの 「オブジェクトを作成する」を参照してください。

2 OpenText Identity Console のホームページで、[権利管理] をクリックします。

3 [トラスティの変更] をクリックします。

4 [検索 ] をクリックして、管理者に管理してもらう最高レベルのコンテナの名前とコンテキストを選択し、[OK] をクリックします。

5 [トラスティの変更] ページで、[トラスティの追加] をクリックし、管理者を表すオブジェクトを選択して [OK] をクリックします。

6 追加したトラスティの [割り当てられた権利] をクリックし、[プロパティの追加] をクリックします。

7 プロパティリストに追加するプロパティを選択して、[OK] をクリックします。

8 管理者が管理するそれぞれのプロパティについて、必要な権利を割り当てます。

それぞれの権利割り当ての [継承可能] チェックボックスを必ずオンにしてください。

9 [完了] をクリックし、[適用] をクリックします。

OpenText eDirectory オブジェクトまたはプロパティへの権利継承をブロックする

OpenText eDirectory では、コンテナでの権利の割り当てが継承可能である場合も継承不能である場合もあります。のファイルシステムでは、フォルダ上のすべての権利の割り当ては継承可能です。OpenText eDirectory では、トラスティが誰であるかに関係なく、個別の従属項目で権利が有効にならないように、それらの項目での権利の継承をブロックできます。

1 OpenText Identity Console のホームページで、[**権利管理**] をクリックします。

2 [**権利継承フィルタの変更**] をクリックします。

3 [**検索** ] をクリックし、権利継承フィルタを変更する対象のオブジェクトの名前とコンテキストを選択して、[**OK**] をクリックします。

これで、すでにオブジェクトに設定された権利継承フィルタのリストが表示されます。

4 プロパティページで、必要に応じて権利継承フィルタのリストを編集します。

フィルタのリストを編集するには、オブジェクトの ACL プロパティに対するスーパバイザ権またはアクセス制御権を持っている必要があります。オブジェクトの継承された権利を全体的にブロックするフィルタは、オブジェクトのすべてのプロパティおよび個々のプロパティに対して設定できます。

注 : このオブジェクトのトラスティに明示的に付与された権利は継承されないため、これらのフィルタでそのような権利を阻止することはできません。

---

5 **適用**をクリックします。

OpenText eDirectory オブジェクトまたはプロパティへの有効な権利を参照する

有効な権利は、ユーザが特定のネットワークリソース上で実行できる実際の権利です。有効な権利は、明示的な権利の割り当て、継承、および同等セキュリティを基に、OpenText eDirectory によって計算されます。システムにクエリを設定すると、リソースへのユーザの有効な権利が決定されます。

1 OpenText Identity Console のホームページで、[**権利管理**] をクリックします。

2 [**有効な権利**] をクリックします。

3 [**検索** ] をクリックし、権利継承フィルタを変更する対象のオブジェクトの名前とコンテキストを選択して、[**OK**] をクリックします。

4 次のオプションから選択します。

オプション	説明
プロパティ名	トラスティが有効な権利を持っているプロパティを表示します。プロパティは OpenText eDirectory から読み込まれるため、通常英語で表示されます。リスト内の各項目は次のタイプのいずれかです。 ［すべての属性権］- オブジェクトのすべてのプロパティを表します。 ［エントリ権］- オブジェクト全体を表します。スーパバイザの場合を除き、このアイテムへの権利にはプロパティの権利は含まれません。 特定のプロパティ - トラスティが個々に権利を持つ特定のプロパティです。デフォルトでは、このオブジェクトクラスのプロパティのみが表示されます (次を参照)。
有効な権利	OpenText eDirectory で計算されたとおりに、選択されたプロパティへのトラスティの有効な権利を表示します。
スキーマ内のすべてのプロパティを表示する	このチェックボックスをオフにしておくと、このオブジェクトクラスのプロパティのみが表示されます。 OpenText eDirectory スキーマで定義されたすべてのクラスのプロパティを表示するには、このチェックボックスをオンにします。追加のプロパティが有効になるのは、このオブジェクトがコンテナである場合、またはそれが拡張されて補助クラスのプロパティを含む場合のみです。追加のプロパティの横には行頭文字は表示されません。

5 **完了**をクリックします。

2 OpenText eDirectory ネットワークの設計

OpenText eDirectory の設計は、実質的に、ネットワークのすべてのユーザとリソースに影響を与えます。OpenText eDirectory の設計が良ければ、ネットワークの効率性、耐障害性、安全性、拡張性、機能性が高まり、それによってネットワーク全体のパフォーマンスと価値が向上します。この章では、OpenText eDirectory ネットワーク設計上のヒントについて説明します。

- 85 ページの 「OpenText eDirectory の設計の基本」
- 86 ページの 「OpenText eDirectory ツリーの設計」
- 93 ページの 「ツリーのパーティション化のガイドライン」
- 95 ページの 「ツリーのレプリカ作成に関するガイドライン」
- 98 ページの 「ユーザ環境についてのプランニング」
- 99 ページの 「e ビジネスに対応する OpenText eDirectory の設計」
- 100 ページの 「OpenText Certificate Server について」
- 105 ページの 「ネットワーク時刻の同期」

OpenText eDirectory の設計の基本

効率のよい OpenText eDirectory 設計の基盤となるのは、ネットワークのレイアウト、会社の組織構造、および適切な準備です。

e ビジネス用に OpenText eDirectory を設計する場合は、99 ページの 「e ビジネスに対応する OpenText eDirectory の設計」を参照してください。

ネットワークのレイアウト

ネットワークのレイアウトとは、ネットワークの物理的な設定のことです。効率的な OpenText eDirectory 設計を作成するには、次の項目について考慮する必要があります。

- WAN リンク
- リモートアクセスを必要とするユーザ
- ネットワークリソース (サーバ数など)
- 頻繁な停電など、ネットワークの状態
- 予測されるネットワークレイアウトの変更

組織の構造

会社の組織構造は、OpenText eDirectory の設計に影響します。効率的な OpenText eDirectory 設計を作成するには、以下が必要になります。

- 組織図および運営形態についての理解
- OpenText eDirectory ツリーの設計および実装を完了するのに必要な技能のある担当者。

  次の技能を持った担当者を選出する必要があります。
  - OpenText eDirectory 設計の焦点およびスケジュールの管理
  - OpenText eDirectory の設計、設計標準、およびセキュリティについての理解
  - 物理的なネットワーク構造の理解および管理
  - インターネットワークのバックボーン、テレコミュニケーション、WAN 設計、およびルータ配置の管理

OpenText eDirectory 設計の準備をする

OpenText eDirectory の設計に実際に着手する前に、次を実行します。

- スコープおよびスケジュールの現実的な見積もりの設定
- OpenText eDirectory 実装の設計によって影響があるすべてのユーザへの連絡
- および 86 ページの 「組織の構造」の情報の参照 85 ページの 「ネットワークのレイアウト」

OpenText eDirectory ツリーの設計

ネットワークの設計と実装で最も重要な作業は、OpenText eDirectory ツリーの設計です。ツリーの設計には、次の作業が含まれます。

- 86 ページの 「命名標準ドキュメントを作成する」
- 89 ページの 「Tree の上位層を設計する」
- 92 ページの 「ツリーの下位層を設計する」

命名標準ドキュメントを作成する

オブジェクト名などの標準名を規定すると、ユーザと管理者の両方が、ネットワークをより直感的に理解できるようになります。書き出された標準では、管理者による、電話番号や住所などの他のプロパティ値の設定方法も指定できます。

ディレクトリの検索とブラウズでは、名前やプロパティ値の整合性が重要になります。

また、標準的な名前を使用することで、OpenText Identity Manager は OpenText eDirectory とその他のアプリケーションの間でデータを容易に移動できます。OpenText Identity Manager の詳細については、『*OpenText Identity Manager* (https://www.netiq.com/documentation/identity-manager-4.10/)』を参照してください。

命名規則

- 87 ページの 「オブジェクト」
- 87 ページの 「サーバオブジェクト」
- 87 ページの 「国オブジェクト」

オブジェクト

- コンテナ内で一意の名前にします。たとえば、同じコンテナ内で Debra Jones と Daniel Jones の両方に「DJONES」という名前を付けることはできません。
- 特殊文字を使用することもできます。ただし、プラス記号 (+)、等号 (=)、およびピリオド (.) を使用する場合は、直前に円記号 (\) を入力する必要があります。サーバオブジェクトとカントリオブジェクト、およびバインダリサービスと多言語環境には、追加の命名規則が適用されます。
- 大文字と小文字、およびアンダースコアとスペースは、入力時にはそのまま表示されますが、実際には区別されません。たとえば Manager_Profile と MANAGER PROFILE は同一と見なされます。
- 名前をコマンドラインやログインスクリプトに入力するときにスペースを使用する場合は、名前全体を引用符で囲む必要があります。

サーバオブジェクト

- 新しいサーバをインストールすると、サーバオブジェクトが自動的に作成されます。
- 既存の Windows サーバ、および他のツリー内の OpenText eDirectory サーバに対して、追加のサーバオブジェクトを作成できますが、それらはすべてバインダリオブジェクトとして扱われます。
- サーバオブジェクトを作成するとき、その名前は物理サーバ名と一致していなければなりません。また、サーバ名には次の規定があります。
  - ネットワーク全体で固有である
  - 2 ～ 47 文字の長さである
  - A ～ Z までの文字、0 ～ 9 までの数字、ハイフン (-)、ピリオド (.)、およびアンダースコア (_) のみを含む
  - 最初の文字にピリオドを使用しない
- 一度付けたサーバオブジェクト名を OpenText Identity Console で変更することはできません。サーバで名前を変更すると、新しい名前が自動的に OpenText Identity Console に表示されます。

国オブジェクト

国オブジェクトは、2 文字の標準 ISO カントリコードに従って命名します。

詳細については、ISO 3166 Code Lists (https://www.iso.org/iso-3166-country-codes.html) を参照してください。

多言語環境の注意事項

複数の言語で稼動しているワークステーションがある場合は、必要に応じて、オブジェクト名に使用する文字を、すべてのワークステーションで表示できる文字のみに制限します。たとえば、ワークステーションを日本語環境で使用している場合には、ヨーロッパの言語で表示できない文字を名前に使用しないよう制限します。

重要 : Tree 名は必ず英語で指定するようにします。

標準ドキュメントの例

次は、最もよく使用される一部のプロパティに関する標準ドキュメントの例です。使用しないプロパティについては、標準を規定する必要はありません。標準ドキュメントは、オブジェクトの作成または修正を担当するすべての管理者に配布します。

オブジェクトクラス	プロパティ	Standard	例	理由
ユーザ	ログイン名	ファーストネームのイニシャル、ミドルネームのイニシャル (ある場合)、姓の組み合わせ (すべて小文字)。最大 8 文字です。各共通名はすべて、社内全体で固有のものにします。	msmith、bjohnson	OpenText eDirectory では会社全体で固有の名前を使用する必要はありませんが、固有にすると、同じコンテキスト (またはバインダリコンテキスト) 内での不整合を避けることができます。
ユーザ	姓	姓 (通常の大文字 / 小文字表記)。	Smith	メールラベルの生成に使用されます。
電話番号および Fax 番号		ハイフンで区切られた番号。	米国 : 123-456-7890、その他 : 44-344-123456	自動ダイヤルソフトウェアで使用されます。
複数のクラス	地域	2 文字の地域コード (大文字)、ハイフン、メール配達地点の組み合わせ。	BA-C23	社内のメール配達で使用されます。
組織	名前	すべてのツリーに与える会社名。	YourCo	独立したツリーがある場合、標準の組織名を使用することで、将来ツリーのマージができます。
部門	名前 (ロケーションに基づく)	2 または 3 文字の地域コードで、すべて大文字。	ATL、CHI、CUP、LA、BAT、BOS、DAL	短く、標準的な名前を使用することで、効率的に検索できます。

オブジェクトクラス	プロパティ	Standard	例	理由
部門 | 名前 (部署に基づく)		部署名または略語。	Sales、Eng	短く、標準的な名前を使用することで、どの部署で使用しているコンテナか見分けやすくなります。
グループ | 名前		わかりやすい名前。	Project Managers	ユーティリティによってはすべて表示できない場合があるため、極端に長い名前を避けてください。
ディレクトリマップ | 名前		ディレクトリマップが示すディレクトリの内容。	DOSAPPS	短く、標準的な名前を使用することで、どの部署で使用しているコンテナか見分けやすくなります。
プロファイル | 名前		プロファイルの目的。	MobileUser	短く、標準的な名前を使用することで、どの部署で使用しているコンテナか見分けやすくなります。
サーバ | 名前		SERV、ハイフン、部署、ハイフン、固有番号の組み合わせ。	SERV-Eng-1	OpenText eDirectory では、ツリー内で固有のサーバ名を使用する必要があります。

Tree の上位層を設計する

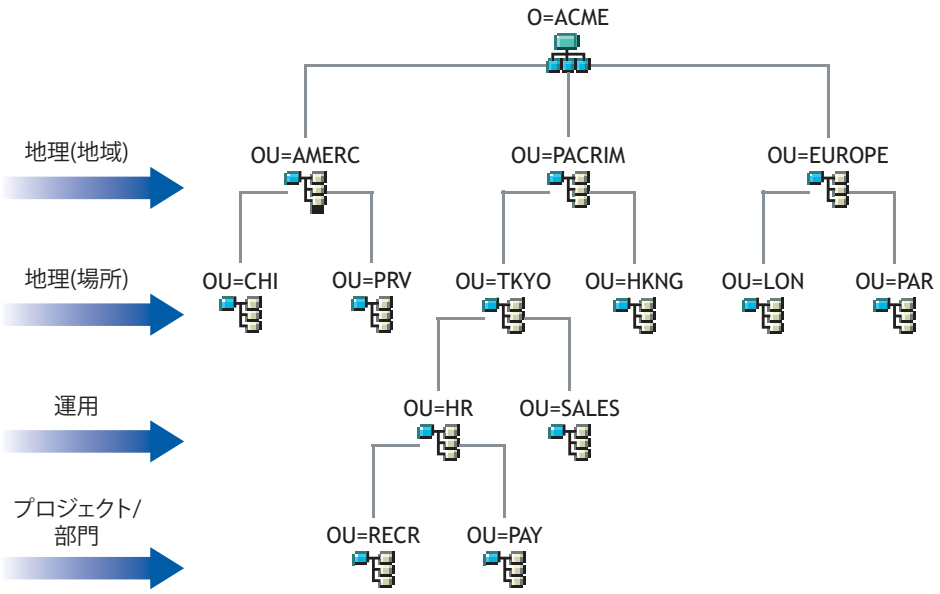
ツリーの上位層を変更するとツリーの残りの部分すべてに影響するため、ツリーの上位層は慎重に設計します。WAN リンクがある場合は、特に注意する必要があります。ツリーの最上部は、後からの変更がなるべく必要でないように設計します。

OpenText eDirectory ツリーの作成時には、次の OpenText eDirectory 設計規則に従います。

- ピラミッド型の設計を使用する。
- OpenText eDirectory ツリーは 1 つにし、固有の名前を付ける。
- 組織オブジェクトを 1 つ作成する。
- 物理的なネットワーク構造を表す第 1 レベルの部門群を作成する。

図 2-1 は、OpenText eDirectory 設計規則を示しています。

図 2-1 OpenText eDirectory 設計規則



ツリーの上位層を作成するには、110 ページの 「オブジェクトを作成する」および 110 ページの 「オブジェクトのプロパティの変更」を参照してください。

ピラミッド型の設計の使用

ピラミッド型の OpenText eDirectory では、管理や、大きなグループに対する変更、および論理パーティションの作成をより容易に実行できます。

ピラミッド型以外の設計として、フラットツリー型があります。この場合、すべてのオブジェクトがツリーの最上部に置かれます。OpenText eDirectory では、フラットツリー設計をサポートできます。ただし、フラットツリー設計では、管理やパーティション化が難しくなる場合があります。

固有の名前を持つ単一の OpenText eDirectory ツリーの使用

ほとんどの組織では、ツリーを 1 つにするのが最適です。デフォルトでは、ツリーは 1 つだけ作成されます。ツリーが 1 つだけだと、ネットワーク内のユーザの識別を統一でき、セキュリティ管理もより容易になります。また、1 箇所で集中管理できます。

業務では単一のツリーの使用を推奨しますが、追加のツリーのテストや開発を否定するわけではありません。

組織によっては、法的または政治的な問題、あるいは社内の事情から、複数のツリーが必要になる場合もあります。たとえば、自立した 7 つの組織から構成される組織では、7 つのツリーが必要になることも考えられます。組織で複数のツリーが必要な場合は、管理を容易にするために OpenText Identity Manager を使用することを検討してください。OpenText Identity Manager の詳細については、『』を参照してください。

ツリーには、他のツリー名と重複しない、固有の名前を付けます。「EDL-TREE」のように、短くてわかりやすい名前を指定します。

同じネットワーク内に同じ名前のツリーが存在すると、次のような問題が発生する場合があります。

- 更新内容が、対象のツリーとは別のツリーに適用される
- リソースが消失する
- 権利が消失する
- データが破損する

DSMerge ユーティリティを使用してツリー名を変更できますが、変更する際には十分に注意する必要があります。ツリー名の変更はネットワーク全体に影響します。ツリー名を変更した場合は、新しいツリー名でクライアントを再設定する必要があります。

単一の組織オブジェクトの作成

通常、1 つの OpenText eDirectory ツリーには、1 つの組織オブジェクトを作成します。デフォルトでは、組織オブジェクトが 1 つ作成され、それに会社名に基づいた名前が付けられます。これにより、会社全体に適用される変更をツリー内の 1 個所から設定できます。

たとえば、ZENworks® を使用して、組織オブジェクトの中にワークステーションインポートポリシーオブジェクトを作成できます。このポリシーは、OpenText eDirectory 内にワークステーションオブジェクトを作成するときの命名方法を定義するもので、組織全体に影響します。

組織コンテナには、次のオブジェクトが作成されます。

- Admin
- サーバ
- ボリューム

  OpenText eDirectory を実行している Windows または Linux サーバのみを含むネットワークには、ボリュームオブジェクトがありません。

次のようなケースでは、組織オブジェクトを複数作成することが必要になる場合があります。

- 複数の会社から構成される企業で、それぞれの会社が個別にネットワークを構成している。
- 社内で、独立した業務単位または組織を表す必要がある。
- 各部門が独立した形態をとることを規定する社内ガイドラインやポリシーがある。

物理的なネットワークを表す部門を作成する

第 1 レベルの部門設計は、OpenText eDirectory の効率性とパーティション化に影響するため重要です。

LAN または WAN を使用して複数のビルや場所などにまたがって構成されているネットワークでは、所在地に基づいて第 1 レベルの部門オブジェクトを設計します。これにより、1 つのパーティション内のすべてのオブジェクトが 1 箇所に維持されるように、OpenText eDirectory を分割できます。また、各場所でのセキュリティの設定や管理者の割り当ても容易になります。

ツリーの下位層を設計する

ツリーの下位層は、ネットワークリソースの編成に基づいて設計します。OpenText eDirectory ツリーの下位層の設計は同じ場所に存在するオブジェクトにのみ影響するため、下位層は上位層よりも自由に設計できます。

ツリーの下位層を作成するには、110 ページの 「オブジェクトを作成する」および 110 ページの 「オブジェクトのプロパティの変更」を参照してください。

コンテナ、ツリー、およびデータベースのサイズを決定する

作成する下位層のコンテナオブジェクトの数は、ツリー内のオブジェクトの総数、空きディスク容量、およびディスクの入出力速度制限によって決まります。eDirectory は、1 つの OpenText eDirectory ツリー内に 10 億以上のオブジェクトを格納できることがテストで確認されています。したがって、実際に制約となる項目は、空きディスク容量とディスクの入出力速度、およびパフォーマンスを維持するための RAM です。大規模なツリーの場合は、レプリカの同期処理が大きく影響することにも注意が必要です。

OpenText eDirectory 内の一般的なオブジェクトのサイズは 3 ～ 5KB です。この数値に基づいて、現在保持している、または新たに作成するすべてのオブジェクトの保管に必要な空きディスク容量をすばやく計算できます。オブジェクトのサイズは、データに付属する属性の数や、データの内容に対応して大きくなります。画像やサウンド、または生物統計学などの BLOB(バイナリラージオブジェクト) データを格納するオブジェクトの場合、そのサイズは増加します。

パーティションのサイズが大きくなるほど、レプリケーションのサイクルは遅くなります。ZENworks や DNS/DHCP サービスなど、OpenText eDirectory を使用する必要がある製品を使用する場合、これらの製品によって作成された OpenText eDirectory オブジェクトにより、格納先のコンテナのサイズが影響を受けます。場合によっては、DNS/DHCP など、管理目的のみで使用するオブジェクトは固有のパーティションに格納するように検討してください。パーティションを別にすれば、レプリカの同期処理の速度の低下によってユーザのアクセスに支障をきたすことを避けることができます。また、パーティションとレプリカの管理も容易になります。

必要な場合は、OpenText eDirectory データベースまたは DIB(ディレクトリ情報データベース) セットのサイズを簡単に判別できます。

- Windows の場合は、\novell\nds\dibfiles にある DIB セットを確認します。
- Linux の場合は、インストール時に指定したディレクトリにある DIB セットを確認します。

作成するコンテナを決定する

通常、同じ必要性からアクセスされる OpenText eDirectory オブジェクトをまとめて格納するコンテナを作成します。それにより、1 つのトラスティ割り当てまたはログインスクリプトで多くのユーザにサービスを提供できます。特にログインスクリプトをより効率的にすることを目的としてコンテナを作成できるほか、1 つのコンテナに 2 つの部署を割り当てることによって、ログインスクリプトの管理をより容易にすることもできます。

ネットワーク内のトラフィックを制限するため、ユーザは各自が必要とするリソースの近くに配置するようにします。たとえば、同じ部署で働く社員は、通常、隣り合った位置で作業します。それらの社員は同じファイルシステムにアクセスし、同じプリンタを使用して印刷します。

一般的なワークグループ境界の例外については、容易に対処できます。たとえば、2 つのワークグループが共通のプリンタを使用するような場合は、一方のワークグループのプリンタに対して別名オブジェクトを作成します。グループオブジェクトを作成することによって、1 つのワークグループ内だけでなく、複数のワークグループに存在する一部のユーザオブジェクトをまとめて管理できます。また、固有のログインスクリプト要件を持つ、一部のユーザ用のプロファイルオブジェクトも作成できます。

ツリーのパーティション化のガイドライン

OpenText eDirectory にパーティションを作成すると、データベースの各部分は複数のサーバに分割して格納されます。パーティション化によって、OpenText eDirectory データ処理と保管の負荷がネットワーク内の複数のサーバに分散されるため、ネットワークの使用を最適化できます。デフォルトでは、パーティションは 1 つだけ作成されます。パーティションの詳細については、59 ページの 「パーティション」を参照してください。パーティションの作成については、159 ページの第 6 章「パーティションおよびレプリカの管理」を参照してください。

次のガイドラインは、ほとんどのネットワークに適用できる一般的な規則です。特定の環境設定、ハードウェア、およびトラフィックの処理量など、それぞれのネットワークの運用条件に合わせて若干の調整が必要になる場合もあります。

ツリーの上位層におけるパーティションを決定する

ツリーをピラミッド型に設計するのと同じように、パーティションもピラミッド型に構成します。パーティションの構造は、ツリーの最上位に少数のパーティションがあり、下位になるに従ってパーティションの数が増えていくようにします。このような設計では、下位層より上位層のほうがパーティション数が多い OpenText eDirectory ツリー構造に比べて、作成されるサブオーディネートリファレンスの数が少なくなります。

ツリーをピラミッド型に設計するには、常にリーフオブジェクト (特にユーザ) の比較的近くにパーティションを作成するようにします

注 : インストール時にツリーのルートに作成されるパーティションについては例外です。

上位層のパーティションを設計する場合は、次の点を考慮します。

- WAN インフラストラクチャに基づいて、ツリーの最上部をパーティション化する。ツリーの最上部にはパーティションを少なめに配置し、下位になるに従ってパーティションの数を増やします。

  WAN リンクによって区切られている各サイトのコンテナを作成し (各サーバオブジェクトをそのローカルコンテナに格納)、その後、各サイトのパーティションを作成します。
- WAN リンクを持つネットワークでは、パーティションが複数の場所にまたがらないようにする。

  パーティションを場所ごとに作成すれば、異なるサイト間のレプリケーショントラフィックによって WAN 帯域幅が不必要に消費されることはありません。
- ローカルサーバを中心にしてパーティションを作成する。物理的に離れた場所にあるサーバは別のパーティションに格納します。

ツリーの下位層におけるパーティションを決定する

OpenText eDirectory ツリーの下位層のパーティションを設計する場合は、次を考慮します。

- 下位層のパーティションは、事業部、部署、およびワークグループと、それらに関連するリソースに基づいて定義する。
- 各パーティションについて、パーティション内のすべてのオブジェクトが同じ場所にあるようにする。それによって、OpenText eDirectory への更新がローカルサーバで行われるようになります。

パーティションサイズを決定する

OpenText eDirectory では、パーティションのサイズについて次の設計制限を推奨しています。

要素	制限容量
パーティションのサイズ	無制限のオブジェクト レプリカ DIB(ディレクトリ情報ベース) は 1TB に制限
ツリー内のパーティションの総数	無制限
ペアレントパーティションごとのチャイルドパーティション数	150
パーティションごとのレプリカ数	50 レプリカ DIB による制限
レプリカサーバごとのレプリカ数	250

NDS® 6 および 7 からのこの設計ガイドラインの変更は、NDS 8 におけるアーキテクチャの変更によるものです。これらの推奨値は、企業などの分散環境に適用されます。ただし、これらは e ビジネスやアプリケーションには適用されない場合があります。

一般的な e ビジネスユーザはすべてのデータを単一のサーバに保管することを必要としますが、OpenText eDirectory では、ツリー内のさまざまなエリアに属するオブジェクトや属性のサブセットを格納するフィルタ済みレプリカを作成できます。この機能を利用すると、1 つのサーバにすべてのデータを保管しなくても、同じように e ビジネスに対応できます。詳細については、66 ページの 「フィルタ済みレプリカ」を参照してください。

ネットワーク変数について

パーティションを設計する場合は、次のネットワーク変数について考慮します。

- サーバの数および速度
- ネットワークアダプタ、ハブ、ルータなどのネットワークインフラストラクチャの速度
- ネットワークトラフィックの量

ツリーのレプリカ作成に関するガイドライン

複数の OpenText eDirectory パーティションを作成しても、そのこと自体で耐障害性が向上したり、ディレクトリのパフォーマンスが向上したりするわけではありません。ただし、戦略的に複数のレプリカを使用すると、それらを実現できます。レプリカの配置は、アクセス可能性や耐障害性にとってきわめて重要です。OpenText eDirectory のデータには、できるだけすばやくアクセスできるようにする必要があります。また、それらのデータは、

耐障害性を確保するために複数の場所にコピーする必要があります。レプリカの作成については、159 ページの第 6 章「パーティションおよびレプリカの管理」を参照してください。

レプリカの配置計画では、次のガイドラインを参考にしてください。

- 96 ページの 「ワークグループのニーズ」
- 96 ページの 「障害対策」
- 97 ページの 「レプリカ数を決定する」
- 98 ページの 「Tree パーティションのレプリカを作成する」
- 98 ページの 「管理用にレプリカを作成する」
- 98 ページの 「WAN トラフィックを管理する」

ワークグループのニーズ

各パーティションのレプリカを、そのパーティション内の情報を使用するワークグループに物理的に近いサーバ上に作成します。WAN リンクの一方の側のユーザが他方の側のサーバに格納されているレプリカに頻繁にアクセスする場合は、WAN リンクの両端のサーバ上にそれぞれレプリカを作成します。

レプリカは、ユーザ、グループ、およびサービスによるアクセスが最も多い場所に配置します。2 つの独立したコンテナ内の各ユーザグループが別のパーティション境界内の同じオブジェクトにアクセスする必要がある場合は、これらのユーザグループが属する 2 つのコンテナより 1 つ上のレベルにあるコンテナ内のサーバ上にレプリカを配置します。

障害対策

ディスクがクラッシュしたり、サーバがダウンした場合、別の場所にあるサーバ上のレプリカによりユーザのネットワークへの認証を行い、使用不能になったサーバに格納されているパーティション内のオブジェクトの情報を取得できます。

いくつかのサーバに同じ情報が配布されていると、ユーザのネットワークへの認証やログインなどのサービスを提供するために、1 つのサーバに依存する必要がありません。

耐障害性を確保するには、ディレクトリツリーに十分な数のサーバがあれば、各パーティションについて 3 つのレプリカを作成します。ローカルパーティションのローカルレプリカは、最低 2 つは必要です。他の場所にあるデータへのアクセスを確保する場合や、負荷分散および障害対策用にデータのインスタンスを複数維持する必要があるアプリケーションを使用している場合、または e ビジネスを展開している場合を除いて、通常は、レプリカの作成は 3 つまでで十分です。

マスタレプリカは 1 つだけ保持できます。その他のレプリカは、読み書き可能、読み込み専用、またはフィルタ済みレプリカとして使用します。通常、ほとんどのレプリカは、読み書き可能レプリカとして使用されます。読み書き可能レプリカは、マスタレプリカ同様、オブジェクトの表示、オブジェクトの管理、およびユーザのログインを処理できます。変更が加えられると、読み書き可能レプリカは同期情報を送信します。

読み込み専用レプリカには、書き込みはできません。読み込み専用レプリカは、オブジェクトを検索および表示できます。また、パーティションのレプリカの同期が実行されると、更新されます。

サブオーディネートリファレンスレプリカやフィルタ済みレプリカは、障害対策用にはなりません。サブオーディネートリファレンスはポインタであり、パーティションのルートオブジェクト以外のオブジェクトは格納されません。フィルタ済みレプリカには、パーティション内の特定の一部のオブジェクトのみが格納されます。

OpenText eDirectory では、パーティションあたりで作成できるレプリカの数に制限はありませんが、レプリカの数が増えるに従って、ネットワークトラフィックの量も増大します。障害対策とネットワークパフォーマンス、それぞれの必要性をバランスよく配慮します。

サーバ上の各パーティションには、レプリカを 1 つだけ格納できます。1 つのサーバには、複数のパーティションのレプリカを格納できます。

組織の障害復旧計画によっては、サーバまたは特定の場所を喪失した場合、ネットワーク再構築のほとんどの作業がパーティションのレプリカを使用して行えます。サーバが 1 つしかない場所では、OpenText eDirectory を定期的にバックアップします障害対策用のレプリカを作成するための、追加サーバの購入を検討してください。

注

- 一部のバックアップソフトウェアでは、OpenText eDirectory が自動的にバックアップされません。
- ウイルス対策ソフトウェアやバックアップソフトウェアのプロセスから、OpenText eDirectory サーバ上の DIB ディレクトリを除外することをお勧めします。DIB ディレクトリのバックアップには、OpenText eDirectory バックアップツールを使用します。OpenText eDirectory のバックアップの詳細については、459 ページの 「OpenText eDirectory のバックアップと復元」を参照してください。

レプリカ数を決定する

複数のレプリカの作成を制限する要素となるのは、それぞれのレプリカの同期で必要となる処理時間とトラフィックの量です。オブジェクトが変更されると、変更内容がレプリカリング内のすべてのレプリカに伝達されます。レプリカリング内のレプリカの数が多いほど、変更内容の同期に必要なデータの伝送量も増えます。WAN リンク経由でレプリカを同期する必要がある場合、同期にかかる時間のコストが高くなります。

地理上の多くの場所でパーティションを作成した場合、一部のサーバではサブオーディネートリファレンスレプリカが多量に生成されることになります。OpenText eDirectory では、地域のパーティションを作成することによって、これらのサブオーディネートリファレンスをより多くのサーバに分散させることができます。

Tree パーティションのレプリカを作成する

Tree パーティションは OpenText eDirectory ツリー内の最も重要なパーティションです。このパーティションの唯一のレプリカが破損した場合、そのパーティションが修復されるか、OpenText eDirectory ツリー全体が再構築されるまで、ネットワークの機能が停止します。また、Tree に関連する設計の変更はまったくできなくなります。

Tree パーティションのレプリカの作成時には、サブオーディネートリファレンスの同期にかかるコストを考慮しながら、レプリカ数を決めてください。

管理用にレプリカを作成する

パーティションの変更はマスタレプリカからのみ発生するため、中央サイトのネットワーク管理者のすぐ近くにあるサーバに各マスタレプリカを保管するようにします。一見、リモートサイトにマスタを保管するほうが理に適っているように思えるかもしれませんが、マスタレプリカはパーティション操作が行われる場所に置いてください。

パーティション作成などの重要な OpenText eDirectory の操作は、中央サイトの特定の管理者または管理者グループが担当することを推奨します。担当者を限定することによって、OpenText eDirectory の動作を悪化させるエラーの発生の可能性を制限できるほか、マスタレプリカを中央で一括してバックアップできます。

ネットワーク管理者は、ネットワークトラフィックが少ない時間帯を選んで、レプリカの作成などのコストの高い操作を実行します。

WAN トラフィックを管理する

現在、ユーザが WAN リンクを使用して特定のディレクトリ情報にアクセスしている場合、必要な情報を格納するレプリカをユーザのローカルサーバに作成することによって、ユーザのアクセス時間と WAN トラフィックの量を削減できます。

マスタレプリカをリモートサイトに複製する場合や、アクセスの提供や障害対策用に WAN 経由でレプリカを配置する場合には、レプリカの同期処理で使用される帯域幅について考慮します。

ローカルサイト以外の場所におけるレプリカの作成は、推奨する 3 つのレプリカの作成がローカルサイトで不可能な場合に耐障害性を確保するためと、アクセスを容易にするため、またマスタレプリカを中央で一括して管理および保管するためにのみ行います。

ユーザ環境についてのプランニング

OpenText eDirectory ツリーの基本構造を設計し、パーティションとレプリカを設定した後は、管理を単純化し、ネットワークリソースへのアクセスを容易にするためのユーザ環境の設定プランを立てます。ユーザ環境プランを立てるには、ユーザのニーズを確認し、エリアごとにアクセシビリティに関するガイドラインを作成します。

ユーザのニーズを確認する

ユーザのニーズを確認する際には、次の点について考慮します。

- プリンタやファイル保管領域など、物理的なネットワークに関する必要条件

  1 つのツリー内の複数のユーザグループ、または複数のコンテナに属する複数のユーザグループによってリソースが共有されていないか確認します。また、リモートユーザが必要とする物理的なリソースについても考慮します。

- ユーザのバインダリサービスに関する必要条件

  バインダリサービスに基づくアプリケーションはどれか、またどのユーザによって使用されるか確認します。

- アプリケーションに関する必要条件

  ユーザが必要とするアプリケーションおよびデータファイルはどれか、オペレーティングシステムは何か、またアプリケーションにアクセスする必要があるユーザまたはユーザグループは誰かを確認します。共用アプリケーションを、ZENworks などのアプリケーションによって自動で起動するのか、または手動で起動するのか確認します。

アクセスに関するガイドラインを作成する

ユーザのニーズについての情報を収集した後、ユーザ環境の作成に使用する OpenText eDirectory オブジェクトを決定します。たとえば、ポリシーパッケージやアプリケーションオブジェクトを作成する場合は、作成数、およびツリー内の保管先について決定します。

また、ユーザのアクセスを制限するセキュリティの実装方法についても決める必要があります。個々のセキュリティ項目に関連する、セキュリティ上の予防措置がないか検討します。たとえば、OpenText eDirectory スーパバイザ権はファイルシステムによって継承されるため、サーバオブジェクトに対してはスーパバイザ権を付与しないよう、ネットワーク管理者に警告する場合があります。

e ビジネスに対応する OpenText eDirectory の設計

OpenText eDirectory を e ビジネスに利用する場合 (サービスのポータルを提供したり、他の企業とデータを共有したりする場合など)、この章ですでに説明している推奨事項は適用されないことがあります。

その場合は、e ビジネスに対応する OpenText eDirectory の設計ガイドラインとして、次に示す推奨事項を代わりに使用できます。

- コンテナ数を制限したツリーを作成する。

  このガイドラインは、使用するアプリケーション、および OpenText eDirectory の実装法に応じて適用します。たとえば、メッセージサーバをグローバルに展開する際には、この章の前半で説明した従来の OpenText eDirectory 設計ガイドラインを適用したほうがより適切な場合があります。また、ユーザの管理を分散させたい場合は、管理責任のエリアごとに独立した OU(部門) を作成することが必要になる場合があります。

- 最低 2 つのパーティションを維持する。

Tree レベルに作成されるデフォルトのパーティションを維持し、その他にツリーの残りの部分のパーティションを作成します。管理目的で OU を個別に作成した場合は、各 OU のパーティションを作成します。

複数のサーバに負荷を分散させるときには、パーティション数の制限を検討しますが、その場合もバックアップや障害回復用に最低 2 つのパーティションは維持します。

- 障害対策および負荷分散のために、ツリーのレプリカを最低 3 つ作成する。

  LDAP は負荷の分散を自動では行いません。LDAP の負荷を分散させるには、第 4 層スイッチの使用を検討します。

- e ビジネス用のツリーを別に作成する。サーバやプリンタなど、ツリーに組み込むネットワークリソースを制限する。ユーザオブジェクトのみを格納するツリーの作成を検討する。

  OpenText Identity Manager を使用すると、このユーザツリーをネットワーク情報を格納する自分の他のツリーにリンクさせることができます。詳細については、を参照してください。

- 補助クラスを使用して、スキーマをカスタマイズする。

  カスタマやアプリケーションで標準 inetOrgPerson とは異なるユーザオブジェクトが必要となる場合は、補助クラスを使用してスキーマをカスタマイズします。補助クラスを使用すると、アプリケーションの設計時にツリーを再作成しなくても、クラスで使用する属性を変更できます。

- LDIF インポートのパフォーマンスを向上させる。

  OpenText インポート / エクスポート変換ユーティリティを使用すると、OpenText eDirectory は処理中に各オブジェクトに索引を付けます。それによって、LDIF インポートプロセスの処理速度が遅くなることがあります。LDIF インポートのパフォーマンスを向上させるには、作成するオブジェクトの属性からのすべての索引付けをいったん中止し、OpenText インポート / エクスポート変換ユーティリティを使用してから、属性の索引付けを再開します。

- NDS 全体で固有の共通名 (CN) を実装する。

  OpenText eDirectory では、異なるコンテナ間で同じ共通名を使用できます。ただし、全体で固有の共通名を使用すれば、共通名の検索で複数の応答を処理するロジックを実装しなくてすみます。

OpenText Certificate Server について

OpenText Certificate Server では、セキュリティコンテナオブジェクトと組織の認証局 (CA) オブジェクトを作成することによって、デジタル証明書を作成、発行、および管理できます。組織の認証局オブジェクトにより、セキュリティで保護されたデータ伝送が可能になります。Web 関連製品には、組織の認証局オブジェクトが必須です。最初の OpenText eDirectory SP4 サーバによって、OpenText eDirectory ツリー全体のセキュリティコンテナオ

ブジェクトと組織の認証局オブジェクトが自動的に作成され、物理的に格納されます。どちらのオブジェクトも OpenText eDirectory ツリーの最上部に作成されます。これらのオブジェクトは、移動せずに作成時の位置にそのまま保管する必要があります。

1 つの OpenText eDirectory ツリー内に存在できる組織の認証局オブジェクトは 1 つだけです。最初のサーバ上にすでに作成されている組織の認証局オブジェクトを、別のサーバに移動することはできません。組織の認証局オブジェクトを削除したり、再作成すると、その組織の認証局に関連する証明書はすべて無効になります。

重要 : 組織の認証局オブジェクトを永続的に保管しようとするサーバを、最初の OpenText eDirectory サーバにするようにします。組織の認証局オブジェクトの保管先のサーバは、ネットワーク内の継続して稼働する部分であり、アクセス可能で、信頼性が高くなければなりません。

インストール中のサーバがネットワーク内の最初の OpenText eDirectory サーバでない場合、インストールプログラムは組織の認証局オブジェクトが存在する OpenText eDirectory サーバを探し、このサーバを参照します。インストールプログラムは、セキュリティコンテナにアクセスし、サーバ証明書オブジェクトを作成します。

ネットワーク内に組織の認証局オブジェクトがまったく存在しないと、Web 関連製品は機能しません。

OpenText Certificate Server でタスクを実行するのに必要な権利

OpenText Certificate Server の設定に関連するタスクを完了するには、次の表に記載されている権利を管理者が持っている必要があります。

OpenText Certificate Server のタスク	必要な権利
1 台目のサーバを新しいツリーにインストールするか、まだベースセキュリティをインストールしていないツリーの 1 台目のサーバをアップグレードするためのベースセキュリティ設定	ツリーのルートのスーパバイザ権 セキュリティコンテナのスーパバイザ権
2 台目以降のサーバをインストールするためのベースセキュリティ設定	サーバコンテナのスーパバイザ権 W0 オブジェクト (セキュリティコンテナ内) のスーパバイザ権
組織の認証局の作成	セキュリティコンテナのスーパバイザ権
サーバ証明書オブジェクトを作成する	サーバコンテナのスーパバイザ権 組織の認証局オブジェクトの NDSPKI: プライベートキー属性の読み込み権

また、ルート管理者は、サブコンテナの管理者に次の権利を割り当てることによって、組織の認証局を使用する権限を委託することもできます。サブコンテナ管理者が SSL セキュリティを使用して OpenText eDirectory をインストールするには、次の権利が必要です。

- セキュリティコンテナにある組織の認証局オブジェクトの NDSPKI: プライベートキー属性の読み込み権
- KAP オブジェクト内のセキュリティコンテナにある W0 オブジェクトのスーパバイザ権

これらの権利はグループや役割に割り当てられ、ここではすべての管理者ユーザが定義されます。OpenText Certificate Server に関連する特定のタスクを実行するのに必要な権利の詳細リストについては、703 ページの第 25 章「Certificate Server について」を参照してください。

Linux コンピュータ上の OpenText eDirectory 操作をセキュリティで保護する

eDirectory には、PKCS (公開鍵暗号化サービス) が付属しています。PKCS には、PKI (公開鍵インフラストラクチャ) サービスを提供する OpenText Certificate Server、NICI (Novell International Cryptographic Infrastructure)、および SAS-SSL サーバが含まれます。

次のセクションでは、OpenText eDirectory の操作におけるセキュリティの確保について説明します。

- 102 ページの 「サーバ上に NICI がインストールおよび初期化されているかどうかを確認する」
- 103 ページの 「サーバ上の NICI モジュールを初期化する」
- 103 ページの 「Certificate Server(PKI サービス) を起動する」
- 103 ページの 「Certificate Server(PKI サービス) を停止する」
- 103 ページの 「組織の認証局オブジェクトを作成する」
- 104 ページの 「サーバ証明書オブジェクトを作成する」
- 104 ページの 「組織の認証局の自己署名証明書のエクスポート」

外部認証局の使用については、703 ページの第 25 章「Certificate Server について」を参照してください。

サーバ上に NICI がインストールおよび初期化されているかどうかを確認する

次の条件を満たしているかチェックします。これらの条件は、NICI モジュールが正しくインストールおよび初期化されていることを表します。

- ファイル /etc/nici.cfg が存在する
- ディレクトリ /var/novell/nici が存在する
- ファイル /var/novell/nici/primenici が存在する

これらの条件を満たしていない場合、次のセクション (103 ページの 「サーバ上の NICI モジュールを初期化する」) の手順に従ってください。

サーバ上の NICI モジュールを初期化する

1 OpenText eDirectory サーバを停止します。

- Linux システムでは、次のコマンドを入力します。

  /etc/init.d/ndsd stop

重要 : ndsd の開始と停止には、ndsmangae を使用することをお勧めします。

---

2 NICI パッケージがインストールされているかどうかを確認します。

- Linux では、次のコマンドを入力します。

  rpm -qa | grep nici

3 (状況によって実行)NICI パッケージがインストールされていない場合、インストールします。

NICI パッケージがインストールされていないと先に進みません。

4 OpenText eDirectory サーバを開始します。

- Linux システムでは、次のように入力します。

  /etc/init.d/ndsd start

重要 : ndsd の開始と停止には、ndsmangae を使用することをお勧めします。

Certificate Server(PKI サービス) を起動する

PKI サービスを開始するには、次のコマンドを入力します。

npki -1

Certificate Server(PKI サービス) を停止する

PKI サービスを停止するには、次のコマンドを入力します。

npki -u

組織の認証局オブジェクトを作成する

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、『*OpenText Certificate Server 3.3 Administration Guide*』の「Creating an Organizational Certificate Authority Object (https://www.netiq.com/documentation/edir88/crtadmin88/data/fbgccghh.html)」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] をクリックします。

4 [**CA 管理**] > [**認証局管理**] の順にクリックします。

組織の認証局オブジェクトが存在しない場合、[Create an Organizational Certificate Authority Object] ダイアログボックスとオブジェクトを作成するウィザードが開きます。メッセージに従ってオブジェクトを作成します。ダイアログボックスまたはウィザードページの具体的な説明については、[**ヘルプ**] をクリックしてください。

注 : OpenText eDirectory ツリーは組織の認証局を 1 つしか格納できません。組織の認証局の作成の詳細については、706 ページの 「組織用の組織認証局を作成する」を参照してください。

サーバ証明書オブジェクトを作成する

サーバ証明書オブジェクトは、OpenText eDirectory サーバオブジェクトを保持するコンテナに作成されます。必要に応じて、サーバ上の個々の暗号化対応アプリケーションに対して、サーバ証明書オブジェクトを個別に作成することができます。あるいは、そのサーバで使用するすべてのアプリケーションに対してサーバ証明書オブジェクトを 1 つ作成することもできます。

注 : サーバ証明書オブジェクトと暗号化キーオブジェクト (KMO) は同じ意味です。OpenText eDirectory オブジェクトのスキーマ名は NDSPKI: 暗号化キーです。

---

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、719 ページの 「サーバ証明書オブジェクトを作成する」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] をクリックします。

4 [**サーバ証明書の管理**] をクリックして、[**サーバ証明書の作成** ] をクリックします。

これによってサーバ証明書の作成ウィザードが開始します。メッセージに従ってオブジェクトを作成します。ウィザードのページに関する具体的な説明については、[**ヘルプ**] をクリックしてください。

組織の認証局の自己署名証明書のエクスポート

自己署名付き証明書は、組織の認証局の識別情報と、組織の認証局によって署名された証明書の有効性を確認するために使用できます。

組織の認証局のプロパティページでは、このオブジェクトに関連付けられた証明書とプロパティを参照できます。[自己署名付き証明書] プロパティページでは、自己署名付き証明書を、暗号化対応のアプリケーションで使用するファイルにエクスポートできます。

組織の認証局に存在する自己署名証明書は、組織の認証局によって署名された証明書を持つサーバ証明書オブジェクトのルート認証局証明書と同じものです。組織の認証局の自己署名付き証明書をルート認証局証明書として認識するサービスでは、組織の認証局によって署名された有効なユーザやサーバが許可されます。

1 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**CA 管理**] の順にクリックします。
2 [**証明書**] タブをクリックして、[**自己署名証明書**] をクリックします。
3 **エクスポート**をクリックします。

   証明書のエクスポートウィザードが開始します。メッセージに従って証明書をエクスポートします。ウィザードのページに関する具体的な説明については、[**ヘルプ**] をクリックしてください。
4 [証明書のエクスポートの概要] ページで、[**Save the Exported Certificate to a File**] をクリックします。

   証明書がファイルに保存され、ルート認証局として暗号化対応アプリケーションにインポートできるようになります。
5 **閉じる**をクリックします。

OpenText eDirectory への保護接続を確立するためのすべてのコマンドライン操作にこのファイルを組み込みます。

ネットワーク時刻の同期

時刻同期は、ネットワーク全体で一貫したサーバ時刻を維持するためのサービスです。時刻同期は、OpenText eDirectory ではなく、サーバのオペレーティングシステムによって提供されます。OpenText eDirectory はその内部時刻に基づいて OpenText eDirectory パケットの正しい順序を維持しますが、その時刻はサーバのオペレーティングシステムから取得されます。

ネットワークで Windows または Linux を使用する場合は、広く利用されている時刻同期の標準である Network Time Protocol (NTP) を使用してサーバを同期してください。

NTP

NTP は、UDP プロトコルスイートの一部として機能します。UDP プロトコルスイートは、TCP/IP プロトコルスイートの一部として機能します。したがって、NTP を使用するコンピュータには、TCP/IP プロトコルスイートがロードされている必要があります。インターネットへアクセスするネットワーク内のコンピュータは、いずれもインターネット上の NTP サーバから時刻を取得できます。

NTP は、国際時刻標準である協定世界時 (UTC) と時刻を同期します。

NTP には、stratum(層) という概念が導入されています。stratum-1 サーバには、電波時計または原子時計などの正確な時計が内蔵されています。stratum-2 サーバは stratum-1 サーバから時刻を取得します。同様に、各層のサーバは 1 つ前の層のサーバから時刻を取得します。

時刻同期ソフトウェアの詳細については、Network Time Protocol (http://www.ntp.org) の Web サイトを参照してください。

Linux コンピュータで時刻を同期する

NTP (Network Time Protocol) デーモン xntpd を使用すると、Linux サーバで時刻を同期できます。xntpd とは、インターネット標準時サーバと同期してシステム時刻を設定および保持するオペレーティングシステムデーモンです。

Linux システムでの ntpd の実行については、「ntpd - Network Time Protocol (NTP) Daemon (http://www.eecis.udel.edu/~mills/ntp/html/ntpd.html)」を参照してください。

時刻同期を確認する

ツリー内の時刻が同期されているか確認するには、Tree オブジェクトに対して読み書き可能以上の権利を持つ、Tree 内のいずれかのサーバから、DSRepair を実行します。

Windows

1 ［**スタート**］>［**設定**］>［**コントロール パネル**］>［**OpenText eDirectory サービス**］の順にクリックします。

2 ［**dsrepair.dlm**］>［**開始**］の順にクリックします。

3 ［**修復**］>［**時刻同期**］の順にクリックします。

Linux

1 次のコマンドを実行します。

```
ndsrepair -T
```

3 オブジェクトの管理

OpenText eDirectory には、Web ベースのネットワーク管理アプリケーションである OpenText Identity Console が含まれています。これを使用すると、OpenText eDirectory ツリーのオブジェクトを管理できます。OpenText Identity Console の特長および利点については、『*OpenText Identity Console 管理ガイド*(https://www.netiq.com/documentation/identity-console/identity_console-admin/data/bookinfo.html)』を参照してください。

OpenText eDirectory オブジェクトの管理には、オブジェクトの作成、変更、および操作が含まれます。たとえば、ユーザアカウントの作成や、ユーザの権利の管理などの作業が必要になることがあります。OpenText Identity Console を使用すると、次の操作を行えます。

- オブジェクトの参照、作成、編集、および編成などの基本的な管理を行う。
- ユーザのログイン名や OpenText eDirectory で使用されるほかの情報を指定して、ユーザアカウントを作成する。
- 権利の割り当て、同等セキュリティの付与、継承の阻止、および有効な権利の参照を行い、権利を管理する。詳細については、78 ページの 「権利の管理」を参照してください。
- 役割ベースサービスオブジェクトを通じて特定の管理アプリケーションでの管理者の役割を定義し、役割ベースの管理を設定する。

この章では次のトピックについての情報を説明します。

- 107 ページの 「オブジェクトに関連する一般的なタスク」
- 112 ページの 「ユーザアカウントの管理」
- 119 ページの 「役割とアクセス制御の設定」

オブジェクトに関連する一般的なタスク

このセクションでは、OpenText eDirectory ツリーの管理で使用する基本的なタスクの手順について説明します。

- 108 ページの 「OpenText eDirectory ツリーをブラウズする」
- 110 ページの 「オブジェクトを作成する」
- 110 ページの 「オブジェクトのプロパティの変更」
- 111 ページの 「オブジェクトをコピーする」
- 111 ページの 「オブジェクトを移動する」
- 111 ページの 「オブジェクトの削除」
- 112 ページの 「オブジェクトをリネームする」

OpenText eDirectory ツリーをブラウズする

OpenText Identity Console のホームページの［**オブジェクトの管理 (Obejct Management)**］タイルを使用すると、OpenText eDirectory ツリー内のオブジェクトを検索またはブラウズできます。ツリーの構造を表示して、タスクを実行できます。実行可能なタスクは、選択するオブジェクトの種類によって異なります。

OpenText Identity Console の OpenText eDirectory の［**オブジェクトの管理 (Obejct Management)**］タイルを使用しても、オブジェクトを検索またはブラウズできます。OpenText Identity Console のほとんどのエントリフィールドでは、オブジェクト名およびコンテキストを指定したり、オブジェクトの［**タイプ**］を選択してオブジェクトを検索またはブラウズしたりできます。OpenText eDirectory の［**オブジェクトの管理 (Obejct Management)**］タイルでオブジェクトを選択すると、オブジェクトとそのオブジェクトのコンテキストが入力フィールドに挿入されます。

このセクションでは、次のことを説明します。

- 108 ページの 「［オブジェクトの表示］ボタンを使用する」
- 109 ページの 「［オブジェクトセレクタ］ボタンを使用する」

［オブジェクトの表示］ボタンを使用する

次で説明する方法を使用して、管理する特定のオブジェクトを検索します。

- 108 ページの 「［参照］を使用する」
- 109 ページの 「検索の使用」

［参照］を使用する

1. OpenText Identity Console のホームページで、［**オブジェクトの管理 (Obejct Management)**］をクリックします。
2. ［**オブジェクト**］ページで、［**名前**］を指定し、［**タイプ**］を選択して、［**コンテキスト**］を指定または検索します。
3. オブジェクトを参照するには、次のオプションを使用します。

オプション	説明
コンテキスト	内容を表示するコンテナの名前を指定します。 このオプションを使用するには、をクリックして、目的のコンテナを選択します。
名前	オブジェクトの名前を指定します。 このフィールドには、ワイルドカード文字としてアスタリスク (*) を使用できます。たとえば「g*」と指定すると、g で始まるすべてのオブジェクト (Germany、Greg など) が検出されます。「*te」と指定すると、te で終わるすべてのオブジェクト (Kate、Corporate など) が検出されます。
タイプ	検索するオブジェクトのタイプを指定します。デフォルト値は、［使用可能なすべてのタイプ］です。 このオプションを使用するには、ドロップダウンリストからオブジェクトタイプを選択します。

4 検索をクリックします。

検索の使用

1 OpenText Identity Console のホームページで、［オブジェクトの管理 (Obejct Management)］タイルをクリックします。

2 ［オブジェクト］ページで、検索するオブジェクトの［名前］を指定します。

このフィールドには、ワイルドカード文字としてアスタリスク (*) を使用できます。たとえば「g*」と指定すると、g で始まるすべてのオブジェクト (Germany、Greg など) が検出されます。「*te」と指定すると、te で終わるすべてのオブジェクト (Kate、Corporate など) が検出されます。

3 ［タイプ］ドロップダウンリストから、検索するオブジェクトのタイプを選択します。

4 ［コンテキスト］フィールドで、検索場所となるコンテナの名前を指定します。

［サブコンテナを検索］をクリックすると、現在のコンテナ内にあるすべてのサブコンテナが検索対象に含まれます。

5 検索をクリックします。

6 目的のオブジェクトを見つけたら、オブジェクトを右クリックして、実行可能なタスクのリストから選択します。

［オブジェクトセレクタ］ボタンを使用する

次で説明する方法を使用して、管理する特定のオブジェクトを検索します。

- 110 ページの 「［参照］を使用する」
- 110 ページの 「検索の使用」

［参照］を使用する

1 OpenText Identity Console のホームページで、［ツリービュー］タイルをクリックします。
2 コンテナおよびユーザをクリックして、必要な操作を実行します。

検索の使用

1 OpenText Identity Console のホームページで、［オブジェクトの管理 (Obejct Management)］タイルをクリックします。

［オブジェクト］ページが開きます。

2 ［名前］フィールドに名前を入力します。このフィールドでは、ワイルドカード文字としてアスタリスク (*) を使用できます。たとえば「g*」と指定すると、g で始まるすべてのオブジェクト (Germany、Greg など) が検出されます。「*te」と指定すると、te で終わるすべてのオブジェクト (Kate、Corporate など) が検出されます。
3 ドロップダウンリストから［タイプ］を選択します。
4 ［コンテキスト］を選択して、検索場所となるコンテナの名前を指定します。

［サブコンテナを検索］をクリックすると、現在のコンテナ内にあるすべてのサブコンテナが検索対象に含まれます。

5 検索をクリックします。

オブジェクトを作成する

1 OpenText Identity Console のホームページで、［オブジェクトの管理 (Obejct Management)］タイル ( ) をクリックします。
2 ［オブジェクトの作成 ］をクリックします。
3 ［名前］を入力して、［タイプ］を選択し、［コンテキスト］の >［次へ］の順にクリックします。
4 作成をクリックします。

必要な情報は、作成するオブジェクトのタイプによって異なります。

オブジェクトのプロパティの変更

1 OpenText Identity Console のホームページで、［オブジェクトの管理 (Obejct Management)］タイル ( ) をクリックします。
2 変更するオブジェクトを選択して、をクリックします。
3 変更する 1 つまたは複数のオブジェクトの名前とコンテキストを指定して、［OK］をクリックします。
4 目的のプロパティページを編集します。
5 ［保存］をクリックします。

オブジェクトをコピーする

このオプションを使用すると、既存のオブジェクトと同じ属性値を持った新しいオブジェクトを作成したり、あるオブジェクトから別のオブジェクトに属性値をコピーしたりできます。

1 OpenText Identity Console のホームページで、[DN 管理] タイルをクリックします。

2 [移動] をクリックし、[検索] をクリックして、コピー先を検索します。

3 [オブジェクト ＋] をクリックし、コピーするオブジェクトの名前とコンテキストを指定します。

4 次のいずれかのオプションを選択します。

- Create New Object and Copy Attribute Values
- Copy Attribute Values to an Existing Object

5 アクセス制御リスト (ACL) の権利を、作成 / 変更しているオブジェクトにコピーする場合、[ACL 権利のコピー] をクリックします。

ACL 権利をコピーする場合、システムやネットワークの環境によってはさらに処理時間がかかる場合があります。

6 OK をクリックします。

オブジェクトを移動する

1 OpenText Identity Console のホームページで、[DN 管理] タイル ( ) をクリックします。

2 [移動] > [検索] の順にクリックして、オブジェクトの移動先コンテナを 1 つ以上指定し、[OK] をクリックします。

3 [オブジェクト] フィールドで、移動する 1 つまたは複数のオブジェクトの名前とコンテキストを指定します。

4 [移動先] フィールドで、オブジェクトの移動先となるコンテナを指定します。

これにより、移動前の場所に依存するあらゆる操作は、操作を更新して移動後の場所を反映できるようになるまで、引き続き実行されます。

5 [保存] をクリックします。

オブジェクトの削除

1 OpenText Identity Console のホームページで、[オブジェクトの管理 (Obejct Management)] タイル ( ) をクリックします。

2 削除するオブジェクトを選択して、 をクリックします。

オブジェクトをリネームする

1 OpenText Identity Console のホームページで、[DN 管理] タイル ( ) をクリックします。

2 [名前変更] > [検索] の順にクリックして、名前を変更するオブジェクトを 1 つ以上選択します。

3 [新しい名前] フィールドで、名前変更するオブジェクトの名前およびコンテキストを指定します。

4 [新規オブジェクト名] フィールドで、オブジェクトの新しい名前を指定します。

[新規オブジェクト名] フィールドには、オブジェクトのコンテキストを含めないでください。

5 名前変更するオブジェクトに別名を作成するには、[リネームしたオブジェクトの代わりに別名を作成] を選択します。

これにより、リネーム前のオブジェクト名に依存するあらゆる操作は、操作を更新してリネーム後のオブジェクト名を反映できるようになるまで、引き続き実行されます。

6 名前変更する前のオブジェクト名を保存するには、[古い名前を保存] をクリックします。

これにより、リネーム前のオブジェクト名が、名前プロパティに未公認の値として追加されます。古い名前を保存すると、この名前に基づいてオブジェクトを検索できます。オブジェクトのリネーム後、そのオブジェクトの [General Identification] タブの [別の名前] フィールドでリネーム前の名前を確認できます。

7 [保存] をクリックします。

ユーザアカウントの管理

OpenText eDirectory のユーザアカウントの設定には、ユーザオブジェクトの作成、およびログイン制御のプロパティやユーザのネットワークコンピューティング環境の設定があります。テンプレートオブジェクトを使用すると、これらのタスクを簡単に実行できます。

ログインスクリプトを作成すると、ユーザがログインしたときに、自動的にファイルやプリンタなどの必要なネットワークリソースに接続できます。同じリソースを使用するユーザが複数いる場合、コンテナにログインスクリプトコマンドを格納して、ログインスクリプトのプロファイルを作成することができます。

このセクションでは、次のことを説明します。

- 113 ページの 「ユーザアカウントを作成および変更する」
- 114 ページの 「オプションのアカウント機能を設定する」
- 117 ページの 「ログインスクリプトを設定する」
- 118 ページの 「リモートユーザのログイン時間制限」
- 119 ページの 「ユーザアカウントを削除する」

ユーザアカウントを作成および変更する

ユーザアカウントは、OpenText eDirectory ツリー内のユーザオブジェクトです。ユーザオブジェクトは、ユーザのログイン名を指定するほか、OpenText eDirectory がネットワークリソースへのユーザアクセスを制御するために使用するその他の情報を提供します。

このセクションでは、次のことを説明します。

- 113 ページの 「ユーザオブジェクトを作成する」
- 113 ページの 「ユーザアカウントの変更」
- 113 ページの 「ユーザアカウントを有効にする」
- 114 ページの 「ユーザアカウントを無効にする」

ユーザオブジェクトを作成する

1 OpenText Identity Console のホームページで、[**ユーザ管理**] タイルをクリックします。

2 [**ユーザの追加** ＋] をクリックします。

3 [ユーザの追加] ページで、必要なユーザ関連情報を入力します。

- **[ユーザ名]**
- **コンテキスト**
- **名字**
- **[パスワード]**

4 ユーザを作成するコンテナを指定します。

5 オプションで追加情報を指定して、[**作成**] をクリックします。
「**ユーザが正常に作成されました**」というメッセージが表示されます。

6 **OK** をクリックします。

ユーザアカウントの変更

1 OpenText Identity Console のホームページで、[**ユーザ管理**] タイルをクリックします。

2 [**ユーザ**] を選択し、[**ユーザの変更**] をクリックします。

3 [**ユーザの変更**] ページで必要な変更を行い、[**保存**] をクリックします。

ユーザアカウントを有効にする

1 OpenText Identity Console のホームページで、[**ツリービュー**] タイルをクリックします。

2 [**novell**] を選択し、ユーザを選択して、[**アクションアイテム**] をクリックします。

3 [**アクションアイテム**] リストで、[**アカウントを有効にする**] を選択します。

4 [**OK**] をクリックします。

ユーザアカウントを無効にする

1 OpenText Identity Console のホームページで、[**ツリービュー**] タイルをクリックします。

2 [**novell**] を選択し、ユーザを選択して、[**アクションアイテム** ] をクリックします。

3 [**アクションアイテム**] リストで、[**アカウントを無効にする**] を選択します。

4 [**OK**] をクリックします。

オプションのアカウント機能を設定する

ユーザオブジェクトを作成した後、ユーザのネットワークコンピューティング環境を設定し、追加のログインセキュリティ機能を実装できます。

ユーザのネットワークコンピューティング環境の設定

1 OpenText Identity Console のホームページで、[**ユーザ管理**] タイルをクリックします。

2 ユーザを選択して、[**変更** ] をクリックします。

3 [**ユーザの変更**] ページで、[**一般**] タブ > [**環境とログインスクリプト**] の順にクリックします。

4 [**ログインスクリプト**] フィールドに必要なスクリプトを入力します。

5 [**保存**] をクリックします。

ユーザに追加のログインセキュリティを設定する

1 OpenText Identity Console のホームページで、[**ユーザ管理**] タイルをクリックします。

2 [**ユーザ**] > [**ユーザの変更** ] の順に選択します。

3 [**ユーザの変更**] ページ > [**制限**] > [**不正侵入者ロックアウト**] の順に移動します。

4 [**不正侵入者ロックアウト**] フィールドに、目的のプロパティページを入力します。
ページの詳細情報については、?をクリックしてください。

ページ	説明	LDAP 属性
パスワード制限	ログインパスワードを設定します。	passwordRequired
ログインの制限	◆ アカウントを有効または無効にします。 ◆ 同時ログインセッション数を制限します。 ◆ ログインの有効期限およびロックアウトする日付を設定します。	loginDisabled loginMaximumSimultaneous loginExpirationTime または loginGraceLimit
ログイン時間制限	ユーザがログインできる時間を制限します。制限を設定し、制限時間になったときにオブジェクトがログインされていると、5 分間警告が表示され、5 分後にまだオブジェクトがログアウトされていない場合は、そのオブジェクトをログアウトします。リモートからログインする場合は、「118 ページの 「リモートユーザのログイン時間制限」」を参照してください。	loginAllowedTimeMap
アドレス制限	このユーザがログインするネットワークの場所 (ワークステーション) を制限します。このページで制限を設定しない場合、ユーザはネットワークのどの場所からでもログインできます。	networkAddressRestriction
アカウントバランス	このユーザのサーバ使用状況のアカウントを設定します。	accountBalance
不正侵入者ロックアウト	不正侵入者が検出されたためにアカウントがロックされた場合、このアカウントを操作します。不正侵入者検出の設定を管理するには、親コンテナの［不正侵入者検出］プロパティページを使用します。	lockedByIntruder

5 ［保存］をクリックします。

コンテナ内のすべてのユーザの不正侵入者検出を設定する

1 OpenText Identity Console のホームページで、［ユーザ管理］タイルをクリックします。

2 ［ユーザ］ > ［ユーザの変更］の順にクリックします。

3 ［ユーザの変更］ページ > ［制限］ > ［不正侵入者ロックアウト］の順に移動して、［OK］をクリックします。

4 必要なパラメータを入力します。

オプション	説明
不正侵入者を検出する	コンテナのユーザアカウントの不正侵入者検出システムを有効にします。
不正ログイン試行回数	連続してログインに失敗して不正侵入者検出がアクティブになるまでのログイン試行回数を指定します。ユーザがログインにこのコンテナ内のユーザアカウントを使用し、連続して失敗した回数がこの値を超えると、不正侵入者検出がアクティブになります。この数値は、コンテナの［Login Intruder Limit(不正ログイン制限)］プロパティに格納されます。
不正ログイン回数のリセット間隔	ここで指定した時間間隔以内に連続してログインに失敗すると、不正侵入者検出がアクティブになります。日、時間、および分を入力します。
検出後にアカウントをロックする	このコンテナ内のユーザアカウントに対する不正侵入者検出がアクティブになった場合にログインを無効にするかどうかを指定します。このチェックボックスがオンになっていない場合、不正侵入者検出がアクティブになってもアカウントはロックされません。このチェックボックスをオンにし、不正侵入者の検出によってユーザアカウントがロックされた場合、ユーザオブジェクトの［不正侵入者ロックアウト］プロパティの［ロックされたアカウント］チェックボックスをオフにして、アカウントのロックを解除できます。
日、時間、分	この 3 つのフィールドでは、不正侵入者検出がこのコンテナ内のユーザアカウントでアクティブになった場合にログインが無効になる時間が指定されます。日、時間、分の値を任意に入力するか、デフォルトの 15 分をそのまま使用します。指定した時間が経過すると、そのユーザアカウントのログインは再び有効になります。これらのフィールドの内容は、コンテナの［アカウントロックアウト期間］プロパティに格納されます。これら 3 つのフィールドの値が 0 と指定されている場合、ユーザアカウントは無期限にロックされます。

5 ［**保存**］をクリックします。

ログイン時間更新間隔の無効化

ユーザのログイン時間属性の更新を無効にする間隔値を指定することができます。ユーザ、コンテナ、およびログインポリシーに対して間隔値を指定できます。セキュリティオブジェクト (LPO) またはサーバ。この機能を有効にするには、nmas.sch ファイルを使ってスキーマを拡張する必要があります。

ユーザに対して間隔を指定するには、次の手順を実行します。

1 OpenText Identity Console のホームページで、［**ユーザ管理**］をクリックします。

2 ログイン間隔を更新する必要があるユーザを選択します。

3 [ユーザの変更] ( ) をクリックして、[その他] タブに移動します。

4 [値がある属性] をクリックし、[属性の選択] リストから [sasUpdateLoginTimeInterval] を選択します。

5 [OK] をクリックして、ログイン間隔および他の必要な情報を指定します。

6 [保存] をクリックし、[OK] をクリックします。

コンテナおよび LPO の更新間隔を指定するには、次の手順を実行します。

1 OpenText Identity Console のホームページで、[オブジェクト管理] をクリックし、[タイプ] ドロップダウンで [部門 (Organization Unit)] を選択して、[検索] をクリックします。

2 コンテナまたはログインポリシーオブジェクトの名前とコンテキストを指定します。

3 [部門 (Organization Unit)] をクリックします。

4 [一般] タブで、[オプション属性の追加] をクリックし、[sasUpdateLoginTimeInterval] を選択して、[OK] をクリックします。

5 必要に応じて、矢印ボタンを使って [sasUpdateLoginTimeInterval] を [値がない属性] リストから [値がある属性] リストに移動し、[保存] をクリックします。

ログインスクリプトを設定する

ログインスクリプトは、ユーザがログインしたときに実行される一連のコマンドです。一般的に、ユーザとファイルやプリンタなどのネットワークリソースとの接続に使用されます。ログインスクリプトは、次の順序でユーザのワークステーション上で実行されます。

1. コンテナログインスクリプト
2. プロファイルログインスクリプト
3. ユーザログインスクリプト

ログイン中に、いずれかのログインスクリプトが見つからない場合、それをスキップしてリスト内の次のスクリプトに移ります。何も見つからない場合、デフォルトのスクリプトが実行され、検索ドライブがユーザのデフォルトサーバ上のフォルダにマップされます。デフォルトサーバは、ユーザオブジェクトの [使用環境] プロパティページで設定されます。

ログインスクリプトを作成する

1 OpenText Identity Console のホームページで、[ユーザ管理] タイルをクリックし、ユーザを選択して、[変更] をクリックします。

2 [ユーザの変更] ページで、[一般] > [環境とログインスクリプト] の順に選択します。

3 [環境とログインスクリプト] ドロップダウンをクリックし、[ログインスクリプト] フィールドにログインスクリプトコマンドを入力します。

詳細については、『*Login Scripts Guide (ログインスクリプトガイド)* (http://www.novell.com/documentation/linux_client/login/data/front.html)』を参照してください。

ログインスクリプトの適用先	作成先
1 人のユーザのみ	ユーザオブジェクト
まだ作成されていない 1 人以上のユーザ	テンプレートオブジェクト
1 つのコンテナ内のすべてのユーザ	コンテナオブジェクト
1 つ以上のコンテナの複数のユーザ	プロファイルオブジェクト

4 ［保存］>［OK］の順にクリックします。

ユーザにプロファイルを割り当てる

プロファイルをユーザオブジェクトと関連付けることによって、ユーザのログイン中にそのプロファイルのログインスクリプトが実行されます。ユーザが、プロファイルオブジェクトのブラウズ権、およびプロファイルオブジェクトのログインスクリプトプロパティの読み込み権を持っていることを確認してください。

詳細については、82 ページの 「OpenText eDirectory オブジェクトまたはプロパティへの有効な権利を参照する」を参照してください。

1 OpenText Identity Console のホームページで、［ユーザ管理］タイルをクリックし、ユーザを選択して、［変更 ］をクリックします。

2 ［ユーザの変更］ページで、［一般］>［環境とログインスクリプト］>［プロファイル ］の順に選択します。

3 ［コンテキストブラウザ］ページで、ログインスクリプトを作成したいユーザオブジェクトを検索できます。

4 ［保存］をクリックします。

リモートユーザのログイン時間制限

ユーザオブジェクトの［ログイン時間制限］プロパティページで、ユーザが OpenText eDirectory にログインできる時間を制限できます。デフォルトでは、ログイン時間に制限はありません。

ログイン時間制限が設定され、制限時間になったときにユーザがログインしていると、5 分以内にログアウトするよう警告が表示されます。ユーザが 5 分経ってもまだログインしている場合、自動的にログアウトされ、保存していないデータは失われます。

ログイン要求を処理するサーバとは異なるタイムゾーンからリモートでログインする場合、ユーザに設定されたログイン時間制限の時差は調整されます。たとえば、月曜日の午前 1 時から午前 6 時までにログインが制限されており、サーバより 1 時間遅いタイムゾーンからリモートでログインする場合、このユーザの制限は午前 2 時から午前 7 時まで有効になります。

1 OpenText Identity Console のホームページで、［管理の使用］タイルをクリックします。

2 ［ユーザ］を選択します。

3 ［制限］タブをクリックし、［時間制限］をクリックします。

4 次のオプションを選択します。

オプション	説明
タイムグリッド	タイムグリッドのそれぞれのセルは、表示されている週の 1 日のうちの 30 分を表します。赤いセルは、このオブジェクトにログインできない制限時間を表します。灰色のセルは制限されていない時間で、このオブジェクトにログインできる時間を表します。時間制限を設定するには、目的の時間をクリックして濃い灰色にします。複数の時間を選択するには、Shift キーを押しながらセルをクリックし、該当するセルまでドラッグします。設定したログイン時間制限は、このオブジェクトの［Login Allowed Time Map(ログイン許可時間マップ)］プロパティに格納されます。
ログイン時間制限の追加	時間制限を追加するには、灰色のセルをクリックして、このオプションを選択します。
ログイン時間制限の削除	時間制限を削除するには、赤いセルをクリックして、このオプションを選択します。
アップデート	このボタンをクリックすると選択が有効になります。
リセット	このボタンをクリックすると、このプロパティページを開く前の状態にタイムグリッドがリセットされます。

5 ［保存］をクリックします。

ユーザアカウントを削除する

1 OpenText Identity Console のホームページで、［ユーザ管理］タイルをクリックし、［ユーザ］を選択します。

2 ［ユーザの削除  ］をクリックします。

ユーザが削除されます。

役割とアクセス制御の設定

OpenText Identity Console を使用すると、管理者は、ユーザに特定の責任を割り当てたり、これらの責任を実行するために必要なツールおよびそれに伴う権利だけを与えたりすることができます。この機能は「役割とアクセス制御」(RAC) と呼ばれています。

RAC を使用することにより、管理者は、特定の機能 (「タスク」と呼ばれる)、および「役割」と呼ばれるタスクの集まりによって決定された特定のオブジェクトだけをユーザが使用できるように設定できます。ユーザが OpenText Identity Console にアクセスしたときの表示内容は、OpenText eDirectory での役割の割り当てによって異なります。OpenText でそのユーザに割り当てられているタスクだけが表示されます。ユーザは、管理対象オブジェク

トを見つけるためにツリーをブラウズする必要はありません。そのタスクの［OpenText Identity Console］タイルで、タスクの実行に必要なツールとインタフェースが提供されます。

1 人のユーザには複数の役割を割り当てることができます。また、複数のユーザに同じ役割を割り当てることもできます。

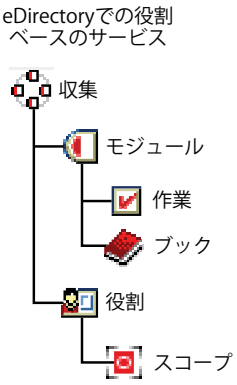
RAC は、OpenText eDirectory で定義されたオブジェクトによって表されます。基本の OpenText eDirectory スキーマは、OpenText Identity Console のインストール時に拡張されます。RAC オブジェクトタイプの一覧を、次の表に示します。

オブジェクト	説明
RAC 環境設定	すべての RAC 役割オブジェクトとモジュールオブジェクトを格納するコンテナオブジェクト。 RAC 環境設定オブジェクトは、すべての RAC オブジェクトの最上位コンテナです。ツリーには RAC 環境設定オブジェクトをいくつでも格納できます。これらのオブジェクトには、RAC 環境設定を管理する権利を持つユーザとして「所有者」が存在します。 RAC 環境設定オブジェクトは次のコンテナのいずれかに作成できます。 ◆ ドメイン ◆ 場所 ◆ 国 ◆ 組織 ◆ 部門
RAC 役割	ユーザ (メンバー) が実行を許可されたタスクを指定するコンテナオブジェクト。役割の定義には、RAC 役割オブジェクトの作成や役割が実行できるタスクの指定が含まれます。 役割メンバーには、ユーザ、グループ、組織、部門があり、ツリーの特定の範囲内の役割に関連付けられています。RAC タスクオブジェクトと RAC ブックオブジェクトは、RAC 役割オブジェクトに割り当てられます。 RAC 役割オブジェクトは、RAC 環境設定コンテナ内でのみ作成できます。
RAC モジュール	RAC タスクオブジェクトと RAC ブックオブジェクトを格納するコンテナオブジェクト。RAC モジュールオブジェクトには、タスクまたはブックを定義する製品名を表すモジュール名属性があります (たとえば、OpenText eDirectory Maintenance、NMAS、OpenText Certificate Access など)。 RAC モジュールオブジェクトは、RAC 環境設定コンテナ内でのみ作成できます。

オブジェクト	説明
RAC タスク	ログインパスワードのリセットなど、特定の機能を表すリーフオブジェクトです。 RAC タスクオブジェクトは、RAC モジュールコンテナ内にのみ格納されます。
RAC ブック	ブックに割り当てられる一連のページを含むリーフオブジェクトです。rbsBook は、1 つ以上の役割および 1 つ以上のオブジェクトクラスタイプに割り当てることができます。 rbsBook オブジェクトは、rbsModule コンテナ内にのみ格納されます。
RAC スコープ	ユーザオブジェクトごとに割り当てずに ACL 割り当てを実行するためのリーフオブジェクトです。RAC スコープオブジェクトは、役割が実行されるツリー内のコンテキストを表し、RAC 役割オブジェクトに関連付けられています。このオブジェクトはグループクラスから権利を継承します。ユーザオブジェクトは RAC スコープオブジェクトに割り当てられます。これらのオブジェクトは、関連付けられるツリーのスコープを参照します。 このオブジェクトは、必要な場合は動的に作成され、不要になれば自動的に削除されます。これらは RAC 役割コンテナ内にのみ格納されます。 **警告 :** Scope オブジェクトの環境設定を変更してはいけません。設定を変更することによって深刻な問題が発生し、システムが故障する可能性があります。

RAC オブジェクトは、次の図に示されているように OpenText eDirectory ツリーに属しています。

図 3-1 *OpenText eDirectory ツリー内の RAC オブジェクト*



RAC 役割の定義

RAC 役割は、ユーザが実行を許可されるタスクを指定します。RAC 役割の定義には、RAC 役割オブジェクトの作成や、役割が実行できるタスク、およびユーザ、グループ、またはこれらのタスクを実行できるコンテナオブジェクトの指定などがあります。場合によっては、OpenText Identity Console のタイル (製品パッケージ) で、変更可能な定義済みの RAC 役割が提供されることがあります。

RAC 役割が実行できるタスクは、OpenText eDirectory ツリー内では RAC タスクオブジェクトとして公開されます。これらのオブジェクトは、製品パッケージのインストールの際に自動的に追加されます。オブジェクトは 1 つ以上の RAC モジュールに編成され、異なる機能を持つ製品モジュールに対応するコンテナとなります。

役割へのメンバーの割り当てについての詳細は、「123 ページの 「RAC 役割のメンバーシップおよびスコープを割り当てる」」を参照してください。

- 122 ページの 「役割オブジェクトを作成する」
- 123 ページの 「役割に関連付けられたタスクを変更する」
- 123 ページの 「RAC 役割のメンバーシップおよびスコープを割り当てる」
- 124 ページの 「RAC 環境設定の役割を削除する」

役割オブジェクトを作成する

RAC 環境設定 (https://www.netiq.com/documentation/identity-console/identity_console-admin/data/b8qqsec.html) を使用して、新しい RAC 役割オブジェクトを作成します。新しい RAC 役割オブジェクトは、他の RAC 役割オブジェクトが保存されているのと同じ RAC 環境設定コンテナに作成することを推奨します (RAC 環境設定コンテナなど)。

1. OpenText Identity Console のホームページで、[**役割とアクセス制御**] をクリックします。
2. [**役割**] タブで、[**役割の作成** ＋] をクリックします。
3. [**役割の作成**] ウィンドウで [**スコープの追加** ] をクリックし、[**スコープ**] を選択します。
   1. [**名前**] を入力します。
   2. 役割に関連付ける**メンバー**またはメンバーのグループを選択します。
4. [**タスクの選択**] メニューから、役割に割り当てる必要があるタスクを選択します。
5. [**作成**] をクリックします。

   新しい役割が作成されます。ウィンドウを閉じてもう一度環境設定を開き、変更を確認します。

役割へのメンバーの追加についての詳細は、「124 ページの 「カスタムの RAC タスクの定義」」を参照してください。

役割に関連付けられたタスクを変更する

各 RAC 役割には、有効なタスクのセットが関連付けられています。特定の役割に割り当てるタスクは、必要に応じてタスクを追加したり削除したりすることで選択できます。

1 OpenText Identity Console のホームページで、[**役割とアクセス制御**] タイルをクリックします。

2 [**RAC 環境設定**] タブで、役割を変更する設定をクリックします。

3 [**役割**] タブで、変更する役割をクリックします。

4 (オプション) タスクを役割に追加するには、次の手順を行います。

4a [**タスクの追加 / 削除** ] をクリックします。

4b タスクを選択して、[**確認**] をクリックします。

5 (オプション) 役割からタスクを削除するには、次の手順を行います。

5a 削除するタスクを選択します。

5b [**タスクの追加 / 削除** ] をクリックします。

6 [**確認**] をクリックします。

RAC 役割のメンバーシップおよびスコープを割り当てる

所属する組織に必要な RAC 役割を定義すると、それぞれの役割にメンバーを割り当てることができます。その際、それぞれのメンバーが役割の機能を使用できるスコープを指定します。スコープとは、この役割を遂行できる OpenText eDirectory ツリー内の場所またはコンテキストです。

役割へのユーザの割り当ては、次の方法で行うことができます。

- 直接
- グループおよび動的グループの割り当てによる方法役割に割り当てられているグループまたはダイナミックグループのメンバーであれば、ユーザはその役割にアクセスできます。
- 職種の割り当てを使用する . 役割に割り当てられている職種に所属する場合は、ユーザはその役割にアクセスできます。
- コンテナの割り当てを使用する。ユーザオブジェクトは、その親コンテナが割り当てられたすべての役割にアクセスできます。さらにツリーのルートまで遡るコンテナの役割にもアクセスできます。

役割との関連付けは、それぞれ異なるスコープで何度も実行できます。また、同じタスクを複数のメンバーに割り当てることもできます。

既存のメンバーに役割を割り当てるには :

1 OpenText Identity Console のホームページで、[**役割とアクセス制御**] タイルをクリックします。

2 [**RAC 環境設定**] タブをクリックします。

3 [メンバー関連付けの編集 (Edit Member Associations) ＋] をクリックして、メンバーを選択します。

4 [役割の追加] ウィンドウで、必要な役割を選択します。

- [役割の追加]：役割メンバーにするオブジェクトを指定するか、オブジェクトセレクタを使用して検索します。
- [スコープの追加]：このメンバーが役割を実行できるスコープを、指定またはオブジェクトセレクタを使用して検索します。

5 [OK] をクリックします。

役割が正常に追加されます。ウィンドウを閉じてもう一度環境設定を開き、変更を確認します。. 詳細については、[ヘルプ] ( ) をクリックします。

RAC 環境設定の役割を削除する

1 OpenText Identity Console のホームページで、[役割とアクセス制御] タイルをクリックします。

2 [RAC] タブをクリックします。

3 RAC 役割を削除したい環境設定をクリックします。

4 変更する役割を選択します。

5 [役割の削除] をクリックします。

6 [OK] をクリックします。

カスタムの RAC タスクの定義

- 124 ページの 「OpenText Identity Console タスクの作成」
- 125 ページの 「関連付けられた役割の変更」
- 125 ページの 「タスクを削除する」

OpenText Identity Console タスクの作成

OpenText Identity Console タスクを作成するには、[カスタムフォーム] または [外部アプリケーション] を使用します。ここで作成されたフォームはタスクとして反映されます。詳細については、「カスタムフォームの管理 (https://www.netiq.com/documentation/identity-console/identity_console-admin/data/alw1iwfbb2eer.html)」および「外部アプリケーションの管理 (https://www.netiq.com/documentation/identity-console/identity_console-admin/data/alw1iwfbb2eerbb3b.html)」を参照してください。

関連付けられた役割の変更

1 OpenText Identity Console のホームページで、[**役割とアクセス制御**] タイルをクリックします。

2 [**メンバー関連付けの編集**] タブをクリックして、[**メンバーの選択** ] をクリックします。

3 ユーザがすでに参加している関連付けのリストが表示されます。

4 必要に応じて、[**スコープ**]、[**割り当てられた権利**]、および [**継承可能**] オブジェクトを変更します。

5 [**OK**] をクリックして閉じます。

タスクを削除する

1 OpenText Identity Console のホームページで、[**役割とアクセス制御**] タイルをクリックします。

2 [**RAC 環境設定**] タブをクリックします。

3 タスクの削除場所となる環境設定をクリックします。

4 削除するタスクを選択します。

5 **削除** をクリックします。

6 **OK** をクリックします。

4 バックグラウンドプロセスの管理

大規模な動的環境に対応する目的で、OpenText eDirectory には、環境に合わせてシステムを調整するための最適化されたバックグラウンドプロセスと環境設定オプションが備わっています。

この章には、次のトピックが含まれています。

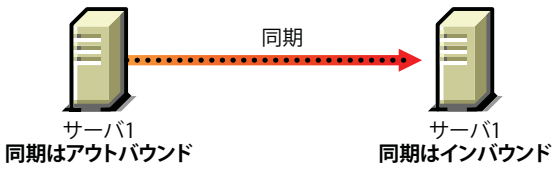
- 127 ページの 「同期」
- 142 ページの 「バックグラウンドプロセスの設定」

同期

同期とは、ディレクトリ情報をレプリカ間で転送し、各パーティション内の情報が互いに整合性を保つようにすることです。OpenText eDirectory により、レプリカリング内のサーバは自動的に同期されます。

同期には、インバウンド同期とアウトバウンド同期があります。たとえば、データに加えられた変更を server1 と server2 の間で同期する必要がある場合、*「アウトバウンド」という用語は、server1 から server2 に送信される同期プロセスを意味します。*「インバウンド」という用語は、server2 によって受信された server1 からの同期プロセスを意味します。

***図 4-1** アウトバウンド同期とインバウンド同期*



同期には、次の 2 つのタイプがあります。

- 通常同期またはレプリカ同期
- 優先度同期

次の表に、通常同期と優先度同期の比較を示します。

***表 4-1** 通常同期またはレプリカ同期と優先度同期の比較*

通常同期またはレプリカ同期	優先度同期
レプリカリング内の任意のサーバのデータに変更が加えられた場合にトリガされます。	重要データとして指定しているデータに変更が加えられた場合のみトリガされます。
詳細については、130 ページの 「通常同期またはレプリカ同期」を参照してください。	詳細については、132 ページの 「優先度同期」を参照してください。

通常同期またはレプリカ同期	優先度同期
データが変更されると、変更内容はバッファに保存されます。通常同期は、変更内容が保存されてから約 30 秒後に開始されます。	重要データへの変更はバッファに保存されません。優先度同期は、データが変更された直後に開始されます。
OpenText eDirectory で最も重要な同期です。変更が優先度同期によって同期されているかどうかにかかわらず実行されます。	通常同期を補完するものです。重要な属性は優先度同期によって同期され、通常同期によって再度同期されます。
OpenText eDirectory 8.8 サーバ間または以前のバージョンの OpenText eDirectory をホストするサーバ間で実行可能です。	同じパーティションを保持している OpenText eDirectory 8.8 以降のサーバ間でのみ実行されます。
機能上、失敗することはありません。 詳細については、128 ページの 「同期の特徴」を参照してください。	優先度同期が失敗した場合、重要なデータへの変更は通常同期によって同期されます。 詳細については、139 ページの 「優先度同期が失敗する場合」を参照してください。

注 : 優先度同期情報は、ndstrace、dstrace、または iMonitor トレース画面内の SYDL または同期詳細タグにあります。

同期の特徴

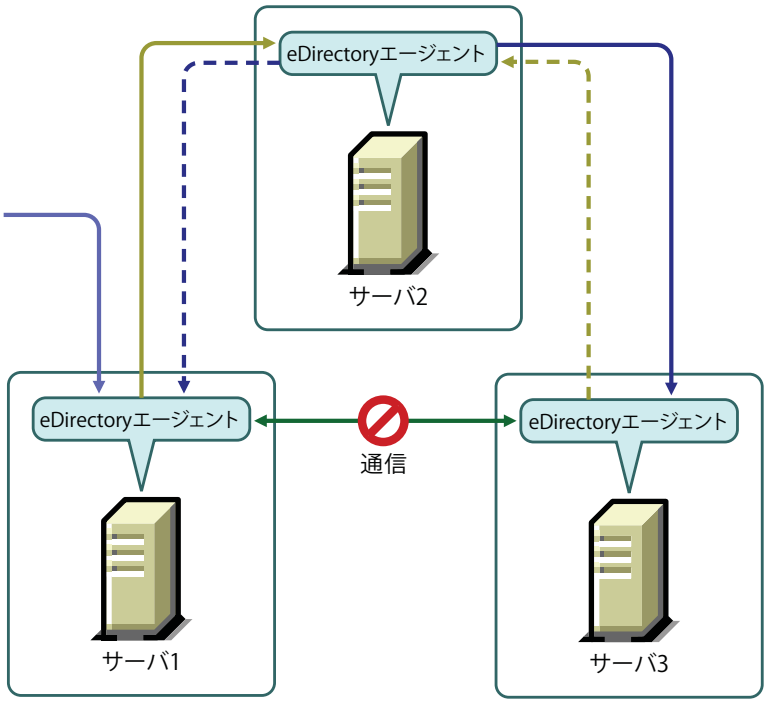
OpenText eDirectory での同期には、次のような特徴があります。

- 遷移同期です。
- オブジェクトトランザクションモデルを維持します。
- 遷移ベクトル、local received up to、および remote received up to などのタイムスタンプがあります。

遷移同期

OpenText eDirectory での同期は遷移同期です。つまり、OpenText eDirectory でデータへの変更を同期する場合、eDirectory エージェントがレプリカリング内の他のすべてのエージェントに直接接続して変更を同期する必要はありません。

図 4-2 遷移同期



たとえば、Server 1 でデータに変更を加えた場合、その変更は Server 1 から Server 2、および Server 2 から Server 3 に同期されます。通信上の問題のため Server 1 が Server 3 に直接接続できなかった場合でも、Server 3 は Server 2 を介してデータへの最新の変更内容を受信します。Server 3 は、変更内容を受信したことを Server 2 に通知します。次に Server 2 は、Server 3 と Server 2 が同期していることを Server 1 に通知します。

オブジェクトトランザクションモデル

OpenText eDirectory での同期では、LDAP および X.500 準拠ディレクトリの標準であるオブジェクトトランザクションモデルが維持されます。オブジェクトトランザクションモデルでは、新しいトランザクションを同期する前に、以前のすべてのトランザクションを同期する必要があります。

たとえば、サーバ上でデータに D1、D2、および D3 という変更を加えたとします。ネットワーク障害のため、これらの変更は他のサーバ間で同期されません。サーバ上で別の D4 という変更を行った場合、D1、D2、および D3 がレプリカリング内のすべてのサーバ間で同期された後でのみ、D4 が同期されます。

遷移ベクトル

遷移ベクトルとは、レプリカのタイムスタンプのことです。レプリカ作成時刻を 1970 年 1 月 1 日からの経過秒数で表したものと、レプリカ番号、および現在のイベント番号を組にして表示されます。例 : s3D35F377 r02 e002

詳細については、474 ページの 「遷移ベクトルと復元後の検証処理」を参照してください。

Local Received Up To

Local Received Up To (LRUT) は、ローカルレプリカが変更内容を受信するまでの時間です。

詳細については、262 ページの 「ツリー内のオブジェクトの参照」を参照してください。

Remote Received Up To

Remote Received Up To (RRUT) は、リモートレプリカの LRUT です。

詳細については、262 ページの 「ツリー内のオブジェクトの参照」を参照してください。

通常同期またはレプリカ同期

通常同期またはレプリカ同期は、eDirectory における 2 つの同期プロセスの 1 つです。通常同期では、サーバ上のデータへの変更がすべて、レプリカリング内の他のサーバと同期されます。

通常同期は、同じパーティションを持つ任意のバージョンの eDirectory が実行されているすべてのサーバ間で行われます。

詳細については、164 ページの 「レプリカの管理」を参照してください。

通常同期を有効または無効にするには、OpenText iMonitor でアウトバウンド同期およびインバウンド同期を有効または無効にします。インバウンド同期およびアウトバウンド同期は両方とも、デフォルトでは有効になっています。通常同期を介して他のサーバ間でデータへの変更を同期するには、iMonitor で同期パラメータを設定する必要があります。詳細については、「257 ページの 「DS エージェントを制御および環境設定する」」を参照してください。

通常同期では、データに変更が加えた場合、変更内容はサーバ間で同期される前に、バッファに保存されます。サーバの同期ステータスは、iMonitor で表示できます。詳細については、「262 ページの 「ツリー内のオブジェクトの参照」」を参照してください。

通常同期は遷移同期であり、オブジェクトトランザクションモデルが維持されます。詳細については、101 ページの「遷移同期」および「オブジェクトトランザクションモデル」を参照してください。

通常同期を設定する

通常同期を設定するには、iMonitor の［エージェント同期］にある［エージェント環境設定］を使用します。

この節では、次のトピックについて説明します。

- 131 ページの 「通常同期を有効 / 無効にする」
- 131 ページの 「インラインキャッシュを有効 / 無効にする」
- 131 ページの 「同期スレッド」
- 131 ページの 「同期メソッド」

通常同期を有効 / 無効にする

通常同期を有効または無効にするには、iMonitor でアウトバウンド同期およびインバウンド同期を有効または無効にします。詳細については、「257 ページの 「DS エージェントを制御および環境設定する」」を参照してください。

アウトバウンド同期は、デフォルトで有効になっています。このオプションをサーバで無効にしている場合、このサーバ上のデータへの変更は、他のサーバと同期されません。アウトバウンド同期を無効にする期間 (単位は時間) を指定できます。デフォルト (最大時間) は 24 時間です。指定された期間が過ぎると、そのサーバでデータに加えられた変更が他のサーバと同期されるようになります。

インバウンド同期は、デフォルトで有効になっています。サーバに対してこのオプションを無効にすると、他のサーバでデータに加えられた変更がそのサーバと同期されなくなります。

インラインキャッシュを有効 / 無効にする

サーバのインラインキャッシュ変更を有効または無効にできます。インラインキャッシュ変更は、アウトバウンド同期が無効になっている場合のみ、無効にできます。アウトバウンド同期を有効にすると、インラインキャッシュ変更も有効になります。

インラインキャッシュ変更を無効にすると、このレプリカの変更キャッシュに無効のマークが付き、[**エージェント環境設定**] > [**パーティション**] に無効フラグが付きます。インラインキャッシュ変更を有効にすると、変更キャッシュの再構築時に、無効な変更キャッシュのフラグが削除されます。

同期スレッド

アウトバウンド同期を行うには、同期スレッドを設定する必要があります。iMonitor で、[**エージェント同期**] の下にある [**エージェント環境設定**] を使用して同期スレッド数を指定できます。有効な値は 1 ～ 16 です。詳細については、257 ページの 「DS エージェントを制御および環境設定する」を参照してください。

同期メソッド

通常、OpenText eDirectory では、レプリカおよびレプリケーションパートナーの数に基づいてメソッドが自動的に選択されます。同期メソッドは、次のとおりです。

- **パーティションごと** : データへの変更は、他のレプリカと同時に同期されます。変更の同期には複数のスレッドが使用されます。たとえば、レプリカ R1 のデータに D1、D2、および D3 という変更が加えられ、これらの変更をレプリカ R2 および R3 の間で同期させる必要がある場合、D1、D2、および D3 は R2 および R3 と同時に同期されます。
- **サーバごと** : データへの変更は順次に同期されます。変更の同期には 1 つのスレッドのみが使用されます。たとえば、レプリカ R1 のデータに D1、D2、および D3 という変更を加えたとします。これらの変更をレプリカ R2 および R3 の間で同期させる必要があります。まず、D1 が R2 および R3 と同期されます。次に、D2 が R2 および R3 と同期されます。

- **ダイナミック調整** : 割り当てたシステムリソースに基づいて、OpenText eDirectory によって同期メソッドが自動的に選択されます。

iMonitor で、［**エージェント同期**］の下の［**エージェント環境設定**］を使用して同期メソッドを指定できます。詳細については、257 ページの 「DS エージェントを制御および環境設定する」を参照してください。

注

- iMonitor には、同期方法としてサーバ別に同期を指定するオプションが用意されています。ただし、Skulker ではバックグラウンドプロセスを同期できないので、この方法は単一のサーバツリーでは推奨されません。
- eDirectory 9.2.2 以降では、Skulker はデータトランザクションが正常に完了した直後には同期を開始しません。すぐに Skulker で同期を実行すると、OpenText eDirectory の操作パフォーマンスに影響を与える場合があります。このため、OpenText eDirectory サーバ間のデータの同期は、デフォルトで 5 秒遅延します。必要に応じて、NDSD_CC_SKULK_DELAY 環境変数をエクスポートして、同期をさらに遅延できます。この変数の値は、次の例のように秒単位でのみ入力できます。

  ```
  NDSD_CC_SKULK_DELAY=<SECONDS>
  NDSD_CC_SKULK_DELAY=
  ```

  上記の環境変数が空白のままの場合、同期はデフォルトで 5 秒遅延します。

優先度同期

優先度同期は、OpenText eDirectory における 2 つの同期プロセスの 1 つです。優先度同期を使用すると、通常同期を待たずに重要なデータを同期することができます。

優先度同期は OpenText eDirectory での通常の同期プロセスを補うものです。通常同期とは異なり、優先度同期では、変更内容はサーバ間で同期される前にバッファに保存されません。そのため、優先度同期は通常同期より高速になります。

通常の同期を待てない場合は、優先度同期によって重要なデータを同期できます。優先度同期プロセスは通常の同期プロセスより高速です。優先度同期は、同じパーティションをホストしている 2 台以上の eDirectory 8.8 以降のサーバ間でのみサポートされます。

次の表に、優先度同期機能をサポートするプラットフォームを示します。

機能リスト	Linux	Windows
優先度同期	✓	✓

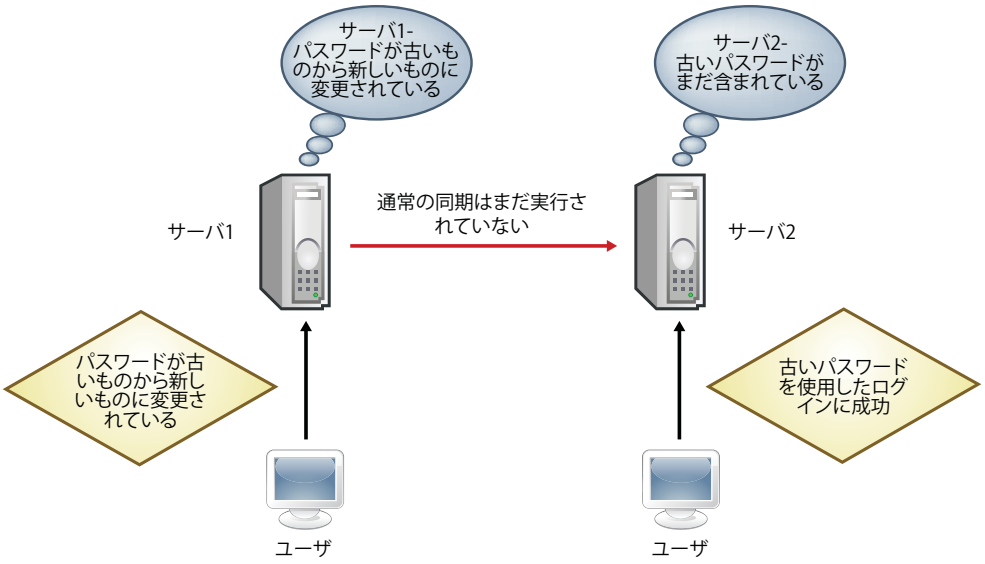
このセクションでは、次の情報を紹介します。

- 133 ページの 「優先度同期の必要性」
- 133 ページの 「優先度同期の使用」

優先度同期の必要性

通常の同期では時間がかかる場合がありますが、その間、変更されたデータは他のサーバで使用できません。たとえば、ディレクトリと通信している異なるアプリケーションがあるとします。サーバ 1 でパスワードを変更します。通常の同期では、この変更がサーバ 2 と同期されるまでしばらく時間がかかります。このため、ユーザはまだ古いパスワードを使用して、サーバ 2 と通信するアプリケーションを通してディレクトリへの認証を行うことができます。

図 4-3 優先度同期の必要性



大規模な展開においては、オブジェクトの重要なデータが変更されたときに、変更が直ちに同期される必要があります。優先度同期プロセスはこの問題を解決します。

優先度同期の使用

優先度同期を使用して日付の変更を同期するには、OpenText Identity Console を使用して優先度同期ポリシーをパーティションに適用する必要があります。

優先度同期は、デフォルトで有効になっています。詳細については、「134 ページの 「インバウンド優先度同期およびアウトバウンド優先度同期を有効 / 無効にする」」を参照してください。

優先度同期を介して重要なデータへの変更を同期するには、次の手順を実行します。

1. 優先度同期のスレッド数を指定します。

   詳細については、134 ページの 「優先度同期スレッド」を参照してください。

2. 優先度同期キューサイズを指定します。

   詳細については、135 ページの 「優先度同期キューサイズ」を参照してください。

3. 優先度同期ポリシーを作成して定義するには、優先度同期を介して同期する重要な属性を指定します。

   詳細については、137 ページの 「優先度同期ポリシーを作成および定義する」を参照してください。

4 優先度同期ポリシーを 1 つ以上のパーティションに適用します。

詳細については、138 ページの 「優先度同期ポリシーを適用する」を参照してください。

優先度同期プロセスでは、重要な属性への変更のみが同期されます。優先度同期では、オブジェクトトランザクションモデルが維持されます。したがって、重要ではないデータが変更され、まだ同期されていない場合、および重要なデータが同じエントリで変更された場合には、重要ではないデータが重要なデータとともに同期されます。

たとえば、ユーザが Income、Employee No、Address、Cube No という属性を持っているとします。その中の Income と Address が重要な属性です。Employee No および Cube No は変更されていますが、これらの変更はまだ同期されていません。Income および Address への変更が優先度同期を介して同期されると、Employee No および Cube No も (重要データとして指定されていませんが) 同期されます。

このセクションでは、次の情報について説明します。

- 134 ページの「インバウンド優先度同期およびアウトバウンド優先度同期を有効 / 無効にする」
- 134 ページの 「優先度同期スレッド」
- 135 ページの 「優先度同期キューサイズ」
- 135 ページの 「優先度同期ポリシーを管理する」
- 139 ページの 「優先度同期が失敗する場合」

インバウンド優先度同期およびアウトバウンド優先度同期を有効 / 無効にする

iMonitor を使って、OpenText eDirectory でのインバウンド / アウトバウンド優先度同期を有効または無効にすることができます。詳細については、「257 ページの 「DS エージェントを制御および環境設定する」」を参照してください。

インバウンド優先度同期は、デフォルトで有効になっています。インバウンド優先度同期をサーバで無効にしている場合、他のサーバ上の重要なデータへの変更は、優先度同期ではこのサーバと同期されません。ただし、変更は通常同期プロセスによって同期されます。

アウトバウンド優先度同期は、デフォルトで有効になっています。このオプションをサーバで無効にしている場合、このサーバ上の重要なデータへの変更は、優先度同期では他のサーバと同期されません。ただし、変更は通常同期プロセスによって同期されます。

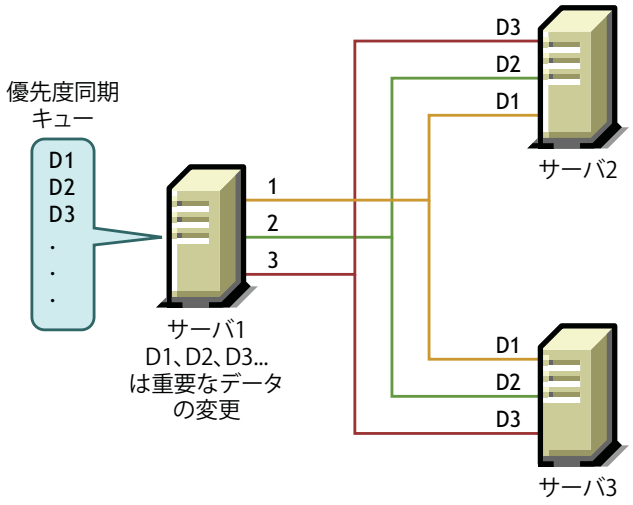
優先度同期スレッド

アウトバウンド優先度同期に使用するスレッド数を設定する必要があります。iMonitor で、[**エージェント同期**] の下にある [**エージェント環境設定**] を使用して、優先度同期スレッド数を指定できます。詳細については、257 ページの 「DS エージェントを制御および環境設定する」を参照してください。有効な値は 1 ～ 32 です。デフォルト値は 4 です。

優先度同期キューサイズ

これは、同期の前にキューで保持可能な、変更された重要なエントリの最大数です。重要なエントリは、変更されるとすぐに優先度同期キューに入れられ、順次に同期されます。たとえば、server1 で D1、D2、および D3 という重要なエントリが変更されており、これらのエントリを、優先度同期を介して server2 および server3 間で同期させる必要がある場合には、まず D1 が server2 および server3 と同期されます。次に D2 が server2 および server3 と同期され、その後、D3 が server2 および server3 と同期されます。キュー内の以前のエントリがサーバのいずれかと正常に同期していない場合でも、その他のエントリの同期には影響しません。

***図 4-4** 優先度同期キュー*



iMonitor で優先度同期キューサイズを指定するには、[**エージェント同期**] の下にある [**エージェント環境設定**] を使用します。詳細については、257 ページの 「DS エージェントを制御および環境設定する」を参照してください。

優先度同期プロセス中、多数の変更が短い間隔で行われ、キューが最大サイズに達した場合には、キューは期限切れになり、新しいキューが形成されます。まだ同期されていない古いキュー内の変更は、通常同期によって同期されます。

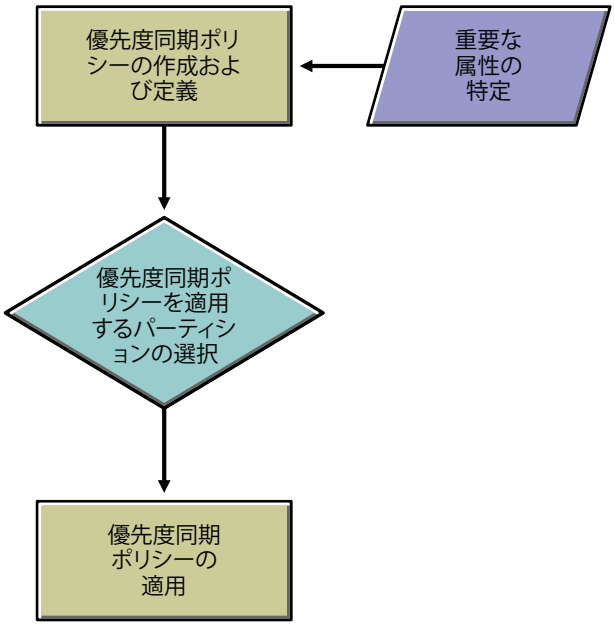
優先度同期のキューサイズは、0 から $2^{32} - 1$ までの範囲で変更できます。デフォルトでは、この値は $2^{32} - 1$ に設定されています。優先度同期キューサイズが 0 に設定されている場合、変更は優先度同期では同期されません。これらの変更は通常同期によって同期されます。

-1 を指定すると、キューサイズが無限大になります。-1 は $2^{32} - 1$ です。-3 などの負の値を指定した場合は、-3 = -1-2 になり、$2^{32} - 1-2$ になります。

優先度同期ポリシーを管理する

優先度同期の管理は、LDAP を使用して、ポリシーを作成および定義し、パーティションに適用することで行えます。優先度同期ポリシーを定義するには、重要な属性を指定します。

図4-5 優先度同期プロセス



たとえば、Password および Account Number という属性が重要である場合、これらの属性を含む優先度同期ポリシー PS1 を作成できます。その後、ポリシー PS1 をパーティション P1 に適用できます。サーバ上のエントリのパスワードまたはアカウント番号を変更した場合、その変更は、パーティション P1 を持つ他のサーバと直ちに同期されます。

優先度同期が実行されるようにするには、iMonitor でアウトバウンド優先度同期およびインバウンド優先度同期が有効になっていることを確認する必要があります。インバウンド優先度同期およびアウトバウンド優先度同期は、デフォルトで有効になっています。インバウンド優先度同期およびアウトバウンド優先度同期を無効にしている場合、データへの変更は通常同期によって同期されます。

詳細については、257 ページの 「DS エージェントを制御および環境設定する」を参照してください。

この節では、次のトピックについて説明します。

- 137 ページの 「優先度同期ポリシーを作成および定義する」
- 137 ページの 「優先度同期ポリシーを編集する」
- 138 ページの 「優先度同期ポリシーを適用する」
- 138 ページの 「優先度同期ポリシーを削除する」

子パーティションを作成した場合、親に適用されている優先度同期ポリシーが子パーティションに継承されます。パーティションをマージした場合、親の優先度同期ポリシーが保持されます。

優先度同期ポリシーを作成および定義する

優先度同期ポリシーの定義は、属性を直接選択するか、オブジェクトクラスを介して選択することで行えます。オブジェクトクラスを介して属性を選択する場合は、オブジェクトクラスの下にあるすべての属性が優先度同期に選択されます。優先度同期では必須属性またはオプション属性を選択することができます。

優先度同期ポリシーは、LDAP を使用して、OpenText eDirectory ツリー内の任意の場所で作成できます。

空の優先度同期ポリシーを作成するには、次のように指定します。

```
dn:cn=policy1,o=policies

changetype:add

objectclass:prsyncpolicy
```

優先度同期ポリシーを定義するには、優先度同期の属性をマークします。

```
dn:cn=policy2,o=policies

changetype:add

objectclass:prsyncpolicy

prsyncattributes:description
```

上の例では、Description が優先度同期としてマークされた属性です。

優先度同期ポリシーを編集する

優先度同期ポリシーオブジェクトは、LDAP を使用し編集できます。

次の例では、Description ではなく Surname を優先度同期としてマークすることで、優先度同期ポリシーが変更されています。

```
dn:cn=policy2,o=policies

changetype:modify

add:prsyncattributes

prsyncattributes:surname
```

優先度同期としてマークされている属性を優先度同期ポリシーから削除するには、次のように指定します。

```
dn:cn=policy2,o=policies

changetype:modify

add:prsyncattributes

prsyncattributes:description
```

上の例では、属性 Description が優先度同期ポリシーから削除されています。

優先度同期ポリシーを適用する

1 つの優先度同期ポリシーを多数のパーティションに適用できますが、複数のポリシーを 1 つのパーティションに適用することはできません。

優先度同期ポリシーは、LDAP を使用してパーティションに適用できます。

優先度同期ポリシーをルートパーティションに適用するには、次のように指定します。

```
dn:

changetype:modify

add:prsyncpolicydn

prsyncpolicydn:cn=policy2,o=policies
```

上の例では、policy2 がルートパーティションに適用されています。

優先度同期ポリシーを非ルートパーティションに適用するには、次のように指定します。

```
dn:o=org

changetype:modify

add:prsyncpolicydn

prsyncpolicydn:cn=policy2,o=policies
```

上の例では、policy2 が非ルートパーティションに適用されています。

非ルートパーティションの優先度同期ポリシーを置き換えるには、次のように指定します。

```
dn:o=org

changetype:modify

replace:prsyncpolicydn

prsyncpolicydn:cn=policy1,o=policies
```

上の例では、policy2 が policy1 に置き換えられています。

優先度同期ポリシーと非ルートパーティションとの関連付けを解除するには、次のように指定します。

```
dn:o=org

changetype:modify

delete:prsyncpolicydn
```

上の例では、優先度同期ポリシーと非ルートパーティション O=Org との関連付けが解除されています。

優先度同期ポリシーを削除する

優先度同期ポリシーは、LDAP を使用し削除できます。

```
dn:cn=policy1,o=policies

changetype:delete
```

注 : 優先度同期ポリシーの作成と管理の詳細については、386 ページの 「Linux での LDAP ツールの使用」および 175 ページの 「OpenText インポート / エクスポート変換ユーティリティ」を参照してください。

優先度同期が失敗する場合

優先度同期は、次のような状況下で失敗する可能性があります。

- ネットワーク障害 : ネットワークに障害が発生し、変更内容をリモートサーバに送信できない場合、優先度同期では変更内容が保存されません。
- 優先度同期キュー最大サイズに到達: エントリ数が優先度同期キューサイズを超える場合、優先度同期では、優先度同期キュー内の変更が無視されます。
- スキーマ同期の失敗 : スキーマが同期されない場合、優先度同期プロセスが失敗します。
- オブジェクトが他のサーバ上に存在しない: オブジェクトの作成そのものが同期されない場合、優先度同期は失敗します。
- レプリカリング内のサーバの混在 : eDirectory 8.8 と eDirectory 8.8 より前の両方のサーバがある場合、優先度同期が失敗します。

これらの理由のいずれかのために優先度同期が失敗した場合には、重要なデータへの変更は通常同期によって同期されます。

ポリシーベースのレプリケーション

OpenText eDirectory のレプリケーションは、デフォルトでメッシュトポロジに従います。つまり、レプリカリング内のすべてのレプリカが、相互にアウトバウンドおよびインバウンドになることができます。メッシュモデルはすべての環境に適しているとは限りません。*ポリシーベースのレプリケーション*を使用すると、管理者はレプリケーショントポロジを設定し、レプリケーショントラフィックを最適化することができます。

レプリケーショントポロジを設定するには、ポリシーファイルを作成し、1 つのファイルですべてのパーティションに対するポリシーを指定して、必要なサーバにそれをコピーします。

Linux の場合

XML 形式のポリシーファイルを作成し、それに selectivesync.xml という名前を付けて、nds.conf ファイルと一緒に配置します。

ポリシーの XML 定義のサンプルを次に示します。

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<SelectiveSync   xmlns="http://www.novell.com/nds"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.novell.com/nds
  file:/opt/novell/eDirectory/lib64/nds-schema/xsd/selectivesync.xsd"
config-version="0.1">

    <Partition DN=".novell.TREE.">

         <SourceServer DN=".server1.novell.TREE.">

                <SyncronizeTo>.server2.novell.TREE.</SyncronizeTo>

         </SourceServer>

         <SourceServer DN=".server2.novell.TREE.">

                <SyncronizeTo>.server3.novell.TREE.</SyncronizeTo>

         </SourceServer>

         <SourceServer DN=".server3.novell.TREE.">

                <SyncronizeTo>.server1.novell.TREE.</SyncronizeTo>

         </SourceServer>

    </Partition>

</SelectiveSync>
```

Windows の場合

XML 形式のポリシーファイルを作成し、それに selectivesync.xml という名前を付けて、インストール場所 (たとえば C:\Novell\NDS) に配置します。

ポリシーの XML 定義のサンプルを次に示します。

```
<?xml version="1.0" encoding="utf-8" ?>

<SelectiveSync   xmlns="http://www.novell.com/nds"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.novell.com/nds
C:\Novell\NDS\selectivesync.xsd" config-version="0.1">

    <Partition DN=".novell.TREE.">

         <SourceServer DN=".server1.novell.TREE.">

                <SyncronizeTo>.server2.novell.TREE.</SyncronizeTo>

         </SourceServer>

         <SourceServer DN=".server2.novell.TREE.">

                <SyncronizeTo>.server3.novell.TREE.</SyncronizeTo>

         </SourceServer>

         <SourceServer DN=".server3.novell.TREE.">

                <SyncronizeTo>.server1.novell.TREE.</SyncronizeTo>
```

```
        </SourceServer>
    </Partition>
</SelectiveSync>
```

Windows では、xsd パスを指定するときにファイルがないことに注意してください。

同期スレッドの手動設定

作成されるスレッドの最大数を設定することで、より多くのサーバに同時に複製されるよう、作成されるスレッド数を手動で増やすことができます。この設定は、サーバ上のすべてのパーティションに適用できます。

作成されるスレッドの最大数を設定するには、次の手順を実行します。

1. iMonitor にログインします。
2. ［**エージェント環境設定**］>［**エージェント同期**］に移動します。
3. オプションで、［**同期メソッド**］セクションの［**サーバごと**］を選択します。
4. ［**システム計測同期スレッド**］セクションで［**無効**］を選択します。
5. ［**最大手動設定同期スレッド**］セクションで、所要のスレッド数を設定します。

システム計測同期

システム計測同期では、次の 2 つの式によって Skulker スレッドの数が計算されます。

- **パーティションモードの場合** : Skulker スレッドの数 = そのサーバ上のパーティションの数
- **サーバモードの場合** : Skulker スレッドの数 =(サーバに認識されているサーバの数 +1)/2

ただし［**最大システム計測同期スレッド**］が無効になっている場合は、上記の 2 つの式は使用されません。代わりに［**最大手動設定同期スレッド**］が使用されます。

たとえば、5 つのサーバと 3 つのパーティションを使って環境がセットアップされているとします。ここで［**最大システム計測同期スレッド**］を有効にした場合、パーティションモードでは、最大で 3 つの Skulker スレッドがサーバによって作成される可能性があり、サーバモードでも、最大で 3 つの Skulker スレッドが作成される可能性があります。しかし、最大数の 3 つの Skulker スレッドが存在しても、1 つのサーバがすべてのパーティション上の他の 4 つのサーバに並列的に更新を送ることはできません。この場合は、［**最大システム計測同期スレッド**］を無効にして、［**最大手動設定同期スレッド**］で Skulker スレッドの数を増やします。

Skulker スレッドの最大数

［**最大手動設定同期スレッド**］を 12 に設定すれば、1 つのサーバはすべてのパーティションのすべてのサーバに更新を並行して送ることができます。ただし、このセットアップでは、サーバモードで 12 個を超える Skulker スレッド、パーティションモードで 3 個を超える Skulker スレッドを作成できません。たとえ［**最大手動設定同期スレッド**］を 12 より高い値に設定したとしてもです。

非同期アウトバウンド同期の設定

旧リリースの OpenText eDirectory では、1 つのサーバから別のサーバへのアウトバウンド同期が単一のスレッドで順次的に実行されたため、変更の複製に長時間かかりました。

OpenText eDirectory に備わっている 1 つのスレッドは、変更キャッシュを分析し、他のサーバに送るパケットを準備した後、パケットのキューを満たします。別のスレッドがパケットを取得し、それらを他のサーバに 1 つずつ送ります。これにより同期が最適化され、時間が短縮されます。

1 つのサーバから別のサーバへのアウトバウンド同期を設定するには、次の手順を実行します。

1 iMonitor にログインします。

2 [**エージェント環境設定**] > [**バックグラウンドプロセスの設定**] の順に選択します。

3 [**非同期アウトバウンド同期設定**] セクションで、[**有効**] を選択します。

注 : 非同期アウトバウンド同期を有効にすると、受信側サーバで CPU と I/O の使用率が増加する可能性があります。これを回避するには、[**非同期ディスパッチャスレッド遅延**] で遅延時間を指定することにより、パケットを送信するときの遅延を設定できます。この遅延時間として 0 ～ 999 ミリ秒の値を設定できます。デフォルト値は 0 ミリ秒です。

バックグラウンドプロセスの設定

次の設定を使用して、Skulker、パージャ、および破損通知のバックグラウンドプロセスの速度を制御できます。

- CPU は、同じプロセス (Skulker、パージャ、または破損通知) の実行と実行の間の遅延時間とコンピュータリソースの最大使用率を指定します。
- ハード制限は、各 Skulker、Purger、および破損通知プロセスの静的遅延設定を指定します。

バックグラウンドプロセスを設定する方法については、261 ページの 「バックグラウンドプロセスの設定」を参照してください。

ハード制限ポリシー

ハード制限ポリシーはデフォルトで有効です。バックグラウンドプロセスは、特定数のオブジェクトを処理した後、100 ミリ秒 (デフォルト値) にわたってスリープ状態になります。遅延 (スリープ) 時間を短くして、システムのパフォーマンスを向上させることができます。遅延が 0 ミリ秒に近く、これらのプロセスの 1 つ以上がバックグラウンドで実行されている場合には、CPU 使用率を増やすことができます。これを監視し、必要に応じて調整する必要があります。

CPU ベースの動的ポリシー

CPU ベースのポリシーにより、システムで次の 3 つのバックグラウンドプロセスの遅延を動的に調整して、CPU 最大使用率を制限できます。

- 変更キャッシュ処理遅延 (アウトバウンド同期の一部)
- ObitProc 遅延 (破損通知処理)
- パージャ遅延 (変更キャッシュのプルーニング)

CPU 使用率が設定されたレベルに自動的に制限されます。クライアントの負荷が高い場合はバックグラウンドプロセスが遅くなり、クライアントの負荷が減少するとバックグラウンドプロセスの速度が上がります。バックグラウンドプロセスを遅くすべきでない場合は、このポリシーでスリープ時間を短くすることにより、最大遅延限度を設定できます。ただし、スリープ時間を小さく設定すると、CPU 制限の違反が発生する可能性があります。

バックグラウンドプロセスの間隔

次のバックグラウンドプロセスの間隔値を設定できます。

- バックリンク /DRL 間隔
- クリーン間隔
- アウトバウンド同期間隔
- スキーマ同期間隔
- ジャニタ間隔
- パージャ間隔

バックグラウンドプロセスの間隔を設定するには、次の操作を行います。

1 iMonitor にログインします。

2 [**エージェント環境設定**] > [**バックグラウンドプロセスの設定**] の順に選択します。

3 [**バックグラウンドプロセス間隔**] セクションで、間隔の値を指定します。

5 スキーマの管理

OpenText eDirectory ツリーのスキーマは、このツリーに含めることができるオブジェクト (ユーザ、グループ、プリンタなど) のクラスを定義します。スキーマによって、各オブジェクトタイプを構成する属性 (プロパティ) が指定されます。属性には、オブジェクトの作成に不可欠な必須属性と、必要に応じて指定できるオプション属性があります。

OpenText eDirectory オブジェクトはそれぞれオブジェクトクラスに属し、オブジェクトクラスはオブジェクトに関連付けることのできる属性を指定します。すべての属性は一連の属性タイプに基づくもので、属性タイプもまた、一連の標準的な属性構文に基づいています。

OpenText eDirectory スキーマは、各オブジェクトの構造を制御するだけではなく、OpenText eDirectory ツリー内でのオブジェクト間の関係も制御します。スキーマルールを設定すると、オブジェクトは他のサブオーディネートオブジェクトを含むことができます。このように、スキーマによって OpenText eDirectory ツリーの構造が決まります。

組織が必要とする情報の変化に応じて、スキーマに変更を加える必要が出てくることがあります。たとえば、ユーザオブジェクトに、以前は FAX 番号が不要であっても、現在は必要であるとします。この場合、FAX 番号を必須属性とした新しいユーザクラスを作成し、ユーザオブジェクトの作成に、この新しいユーザクラスを使用できます。

ツリーに対してスーパバイザ権を持つユーザは、OpenText Identity Console のスキーマ管理の役割により、ツリーのスキーマをカスタマイズして次のようなタスクを実行することができます。

- スキーマ内のすべてのクラスおよび属性の一覧を表示する。
- 既存のスキーマにクラスまたは属性を追加して、スキーマを拡張する。
- クラスを作成する。名前を付けてから、属性、フラグ、追加先コンテナ、および属性の継承元のペアレントクラスを指定することにより行います。
- 名前を付けてから、構文およびフラグを設定して、属性を作成する。
- 既存クラスへ属性を追加する。
- 使用されていない、あるいは古くなったクラスまたは属性を削除する。
- 潜在的な問題を発見および解決する。

この章では次のトピックについての情報を説明します。

- 146 ページの 「スキーマの拡張」
- 150 ページの 「スキーマの表示」
- 151 ページの 「手動でスキーマを拡張する」
- 154 ページの 「バージョン 8.7 以降に追加されたスキーマフラグ」
- 155 ページの 「クライアントを使用してスキーマ操作を実行する」

スキーマ情報の詳細については、『*eDirectory Schema Reference (OpenText eDirectory スキーマリファレンス)* (https://www.novell.com/documentation/developer/ndslib/schm_enu/data/h4q1mn1i.html)』を参照してください。

スキーマの拡張

新しいクラスまたは属性を作成することにより、ツリーのスキーマを拡張できます。OpenText eDirectory ツリーのスキーマを拡張するには、ツリー全体に対するスーパバイザ権が必要です。

次の作業により、スキーマを拡張できます。

- クラスを作成する
- クラスを削除する
- 属性を作成する
- クラスへオプション属性を追加する
- 属性を削除する

次の作業により、補助属性のスキーマを拡張できます。

- 補助クラスを作成する
- 補助クラスのプロパティでオブジェクトを拡張する
- オブジェクトの補助プロパティを変更する
- オブジェクトから補助プロパティを削除する

クラスを作成する

組織の必要条件の変化に応じて、既存のスキーマに対しクラスを追加できます。

1 OpenText Identity Console のホームページで、[**スキーマ管理**] タイルをクリックします。

2 [**クラス**] > [**クラスの作成** ＋] の順にクリックします。

3 属性の作成ウィザードの指示に従って、オブジェクトクラスを定義します。

ウィザードの各段階で、[ヘルプ] が利用できます。詳細については、ヘルプアイコン ( ) をクリックしてください。

オブジェクトクラスに追加するカスタムプロパティを定義する場合は、ウィザードを終了し、最初にカスタムプロパティを定義します。詳細については、属性を作成するを参照してください。

クラスを削除する

使用されていないクラスは、そのクラスが OpenText eDirectory ツリーのベーススキーマの一部でない限り、削除できます。OpenText Identity では、ローカルにレプリカ作成されたパーティションで現在使用されているクラスだけは削除できません。

次のような場合に、スキーマからクラスを削除できます。

- 2 つのツリーをマージし、クラスの違いを解決した場合
- 特定のクラスが不要になった場合

クラスを削除するには、次を実行します。

1. OpenText Identity Console のホームページで、[**スキーマ管理**] タイルをクリックします。
2. [**クラス**] をクリックします。
3. 削除するクラスを選択します。

   削除可能なクラスのみが表示されます。
4. **削除**をクリックします。

属性を作成する

属性のカスタムタイプを独自に定義し、それをオプション属性として既存のオブジェクトクラスに追加できます。ただし、既存のクラスに必須属性を追加することはできません。

注 : OpenText eDirectory では、レプリケーションの問題のため、ストリーム属性タイプ以外の属性に 60KB または 30,000 文字を超える値を含めることができません。ユーザやアプリケーションが、その制限を超える文字列またはバイナリ属性の値を設定すると、OpenText eDirectory は値が長すぎることを示す -649 エラーを返します。

新しい属性を作成するには、次の操作を行います。

1. OpenText Identity Console のホームページで、[**スキーマ管理**] タイルをクリックします。
2. [**属性**] > [**属性の作成** ＋] の順にクリックします。
3. 属性の作成ウィザードの指示に従って、新しい属性を定義します。

   ウィザードの各段階で、[ヘルプ] が利用できます。詳細については、ヘルプアイコン ( ) をクリックしてください。

クラスへオプション属性を追加する

既存のクラスにオプションの属性を追加できます。これは、次のような場合に必要になります。

- 組織の必要とする情報が変化した場合
- ツリーのマージを準備している場合

注 : 必須属性は、クラスの作成時にのみ定義できます。

オプション属性クラスを追加するには、次を実行します。

1 OpenText Identity Console のホームページで、[**スキーマ管理**] をクリックします。

2 [**クラス**] をクリックして、[**クラスの選択**] リストからクラスを選択します。

3 [**クラス情報**] ページで、[**属性の追加** ＋] をクリックします。

4 [**属性の選択**] ページで、属性を選択し、[**追加**] をクリックします。

属性がクラスに追加されます。

5 [**保存**] をクリックし、[**OK**] をクリックします。

誤って属性を追加した場合、または追加した属性を削除する場合は、[**これらのオプション属性を追加**] リストで属性を選択して、🗑 をクリックし、追加する属性のリストから削除します。

このクラスにオブジェクトを作成すると、ここで追加したプロパティを含むオブジェクトが作成されます。追加したプロパティの値を設定するには、オブジェクトの [その他] 一般プロパティページを使用します。

ヒント : 既存のクラスを変更するには、このページを使用して [**Current Attributes**] リストに追加します。削除できる属性は、[**OK**] をクリックする前に追加した属性のみです。前に追加した属性や保存した属性は削除できません。

属性を削除する

使用されていない属性は、その属性が OpenText eDirectory ツリーのベーススキーマの一部でない限り、削除できます。

次のような場合に、スキーマから属性を削除できます。

- 2 つのツリーをマージし、属性の違いを解決した場合
- 特定の属性が不要になった場合

属性を削除するには、次を実行します。

1 OpenText Identity Console のホームページで、[**スキーマ管理**] タイルをクリックします。

2 [**Attributes (属性)**] をクリックします。

3 削除する属性を選択します。

削除可能な属性のみが表示されます。

4 **削除** 🗑 をクリックします。

補助クラスを作成する

補助クラスとは、あるオブジェクトクラス全体ではなく、特定の OpenText eDirectory オブジェクトインスタンスに追加される一連のプロパティ (属性) です。たとえば、電子メールアプリケーションの場合、電子メールプロパティという補助クラスを含むように OpenText eDirectory ツリーのスキーマを拡張し、必要に応じてこれらのプロパティを使用して個々のオブジェクトを拡張できます。

スキーママネージャを使用すると、独自の補助クラスを定義できます。補助クラスで定義したプロパティを使用して、個別のオブジェクトを拡張できます。

補助クラスを作成するには、次を実行します。

1 OpenText Identity Console で、[**役割とタスク**] ボタン () をクリックします。

2 [**スキーマ**] > [**クラスの作成**] の順にクリックします。

3 クラス名と (オプションの)ASN1 ID を指定し、[**次へ**] をクリックします。

4 クラスフラグを設定する場合は、[**補助クラス**] を選択して [**次へ**] をクリックします。

5 クラスの作成ウィザードの指示に従って、新しい補助クラスを定義します。

ウィザードの各段階で、[ヘルプ] が利用できます。

補助クラスのプロパティでオブジェクトを拡張する

1 OpenText Identity Console のホームページで、[**スキーマ管理**] タイルをクリックします。

2 [**オブジェクトの拡張**] > [**オブジェクト名** ] の順にクリックします。

3 ユーザまたはグループを選択し、[**現在の補助クラス拡張** ] をクリックします。

「オブジェクトが正常に拡張されました」というメッセージが表示されます。

4 [**OK**] をクリックします。

5 使用する補助クラスが [**現在の補助クラスの拡張**] の下にすでに表示されているかどうかに応じて、適切な操作を実行します。

補助クラスがリストに表示されている	アクション
○	この手順を終了します。代わりに実行する手順については、150 ページの 「オブジェクトの補助プロパティを変更する」を参照してください。
いいえ	[**追加**] をクリックし、補助クラスを選択して [**OK**] をクリックします。

6 **閉じる**をクリックします。

オブジェクトの補助プロパティを変更する

1 OpenText Identity Console のホームページで、[**オブジェクトの管理 (Obejct Management)**] をクリックします。

2 [**オブジェクト**] リストで、オブジェクトを選択して、[**オブジェクトの変更**] を選択します。

3 変更するオブジェクトの名前およびコンテキストを指定して、[**保存**] をクリックします。

4 表示された画面で、必要な属性値を設定します。

- 値のない属性をクリックして、値がある属性のリストに追加します。
- 値がある属性を選択し、[**変更**] をクリックして属性を編集するか、[**削除**] をクリックして属性を削除します。
- 正しく設定するためには、各プロパティの構文について理解している必要があります。詳細については、『*OpenText eDirectory Schema Reference (OpenText eDirectory スキーマリファレンス)* (http://developer.novell.com/documentation/ndslib/schm_enu/data/h4q1mn1i.html)』を参照してください。

5 [**保存**] > [**OK**] の順にクリックします。

オブジェクトから補助プロパティを削除する

1 OpenText Identity Console のホームページで、[**スキーマ管理**] タイルをクリックします。

2 [**オブジェクトの拡張**] をクリックします。

3 現在の補助クラス拡張のリストで、プロパティを削除する [**現在の補助クラス拡張**] を選択し、[**削除**] をクリックします。

補助クラスが正常に削除されました。

4 [**OK**] をクリックします。

これにより、オブジェクトに最初から定義されていたプロパティを除き、補助クラスによって追加されたすべてのプロパティが削除されます。

5 **閉じる**をクリックします。

スキーマの表示

スキーマが組織の情報のニーズに合ったものかどうかを評価するために、スキーマを表示したり印刷することができます。組織が大きく、また複雑になると、スキーマをカスタマイズする必要も大きくなります。しかし、小規模な組織でも、特別な記録を必要とする場合があるかもしれません。このような場合、スキーマの表示や印刷は、ベーススキーマにどのような拡張が必要かを決めるのに役立ちます。

クラス情報を参照する

OpenText Identity Console の［クラスの情報］ページでは、選択したクラスに関する情報が表示され、属性を追加できます。このページに表示されているほとんどの情報は、クラスが作成されたときに指定されたものです。オプション属性の中には、後で追加されたものもあります。

クラスの作成中に、そのクラスを他のクラスの属性を継承するように指定した場合は、継承された属性はペアレントクラスの中に分類されます。たとえば、オブジェクトクラスがペアレントクラスで必須属性である場合、このページには、選択されたクラスの必須属性として表示されます。

1 OpenText Identity Console のホームページで、［**スキーマ管理**］タイルをクリックします。
2 ［**クラス**］をクリックして、情報を確認したい［**クラス**］を選択します。
詳細情報を表示するには、? をクリックします。

属性情報を表示する

1 OpenText Identity Console のホームページで、［**スキーマ管理**］タイルをクリックします。
2 ［**属性**］をクリックし、情報を確認したい［**属性**］を選択します。
詳細情報を表示するには、? をクリックします。

手動でスキーマを拡張する

.sch 拡張子の付いたファイルを使用して、手動で OpenText eDirectory スキーマを拡張できます。

このセクションでは、次のことを説明します。

- 151 ページの 「Windows でスキーマを拡張する」
- 152 ページの 「Linux でスキーマを拡張する」

Windows でスキーマを拡張する

Windows サーバのスキーマを拡張するには、NDSCons.exe を使用します。OpenText eDirectory に付属しているスキーマファイル (*.sch) は、デフォルトで C:\Novell\NDS ディレクトリにインストールされています。

1 ［**スタート**］>［**設定**］>［**コントロールパネル**］>［**OpenText eDirectory サービス**］の順にクリックします。
2 **install.dlm** をクリックし、［**開始**］をクリックします。
3 ［**追加のスキーマファイルのインストール**］をクリックし、［**次へ**］をクリックします。
4 管理権を持つユーザとしてログインし、［**OK**］をクリックします。

5 スキーマファイルのパスと名前を指定します。

6 完了をクリックします。

Linux でスキーマを拡張する

次のセクションでは、Linux コンピュータでのスキーマ拡張について説明します。

- 152 ページの 「ndssch ユーティリティを使用して、Linux でスキーマを拡張する」
- 153 ページの 「RFC 2307 スキーマを拡張する」

ndssch ユーティリティを使用して、Linux でスキーマを拡張する

OpenText Identity Console のほかにも、OpenText eDirectory スキーマ拡張ユーティリティの ndssch を使用して、Linux コンピュータ上のスキーマを拡張することができます。ツリーのスキーマの変更には、スキーマファイル (.sch) に指定された属性とクラスが使用されます。.sch ファイルで指定した内容に従って、属性とクラスの関連付けが作成されます。

使用する構文は次のとおりです。

```
ndssch [-h hostname[:port]] [-t tree_name] [-F <logfile>] admin-FDN
schemafile...
```

```
ndssch [-h hostname[:port]] [-t tree_name] [-d] admin_FDN schemafile
[schema_description]...
```

ndssch のパラメータ	説明
-h *hostname*	スキーマを拡張するサーバの名前または IP アドレス。指定したサーバが属しているツリーのスキーマが拡張されます。スキーマを拡張するホスト上にツリーがある場合、このパラメータの指定はオプションです。それ以外の場合、このパラメータの指定は必須です。
port	サーバポート。
-t *tree_name*	スキーマを拡張するツリーの名前。このパラメータの指定は任意です。/etc/opt/novell/eDirectory/conf/nds.conf ファイルに指定された値が、デフォルトのツリー名になります。詳細については、『*OpenText™ eDirectory CE 24.4 (v9.3) インストールガイド*』の「「環境設定パラメータ」」を参照してください。
-F *logfile*	ndssch ログファイルのパス名を指定します。
admin-FDN	ツリーに対する OpenText eDirectory 管理権を持つユーザのフルコンテキスト付きの名前。
schemafile	拡張するスキーマについての情報が入力されたファイルの名前。
-d, *schema_description*	このオプションが使用されている場合、各スキーマファイルにはスキーマファイルの説明が付属しています。

RFC 2307 スキーマを拡張する

RFC 2307 に定義されている属性とオブジェクトクラスは、ユーザまたはグループ関連、および NIS 関連のものです。(http://www.ietf.org/rfc/rfc2307.txt) ユーザまたはグループ関連の定義は、/opt/novell/eDirectory/lib/nds-modules/schema/rfc2307-usergroup.sch ファイルにコンパイルされます。NIS 関連の定義は、/opt/novell/eDirectory/lib/nds-modules/schema/rfc2307-nis.sch ファイルにコンパイルされます。それぞれに対応する LDIF 形式のファイルもあります (ユーザ / グループ関連は /opt/novell/eDirectory/lib/nds-modules/schema/rfc2307-usergroup.ldif、NIS 関連は /opt/novell/eDirectory/lib/nds-modules/schema/rfc2307-nis.ldif)。

RFC 2307 スキーマを拡張するには、ndssch ユーティリティまたは ldapmodify ツールを使用します。

- 153 ページの 「ndssch ユーティリティを使用する」
- 153 ページの 「ldapmodify ユーティリティを使用する」

ndssch ユーティリティを使用する

次のいずれかのコマンドを入力します。

ndssch -t tree_name admin-FDN /opt/novell/eDirectory/lib/nds-schema/rfc2307-usergroup.sch

または

ndssch -t tree_name admin-FDN /opt/novell/eDirectory/lib/nds-schema/rfc2307-nis.sch

パラメータ	説明
-t	スキーマを拡張するツリーの名前。このパラメータの指定は任意です。このパラメータが指定されていない場合、/etc/opt/novell/eDirectory/conf/nds.conf ファイルに指定されたツリー名が使用されます。

ldapmodify ユーティリティを使用する

次のいずれかのコマンドを入力します。

ldapmodify -h -D -w -f /opt/novell/eDirectory/lib/nds-schema/rfc2307-usergroup.ldif

または

ldapmodify -h -D -w -f /opt/novell/eDirectory/lib/nds-schema/rfc2307-nis.ldif

パラメータ	説明
-h *ldaphost*	LDAP サーバの実行場所となっている代替ホストを指定します。
-D *binddn*	*binddn* を使用して X.500 ディレクトリにバインドします。RFC 1779 に定義されている文字列表現の DN を指定します。
-w *passwd*	簡易認証のパスワードとして、*passwd* を使用します。
-f *file*	エントリ変更情報を標準入力ではなくファイルから読み込みます。

バージョン 8.7 以降に追加されたスキーマフラグ

READ_FILTERED および BOTH_MANAGED スキーマフラグが、バージョン 8.7 以降に追加されました。

READ_FILTERED は、属性が LDAP オペレーショナル属性であることを示すために使用されます。LDAP がこのフラグを使用するのは、スキーマを読み込んで属性が「オペレーショナル」であることを知らせる必要がある場合です。内部定義のスキーマ属性には、このフラグが設定されたものもあります。LDAP「オペレーショナル」定義には、3 つのスキーマフラグがあります。新しい READ_FILTERED フラグとは別に、既存のフラグとして「オペレーショナル」を示すものには READ_ONLY フラグと HIDDEN フラグがあります。これらのフラグのいずれかがスキーマ定義に存在する場合、LDAP は属性を「オペレーショナル」として扱い、特に必要がない場合はその属性を返しません。

BOTH_MANAGED は、セキュリティ権利を強制する新しいメカニズムです。これは識別名構文の属性にのみ重要なフラグです。この属性に設定された場合、要求している接続には、ターゲットオブジェクトと属性、およびターゲット属性によって参照されているオブジェクトの両方に対する権利が必要になります。これは、現在の WRITE_MANAGED フラグの機能を拡張したものです。このフラグは、現在ベーススキーマ属性には設定されていません。この新しいセキュリティ動作は OpenText eDirectory 8.7.*x* 以降のサーバ上でのみ実行されるため、このフラグに関連する動作に矛盾がないようにするには、ツリー全体を OpenText eDirectory 8.7 以降にアップグレードする必要があります。

8.7.*x* 以降のサーバのみがこれらの新しいフラグを認識するため、ルートパーティションのコピーを格納するバージョン 8.7.*x* 以降のサーバによってのみ、これらのフラグをスキーマ定義に設定できます。これは、ルートを持つサーバのみがスキーマを変更できるためです。通常インストールの新規サーバや、ルートパーティションを保持していない既存サーバをアップグレードしたサーバでは、これらの新しいフラグをツリー内のスキーマに追加することはできません。

これらの新しい機能のいずれかをツリーで有効にする場合は、スキーマが正常に拡張されていて、これらの新しいフラグを追加できることを確認する必要があります。確認する方法には次の 2 つがあります。1 つは、ルートパーティションの書き込み可能なコピーを持つサーバを選択して、バージョン 8.7 以降にアップグレードすることです。これにより、自動的にスキーマが正しく拡張され、新しいフラグに対応します。

2 つ目の方法はさらに込み入ったものです。次の手順に従って操作します。

1. 新しく 8.7.*x* 以降のサーバをインストールするか、ツリー内の既存のサーバをアップグレードします。このサーバが [Root] のコピーを保持している必要はありません。
2. ルートパーティションのコピーをこの新しいサーバに手動で追加します。
3. 次に示す適切なスキーマ拡張ファイルをこのサーバ上で再実行し、スキーマを拡張します。

Platform	指示
Windows	install.dlm をロードし、[**追加のスキーマファイルのインストール**] をクリックします。
Linux	ndssch ユーティリティを使用します。詳細については、152 ページの 「ndssch ユーティリティを使用して、Linux でスキーマを拡張する」を参照してください。

4 これらの新しいフラグが設定された新しいスキーマファイルをインストールします。

5 (オプション) スキーマの同期後は、このサーバからルートレプリカを削除できます。

注 : これらの新しいスキーマフラグにより、オプションの機能が有効になります。新しい機能を必要としない場合は、スキーマ定義にこれらの新しいフラグが存在しなくても、ツリー内での OpenText eDirectory の通常操作に問題が起こることはありません。
READ_FILTERED フラグの場合は、属性の定義によっては表示されないことがあります。このため、オブジェクトのすべての属性に対する LDAP 読み込み要求によって、フラグが存在すれば受け取らないはずのデータが余分に取得される場合があります。
READ_FILTERED フラグを含む属性の中には、READ_ONLY フラグまたは HIDDEN フラグが存在するために、やはりオペレーショナルとして扱われるものもあります。
BOTH_MANAGED フラグは、すべてのサーバがアップグレードされたツリーでのみ有効になります。その環境の中でのみ、この機能を矛盾なく操作することが可能なためです。

クライアントを使用してスキーマ操作を実行する

eMBox (OpenText eDirectory Management Toolbox) クライアントはコマンドライン Java クライアントで、これを使用すると DSSchema 操作にリモートでアクセスできます。DSSchema eMTool を使用すると、スキーマの同期、リモートスキーマのインポート、新規スキーマエポックの宣言、ローカルスキーマのリセット、グローバルスキーマの更新などを実行できます (通常、DSRepair を使用して実行する操作です。詳細については、358 ページの 「スキーマの保守」を参照してください)。

emboxclient.jar ファイルは、OpenText eDirectory の一部としてサーバにインストールされます。JVM をインストールしていれば、どのコンピュータでも実行できます。クライアントの詳細については、618 ページの 「コマンドラインクライアントの使用」を参照してください。

DSSchema eMTool を使用する

1 コマンドラインで次のように入力して、対話式モードでクライアントを実行します。

```
java -cp path_to_the_file/emboxclient.jar  -i
```

(クラスパスに emboxclient.jar ファイルがすでに含まれている場合は、単に「java -i」と入力します。)

クライアントのプロンプトが次のように表示されます。

```
Client>
```

2 修復するサーバにログインするには、次のように入力します。

```
login -sserver_name_or_IP_address -pport_number
-uusername.context -wpassword -n
```

ポート番号は通常 80 または 8028 です。ただし、すでにそのポートを使用している Web サーバが存在する場合は異なります。-n オプションを使用すると、非セキュア接続が開始されます。

クライアントにログインが成功したかどうかが表示されます。

3 次の構文を使用して修復コマンドを入力します。

dsschema.*task options*

次に例を示します。

「dsschema.rst」と入力すると、このサーバのスキーマを同期するようにツリーのルートのマスタレプリカに要求することができます。

「dsschema.irs -n *MyTree*」と入力すると、MyTree というツリーからリモートスキーマをインポートすることができます。

各スイッチの間にはスペースが必要です。スイッチの順序は重要ではありません。

クライアントは修復が成功したかどうかを表示します。

DSSchema eMTool オプションの詳細については、156 ページの 「DSSchema eMTool オプション」を参照してください。

4 クライアントからログアウトするには、次のコマンドを入力します。

```
logout
```

5 クライアントを終了するには、次のコマンドを入力します。

```
exit
```

DSSchema eMTool オプション

次の表に、DSSchema eMTool オプションを示します。クライアントで list -t dsschema コマンドを使用して、DSSchema オプションの詳細を表示することもできます。詳細については、622 ページの 「eMTool とそのサービスを表示する」を参照してください。

オプション	説明
rst	ツリーのルートのマスタレプリカのスキーマをこのサーバに同期します。
irs -n*tree_name*	別のツリーからリモートスキーマをインポートします。
dse	ルートのマスタレプリカを持つサーバ上で新規スキーマエポックを宣言します。
rls	ローカルスキーマを、ルートパーティションのマスタレプリカがあるサーバからのコピーでリセットします。
gsu	グローバルスキーマ更新を実行します。
scc	ドメインクラスのスキーマサーキュラ包含ルールを追加します。

6 パーティションおよびレプリカの管理

パーティションは、OpenText eDirectory ツリー内の個別のデータユニットを構成する OpenText eDirectory データベースの論理区分です。システム管理者は、パーティションを利用して OpenText eDirectory 情報を格納したり、複製したりします。各パーティションは、コンテナオブジェクト、コンテナオブジェクトに含まれるすべてのオブジェクト、およびこれらのオブジェクトについての情報で構成されます。パーティションには、ファイルシステムに関する情報、またはパーティションに含まれるディレクトリやファイルに関する情報はありません。

各サーバに OpenText eDirectory データベース全体のコピーを保存する代わりに、OpenText eDirectory パーティションのコピーを作成してそれをネットワーク内の複数のサーバ上で保存できます。パーティションのコピーは、レプリカと呼ばれています。任意の数の OpenText eDirectory パーティションのレプリカを作成し、任意のサーバに保管できます。レプリカのタイプには、マスタ、読み書き可能、読み込み専用、サブオーディネートリファレンス、フィルタ済み読み書き可能、およびフィルタ済み読み込み専用があります。

次の表で、レプリカタイプについて説明します。

レプリカ	説明
マスタ、読み書き可能、および読み込み専用	特定のパーティションのすべてのオブジェクトおよび属性が含まれます。
サブオーディネートリファレンス	ツリーの接続のために使用されます。
フィルタ済みレプリカ	パーティション全体から取得した情報のサブセットが含まれます。このサブセットは、サーバのレプリケーションフィルタによって定義された必要なクラスおよび属性のみで構成されます。レプリケーションフィルタは、インバウンド同期やローカルでの変更時にレプリカに含めることのできるクラスおよび属性を識別するために使用されます。 フィルタ済みレプリカによって、管理者はまばらで断片的なレプリカを作成できます。 ◆ 指定したオブジェクトクラスだけが含まれるスパースレプリカ ◆ 指定した属性だけが含まれる断片レプリカ フィルタ済みレプリカの機能により、OpenText eDirectory に保存されているデータをアプリケーションがすばやく取得できるようになります。また、フィルタ済みレプリカを使用すると、1 つのサーバにより多くのレプリカを格納できます。

レプリカ	説明
読み書き可能フィルタ済みレプリカ	サーバのレプリケーションフィルタのサブセットであるクラスおよび属性をローカルで変更できます。ただし、これらのレプリカを作成できるのは、レプリケーションフィルタ内にそのクラスの必須属性がすべて含まれている場合のみです。
読み込み専用フィルタ済みレプリカ	ローカルで変更できません。

この章では、パーティションおよびレプリカの管理方法を説明します。

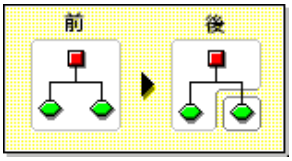
- 160 ページの 「パーティションの作成」
- 161 ページの 「パーティションのマージ」
- 162 ページの 「パーティションの移動」
- 164 ページの 「パーティションの作成操作またはマージ操作のキャンセル」
- 164 ページの 「レプリカの管理」
- 168 ページの 「フィルタ済みレプリカを設定し管理する」
- 171 ページの 「パーティションおよびレプリカを表示する」

パーティションの作成

パーティションを作成すると、ツリーの論理区分が作成されます。これらの論理区分は、ネットワーク内にある別の OpenText eDirectory サーバ間で複製することも配布することもできます。

新しいパーティションを作成すると、ペアレントパーティションが分割されて 2 つのパーティションになります。新しいパーティションは、次の図で示されるように、チャイルドパーティションになります。

***図 6-1** パーティションの分割前と分割後*



たとえば、1 つの部門を選択し、これを新しいパーティションとして作成すると、選択した部門およびそのサブオーディネートオブジェクトすべてがペアレントパーティションから分割されます。

選択した部門は、新しいパーティションのルートになります。新しいパーティションのレプリカは、ペアレントパーティションのレプリカと同じサーバに存在します。また、新しいパーティションのオブジェクトは、そのパーティションのルートオブジェクトに属します。

レプリカすべてを新しいパーティション情報と同期する必要があるため、パーティションの作成には時間がかかる場合があります。パーティションの作成中に別のパーティション操作を実行しようとすると、パーティションが使用中であることを示すメッセージが表示されます。

新しいパーティションのレプリカリストを参照し、リスト内のレプリカがすべてオンの状態であれば、操作が完了していることがわかります。状態は自動的にはリフレッシュされないので、画面を定期的に手動でリフレッシュします。

パーティションを作成するには、次の操作を行います。

1 OpenText Identity Console のホームページで、[**パーティション管理**] タイルをクリックします。

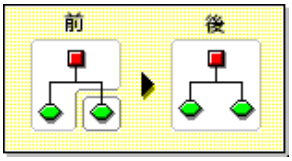
2 [**パーティションの作成** ＋] をクリックし、[**コンテナ** ] を選択します。

3 新しいパーティションの作成元となるコンテナの名前およびコンテキストを指定して、[**作成**] をクリックします。

パーティションのマージ

パーティションをそのペアレントパーティションにマージすると、選択したパーティションおよびそのレプリカがペアレントパーティションに結合されます。パーティションの削除は行いません。次の図のように、パーティションのマージおよび作成のみを実行し、ディレクトリツリーがどのように論理区分に分割されるかを定義します。

***図6-2** パーティションのマージ前とマージ後*



パーティションとそのペアレントパーティションをマージする理由として、次のようなことが考えられます。

- 2 つのパーティションのディレクトリ情報が密接に関連している。
- サブオーディネートパーティションを削除する場合に、その中のオブジェクトを残したい。
- パーティションのオブジェクトを削除する。
- パーティションのすべてのレプリカを削除する。パーティションをその親にマージする操作は、パーティションのマスタレプリカを削除するための唯一の方法です。
- コンテナを移動した後で (パーティションルートに従属パーティションがない場合のみ)、このコンテナをパーティションとする必要がなくなった。
- 会社の組織に変更が生じたため、パーティション構造を変更することでディレクトリツリーを再設計する。

パーティションが大きくて、数百個のオブジェクトが含まれている場合は、ネットワークの応答時間が遅くなるため、パーティションを別々に維持するようにしてください。

ツリーのルートパーティションは最上位のパーティションであり、マージするペアレントパーティションがないため、マージできません。

サーバで処理が完了すると、パーティションがマージされます。パーティションのサイズ、ネットワークトラフィック、サーバの環境設定などによって異なりますが、この操作の完了にはかなり時間がかかる場合があります。

重要 : パーティションのマージを行う前に、両パーティションの同期を点検し、続行する前にすべてのエラーを修正します。エラーを修正することにより、ディレクトリでの問題を切り離し、エラーの伝播や新しいエラーの発生を防ぐことができます。

パーティションのマージを行う前に、マージするパーティションのレプリカ (サブオーディネートリファレンスを含む) を持つサーバすべてが稼働していることを確認します。サーバがダウンしている場合、OpenText eDirectory はサーバのレプリカを読み込むことができず、操作を完了できません。

パーティションのマージの処理中にエラーが表示された場合は、そのつどエラーを解決します。操作を続けながらエラーを修正すると、さらにエラーが発生するため、必ず操作を中断してからエラーを修正してください。

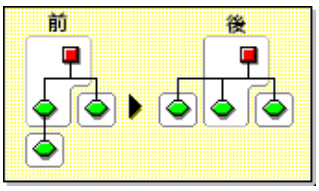
チャイルドパーティションをペアレントパーティションとマージするには、次の操作を行います。

1. OpenText Identity Console のホームページで、[**パーティション管理**] タイルをクリックします。
2. [**パーティション**] ページで、[**パーティション**] > [**パーティションのマージ**] アイコンの順に選択します。
3. 親パーティションとマージするパーティションの名前およびコンテキストを指定して、[**OK**] をクリックします。

パーティションの移動

パーティションを移動することで、ディレクトリツリー内のサブツリーを移動できます。ルートパーティションオブジェクト (コンテナオブジェクト) にサブオーディネートパーティションがない場合にのみ、このオブジェクトを移動できます。

***図 6-3** パーティションの移動前と移動後*



パーティションを移動する場合は、OpenText eDirectory の包含ルールに従う必要があります。たとえば、部門は現在のツリーのルートの直下には移動できません。これは、ルートの包含ルールにより移動が許されているのが、地域、国、または組織であるためです。

パーティションを移動する場合、OpenText eDirectory ではパーティションのルートオブジェクトへのすべての参照が変更されます。オブジェクトの共通名は変更されませんが、コンテナ (およびそのサブオーディネートコンテナすべて) の完全識別名は変更されます。

パーティションを移動するときに、オプションを選択して、移動するコンテナの代わりに別名オブジェクトを作成することもできます。これにより、ユーザが引き続きネットワークにログインして、元のディレクトリの場所でオブジェクトを検索できます。

作成した別名オブジェクトは、移動したコンテナと同じ共通名を持ち、そのコンテナの新しい完全識別名を参照します。

重要 : パーティションを移動したときに、移動したパーティションの代わりとなる別名オブジェクトを作成しないと、パーティションの新しい位置を知らないユーザは元のディレクトリ位置でオブジェクトを見つけようとし、ディレクトリツリーにあるパーティションオブジェクトを見つけることが困難になります。

また、ワークステーションの NAME CONTEXT パラメータがディレクトリツリーコンテナのオリジナルの位置に設定されている場合、これにより、クライアントワークステーションがログインできないという問題が起きるおそれがあります。

オブジェクトを移動すると、オブジェクトのコンテキストが変更されるため、移動したオブジェクトを参照するネームコンテキストを持つユーザの NAME CONTEXT パラメータは、オブジェクトの新しい名前を参照するように更新する必要があります。

コンテナオブジェクトの移動の後に、ユーザの NAME CONTEXT パラメータを自動的に更新するには、NCUPDATE ユーティリティを使用します。

移動したパーティションをパーティションとして使用しない場合は、それをペアレントパーティションとマージします。

パーティションを移動する前に、ディレクトリツリーが正しく同期されていることを確認してください。移動元か移動先のいずれかのパーティションで同期エラーが発生している場合は、パーティションの移動操作を実行しないでください。最初に同期エラーを解決します。

パーティションを移動するには、次の手順を実行します。

1. OpenText Identity Console のホームページで、[**パーティション管理**] タイルをクリックします。
2. [**パーティション**] ページで、[**パーティション**] > [**パーティションの移動**] アイコンの順に選択します。
3. [**オブジェクト名**] フィールドに移動するパーティションオブジェクトの名前およびコンテキストを指定します。
4. [**移動先**] フィールドに、パーティションの移動先となるコンテナの名前とコンテキストを指定します。
5. パーティションの移動元の別名を作成する場合は、[**移動したオブジェクトの代わりに別名を作成します**] を選択します。

   これにより、移動前の場所に依存するあらゆる操作は、操作を更新して移動後の場所を反映できるようになるまで、引き続き実行されます。
6. [**OK**] をクリックします。

パーティションの作成操作またはマージ操作のキャンセル

変更を確定する段階まで操作が到達していない場合、パーティションの作成またはマージをキャンセルできます。この機能を使用して、操作を終了できます。また、OpenText eDirectory ネットワークが、OpenText eDirectory エラーを返したり、パーティション操作に続く同期に失敗した場合にもこの機能を使用します。

ディレクトリツリーのレプリカで同期エラーが生じていると、操作を中止しても問題が解決しない場合があります。ただし、初期のトラブルシューティングオプションとして、この機能を使用できます。

サーバが停止している、またはその他の理由でサーバが使用できない場合、パーティション操作を完了できるようにサーバがネットワークから見えるようにするか、操作を中止します。データベースが壊れているために OpenText eDirectory による同期ができない場合、実行中のパーティション操作すべてを中止する必要があります。

含まれるレプリカの数、サーバの可視性、および既存のワイヤトラフィックの量によりますが、パーティション操作でネットワーク間の完全な同期を行うには、かなりの時間がかかります。

パーティションが使用中であることを示すエラーが表示されても、操作を中止する必要はありません。パーティションのサイズ、接続性の問題などによって異なりますが、通常、パーティション操作は 24 時間以内に完了します。この時間内で操作が完了しない場合には、実行中の操作を中止します。

レプリカの管理

レプリカの追加、削除、またはレプリカタイプの変更をする前に、ターゲットレプリカの位置を慎重に計画します。詳細については、95 ページの 「ツリーのレプリカ作成に関するガイドライン」を参照してください。

レプリカを追加する

次の機能をディレクトリに提供するために、レプリカをサーバに追加します。

- 障害対策
- データへのより高速なアクセス
- WAN リンク上でのより高速なアクセス
- 設定コンテキスト中のオブジェクトへのアクセス (バインダリサービスを使用)

レプリカを追加するには、次の操作を行います。

1. OpenText Identity Console のホームページで、[**パーティション管理**] タイルをクリックします。
2. [**パーティション**] ページで、[**タイプ**] ドロップダウンリストから [サーバ] を選択し、をクリックします。

3 サーバを選択し、[レプリカビュー] ページで [レプリカの追加 ＋] をクリックして、複製する [パーティション名] とサーバを指定します。

4 次のレプリカタイプのいずれかを選択します。

レプリカタイプ	説明
読み書き可能	ユーザは、新しいレプリカの内容の読み込みと変更を両方行うことができます。このパーティションの OpenText eDirectory オブジェクトを管理するユーザの近くに変更可能なレプリカがない場合は、このオプションを選択します。
読み込み専用	ユーザは、新しいレプリカの内容を読み込むことはできますが、変更はできません。このパーティションの OpenText eDirectory オブジェクトを読み込むだけで、変更は行わないユーザの近くにレプリカがない場合は、このオプションを選択します。
フィルタ済み読み書き可能	ユーザは、新しいレプリカの内容の読み込みと変更を両方行うことができますが、このレプリカの内容は、フィルタで指定された OpenText eDirectory オブジェクトおよびプロパティのタイプに制限されます。
フィルタ済み読み込み専用	ユーザは、新しいレプリカの内容を読み込めても変更はできず、このレプリカの内容は、OpenText eDirectory オブジェクトおよびフィルタで指定されたプロパティのタイプに制限されます。

5 OK をクリックします。

詳細については、63 ページの 「レプリカのタイプ」を参照してください。

レプリカを削除する

レプリカを削除すると、パーティションのレプリカはサーバから削除されます。

サーバをディレクトリツリーから削除する場合は、その前に、レプリカをサーバから削除します。レプリカを削除することで、サーバを削除するときに起きる問題を減らすことができます。

また、レプリカの削除により、ネットワークの同期トラフィックの量も削減できます。通常、パーティションに 7 個以上のレプリカは必要ありません。

マスタレプリカまたはサブオーディネートリファレンスは、削除できません。

マスタレプリカを削除するには、次の 2 つのオプションを利用します。

- マスタレプリカをパーティションの別のレプリカを含むサーバへ移動し、そのレプリカを新しいマスタレプリカにします。

  これにより、元のマスタレプリカは自動的に読み書き可能レプリカに変更され、削除できるようになります。

- パーティションをそのペアレントパーティションとマージします。

  これにより、パーティションのレプリカとペアレントパーティションのレプリカがマージされ、そのサーバからレプリカが削除されます。マージによりパーティションの境界は削除されますが、オブジェクトは削除されません。オブジェクトは、「結合」パーティションのレプリカを持つ各サーバに残ります。

レプリカを削除する場合、次の点に注意します。

- 障害対策として、異なるサーバ上に各パーティションのレプリカを 3 つ以上保持します。
- レプリカを削除すると、ターゲットサーバ上のディレクトリデータベースのコピーが削除されます。

  データベースは、ネットワークの別のサーバで引き続きアクセスできます。また、レプリカが含まれていたサーバも、OpenText eDirectory で引き続き機能します。

  サブオーディネートリファレンスレプリカは、削除や管理ができません。サーバがパーティションのレプリカを含む場合、サブオーディネートリファレンスレプリカは、OpenText eDirectory により自動的にサーバ上に作成されます。ただし、サーバがパーティションのチャイルドレプリカを含む場合には、作成されません。

レプリカを削除するには、次の操作を行います。

1 OpenText Identity Console のホームページで、[**パーティション管理**] タイルをクリックします。

2 [**パーティション**] ページで、[**タイプ**] ドロップダウンリストから [サーバ] を選択し、をクリックします。

3 削除するレプリカを保持するパーティションまたはサーバを選択します。

4 [ ] アイコンをクリックします。

レプリカタイプを変更する

レプリカタイプを変更して、レプリカ情報へのアクセスを制御します。たとえば、既存の読み書き可能レプリカを読み込み専用レプリカに変更して、ユーザがレプリカに書き込んだり、ディレクトリデータを変更できないようにします。

読み書き可能レプリカ、または読み込み専用レプリカのタイプを変更できます。マスタレプリカのタイプは変更できませんが、読み書き可能レプリカまたは読み込み専用レプリカは、マスタレプリカに変更できます。これにより、元のマスタレプリカは、自動的に読み書き可能レプリカに変更されます。

通常、ほとんどのレプリカは、読み書き可能レプリカとして使用されます。読み書き可能レプリカには、クライアント操作による書き込みができます。変更が加えられると、読み書き可能レプリカは同期情報を送信します。読み込み専用レプリカには、クライアント操作による書き込みができません。しかし、レプリカを同期すると、読み込み専用レプリカは更新されます。

サブオーディネートリファレンスのレプリカタイプは、変更できません。サブオーディネートリファレンスがあるサーバに、パーティションのレプリカを配置するには、レプリカの追加操作を行う必要があります。サブオーディネートリファレンスレプリカは、パーティションの完全なコピーではありません。サブオーディネートリファレンス (レプリカ)

の配置および管理は、OpenText eDirectory で制御します。サーバがパーティションのレプリカを含む場合、サブオーディネートリファレンスレプリカは、OpenText eDirectory により自動的にサーバ上に作成されます。ただし、サーバがパーティションのチャイルドレプリカを含む場合には、作成されません。

レプリカタイプを変更するには、次の操作を行います。

1 OpenText Identity Console のホームページで、[**パーティション管理**] タイルをクリックします。

2 [**パーティション**] ページで、[**タイプ**] ドロップダウンリストから [サーバ] を選択し、をクリックします。

3 変更するレプリカを保持するパーティションまたはサーバを選択します。

4 [**レプリカビュー**] ページで、変更するレプリカタイプを選択します。

5 レプリカタイプを変更し、[**OK**] をクリックします。

レプリカタイプ	説明
マスタ	ユーザは、このレプリカの内容の読み込みと変更を行うことができ、このレプリカは、下位パーティションの作成やマージなど、このパーティションに影響を与える将来のパーティション処理の出発点となります。1 つのパーティションに設定できるマスタレプリカは 1 つだけです。
読み書き可能	ユーザは新しいレプリカの内容の読み込みと変更を行うことができます。このパーティションの OpenText eDirectory オブジェクトを管理するユーザの近くに変更可能なレプリカがない場合は、このオプションを選択します。
読み込み専用	ユーザは新しいレプリカの内容を読み込むことができますが、変更はできません。このパーティションの OpenText eDirectory オブジェクトを読み込むだけで、変更は行わないユーザの近くにレプリカがない場合は、このオプションを選択します。
フィルタ済み読み書き可能	ユーザは新しいレプリカの内容の読み込みと変更を行うことはできますが、内容はフィルタで指定された OpenText eDirectory オブジェクトとプロパティのタイプに制限されます。
フィルタ済み読み込み専用	ユーザは新しいレプリカの内容を読み込めても変更はできません。内容はフィルタで指定された OpenText eDirectory オブジェクトとプロパティのタイプに制限されます。

「**レプリカタイプが正常に変更されました！**」というメッセージが表示されます。

6 **OK** をクリックします。

詳細については、63 ページの 「レプリカのタイプ」を参照してください。

フィルタ済みレプリカを設定し管理する

フィルタ済みレプリカには、OpenText eDirectory パーティションの情報のフィルタ済みサブセット (オブジェクトまたはオブジェクトクラス、およびこれらのオブジェクトの属性と値のフィルタ済みセット) が保存されています。

管理者は、フィルタ済みレプリカのセットを保持する OpenText eDirectory サーバを作成するためにフィルタ済みレプリカ機能を使用します。フィルタ済みレプリカのセットには、同期するオブジェクトおよび属性のみが含まれます。

このため、OpenText Identity Console では、フィルタ済みレプリカのパーティションスコープおよびフィルタを作成できるツールが用意されています。スコープは、サーバ上でレプリカを保存するパーティションのセットです。レプリケーションフィルタには、サーバのフィルタ済みレプリカセットにホストする OpenText eDirectory のクラスと属性のセットが含まれています。この結果、OpenText eDirectory サーバには、ツリー内の多くのパーティションから明確に定義されたデータセットが格納されるようになります。

サーバのパーティションスコープおよびレプリケーションフィルタの記述は、OpenText eDirectory に格納され、OpenText Identity Console のサーバオブジェクトまたは［パーティションとレプリカ］役割によって管理できます。

- 168 ページの 「フィルタ処理済レプリカウィザードを使用する」
- 169 ページの 「パーティションスコープを定義する」
- 170 ページの 「サーバフィルタを設定する」

フィルタ処理済レプリカウィザードを使用する

フィルタ処理済レプリカウィザードを使用すると、サーバのレプリケーションフィルタおよびパーティションスコープを、表示される手順に従って簡単に設定できます。

1 OpenText Identity Console のホームページで、［**パーティション管理**］タイルをクリックします。

2 ［**パーティション**］ページで、［**タイプ**］ドロップダウンリストから［サーバ］を選択し、をクリックします。

3 フィルタ済みレプリカを設定するサーバを選択します。

4 ［**レプリカビュー**］ページで、［**レプリカの追加** ＋］をクリックし、をクリックして、パーティションコンテナを選択します。

レプリケーションフィルタには、サーバのフィルタ済みレプリカセットに保存する OpenText eDirectory クラスと属性のセットが含まれます。フィルタセットの定義の詳細については、170 ページの 「サーバフィルタを設定する」を参照してください。

5 レプリカタイプを変更します。例 : ［フィルタされた読み込み - 書き込み］、［フィルタされた読み込み専用 (Filtered read-only)］など。

6 ［**OK**］をクリックします。

7 このサーバのパーティションスコープを定義するには、［**編集**］をクリックします。

パーティションスコープの詳細については、「169 ページの 「パーティションスコープを定義する」」を参照してください。

8 ［OK］をクリックします。

パーティションスコープを定義する

パーティションスコープは、サーバ上でレプリカを保存するパーティションのセットです。OpenText Identity Console の［レプリカビュー］ページには、OpenText eDirectory ツリーのパーティションの階層が表示されます。個別のパーティション、指定したブランチのパーティションセット、またはツリー内のすべてのパーティションを選択できます。次に、サーバに追加するこれらのパーティションのレプリカタイプを選択するか、既存のレプリカタイプを変更します。

サーバには、完全なレプリカもフィルタ済みレプリカも保存できます。詳細については、66 ページの 「フィルタ済みレプリカ」を参照してください。

OpenText eDirectory サーバのレプリカを表示する

1 OpenText Identity Console のホームページで、［パーティション管理］タイルをクリックします。

2 ［パーティション］ページで、［タイプ］ドロップダウンリストから［サーバ］を選択し、をクリックします。

［パーティション］ページで、このサーバのレプリカのリストを表示できます。

OpenText eDirectory サーバにフィルタ済みレプリカを追加する

1 OpenText Identity Console のホームページで、［パーティション管理］タイルをクリックします。

2 ［パーティション］ページで、［タイプ］ドロップダウンリストから［サーバ］を選択し、をクリックします。

3 フィルタ済みレプリカの追加先サーバを選択します。

4 ［レプリカビュー］ページで、［レプリカの追加］をクリックします。

5 パーティションの名前およびコンテキストを指定します。

6 ［フィルタ済み読み書き可能］または［フィルタ済み読み込みのみ］をクリックし、［OK］をクリックします。

完全なレプリカをフィルタ済みレプリカへ変更する

1 OpenText Identity Console のホームページで、［パーティション管理］タイルをクリックします。

2 ［パーティション］ページで、［タイプ］ドロップダウンリストから［サーバ］を選択し、をクリックします。

3 変更するレプリカを保持するパーティションまたはサーバを選択し、［OK］をクリックします。

4 変更するレプリカのレプリカタイプ ([タイプ] 列内) をクリックします。

5 [フィルタ済み読み書き可能] または [フィルタ済み読み込みのみ] をクリックし、[OK] をクリックします。

サーバフィルタを設定する

サーバレプリケーションフィルタには、サーバのフィルタ済みレプリカセットにホストする OpenText eDirectory のクラスと属性のセットが含まれています。どのサーバオブジェクトからでもフィルタを設定できます。フィルタ済みレプリカの場合、サーバごとにフィルタを 1 つだけ作成できます。つまり、ある OpenText eDirectory サーバ用に定義されているフィルタは、そのサーバ上のすべてのフィルタ済みレプリカに適用されます。ただし、完全なレプリカにはフィルタは適用されません。

サーバのフィルタは必要に応じて変更できますが、変更するとレプリカの再同期が発生するため時間がかかる可能性があります。サーバの機能に関するプラニングは、慎重に行うようにしてください。

次の方法のいずれかで、サーバのフィルタを設定または変更できます。

- 170 ページの 「レプリカビューを使用する」
- 170 ページの 「サーバオブジェクトを使用する」

レプリカビューを使用する

1 OpenText Identity Console のホームページで、[パーティション管理] タイルをクリックします。

2 [パーティション] ページで、[タイプ] ドロップダウンリストから [サーバ] を選択し、をクリックします。

3 変更するレプリカを格納するパーティションまたはサーバの名前およびコンテキストを選択して、[OK] をクリックします。

4 変更するサーバまたはパーティションの [Edit in the Filter] 列をクリックします。

5 適切なクラスおよび属性を追加し、[OK] をクリックします。

サーバオブジェクトを使用する

1 OpenText Identity Console のホームページで、[オブジェクトの管理 (Obejct Management)] タイルをクリックします。

2 [ディレクトリ管理] > [オブジェクトの変更] の順にクリックします。

3 変更するレプリカを格納するサーバの名前およびコンテキストを指定して、[OK] をクリックします。

4 [レプリカ] タブをクリックします。

5 このサーバにフィルタが定義されていなかった場合、[フィルタは空です] をクリックして [フィルタ編集ダイアログ] ウィンドウを開き、所要のクラスおよび属性を追加します。

または

［Copy Filter From］をクリックし、コピーするフィルタを含むオブジェクト (別のサーバなど) をブラウズします。

6 既存のフィルタを編集するには、フィルタ内のハイパーリンク付きアイテムをクリックし、［フィルタの編集］ダイアログボックスを開いて、目的のクラスおよび属性を追加または削除します。

パーティションおよびレプリカを表示する

このセクションでは、次のことを説明します。

- 171 ページの 「サーバのパーティションを表示する」
- 171 ページの 「パーティションレプリカを表示する」
- 172 ページの 「パーティションに関する情報を表示する」
- 172 ページの 「パーティションの階層を表示する」
- 172 ページの 「レプリカに関する情報を表示する」

サーバのパーティションを表示する

OpenText Identity Console では、サーバに割り当てられたパーティションを表示できます。サーバオブジェクトをディレクトリツリーから削除しようとする場合、サーバに格納されているパーティションの表示が必要となる場合があります。この場合、オブジェクトを削除する前に、削除するレプリカを表示できます。

1 OpenText Identity Console のホームページで、［パーティション管理］タイルをクリックします。
2 サーバオブジェクトの名前とコンテキストを選択します。
3 ［パーティション］ページで、パーティションを表示します。

パーティションレプリカを表示する

この操作により、次を識別できます。

- パーティションのレプリカが存在するサーバ
- パーティションのマスタレプリカのホストとなっているサーバ
- 読み書き可能レプリカ、読み込み専用レプリカ、およびサブオーディネートリファレンスレプリカを含むサーバ
- 各パーティションレプリカの状態

パーティションレプリカを表示するには、次の操作を行います。

1 OpenText Identity Console のホームページで、［パーティション管理］タイルをクリックします。
2 ［パーティション］ページで、パーティションを選択します。
［レプリカビュー］ページの情報を表示します。

パーティションに関する情報を表示する

パーティションに関する情報 (成功した最新の同期や最近試みた同期など) を表示する主な目的は、パーティションの同期情報を確認することです

1 OpenText Identity Console のホームページで、[**パーティション管理**] タイルをクリックします。
2 [**パーティション**] ページで、パーティションを選択します。
3 [**レプリカビュー**] ページで、サーバを選択します。
[**レプリカ情報**] ページが表示されます。

パーティションの階層を表示する

パーティション階層は、OpenText Identity Console で表示できます。ホームページの [**ツリービュー**] タイルをクリックすると、親パーティションと子パーティションを確認できます。

レプリカに関する情報を表示する

レプリカに関する情報を表示する主な目的は、レプリカの状態を確認することです OpenText eDirectory レプリカの状態は、それが実行しているパーティションや複製の操作によってさまざまです。次の表で、OpenText Identity Console に表示されるレプリカの状態を説明します。

状態	説明
オン	現在パーティションや複製の操作を実行していない
New (新規)	サーバに新しいレプリカとして追加中
停止中	サーバから削除中
停止	サーバからの削除が完了
マスタ開始	マスタレプリカへ変更中
マスタ完了	マスタレプリカへの変更が完了
タイプの変更	他のレプリカタイプへの変更中
ロック状態	パーティションの移動または修復の操作の準備が滞っている
移動へ移行	パーティションの移動操作を開始中
Move	パーティションの移動操作中
分割へ移行	パーティションの分割操作 (チャイルドパーティションの作成) を開始中
分割	パーティションの分割 (チャイルドパーティションの作成) 操作中
Join	ペアレントパーティションへのマージ中
オンへ移行	オン状態へ戻る直前

状態	説明
不明	OpenText Identity Console で不明の状態

レプリカの情報を表示するには、次の操作を行います。

1. OpenText Identity Console のホームページで、[**パーティション管理**] タイルをクリックします。
2. [**タイプ**] ドロップダウンメニューで、サーバを選択し、[**検索**] をクリックします。
3. 必要なレプリカをクリックすると、そのレプリカに関する情報が表示されます。

7 OpenText eDirectory 管理ユーティリティ

この章では、次の OpenText eDirectory ユーティリティについて説明します。

- 175 ページの 「OpenText インポート / エクスポート変換ユーティリティ」
- 209 ページの 「インデックスマネージャ」
- 212 ページの 「OpenText eDirectory Service Manager」
- 213 ページの 「オフラインのバルクロードユーティリティ」
- 224 ページの 「LDIF ファイル」

OpenText インポート / エクスポート変換ユーティリティ

OpenText インポート / エクスポート変換ユーティリティは次の操作に使用できます。

- LDIF ファイルから LDAP ディレクトリへのデータのインポート
- LDAP ディレクトリから LDIF ファイルへのデータのエクスポート
- LDAP サーバ間でのデータの移行
- スキーマの比較と更新の実行
- テンプレートを使用した OpenText eDirectory への情報のロード。
- SCH ファイルから LDAP ディレクトリへのスキーマのインポート

OpenText インポート / エクスポート変換ユーティリティは、形式に応じてデータを読み書きするための一連のハンドラを管理します。データを読み込むハンドラをソースハンドラと呼び、データを書き込むハンドラをターゲットハンドラと呼びます。1 つの実行可能モジュールがソースハンドラとターゲットハンドラの両方として機能することもあります。Novell インポート / エクスポート変換エンジンは、ソースハンドラからデータを受け取ってそれを処理し、処理したデータをターゲットハンドラに渡します。

たとえば、LDIF データを LDAP ディレクトリにインポートする場合、OpenText インポート / エクスポート変換エンジンは、LDIF ソースハンドラを使用して LDIF ファイルを読み込み、読み込んだデータを LDAP ターゲットハンドラを使用して LDAP ディレクトリサーバに送信します。LDIF ファイルの構文、構造、およびデバッグの詳細については、899 ページの付録 H「トラブルシューティング」を参照してください。

OpenText インポート / エクスポート変換クライアントユーティリティは、コマンドラインから、または OpenText Identity Console の［インポート / エクスポート変換ウィザード (Import Convert Export Wizard)］から実行できます。ただし、コンマ区切りのデータに対応したハンドラは、コマンドラインユーティリティと OpenText Identity Console でのみ使用できます。

OpenText インポート / エクスポート変換ユーティリティは、次のどちらの方法でも使用できます。

- 176 ページの 「コマンドラインインタフェースを使用する」

OpenText インポート / エクスポート変換エンジンには、ウィザードからもコマンドラインインタフェースからもアクセスできます。ただし、コマンドラインインタフェースの方が、ソースハンドラとターゲットハンドラの組み合わせでより多くの選択肢があります。

NDS および OpenText eDirectory の旧バージョンで提供されていた BULKLOAD ユーティリティと ZONEIMPORT ユーティリティはどちらも、OpenText インポート / エクスポート変換ユーティリティに置き換わっています。

コマンドラインインタフェースを使用する

OpenText インポート / エクスポート変換ユーティリティのコマンドラインバージョンは、次の操作に使用できます。

- LDIF のインポート
- LDIF のエクスポート
- コンマ区切りデータのインポート
- コンマ区切りデータのエクスポート
- LDAP サーバ間でのデータの移行
- スキーマの比較と更新
- テンプレートを使用した OpenText eDirectory への情報のロード
- スキーマのインポート

Windows バージョン (ice.exe) はインストールに組み込まれています。Linux コンピュータでは、インポート / エクスポートユーティリティは NOVLice パッケージに組み込まれています。

OpenText インポート / エクスポート変換構文

OpenText インポート / エクスポート変換ユーティリティは、次の構文で起動します。

```
ice general_options
-S[LDIF | LDAP | DELIM | LOAD | SCH] source_options
-D[LDIF | LDAP | DELIM] destination_options
```

またはスキーマキャッシュを使用する場合は、次の構文になります。

```
ice -C schema_options
-S[LDIF | LDAP] source_options
-D[LDIF | LDAP] destination_options
```

スキーマキャッシュを使用して更新を実行する場合、LDIF ファイルはターゲットとして有効ではありません。

一般オプションの指定は任意です。ただし、指定する場合はソースハンドラオプションやターゲットハンドラオプションより前に指定します。-S(ソース) ハンドラセクションと -D(ターゲット) ハンドラセクションはどちらを先に指定してもかまいません。

利用できるソースハンドラとターゲットハンドラは次のとおりです。

- 179 ページの 「LDIF ソースハンドラのオプション」
- 180 ページの 「LDIF ターゲットハンドラのオプション」
- 181 ページの 「LDAP ソースハンドラのオプション」
- 184 ページの 「LDAP ターゲットハンドラのオプション」
- 185 ページの 「DELIM ソースハンドラのオプション」
- 187 ページの 「DELIM ターゲットハンドラのオプション」
- 188 ページの 「SCH ソースハンドラのオプション」
- 188 ページの 「LOAD ソースハンドラのオプション」

全般オプション

一般オプションは、OpenText インポート / エクスポート変換エンジンの処理全体に影響のあるオプションです。

オプション	説明
-C	スキーマキャッシュを使用してスキーマの比較および更新を実行する場合に指定します。
-l *log_file*	出力メッセージ (エラーメッセージなど) を書き込むログファイルの名前を指定します。このオプションを指定しなかった場合、エラーメッセージは ice.log に出力されます。 Linux コンピュータでこのオプションを指定しなかった場合、エラーメッセージはログに記録されません。
-o	既存のログファイルを上書きします。このフラグを設定しなかった場合、メッセージは既存のログファイルの末尾に追加されます。
-e *LDIF_error_log_file*	正常に処理されなかったエントリを書き込むファイルの名前を指定します。エントリは LDIF 形式で書き込まれます。このファイルは、内容を調べてエラーを修正したうえで元のディレクトリに再適用できます。
-p *URL*	インポート / エクスポート変換エンジンが使用する XML 配置ルールが格納されている場所を指定します。配置ルールはエントリの位置を変更するときに使用します。詳細については、197 ページの 「変換ルール」を参照してください。
-c *URL*	インポート / エクスポート変換エンジンが使用する XML 作成ルールが格納されている場所を指定します。作成ルールは、インポート時にエントリを正しく作成するために必要な情報が欠落している場合にそれを補うために使用します。詳細については、197 ページの 「変換ルール」を参照してください。

オプション	説明
-s *URL*	インポート / エクスポート変換エンジンが使用する XML スキーママッピングルールが格納されている場所を指定します。スキーママッピングルールによって、転送元サーバのスキーマエレメントを、転送先サーバ上の同等ではあるが異なるスキーマエレメントにマッピングできます。 詳細については、197 ページの 「変換ルール」を参照してください。
-h または -?	コマンドラインのヘルプを表示します。

スキーマオプション

スキーマオプションでは、スキーマキャッシュを使用してスキーマの比較および更新操作を実行できます。

オプション	説明
-C -a	ターゲットスキーマを更新します (足りないスキーマを追加します)。
-C -c *filename*	指定したファイルにターゲットスキーマを出力します。
-C -n	スキーマの事前チェックを無効にします。

ソースハンドラのオプション

ソースハンドラオプション (-S) で、インポートするデータのソースを決定します。コマンドラインには、次のうちの 1 つだけを指定できます。

オプション	説明
-SLDIF	LDIF ファイルをソースとして指定します。 サポートされている LDIF オプションのリストについては、179 ページの「LDIF ソースハンドラのオプション」を参照してください。
-SLDAP	LDAP サーバをソースとして指定します。 サポートされている LDAP オプションのリストについては、181 ページの「LDAP ソースハンドラのオプション」を参照してください。
-SDELIM	コンマ区切りのデータファイルをソースとして指定します。 **注 :** パフォーマンスを向上させるには、OpenText インポート / エクスポート変換ユーティリティで DELIM ではなく LDIF ファイルを指定して、データをインポートします。カスタム PERL スクリプトを使用すると、必要なフォーマットで出力を生成できます。 サポートされている DELIM オプションのリストについては、185 ページの 「DELIM ソースハンドラのオプション」を参照してください。

オプション	説明
-SSCH	スキーマファイルをソースとして指定します。 サポートされている SCH オプションのリストについては、188 ページの「SCH ソースハンドラのオプション」を参照してください。
-SLOAD	DirLoad テンプレートをソースとして指定します。 サポートされている LOAD オプションのリストについては、188 ページの「LOAD ソースハンドラのオプション」を参照してください。

ターゲットハンドラのオプション

ターゲットハンドラオプション (-D) で、エクスポートするデータの書き込み先を決定します。コマンドラインには、次のうちの 1 つだけを指定できます。

オプション	説明
-DLDIF	LDIF ファイルを書き込み先として指定します。 サポートされているオプションのリストについては、180 ページの「LDIF ターゲットハンドラのオプション」を参照してください。
-DLDAP	LDAP サーバを書き込み先として指定します。 サポートされているオプションのリストについては、184 ページの「LDAP ターゲットハンドラのオプション」を参照してください。
-DDELIM	コンマ区切りのデータファイルを書き込み先として指定します。 サポートされているオプションのリストについては、187 ページの「DELIM ターゲットハンドラのオプション」を参照してください。

LDIF ソースハンドラのオプション

LDIF ソースハンドラは、LDIF ファイルからデータを読み込んで、それを OpenText インポート / エクスポート変換エンジンに送ります。

オプション	説明
-f *LDIF_file*	LDIF ソースハンドラで読み込んだ LDIF レコードを格納するファイルの名前を指定します。これらの LDIF レコードはインポート / エクスポート変換エンジンに送信されます。 Linux コンピュータでは、このオプションを指定しなかった場合、入力データは stdin から読み込まれます。
-a	このオプションを設定した場合、指定した LDIF ファイル内のレコードが内容レコード (変更タイプが設定されていないレコード) であれば、これらのレコードの変更タイプは「追加」であるとみなされます。

オプション	説明
-c	エラーが発生しても LDIF ソースハンドラの処理を続行します。ここでいうエラーとは、LDIF 解析エラーや、ターゲットハンドラから返されたエラーなどです。このオプションが設定されている場合にエラーが発生すると、LDIF ソースハンドラは、エラーを報告したうえで、LDIF ファイルの次のレコードを検出し、処理を続行します。
-n	実際の更新は行わず、更新を行った場合の結果を表示して確認します。このオプションを設定すると、LDIF ソースハンドラは、LDIF ファイルの解析は行いますが、OpenText インポート / エクスポート変換エンジンやターゲットハンドラへのレコードの送信は行いません。
-m	このオプションを設定した場合、指定した LDIF ファイル内のレコードが内容レコード (変更タイプが設定されていないレコード) であれば、これらのレコードの変更タイプは「変更」であるとみなされます。
-x	このオプションを設定した場合、指定した LDIF ファイル内のレコードが内容レコード (変更タイプが設定されていないレコード) であれば、これらのレコードの変更タイプは「削除」であるとみなされます。
-R *value*	処理するレコードの範囲を指定します。
-v	ハンドラの冗長モードを有効にします。
-e *value*	LDIF ファイル内の属性値を復号化するために使用するスキーム。[des/3des]。
-E *value*	属性の復号化用のパスワード。ADM_E_SRC_PASSWD 変数を使用して、LDIF ソースハンドラを暗号化できます。

LDIF ターゲットハンドラのオプション

LDIF ターゲットハンドラは、OpenText インポート / エクスポート変換エンジンからデータを受け取り、そのデータを LDIF ファイルに書き込みます。

オプション	説明
-f *LDIF_file*	LDIF レコードの書き込み先になるファイルの名前を指定します。 Linux コンピュータでは、このオプションを指定しなかった場合、出力データは stdout に送られます。
-B	バイナリ値も印刷します。
-b	LDIF データの Base64 エンコードを行いません。
-e *value*	LDAP サーバから受け取った属性値を暗号化するために使用するスキーマ。[des/3des]。
-E *value*	属性の暗号化用のパスワード。ADM_E_DEST_PASSWD 変数を使用して、LDIF ターゲットハンドラを暗号化できます。

LDAP ソースハンドラのオプション

LDAP ソースハンドラは、検索要求を該当する LDAP サーバに送信することによってそのサーバからデータを読み込みます。LDAP ソースハンドラは、検索操作の結果として受け取った検索エントリを OpenText インポート / エクスポート変換エンジンに送ります。

オプション	説明
-s *server_name*	ハンドラが検索要求を送るときの送信先 LDAP サーバの DNS 名または IP アドレスを指定します。デフォルトはローカルホストです。 **注 :** バージョン 9.1 以降を使用している場合は、このオプションに対して LDAP サーバの FQDN を指定する必要があります。
-p *port*	*server_name* で指定した LDAP サーバのポート番号を整数で指定します。デフォルトは 389 です。セキュリティで保護された操作の場合、デフォルトポートは 636 です。 ICE が SSL ポート (デフォルトは 636) で LDAP サーバと証明書なしで通信している場合は、サーバの証明書を信頼できるものとして許可します。このオプションは、サーバとクライアントの間で暗号化された通信が行われ、サーバの検証を必要としないような制御された環境でのみ使用してください。
-d *DN*	サーバによって指定されたバインド操作に使用するエントリの識別名を指定します。
-w *password*	*DN* で指定したエントリのパスワード属性を指定します。 ADM_SRC_PASSWD 変数を使用して、LDAP ソースハンドラにパスワードを提供できます。
-W	*DN* で指定したエントリのパスワードの入力を求めるプロンプトが表示されます。 このオプションは Linux でのみ適用できます。
-F *filter*	RFC 1558 準拠の検索フィルタを指定します。このオプションを指定しなかった場合、デフォルトの検索フィルタである「objectclass=*」が使用されます。
-n	実際の検索は行わず、検索の条件などを表示して確認します。
-a *attribute_list*	検索対象の属性をカンマ区切りのリスト形式で指定します。属性名を指定するか、次の 3 つのうちいずれかを指定します。 ◆ 属性を検索しない場合は、(1.1) ◆ すべてのユーザ属性を検索する場合は、(*) ◆ オペレーショナルでない属性をすべて検索する場合は、空のリスト。 このオプションを指定しなかった場合、属性リストは空のリスト (デフォルト) になります。

オプション	説明
-o *attribute_list*	LDAP サーバから受け取った検索結果をインポート / エクスポート変換エンジンに送信する前に、その検索結果から削除する属性をカンマ区切りのリスト形式で指定します。このオプションは、-a オプションでワイルドカードを指定してクラスの属性を初めにすべて検索してから、その検索結果をインポート / エクスポート変換エンジンに渡す前に一部の属性を削除したい場合に便利です。 たとえば、「-a* -o telephoneNumber」と指定すると、ユーザレベルの属性がすべて検索された後で、その検索結果から電話番号にフィルタを適用します。
-R	参照結果を自動的に適用しない場合に指定します。デフォルトでは、-d および -w オプションで指定された名前とパスワードによる参照結果が自動的に適用されます。
-e *value*	LDAP クライアント SDK で有効にするデバッグフラグを指定します。 詳細については、「899 ページの 「トラブルシューティング」」を参照してください。
-b *base_DN*	検索要求のベース識別名を指定します。このオプションを指定しなかった場合、ベース DN はデフォルトの " "(空の文字列) になります。
-c *search_scope*	検索要求のスコープを指定します。有効な値は次のとおりです。 ◆ One: この値を指定すると、ベースオブジェクトの直接の子だけが検索の対象になります。 ◆ Base: この値を指定すると、ベースオブジェクトのエントリだけが検索の対象になります。 ◆ Sub: この値を指定すると、ベースオブジェクトをルートとし、ベースオブジェクトを含む LDAP サブツリーが検索の対象になります。 このオプションを指定しなかった場合は、検索スコープのデフォルトの「Sub」が使用されます。

オプション	説明
-r *deref_aliases*	検索時に別名を逆参照する方法を指定します。指定できる値は次のとおりです。 ◆ Never: この値を指定すると、サーバは別名の逆参照を行いません。 ◆ Always: この値を指定すると、検索のベースオブジェクトを検索するときと検索フィルタに一致するエントリを評価するときの両方で、別名の逆参照が行われます。 ◆ Search: この値を指定すると、ベースオブジェクトを検出した後で検索スコープ内のエントリにフィルタを適用するときには別名の逆参照が行われますが、ベースオブジェクト自体の検索時には別名の逆参照は行われません。 ◆ 検索 : この値を指定すると、検索のベースオブジェクトを検索するときには別名の逆参照が行われますが、検索フィルタに一致するエントリを実際に評価するときには別名の逆参照は行われません。 このオプションを指定しなかった場合は、別名逆参照の方式のデフォルトである「Never」が使用されます。
-l *time_limit*	検索の制限時間を秒単位で指定します。
-z *size _limit*	検索結果として取得できるエントリの最大数を指定します。
-V *version*	接続に使用する LDAP プロトコルのバージョンを指定します。この値は 2 または 3 にする必要があります。このオプションを指定しなかった場合のデフォルトは 3 です。
-v	ハンドラの冗長モードを有効にします。
-L *filename*	SSL 認証に使用されるサーバキーが格納されている PEM 形式のファイルを指定します。デフォルト値は /var/opt/novell/eDirectory/data/SSCert.pem です。 **注 :** LDAP サーバで EC 証明書を使用している場合は、このオプションとともに SSECCert.pem を渡す必要があります。
-A	属性名のみを取得します。このオプションを指定した場合、出力される検索結果に属性値は含まれません。
-t	エラーが発生しても LDAP ハンドラの処理を続行したい場合に指定します。
-m	LDAP 操作は「変更」になります。
-x	LDAP 操作は「削除」になります。
-k	このオプションはサポートされなくなりました。SSL を使用するには、-L オプションを使用して有効な証明書を指定します。
-M	Manage DSA IT コントロールを有効にします。
-MM	Manage DSA IT コントロールを有効にし、重要度を高く設定します。

LDAP ターゲットハンドラのオプション

LDAP ターゲットハンドラは、OpenText インポート / エクスポート変換エンジンからデータを受け取り、それを LDAP サーバに送信します。データは更新操作の形で送信され、送信先サーバによって実行されます。

LDIF ファイル内のハッシュ化パスワードについては、「899 ページの付録 H「トラブルシューティング」」を参照してください。

オプション	説明
-s *server_name*	ハンドラが検索要求を送るときの送信先 LDAP サーバの DNS 名または IP アドレスを指定します。デフォルトはローカルホストです。
-p *port*	*server_name* で指定した LDAP サーバのポート番号を整数で指定します。デフォルトは 389 です。セキュリティで保護された操作の場合、デフォルトポートは 636 です。
-d *DN*	サーバによって指定されたバインド操作に使用するエントリの識別名を指定します。
-w *password*	*DN* で指定したエントリのパスワード属性を指定します。 ADM_DEST_PASSWD 変数を使用して、LDAP ターゲットハンドラにパスワードを提供できます。
-W	*DN* で指定したエントリのパスワードの入力を求めるプロンプトが表示されます。 このオプションは Linux でのみ適用できます。
-B	サーバへの更新操作の転送に非同期 LBURP(LDAP Bulk Update/Replication Protocol) 要求を使用しない場合は、このオプションを指定します。代わりに、標準の同期 LDAP 更新操作要求を使用します。 詳細については、207 ページの 「LBURP (LDAP Bulk Update/Replication Protocol)」を参照してください。
-F	前方向参照を作成できるようにします。作成するエントリの親が存在しない場合、エントリが正常に作成できるよう、そのエントリの親に対応するプレースフォルダが作成されます。このプレースホルダを**前方向参照**といいます。以後の操作で親が作成されると、前方向参照は通常のエントリに変更されます。
-l	パスワード値の格納に NMAS (NetIQ モジュラー認証サービス) の簡易パスワードメソッドを使用する場合は、このオプションを指定します。簡易パスワードを使用する場合、パスワードはディレクトリ内の安全な場所に保持されますが、鍵のペアはサーバ間での認証で実際に必要になるまで生成されません。
-e *value*	LDAP クライアント SDK で有効にするデバッグフラグを指定します。 詳細については、「899 ページの付録 H「トラブルシューティング」」を参照してください。

オプション	説明
-V *version*	接続に使用する LDAP プロトコルのバージョンを指定します。この値は 2 または 3 にする必要があります。このオプションを指定しなかった場合のデフォルトは 3 です。
-L *filename*	SSL 認証に使用されるサーバキーが格納されている PEM 形式のファイルを指定します。デフォルト値は /var/opt/novell/eDirectory/data/SSCert.pem です。 **注 :** LDAP サーバで EC 証明書を使用している場合は、このオプションとともに SSECCert.pem を渡す必要があります。
-k	このオプションはサポートされなくなりました。SSL を使用するには、-L オプションを使用して有効な証明書を指定します。
-M	Manage DSA IT コントロールを有効にします。
-MM	Manage DSA IT コントロールを有効にし、重要度を高く設定します。
-P	同時 LBURP 処理を有効にします。このオプションは、LDIF 内のすべての処理を追加する場合にだけ有効にします。–F オプションを使用する場合は、-P はデフォルトで有効になります。
-Z	非同期要求の数を指定します。この数値は、LDAP サーバから返される結果を待たずに、ICE クライアントから LDAP サーバに非同期に送信できるエントリの数を示します。

DELIM ソースハンドラのオプション

DELIM ソースハンドラはコンマ区切りのデータファイルからデータを読み込み、それをターゲットハンドラに送信します。

オプション	説明
-f *filename*	DELIM ソースハンドラによって読み込まれるコンマ区切りのレコードを含んだファイルの名前を指定します。これらの DELIM レコードはターゲットハンドラに送信されます。
-F *value*	-f で指定したファイルに対する属性のデータオーダを含んだファイルを指定します。 デフォルトでは、区切り文字で区切られたファイル内の属性の列数は、属性値の最大数と一致します。属性が繰り返されている場合、この列数はテンプレート内で属性が繰り返される回数に一致します。このオプションが指定されていない場合、-t を使用してこの情報を直接入力する必要があります。 詳細については、192 ページの 「コンマ区切りのインポートを実行する」を参照してください。

オプション	説明
-t *value*	コンマ区切りの属性リストです。このリストによって、-f で指定されたファイルに対する属性のデータオーダを指定します。 デフォルトでは、区切り文字で区切られたファイル内の属性の列数は、属性値の最大数と一致します。属性が繰り返されている場合、この列数はテンプレート内で属性が繰り返される回数に一致します。このオプションまたは -F のいずれかを指定する必要があります。 詳細については、192 ページの 「コンマ区切りのインポートを実行する」を参照してください。
-c	エラーが発生しても DELIM ソースハンドラの処理を続行したい場合に指定します。ここでいうエラーとは、カンマ区切りデータファイルの解析エラーや、ターゲットハンドラから返されたエラーなどです。このオプションが設定されている場合にエラーが発生すると、DELIM ソースハンドラは、エラーを報告したうえで、コンマ区切りデータファイル内の次のレコードを検出し、処理を続行します。
-n *value*	新しいオブジェクトに LDAP ネーミング属性を指定します。この属性は、-F または -t を使用して指定する属性データに含まれている必要があります。
-l *value*	RDN の追加先のパスを指定します (o=myCompany など)。DN を渡す場合、この値は必要ありません。
-o *value*	オブジェクトクラスのコンマ区切りのリスト (入力ファイルに含まれていない場合)、または補助クラスなどその他のオブジェクトクラスを指定します。デフォルト値は「inetorgperson」です。
-i *value*	スキップする列のコンマ区切りのリストです。この値には、スキップする列の数を整数で指定します。たとえば 3 列目と 5 列目をスキップする場合は、「i3,5」と指定します。
-d *value*	区切り記号を指定します。デフォルトの区切り記号はカンマ (,) です。 次に示すのは、特別な場合の区切り記号です。 ◆ [q] = 引用符 (区切り記号としての単一の '') ◆ [t] = タブ たとえば、タブを区切り記号として指定するには、-d[t] を渡します。
-q *value*	2 次的区切り記号を指定します。デフォルトのセカンダリ区切り記号は一重引用符 ('') です。 次に示すのは、特別な場合の区切り記号です。 ◆ [q] = 引用符 (区切り記号としての単一の '') ◆ [t] = タブ たとえば、タブを区切り記号として指定するには、-q[t] を渡します。
-v	冗長モードで実行します。

オプション	説明
-k *value*	区切りファイルの最初の行をテンプレートにする場合に指定します。このオプションを -t または -F とともに指定した場合、指定したテンプレートと区切りファイル内のテンプレートの整合性がチェックされます。

DELIM ターゲットハンドラのオプション

DELIM ターゲットハンドラはソースハンドラからデータを受け取り、そのデータをカンマ区切りのデータファイルに書き込みます。

オプション	説明
-f *filename*	コンマ区切りレコードの書き込み先になるファイルの名前を指定します。
-F *value*	-f で指定したファイルに対する属性のデータオーダを含んだファイルを指定します。 デフォルトでは、区切り文字で区切られたファイル内の属性の列数は、属性値の最大数と一致します。属性が繰り返されている場合、この列数はテンプレート内で属性が繰り返される回数に一致します。このオプションが指定されていない場合、-t を使用してこの情報を直接入力する必要があります。
-t *value*	コンマ区切りの属性リストです。このリストによって、-f で指定されたファイルに対する属性のデータオーダを指定します。 デフォルトでは、区切り文字で区切られたファイル内の属性の列数は、属性値の最大数と一致します。属性が繰り返されている場合、この列数はテンプレート内で属性が繰り返される回数に一致します。このオプションまたは -F のいずれかを指定する必要があります。
-l *value*	RDN または DN のいずれかになります。ドライバによってデータ内に配置されるのが、DN 全体または RDN のみのどちらになるかを指定します。デフォルト値は RDN です。
-d *value*	区切り記号を指定します。デフォルトの区切り記号はカンマ (,) です。 次に示すのは、特別な場合の区切り記号です。 ◆ [q] = 引用符 (区切り記号としての単一の ") ◆ [t] = タブ たとえば、タブを区切り記号として指定するには、-d[t] を渡します。

オプション	説明
-q *value*	2 次的区切り記号を指定します。デフォルトのセカンダリ区切り記号は一重引用符 (') です。 次に示すのは、特別な場合の区切り記号です。 ◆ [q] = 引用符 (区切り記号としての単一の ') ◆ [t] = タブ たとえば、タブを区切り記号として指定するには、-q[t] を渡します。
-n *value*	インポート処理中に追加されるネーミング属性を指定します。たとえば cn などです。

SCH ソースハンドラのオプション

SCH ハンドラは、古い NDS や OpenText eDirectory のスキーマファイル (拡張子 *.sch がついたファイル) からデータを読み込んで、それを OpenText インポート / エクスポート変換エンジンに送ります。このハンドラを使用すれば、拡張子 *.sch がついたファイルを入力として、スキーマ関連の操作を LDAP サーバに実装できます。

SCH ハンドラは、ソースハンドラだけのハンドラです。スキーマハンドラを使用すると、LDAP サーバに *.sch ファイルをインポートできますが、*.sch ファイルをエクスポートすることはできません。

SCH ハンドラでサポートされているオプションを次の表に示します。

オプション	説明
-f *filename*	*.sch ファイルの完全なパス名を指定します。
-v	(オプション) 冗長モードで実行します。

LOAD ソースハンドラのオプション

DirLoad ハンドラはテンプレートのコマンドから OpenText eDirectory 情報を生成します。このテンプレートファイルは -f 引数で指定します。このファイルには、属性仕様の情報とプログラム制御の情報が保持されます。

オプション	説明
-f *filename*	すべての属性仕様とプログラムの実行を制御するすべての情報を含んだテンプレートファイルを指定します。
-c	エラーが通知された場合、次のレコードから続行します。
-v	冗長モードで実行します。
-r	データが追加されずに削除されるように、要求を削除要求に変更します。このオプションにより、DirLoad テンプレートを使用して追加されたレコードを削除できます。
-m	テンプレートファイル内に変更要求を作成します。

属性仕様 新しいオブジェクトのコンテキストを決定します。

次の属性仕様ファイルのサンプルを参照してください。

```
givenname: $R(first)
initial: $R(initial)
sn: $R(last)
dn:cn=$A(givenname,%.1s)$A(initial,%.1s)$A(sn),ou=dev,ou=ds,o=novell
objectclass: inetorgperson
telephonenumber: 1-800-$N(1-999,%03d)-$C(%04d)
title: $R(titles)
locality: Our location
```

属性仕様ファイルの形式は LDIF ファイルに似ていますが、属性仕様ファイルでは強力な構成体を使用して、詳細な情報と属性間の関係を指定することができます。

固有の数値 指定されたオブジェクトに対する固有の数値を属性値に挿入します。

Syntax: $C[(<*format*)]

オプションの *<format* は、値に適用された出力形式を指定します。形式を指定しない場合、カッコは 2 種類とも必要ありません。

```
$C
$C(%d)
$C(%04d)
```

$C のみを指定すると、現在の数値が属性値に挿入されます。「%d」は、何も指定されなかった場合にプログラムが使用するデフォルト形式であるため、$C は $C(%d) と同じです。数値は各オブジェクトの後で増加するため、属性仕様で $C を複数回使用しても、単一オブジェクト内では数値は変わりません。開始値は、!COUNTER=*value* の構文を使用して、設定ファイル内で指定できます。

任意の数値 次の構文を使用して、属性値に任意の数値を挿入します。

$N(<*low*-<*high*[,<format])]

<low と *<high* では、下限値と上限値を指定します。任意の数字が生成される際に各々の値を使用します。オプションの *<format* は、リストの値に適用される出力形式を指定します。

```
$N(1-999)
$N(1-999,%d)
$N(1-999,%03d)
```

リストの任意の文字列 次の構文を使用して、指定したリストから任意に選択された文字列を属性値に挿入します。

```
$R(<filename[,<format])]
```

<filename には、値を格納しているファイルを指定します。ファイルへのパスは絶対パスまたは相対パスのどちらでも指定できます。リストを格納しているファイルには、このパッケージに含まれているものがあります。値は改行文字で区切る必要があります。

オプションの *<format* は、値に適用された出力形式を指定します。

```
$A(givenname)
$A(givenname,%s)
$A(givenname,%.1s)
```

前方向参照は使用できませんので注意してください。属性値を使用する場合、その属性は全て、属性仕様ファイル内で現在の属性より前にある必要があります。下記の例では、DN の一部である cn は、givenname、initial、および sn から構築されます。したがって設定ファイルでは、これらの属性は DN の前に存在する必要があります。

```
givenname: $R(first)
initial: $R(initial)
sn: $R(last)
dn:o=novell,ou=dev,ou=ds,cn=$A(givenname,%.1s)$A(initial,%.1s)$A(sn)
```

DN は LDIF ファイル内で特殊処理されます。設定内での DN の場所に関係なく、DN が最初に (LDIF 構文に従って)LDIF ファイルに書き込まれます。その他のすべての属性は、表示された順に書き込まれます。

制御設定 オブジェクト作成の際の制御を追加します。すべての制御には、属性設定と区別するために、行頭の文字として感嘆符 (!) が付いています。制御はファイル内の任意の場所に置くことができます。

```
!COUNTER=300
!OBJECTCOUNT=2
!CYCLE=title
!UNICYCLE=first,last
!CYCLE=ou,BLOCK=10
```

- カウンタ

  固有のカウンタ値の開始値を提供します。カウンタ値は、$C 構文内の任意の属性に挿入されます。

- オブジェクト数

  OBJECTCOUNT は、テンプレートから作成されるオブジェクトの数を決定します。

- サイクル

  CYCLE は、ファイル ($R 構文) から任意の値を抜き出す方法を変更するときに使用できます。この設定には異なる 3 つの値があります。

  ```
  !CYCLE=title
  ```

「title」というリスト名が使用される場合は常に、値が任意で選択されるのではなく、リストの順に次の値が抜き出されます。順番に値がすべて使用された場合には、再度リストの最初から開始します。

```
!CYCLE=ou,BLOCK=10
```

リスト「ou」のそれぞれの値が 10 回ずつ使用され、その後次の値に移動します。

CYCLE 制御設定のうちで最も興味深いバリアントは、UNICYCLE です。UNICYCLE は、一連のソースを左から右の順序で繰り返すように指定します。このため、必要な場合に一意の値が必ず作成されます。UNICYCLE 制御を使用する場合、OBJECTCOUNT 制御は、オブジェクト数を、リストから作成できる固有のオブジェクトの最大数に制限するためだけに使用します。つまり、UNICYCLE に含まれるリストが 15000 オブジェクトを作成できる場合、OBJECTCOUNT はその数を減らすことはできますが、増やすことはできません。

たとえば、givenname ファイルに 2 つの値 (「Doug」および「Karl」) があり、かつ sn ファイルに 3 つの値 (「Hoffman」、「Schultz」、および「Grieger」) があるとします。制御設定が「!UNICYCLE=givenname,sn」であり、属性定義が「cn: $R(givenname) $R(sn)」である場合、次の cn が作成されます。

cn: Doug Hoffmancn cn: Karl Hoffmancn cn: Doug Schultzcn cn: Karl Schultzcn cn: Doug Griegercn cn: Karl Grieger

例

ここでは、OpenText インポート / エクスポート変換ユーティリティのコマンドラインユーティリティで次の操作を行う場合のコマンドの例を紹介します。

- 191 ページの 「LDIF インポートの実行」
- 192 ページの 「LDIF エクスポートを実行する」
- 192 ページの 「コンマ区切りのインポートを実行する」
- 193 ページの 「コンマ区切りのエクスポートを実行する」
- 193 ページの 「LDAP サーバ間でデータを移行する」
- 194 ページの 「スキーマのインポートを実行する」
- 194 ページの 「LOAD ファイルのインポートを実行する」
- 197 ページの「暗号化された属性を含む LDIF エクスポートを LDAP サーバから実行する」
- 197 ページの 「暗号化された属性を含む LDIF インポートを実行する」

LDIF インポートの実行

LDIF のインポートを実行するには、LDIF ソースハンドラと LDAP ターゲットハンドラを組み合わせて次の例のように指定します。

```
ice -S LDIF -f entries.ldif -D LDAP -s server1.acme.com -p 389 -d
cn=admin,c=us -w secret
```

コマンドラインでこのように指定すると、LDIF データが entries.ldif から読み込まれ、ポート 389 にある LDAP サーバ server1.acme.com に送られます。送信時の識別子は「cn=admin,c=us」、パスワードは「secret」になります。

LDIF エクスポートを実行する

LDIF のエクスポートを実行するには、LDAP ソースハンドラと LDIF ターゲットハンドラを組み合わせます。次に例を示します。

```
ice -S LDAP -s server1.acme.com -p 389 -d cn=admin,c=us -w password -F
objectClass=* -c sub -D LDIF -f server1.ldif
```

コマンドラインでこのように指定すると、識別子「cn=admin,c=us」およびパスワード「password」を使用してサブツリー検索が実行され、ポート 389 にある LDAP サーバ server1.acme.com 内のオブジェクトがすべて検出されます。結果のデータは LDIF 形式で server1.ldif に出力されます。

コンマ区切りのインポートを実行する

コンマ区切りのインポートを実行するには、コマンドを次の例のように指定します。

```
ice -S DELIM -f/tmp/in.csv -F /tmp/order.csv -ncn -lo=acme -D LDAP -s
server1.acme.com -p389 -d cn=admin,c=us -w secret
```

コマンドをこのように指定すると、/tmp/in.csv ファイルからコンマ区切りの値が読み込まれ、/tmp/order.csv ファイルから属性の順序が読み込まれます。in.csv 内の各属性エントリに対して、order.csv で属性タイプが指定されます。たとえば、in.csv に次の値があるとします。

```
pat,pat,engineer,john
```

この場合、order.csv に含まれる値は次のようになります。

```
dn,cn,title,sn
```

order.csv の情報は、-t オプションを使用して直接入力することもできます。

次にデータは、識別子「cn=admin,c=us」、およびパスワード「secret」を使用して、ポート 389 で LDAP サーバ server1.acme.com に送られます。

この例では、-n オプションを使用して、cn がオブジェクトの新しい DN になるように指定し、-l オプションを使用して、このオブジェクトが組織コンテナ acme に追加されるようにします。

OpenText インポート / エクスポート変換ユーティリティを使用して生成されたコンマ区切りファイルは、生成するために使用されたテンプレートを最初の行に含みます。コンマ区切りファイルの最初の行がテンプレートであることを指定するには、-k オプションを使用します。-F または -t を -k とともに使用する場合、指定するテンプレートは、コンマ区切りファイル内のテンプレートと整合していなければなりません。つまり、両方の属性が正確に一致する必要があります。ただし、各属性の出現数や順序は異なっていても構いません。上の例では、in.csv の

最初の行に dn,cn,title,title,title,sn が含まれています。以下のテンプレートは整合しており、-k の使用時に -t または -F もともに使用できます。

dn,cn,title,sn (属性 title の反復回数が異なる)

dn,sn,title,cn (属性の順序が異なる)

しかし以下のテンプレートは、in.csv 内のテンプレートと整合していないため、-k の使用時に -t または -F を指定することはできません。

dn,cn,title,sn,objectclass (新しい属性 objectclass が含まれている)

dn,cn,title (属性 sn が含まれていない)

コンマ区切りのエクスポートを実行する

コンマ区切りのエクスポートを実行するには、コマンドを次の例のように指定します。

```
ice -S LDAP -s server1.acme.com -p 389 -d cn=admin,c=us -w password -F
objectClass=* -c sub -D DELIM -f /tmp/server1.csv -F order.csv
```

コマンドラインでこのように指定すると、識別子「cn=admin,c=us」およびパスワード「password」を使用してサブツリー検索が実行され、ポート 389 にある LDAP サーバ server1.acme.com 内のオブジェクトがすべて検出されます。結果のデータはコンマ区切りの形式で /tmp/server1.csv ファイルに出力されます。

order.csv 内の属性に複数の値が含まれている場合、/tmp/server1.csv では、この属性の列数が、属性値の最大数と一致します。order.csv 内で属性が繰り返されている場合、この属性の列数は、属性が繰り返される回数と一致します。

たとえば、order.csv に dn,sn,objectclass が含まれている場合、エクスポートされたすべてのエントリについて、objectclass の値が 4 つあるのに対し、dn と sn の値は 1 つずつしかない場合、dn と sn の列数はそれぞれ 1 つずつになるのに対し、objectclass の列数は 4 つになります。objectclass の値を 2 つだけコンマ区切りファイルに出力するには、order.csv には dn,sn,objectclass,objectclass が含まれている必要があります。

どちらの場合も、属性が /tmp/server1.csv の最初の行に書き込まれます。最初の例では、/tmp/server1.csv の最初の行に dn,sn,objectclass,objectclass,objectclass,objectclasss が含まれており、2 番目の例では、/tmp/server1.csv の最初の行に dn,sn,objectclass,objectclass が含まれています。

2 回目以降のインポート時に、最初の行を一連の属性として処理させないようにするには、-k オプションを使用します。詳細については、192 ページの 「コンマ区切りのインポートを実行する」を参照してください。

LDAP サーバ間でデータを移行する

LDAP サーバ間でデータを移行するには、LDAP ソースハンドラと LDAP ターゲットハンドラを組み合わせます。次に例を示します。

```
ice -S LDAP -s server1.acme.com -p 389 -d cn=admin,c=us -w password -F
objectClass=* -c sub -D LDAP -s server2.acme.com -p 389 -d cn=admin,c=us -
w secret
```

このコマンドを実行すると、識別子「cn=admin,c=us」およびパスワード「password」を使用してサブツリー検索が実行され、ポート 389 にある LDAP サーバ server1.acme.com 内のオブジェクトがすべて検出されます。結果のデータは、識別子「cn=admin,c=us」およびパスワード「secret」で、ポート 389 にある LDAP サーバ server2.acme.com に送られます。

スキーマのインポートを実行する

スキーマファイルのインポートを実行するには、コマンドを次の例のように指定します。

```
ice -S SCH -f $HOME/myfile.sch -D LDAP -s myserver -d cn=admin,o=novell -w
passwd
```

このコマンドを実行すると、スキーマデータが myfile.sch から読み込まれ、LDAP サーバ「myserver」に送られます。送信時の識別子は「cn=admin,o=novell」、パスワードは「passwd」になります。

LOAD ファイルのインポートを実行する

LOAD ファイルのインポートを実行するには、コマンドを次の例のように指定します。

```
ice -S LOAD -f attrs -D LDIF -f new.ldf
```

次に、属性ファイル「attrs」の内容の例を示します。

```
#=============================================================================

#  DirLoad 1.00
#=============================================================================

!COUNTER=300

!OBJECTCOUNT=2
#-----------------------------------------------------------------------------

#  ATTRIBUTE TEMPLATE
# -----------------------------------------------------------------------------

objectclass: inetorgperson

givenname: $R(first)

initials: $R(initial)

sn: $R(last)

dn: cn=$A(givenname,%.1s)$A(initial,%.1s)$A(sn),ou=$R(ou),ou=dev,o=novell,

telephonenumber: 1-800-$N(1-999,%03d)-$C(%04d)

title: $R(titles)
```

コマンドプロンプトから前のコマンドを実行すると、次の LDIF ファイルが作成されます。

```
version: 1

dn: cn=JohnBBill,ou=ds,ou=dev,o=novell

changetype: add
```

```
objectclass: inetorgperson

givenname: John

initials: B

sn: Bill

telephonenumber: 1-800-290-0300

title: Amigo

dn: cn=BobJAmy,ou=ds,ou=dev,o=novell

changetype: add

objectclass: inetorgperson

givenname: Bob

initials: J

sn: Amy

telephonenumber: 1-800-486-0301

title: Pomo
```

コマンドプロンプトから次のコマンドを実行すると、データが LDAP ハンドラを経由して LDAP サーバに送られます。

```
ice -S LOAD -f attrs -D LDAP -s www.novell.com -d cn=admin,o=novell -w
admin
```

次のコマンドを使用する際に前のテンプレートファイルを使用すると、前のコマンドで追加したすべてのレコードが削除されます。

```
ice -S LOAD -f attrs -r -D LDAP -s www.novell.com -d cn=admin,o=novell -w
admin
```

-m を使用して変更する場合は、次の例のようにレコードを変更します。

```
# ==============================================================================

#  DirLoad 1.00

# ==============================================================================
!COUNTER=300

!OBJECTCOUNT=2

#------------------------------------------------------------------------------

#  ATTRIBUTE TEMPLATE

# ------------------------------------------------------------------------------

dn: cn=$R(first),%.1s)($R(initial),%.1s)$R(last),ou=$R(ou),ou=dev,o=novell

delete: givenname

add: givenname
```

```
givenname: test1
replace: givenname
givenname: test2
givenname: test3
```

「attrs」ファイルが上のデータを格納しているときに次のコマンドを使用した場合の例を示します。

```
ice -S LOAD -f attrs -m -D LDIF -f new.ldf
```

LDIF データは次のような結果になります。

```
version: 1
dn: cn=BillTSmith,ou=ds,ou=dev,o=novell
changetype: modify
delete: givenname
-
add: givenname
givenname: test1
-
replace: givenname
givenname: test2
givenname: test3
-
dn: cn=JohnAWilliams,ou=ldap,ou=dev,o=novell
changetype: modify
delete: givenname
-
add: givenname
givenname: test1
-
replace: givenname
givenname: test2
givenname: test3
-
```

暗号化された属性を含む LDIF エクスポートを LDAP サーバから実行する

暗号化された属性を含む LDIF のエクスポートを LDAP サーバから実行するには、次のように、LDAP ソースハンドラおよび LDIF ターゲットハンドラを暗号化用のスキームおよびパスワードと組み合わせて使用します。

```
ice -S LDAP -s server1.acme.com -p 636 -L cert-server1.pem -d cn=admin,c=us
-w password -F objectClass=* -c sub -D LDIF -f server1.ldif -e des -E
secret
```

暗号化された属性を含む LDIF インポートを実行する

ICE によってあらかじめ暗号化されている属性を持つファイルの LDIF インポートを実行するには、次のように、LDIF ソースを、ファイルと LDAP ターゲットハンドラをエクスポートするために以前使用したスキームおよびパスワードと組み合わせて使用します。

```
ice -S LDIF -f server1.ldif -e des -E secret -D LDAP -s server2.acme.com -
p 636 -L cert-server2.pem -d cn=admin,c=us -w password
```

変換ルール

OpenText インポート / エクスポート変換エンジンは、ソースハンドラから受け取ったレコードをターゲットハンドラに送る前に、レコードに対して変換処理を行います。この変換処理の内容は一連のルールを使用して指定できます。これらのルールは XML で記述します (XML ファイルとして作成される場合と、XML 用ディレクトリ内に格納された XML データとして作成される場合があります)。このルールにより、LDAP ディレクトリ間でのエントリのインポート時に、次の問題が解決されます。

- 不足している情報
- 階層の違い
- スキーマの違い

次の 3 種類の変換ルールがあります。

ルール	説明
配置	エントリの位置を変更します。 たとえば、あるユーザグループをいったん「l=San Francisco, c=US」というコンテナにインポートし、インポートが終わった後で「l=Los Angeles, c=US」というコンテナに移す場合などに、配置ルールを利用できます。 これらのルールの形式については、203 ページの 「配置ルール」を参照してください。

ルール	説明
作成	インポート時にエントリを正しく作成するために必要な情報が欠落している場合にそれを補います。 たとえば、LDIF データのエクスポート元サーバのスキーマではユーザエントリに必要とされる属性が cn(commonName) 属性だけであるのに対し、LDIF データのインポート先サーバのスキーマでは cn 属性の他に sn(surname) 属性も必要とされる場合が考えられます。このような場合は作成ルールを使用すれば、インポート / エクスポート変換エンジンが各エントリを処理するときに、そのエントリにデフォルトの sn 値 (" " など) が設定されるようにすることができます。これにより、各エントリはインポート先サーバに送信されるときには必要な属性である sn 属性を持つことになり、エントリの正常な追加が保証されます。 これらのルールの形式については、201 ページの 「作成ルール」を参照してください。
スキーママッピング	サーバ間でデータを転送する場合 (直接転送するか LDIF を使用するかに関係なく) で、転送元サーバのスキーマと転送先サーバのスキーマが異なっている場合、次のようにスキーママッピングを使用できます。 ◆ エクスポート元サーバからインポートするエントリのオブジェクトクラスと属性タイプをすべて受け付けることができるように、インポート先サーバ上のスキーマを拡張します。 ◆ 転送元サーバのスキーマエレメントを、転送先サーバ上の同等ではあるが異なるスキーマエレメントにマッピングします。 これらのルールの形式については、199 ページの 「スキーママッピングルール」を参照してください。

これらの変換ルールは、OpenText eDirectory インポート / エクスポートウィザードとコマンドラインインタフェースのどちらでも利用できます。XML ルールの詳細については、199 ページの 「XML ルールを使用する」を参照してください。

コマンドラインインタフェースを使用する

コマンドラインバージョンで変換ルールを使用するには、OpenText インポート / エクスポート変換ユーティリティの実行ファイルを起動するときに、使用したいルールに対応する一般オプション (-p、-c、または -s) を指定します。詳細については、177 ページの 「全般オプション」を参照してください。

オプション	説明
-p *URL*	インポート / エクスポート変換エンジンが使用する XML 配置ルールが格納されている場所です。
-c *URL*	インポート / エクスポート変換エンジンが使用する XML 作成ルールが格納されている場所です。
-s *URL*	インポート / エクスポート変換エンジンが使用する XML スキーママッピングルールが格納されている場所です。

すべての 3 つのオプションで、*URL* を次のいずれかに指定します。

- 次の形式の URL:

  `file://[path/]filename`

  ファイルは、ローカルファイルシステムに存在する必要があります。
- ベースレベルの検索を指定する RFC 2255 準拠の LDAP URL と、1 つの値の属性タイプに対して 1 つの属性記述が定義されている属性リスト。

XML ルールを使用する

OpenText インポート / エクスポート変換ルールで使用される XML 形式は、OpenText Identity Manager の場合と同じです。OpenText Identity Manager の詳細については、*OpenText Identity Manager マニュアルサイト* (https://www.netiq.com/documentation/identity-manager-4.10/) を参照してください。

スキーママッピングルール

スキーママッピングルールの最上位エレメントは、<attr-name-map> です。インポートスキーマとエクスポートスキーマの相互関係は、マッピングルールによって決まります。マッピングルールは、指定されたインポートクラスの定義や属性を、対応するエクスポートスキーマの定義に関連付けます。

マッピングルールは、属性名またはクラス名に対応させて設定します。

- 属性マッピングの場合、マッピングルールでは、それが属性マッピングであること、ネームスペース (nds-name はソース名のタグ)、OpenText eDirectory ネームスペース内での名前、および他のネームスペース (app-name はターゲット名のタグ) とそのネームスペース内での名前を指定する必要があります。マッピングルールでは、マッピングが特定のクラスに適用されることを指定することも、その属性を持つすべてのクラスに適用されることを指定することもできます。
- クラスマッピングの場合、マッピングルールでは、それがクラスマッピングルールであること、ネームスペース (OpenText eDirectory またはアプリケーション) とそのネームスペース内での名前、およびその他のネームスペースとそのネームスペース内での名前を指定する必要があります。

スキーママッピングルールの正式な DTD 定義を次に示します。

```
<!ELEMENT attr-name-map (attr-name | class-name)*>

<!ELEMENT attr-name (nds-name, app-name)>
<!ATTLIST attr-name
          class-name    CDATA    #IMPLIED>

<!ELEMENT class-name (nds-name, app-name)>

<!ELEMENT nds-name (#PCDATA)>

<!ELEMENT app-name (#PCDATA)>
```

複数のマッピングエレメントをファイルに定義できます。各エレメントは、ファイルに定義されている順番で処理されます。1 つのクラスまたは属性を複数回マッピングした場合は、最初のマッピングが優先されます。

スキーママッピングルールの作成例を次に示します。

スキーマルール 1: 次のルールでは、ソースの surname 属性をターゲットの inetOrgPerson クラスの sn 属性にマッピングします。

```
<attr-name-map>
   <attr-name class-name="inetOrgPperson">
      <nds-name>surname</nds-name>
      <app-name>sn</app-name>
   </attr-name>
</attr-name-map>
```

スキーマルール 2: 次のルールでは、ソースの inetOrgPerson クラスの定義をターゲットの User クラスの定義にマッピングします。

```
<attr-name-map>
   <class-name>
      <nds-name>inetOrgPerson</nds-name>
      <app-name>User</app-name>
   </class-name>
</attr-name-map>
```

スキーマルール 3: 次の例では、2 種類のルールを定義します。1 つ目のルールでは、Surname 属性を使用するすべてのクラスについて、ソースの Surname 属性をターゲットの sn 属性にマッピングします。2 つ目のルールでは、ソースの inetOrgPerson クラスの定義をターゲットの User クラスの定義にマッピングします。

```
<attr-name-map>
   <attr-name>
      <nds-name>surname</nds-name>
      <app-name>sn</app-name>
   </attr-name>
   <class-name>
      <nds-name>inetOrgPerson</nds-name>
      <app-name>User</app-name>
   </class-name>
</attr-name-map>
```

コマンド例 : スキーマルールが sr1.xml ファイルに保存されている場合、次のコマンドを指定することにより、1entry.ldf ファイルの処理中にそのルールを使用すること、および結果をターゲットファイル outt1.ldf に送ることがインポート / エクスポート変換ユーティリティに指示されます。

```
ice -o -sfile://sr1.xml -SLDIF -f1entry.ldf -c -DLDIF
-foutt1.ldf
```

作成ルール

作成ルールによって、宛先ディレクトリ内に新規エントリを作成する場合の条件が指定されます。次のエレメントがサポートされます。

- **必須属性** すべての必須属性について、追加レコードに値が必要であること、値がない場合には追加が失敗することを指定します。作成ルールでは、必須属性のデフォルト値を指定できます。レコードに属性値がない場合、そのエントリにはデフォルト値が使用されます。レコードに属性値がある場合は、そのレコード値が使用されます。
- **一致属性** 追加レコードに特定の属性が必要であり、特定の値に一致すること、そうでない場合には追加が失敗することを指定します。
- **テンプレート** OpenText eDirectory 内のテンプレートオブジェクトの識別名を指定します。現時点では、OpenText インポート / エクスポート変換ユーティリティの作成ルールにテンプレートを指定することはできません。

作成ルールの正式な DTD 定義を次に示します。

```
<!ELEMENT create-rules (create-rule)*>

<!ELEMENT create-rule (match-attr*,
                       required-attr*,
                       template?) >
<!ATTLIST create-rule
          class-name    CDATA   #IMPLIED
          description   CDATA   #IMPLIED>

<!ELEMENT match-attr    (value)+ >
<!ATTLIST match-attr
          attr-name      CDATA    #REQUIRED>

<!ELEMENT required-attr (value)*>
<!ATTLIST required-attr
          attr-name     CDATA   #REQUIRED>

<!ELEMENT template EMPTY>
<!ATTLIST template
          template-dn   CDATA   #REQUIRED>
```

複数の作成ルールをファイルに定義できます。各ルールは、ファイルに定義されている順番で処理されます。ルールに適合しないレコードがあると、そのレコードはスキップされますが、レコードのスキップによるエラーは生成されません。

作成ルールの形式例を次に示します。

作成ルール 1: 次に紹介するルールでは、inetOrgPerson クラスの追加レコードに次の 3 つの条件が適用されます。追加レコードには、givenName 属性および Surname 属性が必要です。追加レコードには L 属性が必要ですが、この属性値がない場合には、作成ルールによってデフォルト値「Provo」に設定されます。

```
<create-rules>
   <create-rule class-name="inetOrgPerson">
      <required-attr attr-name="givenName"/>
      <required-attr attr-name="surname"/>
      <required-attr attr-name="L">
         <value>Provo</value>
      </required-attr>
   </create-rule>
</create-rules>
```

作成ルール 2: 次に紹介する作成ルールでは、ベースクラスの種類に関係なく、すべての追加レコードに次の 3 つの条件が適用されます。

- 追加レコードには、givenName 属性が必要です。この属性が含まれていない場合、追加は失敗します。
- 追加レコードには、Surname 属性が必要です。この属性が含まれていない場合、追加は失敗します。
- 追加レコードには、L 属性が必要です。この属性が含まれていない場合、L 属性はデフォルト値「Provo」に設定されます。

```
<create-rules>
   <create-rule>
      <required-attr attr-name="givenName"/>
      <required-attr attr-name="Surname"/>
      <required-attr attr-name="L">
         <value>Provo</value>
      </required-attr>
   </create-rule>
</create-rules>
```

作成ルール 3: 次に紹介する作成ルールでは、ベースクラスの種類に関係なく、すべての追加レコードに次の 2 つの条件が適用されます。

- 作成ルールは、レコードに uid 属性として ratuid が指定されているかチェックします。この属性が含まれていない場合、追加は失敗します。
- 作成ルールは、レコードに L 属性が指定されているかチェックします。この属性がない場合、L 属性はデフォルト値「Provo」に設定されます。

```
<create-rules>
   <create-rule>
      <match-attr attr-name="uid">
         <value>cn=ratuid</value>
      </match-attr>
      <required-attr attr-name="L">
         <value>Provo</value>
      </required-attr>
   </create-rule>
</create-rules>
```

コマンド例 : 作成ルールを crl.xml ファイルに保存し、次のコマンドを指定することにより、1entry.ldf ファイルの処理中にそのルールを使用すること、および結果をターゲットファイル outt1.ldf に送ることがインポート / エクスポート変換ユーティリティに指示されます。

```
ice -o -cfile://cr1.xml -SLDIF -f1entry.ldf -c -DLDIF
-foutt1.ldf
```

配置ルール

配置ルールによって、ターゲットディレクトリ内でエントリが作成される位置が決まります。配置ルールでは、次の 3 つの条件を使用して、エントリの配置にそのルールを適用すべきかどうかを決定します。

- **クラス一致 :** 配置ルールに match class エレメントが定義されている場合、レコードに定義されている objectClass は、ルールの class-name 属性に一致する必要があります。一致しない場合、そのレコードには配置ルールが使用されません。
- **属性一致 :** 配置ルールに match attribute エレメントが定義されている場合、レコードでは、match attribute エレメントに定義されている各属性について属性値が必要です。一致しない場合、そのレコードには配置ルールが使用されません。
- **パス一致 :** 配置ルールに match path エレメントが定義されている場合、レコードの dn 部分は、match path エレメントに定義されているプリフィックスに一致する必要があります。一致しない場合、そのレコードには配置ルールが使用されません。

ルールの最後のエレメントによって、エントリの配置場所が決まります。配置ルールでは、必要に応じて次のオプションを指定できます。

- **解析済み文字データ** 解析済み文字データを使用して、エントリに使用するコンテナの DN を指定します。
- **名前をコピー** 古い DN のネーミング属性を、エントリの新しい DN で使用することを指定します。
- **属性をコピー** エントリの新しい DN で使用するネーミング属性を指定します。指定されたネーミング属性は、エントリのベースクラスのネーミング属性として有効でなければなりません。
- **パスをコピー** ソース DN をターゲット DN として使用することを指定します。
- **パスサフィックスをコピー** ソース DN のパスの一部をターゲット DN として使用することを指定します。match-path エレメントを指定した場合、古い DN のパスの一部、つまり、match-path エレメントのプリフィックス属性に一致しない部分だけが、エントリの DN の一部として使用されます。

配置ルールの正式な DTD 定義を次に示します。

```
<!ELEMENT placement-rules (placement-rule*)>
<!ATTLIST placement-rules
          src-dn-format    (%dn-format;)   "slash"
          dest-dn-format   (%dn-format;)   "slash"
          src-dn-delims    CDATA           #IMPLIED
          dest-dn-delims   CDATA           #IMPLIED>

<!ELEMENT placement-rule (match-class*,
                          match-path*,
                          match-attr*,
                          placement)>
<!ATTLIST placement-rule
          description      CDATA            #IMPLIED>

<!ELEMENT match-class      EMPTY>
<!ATTLIST match-class
          class-name       CDATA            #REQUIRED>

<!ELEMENT match-path       EMPTY>
<!ATTLIST match-path
          prefix           CDATA            #REQUIRED>

<!ELEMENT match-attr       (value)+ >
<!ATTLIST match-attr
          attr-name        CDATA            #REQUIRED>

<!ELEMENT placement        (#PCDATA |
                            copy-name |
                            copy-attr |
                            copy-path |
                            copy-path-suffix)* >
```

複数の配置ルールエレメントをファイルに定義できます。各ルールは、ファイルに定義されている順番で処理されます。ルールに適合しないレコードがあると、そのレコードはスキップされますが、レコードのスキップによるエラーは生成されません。

配置ルールの形式例を次に示します。scr-dn-format="ldap" 属性および dest-dn-format="ldap" 属性によって、ソース DN およびターゲット DN のネームスペースが LDAP 形式として定義されます。

OpenText インポート / エクスポート変換ユーティリティがサポートするソース名およびターゲット名は、LDAP 形式だけです。

配置例 1: 次の配置ルールでは、レコードはベースクラス inetOrgPerson を持つ必要があります。レコードがこの条件に適合する場合、そのエントリは test コンテナの直下に置かれ、ソース DN の最上位コンポーネントがエントリの DN の一部として使用されます。

```
<placement-rules src-dn-format="ldap" dest-dn-format="ldap">
    <placement-rule>
       <match-class class-name="inetOrgPerson"></match-class>
       <placement>cn=<copy-name/>,o=test</placement>
    </placement-rule>
 </placement-rules>
```

ベースクラス inetOrgPerson および次の DN を持つレコードがあるとします。

```
dn: cn=Kim Jones, ou=English, ou=Humanities, o=UofZ
```

このレコードは、例に示したルールに従って、ターゲットディレクトリ内で次の DN を持ちます。

```
dn: cn=Kim Jones, o=test
```

配置例 2: 次の配置ルールでは、レコードは sn 属性を持つ必要があります。レコードがこの条件に適合する場合、そのエントリは test コンテナの直下に置かれ、ソース DN の最上位コンポーネントがエントリの DN の一部として使用されます。

```
<placement-rules src-dn-format="ldap" dest-dn-format="ldap">
    <placement-rule>
       <match-attr attr-name="sn"></match-attr>
       <placement>cn=<copy-name/>,o=test</placement>
    </placement-rule>
 </placement-rules>
```

次の DN および sn 属性を持つレコードがあるとします。

```
dn: cn=Kim Jones, ou=English, ou=Humanities, o=UofZ
sn: Jones
```

このレコードは、例に示したルールに従って、ターゲットディレクトリ内で次の DN を持ちます。

```
dn: cn=Kim Jones, o=test
```

配置例 3: 次の配置ルールでは、レコードは sn 属性を持つ必要があります。レコードがこの条件に適合する場合、そのエントリは test コンテナの直下に置かれ、sn 属性がエントリの DN の一部として使用されます。copy-attr エレメントに指定された属性は、エントリのベースクラスのネーミング属性でなければなりません。

```
<placement-rules src-dn-format="ldap" dest-dn-format="ldap">
  <placement-rule>
    <match-attr attr-name="sn"></match-attr>
    <placement>cn=<copy-attr attr-name="sn"/>,o=test</placement>
  </placement-rule>
</placement-rules>
```

次の DN および sn 属性を持つレコードがあるとします。

```
dn: cn=Kim Jones, ou=English, ou=Humanities, o=UofZ
sn: Jones
```

このレコードは、例に示したルールに従って、ターゲットディレクトリ内で次の DN を持ちます。

```
dn: cn=Jones, o=test
```

配置例 4: 次の配置ルールでは、レコードは sn 属性を持つ必要があります。レコードがこの条件に適合する場合、ソース DN がターゲット DN として使用されます。

```
<placement-rules src-dn-format="ldap" dest-dn-format="ldap">
    <placement-rule>
       <match-attr attr-name="sn"></match-attr>
       <placement><copy-path/></placement>
    </placement-rule>
 </placement-rules>
```

配置例 5: 次の配置ルールでは、レコードは sn 属性を持つ必要があります。レコードがこの条件に適合する場合、エントリの DN 全体が test コンテナにコピーされます。

```
<placement-rules src-dn-format="ldap" dest-dn-format="ldap">
    <placement-rule>
       <match-attr attr-name="sn"></match-attr>
       <placement><copy-path-suffix/>,o=test</placement>
    </placement-rule>
 </placement-rules>
```

次の DN および sn 属性を持つレコードがあるとします。

```
dn: cn=Kim Jones, ou=English, ou=Humanities, o=UofZ
sn: Jones
```

このレコードは、例に示したルールに従って、ターゲットディレクトリ内で次の DN を持ちます。

```
dn: cn=Kim Jones, ou=English, ou=Humanities, o=UofZ, o=test
```

配置例 6: 次の配置ルールでは、レコードは sn 属性を持つ必要があります。レコードがこの条件に適合する場合、エントリの DN 全体が neworg コンテナにコピーされます。

```
<placement-rules>
 <placement-rule>
  <match-path prefix="o=engineering"/>
  <placement><copy-path-suffix/>o=neworg</placement>
 </placement-rule>
</placement-rules>
```

次に例を示します。

```
dn: cn=bob,o=engineering
```

は、次のようになります。

```
dn: cn=bob,o=neworg
```

コマンド例 : 配置ルールが pr1.xml ファイルに保存されている場合、次のコマンドを指定することにより、1entry.ldf ファイルの処理中にそのルールを使用すること、および結果をターゲットファイル foutt1.ldf に送ることがインポート / エクスポート変換ユーティリティに指示されます。

```
ice -o -pfile://pr1.xml -SLDIF -f1entry.ldf -c -DLDIF
-foutt1.ldf
```

LBURP (LDAP Bulk Update/Replication Protocol)

OpenText インポート / エクスポート変換ユーティリティでは、LDAP サーバへの非同期要求の送信に、LBURP を使用します。これにより、要求は常にプロトコルで指定された順序で処理されます。複数プロセッサ間の相互関係やオペレーティングシステムのスケジューラの設定によって処理順序が変わることはありません。

LBURP により、OpenText インポート / エクスポート変換ユーティリティは、複数の更新操作を 1 つの要求として送信したり、これらすべての更新操作に対する応答を 1 つのレスポンスとして受け取ることができます。これにより、プロトコルのネットワーク処理効率が向上します。

LBURP は次のように機能します。

1. OpenText インポート / エクスポート変換ユーティリティが LDAP サーバにバインドします。
2. サーバからクライアントにバインドレスポンスが送られます。
3. クライアントからサーバに開始 LBURP 拡張要求が送られます。
4. サーバからクライアントに開始 LBURP 拡張レスポンスが送られます。
5. 必要に応じてクライアントからサーバに LBURP 操作拡張要求が送られます。

   これらの要求は非同期で送信することもできます。要求ごとに通し番号が付けられ、同じクライアントから同じ接続を介して送信された個々の要求の順序はこの通し番号によって特定されます。各要求には、少なくとも 1 つの LDAP 更新操作が設定されます。
6. サーバは、受け取った各 LBURP 操作拡張要求を通し番号に従って順番に処理し、要求ごとに LBURP 操作拡張レスポンスを送信します。
7. サーバへの更新操作の送信がすべて終了すると、クライアントはサーバに終了 LBURP 拡張要求を送ります。
8. サーバからクライアントに終了 LBURP 拡張レスポンスが送られます。

LBURP プロトコルにより、OpenText インポート / エクスポート変換機能は、サーバにデータを転送するときに送信元と送信先の間のネットワーク接続の限界まで転送速度を上げることができます。ネットワーク接続が十分に高速であれば、OpenText インポート / エクスポート変換機能から要求が送られてくるのを待つ必要がないため、サーバはすべての処理時間を更新操作の処理だけに費やすことができます。

更新操作の処理効率をさらに上げるため、OpenText eDirectory の LBURP プロセッサは、データベースへの更新操作をグループに分けて行います。LBURP の採用により、従来の同期処理の場合と比べて、LDIF のインポート処理の効率は大幅に改善されています。

LBURP はデフォルトで有効になっていますが、LDIF のインポート中に無効にすることもできます。

コマンドラインオプションを使用して、LDIF のインポート中に LBURP の有効 / 無効を切り替えることができます。詳細については、184 ページの 「-B」 を参照してください。

LDIF のインポートを高速化する

1 つの LDIF ファイルに数千または数百万のレコードがある場合は、次のことを検討してください。

- 208 ページの 「読み書き可能レプリカを持つサーバに直接インポートする」
- 208 ページの 「LBURP を使用する」
- 208 ページの 「データベースキャッシュを設定する」
- 209 ページの 「簡易パスワードを使用する」
- 209 ページの 「インデックスを使用する場合の注意」

読み書き可能レプリカを持つサーバに直接インポートする

実行が可能な場合は、LDIF ファイルで示されているすべてのエントリを含む、読み書き可能レプリカを持つサーバを LDIF のインポート先に選択します。これによりネットワーク効率を大幅に高めることができます。

更新時には、インポート先サーバから他の OpenText eDirectory サーバへのチェーン接続は行わないでください。これにより、パフォーマンスはかなり低下します。ただし、一部の更新対象エントリが LDAP を実行していない OpenText eDirectory サーバ上だけに存在する場合には、LDIF ファイルをインポートするためにチェーン接続が必要になることもあります。

レプリカとパーティション管理の詳細については、159 ページの第 6 章「パーティションおよびレプリカの管理」を参照してください。

LBURP を使用する

インポート / エクスポート変換機能では、ネットワークと OpenText eDirectory サーバの処理をできるだけ効率化するために、ウィザードとサーバの間でのデータ転送に LBURP を使用します。LDIF のインポート時に LBURP を使用することにより、LDIF のインポートにかかる時間が大幅に短縮されます。

LBURP の詳細については、207 ページの 「LBURP (LDAP Bulk Update/Replication Protocol)」を参照してください。

データベースキャッシュを設定する

OpenText eDirectory で使用できるデータベースキャッシュの容量は、LDIF インポートの処理速度に大きく影響します。特に、サーバ上のエントリの総数が多いほど影響は大きくなります。LDIF のインポートでは、インポート実行中にはできるだけ多くのメモリを OpenText eDirectory に割り当てると効率的です。インポートが完了してサーバの負荷が通常レベルに戻ったら、メモリの設定を元にもどすことができます。この方法は、OpenText eDirectory サーバで実行する処理がインポートだけの場合に特に効果があります。

OpenText eDirectory データベースキャッシュの設定の詳細については、583 ページの第 19 章「OpenText eDirectory のメンテナンス」を参照してください。

簡易パスワードを使用する

OpenText eDirectory では、パブリックキーとプライベートキーのペアを使用して認証を行います。これらの鍵の生成は、CPU に大きな負担のかかる処理です。バージョン 8.7.3 以降では、パスワードの格納に、NMAS (OpenText モジュラー認証サービス) の簡易パスワード機能を使用できます。簡易パスワードを使用する場合、パスワードはディレクトリ内の安全な場所に保持されますが、鍵のペアはサーバ間での認証で実際に必要になるまで生成されません。これにより、パスワード情報を持つオブジェクトをロードする速度を大幅にアップできます。

コマンドラインバージョンの OpenText インポート / エクスポート変換ユーティリティを使用して、LDIF インポートを実行できます (176 ページの 「コマンドラインインタフェースを使用する」を参照)。

パスワードの格納に簡易パスワードを使用する場合は、OpenText eDirectory ツリーへのログインおよび従来型のファイルサービスやプリントサービスへのアクセスには、NMAS 対応の Novell Client を使用する必要があります。またサーバには NMAS がインストールされている必要があります。名前とパスワードのバインドを行う LDAP アプリケーションは、簡易パスワード機能とスムーズに連携します。

NMAS の詳細については、667 ページの第 24 章「OpenText eDirectory の認証フレームワークについて」を参照してください。

インデックスを使用する場合の注意

不要なインデックスがあると、LDIF のインポートにかかる時間が長くなります。これは、定義されているすべてのインデックスで、設定されている属性値を持つエントリごとに追加の処理が実行されるためです。LDIF をインポートする前に、不要なインデックスがないことを確認します。インデックスを作成するときは、あらかじめデータ確認済みのプレディケート統計をロードしてインデックスが本当に必要な箇所を確認すると、不要なインデックスを減らすことができます。

インデックスの調整の詳細については、209 ページの 「インデックスマネージャ」を参照してください。

インデックスマネージャ

インデックスマネージャは、サーバオブジェクトの属性の 1 つで、データベースインデックスの管理に使用します。OpenText eDirectory では、データベースインデックスを使用することによって、クエリの処理速度が大幅に向上します。

OpenText eDirectory には、基本的なクエリの機能を提供する一連のインデックスが付属しています。これらデフォルトのインデックスの対象となる属性を次に示します。

CN	Aliased Object Name
dc	破損通知
名	Member (メンバー)

Surname (名字)	リファレンス
uniqueID (固有 ID)	Equivalent to Me
GUID	NLS: 共通証明書
cn_SS	Revision (改訂数)
uniqueID_SS	extensionInfo
ldapAttributeList	ldapClassList

またカスタマイズされたインデックスを作成して、ユーザの環境における OpenText eDirectory のパフォーマンスをさらに向上させることができます。たとえば、デフォルトでインデックス付けされていない属性を検索する新しい LDAP アプリケーションが組織に導入された場合、その属性に対するインデックスを作成すると便利です。

注 : インデックスを使用することにより検索の処理速度は上がりますが、インデックスの数が増えるほど更新にかかる時間が長くなります。一般には、パフォーマンスの問題が特定のディレクトリの検索に関係すると思われる場合に、新しいインデックスを作成します。

OpenText Identity Console を使用して、インデックスを作成または削除できます。インデックス名、状態、タイプ、ルール、インデックス付き属性など、インデックスのプロパティを表示したり、管理することができます。

インデックスを作成する

1. OpenText Identity Console のホームページで、[**インデックス管理**] をクリックします。
2. [**インデックスの作成** +] をクリックします。
3. [**インデックスの作成**] ページで、[**名前**] フィールドに名前を入力します。
4. [**サーバの選択**] リストからサーバを選択します。
5. [**属性の選択**] リストから属性を選択します。
6. インデックスのルールを選択します。
   - **値** 属性の値全体または値の最初の部分を照合します。たとえば、値一致は、「Jensen」に一致する「LastName」のあるエントリの検索や、「Jen」で始まる「LastName」があるエントリの検索に使用できます。
   - **存在** 特定の属性値ではなく、属性の存在のみを検索します。Login Script 属性を持つエントリをすべて検索するクエリは、存在インデックスを使用します。
   - **部分文字列** 属性値文字列のサブセットを照合します。たとえば、「der」という値を含む「LastName」を検索するクエリを実行すると、「Derington」、「Anderson」、および「Lauder」が照合の結果として返されます。

     下位文字列インデックスは、作成や維持を行うときに最も多くのリソースが消費されるインデックスです。
7. **作成**をクリックします。

「次のサーバでインデックスが正常に作成されました : 1 サーバ (Index Created successfully on 1 server)」というメッセージが表示されます。

リンバをバックグラウンドプロセスとして再起動し、変更を開始します。

重要 :「$」文字は属性値の区切り記号として使用されます。インデックス名に「$」文字を使用する場合、前に円記号 (\) を付けて、LDAP でインデックスを作成するときに「$」文字をエスケープします。

インデックスを削除する

作成したインデックスが不要になる場合があります。必要のない、ユーザ定義または自動で作成したインデックスは、削除できます。

1. OpenText Identity Console のホームページで、[インデックス管理] をクリックします。
2. [インデックス] ページで、削除するユーザインデックスまたは自動追加インデックスを選択します。
3. [削除 ] をクリックすると、インデックステーブルが更新されます。
4. [削除の確認] ポップアップで [OK] をクリックして、削除を確認します。

   リンバをバックグラウンドプロセスとして再起動し、変更を開始します。

インデックスをオフラインにする

一時的にインデックスをオフラインにすることで、処理のピーク時にパフォーマンスを調整できます。たとえば、ユーザ定義のインデックスの使用をすべて中断すると、バルクロードを高速化できます。オブジェクトを追加または変更するときは定義されているインデックスを更新する必要があり、すべてのインデックスをアクティブにするとデータのバルクロードの速度が遅くなるためです。バルクロードが完了すると、再びインデックスをオンラインにできます。

1. OpenText Identity Console のホームページで、[インデックス管理] をクリックします。
2. [インデックス] ページで、オフラインにするインデックスのチェックボックスを選択し、[状態 (State)] ドロップダウンメニューをクリックして、[オフライン] を選択します。

   表示されているテーブルでは、インデックスの状態が [オンライン] から [オフライン] に変わります。インデックスは、次のいずれかの状態になります。

   - **オンライン** : 現在実行中 .
   - **オフライン** : [一時停止中] . インデックスを再開するには、[Bring Online] をクリックします。
   - **New (新規)** : [オンライン] 状態になるのを待機中 .
   - **削除済み** : インデックステーブルから削除されるのを待機中 .

他のサーバ上でインデックスを管理する

あるサーバで便利に使用されているインデックスがあり、このインデックスを他のサーバでも使用する場合は、他のサーバにインデックス定義をコピーできます。またプレディケートデータを調べると、これとは逆のケースが発生する場合もあります。つまり、複数のサーバで使用されていたインデックスが、そのいずれかのサーバで不要になるといったケースです。このような場合、インデックスが不要になったサーバからインデックスを削除できます。

インデックスマネージャを使用すると、他のインスタンスに影響を与えずに、インデックスの 1 つのインスタンスを処理できます。

1 OpenText Identity Console のホームページで、[**インデックス管理**] をクリックします。

2 [**インデックス**] ページで、変更するインデックスを選択します。

インデックスを 1 つ選択すると、そのインデックスを提供するツリー内のサーバが一覧表示されます。

3 [**インデックスの変更**] ページで、追加のサーバを選択します。

4 [**保存**] をクリックします。

「**インデックスが正常に変更されました！**」というメッセージが表示されます。

OpenText eDirectory Service Manager

OpenText eDirectory Service Manager では、使用可能な OpenText eDirectory サービスおよびそのステータスについての情報が提供されます。また、Service Manager からこれらのサービスを開始または停止できます。

Service Manager は OpenText eDirectory サービスのみを管理します。dsservcfg.xml 設定ファイルを使用して管理を行います。このファイルには、各プラットフォームで管理できるサービスを表示します。リストにサービスを追加または削除することもできます。

OpenText eDirectory Service Manager には次の方法でアクセスできます。

- 212 ページの 「クライアントのサービスマネージャ eMTool を使用する」

クライアントのサービスマネージャ eMTool を使用する

OpenText eDirectory Management Toolbox (eMBox) クライアントはコマンドラインで実行される Java クライアントで、OpenText eDirectory Service Manager eMTool にリモートアクセスできます。emboxclient.jar ファイルは、OpenText eDirectory の一部としてサーバにインストールされます。JVM をインストールしていれば、どのコンピュータでも実行できます。クライアントの詳細については、618 ページの 「コマンドラインクライアントの使用」を参照してください。

クライアントのサービスマネージャ eMTool を使用するには、次の操作を行います。

1 コマンドラインで次のように入力して、対話式モードでクライアントを実行します。

```
java -cp path_to_the_file/emboxclient.jar  -i
```

(クラスパスに emboxclient.jar ファイルがすでに含まれている場合は、単に「java -i」と入力します。)

クライアントのプロンプトが次のように表示されます。

```
Client>
```

2 Service Manager を実行するサーバにログインするには、次のコマンドを入力します。

```
login -s server_name_or_IP_address -p port_number
-u username.context -w password -n
```

ポート番号は通常 80 または 8028 です。ただし、すでにそのポートを使用している Web サーバが存在する場合は異なります。-n オプションを使用すると、非セキュア接続が開始されます。

クライアントにログインが成功したかどうかが表示されます。

3 次のいずれかの Service Manager コマンドを入力します。

コマンド	説明
service.serviceList	利用できる OpenText eDirectory サービスを表示します。
service.serviceStart -n *Module_name*	指定された OpenText eDirectory サービスを開始します。
service.serviceStop -n *Module_name*	指定された OpenText eDirectory サービスを停止します。
service.serviceInfo -n *Module_name*	指定されたサービスに関する情報を表示します。

クライアントで list -t service コマンドを使用して、Service Manager オプションの詳細を表示することもできます。詳細については、622 ページの 「eMTool とそのサービスを表示する」を参照してください。

4 クライアントからログアウトするには、次のコマンドを入力します。

```
logout
```

5 クライアントを終了するには、次のコマンドを入力します。

```
exit
```

オフラインのバルクロードユーティリティ

ldif2dib ユーティリティでは、OpenText eDirectory サーバがオフラインになったときに、LDIF ファイルから OpenText eDirectory データベース (DIB) へのデータのバルクロードを実行できます。OpenText eDirectory は、Linux と Windows の両方のプラットフォーム上でこのユーティリティをサポートします。これはオフラインユーティリティであり、他のオンラ

インツールより高速でバルクロードを実行できます。このユーティリティは、LDIF ファイルから DIB にエントリをインポートするときに、既存のディレクトリを使用し、データベースを新規作成しません。

LDIF ファイルのエントリを大規模なユーザデータベースに取り込む必要があるときには、ldif2dib ユーティリティが必要です。ice や ldapmodify などのオンラインツールはオンラインでのバルクロード時にスキーマチェックやプロトコル変換やアクセス制御チェックなどのオーバーヘッド処理を伴うため、ldif2dib より処理が遅くなります。ldif2dib は、ユーザの大きなデータベースを構築し、初期のダウンタイムが問題とならないときは、高速な動作可能時間を実現します。

eDirectory のパフォーマンスの改善

OpenText eDirectory には、バルクロードのパフォーマンスを向上するための新しいオプションが追加されました。OpenText インポート / エクスポート変換 (ICE) ユーティリティを使用したバルクロードのパフォーマンスを向上するために調整できるパラメータは、以下のとおりです。

- 214 ページの 「OpenText eDirectory キャッシュの設定」
- 214 ページの 「LBURP トランザクションサイズの設定」
- 215 ページの 「ICE の非同期要求の数を増やす」
- 216 ページの 「LDAP 書き込みスレッド数の増加」
- 216 ページの 「ICE のスキーマ検証を無効にする」
- 216 ページの 「バックリンカ」
- 216 ページの 「ACL テンプレートを無効にする」
- 218 ページの 「インラインキャッシュを有効 / 無効にする」
- 218 ページの 「LBURP のタイムアウト周期を拡大する」

各オペレーティングシステムの調整可能パラメータも参照してください。

OpenText eDirectory キャッシュの設定

バルクロードのパフォーマンスを最適にするには、OpenText eDirectory キャッシュの割り当てで、ブロックキャッシュにより高い割り当て率を設定します。詳細については、「OpenText eDirectory サブシステムのチューニング」を参照してください。

LBURP トランザクションサイズの設定

LBURP トランザクションサイズによって、1 つのトランザクションにおいて ICE から LDAP サーバに送信されるレコード数が設定されます。十分なメモリがあり、この値を大きくしても I/O 競合が発生しない場合、この値を大きくすることでバルクロードのパフォーマンスを向上できます。デフォルトの LBURP トランザクションサイズは 25 です。この値は LDIF ファイルが少ない (操作数が 100,000 より少ない) 場合には適切ですが、レコード数が多い場合には不適切です。LBURP トランザクションサイズは、1 ～ 350 の範囲で設定できます。

トランザクションサイズの変更

トランザクションサイズを変更するには、/etc/opt/novell/eDirectory/conf/nds.conf ファイルで n4u.ldap.lburp.transize パラメータの値を変更します。理想的なシナリオでは、トランザクションサイズが大きいほど、パフォーマンスはより高くなります。ただし、次の理由のため、トランザクションサイズには必要以上に大きな値を設定しないようにします。

- トランザクションサイズが大きいほど、サーバはトランザクションを処理するためにより多くのメモリを割り当てる必要があります。システムが少ないメモリで稼働している場合、スワッピングのために処理が遅くなることがあります。
- LDIF ファイルにエラーがなく、OpenText eDirectory にある既存のエントリがコメント化されていることが必要です。トランザクションに 1 つでもエラーがあると (追加しようとしたオブジェクトがすでにディレクトリに存在するという場合も含めて)、OpenText eDirectory は LBURP トランザクション設定を無視し、操作ごとにコミットを実行してデータの整合性を確認します。

  詳細については、899 ページの付録 H「トラブルシューティング」を参照してください。
- LBURP の最適化は、リーフオブジェクトに対してのみ有効です。トランザクションにコンテナオブジェクトとその従属オブジェクトが含まれている場合、OpenText eDirectory はこれをエラーとみなします。これを回避するには、最初に別の LDIF ファイルを使用してコンテナオブジェクトをロードするか、前方向参照の使用を有効にします。

  詳細については、899 ページの付録 H「トラブルシューティング」の「前方向参照の有効化」を参照してください。

ICE の非同期要求の数を増やす

この数値は、LDAP サーバから返される結果を待たずに、ICE クライアントから LDAP サーバに非同期に送信できるエントリの数を示します。非同期要求の数は、10 ～ 200 の範囲内で設定できます。デフォルト値は「100」です。最小値 (10) よりも小さい値はデフォルトに戻されます。小さな LDIF ファイルには最小値が適切です。理想的なシナリオでは、ウィンドウサイズが大きいほど、パフォーマンスはより高くなります。ただし、ウィンドウサイズが大きくなるほど、クライアントが LDIF ファイルのエントリを処理するために割り当てるメモリの量が多くなるため、ウィンドウサイズを必要以上に大きく設定しないでください。システムが少ないメモリで稼働している場合、スワッピングのために処理が遅くなることがあります。ICE 内の非同期要求の数を変更するには、ICE コマンドラインオプションを使用します。

ICE コマンドラインオプションを使用する場合

非同期要求の数は、ICE コマンドラインオプション –Z を使って指定できます。これは、LDAP ターゲットハンドラの一部として使用できます。

ICE クライアントによって送信される非同期要求の数を 50 に設定するには、次のコマンドを入力します。

```
ice -SLDIF -f LDIF_file -a -c -DLDAP -d cn_of_admin -Z50 -w password
```

LDAP 書き込みスレッド数の増加

LDAP サーバで複数の書込みスレッドを使えるようになりました。同時処理によって発生するエラーを避けるために前方参照を有効にするには、次のように ICE コマンドラインオプション –F を使用します。

```
ice -SLDIF -f LDIF_file -a -c -DLDAP -d cn_of_admin -w password  -F
```

ICE のスキーマ検証を無効にする

ICE クライアントのスキーマ検証を無効にするには、次のように ICE コマンドラインオプション –C と –n を使用します。

```
ice -C -n -SLDIF -f LDIF_file -a -c -DLDAP -d cn_of_admin -w password
```

バックリンカ

バックリンクは、特に参照整合性をチェックするバックグラウンドプロセスであり、OpenText eDirectory サーバが起動してから 50 分後に実行されます。その後は 13 時間後に実行されます。バルクロードの処理中にバックリンカが実行されないように注意してください。ロードされるオブジェクトの数やオブジェクトがロードされる回数によっては、バックリンカが実行されると、バルクロードの処理速度が低下することがあります。

ACL テンプレートを無効にする

バルクロードのパフォーマンスを向上させるために、ACL(アクセス制御リスト) テンプレートを無効にすることができます。これによりいくつかの ACL が見つからなくなりますが、必要な ACL を LDIF ファイルに追加するか、それらの ACL を後から適用することで、この問題は解決できます。

1 次のコマンドを実行します。

```
ldapsearch -D cn_of_admin -w password -b cn=schema -s base
objectclasses=inetorgperson
```

このコマンドの出力は次のようになります。

```
dn: cn=schema
objectClasses: ( 2.16.840.1.113730.3.2.2 NAME 'inetOrgPerson' SUP
 organizationalPerson STRUCTURAL MAY ( groupMembership $
ndsHomeDirectory
 $ loginAllowedTi meMap $ loginDisabled $ loginExpirationTime $
 loginGraceLimit $ loginGraceRem aining $ loginIntruderAddress $
 loginIntruderAttempts $ loginIntruderResetTim e $
 loginMaximumSimultaneous $ loginScript $ loginTime $
 networkAddressRestri ction $ networkAddress $ passwordsUsed $
 passwordAllowChange $ passwordExpirationInterval $
 passwordExpirationTime $passwordMinimumLength $ passwordRequired $
 passwordUniqueRequired $ printJobConfiguration $ privateKey $ Profile
$
 publicKey $ securityEquals $ accountBalance $ allowUnlimitedCredit $
 minimum AccountBalance $ messageServer $ Language $ UID $
```

```
 lockedByIntruder $ serverHolds $ lastLoginTime $ typeCreatorMap $
 higherPrivileges $ printerControl $ securityFlags $ profileMembership
$
 Timezone $ sASServiceDN $ sASSecretStore $ sASSecretStoreKey $
 sASSecretStoreData $ sASPKIStoreKeys $ userCertificate
 $nDSPKIUserCertificateInfo $ nDSPKIKeystore $ rADIUSActiveConnections
$
 rADIUS AttributeLists $ rADIUSConcurrentLimit $
rADIUSConnectionHistory
 $ rADIUSDefa ultProfile $ rADIUSDialAccessGroup $
rADIUSEnableDialAccess
 $ rADIUSPassword $ rADIUSServiceList $ audio $ businessCategory $
 carLicense $ departmentNumbe r $ employeeNumber $ employeeType $
 givenName $ homePhone $ homePostalAddress  $ initials $ jpegPhoto $
 labeledUri $ mail $ manager $ mobile $ pager $ ldap Photo $
 preferredLanguage $ roomNumber $ secretary $ uid $ userSMIMECertifica
te
 $ x500UniqueIdentifier $ displayName $ userPKCS12 ) X-NDS_NAME 'User'
X
-NDS_NOT_CONTAINER '1' X-NDS_NONREMOVABLE '1' X-NDS_ACL_TEMPLATES (
'2#subtree#[Self]#[All Attributes Rights]' '6#entry#[Self]#loginScript'
'1#subtree#[Root Template]#[Entry Rights]'
'2#entry#[Public]#messageServer' '2#entry#[Root
Template]#groupMembership' '6#entry#[Self]#printJobConfiguration'
'2#entry#[Root  Template]#networkAddress') )
```

2 この出力から、太字で示されている情報を削除します。

3 変更を加えた出力を LDIF ファイルとして保存します。

4 新しく保存した LDIF ファイルに次の情報を追加します。

```
dn: cn=schema
changetype: modify
delete: objectclasses
objectclasses: ( 2.16.840.1.113730.3.2.2 )-add:objectclasses
```

これにより、新しい LDIF は次のようになります。

```
dn: cn=schema
changetype: modify
delete: objectclasses
objectclasses: ( 2.16.840.1.113730.3.2.2)
-
add:objectclasses
objectClasses: ( 2.16.840.1.113730.3.2.2 NAME 'inetOrgPerson' SUP
 organization alPerson STRUCTURAL MAY ( groupMembership $
ndsHomeDirectory
 $ loginAllowedTimeMap $ loginDisabled $ loginExpirationTime $
 loginGraceLimit $ loginGraceRem aining $ loginIntruderAddress $
 loginIntruderAttempts $ loginIntruderResetTime $
 loginMaximumSimultaneous $ loginScript $ loginTime $
 networkAddressRestri ction $ networkAddress $ passwordsUsed $
 passwordAllowChange $ passwordExpirationInterval $
 passwordExpirationTime $ passwordMinimumLength $ passwordRequired
 $passwordUniqueRequired $ printJobConfiguration $ privateKey $ Profile
$
```

```
 publicKey $ securityEquals $ accountBalance $ allowUnlimitedCredit $
 minimum AccountBalance $ messageServer $ Language $ UID $
 lockedByIntruder $ serverHolds $ lastLoginTime $ typeCreatorMap $
 higherPrivileges $ printerControl $ securityFlags $ profileMembership
$
 Timezone $ sASServiceDN $ sASSecretStore $ sASSecretStoreKey $
 sASSecretStoreData $ sASPKIStoreKeys $ userCertificate $
 nDSPKIUserCertificateInfo $ nDSPKIKeystore $ rADIUSActiveConnections $
 rADIUSAttributeLists $ rADIUSConcurrentLimit $ rADIUSConnectionHistory
$
 rADIUSDefa ultProfile $ rADIUSDialAccessGroup $ rADIUSEnableDialAccess
 $rADIUSPassword $ rADIUSServiceList $ audio $ businessCategory $
 carLicense
 $ departmentNumbe r $ employeeNumber $ employeeType $ givenName $
 homePhone $ homePostalAddress  $ initials $ jpegPhoto $ labeledUri $
mail
 $ manager $ mobile $ pager $ ldap Photo $ preferredLanguage $
roomNumber
 $ secretary $ uid $ userSMIMECertifica te $ x500UniqueIdentifier $
 displayName $ userPKCS12 ) X-NDS_NAME 'User' X-ND S_NOT_CONTAINER '1'
X
-NDS_NONREMOVABLE '1')
```

5 次のコマンドを入力します。

```
ldapmodify -D cn_of_admin -w password -f LDIF_file_name
```

ACL での作業の詳細については、『OpenText™ eDirectory CE 24.4 (v9.3) チューニングガイド』を参照してください。

インラインキャッシュを有効 / 無効にする

サーバのインラインキャッシュ変更を有効または無効にできます。インラインキャッシュ変更は、アウトバウンド同期が無効になっている場合のみ、無効にできます。[アウトバウンド同期] を有効にすると、[インラインキャッシュ変更] も有効になります。[インラインキャッシュ変更] を無効にすると、このレプリカの変更キャッシュに無効のマークが付けられ、**[エージェント環境設定] > [パーティション]** で無効のフラグが付けられます。インラインキャッシュ変更を有効にすると、変更キャッシュの再構築時に、無効な変更キャッシュのフラグが削除されます。

LBURP のタイムアウト周期を拡大する

デフォルトでは、クライアントのタイムアウト周期は 20 分 (1200 秒) です。ただし、バルクロードの処理の際、LBURP トランザクションサイズが 250 であり、非常に大きな値を持つ多数の属性が対象であり、しかも、サーバで同時 LBURP 処理が有効になっている場合、

サーバは ICE クライアントによって送信されるデータを処理するために使用中になり、クライアントへの応答は規定の時間内に行われません。この場合、ICE クライアントはタイムアウトになります。

このため、タイムアウト周期を拡大することをお勧めします。拡大するには、より大きい値 (秒単位) を指定して環境変数 LBURP_TIMEOUT をエクスポートします。たとえば、1200 秒を指定して LBURP_TIMEOUT 変数をエクスポートするには、「export ICE_LBURP_TIMEOUT=1200」と入力します。

バルクロードに ldif2dib を使用する

コマンドラインインタフェースで、インポートするデータが含まれている LDIF ファイルと、データをインポートする必要のあるデータベースファイルのパスを指定できます。ldif2dib を使用してデータのバルクロードを実行するには、次の操作を行います。

1. DIB のバックアップを作成します。

   バックアップと復元のプロセスの詳細については、459 ページの第 15 章「OpenText eDirectory のバックアップと復元」を参照してください。

2. OpenText eDirectory サーバを停止します。

3. LDIF ファイルからのバルクロードを開始するには、コマンドプロンプトで次のコマンドを入力します。

   ```
   ldif2dib <LDIF File Name> [Options]
   ```

   各要素の内容は次のとおりです。

   - **LDIF File Name:** バルクロードを実行する LDIF ファイルの名前を指定します。
   - **Options:** これらはオプションです。このユーティリティを調整するために使用できる各種パラメータを指定します。ldif2dib ユーティリティでサポートされるオプションは以下のとおりです。

   たとえば、バッチモード、キャッシュサイズ、およびブロックキャッシュパーセンテージオプションを指定するためのオプションを設定するには、次のコマンドを入力します。

   ```
   ldif2dib 1MillionUsers.ldif -b/novell/log/logfile.txt -c314572800 -p90
   ```

ヒント : バルクロード処理を一時的に停止するには、s キーまたは S キーを押します。バルクロード処理を停止するには、エスケープ (Esc) キーを使用します。

複数のインスタンス

ldif2dib を使用して、LDIF ファイルから OpenText eDirectory (DIB) の特定のインスタンスへのエントリのバルクロードを実行するには、-n オプションで nds.db ファイルの場所を指定します。-n オプションで nds.db ファイルの場所を指定せず、システムで OpenText eDirectory のインスタンスが 1 つだけ構成されている場合は、ldif2dib がそのデータベースファイルの

場所を自動的に検出します。ただし、インスタンスが複数ある場合は、ldif2dib を実行するとメニューが表示されて構成済みのすべてのインスタンスがリストされ、バルクロードを実行するインスタンスを選択できます。

OpenText eDirectory の複数のインスタンスの詳細については、『OpenText™ eDirectory CE 24.4 (v9.3) インストールガイド』の「ndsconfig の使用による OpenText eDirectory 9.3 の複数インスタンスの設定」を参照してください。

ldif2dib を調整する

このセクションでは、ldif2dib を調整するために使用できるパラメータについて説明します。

- 220 ページの 「キャッシュを調整する」
- 220 ページの 「トランザクションサイズ」
- 221 ページの 「索引」
- 221 ページの 「ブロックキャッシュの比率」
- 221 ページの 「チェックポイント間隔」

キャッシュを調整する

データベースキャッシュの設定は、OpenText eDirectory のパフォーマンスに影響する重要な設定の 1 つです。設定値が小さすぎると、より頻繁にディスクから情報を取得する必要があるため、eDirectory の処理が低速になります。設定値が大きすぎると、他のプロセスの実行に十分なメモリを使用できないため、システム全体が低速になります。キャッシュの詳細については、『OpenText eDirectory チューニングガイド』の「FLAIM キャッシュ設定の変更」を参照してください。

キャッシュサイズを大きくすると、一般的に、バルクロードのパフォーマンスは向上します。ただし、キャッシュサイズを、LDIF ファイルサイズの 3.8 倍を超える値に設定しても、パフォーマンスはそれ以上向上しません。

トランザクションサイズ

トランザクションサイズとは、チャンクサイズをトランザクションごとのオブジェクト数で定義したものです。トランザクションサイズが大きい場合は少数の大規模チャンクが結果を書き込み、サイズが小さい場合は多数の小規模チャンクが結果を書き込みます。

トランザクションサイズが大きくなると、バルクロードのパフォーマンスが向上します。トランザクションサイズがゼロの場合は特別なケースで、トランザクションごとのオブジェクト数が無制限になります。トランザクションサイズがゼロの場合は、バルクロードの終了時にコミットが実行されるため、パフォーマンスが高くなります。ただし、LDIF ファイルが非常に大規模である場合 (オブジェクト数が 100 万個以上の場合) は、トランザクションサイズを 0 に設定することはお勧めしません。LDIF ファイルが非常に大規模である場合は、トランザクションサイズを最大で 4000 程に設定できます。

索引

インデックスを使用すると、検索のパフォーマンスは向上しますが、オブジェクトを DIB にロードするたびにインデックスを更新する必要があるため、バルクロードが低速になります。これは、下位文字列インデックスの場合は特に顕著です。そのため、多数のオブジェクトのバルクロードを実行するときには、インデックスを一時停止すると、バルクロードが高速化します。OpenText eDirectory サーバが起動すると、インデックスが自動的に再開されます。-x オプションを使用すると、インデックスを無効にしてから ldif2dib を使用してエントリをロードします。

ブロックキャッシュの比率

属性の下位文字列インデックスが有効になっている場合は、ブロックキャッシュの比率を 50% に設定することをお勧めします。属性の下位文字列インデックスが無効になっている場合は、ブロックキャッシュの比率を 90% に設定できます。

チェックポイント間隔

チェックポイント間隔は、チェックポイントバックグラウンドスレッドが開始されるまでデータベースが待機する時間です。このスレッドが開始されると、オンディスクバージョンのデータベースとインメモリ (キャッシュに保存された) データベースの整合性が保たれます。このチェックポイントスレッドは、ダーティキャッシュをディスクにフラッシュし、その後ロールフォワードログをクリーンアップします。チェックポイントスレッドの実行中にはバルクロードは一時的に停止するため、チェックポイント間隔の値を大きくして、バルクロードを高速にすることをお勧めします。

制限

このセクションでは、ldif2dib ユーティリティの制限について説明します。

- 221 ページの 「スキーマ」
- 222 ページの 「ACL テンプレート」
- 222 ページの 「オプション」
- 222 ページの 「簡易パスワード LDIF」
- 222 ページの 「カスタムクラス」
- 223 ページの 「フィルタ済みレプリカ」

スキーマ

- LDIF ファイルには、エントリが属するすべてのオブジェクトクラスを記述する必要があります。エントリは、継承のために複数のオブジェクトクラスに属すことができます。たとえば、inetOrgPerson タイプのエントリの構文は、LDIF ファイルでは次のようになります。

```
objectclass: inetorgperson
objectclass: organizationalPerson
objectclass: person
objectclass: top
```

- 現在、以下の構文はサポートされていません。

ACL テンプレート

オブジェクトクラスの ACL テンプレートで指定されている ACL は、ldif2dib を使用してバルクロードされたオブジェクトには自動的には追加されません。

オプション

Linux 上で -b オプションを使用すると、バルクロードの完了後に、統計を表示する画面が表示されなくなります。ただし参照用に、最終統計がログファイルに書き込まれます。

簡易パスワード LDIF

Windows では、簡易パスワードを持つ LDIF をアップロードするときに、System および Administrator フォルダに格納されている NICI キーが同期されていない場合は、ldif2dib が失敗することがあります。この問題に対処するには、次のように、nici/system フォルダに存在するキーにアクセスします

1. C:\Windows\system32\novell\nici\ フォルダに移動します。
2. **Administrator** フォルダに存在するファイルをバックアップします。
3. 次の手順に従って、system フォルダとそのファイルにアクセスします。
   - **3a** System フォルダの［プロパティ］ウィンドウにある［セキュリティ］タブに移動します。
   - **3b** ［**詳細オプション**］を選択し、［**所有者**］タブに移動します。
   - **3c** **Administrator** を選択します。
   - **3d** ［**セキュリティ**］タブに戻り、一覧に **Administrator** を追加します。

     同じ手順を繰り返し、System フォルダに存在するすべてのファイルに対して読み取りアクセスを許可します。
4. **Administrator** フォルダのファイルを system フォルダのファイルで上書きします。
5. アップロードした後に、バックアップしたファイルを **Administrator** フォルダにコピーします。
6. System フォルダおよびフォルダ内のファイルに対する Administrator のアクセス権を元に戻します。

カスタムクラス

ldif2dib を使用して、コンテナオブジェクトが多数含まれている LDIF のバルクロードを実行すると、メモリ内で構築が行われ、結果的に -150 エラーが報告されます。

フィルタ済みレプリカ

OpenText eDirectory は、フィルタ適用後のレプリカのバルクロードをサポートしません。

注意事項

以下のシナリオでは、ldif2dib の動作は未定義です。

- 223 ページの 「エントリが重複している」
- 223 ページの 「スキーマチェックが実行されない」
- 223 ページの 「ハードドライブ上のスペースが不足している」
- 223 ページの 「強制終了」
- 223 ページの 「端末のサイズを変更する」

エントリが重複している

重複するエントリまたは DIB にすでに存在するエントリを含む LDIF を、-u オプションを指定せずにアップロードすると、エントリが複数回追加され、DIB の一貫性が失われます。そのため、バルクロードを実行する前に、LDIF 内でエントリが繰り返されているかどうか、または DIB にすでに存在しているかどうかを把握できていない場合は、バルクロード処理中に -u オプションを使用します。

スキーマチェックが実行されない

ldif2dib はスキーマチェックを実行しません。結果として、属性がオブジェクトのスキーマに属していなくても、オブジェクトに属性を追加できます。これにより、DIB の一貫性が失われます。ldif2dib は、スキーマチェックを実行する必要のない LDIF データに対してのみ使用してください。

ハードドライブ上のスペースが不足している

ハードドライブ上に十分なスペースがないためにすべてのオブジェクトをロードできない場合の ldif2dib の動作は未定義です。バルクロード処理を開始する前に、すべてのオブジェクトをロードできるだけの十分なスペースがあることを確認する必要があります。

強制終了

ldif2dib プロセスを強制終了すると、DIB の一貫性が失われます。バルクロード処理を正常終了するには、<Esc> キーを使用します。

端末のサイズを変更する

バルクロード処理中に端末のサイズを変更すると、ユーザインタフェースに表示される統計が乱れる可能性があります。バルクロードの進行中は、端末のサイズ変更を避ける必要があります。

LDIF ファイル

OpenText インポート / エクスポート変換ユーティリティを使用すると、eDirectory との間での LDIF ファイルのインポートおよびエクスポートが簡単になります。詳細については、『*OpenText™ eDirectory CE 24.4 (v9.3) 管理ガイド*』の「「OpenText™ インポート / エクスポート変換ユーティリティ」」を参照してください。

LDIF インポートを正しく機能させるには、OpenText インポート / エクスポート変換ユーティリティが読み込み、および処理できる LDIF ファイルを最初に作成する必要があります。このセクションでは、LDIF ファイル形式および構文について説明し、正しい LDIF ファイルの例を示します。

- 224 ページの 「LDIF について」
- 233 ページの 「LDIF ファイルのデバッグ」
- 236 ページの 「LDIF を使用してスキーマを拡張する」
- 240 ページの 「ldif2dib の制限」

LDIF について

LDIF は、広く一般的に使用されているファイル形式で、ディレクトリ情報およびディレクトリで実行可能な変更操作について記述します。LDIF は、実際のディレクトリ内で使用されている記憶フォーマットとは完全に独立していて、通常は、LDAP サーバとの間でディレクトリ情報をエクスポートまたはインポートするために使用します。

一般的に、LDIF は簡単に生成できます。そのため、awk や perl などのツールを使用して、固有の形式のデータを LDAP ディレクトリに移動できます。また、LDIF 形式でテストデータを生成するスクリプトを作成することもできます。

LDIF ファイル形式

OpenText インポート / エクスポート変換ユーティリティを使用してインポートするファイルの形式は、LDIF 1 である必要があります。次に LDIF 1 形式のファイルの基本ルールを示します。

- コメント行以外の第 1 行目には、「version: 1」と記述します。
- バージョンの指定の後に、1 つ以上のレコードを定義します。
- 各レコードは、フィールドで構成されます。1 行に 1 フィールドずつ指定します。
- 各行は、改行またはキャリッジリターンと改行の組み合わせのどちらかで区切られます。
- レコードは、1 行以上の空白行で区切られます。

- LDIF レコードには、内容レコードと変更レコードの 2 つのタイプがあります。LDIF ファイルに記述するレコード数に制限はありませんが、記述されたすべてのレコードのタイプが一致している必要があります。同じ LDIF ファイル内に、内容レコードと変更レコードの両方を記述することはできません。
- シャープ記号 (#) で始まる行はコメント行です。この行は、LDIF ファイルの処理時には無視されます。

LDIF 内容レコード

LDIF 内容レコードは、エントリ全体の内容を表します。次に、4 つの内容レコードが定義された LDIF ファイルの例を示します。

```
 1  version: 1
 2  dn: c=US
 3  objectClass: top
 4  objectClass: country
 5
 6  dn: l=San Francisco, c=US
 7  objectClass: top
 8  objectClass: locality
 9  st: San Francisco
10
11 dn: ou=Artists, l=San Francisco, c=US
12 objectClass: top
13 objectClass: organizationalUnit
14 telephoneNumber: +1 415 555 0000
15
16 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
17 sn: Michaels
18 givenname: Peter
19 objectClass: top
20 objectClass: person
21 objectClass: organizationalPerson
22 objectClass: iNetOrgPerson
23 telephonenumber: +1 415 555 0001
24 mail: Peter.Michaels@aaa.com
25 userpassword: Peter123
26
```

この LDIF ファイルは、次の部分から構成されています。

コンポーネント	説明
バージョン指定子	LDIF ファイルの第 1 行目にはバージョンが記述されます。コロンとバージョン番号 (現在の定義は 1) の間には、1 つ以上のスペースを指定できますが、スペースを指定しなくても問題はありません。 バージョンを指定した行がない場合、LDIF ファイルを処理するアプリケーションはそのファイルのバージョンを 0 とみなすことができます。また、構文エラーとして LDIF ファイルが拒否される可能性もあります。LDIF を処理する OpenText のユーティリティは、バージョンを指定する行がない場合、ファイルのバージョンを 0 とみなします。
識別名指定子	各内容レコードの先頭の行 (この例では、2、6、11、および 16 行目) には、そのレコードが表すエントリの DN(識別名) を指定します。 DN 指定子は、次の 2 つのどちらかの形式をとる必要があります。 ◆ dn: *safe_UTF-8_distinguished_name* ◆ dn:: *Base64_encoded_distinguished_name*
行区切り記号	行区切り記号としては、改行、またはキャリッジリターンと改行の組み合わせのどちらかを使用できます。これにより、行区切りとして改行を使用する Linux および Solaris テキストファイルと、キャリッジリターンと改行の組み合わせを使用する MS-DOS* および Windows テキストファイルとの間の非互換性を解決できます。
レコード区切り記号	レコード区切りとしては、空白行 (この例では 5、10、15 および 26 行目) を使用します。 LDIF ファイル内の各レコード (最後のレコードも含む) の終わりには、レコード区切り記号として 1 行以上の空白行を挿入する必要があります。一部のアプリケーションでは、レコード区切りを指定していない LDIF ファイルもそのまま受け入れられますが、LDIF の仕様ではレコード区切りは必須です。
属性値指定子	内容レコード内のその他すべての行は、値指定子です。値指定子は、次の 3 つの形式のいずれかをとる必要があります。 ◆ 属性の記述 : *value* ◆ 属性の記述 :: *Base64_encoded_value* ◆ 属性の記述 : < *URL*

LDIF 変更レコード

LDIF 変更レコードには、ディレクトリに加えられる変更が記述されます。LDAP の更新操作 (追加、削除、変更、および DN の変更) はすべて、LDIF 変更レコードに記述できます。

LDIF 変更レコードでは、LDIF 内容レコードと同じ形式の識別名指定子、属性値指定子、およびレコード区切り記号を使用します。(詳細については、225 ページの 「LDIF 内容レコード」を参照してください。)LDIF 内容レコードとの違いは、LDIF 変更レコードには changetype フィールドがあることです。changetype フィールドは、変更レコードが指定する操作を識別します。

changetype フィールドは、次の 5 つの形式のいずれかである必要があります。

フォーム	説明
changetype: add	この変更レコードで LDAP の追加操作が指定されていることを示すキーワードです。
changetype: delete	この変更レコードで LDAP の削除操作が指定されていることを示すキーワードです。
changetype: moddn	この変更レコードで、LDIF プロセッサがバージョン 3 クライアントとして LDAP サーバにバインドされている場合は LDAP の DN 変更操作が、バージョン 2 クライアントとして LDAP サーバにバインドされている場合は RDN 変更操作が指定されていることを示すキーワードです。
changetype: modrdn	moddn 変更タイプと同義です。
changetype: modify	この変更レコードで LDAP の変更操作が指定されていることを示すキーワードです。

「追加」変更タイプ

追加変更レコードは、内容変更レコード (「225 ページの 「LDIF 内容レコード」」を参照) に、changetype: add フィールドを属性値フィールドの直前に追加したものと同じです。

すべてのレコードのタイプが一致している必要があります。内容レコードと変更レコードを同じファイルに記述することはできません。

```
 1 version: 1
 2 dn: c=US
 3 changetype: add
 4 objectClass: top
 5 objectClass: country
 6
 7 dn: l=San Francisco, c=US
 8 changetype: add
 9 objectClass: top
10 objectClass: locality
11 st: San Francisco
12
14 dn: ou=Artists, l=San Francisco, c=US
15   changetype: add
```

```
16 objectClass: top
17 objectClass: organizationalUnit
18 telephoneNumber: +1 415 555 0000
19
20 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
21 changetype: add
22 sn: Michaels
23 givenname: Peter
24 objectClass: top
25 objectClass: person
26 objectClass: organizationalPerson
27 objectClass: iNetOrgPerson
28 telephonenumber: +1 415 555 0001
29 mail: Peter.Michaels@aaa.com
30 userpassword: Peter123
31
```

「削除」変更タイプ

削除変更レコードはエントリの削除を指定するので、削除変更レコードに必要なフィールドは識別名指定子と「削除」変更タイプだけです。

次に、「227 ページの 「「追加」変更タイプ」」の LDIF ファイルで作成した 4 つのエントリを削除する LDIF ファイルの例を示します。

重要: 以前に追加したエントリを削除するには、エントリの指定順序を逆にする必要があります。順序を逆にしないと、コンテナ内のエントリが空でないため削除操作が失敗します。

```
 1 version: 1
 2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
 3 changetype: delete
 4
 5 dn: ou=Artists, l=San Francisco, c=US
 8   changetype: delete
 9
10 dn: l=San Francisco, c=US
11 changetype: delete
12
13 dn: c=US
14 changetype: delete
15
```

「変更」変更タイプ

「変更」変更タイプでは、すでに存在するエントリに対して属性値の追加、削除、および置換を指定できます。変更指定子は、次の 3 つの形式のいずれかをとる必要があります。

要素	説明
add: 属性タイプ	この属性タイプに対する後続の属性値指定子がエントリに追加されるように指定する必要があることを示すキーワードです。

要素	説明
delete: 属性タイプ	この属性タイプの値が削除されることを示すキーワードです。delete フィールドの後に属性値指定子が続く場合は、その指定された値が削除されます。 delete フィールドの後に属性値指定子がない場合は、すべての値が削除されます。属性に値がない場合、この操作は失敗しますが、属性には削除する値がないので結果的にはこの操作が成功したときと同じです。
replace: 属性タイプ	属性タイプの値が置き換えられることを示すキーワードです。replace フィールドに続く属性値指定子が、その属性タイプの新しい値になります。 replace フィールドの後に属性値指定子がない場合は、現在の値のセットが空の値のセットに置き換えられます (結果的に、属性が削除されます)。delete 変更指定子とは異なり、属性に値が設定されていない場合でも replace は成功します。どちらの場合も実際に得られる結果は同じです。

次の「変更」変更タイプの例では、cn=Peter Michaels エントリに別の電話番号を追加します。

```
 1 version: 1
 2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
 3 changetype: modify
 4 # add the telephone number to cn=Peter Michaels
 4 add: telephonenumber
 5 telephonenumber: +1 415 555 0002
 6
```

1 つの LDAP 変更要求にさまざまな変更を組み合わせて指定できるのと同じように、1 つの LDIF レコードに複数の変更指定子を指定できます。ハイフン (-) だけが記述されている行は、各変更指定子に対する属性値指定の終わりを示します。

次の LDIF ファイルの例では、複数の変更を組み合わせて指定しています。

```
 1 version: 1
 2
 3 # An empty line to demonstrate that one or more
 4 # line separators between the version identifier
 5 # and the first record is legal.
 6
 7 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
 8 changetype: modify
 9 # Add an additional telephone number value.
10 add: telephonenumber
11 telephonenumber: +1 415 555 0002
12 -
13 # Delete the entire fascimiletelephonenumber attribute.
14 delete: facsimileTelephoneNumber
15 -
16 # Replace the existing description (if any exists)
```

```
17 # with two new values.
18 replace: description
19 description: guitar player
20 description: solo performer
21 -
22 # Delete a specific value from the telephonenumber
23 # attribute.
24 delete: telephonenumber
25 telephonenumber: +1 415 555 0001
26 -
27 # Replace the existing title attribute with an empty
28 # set of values, thereby causing the title attribute to
29 # be removed.
30 replace: title
31 -
32
```

「DN 変更」変更タイプ

「DN 変更」変更タイプでは、エントリのリネーム、移動、またはその両方ができます。この変更タイプは、2 つの必須フィールドと 1 つのオプションフィールドで構成されます。

フィールド	説明
newrdn (必須)	このレコードの処理の実行中にエントリに割り当てられる新しい名前を指定します。新規 RDN(newrdn) 指定子は、次の 2 つのどちらかの形式をとる必要があります。 ◆ newrdn: *safe_UTF-8_relative_distinguished_name* ◆ newrdn:: *Base64_encoded_relative_ distinguished_name* 新規 RDN 指定子は、「DN 変更」変更タイプが指定されたすべての LDIF レコードで指定されている必要があります。
deleteoldrdn (必須)	旧 RDN 削除 (deleteoldrdn) 指定子は、古い RDN を newrdn(新規 RDN) に置き換えるか、残しておくかを指定するフラグです。これは、次の 2 つのどちらかの形式をとります。 ◆ deleteoldrdn: 0 リネーム後も古い RDN の値をエントリ内に残しておくことを指定します。 ◆ deleteoldrdn: 1 エントリのリネーム後に古い RDN の値を削除することを指定します。

フィールド	説明
newsuperior (オプション)	新規スーペリア (newsuperior) 指定子は、この DN 変更レコードの処理時にエントリに割り当てる新しいペアレントの名前を指定します。新規スーペリア指定子は、次の 2 つのどちらかの形式をとります。 ◆ newsuperior: *safe_UTF-8_distinguished_name* ◆ newsuperior:: *Base64_encoded_distinguished_ name* 新規スーペリア指定子は、「DN 変更」変更タイプが指定された LDIF レコードでオプションとして使用できます。これは、エントリのペアレントを変更する場合のみ指定します。

次の「DN 変更」変更タイプの例で、エントリの名前を変更する方法を示します。

```
 1 version: 1
 2
 3 # Rename ou=Artists to ou=West Coast Artists, and leave
 4 # its old RDN value.
 5 dn: ou=Artists,l=San Francisco,c=US
 6 changetype: moddn
 7 newrdn: ou=West Coast Artists
 8 deleteoldrdn: 1
 9
```

次の「DN 変更」変更タイプの例で、エントリを移動する方法を示します。

```
 1 version: 1
 2
 3 # Move cn=Peter Michaels from
 4 # ou=Artists,l=San Francisco,c=US to
 5 # ou=Promotion,l=New York,c=US and delete the old RDN.
 5 dn: cn=Peter Michaels,ou=Artists,l=San Francisco,c=US
 6 changetype: moddn
 7 newrdn: cn=Peter Michaels
 8 deleteoldrdn: 1
 9 newsuperior: ou=Promotion,l=New York,c=US
10
```

次の「DN 変更」変更タイプの例では、エントリを移動し、同時に名前を変更する方法を示します。

```
 1 version: 1
 2
 3 # Move ou=Promotion from l=New York,c=US to
 4 # l=San Francisco,c=US and rename it to
 5 # ou=National Promotion.
 5 dn: ou=Promotion,l=New York,c=US
 6 changetype: moddn
 7 newrdn: ou=National Promotion
 8 deleteoldrdn: 1
 9 newsuperior: l=San Francisco,c=US
10
```

重要 : LDAP 2 の RDN 変更操作では、エントリの移動はサポートされません。LDAP 2 クライアントで LDIF newsuperior 構文を使用してエントリを移動しようとすると、その要求は失敗します。

LDIF ファイル内での行の折り返し

LDIF ファイル内で行を折り返すには、行を折り返したい場所で単に行区切り記号 (改行、またはキャリッジリターンと改行の組み合わせ) を挿入し、その後にスペースを追加します。行の先頭にスペースがある場合、LDIF パーサではスペースの後のデータとその前の行のデータを結合して解析します。したがって、先頭のスペースは無視されます。

マルチバイトの UTF-8 文字の途中では、行を折り返さないでください。

次に、行の折り返しを含む (13 および 14 行目)LDIF ファイルの例を示します。

```
 1 version: 1
 2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US
 3 sn: Michaels
 4 givenname: Peter
 5 objectClass: top
 6 objectClass: person
 7 objectClass: organizationalPerson
 8 objectClass: iNetOrgPerson
 9 telephonenumber: +1 415 555 0001
10 mail: Peter.Michaels@aaa.com
11 userpassword: Peter123
12 description: Peter is one of the most popular music
13  ians recording on our label. He's a big concert dr
14  aw, and his fans adore him.
15
```

LDIF ファイル内でのハッシュ化パスワードの表記

LDIF ファイル内では、ハッシュ化パスワードは Base64 データとして表記されます。属性名 userpassword に続けて、パスワードをハッシュ化するために使用される暗号化方式の名前を記述する必要があります。この名前は、次に示すように中カッコ「{}」で囲んで記述します。

例 1

SHA ハッシュ化パスワードの場合 :

```
1 version: 1 2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US 3
sn: Michaels 4 userpassword: {SHA}xcbdh46ngh37jsd0naSFDedjAS30dm5
objectclass: inetOrgPerson
```

例 2

SHA ハッシュ化パスワードの場合 :

```
1 version: 1 2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US 3
sn: Michaels 4 userpassword: {SHA}sGs948DFGkakdfkasdDF34DF4dS3skl5DFS5
objectclass: inetOrgPerson
```

例 3

Digest MD5 ハッシュ化パスワードの場合 :

```
1 version: 1 2 dn: cn=Peter Michaels, ou=Artists, l=San Francisco, c=US 3
sn: Michaels 4 userpassword: {MD5}a45lkSDF234SDFG62dsfsf2DG2QEvgdmnk4305
objectclass: inetOrgPerson
```

LDIF ファイルのデバッグ

- 233 ページの 「前方参照を有効化する」
- 234 ページの 「LDIF ファイルの構文をチェックする」
- 234 ページの 「LDIF エラーファイルを使用する」
- 235 ページの 「LDAP SDK デバッギングフラグを使用する」

LDIF ファイルで問題が発生した場合、次のことを考慮してください。

前方参照を有効化する

LDIF ファイルで、あるエントリを追加するレコードを、そのエントリのペアレントを追加するレコードの前に記述してしまう場合があります。この場合、LDAP サーバが新しいエントリを追加しようとすると、そのエントリのペアレントが存在しないためエラーが発生します。

この問題は、前方参照の使用を有効にするだけで解決できます。前方参照の作成を有効にすると、エントリを作成するときにそのペアレントがまだ存在していない場合でも、このペアレント用に前方参照というプレースホルダが作成されるため、エントリを正常に作成できます。以後の操作で親が作成されると、前方向参照は通常のエントリに変更されます。

LDIF のインポートが完了した後でも、1 つ以上の前方参照が残っている場合があります (たとえば、LDIF ファイルでエントリのペアレントが作成されなかった場合など)。この場合、前方向参照は OpenText Identity Console に不明オブジェクトとして表示されます。前方参照エントリを検索することはできますが、前方参照エントリには属性も属性値もないため、objectClass 以外の属性を読み込むことはできません。ただし、前方参照の下に位置する実オブジェクトエントリ上では、すべての LDAP 操作が正常に機能します。

前方参照エントリを識別する

前方参照エントリは「不明」のオブジェクトクラスを持ち、また、内部 NDS EF_REFERENCE エントリにフラグが設定されています。LDAP を使用して不明オブジェクトクラスのオブジェクトを検索することもできますが、現時点では LDAP からエントリフラグの設定にアクセスしてそれが前方参照エントリであることを確認する方法はありません。

前方参照エントリを通常オブジェクトへ変更する

(LDIF ファイルまたは LDAP クライアント要求などを使用して) オブジェクトを作成するだけで、前方参照エントリを通常のオブジェクトに変更できます。OpenText eDirectory で作成するように指定したエントリが前方向参照としてすでに存在する場合、OpenText eDirectory では既存の前方向参照エントリが、作成を指定したオブジェクトに変換されます。

OpenText インポート / エクスポート変換ユーティリティのコマンドラインインタフェースの使用

コマンドラインインタフェースで前方参照を有効にするには、-F LDAP ターゲットハンドラオプションを使用します。

詳細については、『*OpenText™ eDirectory CE 24.4 (v9.3) 管理ガイド*』の「「LDIF ターゲットハンドラのオプション」」を参照してください。

LDIF ファイルの構文をチェックする

ファイル内のレコードを処理する前に、[操作を表示するが実行しない] LDIF ソースハンドラオプションを使用して LDIF ファイルの構文をチェックできます。

LDIF ソースハンドラは、LDIF ファイル内のレコードを処理するときに常に構文をチェックします。このオプションを使用すると、レコードの処理を無効にして、構文を検証できます。

OpenText インポート / エクスポート変換ユーティリティのコマンドラインインタフェースの使用

コマンドラインインタフェースで LDIF ファイルの構文をチェックするには、-n LDIF ソースハンドラオプションを使用します。

詳細については、『*OpenText™ eDirectory CE 24.4 (v9.3) 管理ガイド*』の「「LDIF ソースハンドラのオプション」」を参照してください。

LDIF エラーファイルを使用する

OpenText インポート / エクスポート変換ユーティリティは、ターゲットハンドラによる処理に失敗したレコードをすべてリストした LDIF ファイルを自動的に作成します。ユーティリティによって生成された LDIF エラーファイルを編集してエラーを修正し、サーバに再適用することで、失敗したレコードに含まれているインポートまたはデータの移行を完了できます。

OpenText インポート / エクスポート変換ユーティリティのコマンドラインインタフェースの使用

コマンドラインユーティリティでエラーログオプションを設定するには、-l 一般オプションを使用します。

詳細については、『*OpenText™ eDirectory CE 24.4 (v9.3) 管理ガイド*』の「「一般オプション」」を参照してください。

LDAP SDK デバッギングフラグを使用する

一部の LDIF の問題を理解するには、LDAP クライアント SDK がどのように機能するかを理解する必要があります。LDAP ソースハンドラ、LDAP ターゲットハンドラ、またはその両方に、次のデバッギングフラグを設定できます。

値	説明
0x0001	LDAP ファンクションコールをトレースします。
0x0002	パケットに関する情報を出力します。
0x0004	引数に関する情報を出力します。
0x0008	接続情報を出力します。
0x0010	BER のエンコーディングおよびデコーディング情報を出力します。
0x0020	検索フィルタ情報を出力します。
0x0040	設定情報を出力します。
0x0080	ACL 情報を出力します。
0x0100	統計情報を出力します。
0x0200	追加の統計情報を出力します。
0x0400	シェル情報を出力します。
0x0800	解析情報を出力します。
0xFFFF (10 進数では、-1)	すべてのデバッギングオプションを有効にします。

この機能を有効にするには、LDAP ソースハンドラおよびターゲットハンドラで-e オプションを使用します。-e オプションに指定する整数の値は、LDAP SDK でさまざまな種類のデバッグ情報を有効にするビットマスクです。

詳細については、『*OpenText™ eDirectory CE 24.4 (v9.3) 管理ガイド*』の「「LDAP ソースハンドラのオプション」」および「「LDAP ターゲットハンドラのオプション」」を参照してください。

LDIF を使用してスキーマを拡張する

LDIF では LDAP 更新操作を表すことができるので、LDIF を使用してスキーマを変更できます。

新しいオブジェクトクラスを追加する

クラスを追加するには、単に、NDSObjectClassDescription の仕様に従った属性値を subschemaSubentry の objectClasses 属性に追加します。

```
NDSObjectClassDescription = "(" whsp
   numericoid whsp
   [ "NAME" qdescrs ]
   [ "DESC" qdstring ]
   [ "OBSOLETE" whsp ]
   [ "SUP" oids ]
   [ ( "ABSTRACT" / "STRUCTURAL" / "AUXILIARY" ) whsp ]
   [ "MUST" oids ]
   [ "MAY" oids ]
   [ "X-NDS_NOT_CONTAINER" qdstrings ]
   [ "X-NDS_NONREMOVABLE" qdstrings ]
   [ "X-NDS_CONTAINMENT" qdstrings ]
   [ "X-NDS_NAMING" qdstrings ]
   [ "X-NDS_NAME" qdstrings ]
   whsp ")"
```

次の LDIF ファイルの例では、person objectClass をスキーマに追加します。

```
 1 version: 1
 2 dn: cn=schema
 3 changetype: add
 4 objectClasses: ( 2.5.6.6 NAME 'person' DESC 'Standard
 5   ObjectClass' SUP ndsLoginProperties STRUCTURAL MUST
 6   (cn $ sn) MAY (description $ seeAlso $ telephoneNum
 7  ber $ fullName $ givenName $ initials $ uid $ userPa
 8  ssword) X-NDS_NAMING ('cn' 'uid') X-NDS_CONTAINMENT
 9  ('organization' 'organizationalUnit' 'domain') X-NDS
10  _NAME 'Person' X-NDS_NOT_CONTAINER '1' X-NDS_NONREMO
11  VABLE '1')
12
```

必須属性

必須属性は、オブジェクトクラス記述の MUST セクションにリストします。person オブジェクトクラスの場合、必須属性は cn と sn です。

オプション属性

オプション属性のリストは、オブジェクトクラス記述の MAY セクションに記述します。person オブジェクトクラスのオプション属性は、description、seeAlso、telephoneNumber、fullName、givenName、initials、uid、および userPassword です。

注 : userPassword 属性は、オプション (MAY) 属性には使用できません。この LDIF 形式を使用して、新しい objectClass でこの属性を必須 (MUST) 属性に使用してスキーマを拡張しようとしても、操作は失敗します。

包含ルール

定義されているオブジェクトクラスを包含するオブジェクトクラスは、オブジェクトクラス記述の X-NDS_CONTAINMENT セクションで指定します。person オブジェクトクラスを包含するオブジェクトクラスは、organization、organizationalUnit、および domain です。

新しい属性を追加する

属性を追加するには、NDSAttributeTypeDescription の仕様に従って属性値を subschemaSubentry の attributes 属性に追加します。

```
NDSAttributeTypeDescription = "(" whsp
  numericoid whsp  ; AttributeType identifier
  [ "NAME" qdescrs ]  ; name used in AttributeType
  [ "DESC" qdstring ]  ; description
  [ "OBSOLETE" whsp ]
  [ "SUP" woid ]  ; derived from this other AttributeType
  [ "EQUALITY" woid]  ; Matching Rule name
  [ "ORDERING" woid]  ; Matching Rule name
  [ "SUBSTR" woid ]  ; Matching Rule name
  [ "SYNTAX" whsp noidlen whsp ] ;  Syntax OID
  [ "SINGLE-VALUE" whsp ]  ; default multi-valued
  [ "COLLECTIVE" whsp ]  ; default not collective
  [ "NO-USER-MODIFICATION" whsp ] ;  default user modifiable
  [ "USAGE" whsp AttributeUsage ] ;  default userApplications
  [ "X-NDS_PUBLIC_READ" qdstrings ]
                              ; default not public read ('0')
  [ "X-NDS_SERVER_READ" qdstrings ]
                               ; default not server read ('0')
  [ "X-NDS_NEVER_SYNC" qdstrings ]
                                ; default not never sync ('0')
  [ "X-NDS_NOT_SCHED_SYNC_IMMEDIATE" qdstrings ]
                             ;  default sched sync immediate ('0')
  [ "X-NDS_SCHED_SYNC_NEVER" qdstrings ]
                                  ;  default schedule sync ('0')
  [ "X-NDS_LOWER_BOUND" qdstrings ]
                               ;  default no lower bound('0')
                                     ;(upper is specified in SYNTAX)
  [ "X-NDS_NAME_VALUE_ACCESS" qdstrings ]
                          ; default not name value access ('0')
  [ "X-NDS_NAME" qdstrings ]  ; legacy NDS name
  whsp ")"
```

次の LDIF ファイルの例では、title 属性タイプをスキーマに追加します。

```
1 version: 1
2 dn: cn=schema
3 changetype: add
4 attributeTypes: ( 2.5.4.12 NAME 'title' DESC 'Standa
5  rd Attribute' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{
6  64} X-NDS_NAME 'Title' X-NDS_NOT_SCHED_SYNC_IMMEDIA
7  TE '1' X-NDS_LOWER_BOUND '1')
8
```

単一値と複数値

属性は、明示的に単一値として定義されない限り、デフォルトでは複数値です。次の LDIF ファイルの例では、SYNTAX セクションの後に SINGLE-VALUE キーワードを追加することによって、title を単一値として定義しています。

```
1 version: 1
2 dn: cn=schema
3 changetype: add
4 attributeTypes: ( 2.5.4.12 NAME 'title' DESC 'Standa
5  rd Attribute' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{
6  64} SINGLE-VALUE X-NDS_NAME 'Title' X-NDS_NOT_SCHED
7  _SYNC_IMMEDIATE '1' X-NDS_LOWER_BOUND '1')
8
```

既存のオブジェクトクラスへオプション属性を追加する

新しいスキーマエレメントを追加する場合は問題ありませんが、通常、既存のスキーマエレメントを変更または拡張する場合には注意が必要です。すべてのスキーマエレメントは OID によって固有に識別されるため、標準スキーマエレメントを拡張すると、元の OID を使用する場合でも実際にはそのエレメントに対して 2 つめの定義が作成されます。このため、不整合が発生することがあります。

スキーマエレメントの変更が必要な場合もあります。たとえば、開発しながらスキーマエレメントを洗練していくときに、新しいスキーマエレメントの拡張または修正が必要な場合があります。次のような場合は、クラスに直接新しい属性を追加せずに、通常は補助クラスのみを使用します。

- 既存のオブジェクトクラスに新しい属性を追加する場合。
- 既存のオブジェクトクラスのサブクラスを作成する場合。

補助クラスを追加または削除する

次のサンプル LDIF ファイルは、2 つの新しい属性、およびこの新しい属性に付随する補助クラスを作成してから、inetOrgPerson エントリをエントリのオブジェクトクラスとして auxiliary クラスと auxiliary クラスの属性値に追加します。

```
version: 1
# Add an attribute to track a bear's hair. The attribute is
# multi-valued, uses a case ignore string syntax,
# and has public read rights
# Values may include: long hair, short, curly, straight,
# none, black, and brown
# X-NDS_PUBLIC_READ '1' The 1 allows public read,
# 0 denies public read
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: ( 2.16.840.1.113719.1.186.4.10 NAME
'bearHair' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
X-NDS_PUBLIC_READ '1' )

# add an attribute to store a bear's picture
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: ( 2.16.840.1.113719.1.186.4.11 NAME
'bearPicture' SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
SINGLE-VALUE )

# create an Auxiliary class for the bearfeatures
dn: cn=schema
changetype: modify
add: objectclasses
objectclasses: (2.16.840.1.113719.1.186.6.101 NAME
'bearFeatures' MAY (bearHair $ bearPicture) AUXILIARY)

# now create a user named bobby
dn: cn=bobby,o=bearcave
changetype: add
cn: bobby
sn: bear
givenName: bobby
bearHair: Short
bearHair: Brown
bearHair: Curly
bearPicture:< file:///c:/tmp/alien.jpg
objectClass: top
objectClass: person
objectClass: inetOrgPerson
objectClass: bearFeatures

# now create a person named john that will later be changed
# into a bear when bearFeatures is added to its objectClass
# list
dn: cn=john,o=bearcave
changetype: add
cn: John
sn: bear
givenName: john
objectClass: top
objectClass: person
```

```
objectClass: inetOrgPerson

# now morph john into a bear by adding bearFeatures
dn: cn=john,o=bearcave
changetype: modify
add: objectClass
objectClass: bearFeatures
-
add: bearHair
bearHair: long
bearHair: black
#bearPicture:< file:///c:/tmp/john.jpg>
-

# to morph john back to a person, simply delete the
# objectClass bearFeatures
dn: cn=john,o=bearcave
changetype: modify
delete: objectClass
objectClass: bearFeatures
```

補助クラスの削除にあたって、objectClass リストから auxiliary クラスを削除する場合には、auxiliary クラスに関連するすべての値を削除する必要はありません。この処理は OpenText eDirectory によって自動的に行われます。

auxiliary クラスに MUST 属性がある場合、auxiliary クラスを objectClass リストへ追加する変更操作でもこれらの属性を指定する必要があります。これらの属性が指定されていない場合、変更は失敗します。

XML 解析で発生する既知の問題

個々のレコードが XML ファイルで指定されたすべての XML ルールを遵守していない場合、LDIF レコード (LDAP サーバで生成された LDIF 形式またはレコード) の XML 処理は成功しません。

ldif2dib の制限

- 241 ページの 「簡易パスワード LDIF」
- 241 ページの 「スキーマ」
- 241 ページの 「ACL テンプレート」
- 242 ページの 「シグナルハンドラ」

簡易パスワード LDIF

Windows では、簡易パスワードを持つ LDIF をアップロードするときに、system フォルダおよび *Administrator* フォルダに格納されている NICI キーが同期されていない場合、ldif2dib が失敗することがあります。

この問題を回避するには、次の手順で nici/system フォルダ内のキーにアクセスします。

1. C:\Windows\system32\novell\nici\ フォルダに移動します (32 ビット NICI の場合)。

   または

   C:\Windows\SysWOW64\novell\nici\ フォルダに移動します (64 ビット NICI の場合)。
2. *Administrator* フォルダのファイルをバックアップします。
3. System フォルダの［プロパティ］ウィンドウにある［セキュリティ］タブに移動します。
4. ［詳細設定］を選択し、［所有者］タブに移動します。
5. Administrator を選択します。
6. ［セキュリティ］タブに戻り、Administrator を一覧に追加します。
7. 手順ステップ 3 ～ステップ 6 を繰り返し、system フォルダ内にあるすべてのファイルに対して読み取りアクセス権を取得します。
8. *Administrator* フォルダのファイルを system フォルダのファイルで上書きします。
9. アップロードした後に、バックアップファイルを *Administrator* フォルダにコピーします。
10. system フォルダおよびフォルダ内のファイルへの Administrator のアクセス権を変更します。

スキーマ

LDIF ファイルには、エントリが属するすべてのオブジェクトクラスを記述する必要があります。また、クラスの継承によってエントリが属することになるクラスも記述する必要があります。たとえば、inetOrgPerson タイプのエントリの構文は LDIF ファイルでは次のようになります。

- objectclass: inetorgperson
- objectclass: organizationalPerson
- objectclass: person
- objectclass: top

ACL テンプレート

ldif2dib ユーティリティを使用してバルクロード処理を行ったオブジェクトは、指定された ACL と一緒にオブジェクトのオブジェクトクラス用の ACL テンプレートには追加されません。

シグナルハンドラ

s キーまたは S キーを押すと、オフラインのバルクロード処理を一時的に停止することができます。バルクロード処理を停止する際はエスケープキー (Esc) を使用することができます。

8 OpenText の監視

OpenText には、OpenText ツリー内にあるすべてのサーバに対して、クロスプラットフォーム対応の監視と診断を行う機能があります。この機能を使用すると、各種ツールやプロトコルハンドラから複数のインタフェースを使用して、または Web ブラウザが使えるネットワーク上で、どの場所からでもサーバを監視することができます。ndscheck、iMonitor、LDAP rootdse 検索、ndsrepair などの OpenText ユーティリティが、監視データの収集に役立ちます。

また、OpenText には、OpenText サブシステムとバックグラウンドプロセスのリアルタイム統計情報を取得するための LDAP 検索方式があります。この方式では、OpenText プロセスと操作の状態が、cn=monitor というベース DN を持つエントリとして OpenText により記録されます。OpenText 管理者は、このインタフェースを使用して OpenText のモジュールと操作のステータスを監視できます。OpenText は、LDAP プロトコル上でこの機能をサポートするため、データの監視要求を出すことができるのは、LDAP クライアントのみです。

- 243 ページの 「OpenText iMonitor の使用」
- 280 ページの 「cn=monitor を使用した監視」
- 292 ページの 「DSTrace の使用」
- 301 ページの 「DSTrace メッセージ」
- 305 ページの 「OpenText iMonitor のメッセージフィルタリング」
- 305 ページの 「SAL メッセージのフィルタ」

OpenText iMonitor の使用

OpenText iMonitor には、OpenText ツリー内にあるすべてのサーバに対して、クロスプラットフォームの監視と診断を行う機能があります。このユーティリティを使用すると、Web ブラウザを使用できるネットワーク上の場所ならどこからでもサーバを監視できます。

また、OpenText iMonitor を使用すると、OpenText 環境に対して、パーティション、レプリカ、またはサーバベースの詳細な管理が可能になります。また、実行しているタスクの種類、タスクの開始時間、結果、および実行時間を検証できます。

OpenText iMonitor は、OpenText が従来提供していたサーバベースの OpenText ツール (DSBrowse、DSTrace、DSDiag、および DSRepair の診断機能など) の多くに取って代わるものであり、Web ベースで使用できます。このため、OpenText iMonitor の機能は主にサーバで動作することに重点を置いています。つまり OpenText ツリー全体ではなく、個々の OpenText エージェント (ディレクトリサービスで実行しているインスタンス) のヘルスが、iMonitor の機能に対して重要な要素となります。

OpenText iMonitor には次のような機能があります。

- OpenText eDirectory ヘルスサマリ
  - 同期情報
  - 認識されているサーバ
  - エージェントの環境設定
- OpenText eDirectory ヘルスチェック
- ハイパーリンク付きの DS トレース
- エージェントの環境設定
- エージェントアクティビティおよび Verb 統計
- Reports (レポート)
- エージェント情報
- エラー情報
- オブジェクト / スキーマブラウザ
- OpenText Identity Manager 監視
- 検索
- パーティションリスト
- エージェントプロセスのステータス
- バックグラウンドプロセスのスケジュール
- DSRepair
- 接続監視

OpenText iMonitor の情報は、次の要素に基づいて表示されます。

- 確立された識別情報

  OpenText iMonitor で実行するすべての要求には、識別情報に基づく eDirectory 権が適用されます。たとえば、［DSRepair］ページにアクセスするには、アクセスを行うサーバに対して、サーバの管理者またはコンソールオペレータとしてログインする必要があります。
- 監視している OpenText eDirectory エージェントのバージョン

  新しいバージョンの NDS および OpenText eDirectory には、以前のバージョンにはない機能とオプションがあります。

OpenText iMonitor に表示された情報から、ローカルサーバの状態が一目でわかります。

この章では次のトピックについての情報を説明します。

- 245 ページの 「システム要件」
- 245 ページの 「OpenText iMonitor へのアクセス」
- 246 ページの 「OpenText iMonitor のアーキテクチャ」
- 252 ページの 「OpenText iMonitor の機能」

- 277 ページの 「OpenText iMonitor の操作のセキュリティ保護」
- 278 ページの 「HTTP サーバオブジェクトの設定」
- 279 ページの 「ndsconfig を使用する HTTP スタックパラメータの設定」

システム要件

OpenText iMonitor を使用するには次のものが必要です。

- OpenText eDirectory 8.7.1 以降
- サポートされる Web ブラウザ (Microsoft Internet Explorer、Firefox など)

プラットフォーム

OpenText iMonitor ユーティリティは次のプラットフォームで動作します。

- Windows 2000、および 2003 Server (SSL なし)
- Linux

Windows では、OpenText eDirectory が実行されると、OpenText iMonitor は自動的にロードされます。Linux で OpenText iMonitor をロードするには、ndsimonitor -l コマンドを使用します。また、/etc/opt/novell/eDirectory/conf/ndsimon.conf ファイルに［ndsimonitor］を追加して、OpenText eDirectory サーバを開始する前に OpenText iMonitor を自動的にロードすることもできます。

OpenText iMonitor ユーティリティは次の Web ブラウザで動作します。

- Microsoft IE 10 以上
- Firefox* 40 以上

監視できる OpenText eDirectory のバージョン

OpenText iMonitor を使用して監視できる NDS および OpenText eDirectory のバージョンは次のとおりです。

- Windows 用のすべてのバージョンの NDS および OpenText eDirectory
- Linux 用のすべてのバージョンの NDS および OpenText eDirectory

OpenText iMonitor へのアクセス

1. OpenText iMonitor の実行ファイルが OpenText eDirectory サーバで実行されていることを確認します。
2. Web ブラウザを開きます。
3. アドレス (URL) のフィールドに、次の形式で入力します。

   `http://server's_TCPIP_address:httpstack_port/nds`

   たとえば、次のように入力します。

```
http://137.65.135.150:8028/nds
```

DNS 名は、iMonitor 内でサーバの IP、IPX アドレス、または識別名を使用できる箇所であればどこでも使用できます。たとえば、次のような DNS が設定されているとします。

```
http://prv-gromit.provo.novell.com/nds?server=prv-
igloo.provo.novell.com
```

これは、次の設定と同等です。

```
http://prv-gromit.provo.novell.com/nds?server=IP_or_IPX address
```

または

```
http://prv-gromit.provo.novell.com/nds?server=/cn=prv-
igloo,ou=ds,ou=dev,o=novell,t=novell_inc
```

OpenText eDirectory HTTPS スタックが有効であれば、HTTPS を通して OpenText iMonitor を使用できます。

4 ユーザ名、コンテキスト、パスワードを指定します。例 : login cn=admin.o=novell

すべての機能にアクセスするには、完全識別名を指定して管理者としてログインするか、管理者と同等のアクセス権でログインします。

5 [ログイン] をクリックします。

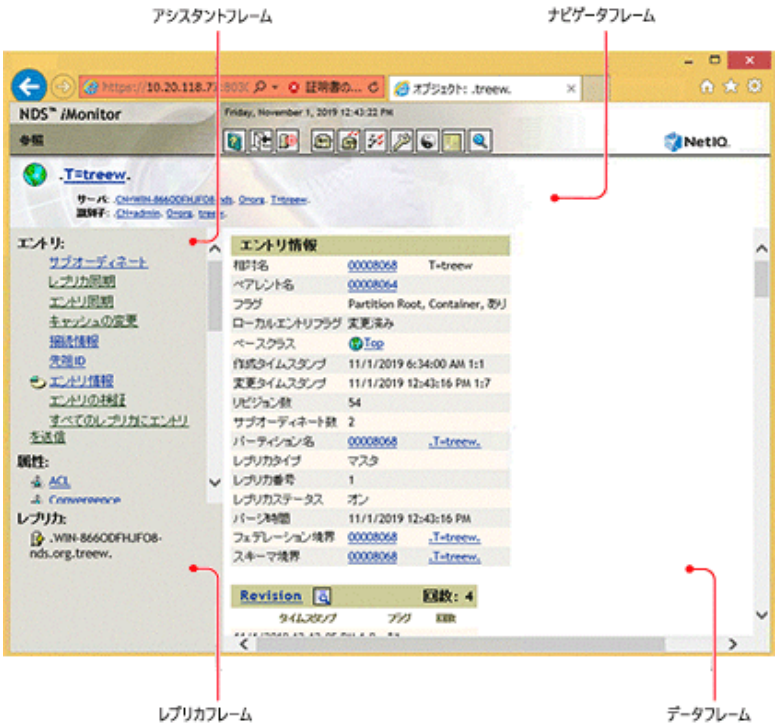
OpenText iMonitor のアーキテクチャ

- 246 ページの 「OpenText iMonitor ページの構成」
- 247 ページの 「動作モード」
- 249 ページの 「すべてのページからアクセス可能な iMonitor の機能」
- 249 ページの 「環境設定ファイル」

OpenText iMonitor ページの構成

OpenText iMonitor の各ページは、ナビゲータフレーム、アシスタントフレーム、データフレーム、およびレプリカフレームの 4 つのフレームまたはセクションに分かれています。

図 8-1 OpenText iMonitor のフレーム



ナビゲータフレーム : ページの上部にあります。このフレームには、データの読み込み元のサーバ名、ユーザの識別情報、および他の画面 (オンラインヘルプ、ログイン、サーバポータルなどの OpenText iMonitor ページ) にリンクするためのアイコンが表示されます。

アシスタントフレーム : ページの左側にあります。このフレームには、ナビゲーション用の項目 (他のページへのリンクなど)、データフレームでのデータの検索に使用する項目、および表示されているページでのデータの取得や解釈に使用する項目が含まれます。

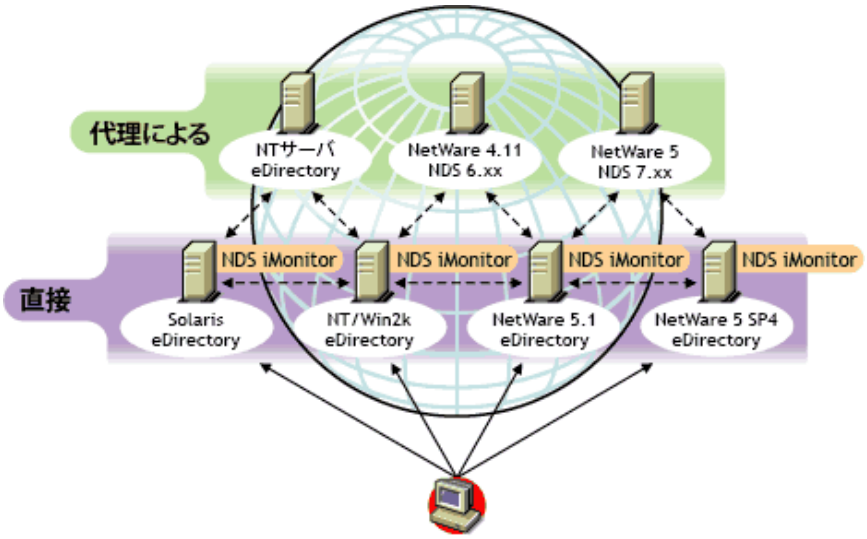
データフレーム : 上部にあるリンクをクリックすると、ローカルサーバに関する詳細情報が表示されます。Web ブラウザがフレームをサポートしていない場合には、このページだけが表示されます。

レプリカフレーム : 現在表示されているレプリカを判別できます。またリンクを使用して、現在表示されている情報が、他のサーバやレプリカを基準としたときに、どのような状態になっているかを確認できます。表示したページに、要求したデータの他のレプリカが存在する場合、またはデータフレームに表示されている情報を別の状態で表示するレプリカが存在する可能性がある場合にのみ、レプリカフレームが表示されます。

動作モード

OpenText iMonitor には、ダイレクトモードとプロキシモードという 2 種類の動作モードがあります。モードを切り替えるために環境設定情報を変更する必要はありません。モードは OpenText iMonitor が自動的に切り替えますが、eDirectory ツリーのナビゲートを正しく効率的に行うために、これらのモードについて理解しておくことをお勧めします。

図8-2 動作モード



ダイレクトモード : このモードが使用されるのは、Web ブラウザが iMonitor の実行ファイルを実行しているマシン上のアドレスまたは DNS 名を直接ポイントしていて、そのマシンのローカル eDirectory DIB 上の情報だけを読み込んでいる場合です。

OpenText iMonitor の一部の機能はサーバ限定で、そのマシン上で動作している OpenText iMonitor でのみ使用できます。サーバ限定の機能では、リモートからアクセスできないローカル API のセットを使用します。サーバ限定の OpenText iMonitor の機能には、DSTrace、DSRepair、および［バックグラウンドプロセスのスケジュール (Background Process Schedule)］ページなどがあります。ダイレクトモードの場合、ローカルコンピュータからは OpenText iMonitor のすべての機能を利用できます。

ダイレクトモードの主な特徴は次のとおりです。

- サーバ限定の機能をすべて使用できます。
- ネットワークの通信量が減少します (高速アクセスが可能)。
- OpenText eDirectory のバージョンに関係なく、プロキシによるアクセスも可能です。

プロキシモード : このモードが使用されるのは、Web ブラウザが、あるマシンで実行されている OpenText iMonitor をポイントしていて、同時に別のマシンから情報を集めている場合です。OpenText iMonitor では、サーバに限定されない機能に対してはサーバに限定されない従来の OpenText eDirectory プロトコルを使用します。このため、「NDS 6.*x*」で始まる以前のバージョンの OpenText eDirectory をすべて監視および診断できます。ただし、サーバ限定の機能では、リモートからアクセスできない API が使用されます。

プロキシモードが有効なときに、他のサーバの動作モードをダイレクトモードに切り替えることもできます。ただし、そのサーバの OpenText eDirectory のバージョンで OpenText iMonitor がサポートされていることが条件です。プロキシによる情報収集対象のサーバ上で OpenText iMonitor が実行されている場合は、ナビゲータフレームに追加のアイコンボタンが表示されます。カーソルをこのアイコン上に移動すると、そのリモートサーバ上で実行されているリモート OpenText iMonitor へのリンクが表示されます。ただし、プロキシによる情報収集対象のサーバ上で以前のバージョンの OpenText eDirectory が実行されている

場合は、このアイコンは表示されません。そのリモートサーバが OpenText iMonitor をサポートしているバージョンの OpenText eDirectory にアップグレードされるまでは、そのサーバからの情報収集には常にプロキシを使用する必要があります。

プロキシモードの主な特徴は次のとおりです。

- ツリー内のすべてのサーバで OpenText iMonitor を実行しなくても、OpenText iMonitor の機能の大部分を利用できます
- 1 つのサーバをアップグレードするだけで済みます
- 1 つのアクセスポイントでダイヤルインが可能です
- OpenText iMonitor 自体へのアクセスには低速なリンクを使用し、OpenText iMonitor から OpenText eDirectory 情報へのアクセスには高速なリンクを使用できます
- 以前のバージョンの NDS の情報にアクセスできます
- サーバ限定の機能は、iMonitor がインストールされているコンピュータ以外では使用できません

すべてのページからアクセス可能な iMonitor の機能

エージェントの要約、エージェント情報、エージェントの環境設定、トレースの環境設定、DSRepair、レポート、および検索の各ページには、ナビゲータフレームを使用することによってどの OpenText iMonitor ページからでもリンクできます。その他、どの OpenText iMonitor ページからでも、ログインしたり、OpenText サポートの Web ページにリンクしたりできます。

ログイン / ログアウト : ログインしていない状態では、[**ログイン**] ボタンが有効になります。ログインしている間は [**ログアウト**] ボタンが表示され、ログアウトするとブラウザウィンドウが閉じます。ブラウザウィンドウがすべて閉じられるまでは OpenText iMonitor セッションは開いたままになるため、再びログインする必要はありません。自分のログインステータスは、ナビゲータフレームに表示された識別情報を調べることによって、どのページからでも確認できます。

サポート接続リンク : ページ右上に表示される OpenText のロゴは、OpenText サポート接続 Web ページへのリンクになっています。ここから OpenText の Web サイトに直接リンクして、最新のサーバパッチキット、更新データ、各製品に固有なサポート情報などを取得できます。

環境設定ファイル

OpenText iMonitor に含まれる環境設定ファイルを使用すると、ユーティリティのデフォルトの動作や値を変更したり、設定したりできます。

環境設定ファイルはテキストファイルで、必要な値が指定された環境設定パラメータタグが含まれています。このファイルは、Windows 上では OpenText iMonitor の実行可能ファイルと同じディレクトリ (通常は OpenText eDirectory の実行可能ファイルと同じ場所) にあり、Linux 上では /etc ディレクトリにあります。

- 250 ページの 「ndsimon」
- 250 ページの 「ndsimonhealth」

ndsimon

ndsimon の環境設定ファイルでは、トレースファイルの設定の変更、サーバへのアクセス制御、コンテナのリスト表示または検索結果を表示する際のオブジェクトの最大表示数の設定、およびアイドル状態が何分続くと接続がログアウトするかを指定できます。

サーバ	設定ファイル
Windows	*install directory*\novell\NDS\ndsimon.ini
Linux	/etc/opt/novell/eDirectory/conf/ndsimon.conf

ndsimon の環境設定ファイルに設定するパラメータには、次のような 2 種類のグループがあります。

- OpenText iMonitor の実行可能ファイル自体の実行方法に適用されるパラメータ

  OpenText iMonitor の実行可能ファイルはロードされると、従来の HTTP ポート 80 でリスンしようとします。このポートが使用中の場合、待機するポートを 8028 に切り替えます。そのポートが使用中の場合は、OpenText iMonitor はさらにポートを切り替え、番号を 2 ずつ増やしながら (8010、8012 など)、8078 に達するまで使用可能なポート番号を検索します。最大 8078.

  SSL が設定され使用可能になっている場合も、同様のパターンでバインドが実行されます。この場合、最初にポート 81 がバインドされ、次に 8009、8011、8013 と続きます。

  これにより、OpenText iMonitor と、同じサーバで実行している Web サーバとの共存が可能になります。プラットフォームによっては、インストールされた Web サーバをロードする前に OpenText iMonitor をロードできます。また、OpenText iMonitor をバインドするポートを選択することもできます。通常のポートおよび SSL ポートは、HttpPort パラメータおよび HttpsPort パラメータをそれぞれに使用して設定できます。

- 特定の機能またはページに適用されるパラメータ

  OpenText iMonitor に付属する環境設定ファイルには、変更可能なパラメータのサンプルが含まれています。これらのパラメータの先頭にはシャープ記号 (#) が付いています。これは、パラメータがコメントアウトされていることを示していて、OpenText iMonitor が環境設定ファイルを解析するときには、これらのパラメータは無視されます。付属する環境設定ファイルでは、これらのパラメータにはすべて、内部でバインドされたデフォルト値が使用されます。これらのパラメータを使用可能にする、またはパラメータを追加するには、行の先頭の「#」を削除します。

ndsimonhealth

ndsimonhealth の環境設定ファイルでは、［エージェントヘルス］ページのデフォルト設定を変更できます。［エージェントヘルス］オプションを有効または無効にしたり、オプションのレポートレベルおよび範囲を設定したり、サーバのレポートレベルを設定できます。

サーバ	設定ファイル
Windows	*install directory*\novell\NDS\ndsimonhealth.ini
Linux	/etc/opt/novell/eDirectory/conf/ndsimonhealth.conf

ndsimonhealth の環境設定ファイルに設定するオプションには、次のような 3 種類のオプションがあります。

- オプションのみを有効または無効にする

  オプションを無効にするには、オプションの前のシャープ記号 (#) を削除し、コロン (:) の後ろにリスト表示されるすべてのレベルを「OFF」に置き換えます。これらのオプションのレポートレベルを設定するには、オプションの前の「#」文字を削除し、コロンの後ろにレポートレベルを追加します。有効なレベルは、WARN、MARGINAL、および SUSPECT です。これらのオプションに入力できるレポートレベルは 1 つだけです。

- 設定の範囲を指定する一般オプション

  これらのオプションでは、レポートレベルの設定を有効または無効にしたり、レポートレベルを設定したりできます。またレポートレベルの範囲の設定もできます。

  これらのすべてのオプションのレポートレベルを設定するには、オプション名の後に -active と記述し、その後ろに設定するレポートレベルを記述します。たとえば、time_delta をアクティブに設定するには、環境設定ファイルに次の行を追加します。

  ```
  time_delta-active: WARN
  ```

  time_delta を非アクティブに設定するには、環境設定ファイルに次の行を追加します。

  ```
  time_delta-active: OFF
  ```

  範囲を入力する場合、指定する範囲はこのレポートレベルを表示しない範囲です。

  3 つすべてのレポートレベルをアクティブにするオプションの設定方法、および範囲の設定方法については、次の time_delta の例を参照してください。この例では、-2 ～ 2 の範囲外では少なくとも marginal のレベルが表示され、-5 ～ 5 の範囲外では少なくとも suspect のレベルが表示され、-10 ～ 10 の範囲外では warning のレベルが表示されます。

  ```
  time_delta-active: WARN | SUSPECT | MARGINAL
  time_delta-Min_Warn:      -10
  time_delta-Min_Suspect:    -5
  time_delta-Min_Marginal:   -2
  time_delta-Max_Marginal:    2
  time_delta-Max_Suspect:     5
  time_delta-Max_Warn:        10
  ```

  これらのオプションのヘルプを表示するには、iMonitor で次の URL を入力します。

  ```
  http://XXX.XXX.XXX.XXX:PORT/nds/help?hbase=/nds/health/OPTION_NAME
  ```

  XXX.XXX.XXX.XXX:*PORT* には、OpenText iMonitor がアクセスできる IP アドレスとポート、*OPTION_NAME* にはヘルプ表示するオプション名 (time_delta など) を入力します。

現在の設定レベルと範囲を表示するには、ブラウザを使用して表示するオプションを含むヘルスページへ進み、ブラウザの URL 行の最後に次を追加します。

```
&op=setup
```

- カスタム設定または複合設定が必要なオプション

  設定できるレポートレベルには次の 3 種類があります。

  - WARN は、すぐにアップグレードする必要があるバージョンの OpenText eDirectory を実行しているサーバを検出します。
  - SUSPECT は、アップグレードが望まれるバージョンの OpenText eDirectory を実行しているサーバを検出します。
  - MARGINAL は、最新バージョンではない OpenText eDirectory を実行しているサーバを検出します。

  これらのオプションは、サーバのバージョンが指定された許容範囲にあるかどうかのレポートレベルを設定します。

OpenText iMonitor の機能

このセクションでは OpenText iMonitor の機能について簡単に説明します。

OpenText iMonitor が持つ各機能の詳細については、オンラインヘルプの該当するセクションを参照してください。

- 253 ページの 「OpenText eDirectory サーバのヘルス情報の表示」
- 253 ページの 「パーティション同期ステータスの表示」
- 254 ページの 「破損通知プロセスステータスおよびキャッシュの変更回数を表示する」
- 255 ページの 「サーバ接続情報の表示」
- 256 ページの 「認識されているサーバの表示」
- 256 ページの 「レプリカ情報の表示」
- 257 ページの 「DS エージェントを制御および環境設定する」
- 258 ページの 「トレースを環境設定する」
- 259 ページの 「プロセスステータス情報の表示」
- 259 ページの 「エージェントアクティビティの表示」
- 260 ページの 「トラフィックパターンの表示」
- 260 ページの 「バックグラウンドプロセスの表示」
- 261 ページの 「バックグラウンドプロセスの設定」
- 261 ページの 「OpenText eDirectory サーバエラーの表示」
- 262 ページの 「DSRepair 情報の表示」
- 262 ページの 「エージェントのヘルス情報を表示する」
- 262 ページの 「ツリー内のオブジェクトの参照」
- 263 ページの 「同期またはパージのためのエントリの表示」

- 264 ページの 「OpenText Identity Manager の詳細の表示」
- 264 ページの 「レプリカの同期ステータスの表示」
- 264 ページの 「レポートの設定と表示」
- 267 ページの 「スキーマ、クラス、および属性定義の表示」
- 267 ページの 「オブジェクトの検索」
- 268 ページの 「ストリームビューアの使用」
- 268 ページの 「DIB セットのクローン」

OpenText eDirectory サーバのヘルス情報の表示

［エージェントサマリ (Agent Summary)］ページでは、同期設定、エージェントプロセスのステータス、データベースで認識されているサーバの総数など、OpenText eDirectory サーバのヘルス情報を表示できます。

1 iMonitor で、［**エージェントの概要**］をクリックします。

2 次のオプションから選択します。

［**エージェント同期の概要**］では、レプリカの数とタイプ、およびこれらレプリカが正常に同期されてから経過した時間を表示できます。その他、レプリカのタイプ別にエラーの数を表示することもできます。表示できるレプリカまたはパーティションが 1 つだけの場合、見出しは「**パーティション同期ステータス**」になります。

［エージェント同期の概要］が表示されない場合、ユーザの識別情報に基づいて見れるレプリカがないということです。

［**データベースで認識されているサーバ合計**］では、ローカルデータベースが認識しているサーバのタイプと数、および各サーバが実行中であるかどうかを表示できます。

［**エージェントプロセスステータス合計**］では、エージェント上で実行されているプロセスのステータスを、管理者に依頼せずに自分で調べることができます。ステータス情報は問題や重要な情報が発生したときに記録されます。表示される表のサイズは、記録されているステータスの数によって異なります。

パーティション同期ステータスの表示

［エージェント同期］ページでは、パーティションの同期状態を表示できます。ページの左側のアシスタントフレームに一覧表示されているオプションから選択して、表示する情報を絞り込むこともできます。

1 OpenText iMonitor で、アシスタントフレームの［**エージェント同期**］をクリックします。

2 次のオプションから選択します。

［**パーティション同期ステータス**］では、パーティション、エラーの数、最終同期時刻、最大リングデルタを表示できます。

［**パーティション**］では、各パーティションの［レプリカ同期］ページへのリンクを表示できます。

［**最終同期日時**］では、サーバから個々のパーティションのレプリカをすべて同期できたときから経過した時間を表示できます。

［**最大リングデルタ**］では、リング内にあるすべてのレプリカに対して同期できない可能性があるデータ量を表示します。たとえば、ユーザが自分のログインスクリプトを変更してから 30 分が経過していない場合、最大リングデルタへの割り当てが 45 分であれば、このユーザのログインは正常に同期されないおそれがあります。その場合、このユーザがログインしようとすると以前のログインスクリプトを受け取ることになります。ユーザが 45 分以上前にログインスクリプトを変更している場合は、このユーザはすべてのレプリカから常に新しいログインスクリプトを受け取ることになります。

［**最大リングデルタ**］に［**不明**］が表示されている場合は、遷移同期ベクトルに不整合があり、レプリカ / パーティション操作が実行中などの理由で最大リングデルタを計算できないことを意味します。

破損通知プロセスステータスおよびキャッシュの変更回数を表示する

特定のパーティションの破損通知プロセスステータスおよびキャッシュの変更回数を表示するには、そのパーティションのパーティションルートオブジェクトに移動します。次の 3 種類の破損通知のデータが表示されます。

- OBIT_DEAD: オブジェクトが削除されるときに作成されます。
- OBIT_NEWRDN: オブジェクトの名前が変更されるときに作成されます。
- OBIT_MOVED: オブジェクトがある場所から別の場所に移されると作成されます。

オブジェクトが処理される過程で、4 つの状態を経ます。オブジェクトは ISSUED 状態から PURGEABLE 状態に移り、最終的にパージされます。4 つの状態は次のとおりです。

- ISSUED
- NOTIFIED
- OK_TO_PURGE
- PURGEABLE

特定のオブジェクトに 12 種類の組み合わせがあります。これらの組み合わせは次のとおりです。

- OBIT_DEAD_ISSUED
- OBIT_DEAD_NOTIFIED
- OBIT_DEAD_OK_TO_PURGE
- OBIT_DEAD_PURGEABLE
- OBIT_NEWRDN_ISSUED
- OBIT_NEWRDN_NOTIFIED
- OBIT_NEWRDN_OK_TO_PURGE
- OBIT_NEWRDN_PURGEABLE
- OBIT_MOVED_ISSUED
- OBIT_MOVED_NOTIFIED

- OBIT_MOVED_OK_TO_PURGE
- OBIT_MOVED_PURGEABLE

これらの組み合わせのそれぞれに対して番号が表示されます。番号は、最後の破損通知処理サイクルが終了した時点で、その特定の状態にあるオブジェクトの合計数を示します。

変更キャッシュカウントは、現行サーバのパーティションの変更キャッシュに入っているオブジェクトの数を表します。以下の図に、そのパーティションの特定のパーティションルートオブジェクトについての破損通知カウントと変更キャッシュカウントを示します。

***図8-3** 破損通知および変更キャッシュカウントの情報*

Obit and Change Cache Count Information	
OBIT_DEAD_ISSUED	8318
OBIT_DEAD_NOTIFIED	0
OBIT_DEAD_OK_TO_PURGE	1682
OBIT_DEAD_PURGEABLE	0
OBIT_NEWRDN_ISSUED	0
OBIT_NEWRDN_NOTIFIED	0
OBIT_NEWRDN_OK_TO_PURGE	0
OBIT_NEWRDN_PURGEABLE	0
OBIT_MOVED_ISSUED	0
OBIT_MOVED_NOTIFIED	0
OBIT_MOVED_OK_TO_PURGE	0
OBIT_MOVED_PURGEABLE	0
Obit Count from database index	10000
Change Cache Count	10002

サーバ接続情報の表示

［エージェント情報］ページでは、ローカルサーバの接続情報を表示できます。

1 OpenText iMonitor で、アシスタントフレームの［**エージェント情報**］をクリックします。

2 次のオプションから選択します。

［**Ping 情報**］には、サーバからの通知先アドレスのセットに対して、OpenText iMonitor が IP ping を試みたことが表示されます。表示されるのは、応答があった場合です。

［**DNS 名**］には、OpenText iMonitor がサーバによってサポートされている IP アドレスに対してアドレスの反転を試みたことを表示します。対応する DNS 名が表示されます。

使用しているトランスポート、環境設定、およびプラットフォームによっては、この情報が表示されない場合もあります。

［**接続情報**］サーバの参照、タイムデルタ、一番ルート側のマスタレプリカ、およびレプリカ深さなどのサーバの情報を表示できます。

使用しているトランスポート、環境設定、およびプラットフォームによっては、この情報が表示されない場合もあります。

［サーバ照会］ローカルサーバへのアクセスに使用できるアドレスを一括して表示できます。

時刻同期は、最後に発行されたレプリカのタイムスタンプが現在の時刻より遅れない限り、合成時刻や未来の時刻を使用しないことを示します。

OpenText eDirectory は、サーバの現在時刻に基づいてタイムスタンプを発行できる程度に時刻が同期されていることを想定しています。ただし、時刻同期プロトコルが同期状態にあることは保証されていません。

［タイムデルタ］では、OpenText iMonitor とリモートサーバの時刻の差を秒単位で表示できます。負の整数は OpenText iMonitor の時刻がサーバの時刻より進んでいることを意味します。正の整数は OpenText iMonitor の時刻がサーバの時刻より遅れていることを意味します。

［最もルートに近いマスタレプリカ］が表示されている場合、最上位のレプリカつまりネーミングツリーのルートに最も近いレプリカがマスタレプリカであることを意味します。

［レプリカ深さ］最上位レプリカの深さ (最上位レプリカとツリーのルートの間のレベル数) を表示します。

認識されているサーバの表示

［認識サーバ］のリストには、ソースサーバのデータベースが認識しているすべてのサーバがリストされます。フィルタ条件を指定して、データベースで認識されているすべてのサーバまたはレプリカリング内のすべてのサーバのリストを表示できます。サーバの横にアイコンが表示された場合、そのサーバはレプリカリングのメンバーです。

1 OpenText iMonitor で、アシスタントフレームの［認識サーバ］をクリックします。

2 次のオプションから選択します。

［エントリ ID］ローカルサーバのオブジェクト識別子を表示します。エントリ ID は、複数のサーバ間で共用することはできません。

［NDS リビジョン］には、通信相手のサーバにキャッシュまたは保存されている OpenText eDirectory ビルド番号または NDS バージョンがリスト表示されます。

［ステータス］サーバのステータス (稼動中、停止中、不明) を表示します。通信相手のサーバのステータスが不明となっている場合、過去にこのサーバと通信する必要がなかったことを表します。

［最終更新時刻］該当するサーバが相手サーバと最後に通信を試みて、その相手サーバが停止中であることを検出したときの時刻を表示します。このカラムが表示されない場合は、すべてのサーバが稼働中であることを意味しています。

レプリカ情報の表示

［パーティション］ページでは、通信相手のサーバ上にあるレプリカに関する情報を表示できます。ページの左側のアシスタントフレームに一覧表示されているオプションから選択して、表示するページの情報を絞り込むことができます。

［Server Partition Information］では、該当するサーバのパーティションについての情報 (エントリ ID、レプリカステータス、パージ時間、最終変更時刻など) を表示できます。

[パーティション] サーバのパーティションの Tree オブジェクトに関する情報を表示できます。

[パージ時間] すべてのレプリカが削除を認識しているためにすでに削除されたデータを、データベースから削除できる時間を示します。

[最終変更時刻] このレプリカのデータベースに書き込まれたデータの、最後に発行されたタイムスタンプを表示できます。これにより、将来の時刻が設定されていないかどうか、および合成時刻が使用されていないかどうかを確認できます。

[レプリカ同期] パーティションに対応する［レプリカ同期サマリ］ページを表示できます。［レプリカ同期］ページには、パーティション同期ステータスとレプリカステータスについての情報が表示されます。また、パーティションとレプリカのリストを表示することもできます。

DS エージェントを制御および環境設定する

［エージェント環境設定］ページでは、DS エージェントの制御および環境設定ができます。このページで利用できる機能は、現在の識別情報に基づく権利および使用している eDirectory のバージョンによって異なります。

1 OpenText iMonitor で、[エージェント環境設定 をクリックします。

2 次のオプションから選択します。

- [エージェント情報] ローカルサーバの接続情報を表示できます。
- [パーティション] 通信しているサーバ上にあるレプリカを表示できます。
- [レプリケーションフィルタ］では、指定した OpenText eDirectory エージェントに対して設定されたレプリケーションフィルタを表示できます。NDS OpenText eDirectory 8.5 (ビルドバージョン 85.*xx*) は、「フィルタ済みレプリカ」と呼ばれる機能を初めて実装した OpenText eDirectory バージョンです。フィルタ済みレプリカの使用と設定方法の詳細については、「66 ページの 「フィルタ済みレプリカ」」を参照してください。
- [エージェントトリガ] バックグラウンドプロセスを開始します。エージェントトリガは、機能的には SET DSTRACE=**option* コマンドと同じです。
- [バックグラウンドプロセス処理設定] 特定のバックグラウンドプロセスを実行する時間間隔を変更します。バックグラウンドプロセスの設定は、機能的には SET DSTRACE=!*option* コマンドと同じです。
- [エージェント同期] インバウンド同期やアウトバウンド同期を無効または有効にします。同期を無効にする期間 (単位は時間) を指定できます。
- [データベースキャッシュ］DS データベースエンジンが使用するデータベースキャッシュのサイズを設定します。提供されるさまざまなキャッシュ統計情報により、適切な量のキャッシュが利用可能であるかを判断できます。十分なキャッシュがないと、システムのパフォーマンスが悪化する原因となります。

注 : ファイルシステムキャッシュと使用可能な RAM を考慮してから、データベースキャッシュを割り当てることをお勧めします。

- ［**ログイン設定**］では、ユーザのログイン時に OpenText eDirectory がログイン属性を更新するかどうかを指定できます。次のオプションは、ユーザがログインする時の OpenText eDirectory の応答方法を制御します。
  - ［**ログイン更新の遅延**］は、更新の時間間隔 (秒) を指定します。たとえば、1 人以上のユーザが遅延の間にログインした場合、OpenText eDirectory はすべての変更をキューに追加します。遅延が終了すると、OpenText eDirectory はキューにあるすべての変更を適用します。
  - ［**ログイン更新無効間隔**］は、特定のユーザのログイン属性が更新されない時間間隔 (秒) を指定します。標準的な間隔は 3600 秒 (1 時間) です。たとえば、ユーザが午前 8 時に初めてログインした場合、OpenText eDirectory は属性を更新し、そこから間隔が開始します。ユーザが午前 9 時前に再びログインした場合、OpenText eDirectory は属性を更新しません。デフォルトは 0 です。無効間隔は設定されていません。

トレースを環境設定する

［トレースの環境設定］ページの情報にアクセスするには、サーバの管理者と同等の権利またはコンソールオペレータの権利が必要です。このページの情報にアクセスするには、その前に、資格情報を検証できるよう、ユーザ名とパスワードを入力する必要があります。

［トレースの環境設定］ページでは、トレースを設定できます。OpenText iMonitor の DSTrace は、サーバ限定機能です。つまり、この機能は iMonitor が動作しているサーバ以外からは起動できません。他のサーバで実行されているこの機能にアクセスするには、そのサーバで実行されている iMonitor に切り替える必要があります。

1 iMonitor で、［**トレースの環境設定**］をクリックします。

2 次のオプションから選択します。
- ［**更新**］トレースオプションおよびトレース行プリフィックスに変更を送信できます。DS トレースがオフになっている場合は、［**オン**］をクリックして DS トレースをオンにします。DSTrace がすでにオンになっている場合は、［**更新**］をクリックして現在のトレースに変更内容を送信します。
- ［**オン**］/［**オフ**］DS トレースをオンまたはオフにします。ボタン上に表示される文字列は、DS トレースが現在オンかオフかによって異なります。DSTrace がオンになっている場合、ボタンには［**オフ**］というテキストが表示されます。DS トレースのオンとオフを切り替えるには、このボタンをクリックします。DSTrace がオフになっているときに［**オン**］をクリックした場合の動作は、［**更新**］をクリックした場合と同じです。
- ［**トレース行プレフィックス**］このオプションを使用すると、トレース行の先頭に追加するデータを選択することができます。
- ［**DS トレースオプション**］トレースを開始したローカル DS エージェント上のイベントに対して適用されます。［DS トレースオプショ］ンを使用すると、エラーと潜在的な問題、およびローカルサーバ上の OpenText eDirectory に関する情報を表示できます。［DS トレースオプション］をオンにすると、CPU の負荷が増え、システムのパフォーマンスが低下します。このため、DS トレースオプションは診断の

みに使用して、通常はオフにします。DS トレースオプションは、SET DSTRACE=+*option* コマンドと機能的には同じですが、使いやすさの点では優れています。

- ［**イベント環境設定**］には、DSTrace で監視のために有効または無効にする OpenText eDirectory と NMAS のイベントオプションがリスト表示されます。イベントシステムは、オブジェクトの追加、削除、属性値の変更など、ローカルな操作に対してイベントを生成します。個々のイベントタイプ対して、そのイベントタイプに固有の情報が含まれた構造が返されます。
- **［トレース履歴］**以前に実行したトレースのリストを表示できます。各トレースログは、トレースデータの収集期間によって識別されます。
- **［トレーストリガ］**DS トレース内で、指定した DS エージェント情報を表示するために設定する必要があるフラグを表示できます。トレーストリガを有効にした場合、トレースに書き込まれる情報の量が多くなります。OpenText サポート部門からの指示があった場合にのみ、トレーストリガを有効にすることをお勧めします。

3 ［**オン**］をクリックして DSTrace をオンにし、変更内容を送信します。

4 または［**トレースライブ**］をクリックして、OpenText iMonitor で DS トレースを表示します。

プロセスステータス情報の表示

［エージェントプロセスのステータス］ページでは、バックグラウンドプロセスのステータスに関するエラーと発生した各エラーの詳細情報を表示できます。ページの左側のアシスタントフレームに一覧表示されているオプションから選択して、このページに表示する情報を絞り込むことができます。

OpenText iMonitor で、アシスタントフレームの［**エージェントプロセスのステータス**］をクリックします。現在、バックグラウンドプロセスのステータスで報告される情報には、次のものが含まれます。

- スキーマの同期
- 破損通知処理
- 外部参照 /DRL
- リンバ
- ［Repair］

エージェントアクティビティの表示

［エージェントアクティビティ］ページでは、トラフィックパターンや考えられるシステムボトルネックを調べることができます。このページでは、現在 OpenText eDirectory で処理されているバーブおよび要求が表示されます。また、これらのうち、データベースへの書き込みのために DIB ロックを取得しようとしている要求がどれかを特定したり、DIB ロックの取得待機中の要求数を調べることもできます。

OpenText eDirectory 8.6 以降を実行しているサーバを表示すると、パーティションのリスト、およびナビゲータフレームで指定したサーバが含まれるレプリカリングに参加しているサーバも表示されます。OpenText eDirectory 8.6 を導入すると、同期処理はシングルス

レッドではなくなります。OpenText eDirectory 8.6 以降のバージョンのサーバは、1 つ以上のレプリケーションパートナーに複数のパーティションを同時に発信する可能性があります。このため、このような並行同期処理の監視がさらに容易になるよう、［同期アクティビティ］ページが作成されています。

1 OpenText iMonitor で、アシスタントフレームの［**エージェントアクティビティ**］をクリックします。

2 次のオプションから選択します。

- ［**バーブアクティビティおよび統計情報 (Verb Activity and Statistics)**］では、OpenText eDirectory の最後の初期化以降に呼び出されたバーブの総数や発行された要求の数をリアルタイムで表示できます。また、現在アクティブになっている要求の数と、これらの要求を処理するための最小、最大、および平均時間 (ミリ秒単位) も表示されます。
- ［**Synchronization Current and Schedule(現在同期およびスケジュール)**］インバウンド同期およびアウトバウンド同期が発生したさまざまな時刻のリストを表示します。インバウンド同期またはアウトバウンド同期が現在実行中の場合、プロセスが実行中であること、そのサイクルが開始された時刻、およびそのプロセスを実行しているサーバを示すアイコンも表示されます。

  インバウンド同期およびアウトバウンド同期が無効になっている場合は、現在同期が無効であることおよび再び有効になる予定の時刻を示すアイコンが表示されます。アウトバウンド同期では、次に再び有効になる予定時刻も表示されます。
- **イベント**現在アクティブな状態にあるイベント、イベントハンドラの統計情報、イベント統計情報の概要、および呼び出された現在のイベント権利機能のリストを表示できます。
- ［**バックグラウンド処理スケジュール**］スケジュールされているバックグラウンド処理、その現在の状態、および再実行のスケジュールを表示できます。

トラフィックパターンの表示

［Verb 統計］ページでは、トラフィックパターンや考えられるシステムボトルネックを調べることができます。このページでは、OpenText eDirectory の最後に初期化されてから呼び出されたバーブの総数や発行された要求の数がリアルタイムで表示されます。その他、これら要求のうち現在アクティブなものの数や、これら要求の処理にかかる時間の最大値、平均値、最小値 (それぞれミリ秒単位) も表示できます。バックグラウンドプロセス、バインダリ、および標準 OpenText eDirectory 要求が追跡されます。

このページを古いバージョンの OpenText eDirectory で表示すると、OpenText eDirectory 8.5 以降で表示する場合より情報量が少なくなります。

バックグラウンドプロセスの表示

［バックグラウンド処理スケジュール］ページでは、スケジュールされているバックグラウンドプロセスを、現在の状態と次回の実行予定時刻とともに表示できます。OpenText iMonitor の［バックグラウンドプロセスのスケジュール (Background Process Schedule)］は、サーバ限定機能です。つまり、この機能は OpenText iMonitor が動作しているサーバ以外からは表示できません。他のサーバで実行されているバックグラウンドプロセスのスケジュール機能にアクセスするには、そのサーバで実行されている OpenText iMonitor に切り

替える必要があります。OpenText eDirectory 8.5 以降のバージョンにアップグレードしたサーバの数が増えれば、OpenText iMonitor のサーバ限定機能の利用範囲は広がります。その他のサーバ限定機能には、［DS トレース］ページおよび［DSRepair］ページなどがあります。

［バックグラウンド処理スケジュール］ページの情報にアクセスするには、サーバの管理者と同等の権利またはコンソールオペレータの権利が必要です。このページの情報にアクセスするには、資格情報の確認のためにログインする必要があります。

バックグラウンドプロセスの設定

管理者は、OpenText iMonitor の［バックグラウンドプロセスの設定］ウィンドウで次のいずれかのバックグラウンドプロセス遅延設定ポリシーを設定することで、バックグラウンドプロセスサイクルの実行時間を短縮することができます。

- CPU
- ハードメモリ制限
- パージャ処理遅延

バックグラウンドプロセスを設定するには、次の手順に従います。

1. OpenText iMonitor にログインします。
2. ［**エージェント環境設定**］>［**バックグラウンドプロセスの設定**］の順に選択します。
3. ［**バックグラウンドプロセス遅延設定 (ミリ秒)**］セクションまでスクロールダウンし、遅延間隔を 0 秒から 100 ミリ秒までの任意の値に設定します。

   デフォルトでは、3 つすべてのプロセスのスリープ時間を 100 秒に設定した［**ハード制限**］ポリシーが有効になっています。

   または

   ［**CPU Policy**］を選択し、必要に応じて適切に設定します。

   デフォルトでは、［最大 CPU 使用率］パラメータが 80% に設定され、［遅延上限］が 100 ミリ秒に設定されています。
4. ［**パージャ間隔**］フィールドに、遅延間隔を入力します。

   デフォルトでは、30 分に設定されています。この値は、要件に合わせて変更できます。

OpenText eDirectory サーバエラーの表示

［エラー索引］ページでは、OpenText eDirectory サーバ上で検出されたエラーについての情報を表示できます。検出されたエラーは、OpenText eDirectory 固有のエラー、および関連のあるその他のエラーの 2 つのフィールドに分けて表示されます。表示されるエラーのそれぞれに説明がハイパーリンクされていて、エラーの内容、考えられる原因、回復手段などがわかります。

1. OpenText iMonitor で、アシスタントフレームの［**エラー索引**］をクリックします。

   ［エラー索引］ページからは、エラーや技術情報に関して OpenText が提供している最新のドキュメントやホワイトペーパーにリンクできます。

DSRepair 情報の表示

［DSRepair］ページでは、検出された問題を表示したり、DIB セットのバックアップやクリーンアップを実行できます。OpenText iMonitor の DSRepair は、サーバ限定機能です。つまり、この機能は OpenText iMonitor が動作しているサーバ以外からは起動できません。他のサーバで実行されている DSRepair 情報にアクセスするには、そのサーバで実行されている OpenText iMonitor に切り替える必要があります。OpenText eDirectory の新しいバージョンにアップグレードしたサーバの数が増えれば、iMonitor のサーバ中心の機能がさらに使えるようになります。この機能以外には、［DS トレース］ページや［バックグラウンド処理スケジュール］ページなどがサーバ限定の機能です。

［DSRepair］ページの情報にアクセスするには、サーバの管理者と同等の権利またはコンソールオペレータの権利が必要です。このページの情報にアクセスするには、資格情報の確認のためにログインする必要があります。

1 OpenText iMonitor で、［**DSRepair** ］をクリックします。

2 次のオプションから選択します。

- ［**ダウンロード**］ファイルサーバから修復関連のファイルを取得できます。DSRepair ユーティリティが実行中の場合や、OpenText iMonitor の［DSRepair］ページから修復を開始した場合には、操作が完了するまでは dsrepair.log にアクセスできません。
- ［**古い DIB セットを削除**］では、赤い［**X**］をクリックすることで古い DIB セットを削除できます。

  警告 : この操作は元に戻すことができません。このオプションを選択すると、古い DIB セットがファイルシステムからパージされます。

- ［**DS Repair 拡張スイッチ**］問題の修正、問題のチェック、データベースのバックアップ作成などを実行できます。OpenText サポート部門から指示がない限り、［**サポートオプション**］フィールドに情報を入力する必要はありません。

3 ［**修復の開始**］をクリックして、サーバ上で DS Repair を実行します。

エージェントのヘルス情報を表示する

［エージェントヘルス］ページでは、指定した OpenText eDirectory エージェントのヘルス情報およびそのエージェントに関連するパーティションおよびレプリカリングを表示できます。

1 OpenText iMonitor で、アシスタントフレームの［**エージェントヘルス**］をクリックします。

2 リンクをクリックすると、詳細な情報が表示されます。

ツリー内のオブジェクトの参照

［参照］ページでは、ユーザのツリー内にある任意のオブジェクトを参照できます。ページ最上部のナビゲーションバーには、表示中のオブジェクトが存在するサーバ、およびオブジェクトへのパスが表示されます。ページの左側にある［レプリカ］フレームでは、実

パーティション上にある同じオブジェクトを表示またはアクセスできます。ページ内の下線付きオブジェクトをクリックすると、オブジェクトに関する詳しい情報が表示されます。また、ナビゲータフレーム内にある名前の任意の一部分をクリックすると、ツリーの上の階層を参照できます。

このページに表示される情報は、ログイン時の OpenText eDirectory 権、ブラウズするオブジェクトのタイプ、および実行している NDS または OpenText eDirectory のバージョンによって異なります。スーパーバイザ権でログインした場合、このページには XRef オブジェクトが表示されます。レプリカリストを使用して、レプリカの実コピーへジャンプできます。ダイナミックグループのオブジェクトを参照している場合、ダイナミックメンバーに対してタイムスタンプは表示されません。

[レプリカ同期] このオブジェクトを含むレプリカの同期ステータスを表示します。

[エントリ同期] サーバ側から見て同期が必要である属性を表示します。

[接続情報] には、OpenText iMonitor がこのオブジェクトの情報をどこで取得したかが表示されます。

[エントリ情報] オブジェクトの名前、フラグ、ベースクラス、変更タイムスタンプ、および接続情報のサマリを表示します。

[すべてのレプリカにエントリを送信] このエントリの属性を他のすべてのレプリカに再送信します。オブジェクトに多数の属性値がある場合、この処理には時間がかかることがあります。この処理では、そのオブジェクトの、他のすべてのコピーが同一になるわけではありません。他のレプリカが各属性を再考慮できるようにするだけです。

[すべて送信] (参照しているオブジェクトがパーティションルートの場合および［カスタムモード］オプションが有効になっている場合のみ表示されます)。このパーティション内のすべてのエントリを、パーティションのレプリカを保持しているすべてのサーバに再送信します。この処理では、そのオブジェクトの、送信されたすべてのコピーが同一になるわけではありません。他のレプリカが各オブジェクトとその属性を再検討できるようにするだけです。

同期またはパージのためのエントリの表示

［変更キャッシュ］ページでは、同期またはパージにおいてこのサーバが検討する必要のあるエンティティのリストを表示できます。このオプションを使用できるのは、ユーザがアクセスしているサーバが OpenText eDirectory 8.6 以降を実行しており、また、表示中のオブジェクトがパーティションルートである場合だけです。このページを表示するには、OpenText eDirectory サーバに対するスーパバイザ権が必要です。

[エントリ同期] エントリが同期を必要とする理由を判別できます。

注 : OpenText iMonitor の［キャッシュの変更］ページには、限られた数のオブジェクトだけが表示されます。サーバ上の特定のパーティションまたはすべてのパーティションの、変更キャッシュに入っているすべてのオブジェクトを表示するには、［レポート］ページで［変更キャッシュのダンプ］レポートを実行します。OpenText iMonitor のレポートの設定および実行方法の詳細については、264 ページの 「レポートの設定と表示」を参照してください。

OpenText Identity Manager の詳細の表示

［DirXML の概要］ページでは、ユーザのサーバで実行中のすべての DirXML ドライバ、各ドライバのステータス、保留中の関連付け、およびドライバの詳細のリストを表示できます。

1 OpenText iMonitor で、［**DirXML の概要** をクリックします。

2 次のオプションから選択します。

［ステータス］ 指定したドライバの現在の状態を表示します。表示されるステータスは、［停止］、［開始します］、［稼働中］、［シャットダウン保留中］、および［スキーマ取得中］です。

［起動オプション］ 選択したドライバの現在の起動オプションを表示します。

［保留中］ まだ作成されていない関連付けの数を表示します。

［ドライバ詳細］アイコンは、ユーザのサーバで実行中の DirXML ドライバに関する、加入者および発行者の詳細、XML ルール、フィルタ、および保留中の関連付けリストを表示します。このページには、最初の 50 個の保留中オブジェクトに関する詳細も表示されます。このページに表示される XML ルールを使用すると、指定した DirXML ドライバに対するオブジェクトの作成を続行するために必要な、保留中のオブジェクト内で検索すべき情報を判断できます。

レプリカの同期ステータスの表示

［レプリカ同期］ページでは、レプリカの同期ステータスを表示できます。

1 OpenText iMonitor で、アシスタントフレームの［**エージェント同期**］をクリックします。

2 表示するパーティションの［**レプリカ同期**］をクリックします。

3 このページにあるリンク、および左側のナビゲーションバーにあるリンクを使用すると、他のパーティションにアクセスしたり、レプリカリング内でジャンプすることができます。

レポートの設定と表示

［レポート］ページでは、このサーバで直接実行されているレポートを表示および削除できます。一部のレポートでは、実行に長時間を要し、多くのリソースを消費する場合があります。

スケジュールされたレポートは、［パブリック］識別情報を使用して、ユーザ認証なしで実行されます。ユーザが実行するレポートはすべて、ユーザの権利で直接実行されます。すべてのレポートデータは、レポートを実行したサーバに保管されます。デフォルトでは、iMonitor はオペレーティングシステムに応じて次のディレクトリにレポートデータを格納します。

プラットフォーム	ディレクトリ
Windows	C:\Novell\NDS\ndsimon\dsreports\
Linux	/var/opt/novell/eDirectory/data/dsreports

［レポートの環境設定］ページでは、事前に設定されたレポート、カスタムレポート、およびスケジュール設定されたレポートのリストを表示できます。このページを使用して、レポートを変更および実行できます。また、OpenText iMonitor ページ用のカスタムレポートの作成もできます。次の表に、OpenText iMonitor に含まれている事前設定のレポートを示します。

レポート	説明
サーバ情報	ツリー全体を調べて、検索可能な各 NCP サーバと通信し、検知したすべてのエラーをレポートします。このレポートを使用して、時刻同期および Limber の問題を診断できます。また、現在のサーバ自体が他のすべてのサーバと通信可能であるかどうかも知ることができます。環境設定ページで選択されている場合、このサーバはツリー内にある各サーバの NDS エージェントヘルス情報を生成することもできます。
破損通知リスティング	このサーバ上のすべての破損通知を表示します。
オブジェクト統計情報	オブジェクトを指定したスコープで調べて、要求される条件に一致したオブジェクトのリストを生成します。この条件には、将来の時刻、不明なオブジェクト、名前が変更されたオブジェクト、ベースクラス数、コンテナ、別名、外部参照などがあります。
変更キャッシュのダンプ	サーバ上の選択されたパーティションまたはすべてのパーティションの変更キャッシュにあるすべてのオブジェクトをリストします。このレポートは、サーバ間で同期する必要がある属性や値と共に、変更キャッシュにあるオブジェクトの XML ダンプも生成します。レポートには、変更キャッシュに含まれている、すべてのオブジェクトの分析に関する情報が示されます。 **注 :** OpenText iMonitor が変更キャッシュダンプを格納する場所は、上記の表にリストされている実際の変更キャッシュダンプレポートを格納するディレクトリと同じです。
サービスアドバタイジング	SLP または SAP を使用して現在のサーバに認識されている、すべてのディレクトリとサーバを表示します。
エージェントヘルス	現在のサーバのヘルス情報を収集します。
値数	指定した値より値数が多い属性を持つオブジェクトのリストを生成します。

レポートの表示と削除

1 OpenText iMonitor で、[レポート]をクリックします。

2 レポートを削除するには、をクリックし、レポートを表示するには をクリックします。

レポートの実行

1 OpenText iMonitor で、[レポート] > [レポート設定] の順にクリックします。

2 をクリックしてレポートを実行します。

レポートの設定またはスケジュールを行う

1 OpenText iMonitor で、[レポート] > [レポート設定] の順にクリックします。

2 をクリックし、レポートを設定およびスケジュールします。

3 目的のオプションを選択し、[デフォルトの保存] をクリックして選択したオプションを保存します。

4 (オプション) レポートが定期的に、または後で実行されるように設定します。

   4a レポートの頻度、開始時刻、および開始日を指定します。

   4b [スケジュール] をクリックします。

5 [レポートの実行] をクリックしてレポートを開始します。

カスタムレポートの作成

カスタムレポートを作成すると、OpenText iMonitor の任意のページをレポートとして起動できます。

1 OpenText iMonitor で、[レポート] > [レポート設定] の順にクリックします。

2 [実行可能レポート] リストで、 [カスタムレポート] をクリックします。

3 レポートの名前を入力し、レポートとして起動する OpenText iMonitor ページの URL を入力します。

   カスタムレポートを実行する場合は、次の URL を入力します。

   /nds/*required page*

4 [保存されるレポート] フィールドに、保持または保存するレポートのバージョンの数を指定します。

5 (オプション) [保存] をクリックしてレポートを保存します。

6 (オプション) レポートが定期的に、または後で実行されるように設定します。

   6a レポートの頻度、開始時刻、および開始日を指定します。

   6b [スケジュール] をクリックします。

7 [レポートの実行] をクリックしてレポートを開始します。

スキーマ、クラス、および属性定義の表示

［スキーマ］ページでは、ユーザのスキーマ、クラス、および属性の定義を表示できます。すでに作成されている拡張や特定のスキーマに固有の情報 (スキーマに行った変更や拡張など) を添付して、ツリー上にロードされているスキーマを表示できます。

1 OpenText iMonitor で、アシスタントフレームの［**スキーマ**］をクリックします。

2 次のオプションから選択します。

［同期リスト］このサーバと同期する相手のサーバを表示します。このオプションは、NDS OpenText eDirectory8.5 以降を実行しているサーバに対してのみ使用できます。この情報を表示するには、サーバに対するスーパーバイザ権が必要です。

［スキーマルート］ツリーのルートに最も近いスキーマレプリカに関する情報を表示します。

各 OpenText eDirectory サーバには、そのスキーマ全体のレプリカが保存されています。スキーマレプリカは、ディレクトリオブジェクトを格納しているパーティションから分割されて保存されます。任意のスキーマレプリカへの変更内容は、すべてのレプリカに伝えられます。スキーマの変更は、ルートパーティションの書き込み可能なレプリカを保存するサーバを通してのみ実行できます。ルートパーティションの読み込み可能なレプリカを保存しているサーバは、スキーマ情報を読み込むことはできますが、変更はできません。

属性定義各属性の名前、属性値が含まれる構文、および属性が受ける制約がリストされます。左側のナビゲーションフレームを使用すると、個々の属性をブラウズしたり、それらにアクセスしたりすることができます。

［クラス定義］各クラスの名前、ルール、および属性を表示します。左側のナビゲーションフレームを使用すると、個々の属性を参照したり、それらにアクセスすることができます。

オブジェクトの検索

［検索］ページでは、さまざまなクエリオプションおよびフィルタに基づいて、オブジェクトを検索できます。検索クエリオプションおよびフィルタは、基本フォームとカスタムフォームという 2 つのレベルの検索要求フォームに分けられます。単純検索要求フォームは、OpenText eDirectory の一般ユーザ向けであり、単純な検索に使用します。カスタム検索要求フォームは、熟練ユーザ向けであり、複雑な検索に使用します。現在はサーバレベルの検索のみがサポートされています。

4 つのセクション内の検索オプションおよび検索フィルタは、すべて結合可能です。空白フィールド (相対識別名を除く) は無視されます。<Ctrl> キーを使用して、マルチリスト上でアイテムを選択解除したり、複数のアイテムを選択することができます。選択解除したマルチリストも無視されます。

1 OpenText iMonitor で、［**検索** をクリックします。

2 次のオプションから選択します。

- **［スコープオプション］**検索のスコープを指定できます。
- **［エントリフィルタ］**エントリ情報に関連する検索クエリフィルタを指定できます。

- ［属性と値のフィルタ］属性および値に関連する検索クエリフィルタを指定できます。
- ［表示オプション］検索結果の表示形式を制御するオプションを指定できます。

注 : ［表示オプション］設定を使用できるのは、［詳細］をクリックしてすべての［高度な検索］オプションを表示している場合のみです。

3 検索要求フォームの一番下にある［ヘルプ］ボタンをクリックすると、そのフォームに関連する簡潔なヘルプ情報が表示されます。

ヘルプ情報をクリアするには、［再ロード］または［リフレッシュ］をクリックします。

ストリームビューアの使用

［ストリームビューア］ページでは、次の形式で現在のストリームを表示できます。

- プレーンテキスト
- HTML
- GIF
- JPEG
- BMP
- WAV
- 16 進ダンプ
- その他

特定の形式で常に表示したいストリーム属性がある場合、［ストリームビューア］を使用してデフォルトの表示設定を選択します。

［NDS ストリーム属性セットアップ］ブラウザでストリームを表示するためのデフォルトの形式を変更します。ストリームが正しく表示されるかどうかはブラウザに依存します。ブラウザによっては、ユーザが選択した設定が適用されない場合があります。

デフォルト設定に加えた変更を適用するには、ユーザがサーバに認証される必要があります。変更内容は streams.ini (Windows サーバの場合) または streams.conf (Linux サーバの場合) に保存されます。デフォルトの設定を手動で編集することもできます。

DIB セットのクローン

このオプションでは、1 つのサーバ (ソースサーバ) に保存されている OpenText eDirectory データベースの DIB ファイルセットを完全に複製できます。DIB クローンは、ツリー内のすべてのマスタレプリカが保持されているソースサーバから複製する必要があります。クローンは別のサーバ (ターゲットサーバ) に配置することができます。ターゲッ

トサーバが OpenText eDirectory を開始すると、サーバは DIB ファイルセットをロードし、サーバオブジェクトのマスタレプリカに接続し、名前を解決し、クローン作成後に行われた DIB ファイルセットのすべての変更を同期します。

OpenText eDirectory DIB セットのクローンは、クローンを作成したサーバのオペレーティングシステムと同じオペレーティングシステムが稼働するサーバ上にだけ配置する必要があります。たとえば、DIB ファイルセットのクローンを Linux サーバに復元する場合は、Windows サーバではなく Linux サーバでクローンを作成します。

この機能のバックエンドは eDirectory 8.7 に搭載されていましたが、OpenText eDirectory 8.7.1 で OpenText iMonitor 2.4 以降が稼働するようになるまでサポートされていませんでした。このオプションは、バージョンが 8.7 以前の OpenText eDirectory や NDS では使用できません。

図 8-4 *OpenText iMonitor の［DIB セットのクローン］ページ*



このセクションでは、次の情報を紹介します。

- 269 ページの 「DIB セットのクローンの使用事例」
- 270 ページの 「クローンを作成する」

DIB セットのクローンの使用事例

DIB セットのクローンは次のような場合に使用します。

- すでに「オン」の状態になっているパーティションで新しいサーバを作成します。

次の利点があります。

- レプリカリングに新しいサーバを追加する際に、リング内のすべてのサーバが稼動中または実行中である必要がありません。
- 新しいサーバは自動的にすべてのパーティションを保持しますが、同期する必要はありません。
- すばやく処理できます。

- 障害回復

長所	短所
• パーティションを1度コピーするだけで成功します。 • 複数のパーティションをもつ大きなサーバを停止させる時間が少なくてすみます。	• 対象のパーティションを少なくとも1度は正しくコピーする必要があります。 • SSL やセキュリティバックアップに対応しません。 • ファイルシステムを処理しません。

- バックアップおよび復元

長所	短所
• 大規模のデータベースでは特に、処理に時間がかかりません。	• OpenText eDirectory のコアを追加するだけです。LDAP、SNMP、SSL などはインストールされません。また設定も行われません。 • 最新の変更は取得されません。スナップショットのみが取得されます。ロールフォワードログは実行されません。

このような欠点があるため、バックアップ処理および復元処理のために DIB セットのクローンを使用することはお勧めできません。

クローンを作成する

DIB ファイルセットのクローンは、元のサーバでオンラインまたはオフラインのいずれでも作成できます。オフラインで行う場合は、OpenText eDirectory を停止させておく必要があります。オンラインモードでは、OpenText eDirectory は稼働状態になり、ロックされません。

- 271 ページの 「オンラインによる方法」
- 272 ページの 「EBA を無効にしたオフラインによる方法」
- 274 ページの 「EBA を有効にしたオフラインによる方法」
- 275 ページの 「OpenText eDirectory の設定を完了する」

警告 : 別のサーバのクローンを作るために、識別情報管理サーバで Dibclone ユーティリティを使用しないでください。これを行うと、クローンのサーバに不要な TAO ファイルが生成されてしまいます。

オンラインによる方法

1 ndsclone モジュールをソースサーバにロードします。

プラットフォーム	スキーマを拡張するには、次の操作を行います。
Windows	NDSCons.exe で **dsclone.dll** を選択し、[**開始**] をクリックします。
Linux	ndsmodules.conf ファイルに「ndsclone」エントリを追加し、http://IP address:port/dhost ページを使用して Directory Clone Agent をロードします。 **注 :** ndsclone モジュールは、ndstrace -c "load ndsclone" コマンドでロードすることもできます。

2 DIB クローンプロセスをソースサーバで開始する前に、OpenText iMonitor の [エージェント環境設定] ページで、インバウンド同期を無効にします。

3 DIB ファイルセットのクローンを作成します。

3a OpenText iMonitor で、DIB 環境設定のクローンを実行します。

[**エージェント環境設定**] > [**DIB セットのクローン**] > [**新しいクローンの作成**] の順にクリックします。

3b ターゲットサーバの完全修飾名と、DIB ファイルのクローンが配置される場所のファイルパスを指定してから、[**クローンオブジェクトの作成**] チェックボックスと [**DIB をオンラインでクローン**] チェックボックスをオンにします。

ターゲットサーバの NCP サーバ名 (クローンオブジェクト) は、ターゲットサーバの名前と一致させる必要があります。

3c **送信**をクリックします。

NDS クローンオブジェクトが作成され、DIB ファイルセットが指定された配置先にコピーされます。

4 ターゲットサーバで OpenText eDirectory のインストールと構成を行い、サーバを停止します。

5 クローンの DIB ファイルセットが格納されている DIB ディレクトリをターゲットサーバにコピーします。

さらに、Linux システムの場合は、/etc/opt/novell/eDirectory/conf/nds.conf ファイルをソースサーバからターゲットサーバにコピーし、ターゲットサーバへの次の参照を更新します。

- 次のパラメータに関する IP アドレスの変更
  - n4u.server.interfaces
  - http.server.interfaces
  - https.server.interfaces

- 手順 3b で作成された NCP サーバ名を n4u.nds.server-name パラメータに指定します。
- 優先サーバ名を n4u.nds.preferred-server パラメータに指定します。通常、ターゲットサーバのホスト名が優先サーバ名と見なされます。

6 ターゲットサーバの /var/opt/novell/nici/0 と /var/opt/novell/nici/0/backup から nicisdi.key を削除します。

7 次に、ターゲットサーバを起動し、ndsconfig upgrade コマンドを実行します。

注 : Windows では、サイレントインストーラを使用して OpenText eDirectory サーバをアップグレードするために、EConfig.ps1 コマンドを実行する必要があります。アップグレードのときには、upgrade.ni レスポンスファイルで、クローンされた DIB のツリー名、サーバ名、および ADMIN 資格情報を指定しなければなりません。また、ツリーの他のサーバの IP が含まれる既存のサーバ IP も指定する必要があります。詳細については、『*OpenText™ eDirectory CE 24.4 (v9.3) インストールガイド*』の「Windows での OpenText eDirectory の無人アップグレード」を参照してください。

---

8 ターゲットサーバオブジェクトのマスタレプリカが OpenText eDirectory を実行していて、使用可能になっていることを確認します。ターゲットサーバ上で OpenText eDirectory が初期化されると、OpenText eDirectory はターゲットサーバの最終的な名前が解決できるマスタレプリカと通信を行います。

9 ターゲットサーバのレプリカ属性値が、すべてのサーバと同期されていることを確認します。すべてのサーバで属性の変更が適用されたら、ソースサーバのインバウンド同期を再び有効にします。インバウンド同期は、OpenText iMonitor の［エージェント環境設定］ページまたは DSTrace で有効にできます。

10 OpenText eDirectory の環境設定を完了するには、275 ページの 「OpenText eDirectory の設定を完了する」を参照してください。

EBA を無効にしたオフラインによる方法

1 DIB ファイルセットのクローンを作成します。

1a OpenText iMonitor で、DIB 環境設定のクローンを実行します。

［**エージェント環境設定**］>［**DIB セットのクローン**］>［**新しいクローンの作成**］の順にクリックします。

1b ターゲットサーバの完全修飾名を指定し、［**クローンオブジェクトの作成**］チェックボックスをオンにして、［**DIB をオンラインでクローン**］チェックボックスをオフにします。

ターゲットサーバの NCP サーバ名は、ターゲットサーバの名前と一致させる必要があります。

1c **送信**をクリックします。

NDS クローンオブジェクトが作成されます。ソースサーバ上の DIB はロックされているため、OpenText eDirectory がロックされているというエラーが報告されます。

2 ターゲットサーバで OpenText eDirectory のインストールと構成を行い、サーバを停止します。

3 ソースサーバの DIB ディレクトリにある *.nds、nds*、および nds.rfl/*.* ファイルを、ターゲットサーバ上のコピー先またはメディアに手動でコピーします。セットをターゲットサーバの DIB ディレクトリに移動するのに都合のよい場所にします。さらに、Linux システムの場合は、/etc/opt/novell/eDirectory/conf/nds.conf ファイルをターゲットサーバに転送し、ターゲットサーバに対する次の参照を更新します。

- 次のパラメータに関する IP アドレスの変更
  - n4u.server.interfaces
  - http.server.interfaces
  - https.server.interfaces
- 手順 1b で作成された NCP サーバ名を n4u.nds.server-name パラメータに指定します。
- 優先サーバ名を n4u.nds.preferred-server パラメータに指定します。通常、ターゲットサーバのホスト名が優先サーバ名と見なされます。

4 ターゲットサーバの /var/opt/novell/nici/0 と /var/opt/novell/nici/0/backup から nicisdi.key を削除します。

5 NDSD_DISABLE_INBOUND=Y 環境変数をエクスポートしてから、ndsd を起動してソースサーバ上のインバウンド同期を無効にします。

6 ソースサーバ上で OpenText eDirectory を再起動します。

ファイルがコピーされる前にソースサーバ上で OpenText eDirectory が再起動された場合、このクローンは無効とみなされます。その場合は、新しい NCP サーバオブジェクトを削除してクローンを再作成する必要があります。

7 次に、ターゲットサーバを起動し、ndsconfig upgrade コマンドを実行します。

注 : Windows の場合、eDirectory セットアップファイルを実行する必要があります。また、セットアップファイルを実行して OpenText eDirectory サーバをアップグレードするときに、eDirectory ツリーを選択してログインすることも必要です。

---

8 ターゲットサーバのレプリカ属性値が、すべてのサーバと同期されていることを確認します。すべてのサーバで属性の変更が適用されたら、ソースサーバのインバウンド同期を再び有効にします。インバウンド同期は、OpenText iMonitor の［エージェント環境設定］ページまたは DSTrace で有効にできます。

9 DIB ディレクトリにクローンの DIB ファイルセットが格納されているターゲットサーバに、OpenText eDirectory をインストールし、サーバを起動します。

新しいターゲットサーバオブジェクトのマスタレプリカが OpenText eDirectory を実行していて、使用可能になっていることを確認します。ターゲットサーバ上で OpenText eDirectory が初期化されると、OpenText eDirectory はターゲットサーバの最終的な名前が解決できるマスタレプリカと通信を行います。

10 OpenText eDirectory の環境設定を完了するには、275 ページの 「OpenText eDirectory の設定を完了する」を参照してください。

EBA を有効にしたオフラインによる方法

前提条件

- DIB セットのクローンはマスタレプリカから作成する必要があります。EBA を有効にして DIB のクローンを作成している間、R/W レプリカはサポートされません。マスタレプリカのステータスを確認するには、/etc/opt/novell/eDirectory/conf フォルダで次のコマンドを実行します。

  ```
  ndsstat -r
  ```

- ソースサーバ上で ndscheck コマンドを実行して、EBA のステータスをキャプチャします。

DIB のクローン作成の実行

1 DIB ファイルセットのクローンを作成します。

  1a OpenText iMonitor で、DIB 環境設定のクローンを実行します。

  ［**エージェント環境設定**］>［**DIB セットのクローン**］>［**新しいクローンの作成**］の順にクリックします。

  1b ターゲットサーバの完全修飾名を指定し、［**クローンオブジェクトの作成**］チェックボックスをオンにして、［**DIB をオンラインでクローン**］チェックボックスをオフにします。

  ターゲットサーバの NCP サーバ名は、ターゲットサーバの名前と一致させる必要があります。

  1c **送信**をクリックします。

  NDS クローンオブジェクトが作成されます。ソースサーバ上の DIB はロックされているため、OpenText eDirectory がロックされているというエラーが報告されます。

2 EBA を有効にせずに、ターゲットサーバで OpenText eDirectory のインストールと構成を行い、サーバを停止します。

3 ソースサーバの DIB ディレクトリにある *.nds、nds*、および nds.rfl/*.* ファイルを、ターゲットサーバ上のコピー先またはメディアに手動でコピーします。セットをターゲットサーバの DIB ディレクトリに移動するのに都合のよい場所にします。さらに、Linux システムの場合は、/etc/opt/novell/eDirectory/conf/nds.conf ファイルをターゲットサーバに転送し、ターゲットサーバに対する次の参照を更新します。

  - 次のパラメータに関する IP アドレスの変更
    - n4u.server.interfaces
    - http.server.interfaces
    - https.server.interfaces
  - 手順 1b で作成された NCP サーバ名を n4u.nds.server-name パラメータに指定します。
  - 優先サーバ名を n4u.nds.preferred-server パラメータに指定します。通常、ターゲットサーバのホスト名が優先サーバ名と見なされます。
  - パラメータエントリ n4u.server.eba_enabled をファイルから削除します。

4 ターゲットサーバの /var/opt/novell/nici/0 と /var/opt/novell/nici/0/backup から nicisdi.key を削除します。

5 ソースサーバ上でインバウンド同期を無効にするため、/etc/opt/novell/eDirectory/conf にある env ファイルにエントリを追加して、NDSD_DISABLE_INBOUND=Y 環境変数をエクスポートします。

6 ソースサーバ上で OpenText eDirectory を再起動します。

6a ndscheck コマンドを実行して、EBA ステータスをキャプチャします。

6b 前提条件とステップ 6a でキャプチャされた EBA のステータスを比較します。両方の状況で EBA のステータスに違いがある場合は、ソースサーバで eDirectory を再起動します。

6c ソースサーバで OpenText eDirectory の再起動が完了したら、ndscheck コマンドを再度実行して、EBA のステータスが前提条件と同じであることを確認します。

注 : ファイルをコピーする前にソースサーバ上で OpenText eDirectory が再起動された場合、クローンは無効とみなされます。その場合は、新しい NCP サーバオブジェクトを削除してクローンを再作成する必要があります。

---

7 次に、ターゲットサーバで OpenText eDirectory を起動し、ndsconfig upgrade コマンドを実行します。

7a EBA を設定するように求めるプロンプトが表示されたら、[n] を選択します。

7b ndsconfig upgrade の実行中に -601 エラーが発生した場合は、ソースサーバから NDSD_DISABLE_INBOUND=Y を削除してソースサーバ上で同期を有効にし、ソースサーバで OpenText eDirectory を再起動します。

7c ソースサーバで OpenText eDirectory を再起動したら、ターゲットサーバで ndsconfig upgrade コマンドを実行します。EBA を設定するように求めるプロンプトが表示されたら、[n] を選択します。

注 : Windows の場合、OpenText eDirectory セットアップファイルを実行する必要があります。また、セットアップファイルを実行して OpenText eDirectory サーバをアップグレードするときに、OpenText eDirectory ツリーを選択してログインすることも必要です。

---

8 ndsconfig upgrade コマンドを再度実行し、オプション [y] を選択して、ターゲットサーバで EBA を有効にします。

新しいターゲットサーバオブジェクトのマスタレプリカが OpenText eDirectory を実行していて、使用可能になっていることを確認します。ターゲットサーバ上で OpenText eDirectory が初期化されると、OpenText eDirectory はターゲットサーバの最終的な名前が解決できるマスタレプリカと通信を行います。

9 OpenText eDirectory の環境設定を完了するには、275 ページの 「OpenText eDirectory の設定を完了する」を参照してください。

OpenText eDirectory の設定を完了する

- 「SIDKEY」(237 ページ)
- 「SAS、LDAP、および SNMP サービスを設定する」(237 ページ)

SDIKEY

1 ターゲットサーバ上で、OpenText eDirectory を停止します。

2 ターゲットサーバのファイルシステムで、/var/opt/novell/nici/0/nicisdi.key ファイルと /var/opt/novell/nici/0/backup/nicisdi.key ファイルを移動または名前変更します。

プラットフォーム	ディレクトリ
Windows	C:\Windows\SysWOW64\novell\nici\nicisdi.key
Linux	/var/opt/novell/nici/0/nicisdi.key
	/var/opt/novell/nici/0/backup/nicisdi.key

3 ターゲットサーバ上で、OpenText eDirectory を開始します。

SAS、LDAP、HTTP、および SNMP サービスを設定する

Linux: コマンドラインで次のコマンドを入力することで、1 つの操作で SAS、LDAP、SNMP、および HTTP サービスを設定できます。

ndsconfig upgrade [-a admin FDN]

Windows: OpenText eDirectory インストーラを実行し、SAS、LDAP、SNMP、および HTTP サービスの設定を完了します。

設定の完了後、デフォルトでは HTTP がポート 80 および 443 でリスンします。OpenText eDirectory は、HTTP のポート設定を HTTP サーバオブジェクトに格納します。必要に応じて、管理者ユーザとしてポート設定を変更できます。

サービスを個別に設定する場合は、次の表を参照してください。

SAS

プラットフォーム	コマンドまたはツール
Windows	OpenText Identity Console を使用して、SAS サービスオブジェクトと証明書を作成します。

LDAP

プラットフォーム	コマンドまたはツール
Windows	OpenText Identity Console を使用して、LDAP サーバとグループオブジェクトを作成します。

［SNMP］

プラットフォーム	コマンドまたはツール
Windows	rundll32 snmpinst, snmpinst -c オブジェクトの作成 -a ユーザ FDN -p パスワード -h ホスト名または IP アドレス

OpenText iMonitor の操作のセキュリティ保護

OpenText iMonitor 環境へのアクセスをセキュリティ保護するには、次の保護手順を実行します。

1. ファイアウォールを使用して VPN アクセスを準備します。これは、OpenText Identity Console および、アクセス制限が必要な他のすべての Web ベースのサービスの場合も同様です。
2. ファイアウォールが設置されているかどうかに関係なく、OpenText iMonitor 経由で許可されるアクセスの種類を制限することによって、DoS(Denial of Service) 攻撃に対する保護を強化できます。

   OpenText iMonitor は URL 要求を経由して受け取るデータを可能な限り検証しますが、すべての不正な入力を拒否することはほぼ不可能です。無効な URL を通じた DoS 攻撃の危険を減らすため、OpenText iMonitor の環境設定ファイルの［LockMask:］オプションを使用して 3 つのレベルのアクセスが制御されます。

アクセスレベル	説明
0	OpenText iMonitor の URL 処理において事前の認証は不要です。この場合、［パブリック］識別子の OpenText eDirectory 権利がすべての要求に適用され、iMonitor が表示する情報は［パブリック］ユーザの権利で表示できるものに限定されます。ただし、OpenText iMonitor に URL を送る際に認証が不要であるため、OpenText iMonitor は、不正な URL の送信による DoS 攻撃を受けやすくなる可能性があります。
1(デフォルト)	OpenText iMonitor が URL を処理する前に、OpenText eDirectory 識別子としての認証が必要です。この場合、その識別子の eDirectory 権利はすべての要求に適用されるため、eDirectory 権利によって制限を受けます。DoS 攻撃を受ける危険性はレベル 0 と同様ですが、DoS 攻撃は実際にサーバに認証を受けたものでないと行うことができないことが異なります。認証が正常に実行されるまでは、すべての OpenText iMonitor の URL 要求への返答は［ログイン］ダイアログボックスで行われます。したがってこの状態に設定されているときには、OpenText iMonitor は認証されていないユーザによる攻撃を通さないようにする必要があります。

アクセスレベル	説明
2	OpenText iMonitor が URL を処理する前に、OpenText iMonitor が認証しているサーバ上のスーパバイザに相当する OpenText eDirectory 識別子としての認証が必要です。DoS 攻撃を受ける危険性はレベル 1 と同様ですが、DoS 攻撃を行うには実際にサーバのスーパバイザとして認証される必要があります。認証が正常に実行されるまでは、すべての OpenText iMonitor の URL 要求への返答は［ログイン］ダイアログボックスで行われます。したがってこの状態に設定されているときには、OpenText iMonitor は認証されていないユーザおよびスーパバイザとして認証されていないユーザからの攻撃を通さないようにする必要があります。

レベル 1 はデフォルトです。多くの管理者はツリー内のすべてのサーバにアクセスできるスーパバイザ権を持っていませんが、管理しているサーバと通信するサーバ上の OpenText iMonitor サービスを使用する必要が生じる可能性があるためです。

注 : OpenText iMonitor には Repair、Trace など複数の機能があり、これらの機能にアクセスするには、LockMask の設定に関係なくスーパバイザに相当する権利が必要です。

HTTP サーバオブジェクトの設定

OpenText eDirectory をインストールすると、HTTP サーバオブジェクトが作成されます。このオブジェクトのディレクトリに、HTTP サーバのデフォルトの設定があります。ただし、このデフォルト設定は、OpenText Identity Console を使用して変更することができます。HTTP サーバオブジェクトとは、サーバ固有の環境設定データのことです。

HTTP サーバオブジェクトには、次の属性があります。

- **httpDefaultTLSPort:** HTTP サーバがリスンするセキュアポートを指定します。
- **httpDefaultClearPort:** HTTP サーバがリスンするクリアテキストポートを指定します。
- **httpAuthRequiresTLS:** クリアテキストポート経由の要求をセキュアポートにリダイレクトするかどうかを指定します。
- **httpTraceLevel:** DSTrace での HTTP サーバのデバッグレベルを指定します。デフォルトのトレースレベルは 2 です。
- **httpKeyMaterialObject:** HTTP サーバがセキュア接続を処理するときに使用する必要がある証明書オブジェクトの DN が格納されます。Suite B モードで OpenText iMonitor インタフェースを設定するには、httpBindRestrictions の値を Suite B モードに設定して目的の Suite B モードを有効にしてから、httpKeyMaterialObject に適切な ECDSA サーバ証明書を関連付けます。デフォルトでは、httpKeyMaterialObject は RSA 証明書を使用するように設定されています。
- **httpSessionTimeout:** HTTP セッションのタイムアウトを指定します。デフォルト値は 900 秒です。
- **httpKeepAliveRequestTimeout:** 各 HTTP 要求のキープアライブタイムアウトを指定します。デフォルト値は 15 秒です。

- **httpRequestTimeout:** 各 HTTP 要求のタイムアウトを指定します。デフォルト値は 300 秒です。
- **httpIOBufferSize:** HTTP サーバの入力および出力バッファサイズを指定します。デフォルトは 8192 バイトです。
- **httpThreadsPerCPU:** CPU ごとに生成する HTTP スレッドの数を指定します。デフォルト値は 2 スレッドです。
- **httpHostServerDN:** 関連付けられている NCP サーバオブジェクトの DN が格納されます。
- **httpBindRestrictions:** サイファの暗号化レベルを設定するために使用できます。
  - **RSA:** 次の値を使用して、サイファの使用を制限できます。
    - 0 - HIGH、MEDIUM、LOW、および EXPORT サイファを許容
    - 1 - HIGH、MEDIUM、および LOW サイファだけを許容
    - 2 - HIGH および MEDIUM サイファだけを許容
    - 3 - HIGH サイファだけを許容

    デフォルト値は 3 です。
  - **ECDSA 256:** 次の値を使用して、サイファの使用を制限できます。
    - 4 - 128 ビットのサイファまたは 256 ビットのサイファを許可
  - **ECDSA 384:** 次の値を使用して、サイファの使用を制限できます。
    - 5 - 128 ビットのサイファまたは 256 ビットのサイファを許可
    - 6 - 256 ビットのサイファを許可

  ECDSA 証明書の場合、OpenText eDirectory は Suite B サイファのみを許可します。

  Suite B モードで LDAP および httpstk インタフェースを設定するには、管理者の権利で OpenText Identity Console にログインし、Suite B モードのいずれかを有効にしてから、これらのインタフェースに適切な ECDSA サーバ証明書を関連付けます。この作業は、ldapServer や httpServer などのサーバの LDAP および httpstk 環境設定オブジェクトを使用するすべての OpenText eDirectory サーバに対して実行する必要があります。
  SuiteB モードをオンにする前に、eDirectory 環境内の LDAP クライアント、LDAP ブラウザ、および Web ブラウザのすべてが、TLS 1.2 証明書と EC 証明書をサポートしていることを確認してください。

ndsconfig を使用する HTTP スタックパラメータの設定

ndsconfig を使用する HTTP スタックパラメータは次のとおりです。

- **http.server.interfaces:** HTTP サーバがリスンするクリアテキストインタフェースが格納されます。これは、ndsconfig による新しいインスタンスの設定中に設定されます。
- **http.server.request-io-buffer-size:** HTTP サーバの入力および出力バッファサイズを指定します。デフォルト値は 8192 バイトです。
- **http.server.request_timeout-seconds:** 各 HTTP 要求のタイムアウトを指定します。デフォルト値は 300 秒です。

- **http.server.keep-timeout-seconds:** 各 HTTP 要求のキープアライブタイムアウトを指定します。デフォルト値は 15 秒です。
- **http.server.threads-per-processor:** CPU ごとに生成する HTTP スレッドの数を指定します。デフォルト値は 2 スレッドです。
- **http.server.session-exp-seconds:** HTTP セッションのタイムアウトを指定します。デフォルト値は 900 秒です。
- **http.server.trace-level:** DSTrace での HTTP スタックのデバッグ レベルを指定します。デフォルトレベルは 2 です。
- **http.server.clear-port:** HTTP サーバがリスンするクリアテキストポートを指定します。
- **http.server.tls-port:** HTTP サーバがリスンするセキュアポートを指定します。
- **http.server.auth-req-tls:** クリアテキストポート経由の要求をセキュアポートにリダイレクトするかどうかを指定します。
- **https.server.interfaces:** HTTP サーバがリスンするセキュアインタフェースが格納されます。これは、ndsconfig による新しいインスタンスの設定中に設定されます。
- **https.server.cached-cert-dn:** HTTP サーバがセキュア接続を処理するときに使用する必要がある証明書オブジェクトの DN が格納されます。

cn=monitor を使用した監視

OpenText eDirectory には、OpenText eDirectory サーバの現在の状態を監視するための LDAP 検索メソッドが用意されています。OpenText eDirectory は、OpenText eDirectory サブシステムとバックグラウンドプロセス (スレッドプール、接続テーブル、DClient、DS エージェント、バックグラウンドプロセス、LDAP サーバなど) に関する有用なパフォーマンスメトリックおよびサーバ状態の情報を、cn=monitor のベース DN を持つエントリとして記録します。これらの統計情報をサーバから取得するには、cn=monitor の検索ベースを使用して検索要求を発行します。取得したこの統計情報を使用して、OpenText eDirectory 環境を監視することができます。

重要 : cn=monitor は仮想オブジェクトであり、OpenText eDirectory ツリーに実際に存在するわけではありません。このメソッドは、LDAP インタフェースを通して OpenText eDirectory を監視する場合に使用できます。

OpenText eDirectory サブシステムは、監視フレームワーク内のデータ生成元として登録されています。OpenText eDirectory の登録済みデータ生成元を、表 8-1 に示します。このフレームワークは、登録済みのすべてのデータ生成元からリアルタイムのデータを収集し、このデータの要求側 (このデータのコンシューマ) と共有します。監視フレームワークは、検索要求に応じて動的にオブジェクトを収集して、cn=monitor のサブツリーに返します。各オブジェクトには、サーバの特定の側面についての情報が含まれています。一部のオブジェクトは他のオブジェクトのコンテナとして機能し、cn=monitor を最上位のオブジェクトとするオブジェクト階層を構築するために使用されます。監視フレームワーク、アクセス対象、およびその他のコントロールが提供する情報 (LDAP サーバ固有の情報や接続固有の情報など) には、LDAP クライアントを使用してアクセスできます。OpenText eDirectory では、この検索要求を、NCP サーバオブジェクトの NDSRightsToMonitor 属性に対する書き込み権を持つユーザに制限しています。

すべての登録済みデータ生成元からのデータには、ldapsearch または汎用の LDAP ブラウザを使用してアクセスできます。

すべての登録済みデータ生成元の監視データを表示するには、ldapsearch コマンドを使用します。

```
ldapsearch -h <SrvIP> -p <port> -D <user dn> -w <password> -s sub -b
cn=monitor
```

注 : OpenText eDirectory では、cn=monitor 検索でのデータフィルタリングをサポートしていません。再帰的に実行するようスケジュールされた一部のバックグラウンドプロセスについては、OpenText eDirectory は cn=monitor 検索の応答で、スケジュールされた回数分のプロセスを表示します。そのようなプロセスの一例は、SkulkerWorkerProc です。

監視統計情報の表示

ldapsearch は、すべての登録済みデータ生成元からのデータを、LDAP 形式で、cn=monitor をベースとして使用して返します。LDAP サーバは、LDAP オブジェクト形式のデータコンシューマとしても機能します。

表 8-1 には、データ生成元と、それに対応する、監視統計情報を格納しているパラメータがリストされます。OpenText eDirectory と一緒に他の製品が設定されている場合は、データ生成元がこれ以外にも存在する可能性があります。

***表 8-1** データ生成元と監視統計情報パラメータ*

データ生成元	監視統計情報パラメータ
エージェント	◆ バックグラウンドプロセス ◆ パーティション ◆ システムの状態
DHOST	次の DHOST プロセスおよび接続情報が監視されます。 ◆ インバウンド接続 ◆ スレッドプール情報 ◆ ThreadsSpawned ◆ ThreadsDied ◆ ThreadsIdle ◆ ThreadsWorkers ◆ ThreadPeakWorkers ◆ ThreadPoolReadyQueueItems ◆ ThreadPoolReadyQueueMaxWaitTime ◆ ThreadMinWaitTime ◆ ThreadMaxWaitTime

データ生成元	監視統計情報パラメータ
DClient	◆ アウトバウンドのコンテキスト ◆ アウトバウンド接続
LDAP	◆ バインディング ◆ エラー ◆ 着信操作 ◆ 発信操作 ◆ レプリケーション ◆ トラフィック量
レコードマネージャ	◆ キャッシュ失敗表示 ◆ キャッシュ障害 ◆ 現在のサイズ、ヒット数 ◆ ヒット表示 ◆ キャッシュに保存された項目 ◆ 最大サイズ ◆ OldVersionCachedCount ◆ OldVersionCachedSize ◆ DIb サイズ ◆ チェックポイントスレッド

cn=monitor の検索ベースを使用して検索要求を発行すると、表 8-2 に示されているように、監視フレームワークは検索要求の応答で、cn=monitor サブツリーのオブジェクトを動的に生成して返します。

表 8-2 *cn=monitor 検索で監視されるオブジェクト*

オブジェクト名	説明
cn=Monitor	データを監視するルートレベルオブジェクト。
cn=Agent,cn=Monitor	ディレクトリサービスエージェントに関する情報を提供します。
cn=BackGroundProcInterval,cn=Agent,cn=Monitor	バックグラウンドプロセスに関する情報を提供します。(特定のプロセス、または通常はすべてのバックグラウンドプロセス)
cn=ARC resolve timer thread,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	詳細参照コストのバックグラウンドプロセスに関する情報を提供します。
cn=BacklinkProc,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	バックリンカのバックグラウンドプロセスに関する情報を提供します。

オブジェクト名	説明
cn=CPU Usage monitor,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	CPU 使用率のバックグラウンドプロセスに関する情報を提供します。
cn=CheckBacklinks,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	バックリンカのチェックのバックグラウンドプロセスに関する情報を提供します。
cn=CheckExtRefProc,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	外部参照のチェックのバックグラウンドプロセスに関する情報を提供します。
cn=ExtRefRefreshProc,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	外部参照の更新のバックグラウンドプロセスに関する情報を提供します。
cn=Janitor,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	ジャニタのバックグラウンドプロセスに関する情報を提供します。
cn=RunLimberUp,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	スケジュールリンバのバックグラウンドプロセスに関する情報を提供します。
cn=Limber,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	リンバ (コネクティビティチェック) のバックグラウンドプロセスに関する情報を提供します。
cn=HiConvergenceHeartBeat,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	Skulker のスケジュールのバックグラウンドプロセスに関する情報を提供します。
cn=ObitProc,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	破損通知のバックグラウンドプロセスに関する情報を提供します。
cn=PartitionPurgeProcess,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	パーティションパージャのバックグラウンドプロセスに関する情報を提供します。
cn=Predicate Statistics Update,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	予測統計情報更新のバックグラウンドプロセスに関する情報を提供します。
cn=RNRAdvertise,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	アドバタイズサービスアドレスのバックグラウンドプロセスに関する情報を提供します。
cn=RefreshBinderyContext,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	バインダリ更新のバックグラウンドプロセスに関する情報を提供します。
cn=Repair Inactive Replicas,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	非アクティブレプリカの修復のバックグラウンドプロセスに関する情報を提供します。
cn=SchemaProc,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	スキーマ同期のバックグラウンドプロセスに関する情報を提供します。
cn=SkulkerProc,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	同期のバックグラウンドプロセスに関する情報を提供します。
cn=SkulkerWorkerProc,cn=BackGroundProcInterval,cn=Agent,cn=Monitor	同期のバックグラウンドプロセスに関する情報を提供します。
cn=Partition,cn=Agent,cn=Monitor	サーバ上のすべてのユーザパーティションに関する情報を提供します。同じ属性の値が複数ある場合、それは複数のパーティションがあることを意味します

オブジェクト名	説明
cn=Status,cn=Agent,cn=Monitor	サーバのステータスに関する情報を提供します。
cn=DHOST,cn=Monitor	DHOST サブシステムに関する情報を提供します。
cn=InBoundConnection,cn=DHOST,cn=Monitor	インバウンド接続テーブルの情報に関する情報を提供します。
cn=ThreadPool,cn=DHOST,cn=Monitor	DHOST スレッドプール統計に関する情報を提供します。
cn=Dclient,cn=Monitor	サーバ側 DClient に関する情報を提供します。
cn=OutBoundConnection,cn=Dclient,cn=Monitor	アウトバウンド接続テーブルの情報に関する情報を提供します。
cn=OutBoundContext,cn=Dclient,cn=Monitor	アウトバウンドコンテキストテーブルの情報に関する情報を提供します。
cn=LDAP,cn=Monitor	LDAP サーバの情報を提供します。
cn=LDAPStatistics,cn=LDAP,cn=Monitor	LDAP サーバの統計に関する情報を提供します。
cn=Bindings,cn=LDAPStatistics,cn=LDAP,cn=Monitor	LDAP サーバ上のバインディング統計に関する情報を提供します。
cn=Errors,cn=LDAPStatistics,cn=LDAP,cn=Monitor	LDAP 要求中に発生したエラーに関する情報を提供します。
cn=IncomingOperations,cn=LDAPStatistics,cn=LDAP,cn=Monitor	LDAP サーバ上の着信操作統計に関する情報を提供します。
cn=OutgoingOperations,cn=LDAPStatistics,cn=LDAP,cn=Monitor	LDAP サーバ上の発信操作統計に関する情報を提供します。
cn=Replications,cn=LDAPStatistics,cn=LDAP,cn=Monitor	LDAP サーバのレプリケーション統計に関する情報を提供します。
cn=TrafficVolume,cn=LDAPStatistics,cn=LDAP,cn=Monitor	LDAP サーバのトラフィック量統計に関する情報を提供します。
cn=RecordManager,cn=Monitor	FLAIM データベースに関する情報を提供します。
cn=Size,cn=RecordManager,cn=Monitor	FLAIM データベースのサイズに関する情報を提供します。
cn=CheckPointThreadData,cn=RecordManager,cn=Monitor	チェックポイントスレッドに関する情報を提供します。
cn=CacheStatistics,cn=RecordManager,cn=Monitor	FLAIM データベースのキャッシュ統計に関する情報を提供します。
cn=CacheFaultLooks,cn=CacheStatistics,cn=RecordManager,cn=Monitor	キャッシュ失敗表示の情報を提供します。
cn=CacheFaults,cn=CacheStatistics,cn=RecordManager,cn=Monitor	キャッシュ失敗の情報を提供します。

オブジェクト名	説明
cn=HitLooks,cn=CacheStatistics,cn=RecordManager,cn=Monitor	キャッシュヒット表示の情報を提供します。
cn=Hits,cn=CacheStatistics,cn=RecordManager,cn=Monitor	キャッシュヒットの情報を提供します。
cn=ItemsCached,cn=CacheStatistics,cn=RecordManager,cn=Monitor	キャッシュに保存された項目数の情報を提供します。
cn=OldVersionCachedCount,cn=CacheStatistics,cn=RecordManager,cn=Monitor	古いバージョンのキャッシュに保存された項目数の情報を提供します。
cn=MaximumSize,cn=CacheStatistics,cn=RecordManager,cn=Monitor	最大キャッシュサイズの情報を提供します。
cn=CurrentSize,cn=CacheStatistics,cn=RecordManager,cn=Monitor	現在のキャッシュサイズの情報を提供します。
cn=OldVersionCachedSize,cn=CacheStatistics,cn=RecordManager,cn=Monitor	古いバージョンのキャッシュサイズの情報を提供します。

各オブジェクトには、サーバの特定の側面 (接続やスレッドなど) についての情報が格納されます。監視統計情報が格納される属性を、表 8-3 に示します。

***表 8-3** 統計情報を監視するための属性*

属性	説明
BackgroundProcScheduled	バックグラウンドプロセスが次にスケジュールされている時刻。複数の値は、バックグラウンドプロセスが複数回スケジュールされていることを意味します。
BackgroundProcStartTime	次のバックグラウンドプロセスの開始時刻。複数の値は、バックグラウンドプロセスが複数回実行されることを意味します。
PerishableData	他のサーバと同期されていない状態のデータの量 (秒単位で示されます)。
OBIT_NEWRDN_PURGEABLE	PURGEABLE 状態の NEWRDN 破損通知の数。
OBIT_NEWRDN_OK_TO_PURGE	パージ可能状態の NEWRDN 破損通知の数。
OBIT_NEWRDN_NOTIFIED	NOTIFIED 状態の NEWRDN 破損通知の数。
OBIT_NEWRDN_ISSUED	ISSUED 状態の NEWRDN 破損通知の数。
OBIT_MOVED_PURGEABLE	PURGEABLE 状態に移された破損通知の数。
OBIT_MOVED_OK_TO_PURGE	パージ可能状態に移された破損通知の数。
OBIT_MOVED_NOTIFIED	NOTIFIED 状態に移された破損通知の数。
OBIT_MOVED_ISSUED	ISSUED 状態に移された破損通知の数。
OBIT_DEAD_PURGEABLE	PURGEABLE 状態で停止している破損通知の数。

属性	説明
OBIT_DEAD_OK_TO_PURGE	パージ可能状態で停止している破損通知の数。
OBIT_DEAD_NOTIFIED	NOTIFIED 状態で停止している破損通知の数。
OBIT_DEAD_ISSUED	ISSUED 状態で停止している破損通知の数。
OBIT_COUNT_FROM_DATABASE_INDEX	破損通知の合計数。
MaxRingDelta	レプリカリングの任意の 2 つのサーバ間で同期されていない状態のデータの最大量 (秒単位で示されます)。
ChangeCacheCount	パーティション上の現在の変更キャッシュカウント。
eDirectoryUpTime	サーバ起動後の経過時間 (秒数)。
eDirectorySystemCurrTime	サーバの現在のシステム時刻。
eDirectoryAgentVersion	カレントディレクトリサーバエージェントのバージョン。
MaxInBoundConnection	最大インバウンド接続。
InBoundConnectionCount	現在のインバウンド接続数。
ThreadsWorkers	スレッドプール内のワーカスレッド数。
ThreadsSpawned	生成されたスレッド数。
ThreadsIdle	アイドルスレッド数。
ThreadsDied	停止スレッド数。
ThreadWaitingQueuePeakItems	待機キュー内のスレッドの最大数。
ThreadWaitingQueueItems	待機キュー内の現在のスレッド数。
ThreadPoolReadyQueueMaxWaitTime	ReadyQueue 内のスレッドの最大待機時間。
ThreadPoolReadyQueueItems	ReadyQueue 内の現在のスレッド数。
ThreadPeakWorkers	プールワーカの最大数。
ThreadMinWaitTime	スケジュールされるまでのスレッドの最小待機時間。
ThreadMaxWaitTime	スケジュールされるまでのスレッドの最大待機時間。
TotalOpenOutBoundConnection	現在のオープンアウトバウンド接続数。
RefusedOutBoundConnection	拒否されたアウトバウンド接続数。
MaxOutBoundConnection	最大アウトバウンド接続数。
TotalOutBoundContextCount	最大アウトバウンドコンテキスト数。
ActiveOutBoundContextCount	現在のアウトバウンドコンテキスト数。
unAuthBinds	認証を省略した匿名バインド要求を受け取った回数。

属性	説明
strongAuthBinds	バインド要求のうち、強度の高い認証手続きである SASL および X.500 の認証に成功したものの回数。外部認証手続きによるものも数に含みます。
simpleAuthBinds	バインド要求のうち、簡易認証手続きにより認証に成功したものの回数。簡易認証手続きとは、パスワードを暗号化して、またはクリアテキストのまま送ることにより行うものです。
bindSecurityErrors	バインド要求のうち、認証手続きが適切でない、あるいは資格情報が無効であるために拒否したものの回数。
エラー	エラーが発生して返された要求の数。Errors 属性には、次のものは含まれません。 ◆ セキュリティ関連のエラー ◆ 参照関連のエラー ◆ 部分的に返された要求
securityErrors	不適切な認証や無効な資格情報など、セキュリティ要件を満たしていない要求の数。
wholeSubtreeSearchOps	受け取ったサブツリー全体の検索要求の数。
searchOps	受け取った検索要求 (ベースオブジェクト検索、1 レベル検索、サブツリー全体の検索) の数。
removeEntryOps	受け取ったエントリ削除要求の数。
readOps	受け取った読み出し要求の数。
oneLevelSearchOps	受け取った 1 レベル検索要求の数。
modifyRDNOps	受け取った RDN(相対識別名) 変更要求の数。
modifyEntryOps	受け取ったエントリ変更要求の数。
listOps	受け取ったリスト要求の数。
inOps	クライアントから受け取った要求の数。
extendedOps	拡張処理の回数。
compareOps	受け取った比較要求の数。
addEntryOps	受け取ったエントリ追加要求の数。
abandonOps	LDAP が破棄した要求の数。
referralsReturned	処理要求に応じて返した参照の個数。
chainings	この OpenText eDirectory サーバから他の OpenText eDirectory サーバに転送された処理の数。
repsUpdatesIn	現在のインバウンドレプリケーション更新要求数。
repsUpdateOut	現在のアウトバウンドレプリケーション更新要求数。

属性	説明
outBytes	インタフェース上の送信トラフィック (バイト単位)。クライアントや OpenText eDirectory サーバへの応答と、他の OpenText eDirectory サーバへの要求がこれに含まれます。
inBytes	インタフェース上の受信トラフィック (バイト単位)。クライアントからの要求と、他の OpenText eDirectory サーバからの応答がこれに含まれます。
合計	FLAIM キャッシュに含まれる項目の合計数。
EntryCache	エントリキャッシュに含まれる項目の合計数。
BlockCache	ブロックキャッシュに含まれる項目の合計数。
TotalSize	FLAIM キャッシュに含まれる項目の合計サイズ。
EntryCacheSize	エントリキャッシュに含まれる項目の合計サイズ。
BlockCacheSize	ブロックキャッシュに含まれる項目の合計サイズ。
CheckPointThreadWritingDataBlocks	0 は、そのチェックポイントがダーティブロックに書き込んでいないことを意味します。1 は、そのチェックポイントがダーティブロックに書き込んでいることを意味します。
CheckPointThreadStartTime	チェックポイントスレッドの開始時刻。この値を確認する必要があるのは、チェックポイントスレッドが実行されている場合のみです。
CheckPointThreadLogBlocksWritten	書き込まれたログブロックの数。
CheckPointThreadIsRunning	0 は、チェックポイントが実行されていないことを意味します。1 は、チェックポイントスレッドが実行されていることを意味します。
CheckPointThreadIsForced	チェックポイントが強制されているかどうかを示します。
CheckPointThreadForceStartTime	チェックポイントの強制開始時刻。この値を確認する必要があるのは、チェックポイントが強制的に開始される場合のみです。
CheckPointThreadDirtyCacheBlocks	ダーティキャッシュブロックの数。
CheckPointThreadDataBlocksWritten	書き込まれたダーティブロックの数。
CheckPointThreadBlockSize	現在のブロックサイズ。
TotalDIBSize	FLAIM データベースの合計サイズ。
DIBStreamFileSize	ストリームファイルの合計サイズ。
DIBRollBackFileSize	ロールバックファイルの合計サイズ。
DIBRflmFileSize	ロールフォワードログファイルの合計サイズ。
DIBFileSize	DIB ファイルの合計サイズ。

属性	説明
CurrentTransactionID	トランザクション ID は、データベースで発生した変更を追跡するために使用されます。各 write 処理は、transactionID と関連付けられたトランザクション内で実行されます。許可されるトランザクション ID の最大値は 0xFFFFE000 [4294959104] です。これに達すると、新しいトランザクションは許可されません。

以下に、LDAP 検索の出力例を示します。

```
# LDAPv3
# base <cn=monitor> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
# BackGroundProcInterval, Agent, Monitor
dn: cn=BackGroundProcInterval,cn=Agent,cn=Monitor
slowSyncInterval: 1800
fastSyncInterval: 5
ServerStateUpThreshold: 1800
JanitorInterval: 120
HeartBeatSkulkInterval: 3600
FlatCleaningInterval: 43200
DRLInterval: 60
BacklinkInterval: 46800
objectclass: Top
objectclass: extensibleObject
# .GOOD-ONE., Partition, Agent, Monitor
dn: cn=.GOOD-ONE.,cn=Partition,cn=Agent,cn=Monitor
ChangeCacheCount: 0
objectclass: Top
objectclass: extensibleObject
# InBoundConnection, DHOST, Monitor
dn: cn=InBoundConnection,cn=DHOST,cn=Monitor
MaxInBoundConnection: 256
InBoundConnectionCount: 20
objectclass: Top
objectclass: extensibleObject
# ThreadPool, DHOST, Monitor
dn: cn=ThreadPool,cn=DHOST,cn=Monitor
ThreadsWorkers: 37
Monitoring
ThreadsSpawned: 3572
ThreadsIdle: 7
ThreadsDied: 3535
ThreadWaitingQueuePeakItems: 24
ThreadWaitingQueueItems: 20
ThreadPoolReadyQueueMaxWaitTime: 574529
ThreadPoolReadyQueueItems: 0
ThreadPeakWorkers: 90
ThreadMinWaitTime: 2
ThreadMaxWaitTime: 16394616
objectclass: Top
objectclass: extensibleObject
# OutBoundConnection, Dclient, Monitor
```

```
dn: cn=OutBoundConnection,cn=Dclient,cn=Monitor
TotalOpenOutBoundConnection: 17
RefusedOutBoundConnection: 0
MaxOutBoundConnection: 4294967295
objectclass: Top
objectclass: extensibleObject
# OutBoundContext, Dclient, Monitor
dn: cn=OutBoundContext,cn=Dclient,cn=Monitor
TotalOutBoundContextCount: 256
objectclass: Top
objectclass: extensibleObject
# Bindings, LDAPStatistics, LDAP, Monitor
dn: cn=Bindings,cn=LDAPStatistics,cn=LDAP,cn=Monitor
unAuthBinds: 6908
strongAuthBinds: 0
simpleAuthBinds: 4433475
bindSecurityErrors: 0
objectclass: Top
objectclass: extensibleObject
# Errors, LDAPStatistics, LDAP, Monitor
dn: cn=Errors,cn=LDAPStatistics,cn=LDAP,cn=Monitor
securityErrors: 0
errors: 1
objectclass: Top
objectclass: extensibleObject
# IncomingOperations, LDAPStatistics, LDAP, Monitor
dn: cn=IncomingOperations,cn=LDAPStatistics,cn=LDAP,cn=Monitor
wholeSubtreeSearchOps: 4426462
searchOps: 4426462
removeEntryOps: 0
readOps: 0
oneLevelSearchOps: 0
modifyRDNOps: 0
modifyEntryOps: 4
listOps: 0
inOps: 8901739
extendedOps: 0
compareOps: 0
addEntryOps: 5
abandonOps: 0
objectclass: Top
objectclass: extensibleObject
# OutgoingOperations, LDAPStatistics, LDAP, Monitor
dn: cn=OutgoingOperations,cn=LDAPStatistics,cn=LDAP,cn=Monitor
chainings: 0
referralsReturned: 0
objectclass: Top
objectclass: extensibleObject
# Replications, LDAPStatistics, LDAP, Monitor
dn: cn=Replications,cn=LDAPStatistics,cn=LDAP,cn=Monitor
repUpdatesIn: 0
repUpdatesOut: 0
objectclass: Top
objectclass: extensibleObject
# TrafficVolume, LDAPStatistics, LDAP, Monitor
```

```
dn: cn=TrafficVolume,cn=LDAPStatistics,cn=LDAP,cn=Monitor
outBytes: 326809576
inBytes: 380249498
objectclass: Top
objectclass: extensibleObject
Monitoring
# CacheFaultLooks, RecordManager, Monitor
dn: cn=CacheFaultLooks,cn=RecordManager,cn=Monitor
TotalSize: 2699
EntryCacheSize: 2539
BlockCacheSize: 160
objectclass: Top
objectclass: extensibleObject
# CacheFaults, RecordManager, Monitor
dn: cn=CacheFaults,cn=RecordManager,cn=Monitor
TotalSize: 1948
EntryCacheSize: 1788
BlockCacheSize: 160
objectclass: Top
objectclass: extensibleObject
# CurrentSize, RecordManager, Monitor
dn: cn=CurrentSize,cn=RecordManager,cn=Monitor
TotalSize: 4849664
EntryCacheSize: 3866624
BlockCacheSize: 983040
objectclass: Top
objectclass: extensibleObject
# HitLooks, RecordManager, Monitor
dn: cn=HitLooks,cn=RecordManager,cn=Monitor
TotalSize: 656418775
EntryCacheSize: 489811630
BlockCacheSize: 166607145
objectclass: Top
objectclass: extensibleObject
# Hits, RecordManager, Monitor
dn: cn=Hits,cn=RecordManager,cn=Monitor
TotalSize: 449815580
EntryCacheSize: 283226835
BlockCacheSize: 166588745
objectclass: Top
objectclass: extensibleObject
# ItemsCached, RecordManager, Monitor
dn: cn=ItemsCached,cn=RecordManager,cn=Monitor
TotalSize: 1865
EntryCacheSize: 1691
BlockCacheSize: 174
objectclass: Top
objectclass: extensibleObject
# MaximumSize, RecordManager, Monitor
dn: cn=MaximumSize,cn=RecordManager,cn=Monitor
TotalSize: 200015872
EntryCacheSize: 100007972
BlockCacheSize: 100007900
objectclass: Top
objectclass: extensibleObject
```

```
# OldVersionCachedCount, RecordManager, Monitor
dn: cn=OldVersionCachedCount,cn=RecordManager,cn=Monitor
TotalSize: 7
EntryCacheSize: 3
BlockCacheSize: 4
objectclass: Top
objectclass: extensibleObject
# OldVersionCachedSize, RecordManager, Monitor
dn: cn=OldVersionCachedSize,cn=RecordManager,cn=Monitor
Monitoring
TotalSize: 21376
EntryCacheSize: 4448
BlockCacheSize: 16928
objectclass: Top
objectclass: extensibleObject
dn: cn=Size,cn=RecordManager,cn=Monitor
CurrentTransactionID: 26552
TotalDIBSize: 1533393 Bytes
DIBStreamFileSize: 781 Bytes
DIBRflmFileSize: 512708 Bytes
DIBRollBackFileSize: 167936 Bytes
DIBFileSize: 851968 Bytes
objectclass: Top
objectclass: extensibleObject
# search result
search: 2
result: 0 Success
# numResponses: 20
# numEntries: 19
```

DSTrace の使用

Linux 環境で DSTrace ユーティリティを使用するには、サーバプロンプトから次のコマンドを実行します。

```
/opt/novell/eDirectory/bin/ndstrace
```

ndstrace コマンドの完全な構文は次のとおりです。

```
ndstrace [-l|-u|-c "command1;....."|--version] [-h
<local_interface:port>] [--config-file <configuration_file_path>] [thrd
<thread ID>] [svty <severity_level>] [conn <connection_ID>]
```

DSTrace ユーティリティは、次の 3 つの主要部分で構成されています。

- 293 ページの 「基本機能」
- 294 ページの 「デバッギングメッセージ」
- 296 ページの 「バックグラウンド処理」

基本機能

DSTrace の基本機能は次のとおりです。

- OpenText eDirectory の内部動作および Linux のデバッグメッセージを表示します。
- 一部の同期処理を開始します。

DSTrace ユーティリティは、UI モードまたはコマンドラインモードのいずれかで使用できます。デフォルトでは、DSTrace は UI モードで実行します。UI モードで DSTrace ユーティリティを開始するには、サーバプロンプトで次のコマンドを実行します。

```
/opt/novell/eDirectory/bin/ndstrace
```

コマンドラインモードで DSTrace ユーティリティを開始するには、サーバプロンプトで次のコマンドを実行します。

```
/opt/novell/eDirectory/bin/ndstrace -l
```

DSTrace の基本機能を開始するには、次の構文を使用してサーバプロンプトでコマンドを入力します。

```
ndstrace command_option
```

次の表では、入力可能なコマンドオプションのリストを示します。

オプション	説明
オン	基本トレースメッセージを含む OpenText eDirectory トレース画面を起動します。
オフ	トレース画面を無効にします。
ALL	OpenText eDirectory トレース画面を起動し、すべてのトレースメッセージを表示します。
AGENT	ON、BACKLINK、DSAGENT、JANITOR、RESNAME、および VCLIENT フラグと同等のトレースメッセージを含む OpenText eDirectory トレース画面を起動します。
DEBUG	デバッグに通常使用する定義済みのトレースメッセージのセットを有効にします。設定されるフラグは、ON、BACKLINK、ERRORS、EMU、FRAGGER、INIT、INSPECTOR、JANITOR、LIMBER、MISC、PART、RECMAN、REPAIR、SCHEMA、SKULKER、STREAMS、および VCLIENT です。
NODEBUG	トレース画面は使用可能なままで、以前に設定したデバッギングメッセージはすべて無効にします。このオプションでは、メッセージも ON のコマンドオプションが設定された状態のままになります。

デバッギングメッセージ

DSTrace 画面が使用可能な場合、デフォルトのフィルタの設定に基づいて情報が表示されます。デフォルトで表示される情報の内容を変更するには、デバッギングメッセージフラグを使用してフィルタを操作します。デバッギングメッセージにより、OpenText eDirectory のステータスを確認し、問題が発生していないかどうかを検証できます。

各 OpenText eDirectory 処理には、デバッギングメッセージのセットが含まれています。個々の処理中にそのデバッギングメッセージを表示するには、プラス記号 (+)、および該当する処理名またはオプションを使用します。処理を表示しない場合は、マイナス記号 (-)、および該当する処理名またはオプションを使用します。次に例を示します。

メッセージ	説明
set ndstrace = +SYNC	同期メッセージを表示します。
set ndstrace = -SYNC	同期メッセージを非表示にします。
set ndstrace = +SCHEMA	スキーマメッセージを表示します。

また、ブール演算子の & (AND) および | (OR) を使用して、デバッギングメッセージのフラグを結合することもできます。サーバコンソールでデバッギングメッセージを制御する構文は、次のとおりです。

```
set ndstrace = <trace_flag> [parameter]
```

次の表では、デバッギングメッセージ用のトレースフラグについて説明します。各トレースフラグは略語で入力できます。

トレースフラグ	説明
ABUF	OpenText eDirectory 要求との連携、または OpenText eDirectory 要求への応答として受信されたデータを含む、インバウンドおよびアウトバウンドパケットバッファに関するメッセージと情報です。
ALOC	メモリ割り当ての詳細について示すメッセージです。
AREQ	他のサーバまたはクライアントからのインバウンド要求に関するメッセージです。
AUTH	認証に関するメッセージとエラーレポートです。
BASE	最小限のデバッギングレベルでのデバッグエラーメッセージ。
BLNK	バックリンクとインバウンドの破損通知メッセージおよびエラーレポートです。
CBUF	アウトバウンド DS クライアント要求に関するメッセージです。
CHNG	キャッシュ変更メッセージです。
COLL	以前に更新内容を受信したときのオブジェクトの更新情報に関するステータスおよびエラーレポートです。

トレースフラグ	説明
CONN	ローカルサーバが接続を試みている相手のサーバ、およびローカルサーバが接続できない原因となっている可能性のあるエラーとタイムアウトについての情報を示すメッセージです。
DNS	OpenText eDirectory 統合 DNS サーバプロセスに関するメッセージです。
DRLK	分散リファレンスリンクメッセージです。
DVRS	OpenText eDirectory が機能している可能性のある DirXML® ドライバ固有のエリアを示すメッセージです。
DXML	DirXML イベントの詳細について示すメッセージです。
FRAG	OpenText eDirectory メッセージを NCP サイズのメッセージに分解する、NCP™ Fragger からのメッセージです。
IN	インバウンドの要求およびプロセスに関するメッセージです。
INIT	OpenText eDirectory の初期化に関するメッセージです。
INSP	ソースサーバのローカルデータベース内のオブジェクトの整合性に関するメッセージです。このフラグを使用すると、ソースサーバのディスクストレージシステム、メモリ、プロセッサの要求量が増加します。オブジェクトが破損しない限り、このフラグは有効に設定しないでください。
JNTR	janitor、レプリカの同期、フラットクリーナなどのバックグラウンド処理に関するメッセージです。
LDAP	LDAP サーバに関するメッセージです。
LMBR	limber 処理に関するメッセージです。
LOCK	ソースサーバのローカルデータベースロックの使用および操作に関するメッセージです。
LOST	消失エントリに関するメッセージです。
MISC	OpenText eDirectory 内の異なるソースからのメッセージです。
MOVE	パーティションの移動操作、またはサブツリーの移動操作からのメッセージです。
NCPE	NCP レベルの要求を受信したサーバを示すメッセージです。
NMON	OpenText iMonitor に関するメッセージです。
OBIT	破損通知処理からのメッセージです。
PART	バックグラウンド処理および要求処理からのパーティション操作に関するメッセージです。
PURG	パージ処理に関するメッセージです。
RECM	ソースサーバのデータベースの操作に関するメッセージです。
RSLV	名前解決要求の処理に関するメッセージです。

トレースフラグ	説明
SADV	SLP (Service Location Protocol) のツリー名とパーティションの登録に関するメッセージです。
SCMA	スキーマの同期処理に関するメッセージです。
SCMD	スキーマ関連の操作の詳細について示すメッセージです。インバウンド同期とアウトバウンド同期の両方についての詳細を示します。
SKLK	レプリカの同期処理に関するメッセージです。
SPKT	OpenText eDirectory NCP サーバレベルの情報に関するメッセージです。
STRM	ストリーム構文の属性の処理に関するメッセージです。
SYDL	レプリケーション処理時の詳細について示すメッセージです。
SYNC	インバウンド同期トラフィック (サーバ側で受信される内容) についてのメッセージです。
TAGS	トレースオプションを識別するタグ文字列が表示されます。このトレースオプションでは、トレース処理で表示される各行のイベントが生成されます。
THRD	バックグラウンド処理 (スレッド) の開始時と終了時を示すメッセージです。
TIME	同期処理時に使用される遷移ベクトルに関するメッセージです。
TVEC	Synchronize Up To、レプリカ、および遷移ベクトルなどの属性に関するメッセージです。
VCLN	他のサーバへの接続の確立または切断に関するメッセージです。

DSTrace でデバッギングメッセージを使用していると、特に便利なトレースフラグがあることが分かります。OpenText サポートで多く使用されている DSTrace 設定には、次のようなショートカットがあります。

```
set ndstrace = A81164B91
```

この設定を使用すると、複数のデバッギングメッセージを 1 つのグループとして使用できます。

バックグラウンド処理

OpenText eDirectory のステータスを確認できるデバッギングメッセージの他に、OpenText eDirectory バックグラウンド処理を強制的に実行するコマンドのセットも用意されています。バックグラウンド処理を強制的に実行するには、コマンドの先頭にアスタリスク (*) を付けます。次に例を示します。

```
set ndstrace = *H
```

また、いくつかのバックグラウンド処理のステータス、タイミング、および制御を変更することもできます。これらの値を変更するには、コマンドの先頭に感嘆符 (!) を付けて新しいパラメータまたは値を入力します。次に例を示します。

`set ndstrace = !H 15 (parameter_value_in_minutes)`

OpenText eDirectory バックグラウンド処理を制御する各ステートメントの構文を次に示します。

`set ndstrace = <trace_flag> [parameter]`

次の表では、バックグラウンド処理のトレースフラグ、必要なパラメータ、およびトレースフラグが表示される処理のリストを示します。

トレースフラグ	パラメータ	説明
*A	なし	ソースサーバのアドレスキャッシュをリセットします。
*AD	なし	ソースサーバのアドレスキャッシュを無効にします。
*AE	なし	ソースサーバのアドレスキャッシュを有効にします。
*B	なし	ソースサーバ上で 1 秒後にバックリンク処理の実行を開始するようにスケジュールします。
!B	時刻	バックリンク処理の実行間隔を分単位で設定します。 デフォルト =1500 分 (25 時間)。範囲 =2 ～ 10080 分 (168 時間)
*CT	なし	ソースサーバのアウトバウンド接続テーブルと、テーブルの現在の統計情報を表示します。これらの統計情報には、他のサーバやクライアントからソースサーバへのインバウンド接続に関する情報は含まれていません。
*CTD	なし	コンマ区切りの形式で、ソースサーバのアウトバウンド接続テーブルと、テーブルの現在の統計情報を表示します。これらの統計情報には、他のサーバやクライアントからソースサーバへのインバウンド接続に関する情報は含まれていません。
*D	レプリカ rootEntry ID	指定したローカルエントリ ID をソースサーバの［すべてのオブジェクトを送信］リストから削除します。エントリ ID では、サーバのローカルデータベースで固有のパーティションルートオブジェクトを指定する必要があります。通常、このコマンドは、サーバのアクセス不能が原因で Send All Updates 処理が何度試みられても失敗する場合にのみ使用します。
!D	時刻	インバウンド同期およびアウトバウンド同期の間隔を分単位で指定された値に設定します。 デフォルト =24 分。範囲 =2 ～ 10080 分 (168 時間)
!DI	時刻	インバウンド同期の間隔を分単位で指定された値に設定します。 デフォルト =24 分。範囲 =2 ～ 10080 分 (168 時間)

トレースフラグ	パラメータ	説明
!DO	時刻	アウトバウンド同期の間隔を分単位で指定された値に設定します。 デフォルト =24 分。範囲 =2 〜 10080 分 (168 時間)
*E	なし	ソースサーバのエントリキャッシュを再初期化します。
!E	なし	インバウンド同期およびアウトバウンド同期処理の実行を開始するようにスケジュールします。
!EI	なし	インバウンド同期処理の実行を開始するようにスケジュールします。
!EO	なし	アウトバウンド同期処理の実行を開始するようにスケジュールします。
*F	なし	janitor 処理の一部として、フラットクリーナ処理の実行がソースサーバ上で 5 秒後に開始されるようにスケジュールします。
!F	時刻	フラットクリーナ処理の実行間隔を分単位で設定します。 デフォルト =240 分 (4 時間)。範囲 =2 〜 10080 分 (168 時間)
*FL	1-10	DSTrace が使用するローリングログファイルの数を設定します。このパラメータを 1 より大きい値に設定した場合、ソースサーバの ndstrace.log ファイルが設定されている最大ファイルサイズに達すると、DSTrace はログファイルの名前を ndstrace1.log に変更して、新しい ndstrace.log ファイルを作成します。このファイルが最大ファイルサイズに達すると、先ほどの ndstrace1.log ファイルが ndstrace2.log に名前を変更され、それより新しい ndstrace.log ファイルが ndstrace1.log に名前を変更されます。 この処理は、DSTrace がこのオプションによって設定されたローリングログファイルの最大数に達するまで継続します。指定された制限に達すると、一番古いログファイルが削除されて、指定された最大数のローリングログファイルのみが保持されます。 最大 10 個のローリングログファイルを設定できます。デフォルトでは、DSTrace はローリングログファイルを少なくとも 1 個使用する必要があります。このパラメータを 0 に設定すると、DSTrace はパラメータ値として 1 を使用します。
*G	レプリカ rootEntry ID	指定したルートパーティション ID の変更キャッシュを再構築します。

トレースフラグ	パラメータ	説明
*H	なし	ソースサーバ上で直ちににレプリカ同期処理の実行を開始するようにスケジュールします。
!H	時刻	Heatbeat 同期処理の実行間隔を分単位で設定します。 デフォルト =30 分。範囲 =2 ～ 1440 分 (24 時間)
*HR	なし	メモリ内で最後に送信されたベクトルを消去します。
*I	レプリカ rootEntry ID	指定したローカルエントリ ID をソースサーバの［すべてのオブジェクトを送信］リストに追加します。エントリ ID では、サーバのローカルデータベースで固有のパーティションルートオブジェクトを指定する必要があります。レプリカの同期処理では、［すべてのオブジェクトを送信］リストがチェックされます。パーティションのルートオブジェクトのエントリ ID がリスト内に存在する場合、Synchronized Up To 属性の値に関係なく、OpenText eDirectory によってパーティション内のすべてのオブジェクトと属性が同期されます。
!I	時刻	Heatbeat 同期処理の実行間隔を分単位で設定します。 デフォルト =30 分。範囲 =2 ～ 1440 分 (24 時間)
*J	なし	レプリカの同期処理の一部として、ソースサーバ上でパージ処理の実行を開始するようにスケジュールします。
!J	時刻	janitor 処理の実行間隔を分単位で設定します。 デフォルト =2 分。範囲 =1 ～ 10080 分 (168 時間)
*L	なし	ソースサーバ上で 5 秒後に limber 処理の実行を開始するようにスケジュールします。
*M	［Bytes］	ソースサーバの ndstrace.log ファイルで使用する最大ファイルサイズを変更します。このコマンドは、デバッグファイルの状態に関係なく使用できます。bytes の値は 10000 バイトと 100MB の間で 10 進の値を指定する必要があります。この範囲外の値が指定された場合、変更は行われません。
!M	なし	OpenText eDirectory で使用されるメモリの最大量をレポートします。
!N	0 | 1	名前の形式を設定します。 0=16 進数のみ。1=full dot 形式
*P	なし	調整可能なパラメータとそのデフォルトの設定を表示します。

トレースフラグ	パラメータ	説明
*R	なし	ndstrace.log ファイルのサイズをゼロバイトに再設定します。このコマンドは、SET パラメータの NDS Trace File Length Set to Zero と同じ働きをします。
*S	なし	サーバ上のレプリカを同期する必要があるかどうかをチェックするスカルク処理をスケジュールします。
!SI	時刻	インバウンドスキーマ同期処理の実行間隔を分単位で設定します。 デフォルト =24 分。範囲 =2 ～ 10080 分 (168 時間)
!SO	時刻	アウトバウンドスキーマ同期処理の実行間隔を分単位で設定します。 デフォルト =24 分。範囲 =2 ～ 10080 分 (168 時間)
!SIO	時刻	時間を分単位で指定し、その間のインバウンドスキーマ同期処理を無効にします。 デフォルト =24 分。範囲 =2 ～ 10080 分 (168 時間)
!SO0	時刻	時間を分単位で指定し、その間のインバウンドスキーマ同期処理を無効にします。 デフォルト =24 分。範囲 =2 ～ 10080 分 (168 時間)
*SS	なし	強制的に即時スキーマの同期を実行します。
*SSA	なし	スキーマの同期処理の実行を即時に開始するようにスケジュールします。過去 24 時間以内に同期が行われていた場合でも、すべてのターゲットサーバでスキーマの同期が強制的に実行されます。
*SSD	なし	ソースサーバの［ターゲットスキーマ同期］リストをリセットします。このリストでは、スキーマの同期処理の実行中にソースサーバと同期する必要のあるサーバが識別されます。レプリカを保持していないサーバは、サーバオブジェクトとレプリカを保持しているサーバのターゲットリストに包含されるように要求を送信します。
*［SSL］	なし	ターゲットサーバのスキーマ同期リストを印刷します。
*ST	なし	ソースサーバ上のバックグラウンド処理のステータス情報を表示します。
*STX	なし	ソースサーバ上のバックリンク処理 (外部参照) のステータス情報を表示します。
*STS	なし	ソースサーバ上のスキーマ同期処理のステータス情報を表示します。
*STO	なし	ソースサーバ上のバックリンク処理 (破損通知) のステータス情報を表示します。

トレースフラグ	パラメータ	説明
*STL	なし	ソースサーバ上の limber 処理のステータス情報を表示します。
!T	時刻	サーバの稼動状態のチェックの実行間隔を分単位で設定します。 デフォルト =30 分。範囲 =1 ～ 720 分 (12 時間)
*U	サーバのオプションの ID	コマンドにエントリ ID が含まれていない場合は、以前に「**down**」から「**up**」にラベルが付加された任意のサーバのステータスを変更します。コマンドにローカルエントリ ID が含まれている場合は、指定されたサーバのステータスを「**down**」から「**up**」に変更します。エントリ ID は、ソースサーバのデータベースで固有であり、サーバを表すオブジェクトを参照する必要があります。
!V	リスト	制限のある OpenText eDirectory バージョンのリストを表示します。バージョンが表示されない場合は、制限がないことを示します。各バージョンはコンマで区切られます。
*Z	なし	現在、スケジュールされているタスクを表示します。

DSTrace メッセージ

スレッド ID、接続 ID、およびメッセージの重大度に基づいて、トレースメッセージをフィルタすることができます。

メッセージにフィルタを指定すると、フィルタに一致するメッセージだけが画面に表示されます。FILE が ON に設定されている場合、タグが有効になっている他のメッセージはすべて ndstrace.log に記録されます。

一度に適用できるのは 1 つのフィルタだけです。フィルタは、DSTrace のセッションごとに指定する必要があります。

デフォルトでは、重大度レベルは INFO に設定されます。これは、重大度レベルが INFO 以上のメッセージはすべて表示されることを意味します。重大度レベルは、svty タグを有効にすると表示できます。

OpenText iMonitor を使用しても、トレースメッセージをフィルタすることができます。詳細については、「305 ページの 「OpenText iMonitor のメッセージフィルタリング」」を参照してください。

Linux

次の手順を完了してトレースメッセージをフィルタします。

1 次のコマンドでフィルタを有効にします。

```
ndstrace tag filter_value
```

フィルタを無効にするには、次のコマンドを入力します。

```
ndstrace tag
```

フィルタを有効にする場合の例 :

- スレッド ID が 35 の場合にフィルタを有効にするには、次のように入力します。

  ```
  ndstrace thrd 35
  ```

- 重大度レベルが致命的エラーの場合にフィルタを有効にするには、次のように入力します。

  ```
  ndstrace svty fatal
  ```

  重大度レベルとして、FATAL、WARN、ERR、INFO、および DEBUG を指定できます。

- 接続 ID が 21 の場合にフィルタを有効にするには、次のように入力します。

  ```
  ndstrace conn 21
  ```

フィルタを無効にする場合の例 :

- スレッド ID に基づいてフィルタを無効にするには、次のように入力します。

  ```
  ndstrace thrd
  ```

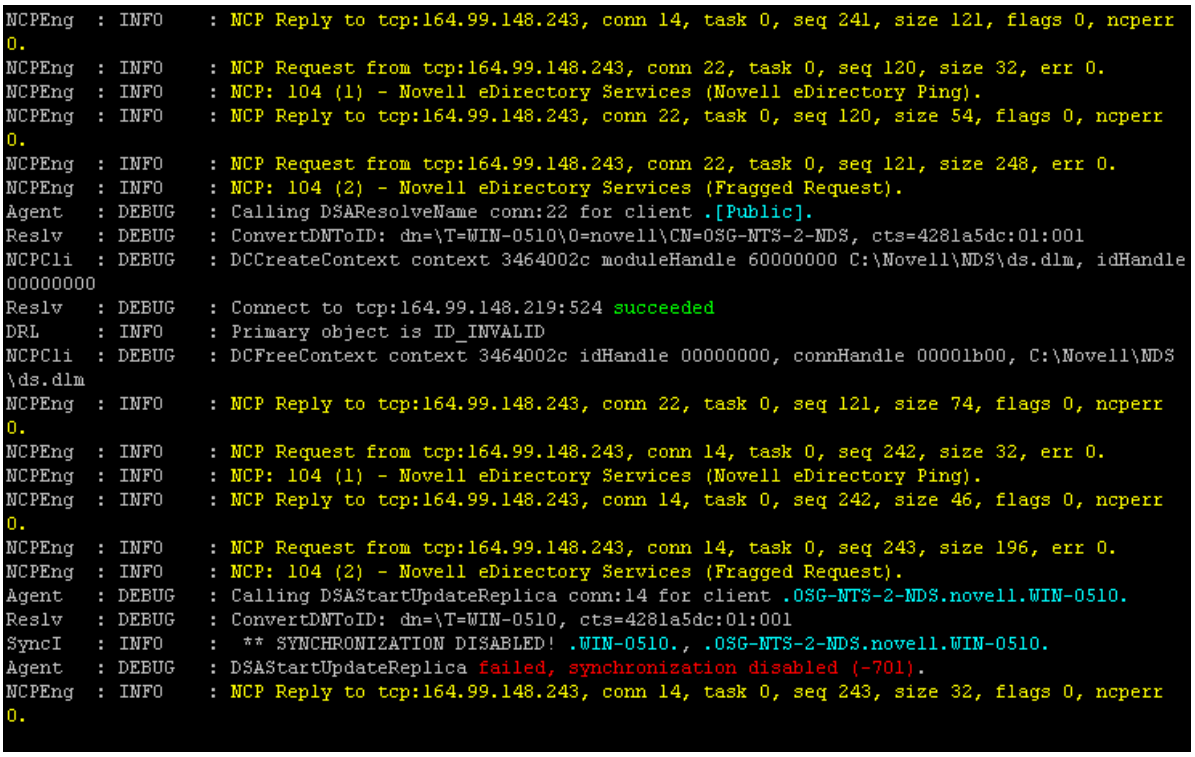
- 接続 ID に基づいてフィルタを無効にするには、次のように入力します。

  ```
  ndstrace conn
  ```

- 重大度に基づいてフィルタを無効にするには、次のように入力します。

  ```
  ndstrace svty
  ```

図 8-5 フィルタを適用したトレースメッセージのサンプル画面



Windows

次の手順を完了してトレースメッセージをフィルタします。

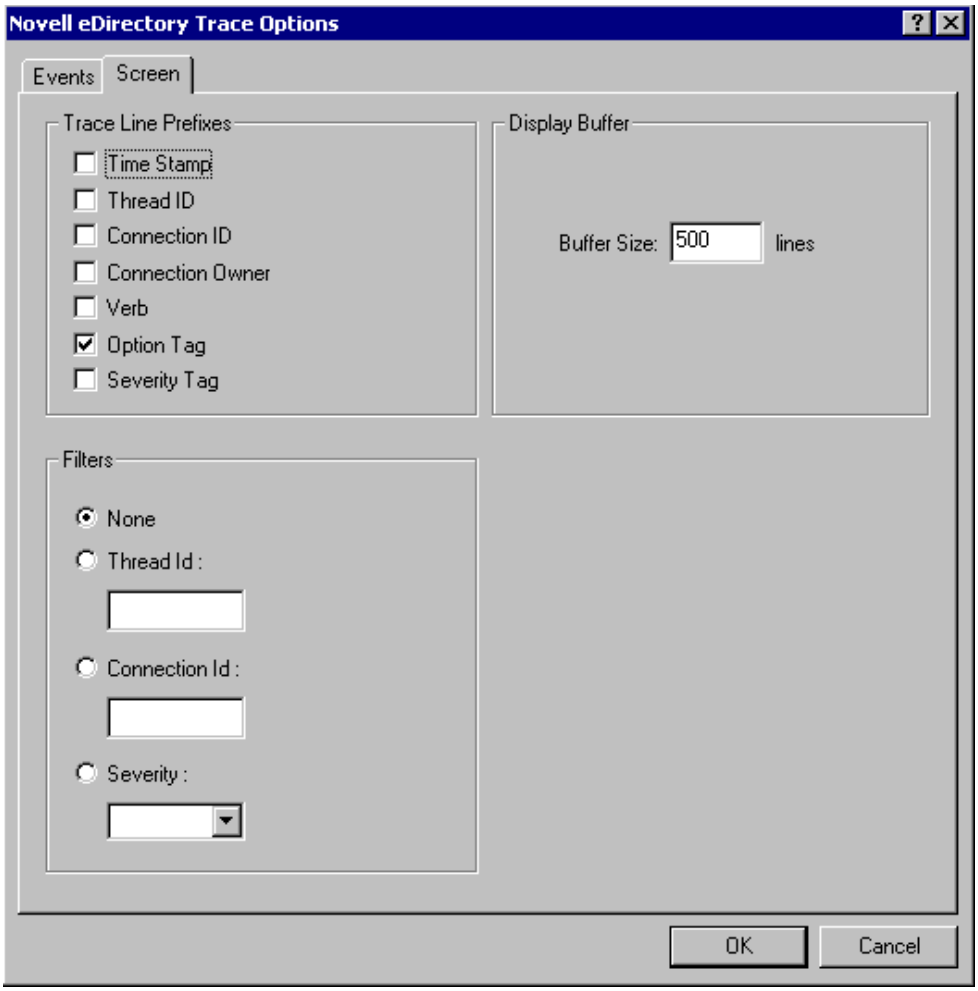
1 ［スタート］>［コントロールパネル］>［OpenText eDirectory サービス］の順にクリックします。

2 ［サービス］タブで、［dstrace.dlm］を選択します。

3 ［トレース］ウィンドウで、［編集］>［オプション］の順にクリックします。

［OpenText eDirectory トレースオプション］ダイアログボックスが表示されます。

図 8-6 Windows でのトレースオプション画面



4 ［画面］タブをクリックします。

5 ［フィルタ］グループからフィルタオプションを選択し、フィルタの値を入力します。

次の項目に基づいてメッセージをフィルタできます。

- スレッド ID
- 接続 ID
- 重大度

いずれかのフィルタを選択する前に、［トレース行プレフィックス］でそのフィルタが有効にされていることを確認します。

［なし］を選択するか、フィルタオプションの選択を解除すると、フィルタを無効にすることもできます。

注

- フィルタオプションとして**スレッド ID** または**接続 ID** を選択し、存在しない値を入力した場合、メッセージは画面に表示されません。ただし、他のメッセージはすべて ndstrace.log ファイルに記録されます。
- Windows では、トレースレベルの重大度は動作しません。

OpenText iMonitor のメッセージフィルタリング

接続 ID、スレッド ID、またはエラー番号に基づいて、OpenText iMonitor のトレースメッセージをフィルタできます。

接続 ID やスレッド ID に基づいてフィルタを行う場合は、［トレースの環境設定］タブでこれらを有効にしたことを確認します。

詳細については、OpenText iMonitor のオンラインヘルプを参照してください。

SAL メッセージのフィルタ

SAL は、エラーに関する包括的な情報を、オンデマンドでログに記録するために拡張されてきました。デバッグビルドでは、引数を使用してファンクションコールをトレースすることができます。

重大度レベルの設定

SAL_LogLevels パラメータを使用すると、SAL メッセージの重大度レベルを設定できます。SAL_LogLevels は、必要なログレベルから構成されたコンマ区切りのリストです。

下の表では、ログレベルについて説明します。

***表 8-4** SAL メッセージのフィルタパラメータ*

パラメータ名	説明
LogCrit	致命的なメッセージ。 デフォルトでは、このレベルは有効になっています。致命的エラーが記録されると、システムはシャットダウンされます。
LogErr	すべてのエラーメッセージ。 システムは機能し続けますが、結果は予測できません。
LogWarn	警告メッセージ。 発生する可能性のあるエラーの存在について通知される警告です。
LogInfo	情報メッセージ。

パラメータ名	説明
LogDbg	開発時のデバッグ用に使用されるデバッグメッセージです。 これらのメッセージは、バイナリサイズを削減するため、コンパイル時にリリースビルドから削除されます。
LogCall	ファンクションコールをトレースします。これらはデバッグメッセージのサブセットです。
LogAll	LogCall 以外のメッセージをすべて有効にします。

特定のログレベルの先頭に「-」を指定すると、そのレベルが無効になります。

例

LogInfo と LogDbg を除いたすべてのログレベルに基づいてフィルタを行う場合は、次の手順を完了させます。

Linux

1. ndsd を停止します。
2. 次のコマンドを入力します。

   ```
   export SAL_LogLevels=LogAll,-LogInfo,-LogDbg
   ```

3. ndsd を起動します。

Windows

1. DHost をシャットダウンします。
2. 次のコマンドをコマンドプロンプトの指示にしたがって入力します。

   ```
   set SAL_LogLevels=LogAll,-LogInfo,-LogDbg
   ```

   ```
   c:\novell\nds>dhost.exe /datadir=c:\novell\nds\DIBFiles\
   ```

3. DHost を再起動します。

ログファイルパスを設定する

SAL_LogFile 環境変数を使用すると、ログファイルの場所を指定できます。場所として指定できるのは、有効なパスの有効なファイル名、または次のいずれかです。

- コンソール : すべてのメッセージはコンソールに出力されます。
- syslog: Linux では、メッセージは syslog に記録されます。Windows では、メッセージは syslog という名前のファイルに記録されます。これはログのデフォルトの動作です。

  致命的なエラーはすべて、明確に無効にされている場合以外は、常に syslog に記録されます。

9 OpenText eDirectory サーバの SecretStore 環境設定

注 : OpenText eDirectory 9.2.7 以降では、SecretStore のサポートは非推奨になります。OpenText eDirectory 9.2.7 以上の SecretStore 関連の問題についてはサポートが提供されなくなります。

OpenText eDirectory をインストールすると、デフォルトで SecretStore の実行可能ファイルとライブラリがインストールされます。ただし、OpenText eDirectory の新しいインストールに対する SecretStore の設定はオプションです。OpenText eDirectory サーバをアップグレードする場合は、既存の設定が変更されることなくそのまま維持されます。Linux および Windows プラットフォームで次のコマンドを使用して、SecretStore の機能に対応するように OpenText eDirectory スキーマを拡張してください。

```
ice -S SCH -f /var/opt/novell/eDirectory/lib/nds-schema/sssv3.sch -D LDAP -
s
<serverIP> -d <adminDN>
```

例 : ice -S SCH -f /var/opt/novell/eDirectory/lib/nds-schema/sssv3.sch -D LDAP -s 1.2.3.4 -d cn=admin,o=administrators

SecretStore の設定および設定解除は、以降のセクションで説明する手順で行うことができます。

- 307 ページの 「Linux」
- 308 ページの 「Windows」

Linux

SecretStore の設定

SecretStore を設定するには、次の手順に従います。

1. 設定する場合は、ssscfg-c を実行します。
2. エントリ「ssncp」を /etc/opt/novell/eDirectory/conf/ndsmodules.conf に追加し、OpenText eDirectory の起動中にデフォルトで SecretStore モジュールがロードされるようにします。後から nss ユーティリティを使用して、SecretStore モジュールをロードまたはアンロードすることもできます。

SecretStore の設定解除

設定解除する場合は、ssscfg-d コマンドを実行します。エントリ「ssncp」が /etc/opt/novell/eDirectory/conf/ndsmodules.conf 内にある場合は、このエントリを削除します。

Windows

次の手順で SecretStore を設定および設定解除します。

1. 設定する場合は、ssscfg.exe-c を実行します。
2. 設定解除する場合は、ssscfg.exe-d を実行します。

ssscfg.exe ユーティリティは、eDirectoryInstallDrive:>\Novell\NDS\ ディレクトリにあります。OpenText eDirectory サーバの起動中に SecretStore モジュールを自動ロードするには、NDSCons.exe の GUI インタフェースで、ssncp.dlm モジュールを auto に設定します。

10 OpenText eDirectory ツリーのマージ

OpenText eDirectory のマージユーティリティを使用すると、2 つの OpenText eDirectory ツリーをマージして、単一の OpenText eDirectory ツリーを作成できます。マージされるのは Tree オブジェクトだけです。コンテナオブジェクトとそのリーフオブジェクトは、マージ後のツリー上でもそれぞれ異なるオブジェクトとして存在します。

ヒント : リーフオブジェクトを移動したり、パーティションをマージしたりするには、OpenText Identity Console を使用します。

マージする 2 つのツリーをそれぞれソースツリーおよびターゲットツリーといいます。1 つのツリーを別のツリーにマージする前に、ターゲットツリーにあるルートパーティションのレプリカを 1 つだけ残し、その他のすべてのレプリカを削除する必要があります。ターゲットツリーにルートパーティションのレプリカが 1 つしかない場合には、マージ処理を続行できます。マージ後は、ルートパーティションのレプリカが 2 つになります。1 つはターゲットツリー上にあったレプリカで、もう 1 つはマージ操作を実行したソースツリーのサーバ上にあったレプリカです。ルートパーティションのレプリカを追加する必要がある場合は、マージが完了した後に保存することができます。

マージ時にターゲットツリーサーバにルートパーティションのレプリカが複数ある場合、マスタレプリカを保持していないサーバで、外部参照オブジェクトの位置に関する問題が発生する可能性があります。外部参照オブジェクトは、サブオーディネートリファレンスのパーティションルートに含まれています。これは、パーティションの境界を表すルートパーティションのレプリカを持つ別のサーバに配置する必要があります。ソースツリーにあるルートパーティションの下位パーティションごとに、ターゲットツリーにあるサブオーディネートリファレンスのパーティションルートを持つ必要があります。エラーが生じた場合、同期ステータスに関する OpenText eDirectory エラーコード -605 が報告されます。この場合、DSRepair を使用して、エラーが発生したサーバのローカルデータベースを修復します。詳細については、350 ページの 「ローカルデータベースの修復の実行」を参照してください。

DSMerge を実行しても、コンテナ内の OpenText eDirectory 名またはコンテキストは変わりません。オブジェクト権とプロパティ権はマージ後のツリーでも保持されます。

この章では、次のトピックについて説明します。

- 310 ページの 「OpenText eDirectory ツリーの統合」
- 316 ページの 「サーバツリーの結合」
- 321 ページの 「ツリー名の変更」
- 322 ページの 「クライアントを使用したツリーのマージ」

OpenText eDirectory ツリーの統合

OpenText eDirectory ツリーをマージするには、Merge Tree コマンドを使用します。このコマンドを実行すると、2 つの OpenText eDirectory ツリーのルートをマージできます。マージされるのは Tree オブジェクトだけです。コンテナオブジェクトとそのリーフオブジェクトは、マージ後のツリー上でもそれぞれ異なるオブジェクトとして存在します。

マージする 2 つのツリーはそれぞれソースツリーおよびターゲットツリーといいます。ソースツリーのマージ先ツリーがターゲットツリーです。

DSMerge を実行しても、コンテナ内のオブジェクトの名前は変わりません。オブジェクト権とプロパティ権はマージ後のツリーでも保持されます。

- 310 ページの 「前提条件」
- 310 ページの 「ターゲットツリーの要件」
- 311 ページの 「スキーマの要件」
- 311 ページの 「ソースツリーをターゲットツリーへマージする」
- 311 ページの 「パーティションの変化」
- 312 ページの 「ソースツリーとターゲットツリーを準備する」
- 313 ページの 「マージする前の時刻の同期」
- 314 ページの 「2 つのツリーのマージ」
- 315 ページの 「マージ後の作業」

前提条件

- ❒ ソースツリーの［ルート］パーティションのマスタレプリカが格納されているサーバに、OpenText eDirectory がインストールされている必要があります。
- ❒ 正しい機能を維持するには、ソースツリーの他のサーバを OpenText eDirectory 8.8 以降にアップグレードする必要があります。

注 : 許可されたログインメソッドを削除するには、ldapdelete ツールまたは OpenText Identity Console を使用します。

ターゲットツリーの要件

- ❒ ターゲットツリーの［ルート］パーティションのマスタレプリカが格納されているサーバに、OpenText eDirectory がインストールされている必要があります。このサーバで他のバージョンの NDS® または OpenText eDirectory が実行されている場合は、マージ操作が正常に完了しません。
- ❒ 正しい機能を維持するには、ターゲットツリーの他のサーバを OpenText eDirectory 8.8 以降にアップグレードする必要があります。

- ソースツリーとターゲットツリーの両方で、Tree のサブオーディネートコンテナを同じ名前で維持することはできません。2 つのツリーをマージする前に、一方のコンテナをリネームする必要があります。
- ソースツリーとターゲットツリーの両方にセキュリティオブジェクトがある場合は、ツリーをマージする前にどちらかを削除する必要があります。

スキーマの要件

マージ操作を実行する前に、2 つのツリーのスキーマが正確に一致している必要があります。ツリーごとに [Root] パーティションのマスタレプリカを含むサーバで DSRepair を実行する必要があります。[リモートスキーマのインポート] オプションを使用して、各ツリーで他のツリーのスキーマがすべて認識されていることを確認します。

1. Ndsrepair -S -Ad を実行します。
2. 管理者の名前とパスワードを指定します。
3. [ツリー (Tree)] からオプション 4 の [Import スキーマ (Import schema)] (グローバルスキーマオプション) を選択します。
4. IP アドレスまたはツリー名を指定します。

ソースツリーをターゲットツリーへマージする

ツリーをマージすると、ソースツリー内のサーバがターゲットツリーに組み込まれます。

ターゲットツリーの Tree オブジェクトがソースツリーのオブジェクトの新しい Tree オブジェクトになり、ソースツリーにあるすべてのサーバのツリー名がターゲットツリーのツリー名に変わります。

ターゲットツリー内のサーバのツリー名はマージ後も変わりません。

ソースツリーの Tree オブジェクトの下位オブジェクトは、ターゲットツリーの Tree オブジェクトの下位オブジェクトになります。

パーティションの変化

マージ実行時には、ソースツリーの Tree オブジェクトの下にあるオブジェクトが DSMerge によって複数のパーティションに分けられます。

続いて、ソースツリー内のサーバから、Tree パーティションのレプリカがマスタレプリカを除いてすべて削除されます。ソースツリーのマスタレプリカを保持していたサーバには、ターゲットツリーのツリーパーティションのレプリカが作成されます。

図 10-1 および図 10-2 は、2 つのツリーをマージした場合のパーティションの変化を示します。

図 10-1　マージ前の OpenText eDirectory ツリー

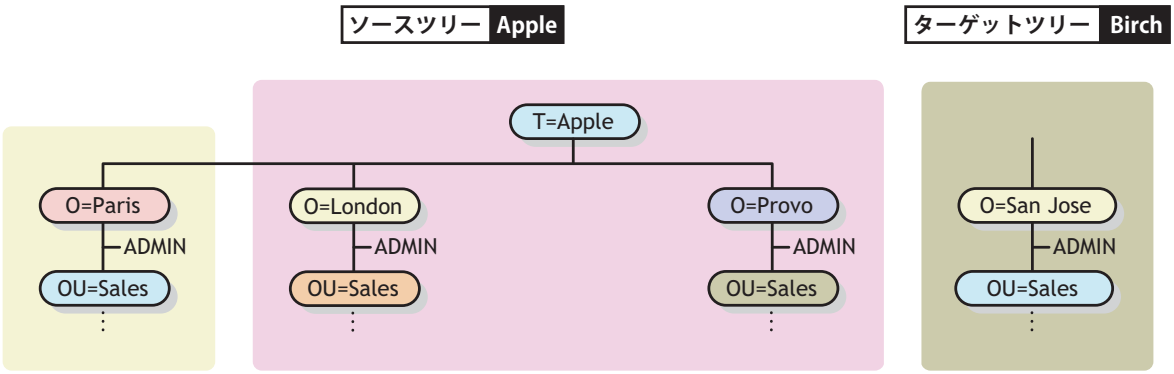
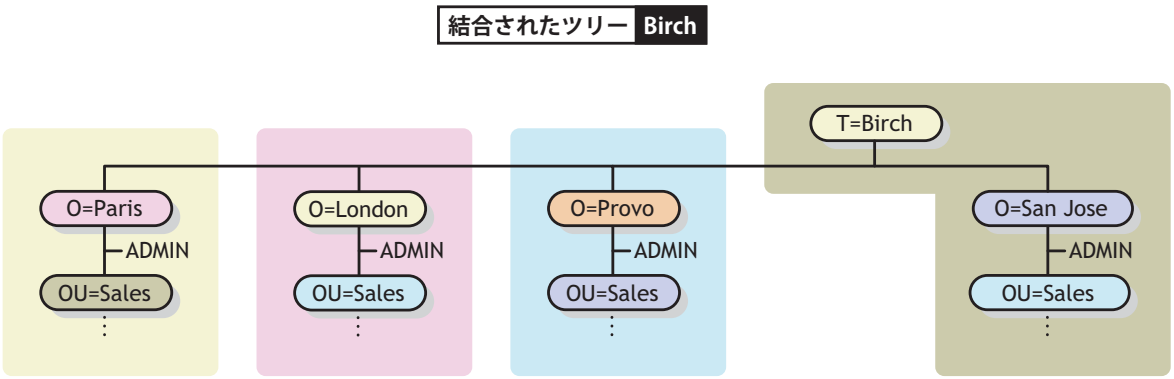


図 10-2　マージ後の OpenText eDirectory ツリー



ソースツリーとターゲットツリーを準備する

マージ操作を開始する前に、そのマージ操作の影響を受けるすべてのサーバが安定した状態で同期していることを確認する必要があります。次の表で、マージ対象のソースツリーとターゲットツリーの準備に関する前提条件について説明します。

必要条件	必要な措置
ソースツリーまたはターゲットツリーのツリーパーティションのレプリカを保持しているすべてのサーバ上で WANMAN がオフになっている。	WANMAN ポリシーを調べて、WAN 通信の制約によるマージ操作への支障がないことを確認します。必要があれば、マージ操作を開始する前に WANMAN をオフにします。
ソースツリーの Tree オブジェクトに別名またはリーフオブジェクトが存在しない。	ソースツリーの Tree オブジェクトに存在する別名またはリーフオブジェクトを削除します。

必要条件	必要な措置
ソースツリーおよびターゲットツリーに同じ名前がない。	同じ名前がある場合には、ソースツリーまたはターゲットツリーのオブジェクト名を変更します。コンテナオブジェクト名を変更したくない場合は、コンテナの 1 つから同じツリー内の別のコンテナにオブジェクトを移動し、空になったコンテナを DSMerge 実行前に削除します。 Tree オブジェクトの直下でないコンテナオブジェクトの場合は、両ツリーに同じコンテナオブジェクトがあってもかまいません。
ソースツリー上にログイン接続が存在しない。	ソースツリーのすべての接続を終了します。
ソースツリーとターゲットツリーの OpenText eDirectory のバージョンが同じである。	ルートパーティションのレプリカを持つ、OpenText eDirectory サーバ以外のすべてのサーバをアップグレードします。
ターゲットツリー内のルートレプリカのコピーが 1 つである。	ターゲットツリー上の、マスタレプリカを除いたすべてのレプリカを削除します。
ソースツリーおよびターゲットツリーのスキーマが同じである。	DSMerge を実行します。出力されるレポートを見てスキーマに問題があることがわかった場合は、DSRepair を使用してスキーマを一致させます詳細については、359 ページの 「リモートスキーマをインポートする」を参照してください。DSMerge を再実行します。
単一のツリーだけが、ツリーのルートの下位にセキュリティコンテナを保持できる。	ソースツリーおよびターゲットツリーの両方がセキュリティコンテナを保持している場合は、「843 ページの付録 A「NMAS の注意事項」」で説明されている手順に従って、一方のコンテナを削除します。

マージ操作は単独のトランザクションなので、実行中に停電やハードウェアエラーが発生しても重大な障害にはつながりません。ただし、DSMerge を実行する前に、あらかじめ eDirectory データベースの通常のバックアップをとっておくことをお勧めします。詳細については、459 ページの第 15 章「OpenText eDirectory のバックアップと復元」を参照してください。

マージする前の時刻の同期

重要 : 時刻同期の正確な環境設定は複雑な作業です。ツリーをマージする前に、両ツリーを同期させるのに十分な時間があることを確認します。

時刻の異なる複数のタイムソースが使用されていたり、ツリー内のサーバの中に時刻が同期されていないものがあると、OpenText eDirectory は正しく機能しません。

マージを実行する前に、両ツリーにあるすべてのサーバの時刻が同期されていること、およびこれらのサーバがタイムソースとして同じタイムサーバを使用していることを確認します。ただし、ターゲットツリーの時刻は 5 分以内であればソースツリーの時刻より進んでいてもかまいません。

一般に、1 つのツリー上に存在できるタイムサーバは、リファレンスタイムサーバまたはシングルタイムサーバのどちらか 1 つだけです。同様に、マージ後のツリー上に存在できるタイムサーバも、リファレンスタイムサーバまたはシングルタイムサーバのどちらか 1 つだけです。

マージする両ツリーのそれぞれにリファンレスタイムサーバまたはシングルタイムサーバがある場合は、どちらかのツリーの設定をもう一方のツリーにあるリファンレスタイムサーバまたはシングルタイムサーバに変更して、マージ後のツリー上でリファレンスタイムサーバまたはシングルタイムサーバが 1 つだけになるようにします。

タイムサーバのタイプの詳細については、『*OES Planning and Implementation Guide (OES プラニングおよび実装ガイド)* (http://www.novell.com/documentation/oes11/oes_implement_lx/data/time.html)』の「Time Services (時刻サービス)」を参照してください。

2 つのツリーのマージ

すべてのメニューオプションの機能を使用できるようにするには、Tree パーティションのマスタレプリカが格納されているサーバ上で DSMerge を実行します。

マスタレプリカが格納されている場所が不明な場合は、マスタレプリカを必要とする操作を実行すると、正確なサーバ名がプロンプトに表示されます。

マージ操作を実行するには、次の方法を使用します。

- コマンドラインクライアント

  詳細については、322 ページの 「クライアントを使用したツリーのマージ」を参照してください。

大きなツリーをマージするときは、Tree オブジェクト直下にあるオブジェクトの数が少ないほうのツリーをソースツリーとして指定したほうが、処理速度が大幅に速くなります。Tree オブジェクトの直下にあるオブジェクトすべてに対して新しいパーティションが作成されるため、この方法をとると、マージ実行時に分割されるパーティションの数が少なくてすむためです。

ソースツリーの名前はマージ後には存在しなくなります。したがって、場合によっては、クライアントワークステーションの環境設定を変更する必要があります。Novell Client for DOS/Windows の場合は、net.cfg ファイルの優先ツリーステートメントおよび優先サーバステートメントを確認します。Novell Client for Windows の場合は、クライアントのプロパティページにある優先ツリーステートメントおよび優先サーバステートメントを確認します。

優先サーバが使用されている場合は、ツリーのマージやツリー名の変更を行っても、そのクライアントは名前によってサーバにログインした状態のままなので操作による影響はクライアント側にはありません。優先ツリーが使用されている場合は、ツリーのマージやツリー名の変更を行うと、元のツリー名はなくなります。マージを行った後にはターゲットツリーの名前だけが残ります。優先ツリーの名前を新しいツリー名に変更します。

ヒント : ターゲットツリーの名前はマージの結果作成されるツリーの名前としてそのまま残るので、アップデートする必要のあるクライアントワークステーションの数を少なくするには、クライアントワークステーションの数が多い方のツリーをターゲットツリーとし

て指定します。または、マージ操作の後でツリー名を変更して、最終的なツリー名が接続されているクライアントワークステーションの数が多い方のツリーに一致するようにします。詳細については、321 ページの 「ツリー名の変更」を参照してください。

次の前提条件の一覧を使用して、マージ操作の準備ができているか確認します。

- ❒ OpenText Identity Console によるソースツリーのサーバへのアクセス権があること
- ❒ マージする両ツリーの Tree オブジェクトに対するスーパバイザオブジェクト権を持つ管理者オブジェクトの名前とパスワードがわかっていること
- ❒ 2 つのツリーの OpenText eDirectory データベースがバックアップされていること
- ❒ 両ツリー内のすべてのサーバが同期されていて、同じタイムソースを使用していること
- ❒ (オプション) ツリー内のサーバがすべて動作可能であること (動作不能状態のサーバは動作可能になった時点で自動的に更新されます)。
- ❒ 「312 ページの 「ソースツリーとターゲットツリーを準備する」」に表示された前提条件を確認すること

マージプロセスには 2、3 分しかかかりませんが、次のような場合は、付随的な作業が必要になるため、マージ操作が完了するまでに要する時間が長くなります。

- Tree オブジェクトの下に多数のオブジェクトがあり、パーティションに分ける必要がある
- ソースツリーに多数のサーバがあり、ソースツリーのツリー名を変更する必要がある

2 つのツリーをマージするには、「DSMerge eMTool の使用」で説明されている手順に従います。

注 : 親パーティションが非 EBA 対応で、子パーティションが EBA 対応の場合は、この 2 つのパーティションをマージしないでください。これを行うと、EBA 機能が停止する可能性があります。

マージ後の作業

2 つのツリーをマージした後は、状況に応じて次の手順を行ってください。

1. すべてのツリー名が正しく変更されたことを確認します。
2. マージ操作によって作成された新しいパーティションを確認します。

   新しいツリーに小さいパーティションが多数存在する場合や、関連する情報が格納されたパーティションが複数存在する場合は、これらのパーティションをマージすることもできます。詳細については、161 ページの 「パーティションのマージ」を参照してください。
3. DSMerge 実行前にリーフオブジェクトや別名をツリーから削除した場合は、ツリーにそれらのリーフオブジェクトや別名を再作成します。
4. OpenText eDirectory ツリーのパーティション構成を調べます。

   マージ後のツリー内でのレプリカの位置はソースツリー内での位置と異なる場合があります。パーティション構成の変更や検査は慎重に行ってください。

5 クライアントワークステーションの環境設定を更新します。

Novell Client for Windows の場合は、クライアントのプロパティページの優先ツリーステートメントおよび優先サーバステートメントを確認するか、またはターゲットツリーの名前を変更します。

優先サーバが使用されている場合は、ツリーのマージやツリー名の変更を行っても、そのクライアントは名前によってサーバにログインした状態のままなので操作による影響はクライアント側にはありません。優先ツリーが使用されている場合は、ツリーのマージやツリー名の変更を行うと、元のツリー名はなくなります。マージを行った後にはターゲットツリーの名前だけが残ります。優先ツリーの名前を新しいツリー名に変更します。

ソースツリーの Tree オブジェクトの ACL(アクセス制御リスト) は保持されます。したがって、この Tree オブジェクトに対してソースツリーのユーザが持つ管理者権限は、引き続き有効です。

マージ後も、両ツリーの管理者ユーザは存在したままで、それぞれのコンテナオブジェクトによって固有のものとして識別されます。

セキュリティ上必要であれば、2 つの管理者ユーザオブジェクトの一方を削除するか、両オブジェクトの権利を制限することもできます。

サーバツリーの結合

[**ツリーの結合**] オプションを使用すると、単一サーバソースツリーの Tree オブジェクトを、ターゲットツリーにある指定のコンテナに結合できます。結合が完了すると、ソースツリーのツリー名は、ターゲットツリーのツリー名になります。

結合操作中に、DSMerge によってソースツリーの Tree オブジェクトのオブジェクトクラスがドメインに変更され、新しいパーティションが作成されます。新しいドメインオブジェクトは、新しいパーティションのパーティションルートになります。ソースツリーのツリーオブジェクトの下にあるすべてのオブジェクトはドメインオブジェクトに属すことになります。

ターゲットツリーの管理者には、作成されたツリーのルートコンテナに対する権利があるため、結合後のソースツリーのルートに対する権利もあります。

注 : 権利継承が再計算されて有効になるには、数時間かかる可能性があります。この時間は、ツリーの複雑さ、サイズ、パーティション数によって異なります。

ソースツリーの管理者は、新しく作成されたドメインオブジェクト内でのみ権利を所有します。

図 10-3 と図 10-4 に、ツリーを特定のコンテナに結合した場合の変化を示します。

図 10-3 結合前の OpenText eDirectory ツリー

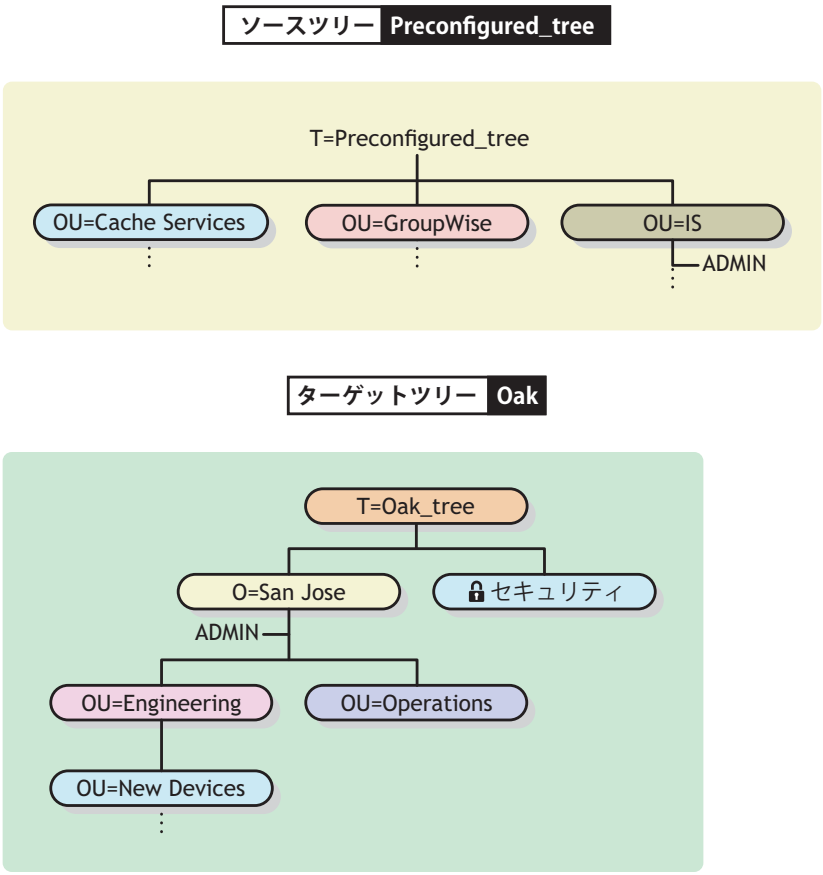
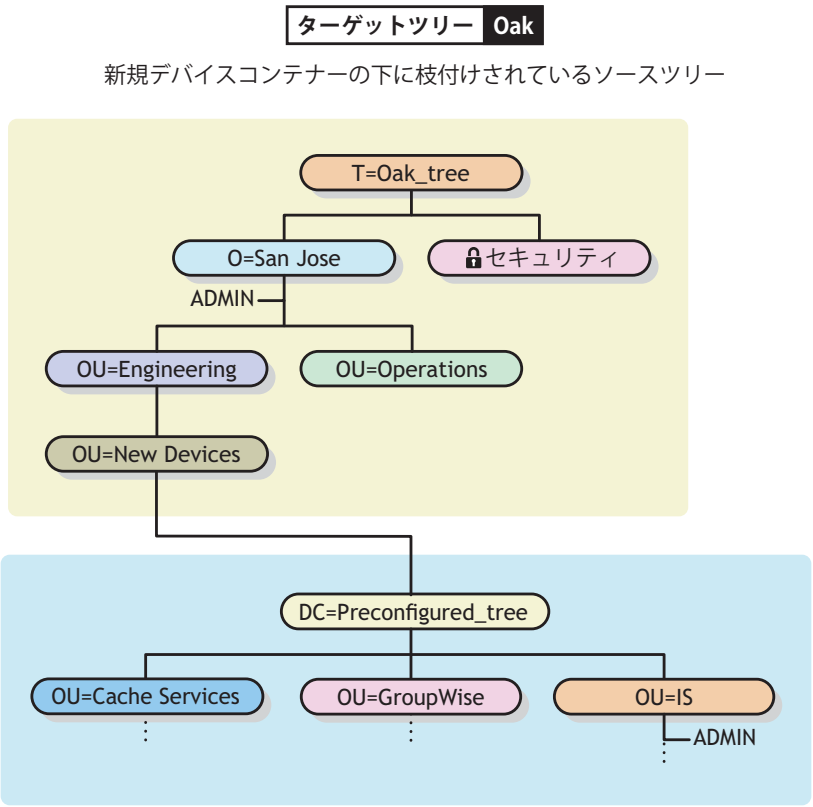


図 10-4 結合後の OpenText eDirectory ツリー



このセクションでは、次の情報について説明します。

- 318 ページの 「コンテキスト名の変更について」
- 319 ページの 「ソースツリーとターゲットツリーを準備する」
- 321 ページの 「結合操作における包含要件」
- 321 ページの 「ソースツリーとターゲットツリーを結合する」

コンテキスト名の変更について

ソースツリーをターゲットツリーのコンテナに結合すると、ソースツリー内にあるオブジェクトの識別名の後ろに、ソースツリー名、続いてソースツリーをマージしたターゲットツリーコンテナの識別名が追加されます。相対識別名は変わりません。

たとえば、区切り文字としてドットを使用している場合、ソースツリー Preconfigured_tree 内にあるオブジェクト Admin のタイプ付きの名前は次のようになります。

```
CN=Admin.OU=IS.T=Preconfigured_tree
```

Preconfigured_tree を Oak_tree の New Devices コンテナにマージした後は、Admin のタイプ付きの名前は次のようになります。-

```
CN=Admin.OU=IS.DC=Preconfigured_tree.OU=Newdevices.
OU=Engineering.O=Sanjose.T=Oak_tree.
```

注 : コンテナ DN を含め、どのタイプの DN でも、使用できる最大文字数は、255 文字です。この制限は、1 つのツリーのルートをターゲットツリーの下端近くにあるコンテナに結合する場合に特に重要です。

Oak_tree の後にある最後のドット (Oak_tree.) は、この識別名を構成する最後の要素がツリー名であることを示しています。このドットを省略する場合は、ツリー名も省略します。

ソースツリーとターゲットツリーを準備する

結合操作を開始する前に、その結合操作の影響を受けるすべてのサーバが安定した状態であることを確認する必要があります。次の表では、結合前のソースツリーとターゲットツリーの準備に関する前提条件について説明します。

必要条件	必要な措置
ソースツリー内のサーバが 1 つである。	ソースツリーから、1 つだけ残してすべてのサーバを削除します。
ソースツリーの Tree オブジェクトに別名またはリーフオブジェクトが存在しない。	ソースツリーの Tree オブジェクトに存在する別名またはリーフオブジェクトを削除します。
結合コンテナ内に同じ名前がない。	同じ名前がある場合は、ターゲットツリーの結合コンテナ内のオブジェクトまたはソースツリー内のオブジェクト名を変更します。 オブジェクト名を変更したくない場合は、コンテナの 1 つから同じツリー内の別のコンテナにオブジェクトを移動し、空になったコンテナを DSMerge 実行前に削除します。詳細については、107 ページの第 3 章「オブジェクトの管理」を参照してください。 同じペアレントオブジェクトの直下でないコンテナオブジェクトの場合は、両ツリーに同じコンテナオブジェクトがあってもかまいません。オブジェクトは、直接のコンテナオブジェクトによって識別されます。
ソースツリーおよびターゲットツリーの eDirectory バージョンが 8.51 SP2a 以降である。	DSMerge は、OpenText eDirectory の適切なバージョンを検索します。使用できるバージョンが見つからない場合、DSMerge はエラーを返します。OpenText eDirectory の最新バージョンは、ソフトウェアライセンスおよびダウンロードポータル (https://www.microfocus.com/en-us/products) から入手できます。

必要条件	必要な措置
ターゲットツリーを結合するコンテナがレプリカを持たないパーティション（単一サーバパーティション）にある。	ターゲットコンテナに複数のレプリカがある場合は、次のうち 1 つを実行します。 ◆ このコンテナに関連付けられたパーティションをマスタレプリカにし、その他のレプリカを削除します。 ◆ または、ターゲットツリーの結合コンテナを別のパーティションとして切り離し、レプリカを削除します。 結合が完了した後は、パーティションの関連付けを再構築できます。
ターゲットコンテナを保持するサーバがルートパーティションのレプリカも保持している。	サーバがルートのレプリカを保持していない場合、ディレクトリはターゲットツリーの管理権を確認できないので、結合は失敗し、エラー「-672 アクセスできません」が表示されます。 OpenText Identity Console を使用してルートのレプリカを追加します。詳細については、164 ページの 「レプリカを追加する」を参照してください。
ソースツリーおよびターゲットツリーのスキーマが同じである。	DSMerge で［結合］オプションを実行します。出力されるレポートからスキーマに問題があることがわかった場合は、ターゲットツリー上で DSRepair を実行してソースツリーからスキーマをインポートします。 結合操作によって、自動的にスキーマがターゲットツリーからソースツリーにインポートされます。 DSMerge を再実行します。
単一のツリーだけが、ツリーのルートの下位にセキュリティコンテナを保持できる。	ソースツリーおよびターゲットツリーの両方がセキュリティコンテナを保持している場合、843 ページの付録 A 「NMAS の注意事項」で説明されている手順に従って、一方のコンテナを削除します。
ソースツリーの時刻基準は、再設定する必要があります。	ソースツリーは、ターゲットツリーのサーバからタイムソースを取得するように、セカンダリサーバとして設定される必要があります。 Timesync を再設定する場合は、『*OES Planning and Implementation Guide*』の「Configuring and Administering Time Synchronization (http://www.novell.com/documentation/oes11/oes_implement_lx/data/time.html#time-cfgnadmin)」を参照してください。

結合操作における包含要件

ソースツリーをターゲットツリーのコンテナに結合するには、ターゲットツリーのコンテナでソースツリーの受け入れ準備を整えておく必要があります。ターゲットツリーのコンテナは、クラスドメインのオブジェクトを包含できなければなりません。包含に問題があると、結合操作中に「-611 不正な包含ルールです」というエラーが発生します。

次の表の情報を使用して、DSRepair を実行して包含リストを変更する必要があるかどうかを判断します。

ターゲットツリーのコンテナ要件	ターゲットツリーのコンテナオブジェクトが、包含リスト内のドメインオブジェクトを包含していること。 これを確認するには、[**OpenText iMonitor**] > [**スキーマ**] を使用します。包含リストにドメインが包含されていない場合、DSRepair を実行してスキーマを拡張します。
ソースツリーの要件	結合操作によって、ソースツリーのルートがクラスツリールートからクラスドメインに変更されます。Tree の下位オブジェクトはすべて、スキーマルールに従い、クラスドメインに適切に包含されている必要があります。 これを確認するには、[**OpenText iMonitor**] > [**スキーマ**] を使用します。包含リストにドメインが包含されていない場合、DSRepair を実行してスキーマを拡張します。

格納条件が満たされていない場合は、「OpenText eDirectory サーバ上で DSRepair を実行する」の説明に従って、DSRepair を実行してスキーマを修正します。

ソースツリーとターゲットツリーを結合する

前提条件を満たしていることを確認し、DSMerge を使用して結合を実行します。詳細については、「DSMerge eMTool のオプション」を参照してください。

ツリー名の変更

マージする 2 つのツリーの名前が同じ場合は、どちらかの名前を変更する必要があります。

ここで名前を変更できるのはソースツリーだけです。ターゲットツリーの名前を変更するには、ターゲットツリー上のいずれかのサーバに対して OpenText Identity Console のツリー名の変更ウィザードを実行する必要があります。

ツリー名を変更しても、バインダリコンテキストは自動的には変更されません。autoexec.ncf ファイルで設定されたバインダリコンテキストセットにもツリー名 (例 : SET Bindery Context = O=n.*test_tree_name*) が含まれるため、ツリー名が最近変更されたサーバでは、ツリー名が変更される前のコンテキストは使用されません。

したがって、ツリー名を変更した場合、クライアントワークステーションの環境設定の変更が必要になる可能性があります。Novell Client for DOS/Windows の場合は、net.cfg ファイルの優先ツリーステートメントおよび優先サーバステートメントを確認します。Novell Client for Windows の場合は、クライアントのプロパティページにある優先ツリーステートメントおよび優先サーバステートメントを確認します。

優先サーバが使用されている場合は、ツリーのマージやツリー名の変更を行っても、そのクライアントは名前によってサーバにログインした状態のままなので操作による影響はクライアント側にはありません。優先ツリーが使用されている場合は、ツリーのマージやツリー名の変更を行うと、元のツリー名はなくなります。マージを行った後にはターゲットツリーの名前だけが残ります。優先ツリーの名前を新しいツリー名に変更します。

2 つのツリーをマージする場合、ターゲットツリーの名前はマージの結果作成されるツリーの名前としてそのまま残るので、アップデートする必要のあるクライアントワークステーションの数を少なくするには、クライアントワークステーションの数が多い方のツリーをターゲットツリーとして指定します。

または、マージの後でツリー名を変更して、最終的なツリー名がクライアントワークステーションの数が多い方のツリーに一致するようにすることもできます。

マージ後のツリーの名前は、元のソースツリーの名前に変更することもできます。このオプションを選択する場合は、ターゲットツリーにあるクライアントワークステーションの net.cfg ファイルを更新する必要があります。

次の前提条件の一覧を使用して、リネーム操作の準備ができているか確認します。

- ソースツリー上のサーバコンソールへのアクセス権があること、またはこのサーバとの RCONSOLE セッションが確立済みであること
- ソースツリーの Tree オブジェクトに対するスーパバイザオブジェクト権があること
- (オプション) ツリー内のサーバがすべて動作可能であること (動作不能状態のサーバは動作可能になった時点で自動的に更新されます)。

ツリーの名前を変更するには、「DSMerge eMTool のオプション」で説明されている手順に従います。

クライアントを使用したツリーのマージ

eMBox (OpenText eDirectory Management Toolbox) クライアントはコマンドライン Java クライアントで、これを使用すると DSMerge にリモートアクセスできます。emboxclient.jar ファイルは、eDirectory の一部としてサーバにインストールされます。JVM をインストールしていれば、どのコンピュータでも実行できます。クライアントの詳細については、「618 ページの 「コマンドラインクライアントの使用」」を参照してください。

DSMerge eMTool を使用する

1 コマンドラインで次のように入力して、対話式モードでクライアントを実行します。

```
java -cp path_to_the_file/emboxclient.jar  -i
```

(クラスパスにすでに emboxclient.jar ファイルが設定されている場合は、「java -i」とだけ入力します)。

クライアントのプロンプトが次のように表示されます。

```
Client>
```

2 DSMerge を実行する (ソースツリーになる) サーバにログインするには、次のコマンドを入力します。

```
login -sserver_name_or_IP_address -pport_number
-uusername.context -wpassword -n
```

ポート番号は通常 80 または 8028 です。ただし、すでにそのポートを使用している Web サーバが存在する場合は異なります。-n オプションを使用すると、非セキュア接続が開始されます。

クライアントはログインが成功したかどうかを表示します。

3 次の構文を使用してマージコマンドを入力します。

dsmerge.*task options*

たとえば、「dsmerge.m -uadmin -ptest -TApple -Uadmin -Ptest」と入力すると、ターゲットツリー Apple (ターゲットツリーのユーザ名は Admin、ユーザパスワードは test) が、現在ログインしているソースツリー (ソースツリーのユーザ名は Admin、ユーザパスワードは test) にマージされます。

「dsmerge.g -uadmin -ptest -TOrange -Uadmin -Ptest -CFruit」と入力すると、現在ログインしているソースツリー (ソースツリーのユーザ名は Admin、ユーザパスワードは test) が、ターゲットツリー Orange(ターゲットツリーのユーザ名は Admin、ユーザパスワードは test) に含まれるコンテナ Fruit に結合されます。

各スイッチの間にはスペースが必要です。スイッチの順序は重要ではありません。

クライアントは DSMerge 操作が成功したかどうかを表示します。

DSMerge eMTool オプションの詳細については、「323 ページの 「DSMerge eMTool オプション」」を参照してください。

4 クライアントからログアウトするには、次のコマンドを入力します。

```
logout
```

5 クライアントを終了するには、次のコマンドを入力します。

```
exit
```

DSMerge eMTool オプション

次の表に、DSMerge eMTool オプションを示します。クライアントで list -t dsmerge コマンドを使用して、DSMerge オプションの詳細を表示することもできます。詳細については、622 ページの 「eMTool とそのサービスを表示する」を参照してください。

マージ操作	クライアントのコマンド
ツリーの名前が変更できるかどうかチェックする	dsmerge.pr -u*User* -p*User_password* -n*New_tree_name*
ツリー名を変更する	dsmerge.r -u*User* -p*User_password* -n*New_tree_name*
2 つのツリーがマージできるかチェックする	dsmerge.pm -u*Source_tree_user*-p*Source_tree_user_password* -T*Target_tree_name* -U*Target_tree_user* -P*Target_tree_password*
2 つのツリーのマージ	dsmerge.m -u*Source_tree_user*-p*Source_tree_user_password* -T*Target_tree_name*-U*Target_tree_user* -P*Target_tree_password*
ソースツリーがターゲットツリーのコンテナに結合できるかチェックする	dsmerge.pg -u*Source_tree_user* -p*Source_tree_user_password* -T*Target_tree_name* -U*Target_tree_user* -P*Target_tree_password* -C*Target_tree_container*
ソースツリーをターゲットツリーのコンテナに結合する	dsmerge.g -u*Source_tree_user* -p*Source_tree_user_password* -T*Target_tree_name* -U*Target_tree_user* -P*Target_tree_password* -C*Target_tree_container*
実行中の DSMerge 操作をキャンセルする	キャンセル

11 OpenText eDirectory のデータを暗号化する

OpenText eDirectory では、次の状態にある特定のデータを暗号化できます。

- ディスクに保存されている。
- 2 つ以上の OpenText eDirectory サーバ間で転送される。そのため、機密データのセキュリティを強化できます。

次のものを暗号化することによって、データを保護できます。

- 属性 : ディスクに格納されている機密データを保護する場合。

  詳細については、325 ページの 「暗号化属性」を参照してください。
- レプリケーション : OpenText eDirectory サーバ間のレプリケーション中に機密データを保護する場合。

  「336 ページの 「暗号化レプリケーション」」を参照してください。

重要 : EBA (Enhanced Background Authentication) を導入すると、EBA 対応の OpenText eDirectory サーバ間でデータレプリケーションが行われる際に、自動的にデータが暗号化されます。サーバの一方が EBA 対応でない場合は、暗号化複製ポリシーを設定してデータを暗号化することができます。

暗号化属性

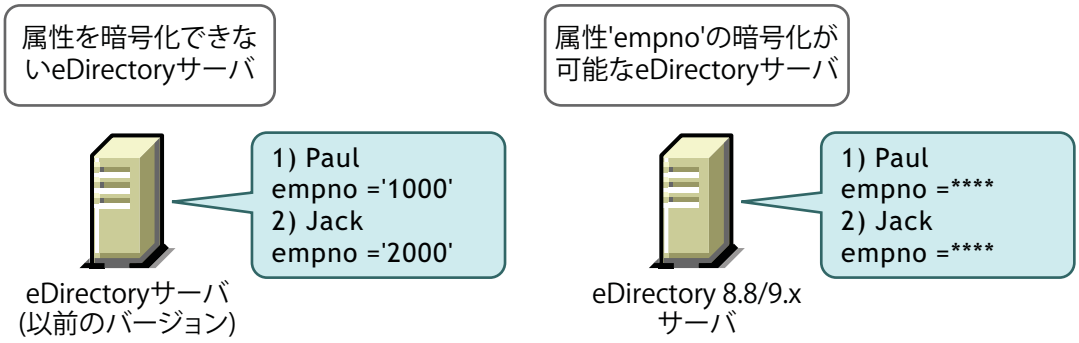
データがディスクに格納されている間は属性を暗号化してデータを保護できます。暗号化属性はサーバ固有の機能です。この機能は、銀行顧客のクレジットカード番号のような機密データを保護する必要がある場合に使用できます。

属性を暗号化すると、属性値がエンコードされます。たとえば、DIB に格納されている属性 empno を暗号化するとします。empno=1000 である場合、属性値 (1000) はクリアテキスト形式ではディスクに保存されません。この暗号化された値は、セキュリティ保護されたチャネルからディレクトリにアクセスした場合にのみ読み取ることができます。

方式内のすべての属性に対して暗号化を有効にすることができます。ただし、共通名 (CN) 属性に対してではなく、重要なデータに対してのみ暗号化を有効にすることをお勧めします。暗号化する属性の決定については、「343 ページの 「データを暗号化するときにデータの完全な安全性を確保する」」を参照してください。

Public およびサーバが読み込める暗号化属性へのアクセスに制限はありません。つまり、クライアントはクリアテキストを使用してこれらの属性にアクセスすることができ、ユーザはこれらの暗号化用の属性に DIB レベルでマークすることができます。スキーマで「パブリック読込み」フラグが設定されている属性に対して暗号を有効にしても、セキュアではない手段によるその属性へのアクセスは防止されません。

図 11-1 暗号化属性



OpenText eDirectory 内のデータは次のいずれかの方法で保存できます。

- DIB(Data Information Base) またはデータベースに保存
- バックアップデータとして保存
- LDIF ファイル

属性を暗号化するには、暗号化属性ポリシーを作成してサーバに適用します。

属性を暗号化するには、OpenText Identity Console を使用して次の手順を実行します。

1 暗号化属性ポリシーを作成して定義します。

1a 暗号化する属性を選択します。

1b 属性の暗号化スキームを選択します。

詳細については、「329 ページの 「暗号化された属性ポリシーの作成」」を参照してください。

2 サーバに暗号化属性ポリシーを適用します。

詳細については、「329 ページの 「暗号化された属性ポリシーの編集 / 変更」」を参照してください。

LDAP を使用して属性を暗号化することもできます。

詳細については、「330 ページの 「LDAP を使用して暗号化属性ポリシーを管理する」」を参照してください。

注 : 暗号化属性ポリシーの割り当ては、リンバの実行時に有効になります。

ベストプラクティスとして、次の操作を行うことをお勧めします。

- 暗号化する属性のうち、重要な属性のみをマークします。暗号化する属性すべて (パブリックやサーバが読み込める属性など) にマークはしないでください。
- 暗号化する属性をマークする場合は、強力な暗号化アルゴリズムである AES を使用してください。

このセクションでは、次の情報について説明します。

- 327 ページの 「暗号化方式を使用する」
- 328 ページの 「暗号化属性ポリシーを管理する」
- 332 ページの 「暗号化属性にアクセスする」
- 334 ページの 「暗号化属性を表示する」
- 335 ページの 「バックアップデータを暗号化 / 復号化する」
- 335 ページの 「暗号化属性を含む DIB ファイルセットのクローンを作成する」
- 335 ページの 「レプリカリングに OpenText eDirectory サーバを追加する」
- 335 ページの 「後方互換性」
- 335 ページの 「暗号化属性に移行する」
- 336 ページの 「暗号化属性のレプリカを作成する」

暗号化方式を使用する

OpenText eDirectory では、属性のセキュリティを確保するために、次の暗号化方式がサポートされています。

- AED(Advanced Encryption Standard)
- トリプル DES
- DES(Data Encryption Standard)

1 つの暗号化属性ポリシーに含まれる各属性に対して、個別に暗号化方式を選択することもできます。たとえば、EP1 という暗号化属性ポリシーに含まれる属性 cubeno に対しては暗号化方式として AES を選択し、属性 empno に対してはトリプル DES を選択することができます。詳細については、「329 ページの 「暗号化された属性ポリシーの作成」」を参照してください。

属性の暗号化方式を変更するには、暗号化属性ポリシーを編集します。暗号化されている属性を復号化することもできます。詳細については、「329 ページの 「暗号化された属性ポリシーの編集 / 変更」」を参照してください。

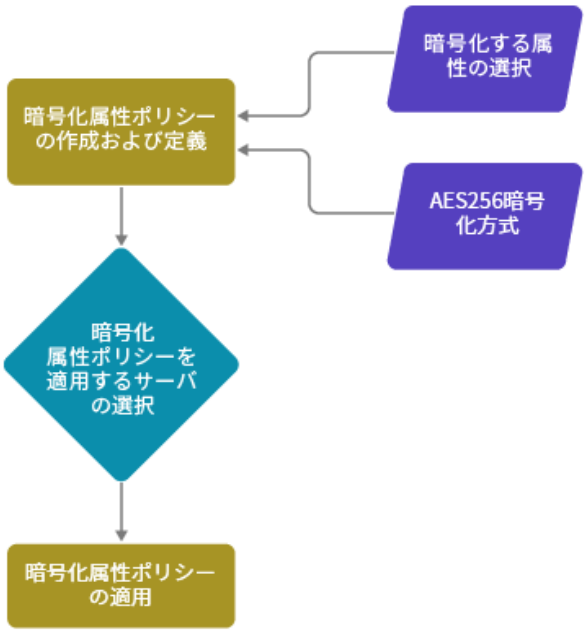
レプリカリング内の各サーバに対して、個別に暗号化方式を選択することもできます。たとえば、属性を暗号化する際に、Server1 では AES を使用し、Server2 ではトリプル DES を使用し、Server3 では暗号化方式を使用しないというような設定が可能です。

暗号化属性ポリシーを管理する

ポリシーを作成して定義し、サーバに適用することで、属性の暗号化を管理できます。

暗号化属性ポリシーを定義するには、暗号化する属性と暗号化方式を選択します。

図 11-2 属性を暗号化する



暗号化された属性ポリシーの管理は、OpenText Identity Console から行えます。このセクションでは、次のトピックについて説明します。

- 328 ページの 「OpenText Identity Console を使用した暗号化された属性ポリシーの管理」
- 330 ページの 「LDAP を使用して暗号化属性ポリシーを管理する」
- 331 ページの 「暗号化属性ポリシーをコピーする」
- 332 ページの 「パーティション操作」

OpenText Identity Console を使用した暗号化された属性ポリシーの管理

このセクションでは、次の手順について説明します。

- 329 ページの 「暗号化された属性ポリシーの作成」
- 329 ページの 「暗号化された属性ポリシーの編集 / 変更」
- 329 ページの 「暗号化属性ポリシーを削除する」

暗号化された属性が OpenText eDirectory サーバに存在する場合、OpenText Identity Console は次の方法で動作します。

1. クリアテキストまたはセキュリティで保護されたチャネルを介して、暗号化属性の読み込み、表示、修正を行うことはできません。
2. クリアテキストまたはセキュリティで保護されたチャネルで OpenText Identity Console を介して、暗号化された属性以外の属性を持つエントリの属性の読み込み、表示、修正を行うことはできません。これは、エントリ全体がブロックされていることを意味します。

暗号化された属性ポリシーの作成

1 OpenText Identity Console のホームページで、[**暗号化された属性**] タイルをクリックします。

2 [**暗号化された属性**] ページで、[**EA ポリシーの作成** +] をクリックします。

3 [**名前**] フィールドにポリシー名を入力します。

4 [コンテキスト] ( ) をクリックして、コンテナを選択します。

5 [NCP サーバ] ( ) をクリックし、サーバを選択して、[**OK**] をクリックします。

6 属性を選択します。選択した属性は、[**選択された属性**] テーブルに表示されます。

7 **作成**をクリックします。

「**ポリシーが正常に作成されました**」というメッセージが表示されます。

暗号化された属性ポリシーの編集 / 変更

1 OpenText Identity Console のホームページで、[**暗号化された属性**] タイルをクリックします。

2 [**暗号化された属性**] ページで、変更する必要があるポリシーを選択し、[オブジェクトの変更] ( ) をクリックします。

3 [**暗号化された属性ポリシーの変更**] 管理ページで必要な情報を変更して、ポリシーを編集します。

4 **完了**をクリックします。

「**ポリシーが正常に変更されました**」というメッセージが表示されます。

5 [**OK**] をクリックします。

暗号化属性ポリシーを削除する

1 OpenText Identity Console のホームページで、[**暗号化された属性**] タイルをクリックします。

2 [**暗号化された属性**] ページで、削除する必要があるポリシーを選択し、[オブジェクトの削除] ( ) をクリックします。[**OK**] をクリックして、選択した属性の削除を確認します。

LDAP を使用して暗号化属性ポリシーを管理する

重要 : 暗号化された属性の管理には、LDAP ではなく、OpenText Identity Console を使用することを強く推奨します。

このセクションでは、次の手順について説明します。

- 330 ページの 「暗号化属性ポリシーを作成して定義する」
- 331 ページの 「暗号化属性ポリシーを編集する」
- 331 ページの 「暗号化属性ポリシーを適用する」
- 331 ページの 「暗号化属性ポリシーを削除する」

注 : LDIF で暗号化する属性をマーキングする場合は、属性と方式のリストではなく、属性と方式のペアを指定する必要があります。これは暗号化属性に付随する現行の制約です。

暗号化属性ポリシーを作成して定義する

1 暗号化属性ポリシーを作成します。

たとえば、AE Policy- test-server という暗号化属性ポリシーの場合は次のようになります。

```
dn: cn=AE Policy - test-server, o=novell
changetype: add
objectClass: encryptionPolicy
```

2 作成したポリシーオブジェクトに attrEncryptionDefinition 属性を追加して、暗号化する属性をマーキングします。

たとえば、暗号化する属性の名前が CRID の場合は、次のように暗号化方式と属性名を指定します。

```
dn: cn=AE Policy - test-server, o=novell
changetype: modify
add: attrEncryptionDefinition
attrEncryptionDefinition: aes$CRID
```

注 : 属性名の指定には、その属性の NDS 名を使用します。OpenText eDirectory では多くの属性が LDAP 名と NDS 名の両方を持っています。ここでは、NDS 名を必要とする属性名を指定してください。

3 attrEncryptionRequiresSecure 属性をポリシーに追加します。

この属性の値によって、暗号化属性にアクセスする際にセキュリティ保護されたチャネルの使用を常に要求するかどうかが指定されます。この値が 0 の場合、常には要求しないことを示します。この値が 1 の場合は、常に要求することを示します。

次に例を示します。

```
dn: cn=AE Policy - test-server, o=novell
changetype: modify
add: attrEncryptionRequiresSecure
attrEncryptionRequiresSecure: 0
```

4 NCP サーバにポリシーを関連付けます。

test-server という NCP サーバの場合は次のようになります。

```
dn: cn=test-server, o=novell
changetype: modify
add: encryptionPolicyDN
encryptionPolicyDN: cn=AE Policy - test-server, o=novell
```

暗号化属性ポリシーを編集する

次の LDIF ファイルは、attrEncryptionRequireSecure 属性の値を変更することで暗号化属性ポリシーを編集する方法を示しています。

```
dn: cn=AE Policy - test-server, o=novell
changetype: modify
replace: attrEncryptionRequiresSecure
attrEncrytionRequiresSecure: 1
```

暗号化属性ポリシーを適用する

次の LDIF ファイルは、AE Policy-test-server という暗号化属性ポリシーを test-server というサーバに適用する方法を示しています。

```
dn: cn=test-server, o=novell
changetype: modify
add: encryptionPolicyDN
encryptionPolicyDN: cn=AE Policy - test-server, o=novell
```

暗号化属性ポリシーを削除する

次の LDIF ファイルは、暗号化属性ポリシーを削除する方法を示しています。

```
dn: cn=AE Policy - test-server, o=novell
changetype: delete
```

注 : LDAP を使用して暗号化属性を管理する方法の詳細については、386 ページの 「Linux での LDAP ツールの使用」および 175 ページの 「OpenText インポート / エクスポート変換ユーティリティ」を参照してください。

暗号化属性ポリシーをコピーする

暗号化属性ポリシーをコピーして多数のサーバにまったく同じ設定内容を反映させることができます。ポリシーは、オブジェクトとして OpenText eDirectory に保存されます。

OpenText Identity Console を使用してポリシーオブジェクトをコピーする手順については、111 ページの 「オブジェクトをコピーする」を参照してください。

パーティション操作

2 つのパーティションをマージすると、ペアレントのポリシーがマージ後のパーティションで保持されます。パーティションを分割すると、ペアレントのポリシーがチャイルドパーティションに継承されます。

推奨 : OpenText eDirectory は、暗号化のマークが付いているべきでない、いくつかの属性を内部操作用に保管しています。これらの属性に暗号化のマークが付けられると、OpenText eDirectory の一部の機能が壊れるか、期待通りの動作をしなくなる可能性があります。

暗号化のマークを付けてはならない属性は次のとおりです。

- federationBoundaryType
- ボリューム
- ACL
- federationBoundary
- member
- federationControl
- federationSearchPath
- encryptionPolicyDN
- indexDefinition
- dgIdentity
- dgAllowUnknown
- agTimeout
- Host Server
- hostResourcePath
- ndsPredicateState
- ndsStatusExternalReference
- ndsStausLimber
- ndsStatusSchema

上記のリストはすべてを網羅しているわけではありませんが、ここにリストされた属性と同様の属性には、暗号化のマークを付けないでください。

暗号化属性にアクセスする

属性を暗号化すると、暗号化属性へのアクセスも保護されます。これは、OpenText eDirectory が、暗号化された属性へのアクセスを、セキュアチャネル (LDAP セキュアチャネル、NCP セキュアチャネルなど) 経由のみに制限できるからです。ただし、セキュア NCP 接続を作成するために使用する DClient アプリケーションは一般利用者向けには提供されていないので、セキュア NCP 接続を設定して使用できるのは、OpenText の内部顧客に限られます。

また、バックアップ (ndsbackup) ユーティリティを使用して、暗号化属性をバックアップすることもできます。

デフォルトでは、セキュリティ保護されたチャネルからしか暗号化属性にアクセスできません。

ただし、クライアントがクリアテキストで暗号化属性にアクセスできるようにするには、[常にセキュアチャネルが必要] オプションを無効にする必要があります。詳細については、333 ページの 「クリアテキストチャネルから暗号化属性へのアクセスを有効 / 無効にする」を参照してください。

注 : 暗号化された属性を持つオブジェクトに対してセキュリティ保護されていない検索を実行した場合、暗号化されていない属性は、明示的に指定しない限り返されません。

クリアテキストチャネルから暗号化属性へのアクセスを有効 / 無効にする

OpenText Identity Console または LDAP のいずれかを使用して、[常にセキュアチャネルが必要] オプション (つまり、attrEncryptionRequireSecure 属性) を有効または無効にすることで、クリアテキストチャネルから暗号化された属性へのアクセスを有効または無効にすることができます。

このセクションでは、次のことを説明します。

- 333 ページの「OpenText Identity Console を使用してクリアテキストチャネルから暗号化された属性へのアクセスを有効 / 無効にする」
- 333 ページの 「LDAP を使用してクリアテキストチャネルから暗号化属性へのアクセスを有効 / 無効にする」

OpenText Identity Console を使用してクリアテキストチャネルから暗号化された属性へのアクセスを有効 / 無効にする

OpenText Identity Console を使用してクリアテキストチャネルから暗号化された属性へのアクセスを有効または無効にするには、次の作業を行う際に、[暗号化された属性ポリシー] 管理ページで [常にセキュアチャネルが必要] を有効または無効にします。

- 暗号化属性ポリシーを作成して定義する。
- 暗号化属性ポリシーを編集する。

LDAP を使用してクリアテキストチャネルから暗号化属性へのアクセスを有効 / 無効にする

LDAP を使用してクリアテキストチャネルから暗号化属性へのアクセスを有効にするには、次の属性を暗号化属性ポリシーに追加します。

attrEncryptionRequiresSecure

この属性を 0 に設定すると、セキュリティ保護されたチャネルを必ずしも使用する必要はなくなります。つまり、クリアテキストチャネルから暗号化属性にアクセスできます。この属性を 1 に設定すると、セキュリティ保護されたチャネルの使用が常に要求されるようになります。つまり、セキュリティ保護されたチャネルからのみ暗号化属性にアクセスできます。

詳細については、「330 ページのステップ 3」を参照してください。

暗号化属性を表示する

暗号化されている属性が表示されるかどうかは、[常にセキュアチャネルが必要] オプションが有効になっているかどうかによって決定されます。つまり、暗号化属性にアクセスする際にセキュリティ保護されたチャネルを使用するかどうかによって決まります。

- 334 ページの 「OpenText Identity Console を使用して暗号化された属性を表示する」
- 334 ページの 「DSBrowse を使用して暗号化属性を表示する」
- 334 ページの 「SNMP トラップ」

OpenText Identity Console を使用して暗号化された属性を表示する

[常にセキュアチャネルが必要] が有効になっている場合は、暗号化属性を表示できません。エラー「–6089」が返されます。これは、暗号化属性にアクセスする際にセキュリティ保護されたチャネルを使用する必要があることを示しています。

[常にセキュアチャネルが必要] が無効になっている場合は、OpenText Identity Console で暗号化された属性値を表示できます。

詳細については、262 ページの 「ツリー内のオブジェクトの参照」を参照してください。

DSBrowse を使用して暗号化属性を表示する

[常にセキュアチャネルが必要] オプションが有効になっている場合、つまり、暗号化属性にアクセスする際にセキュリティ保護されたチャネルの使用が常に要求される場合は、暗号化されたエントリの属性を表示することはできません。ただし、暗号化されていないエントリの属性は表示できます。

SNMP トラップ

暗号化属性にアクセスする際にセキュリティ保護されたチャネルの使用を常に要求するように指定している場合は、NDS® 値イベントがブロックされます。値イベントに関連するトラップの値データは NULL になり、結果は -6089 に設定されます。これは、暗号化属性の値を取得するために、セキュリティ保護されたチャネルを使用する必要があることを示します。次のトラップの値データは NULL になります。

- ndsAddValue
- ndsDeleteValue
- ndsDeleteAttribute

バックアップデータを暗号化 / 復号化する

暗号化属性を含むサーバ上のデータをバックアップする際には、バックアップデータを暗号化または復号化するためのパスワードを入力するよう求められます。これには、バックアップユーティリティの -E オプションを使用して簡単に対応できます。詳細については、ndsbackup のマニュアルページを参照してください。

データのバックアップの詳細については、「459 ページの第 15 章「OpenText eDirectory のバックアップと復元」」を参照してください。

暗号化属性を含む DIB ファイルセットのクローンを作成する

暗号化された属性を含む OpenText eDirectory データベースのクローンを作成すると、DIB ファイルセットのクローンでもそれらの属性値が暗号化されます。DIB ファイルセットのクローンに含まれる値を暗号化するには、OpenText eDirectory で使用されるキーを保護するためのパスワードを設定する必要があります。DIB ファイルセットのクローンを他のサーバに配置する際には、このパスワードの入力を求められます。

詳細については、269 ページの 「DIB セットのクローンの使用事例」を参照してください。

レプリカリングに OpenText eDirectory サーバを追加する

OpenText eDirectory サーバは、レプリカが格納されたいずれかのサーバまたはすべてのサーバ上で属性が暗号化されているかどうか、あるいは［常にセキュアチャネルが必要］が有効になっているかどうかに関わらず、レプリカリングに追加できます。

OpenText eDirectory サーバをレプリカリングに追加する際の詳細については、164 ページの「レプリカを追加する」を参照してください。

後方互換性

暗号化された属性にアクセスするには、Identity Console、SNMP、DirXML®、NSureAudit などのすべての OpenText eDirectory ユーティリティを、セキュリティ保護された NCP™ に変更する必要があります。変更しない場合は、暗号化属性にアクセスする際にセキュリティ保護されたチャネルの使用を要求しないように指定する必要があります。詳細については、「333 ページの 「クリアテキストチャネルから暗号化属性へのアクセスを有効 / 無効にする」」を参照してください。

暗号化属性に移行する

OpenText eDirectory をアップグレードする際は、暗号化された属性ポリシーを作成して定義することで、既存の属性を暗号化できます。詳細については、328 ページの 「暗号化属性ポリシーを管理する」を参照してください。

暗号化属性のレプリカを作成する

デフォルトでは、サーバに暗号化属性が存在している場合でも、暗号化複製は無効になっています。暗号化属性を安全に複製するには、暗号化複製を有効にする必要があります。暗号化複製の設定については、「336 ページの 「暗号化レプリケーション」」を参照してください。

暗号化レプリケーション

OpenText eDirectory では、OpenText eDirectory サーバ間で転送されるデータを暗号化できます。データがクリアテキスト形式で転送されなくなるため、複製時のセキュリティを強化できます。

注 : OpenText eDirectory 9.2.7 リリースでは、暗号化された複製のサポートは非推奨になりました。

***図 11-3** 暗号化レプリケーション*

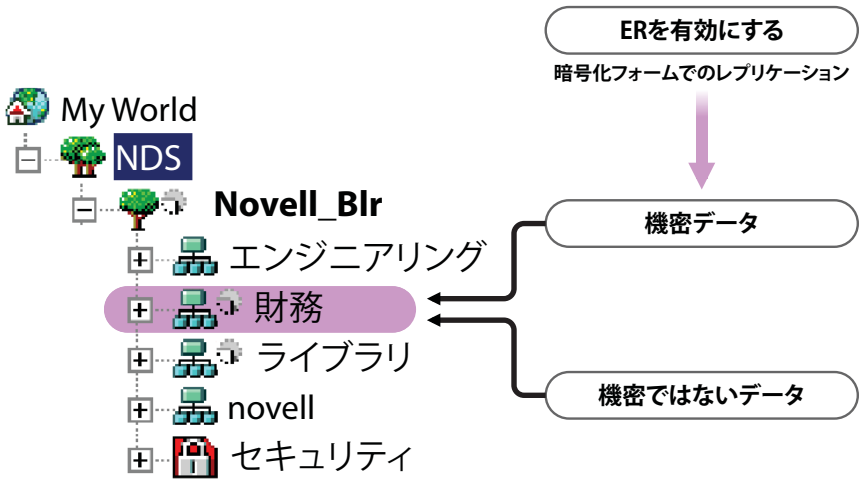


図 11-3 では、「finance」と「library」がツリー内のパーティションに相当します。「finance」には、レプリケーションの際に暗号化しなければならない機密データが含まれる可能性があります。その場合は、パーティション「finance」に対して暗号化複製を有効にすることができます。「library」のように重要データが含まれていないパーティションについては、暗号化複製を有効にする必要はありません。

重要 : パーティションに対して暗号化複製を有効にすると、複製処理の速度が低下する可能性があります。暗号化された複製は、OpenText Identity Console を使用して有効または無効にできます。

この節では、次のトピックについて説明します。

- 337 ページの 「暗号化複製の必要性」
- 337 ページの 「暗号化複製を有効にする」
- 341 ページの 「新しいレプリカをレプリカリングに追加する」

- 342 ページの 「同期と暗号化複製」
- 343 ページの 「暗号化複製ステータスを表示する」

暗号化複製の必要性

OpenText eDirectory 8.8 以前では、複製中のデータはクリアテキストでネットワークに転送されました。レプリカが地理的に離れており、インターネット経由で接続されている場合は特に、ネットワーク上で機密データを暗号化して保護する必要がありました。

この機能は、次のような状況で使用できます。

- ディレクトリサーバが WAN やインターネットを介して地理的に複数の場所にわたって広がっており、ネットワーク上で重要データを暗号化する必要がある。
- ツリーのパーティションの一部だけを保護する場合は、複製のために暗号化する重要データを保持しているパーティションを選択的に指定できます。
- 重要データを含むパーティションの特定のレプリカ間で暗号化複製が必要な場合。
- 現在のネットワーク環境が安全ではないと思われる場合は、複製中に重要データを保護することもできます。

暗号化複製を有効にする

暗号化複製を有効にするには、暗号化複製を有効にするようにパーティションを設定する必要があります。設定はパーティションのルートオブジェクトに保存されます。

暗号化複製は、パーティションレベルでもレプリカレベルでも有効にすることができます。

レプリカレベルの設定は、パーティションレベルの設定よりも優先されます。つまり、次のようになります。

- 暗号化複製がパーティションレベルで有効になっていて、特定のレプリカで無効になっている場合、それらのレプリカの間の複製はクリアテキスト形式で行われます。
- パーティションレベルで暗号化複製が無効になっていて、特定のレプリカで有効になっている場合、それらのレプリカの間の複製は暗号化された形式で行われます。

表 11-1 *パーティションレベルの暗号化複製の設定を上書きする*

パーティションレベル	レプリカレベル	複製
有効	無効	暗号化されない
無効	有効	暗号化される

このセクションでは、次の手順について説明します。

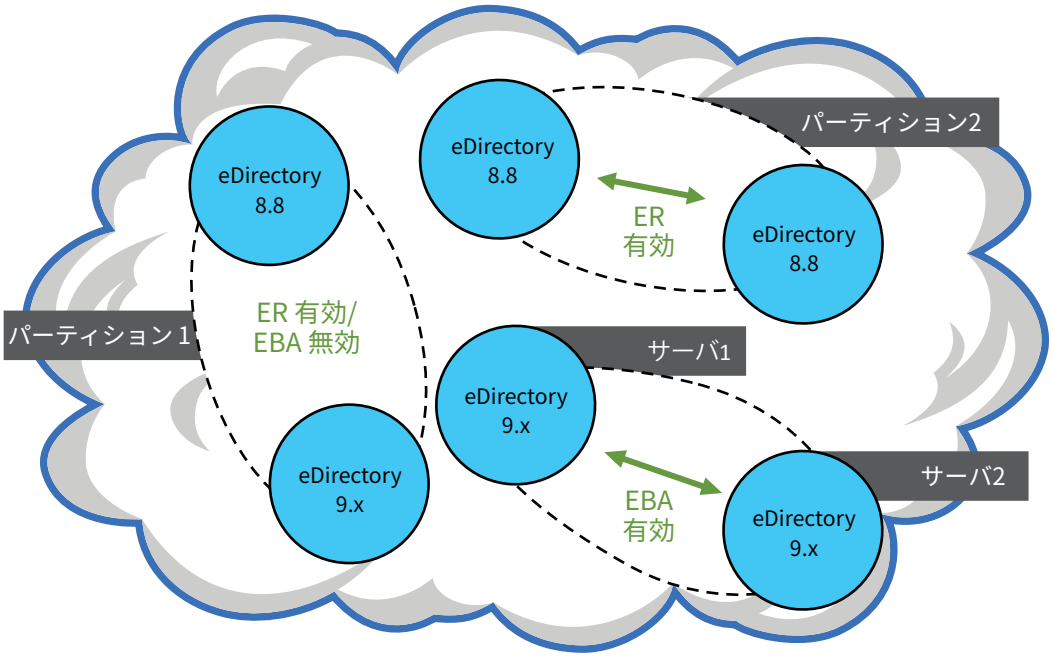
- 338 ページの 「パーティションレベルで暗号化複製を有効にする」
- 340 ページの 「レプリカレベルで暗号化複製を有効にする」

パーティションレベルで暗号化複製を有効にする

パーティションレベルで暗号化複製を有効にすると、そのパーティションをホストしているすべてのレプリカの間で行われる複製が暗号化されます。たとえば、パーティション P1 のレプリカとして、R1、R2、R3、および R4 があるとします。その場合は、これらのレプリカの間のすべての複製 (インバウンドとアウトバウンドの両方) を暗号化できます。

パーティションレベルで暗号化された複製を有効にするには、そのパーティションをホストしているすべてのサーバで OpenText eDirectory 8.8 以降が実行されている必要があります。

図 11-4 暗号化レプリケーション



レプリカレベルで暗号化複製が設定されている場合は、レプリカレベルの設定がパーティションレベルの設定よりも優先されます。「337 ページの 表 11-1」を参照してください。

下位互換性は、暗号化複製がパーティションレベルで有効になっているかどうかに依存します。詳細については、「341 ページの 「新しいレプリカをレプリカリングに追加する」」を参照してください。

パーティションレベルで暗号化された複製を有効にするには、次のセクションで説明するように、OpenText Identity Console または LDAP を使用します。

- 339 ページの「OpenText Identity Console を使用してパーティションレベルで暗号化された複製を有効にする」
- 339 ページの 「LDAP を使用してパーティションレベルで暗号化複製を有効にする」
- 341 ページの 「パーティション操作」

OpenText Identity Console を使用してパーティションレベルで暗号化された複製を有効にする

1 OpenText Identity Console のホームページで、[**暗号化された複製**] タイルをクリックします。

2 [**暗号化された複製**] ページで、[**パーティションの検索** ] をクリックします。

3 ツリーまたはコンテナを選択します。

4 [**複製の暗号化を有効にする**] チェックボックスをオンにし、[**完了**] をクリックします。

注 : パーティションレベルで暗号化された複製を無効にするには、[**複製の暗号化を有効にする**] チェックボックスをオフにします。

---

5 [**レプリケーションの暗号化が正常に有効になりました**] というメッセージが表示されます。

[**OK**] をクリックします。

[暗号化された複製] ページでパーティション全体に対して複製の暗号化を有効にした場合でも、特定のレプリカに対して暗号化された複製を無効にできます。暗号化複製が無効になっているレプリカは、暗号化された形式のデータを送受信しません。パーティション全体の暗号化を無効にするには、[**複製の暗号化を有効にする**] チェックボックスをオフにします。

LDAP を使用してパーティションレベルで暗号化複製を有効にする

重要 : OpenText Identity Console を使用して暗号化された複製を有効にすることを強くお勧めします。

複製を暗号化するには、属性 dsEncryptedReplicationConfig を使用します。構文は次のとおりです。

```
enable/disable flag#destination replica number#source replica number
```

次のいずれかのフラグに置き換えます。

- 0: 暗号化複製が無効になります。
- 1: 暗号化複製が有効になります。

ソースレプリカ番号とターゲットレプリカ番号は、パーティションのソースレプリカ番号とターゲットレプリカ番号です。ソースレプリカ番号とターゲットレプリカ番号はどちらを先に指定しても構いません。レプリカ A から B への複製が暗号化されている場合は、B から A への複製も暗号化されます。

注 : パーティションレベルのソースレプリカ番号とターゲットレプリカ番号を 0 にして、フラグを 1 に設定した場合は、すべてのレプリカで暗号化複製が有効になります。

パーティションレベルで暗号化複製を有効にするには、dsEncryptedReplicationConfig 属性の値を 1#0#0 に設定します。

次に、パーティションレベルで暗号化複製を有効にするための LDIF ファイルの例を示します。

```
dn: o=ou
changetype:modify
replace: dsEncryptedReplicationConfig
dsEncryptedReplicationConfig:1#0#0
```

レプリカレベルの設定は、パーティションレベルの設定よりも優先されます。詳細については、「340 ページの 「LDAP を使用してレプリカレベルで暗号化複製を有効にする」」を参照してください。

レプリカレベルで暗号化複製を有効にする

レプリカレベルで暗号化複製を有効にすると、特定のレプリカの間の複製が暗号化されます。指定したレプリカの間で行われるアウトバウンドおよびインバウンドの複製が暗号化されます。

たとえば、パーティション P1 のレプリカとして、R1、R2、R3、および R4 があるとします。この場合、レプリカ R1 と R2 の間、または R2 と R4 の間の複製を暗号化できます。

1 つのレプリカで暗号化複製を有効にした場合は、次のような複製が行われます。

- サーバからこのレプリカへのインバウンド同期が暗号化されます。
- このレプリカから別のサーバへのアウトバウンド同期が暗号化されます。

暗号化された複製が有効になっているレプリカは、OpenText eDirectory 8.8 以降のサーバ上に配置されている必要があります。レプリカリング内にあるレプリカのうち、暗号化された複製が有効になっていない残りのレプリカは、eDirectory 8.8 より古いバージョンの OpenText eDirectory サーバ上に配置しても構いません。

レプリカレベルで暗号化された複製を無効にするには、OpenText Identity Console の暗号化された複製の環境設定ページで、該当するレプリカに対し［**リンクを暗号化する**］を無効にします。

レプリカレベルで暗号化された複製を有効にするには、次のセクションで説明するように、LDAP を使用します。

- 340 ページの 「LDAP を使用してレプリカレベルで暗号化複製を有効にする」

LDAP を使用してレプリカレベルで暗号化複製を有効にする

重要 : OpenText Identity Console を使用して暗号化された複製を有効にすることを強くお勧めします。

複製を暗号化するには、属性 dsEncryptedReplicationConfig を使用する必要があります。構文は次のとおりです。

```
enable/disable flag#destination replica number#source replica number
```

この構文の詳細については、339 ページの 「LDAP を使用してパーティションレベルで暗号化複製を有効にする」を参照してください。

この構文でレプリカ番号を指定すると、指定したレプリカの間の複製が暗号化されます。次に、この構文の例を示します。

- 1#0#1: レプリカ番号 1 と、パーティション内の他のすべてのレプリカの間で、暗号化複製が有効になります。
- 0#3#1: レプリカ番号 3 と #1 の間で暗号化複製が無効になります。
- 0#1#1: レプリカ番号 1 で暗号化複製が無効になります。

次に、レプリカ番号 1 と 3 の間で暗号化複製を無効にするための LDIF ファイルの例を示します。

```
dn: o=ou
changetype: modify
replace: dsEncryptedReplicationConfig
dsEncryptedReplicationConfig: 0#3#1
```

パーティション操作

パーティションを分割すると、ペアレントパーティションの暗号化複製の設定がチャイルドパーティションに継承されます。パーティションをマージすると、ペアレントパーティションの暗号化複製の設定がマージ後のパーティションで保持されます。

新しいレプリカをレプリカリングに追加する

レプリカリングに新しいレプリカを追加する場合は、パーティションレベルおよびレプリカレベルで暗号化複製が有効になっているかどうかによって、それぞれ異なる影響を受けます。

レプリカをレプリカリングに追加する場合の詳細については、「164 ページの 「レプリカの管理」」を参照してください。

前述のどちらのレベルでも、次のセクションで説明するように、レプリカリングに追加する OpenText eDirectory サーバのバージョンによって、複数のシナリオが考えられます。

- 341 ページの 「パーティションレベルで暗号化複製を有効にする」
- 342 ページの 「レプリカレベルで暗号化複製を有効にする」
- 342 ページの 「追加するサーバで暗号化複製を有効にする」

パーティションレベルで暗号化複製を有効にする

追加しようとしている OpenText eDirectory サーバのバージョンによって、シナリオは異なります。

場面	データの暗号化
EBA なし、暗号化された複製が無効の、OpenText eDirectory 9.1 以降のサーバを追加する	データは平文で受け渡されます。
暗号化された複製あり、EBA なしの、OpenText eDirectory 9.1 以降のサーバを追加する	OpenText eDirectory は、暗号化された複製ポリシーに基づいてデータを暗号化します。
EBA ありの OpenText eDirectory 9.1 以降のサーバを追加する	EBA ベースの暗号化が、暗号化複製より優先されます。

レプリカレベルで暗号化複製を有効にする

ソースレプリカと特定のターゲットレプリカの間で暗号化された複製が有効になっている場合は、OpenText eDirectory 8.8 以降のサーバをレプリカリングに追加できます。

ソースレプリカと、レプリカリング内の他のすべてのレプリカの間で暗号化複製が有効になっている場合、このシナリオは当てはまりません。その場合は、パーティションレベルで暗号化複製が有効または無効になっているレプリカリングにレプリカを追加することと同じになります。詳細については、「341 ページの 「パーティションレベルで暗号化複製を有効にする」」を参照してください。

追加するサーバで暗号化複製を有効にする

追加しようとしているサーバのプラットフォームが Linux の場合は、ndsconfig -E オプションを使用してそのサーバでの暗号化複製を有効にできます。詳細については、ndsconfig のマニュアルページを参照してください。

追加するサーバのプラットフォームが Windows の場合は、インストールページで［複製の暗号化を有効にする］オプションを選択します。

追加しようとしているサーバのプラットフォームが Linux 以外の場合は、OpenText Identity Console または LDAP を使用して暗号化された複製を有効にできます。詳細については、「337 ページの 「暗号化複製を有効にする」」を参照してください。

同期と暗号化複製

特定のレプリカで暗号化複製を有効にして、設定の変更を他のサーバと同期しない場合、レプリカ間の複製は暗号化された形式で行われます。暗号化複製の設定変更が同期されていないレプリカでは、引き続きクリアテキスト形式で複製が行われます。

それまで暗号化複製の設定がレプリカ間で同期されていなかった場合でも、レプリカ間の複製は暗号化された形式で行われます。

暗号化複製ステータスを表示する

次の手順に従って、OpenText iMonitor を介して暗号化された複製ステータスを表示できます。

1 OpenText iMonitor で、アシスタントフレームの［**エージェント同期**］をクリックします。

2 表示するパーティションの［**レプリカ同期**］をクリックします。

レプリカステータス情報が表示されます。現在接続されているレプリカからのリンクが暗号化されているかどうかが［**Encryption Status**］フィールドに表示されます。

基本的に、暗号化複製 (ER) には次の 3 つのシナリオがあります。

- **パーティションレベルで ER が有効 :** 接続先のレプリカの［**暗号化状態**］は［使用可能］として示されます。

  どのレプリカに接続しているかを確認するには、レプリカフレームでハイパーリンクが付いていないレプリカを探します。そのレプリカが、現在接続しているレプリカです。他のレプリカをブラウズすると、そのレプリカの［**暗号化状態**］も［使用可能］として示されます。

- **レプリカレベルで ER が有効 :** 特定のレプリカからすべてのレプリカへの (つまり 1 つからすべてへの)ER が有効になっている場合です。この場合は、そのレプリカに接続すると、レプリカの［**暗号化状態**］が［使用可能］として示されます。

- **一部のレプリカで ER が有効 / 無効 :** 一部のレプリカで ER が有効 / 無効 – パーティション全体では ER が有効になっているが、一部のサーバでは無効になっている場合、またはその逆の場合です。

  たとえば、3 つのレプリカを含むパーティション A で ER が有効に設定されていて、レプリカ 1 とレプリカ 3 の間の ER が無効に設定されている場合、レプリカ 1 に接続すると、［**暗号化状態**］は次のように表示されます。

  Server 1 使用可能

  Server 2

  Server 3 使用不可

  これは、Server 1 からレプリカリング内のすべてのサーバへの ER は有効であるが、Server 1 から Server 3 への ER は管理者によって無効にされていることを意味します。

データを暗号化するときにデータの完全な安全性を確保する

データを暗号化するときには、まず、次の基本的な規則を守ることが重要です。

最終的に暗号化される情報をハードディスク (またはその他の媒体) にクリアテキストの形式で書き込まないこと。

既存のクリアテキストデータを暗号化対象としてマークすれば、そのデータは暗号化されますが、既存のクリアテキストデータは、DIB が存在するハードディスクのどこかに残る可能性があります。

次の操作を実行しようとすると、データベース内のどこかのブロックにデータのクリアテキスト部分が残ります。

- 既存のクリアテキストデータを暗号化対象としてマークする
- 暗号化属性の暗号化方式を変更する

以下のセクションでは、データを暗号化するときのシナリオを想定し、暗号化データの完全な安全性を確保する手順を説明します。

- 344 ページの 「完全に新しい設定でデータを暗号化する」
- 345 ページの 「既存の設定でデータを暗号化する」
- 346 ページの 「結論」

完全に新しい設定でデータを暗号化する

新しい設定では、オペレーティングシステムをインストールしただけの状態で OpenText eDirectory をインストールします。したがって、DIB が存在するハードディスク上にクリアテキストデータが存在しないことが保証されます。

OpenText eDirectory 内の暗号化データの完全な安全性を確保するには、次の手順を実行します。

1 どの属性をどの方式で暗号化するかをあらかじめ決めておきます。

つまり、データをクリアテキスト形式で OpenText eDirectory にアップロードする前に、暗号化する属性を決めておく必要があります。

警告 : いったんデータをクリアテキスト形式で OpenText eDirectory にロードしたら、属性を暗号化対象としてマークしないでください。そうすることもできますが、その場合はセキュリティの問題が発生します。

---

2 OpenText eDirectory を設定し、属性に適用する暗号化スキームを設定します。

3 既存のデータを新しいサーバにロードします。

LDIF ファイルからバルクロードすることと他のサーバから複製するという 2 つがよくあるシナリオです。バルクロードするときは、クリアテキストの LDIF ファイルを DIB が存在するハードディスクにコピーしないでください。

注 : クリアテキストデータをディスクに書き込んではいけないという規則を思い出してください。

---

4 既存のすべてのクリアテキストデータを破壊します。

クリアテキストデータが格納されているディスク (またはその他の媒体) を完全に消去してください。消去する対象には、サーバのバルクロードに使われたクリアテキストの LDIF ファイル、複製に使われた他のサーバ、古いバックアップが残っているテープなどが含まれます。

既存の設定でデータを暗号化する

このシナリオには次のような状況があります。

- 345 ページの 「既存のクリアテキストデータを暗号化データに変換する」
- 346 ページの 「データの暗号化方式を変更する」

既存のクリアテキストデータを暗号化データに変換する

クリアテキストデータを暗号化対象としてマークし、以下の方法でデータの安全性を確保できます。

- 345 ページの 「複製を利用する方法」
- 345 ページの 「バックアップおよび復元を利用する方法」

複製を利用する方法

1 新しいサーバで次のように暗号化を設定します。

1a どの属性をどの方式で暗号化するかをあらかじめ決めておきます。

つまり、データをクリアテキスト形式で OpenText eDirectory にアップロードする前に、暗号化する属性を決めておく必要があります。

警告 : いったんデータをクリアテキスト形式で OpenText eDirectory にロードしたら、属性を暗号化対象としてマークしないでください。そうすることもできますが、その場合はセキュリティの問題が発生します。

1b 新たにフォーマットされ、パーティションが作成されたディスクで、クリアインストール (場合によっては OS も含めて) を行います。

これは、ディスクにクリアテキストデータが存在する可能性を排除するためです。つまり、以前クリアテキストデータが保存されていた既存のコンピュータに OpenText eDirectory を再インストールすることはできません。ディスクからデータのすべての痕跡を完全に消去する必要があります。OpenText eDirectory をインストールする前に、ディスクで安全な消去用ソフトウェアを使用する、磁気バルクイレーサーを使用するなど、データを徹底的に破壊する操作を行います。

1c OpenText eDirectory を設定し、属性に適用する暗号化スキームを設定します。

2 暗号化する既存のデータが存在するレプリカリングにそのサーバを移動し、複製を実行した後、古いサーバをオフラインにします。

3 既存のすべてのクリアテキストデータを破壊します。

クリアテキストデータが格納されているディスク (またはその他の媒体) を完全に消去してください。消去する対象には、サーバのバルクロードに使われたクリアテキストの LDIF ファイル、複製に使われた他のサーバ、古いバックアップが残っているテープなどが含まれます。

バックアップおよび復元を利用する方法

1 新しいサーバで次のように暗号化を設定します。

1a どの属性をどの方式で暗号化するかをあらかじめ決めておきます。

つまり、データをクリアテキスト形式で OpenText eDirectory にアップロードする前に、暗号化する属性を決めておく必要があります。

警告 : いったんデータをクリアテキスト形式で OpenText eDirectory にロードしたら、属性を暗号化対象としてマークしないでください。そうすることもできますが、その場合は Note A で説明しているセキュリティの問題が発生します。

1b 新たにフォーマットされ、パーティション化されたディスクで、クリアインストール (場合によってはオペレーティングシステムも含む) を行います。

これは、ディスクにクリアテキストデータが存在する可能性を排除するためです。つまり、以前クリアテキストデータが保存されていた既存のコンピュータに OpenText eDirectory を再インストールすることはできません。ディスクからデータのすべての痕跡を完全に消去する必要があります。OpenText eDirectory をインストールする前に、ディスクで安全な消去用ソフトウェアを使用する、磁気バルクイレーサーを使用するなど、データを徹底的に破壊する操作を行います。

1c OpenText eDirectory を設定し、属性に適用する暗号化スキームを設定します。

2. 新しいサーバで、バックアップされた DIB(既存のクリアテキストデータが格納されている) を復元します。DIB をバックアップするには、DIB セットのクローンまたはホットバックアップを使用できます。
3. 既存のすべてのクリアテキストデータを破壊します。

   クリアテキストデータが格納されているディスク (またはその他の媒体) を完全に消去してください。消去する対象には、サーバのバルクロードに使われたクリアテキストの LDIF ファイル、複製に使われた他のサーバ、古いバックアップが残っているテープなどが含まれます。

データの暗号化方式を変更する

バックアップと復元を利用してこの操作を行うには、次の手順を実行します。

1. 属性の暗号化アルゴリズムを変更します。
2. DIB のバックアップをとります。DIB をバックアップするには、DIB セットのクローンまたはホットバックアップを使用できます。
3. バックアップされた DIB を新しいサーバ上で復元し、古いサーバを削除します。
4. 古いサーバ上にあるすべての既存のクリアテキストデータを破壊します。そうすることで、古い暗号化方式で暗号化されたデータの断片がハードディスクから一掃されます。

   クリアテキストデータが格納されているディスク (またはその他の媒体) を完全に消去してください。消去する対象には、サーバのバルクロードに使われたクリアテキストの LDIF ファイル、複製に使われた他のサーバ、古いバックアップが残っているテープなどが含まれます。

結論

ここで説明したシナリオ以外にも、この問題が発生する状況はあり得ます。「最終的に暗号化される情報をハードディスク (またはその他の媒体) にクリアテキストの形式で書き込まない」という規則を守っている限り、暗号化データの完全な安全性が確保されます。

12 OpenText eDirectory データベースの修復

DSRepair ユーティリティを使用して、OpenText eDirectory ツリーのデータベースを保守および修復することができます。このユーティリティでは、次の操作を実行できます。

- 不正なレコード、スキーマの不一致、不正なサーバアドレス、外部参照など、OpenText eDirectory の問題の修正
- OpenText eDirectory スキーマへの詳細な変更
- データベースの終了やユーザの介入を伴わない、自動的なデータベースの構造チェック
- データベースのオペレーショナルインデックスの確認
- 空のレコードを破棄することによる空き領域の増量。
- ローカルデータベースを修復します。
- レプリカ、レプリカリングおよびサーバオブジェクトの修復
- 同期エラーに対する、各ローカルパーティションの各サーバの分析
- ローカルデータベース内のオブジェクトの検出と同期

すべての OpenText eDirectory データベースの問題が致命的であるというわけではなく、OpenText eDirectory によって処理を続行できる問題もあります。ただしデータベースが破損すると、ローカルデータベースを開くことができないことを知らせるメッセージがコンソールに表示されます。この場合、修復を実行するか OpenText サポート部門に連絡します。

OpenText eDirectory に問題が発生したか、OpenText サポート部門から指示されたのでない限り、修復操作を実行することはお勧めできません。ただし、修復ユーティリティや他の OpenText ユーティリティ (OpenText iMonitor など) の診断機能を使用することはお勧めします。詳細については、243 ページの第 8 章「OpenText の監視」を参照してください。

修復ウィザードのリスト :

ウィザード	説明
基本修復ウィザード	標準修復、ローカルデータベースの修復、または単一オブジェクトの修復を実行できます。また、外部参照をチェックして不明なリーフオブジェクトを削除できます。
ログファイルウィザード	修復ログファイルを開いて、ログファイルオプションを設定できます。
OpenText iMonitor による修復	OpenText iMonitor を開いて、このプログラムの修復オプションを使用できます。

ウィザード	説明
レプリカ修復ウィザード	すべてまたは選択したレプリカの修復、タイムスタンプの修復と新しいエポックの宣言、現在のサーバを新しいマスタレプリカに設定、および必要に応じて選択したレプリカの削除などを実行できます。
レプリカリングの修復ウィザード	すべてまたは選択したレプリカリングの修復、リング内のすべてのサーバにすべてのオブジェクトを送信、選択したレプリカでマスタレプリカのすべてのオブジェクトを受信、および必要に応じてレプリカリングから現在のサーバの削除などを実行できます。
スキーマの保守ウィザード	ツリーからスキーマを要求、ローカルスキーマのリセット、新規スキーマエポックの宣言、オプションスキーマ拡張機能の実行、リモートスキーマのインポート、およびスキーマの更新などを実行できます。
サーバの修復ウィザード	すべてのネットワークアドレスを修復するか、サーバのネットワークアドレスのみを修復することができます。
同期修復ウィザード	現在のサーバで選択したレプリカの同期、現在のサーバの同期ステータスのレポート、すべてのサーバの同期ステータスのレポート、時刻同期の実行、および即時同期のスケジュールが実行できます。

ウィザードは、次のような操作のときに役立ちます。

- 348 ページの 「基本修復操作の実行」
- 352 ページの 「修復ログファイルの表示と設定」
- 352 ページの 「OpenText iMonitor での修復の実行」
- 353 ページの 「レプリカの修復」
- 355 ページの 「レプリカリングを修復する」
- 358 ページの 「スキーマの保守」
- 360 ページの 「サーバのネットワークアドレスの修復」
- 362 ページの 「同期化操作の実行」
- 364 ページの 「DSRepair オプション」
- 369 ページの 「クライアントを使用したデータベースの修復」
- 372 ページの 「グラフィカルな DS 修復ユーティリティ」

基本修復操作の実行

基本修復ウィザードでは、標準修復、ローカルデータベースの修復、または単一オブジェクトの修復を実行できます。また、外部参照をチェックして不明なリーフオブジェクトを削除できます。

- 349 ページの 「標準修復を実行する」
- 350 ページの 「ローカルデータベースの修復の実行」

- 351 ページの 「外部参照のチェック」
- 351 ページの 「単一オブジェクトの修復」
- 351 ページの 「不明なリーフオブジェクトの削除」

標準修復を実行する

標準修復では、指定されたサーバの OpenText eDirectory データベースファイルに致命的な OpenText eDirectory エラーがないかチェックし、修復します。このオプションは、実行されるたびに 8 つの主要な操作を実行します。これらの操作には、管理者が関与する必要はありません。これらの操作の中には、実行中にデータベースをロックするものがあります。標準修復では、ローカルデータベースファイルのセットが一時的に作成され、修復操作はこれらのファイルに対して実行されます。つまり、重大な問題が発生したとしても、オリジナルのファイルは無事です。

特定の問題をトラブルシューティングし、それを解決することは標準修復を実行することよりはるかに優れています。標準修復を実行すると、データベースファイルが現在使用している 2 倍の容量の空きディスクが必要になる場合があります。詳細については、350 ページの 「ローカルデータベースの修復の実行」を参照してください。

OpenText eDirectory が使用するオペレーショナルインデックスの再構築は、ローカルデータベースがロックされているときのみ可能です。

次の表に、標準修復の実行中に行われる操作について示します。

操作	データベースのロック	説明
データベース構造およびインデックスのチェック	○	データベースレコードとインデックスの構造および形式を調べます。OpenText eDirectory 環境のデータベースレベルで構造的な破損が組み込まれていないことを確認します。
データベース全体を再構築する	○	構造チェックおよびインデックスチェック中に見つかったエラーを解決します。正しいデータ構造を復元し、OpenText eDirectory データベースおよびインデックスファイルを再作成します。
ツリー構造のチェックを実行	○	データベースレコード間のリンクを検証し、各チャイルドレコードに対して有効なペアレントレコードがあることを確認します。これは、データベースの整合性を確認するのに役立ちます。無効なレコードにはマークがつけられ、OpenText eDirectory レプリカ同期処理の実行中に別のパーティションレプリカから復元できます。

操作	データベースのロック	説明
すべてのローカルレプリカを修復	○	各オブジェクトと属性をスキーマ定義でチェックし、OpenText eDirectory データベースの不整合を解決します。ここでは、内部データ構造の形式もすべてチェックします。 この操作では、データベースから無効なレコードを削除することにより、ツリー構造のチェック中に見つかった不整合も解決します。この結果、無効なレコードにリンクされているすべてのチャイルドレコードは、すべてオーファンとしてマークされます。これらのオーファンレコードは失われませんが、この処理によって、データベースの再構築中に多数のエラーが発生する可能性があります。これは正常な反応で、孤立したオブジェクトはレプリカ同期の過程で自動的に再編成されます。
ネットワークアドレスの修復	いいえ	OpenText eDirectory 内で保存しているサーバのネットワークアドレスを、ローカル SAP、SLP、または DNS テーブルで維持されている値でチェックし、OpenText eDirectory が現在も正確な情報を保持していることを確認します。矛盾が見つかった場合、OpenText eDirectory は正しい情報で更新されます。
ストリームシンタックスファイルを確認する	○	ログインスクリプトなどのストリームシンタックスファイルは、OpenText eDirectory データベースの特殊領域に保存されます。この操作では、各ストリームシンタックスファイルが有効な OpenText eDirectory オブジェクトに関連付けられているかどうかをチェックします。関連付けられていないストリームシンタックスファイルは削除され、そのファイルを参照している属性はパージされます。

ローカルデータベースの修復の実行

OpenText eDirectory でオープンおよびアクセスできるように、この修復操作を使用してローカルデータベースの矛盾を解決します。

ローカルデータベースの修復は、一時ファイルセットに対して実行するように指定することもできます。一時ファイルセットを指定しなかった場合、修復操作はアクティブなデータベースに対して実行されます。

一時データベースファイルセットに対して修復操作を実行する場合は、この操作中はデータベースを閉じておく必要があります。操作対象を一時ファイルセットにした場合、修復結果を反映する前に、その確認を求めるメッセージが表示されます。それ以外の場合、修復結果は即座に反映されます。

修復操作が終了すると、その修復操作のログを表示して、修復を完了させるのにさらに必要な操作があるかどうかを確認できます。詳細については、352 ページの 「修復ログファイルの表示と設定」を参照してください。

ndsrepair -J < サーバ固有 ID の 10 進数値 > コマンドを使用して、ローカルデータベースの修復を実行します。

外部参照のチェック

この修復操作は、各外部参照オブジェクトをチェックして、そのオブジェクトを含むレプリカがあるかどうかを調べます。オブジェクトのあるパーティションのレプリカが含まれているすべてのサーバにアクセスできない場合、オブジェクトは見つけられません。オブジェクトが見つからない場合、警告が表示されます。

この操作では破損情報も表示されます。

ndsrepair -C コマンドを使用して外部参照をチェックします。

単一オブジェクトの修復

この修復操作は、OpenText eDirectory がデータへアクセスするのを妨げるような、OpenText eDirectory オブジェクトの不整合を解決します。この操作は、ユーザ作成のパーティションおよび外部参照パーティションでのみ有効です。

この操作は、アクティブなデータベースファイルに対して実行されます。破損が物理的なレベルの場合は、まず物理チェックおよび構造チェックを実行してから単一オブジェクトの修復を行います。

修復操作時点の OpenText eDirectory データベースのバックアップコピーを保持していることを確認します。

ndsrepair -J コマンドを使用して単一オブジェクトを修復します。

不明なリーフオブジェクトの削除

オブジェクトに必須プロパティがない場合、あるいはその他に無効な点がある場合 (プロパティがオブジェクトタイプの最低要件を満たしていない場合)、修復によって一貫性のないオブジェクトが不明なオブジェクトに変更されます。不明なオブジェクトは実際のオブジェクトであり、OpenText eDirectory 側では既知のオブジェクトです。不明なオブジェクトになっているのは、オブジェクトクラスの検証が不完全なためです。疑問符アイコンで表示される不明なオブジェクトは削除できますが、簡単に元のオブジェクトタイプに戻すことはできません。

この修復操作では、ローカル OpenText eDirectory データベースのオブジェクトのうち、オブジェクトクラスが不明で、従属オブジェクトを維持していないオブジェクトをすべて削除します。削除は OpenText eDirectory ツリーの他のレプリカと同時に後で行われます。

重要 : この操作の意味を完全に理解しているか OpenText サポート部門から実行の指示がない限り、この操作は実行しないでください。

---

1. ndsrepair -P -Ad コマンドを実行します。
2. パーティションリストからレプリカオプションを選択します。
3. レプリカオプションリストから 14 を選択します。

修復ログファイルの表示と設定

修復ログファイルには、ローカルパーティションとサーバに関する詳細情報が含まれます。この情報はデータベースの破損を診断するのに役立ちます。ログファイルウィザードでは、修復ログファイルを開いてログファイルオプションを設定できます。

このセクションでは、次の操作について説明します。

- 352 ページの 「ログファイルを開く」
- 352 ページの 「ログファイルオプションを設定する」

ログファイルを開く

この操作を行って、修復ログファイルを表示します。ログファイルのデフォルト名は dsrepair.log です。修復操作の結果は、このログファイルに書き込まれます。

ログファイル操作のオン / オフ切り替え、名前の変更、およびログファイルの削除またはリセットなどができます。詳細については、352 ページの 「ログファイルオプションを設定する」を参照してください。

1 eMBox クライアントにログインし、「list」と入力して、Enter キーを押します。

2 「getloginfo」と入力して、Enter キーを押します。

修復ログファイルが表示されます。

ログファイルオプションを設定する

この操作を行って、修復ログファイルを管理します。ログファイルのオン / オフ切り替え、ログファイルの削除、追加、ファイル名の変更などが行えます。

1 eMBox クライアントにログインし、「list」と入力して、Enter キーを押します。

2 「setloginfo -f filename.log」と入力して、Enter キーを押します。

OpenText iMonitor での修復の実行

修復機能にアクセスするには、iMonitor を使用します。iMonitor の［修復］ページでは、問題を表示したり、OpenText eDirectory データベースのバックアップやクリーンアップを実行できます。

OpenText iMonitor の DSRepair はサーバ限定機能です。つまり、この機能は OpenText iMonitor が実行されているローカルサーバでのみ使用できます。他のサーバで実行されているこの機能にアクセスするには、そのサーバで実行されている OpenText iMonitor に切り替える必要があります。

［DS Repair］ページにアクセスするには、アクセスしようとするサーバの管理者またはコンソールオペレータと同等の権利が必要です。つまり、このページの情報にアクセスするには、まずログインして資格情報のチェックを受ける必要があります。

iMonitor で利用可能な修復機能の詳細については、「262 ページの 「DSRepair 情報の表示」」を参照してください。

レプリカの修復

レプリカの修復操作では、レプリカの各オブジェクトとスキーマとの整合性が保たれているかどうか、オブジェクトの各属性とスキーマとの整合性が保たれているかどうかをチェックし、属性の構文に従ってデータをチェックします。レプリカに関連する他の内部データ構造もチェックされます。

レプリカの修復ウィザードを使用して、次の操作を実行します。

- 353 ページの 「すべてのレプリカを修復する」
- 353 ページの 「選択したレプリカを修復する」
- 353 ページの 「タイムスタンプを修復する」
- 355 ページの 「このサーバを新しいマスタレプリカに設定する」
- 355 ページの 「選択したレプリカを削除する」

すべてのレプリカを修復する

この操作では、レプリカテーブルに表示されたすべてのレプリカを修復します。

30 分前までにローカル OpenText eDirectory データベースの修復操作を行っていない場合、この操作を実行する前にローカルデータベースを修復してください。詳細については、350 ページの 「ローカルデータベースの修復の実行」を参照してください。

すべてのレプリカを修復するには、ndsrepair -R コマンドを実行します。

選択したレプリカを修復する

この操作では、レプリカビューに表示されているレプリカのうち、選択したレプリカのみ修復します。

30 分前までにローカル OpenText eDirectory データベースの修復操作を行っていない場合、この操作を実行する前にローカルデータベースを修復してください。詳細については、350 ページの 「ローカルデータベースの修復の実行」を参照してください。

選択したレプリカを修復するには、「DSRepair eMTool のオプション」セクションを参照してください。

タイムスタンプを修復する

注 : この操作を行う前に、同期修復ウィザードを使用して、レプリカリング内のすべてのサーバが正しく通信していることを確認します。詳細については、362 ページの 「同期化操作の実行」を参照してください。

この操作では、選択したパーティションのレプリカをすべて最新版に更新するために、マスタレプリカの新しい参照ポイントを指定します。

この操作は、常にパーティションのマスタレプリカ上で実行されます。マスタレプリカは、このサーバのローカルレプリカである必要はありません。

オブジェクトが作成または変更されるとタイムスタンプが設定されますが、これは一意でなければなりません。マスタレプリカのタイムスタンプはすべて検査されます。タイムスタンプが現在のネットワーク時間より遅れている場合、新しいタイムスタンプに置き換えられます。タイムスタンプが最新であれば、新しいタイムスタンプは発行されません。すべてのタイムスタンプの時刻が一致すると、新規エポックが宣言されます。

この操作は、レプリカのオブジェクト間、またはオブジェクトのプロパティ間で矛盾が生じている場合に使用します。たとえば、ログインスクリプトを更新したのにログイン時に古いログインスクリプトが表示される場合は、レプリカ間で正しく同期が取られているかどうかを確認してください。将来のタイムスタンプと現在の時刻の時間差が 1 分以内であれば、最終的に OpenText eDirectory 自体がその状況を修正します。新規エポックの宣言は非常に費用のかかる操作であり、定期的な使用はお勧めしていません。

OpenText eDirectory はデータベースとして厳密な整合性はとられていません。したがってそのレプリカ同期の確認には 5 ～ 10 分かかることがあります。この操作を行うと、次の状態になります。

- 新規エポックがマスタレプリカで宣言され、その影響がマスタレプリカのすべてのオブジェクトに及ぶ可能性があります。
- すべてのタイムスタンプが調べられ、必要に応じて修復されます。
- レプリカ間の同期が取られるまで、日付の古いタイムスタンプ (エポック) を保持するレプリカからの更新内容は受け付けられません。
- レプリカは、マスタレプリカまたは新規エポックを受信済みの他のレプリカのすべてのオブジェクトのコピーを受け取ります。
- このレプリカは、マスタレプリカと同じエポックになります。
- 以前のエポックからの変更内容は失われます。
- マスタレプリカが現在のサーバに存在する必要はありませんが、この修復操作を実行するにはマスタレプリカに対するスーパバイザ権が必要です。
- そのほかのレプリカは新しい状態になります。

タイムスタンプを修復して新規エポックを宣言するには、「DSRepair eMTool のオプション」を参照してください。

このサーバを新しいマスタレプリカに設定する

この操作では、選択したパーティションのローカルレプリカをマスタレプリカとして設定します。元のマスタレプリカを損失した場合には、この操作で新しいマスタレプリカを設定できます。マスタレプリカがあるサーバでハードディスク障害が発生すると、そのマスタレプリカが失われることがあります。その場合は、マスタレプリカを変更する必要があります。

詳細については、159 ページの第 6 章「パーティションおよびレプリカの管理」を参照してください。

1. ndsrepair -P -Ad コマンドを実行します。
2. パーティションリストからレプリカオプションを選択します。
3. レプリカオプションリストから 5 を選択します。
4. 管理者の名前とパスワードを指定します。

選択したレプリカを削除する

この操作では、選択したレプリカをこのサーバから削除します。レプリカは削除されるか、サブオーディネートリファレンスに変更されます。

詳細については、159 ページの第 6 章「パーティションおよびレプリカの管理」を参照してください。

1. ndsrepair -P -Ad コマンドを実行します。
2. パーティションリストからレプリカオプションを選択します。
3. レプリカオプションリストから 13 を選択します。
4. 警告メッセージに応答し、管理者の名前とパスワードの詳細を指定します。

レプリカリングを修復する

レプリカリングの修復操作では、レプリカを保持する各サーバのレプリカリング情報をチェックし、リモート ID 情報を検証します。

レプリカリング修復ウィザードを使用して、次の操作を実行します。

- 356 ページの 「すべてのレプリカリングを修復する」
- 356 ページの 「選択したレプリカリングを修復する」
- 356 ページの 「リング内のすべてのサーバにすべてのオブジェクトを送信する」
- 357 ページの 「マスタから選択したレプリカへすべてのオブジェクトを受信する」
- 357 ページの 「レプリカリングからこのサーバを削除する」

すべてのレプリカリングを修復する

この操作では、レプリカビューに表示されたすべてのレプリカのレプリカリングを修復します。

30 分前までにローカル OpenText eDirectory データベースの修復操作を行っていない場合、この操作を実行する前にローカルデータベースを修復してください。詳細については、350 ページの 「ローカルデータベースの修復の実行」を参照してください。

1. ndsrepair -P -Ad コマンドを実行します。
2. パーティションリストからレプリカオプションを選択します。
3. レプリカオプションリストから 8 を選択します。

選択したレプリカリングを修復する

この操作では、レプリカテーブルで選択したレプリカのレプリカリングを修復します。

30 分前までにローカル OpenText eDirectory データベースの修復操作を行っていない場合、この操作を実行する前にローカルデータベースを修復してください。詳細については、350 ページの 「ローカルデータベースの修復の実行」を参照してください。

1. ndsrepair -P -Ad コマンドを実行します。
2. パーティションリストからレプリカオプションを選択します。
3. レプリカオプションリストから 9 を選択します。

リング内のすべてのサーバにすべてのオブジェクトを送信する

この操作では、レプリカリング内で選択したサーバから、選択したパーティションのレプリカを含む他のすべてのサーバに、すべてのオブジェクトを送信します。

レプリカリング内で選択したサーバ上の選択したパーティションのレプリカが、レプリカリング内の他のすべてのサーバと同期していることを確かめるには、この操作を行います。該当するパーティションのサブオーディネートリファレンス (レプリカ) のみを含むサーバでは、この操作は実行できません。

選択したサーバに保持されているレプリカとまだ同期していない他のレプリカに加えた変更は失われます。この操作を実行する前に、同期ステータスを確認してください。

重要 : この操作はレプリカ内のオブジェクトを再作成するため、ネットワークトラフィックの量が大幅に増加する可能性があります。これは診断操作ではありません。

---

1. ndsrepair -P -Ad コマンドを実行します。
2. パーティションリストからレプリカオプションを選択します。
3. レプリカオプションリストから 10 を選択します。
4. サーバの詳細を指定します。

5 3 を選択します。

6 管理者の名前とパスワードを指定します。

マスタから選択したレプリカへすべてのオブジェクトを受信する

この操作では、選択したサーバ上のレプリカで、マスタレプリカのすべてのオブジェクトを受信します。

レプリカリング内で選択したサーバ上の選択したパーティションのレプリカが、マスタレプリカと同期していることを確かめるには、この操作を行います。この操作は、マスタレプリカがあるサーバでは実行できません。

重要: この操作を行うと、ネットワークトラフィックの量が大幅に増加します。この操作を要求することにより、現在のレプリカは、サーバに新しいレプリカがあるかのように動作します。さらに、レプリカの状態は新規になります。

---

1 ndsrepair -P -Ad コマンドを実行します。

2 パーティションリストからレプリカオプションを選択します。

3 レプリカオプションリストから 10 を選択します。

4 サーバの詳細を指定します。

5 4 を選択します。

6 管理者の名前とパスワードを指定します。

レプリカリングからこのサーバを削除する

この操作では、現在のサーバに保存されているレプリカのうち、選択したレプリカから特定のサーバを削除します。

警告: この操作を誤用すると、OpenText eDirectory データベースで致命的な破損が生じることがあります。OpenText サポート担当者からの指示がない限り、この操作は実行しないでください。

---

1 ndsrepair -P -Ad コマンドを実行します。

2 パーティションリストからレプリカオプションを選択します。

3 レプリカオプションリストから 10 を選択します。

4 サーバの詳細を指定します。

5 6 を選択します。

6 管理者の名前とパスワードを指定します。

スキーマの保守

スキーマとはオブジェクト属性のルールおよび定義を体系化したもので、これにより、各オブジェクトの内容と形式が構築され、データベース内でのオブジェクト間の関係が確立されます。

スキーマの保守ウィザードには、OpenText eDirectory サーバのスキーマを［Root］のマスタに合わせるために必要なスキーマ操作がいくつか用意されています。ただし、これらの操作は必要な場合にのみ使用してください。スキーマは、ローカル修復操作および標準修復操作によってすでに検査されています。

OpenText eDirectory スキーマの詳細については、145 ページの第 5 章「スキーマの管理」を参照してください。

スキーマの保守ウィザードを使用して、次の操作を実行します。

- 358 ページの 「ツリーからスキーマを要求する」
- 359 ページの 「ローカルスキーマをリセットする」
- 359 ページの 「オプションスキーマ拡張機能を実行する」
- 359 ページの 「リモートスキーマをインポートする」
- 360 ページの 「新規スキーマエポックを宣言する」

ツリーからスキーマを要求する

この操作を実行すると、ツリーのルートのマスタレプリカが自身のスキーマをこのサーバのスキーマに同期させます。スキーマに対する変更の内容は、24 時間以内に [Root] のマスタレプリカからこのサーバに伝達されます。

重要 : すべてのサーバがマスタレプリカのスキーマを要求すると、ネットワークトラフィックが増加します。そのため、このオプションの使用には細心の注意を払うようにしてください。

---

1. Ndsrepair -S -Ad を実行します。
2. 管理者の名前とパスワードを指定します。
3. 次のオプションを選択します : 1. マスタサーバからスキーマを要求する (Request schema from master server)。

ローカルスキーマをリセットする

この操作は、ローカルスキーマのタイムスタンプをクリアし、着信スキーマの同期を要求するスキーマリセット機能を呼び出します。

［Root］パーティションのマスタレプリカから実行した場合、この操作はできません。ツリー内のすべてのサーバが同時にリセットされるわけではありません。

1 Ndsrepair -S -Ad を実行します。
2 管理者の名前とパスワードを指定します。
3 次のオプションを選択します : 2. ローカルスキーマをリセット (Reset local schema)。

オプションスキーマ拡張機能を実行する

この操作では、包含など拡張機能のためにスキーマを拡張および変更します。

この操作では、このサーバに［Root］パーティションのレプリカが含まれ、そのレプリカが使用可能な状態であることが必要になります。

以前のバージョンの OpenText eDirectory では、このオプションで加えた変更の同期ができません。

1 Ndsrepair -S -Ad を実行します。
2 管理者の名前とパスワードを指定します。
3 次のオプションを選択します : 3. オプションスキーマ拡張機能 .

リモートスキーマをインポートする

この操作では、現在のツリーのスキーマに追加したいスキーマが含まれている OpenText eDirectory ツリーを選択します。

ツリーを選択すると、［Root］パーティションのマスタレプリカを保持するサーバに接続されます。現在のツリー上にあるスキーマの拡張には、そのサーバのスキーマが使用されます。

2 つのツリーをマージするには、一方のツリーからもう一方のツリーへ繰り返しスキーマをインポートする必要があります。

1 Ndsrepair -S -Ad を実行します。
2 管理者の名前とパスワードを指定します。
3 ［ツリー (Tree)］からオプション 4 の［Import スキーマ (Import schema)］(グローバルスキーマオプション) を選択します。
4 IP アドレスまたはツリー名を指定します。

新規スキーマエポックを宣言する

エポックとは、基準点として任意に選択される瞬間のことです。時代や新規バージョンと同義です。エポックは、レプリカの同期を制御します。新規エポックに宣言されると、マスタレプリカで機能します。他のレプリカは、新規エポックを持つレプリカに更新情報を送ることはできませんが、そのレプリカと完全に同期するまでは更新情報を受け取ります。

指定したパーティションの他のレプリカが更新済みレプリカと同期された場合、つまり各レプリカのエポックが同じになると、双方向の同期が再び許可されます。

新規スキーマエポッククを宣言すると、[Root] パーティションのマスタレプリカを保持するサーバに接続され、スキーマレコード上の不正なタイムスタンプが修復されます。次にスキーマの新規エポックがそのサーバで宣言され、ツリー全体に影響を与えます。

他のすべてのサーバは、修復されたタイムスタンプを保持する新しいスキーマのコピーを受け取ります。

受け取る側のサーバが新規エポック内に存在しなかったスキーマを含む場合は、古いスキーマを使用するオブジェクトおよび属性が「不明」オブジェクトクラスまたは属性に変更されます。

重要 : OpenText サポート部門からの指示がない限り、この操作は実行しないでください。

---

1. Ndsrepair -S -Ad を実行します。
2. 管理者の名前とパスワードを指定します。
3. 次のオプションを選択します : 5. 新しいエポックの宣言 (Declare a new epoch)。

サーバのネットワークアドレスの修復

サーバの修復ウィザードでは、レプリカリングのすべてのサーバのネットワークアドレスとローカルデータベースのサーバオブジェクトを修復します。さらに、レプリカリングの選択したサーバのネットワークアドレスとローカルデータベースのサーバオブジェクトも修復できます。

サーバの修復ウィザードを使用して、次の操作を実行します。

- 361 ページの 「すべてのネットワークアドレスを修復する」
- 361 ページの 「サーバのネットワークアドレスの修復」

すべてのネットワークアドレスを修復する

この操作では、ローカル OpenText eDirectory データベース内で、すべてのサーバのネットワークアドレスをチェックします。使用できるトランスポートプロトコルに応じて、SAP テーブル、SLP ディレクトリエージェント、および DNS ローカルまたはリモート情報でサーバ名を検索します。

その後、OpenText eDirectory サーバオブジェクトのネットワークアドレス属性、およびすべてのパーティション［Root］オブジェクトの各レプリカ属性のアドレスレコードと、各アドレスが比較されます。アドレスが異なる場合は、同じになるように更新されます。

SAP テーブル、ローカル / リモート DNS 情報、または SLP ディレクトリエージェントにサーバのネットワークアドレスが見つからなければ、修復は行われません。

1. ndsrepair -N -Ad コマンドを実行します。
2. サーバリストから、必要なサーバを選択します。
3. サーバオプションリストから、［1: すべてのネットワークアドレスを修復する (Repair all network addresses)］を選択します。

サーバのネットワークアドレスの修復

この操作では、ローカル OpenText eDirectory データベースファイル内で、選択したサーバのネットワークアドレスをチェックします。現在バインドされているトランスポートプロトコルに応じて、ローカル SAP テーブル、SLP ディレクトリエージェント、およびローカルまたはリモートの DNS 情報でサーバ名を検索します。ネットワークアドレスが見つかれば、そのアドレスを OpenText eDirectory サーバオブジェクトのネットワークアドレス属性、およびすべてのパーティション［Root］オブジェクトの各レプリカ属性のアドレスレコードと照合します。アドレスが異なる場合は、同じになるように更新されます。

SAP テーブル、SLP ディレクトリエージェント、またはローカル / リモート DNS 情報にサーバのネットワークアドレスが見つからなければ、修復は行われません。

1. ndsrepair -P -Ad コマンドを実行します。
2. パーティションリストからレプリカオプションを選択します。
3. レプリカオプションリストから 2 を選択します。

問題

OpenText SLP はオプションのパッケージです。認証機能は、OpenText SLP パッケージの一部としては実装されません。

OpenText eDirectory は現在 OpenSLP との互換性があり、OpenSLP の認証機能が使用されます。

注 : Linux では、OpenText eDirectory はすべてのインタフェースをリスンするわけではありませんが、nds.conf ファイルで指定された特定の IP は例外です。新しい IPV6 アドレスを追加する場合は、リスナを開始する新しいアドレスと、対応する追加対象の参照を反映するよう nds.conf ファイルが変更されていることを確認します。

同期化操作の実行

同期修復ウィザードでは、現在のサーバで選択したレプリカの同期、現在のサーバの同期ステータスのレポート、すべてのサーバの同期ステータスのレポート、時刻同期の実行、および即時同期のスケジュールができます。

同期修復ウィザードを使用して、次の操作を実行します。

- 362 ページの 「選択したレプリカをこのサーバで同期する」
- 362 ページの 「このサーバの同期ステータスをレポートする」
- 363 ページの 「すべてのサーバの同期ステータスをレポートする」
- 363 ページの 「時刻同期を実行する」
- 364 ページの 「即時同期をスケジュールする」

選択したレプリカをこのサーバで同期する

この操作では、選択したパーティションのレプリカを持つすべてのサーバ上で、完全な同期ステータスを確保します。

このオプションにより、パーティションの状態を確認できます。そのパーティションのレプリカを持つサーバがすべて正常に同期していれば、そのパーティションは異常なしと見なされます。レプリカリング内の各サーバが接続されると、次に、接続された各サーバは、レプリカリング内の他のすべてのサーバに対して、即時に同期を実行します。

サーバは、そのサーバ自身とは同期されません。そのため、現在のサーバのレプリカのステータスは［ホスト］として示されます。

1. ndsrepair -P -Ad コマンドを実行します。
2. パーティションリストからレプリカオプションを選択します。
3. レプリカオプションリストから 10 を選択します。
4. サーバの詳細を指定します。
5. 2 を選択します。
6. 管理者の名前とパスワードを指定します。

このサーバの同期ステータスをレポートする

この操作では、現在のサーバ上にレプリカを持つすべてのパーティションのレプリカ同期ステータスをレポートします。

この操作では、パーティションのレプリカを保持する各サーバのレプリカの［Root］オブジェクトから同期ステータス属性を読み込みます。レポートには、すべてのサーバに対して正常に同期が行われた最終時刻と、最終同期以降発生したエラーが表示されます。

12 時間以内に同期が完了していない場合は、警告メッセージが表示されます。

1. ndsrepair -P -Ad コマンドを実行します。
2. パーティションリストから、そのパーティションを選択します。

3 レプリカオプションリストから、［10: レプリカリングの表示 (View Replica Ring)］を選択します。
4 サーバのリストから、必要なサーバを選択します。
5 サーバオプションから、［1: 選択したサーバの同期ステータスのレポート (Report synchronization status on the selected server)］を選択します。

すべてのサーバの同期ステータスをレポートする

この操作では、現在のサーバ上にレプリカを持つすべてのパーティションのレプリカ同期ステータスを確保します。

この操作では、パーティションのレプリカを保持する各サーバのレプリカの［Root］オブジェクトから同期ステータス属性を読み込みます。レポートには、すべてのサーバに対して正常に同期が行われた最終時刻と、最終同期以降発生したエラーが表示されます。

12 時間以内に同期が完了していない場合は、警告メッセージが表示されます。

1 ndsrepair -P -Ad コマンドを実行します。
2 パーティションリストから、そのパーティションを選択します。
3 レプリカオプションリストから、［6: すべてのサーバの同期ステータスのレポート (Report synchronization status of all servers)］を選択します。

時刻同期を実行する

この操作では、ローカル OpenText eDirectory データベースに登録されているすべてのサーバに接続し、各サーバの OpenText eDirectory と時刻同期ステータスに関する情報を要求します。

各サーバ上で実行している OpenText eDirectory のバージョンが［DS バージョン］フィールドに表示されます。

サーバに何のレプリカも含まれていない場合は、［レプリカの深さ］フィールドに「-1」と表示されます。［Root］パーティションのレプリカが含まれている場合は、「0」と表示されます。指定したサーバ上にレプリカがある場合は、［Root］に最も近いレプリカが［Root］からオブジェクト何個分離れているかを示す正の整数が表示されます。

OpenText eDirectory ツリー内のすべてのサーバは、同じタイムソースと同期している必要があります。そうしなければ、矛盾が発生したときに、レプリカ間でのオブジェクトの同期が正確に管理されなくなります。

同期修復ウィザードでは、各サーバのタイムソースはレポートできません。ただし、タイムサーバのタイプは報告します。この情報を参照すれば、時刻同期が正確に設定されているかどうかを確認できます。

重要 : 時刻同期のステータス「ほぼ同期 (Nearly-In-Sync)」を監視するには、DSRepair ではなく OpenText iMonitor を使用してください。詳細については、243 ページの第 8 章「OpenText の監視」を参照してください。

時刻同期を実行するには、ndsrepair -T コマンドを実行します。

詳細については、105 ページの 「ネットワーク時刻の同期」を参照してください。

即時同期をスケジュールする

この操作では、すべてのレプリカの同期を即座に行うようスケジュールします。この操作は、同期プロセスが通常のスケジュールどおりに実行されるのを待つことなく、同期情報を確認したい場合に使用します。

すべてのレプリカの同期を即座に行うようスケジュールするには :

1. ndsrepair -P -Ad コマンドを実行します。
2. パーティションリストからレプリカオプションを選択します。
3. レプリカオプションリストから、[3: 即時同期のスケジュール (Scheduling an Immediate Synchronization)] を選択します。

DSRepair オプション

各 OpenText eDirectory プラットフォームの DSRepair ユーティリティには、通常の使用時には非表示になっている高度な機能がいくつか含まれています。これらの拡張機能は、さまざまなプラットフォームで DSRepair ユーティリティをロードする際にスイッチを使用して利用可能にできます。

- 364 ページの 「OpenText eDirectory サーバ上で DSRepair を実行する」
- 366 ページの 「DSRepair コマンドラインオプション」
- 368 ページの 「DSRepair 詳細設定スイッチの使用」

OpenText eDirectory サーバ上で DSRepair を実行する

- 364 ページの 「Windows」
- 364 ページの 「Linux」

Windows

1. [スタート] > [設定] > [コントロールパネル] > [OpenText eDirectory サービス] の順にクリックします。
2. [dsrepair.dlm] をクリックして、[開始] をクリックします。

   DSRepair の詳細オプションを開くには、OpenText eDirectory Services コンソールの [起動パラメータ] フィールドに「–a」と入力し、dsrepair.dlm を開始します。

Linux

DSRepair を実行するには、サーバコンソールに次の構文を使用して「ndsrepair」と入力します。

```
ndsrepair {-U |-E |-C |-P [-Ad] |-S [-Ad]|-N |-T |-J <entry_id> [-Ad -AM
<attribute name>]} [-A <yes/no>] [-O <yes/no>][-F filename] [-h
<local_interface:port>] [--config-file <configuration_file_path>]
```

または

```
ndsrepair -R [-l yes|no] [-u yes|no] [-m yes|no] [-i yes|no] [-f yes|no][-
d yes|no] [-t yes|no] [-o yes|no][-r yes|no] [-v yes|no] [-c yes|no] [-F
filename] [-A yes|no] [-O yes|no]
```

重要 : 詳細設定スイッチ［-Ad］は最後の引数として指定する必要があります。［-Ad］詳細設定スイッチオプションは、OpenText サポート技術者からの指示があった場合にのみ有効にすることをお勧めします。config-file を引数として指定する場合は、詳細設定スイッチ［-Ad］の前に指定する必要があります。

例

標準修復を実行し、/root/ndsrepair.log ファイルにイベントを記録する場合 (またはログファイルがすでに存在していればそのログファイルに追加してイベントを記録する場合) は、次のコマンドを入力します。

```
ndsrepair -U -A no -F /root/ndsrepair.log
```

すべてのグローバルスキーマ操作とその詳細設定オプションのリストを表示するには、次のコマンドを入力します。

```
ndsrepair -S -Ad
```

データベースを強制ロックしてローカルデータベースを修復するには、次のコマンドを入力します。

```
ndsrepair -R -l yes
```

オブジェクトのエントリ ID が分かっている場合に 1 つのオブジェクトを修復するには、次のコマンドを入力します。

```
ndsrepair -J <entry ID in hex>
```

特定のパーティションまたはレプリカを修復するには、次のコマンドを入力します。

```
ndsrepair -P
```

このコマンドは、サーバに存在するすべてのパーティションのリストを返します。いずれかのパーティションを選択して実行可能な操作のリストを取得できます。

使用するために解放可能なデータベースの空き領域に関する情報を表示するには、次のコマンドを入力します。

```
ndsrepair -I
```

ネットワークアドレスを修復するには、次のコマンドを入力します。

```
ndsrepair -N
```

注 : ndsrepair コマンドの入力内容は、オプションファイルによってリダイレクトできます。オプションファイルは、レプリカおよびパーティション操作に関連するオプションやサブオプションを含むテキストファイルです。これらはサーバに対する認証を必要としません。各オプションまたはサブオプションは改行によって区切られます。ファイルの内容が、適切な順序で指定されていることを確認します。適切な順序になっていないと、予期しない結果が発生する場合があります。

DSRepair コマンドラインオプション

オプション	説明
-U	［標準修復］オプションです。ユーザの操作なしに DSRepair を実行して終了します。修復が完了したらログファイルをチェックして、DSRepair で実行された処理を確認できます。 このオプションの指定時に実行される修復は、デフォルトの推奨されている通常の修復ではありません。特定の問題をトラブルシューティングし、それを解決することは標準修復を実行することよりはるかに優れています。
-P	［レプリカ操作とパーティション操作］オプションです。現在のサーバの OpenText eDirectory データベースファイルにレプリカが保存されているパーティションが表示されます。［レプリカオプション］メニューには、「レプリカの修復」、「パーティション操作のキャンセル」、「同期のスケジュール」、および「ローカルレプリカをマスタレプリカとして指定」を実行するオプションがあります。
-S	［グローバルスキーマの操作］オプションです。このサーバのスキーマを Tree オブジェクトのマスタに準拠させるのに必要なスキーマ操作がいくつか含まれています。ただし、これらの操作は必要な場合にのみ使用してください。スキーマは、ローカル修復操作および標準修復操作によってすでに検査されています。
-C	［外部参照オブジェクトのチェック］オプションです。各外部参照オブジェクトをチェックして、そのオブジェクトを含むレプリカがあるかどうかを調べます。オブジェクトのあるパーティションレプリカを含むすべてのサーバがアクセス不能の場合、オブジェクトは見つかりません。オブジェクトが見つからない場合、警告が表示されます。
-E	［レプリカ同期のレポート］オプションです。現在のサーバ上にレプリカを持つすべてのパーティションのレプリカ同期ステータスをレポートします。この操作により、パーティションのレプリカを保持する各サーバ上にあるレプリカのツリーオブジェクトから、同期ステータス属性が読み込まれます。レポートには、すべてのサーバに対して正常に同期が行われた最終時刻と、最終同期以降発生したエラーが表示されます。12 時間以内に同期が完了していない場合は、警告メッセージが表示されます。

オプション	説明
-N	［このデータベースに認識されているサーバ］オプションです。ローカル OpenText eDirectory データベースに登録されているすべてのサーバが表示されます。現在のサーバに Tree パーティションのレプリカがある場合、このサーバには OpenText eDirectory ツリー内のすべてのサーバのリストが表示されます。サーバオプションを実行するサーバを 1 つ選択します。
-J	ローカルサーバ上の 1 つのオブジェクトを修復します。修復するオブジェクトのエントリ ID(16 進形式で) を指定する必要があります。破損している 1 つの特定のオブジェクトを修復する場合、標準修復 (-U) オプションの代わりに、このオプションを使用できます。データベースのサイズによっては、［標準修復］オプションの完了に何時間もかかる場合があります。このオプションを使用して、時間を節約することができます。
-T	［時刻同期］オプションです。ローカル OpenText eDirectory データベースに登録されているすべてのサーバにアクセスして、各サーバの時刻同期ステータスの情報を要求します。このサーバに Tree パーティションのレプリカがある場合は、OpenText eDirectory ツリー内のすべてのサーバがポーリングされます。各サーバ上で実行されている OpenText eDirectory のバージョンもレポートされます。
-A	既存のログファイルに付加します。情報は既存のログファイルに追加されます。デフォルトでは、このオプションは有効です。
-O	出力をファイルに記録します。デフォルトでは、このオプションは有効です。
-F *filename*	出力を指定したファイルに記録します。
-R	［ローカルデータベースの修復］オプションです。ローカル OpenText eDirectory データベースを修復します。OpenText eDirectory でオープンおよびアクセスできるように、この修復操作を使用してローカルデータベースの矛盾を解決します。このオプションには、データベースの修復操作を容易にするサブオプションがあります。このオプションにはファンクション修飾子があります。ファンクション修飾子については、次の表で説明します。
-I	使用するために解放可能なデータベース内の空き領域に関する情報を表示します。OpenText eDirectory を使用すると、空のレコードを取得し、ndsrepair コマンドの再利用オプションを使用して、空き領域を再利用することができます。

-R オプションで使用するファンクション修飾子を次に示します。

オプション	説明
-l	修復操作中に OpenText eDirectory データベースをロックします。

オプション	説明
-u	修復操作中に一時 OpenText eDirectory データベースを使用します。変更の保存または破棄するためのプロンプト、およびログファイルの表示のプロンプトが表示されます。
-m	修復されていない元のデータベースを維持します。
-i	OpenText eDirectory データベース構造とインデックスをチェックします。
-f	データベースの空き領域を増やします。
-d	データベース全体を再構築します。
-t	ツリー構造のチェックを実行します。データベース内での接続状況が正しいかどうかを調べるため、ツリー構造のリンクをすべてチェックするには、「はい」を設定します。チェックを省略するには、「いいえ」を設定します。デフォルト値は「はい」です。
-o	オペレーショナルスキーマを再構築します。
-r	すべてのローカルレプリカを修復します。
-v	ストリームファイルを確認します。
-c	ローカル参照をチェックします。

DSRepair 詳細設定スイッチの使用

警告 : このセクションで説明する機能は、正しく使用しないと OpenText eDirectory ツリーが破損して元に戻せないことがあります。OpenText サポート担当者からの指示がない限り、この機能は使用しないでください。

運用環境でこれらのうちのいずれかの機能を使用する前に、あらかじめサーバ上の OpenText eDirectory のフルバックアップをとっておくことをお勧めします。詳細については、459 ページの第 15 章「OpenText eDirectory のバックアップと復元」を参照してください。

Linux では、「ndsrepair -R -Ad -XK2」と入力します。

Windows では、dsrepair.dlm を開始する前に、NDS コンソールの［**起動パラメータ**］フィールドにこれらのオプションを入力します。詳細については、364 ページの 「OpenText eDirectory サーバ上で DSRepair を実行する」を参照してください。

スイッチ	説明
-P	タイプが［不明］であるすべての OpenText eDirectory オブジェクトを、参照済みとしてマークします。参照されたオブジェクトは、OpenText eDirectory レプリカ同期処理の対象にはなりません。

スイッチ	説明
-WM	多くの場合、ZENworks ® 2.0 を使用していると、WM: Registered Workstations 属性が非常に高くなります。DSRepair を -WM オプションを指定して実行すると、こうした高い値がクリアされます。
-XK2	このサーバの OpenText eDirectory データベース内のすべての OpenText eDirectory オブジェクトを削除します。この操作では、破損したレプリカがどんな方法を使っても削除できない場合に、破損レプリカを削除できます。
-XK3	このサーバの OpenText eDirectory データベース内のすべての外部参照を削除します。この操作では、機能していないレプリカ内の外部参照をすべて削除できます。参照が原因で問題が発生している場合、レプリカが再度機能するために、OpenText eDirectory は参照を再作成できます。
-RC	DIB をバックアップします。このオプションは、Windows でのみ使用できます。
-OT	ローカルデータベース修復の実行中、破損通知にタイムスタンプを含めます。INHIBIT MOVE を除くすべての破損通知にタイムスタンプを含めます。
-NLD	NLS: ライセンス許可証オブジェクトと NLS: 製品コンテナオブジェクトから IRF を削除します。
-AM	特定の条件に一致する属性を FLAIM データベース内の別のコンテナに移動します。どの OpenText eDirectory 属性が別のコンテナへの移動に適格であるかについては、「FLAIM Attribute Containerization」を参照してください。
-AH	DIB サイズが 1GB よりも小さく、古い NDO ファイルが 72 時間よりも古い場合、NDO ファイルは作成されません。

クライアントを使用したデータベースの修復

eMBox (OpenText eDirectory Management Toolbox) クライアントはコマンドライン Java クライアントで、これを使用すると DSRepair にリモートアクセスできます。クライアントはバッチモードで実行できるため、これを使用して OpenText eDirectory DSRepair eMTool で標準修復を行うことができます。

emboxclient.jar ファイルは、OpenText eDirectory の一部としてサーバにインストールされます。JVM をインストールしていれば、どのコンピュータでも実行できます。クライアントの詳細については、「618 ページの 「コマンドラインクライアントの使用」」を参照してください。

DSRepair eMTool を使用する

1 コマンドラインで次のように入力して、対話式モードでクライアントを実行します。

```
java -cp path_to_the_file/emboxclient.jar -i
```

(クラスパスに emboxclient.jar ファイルがすでに含まれている場合は、単に「java -i」と入力します。)

クライアントのプロンプトが次のように表示されます。

```
Client>
```

2 修復するサーバにログインするには、次のように入力します。

```
login -s server_name_or_IP_address -p port_number
-u username.context -w password -n
```

ポート番号は通常 80 または 8028 です。ただし、すでにそのポートを使用している Web サーバが存在する場合は異なります。-n オプションを使用すると、非セキュア接続が開始されます。

クライアントはログインが成功したかどうかを表示します。

3 次の構文を使用して修復コマンドを入力します。

dsrepair.*task options*

たとえば、dsrepair.ufr は標準修復を実行します。

dsrepair.rld -a –v は、［すべてのローカルレプリカを修復］オプションおよび［ローカル参照をチェックする］オプションを使用して、ローカルデータベースを修復します。

各スイッチの間にはスペースが必要です。スイッチの順序は重要ではありません。

クライアントは修復が成功したかどうかを表示します。

DSRepair eMTool オプションの詳細については、「370 ページの 「DSRepair eMTool のオプション」」を参照してください。

4 クライアントからログアウトするには、次のコマンドを入力します。

```
logout
```

5 クライアントを終了するには、次のコマンドを入力します。

```
exit
```

DSRepair eMTool のオプション

次の表に DSRepair eMTool のオプションを示します。クライアントで list -t dsrepair コマンドを使用して DSRepair オプションの詳細を表示することもできます。詳細については、622 ページの 「eMTool とそのサービスを表示する」を参照してください。

オプション	説明
rso -o -d	単一オブジェクトの修復 (16 進数のオブジェクト ID、オブジェクト DN)
rts	時刻同期
rss	すべてのパーティションの同期ステータスのレポート

オプション	説明
rld -l -t -d -p -i -f -c -o -a -m -v	◆ ローカルデータベースを修復する ◆ 全体の修復中に OpenText eDirectory データベースをロックする ◆ 修復中に一時 OpenText eDirectory データベースを使用する ◆ 修復されていない元のデータベースを維持する ◆ データベース構造のチェックを実行する ◆ データベース構造とインデックスのチェックを実行する ◆ データベースの空き領域を増やす ◆ ツリー構造のチェックを実行する ◆ オペレーショナルスキーマを再構築する ◆ すべてのローカルレプリカを修復する ◆ メールディレクトリとストリームファイルを検証する ◆ ローカル参照をチェックする
ufr	標準修復
rsn -o -d	選択したサーバのネットワークアドレスを修復する (16 進数のオブジェクト ID、オブジェクト DN)
ran	すべてのネットワークアドレスの修復
rsr -p -d	選択したレプリカの修復 (パーティション ID、パーティション DN)
rer	すべてのレプリカの修復
ror -p -d	選択したレプリカリングの修復 (パーティション ID、パーティション DN)
rar	すべてのレプリカのレプリカリングの修復
ssa -p -d	すべてのサーバのレプリカ同期ステータスのレポート (パーティション ID、パーティション DN)
cer	外部参照のチェック
rao -p -d -s -d	このレプリカのすべてのオブジェクトの受信 (パーティション ID、パーティション DN、サーバ ID、サーバ DN)
sao -p -d -s -d	リング内のすべてのレプリカに対するすべてのオブジェクトの送信 (パーティション ID、パーティション DN、サーバ ID、サーバ DN)
dne -p -d	タイムスタンプの修復と新しいエポックの宣言 (パーティション ID、パーティション DN)
sri -p -d	即時同期のスケジュール (パーティション ID、パーティション DN、サーバ ID、サーバ DN)
sks -p -d -s -d	選択したサーバでのレプリカの同期 (パーティション ID、パーティション DN、サーバ ID、サーバ DN)

オプション	説明
ske -p -d	すべてのサーバでのレプリカの同期 (パーティション ID、パーティション DN)
dsr -p -d	このサーバで選択したレプリカの削除 (パーティション ID、パーティション DN)
xsr -p -d -s -d	レプリカリングからのサーバの削除 (パーティション ID、パーティション DN、サーバ ID、サーバ DN)
dnm -p -d	このサーバを新しいマスタレプリカとして設定 (パーティション ID、パーティション DN)
dul	不明リーフオブジェクトの削除

グラフィカルな DS 修復ユーティリティ

グラフィカルな DS 修復ユーティリティが OES 11 SP1 に追加されました。このツールは新しい OES 11 SP1 のインストール時に自動的にインストールされます。

ユーザインタフェースを起動するには、サーバコンソールで ndsgrepair コマンドを実行します。コンソールを使用して実行可能な修復操作のほとんどは、このグラフィカルインタフェースを使用して実行できます。すべてのヘルプトピック (メニューオプションなど) を参照するには、F1 キーを押すか、または UI メインメニューで、[**ヘルプ**] > [**ヘルプコンテンツ**] の順にクリックします。

OES 11 SP1 にアップグレードする場合は、次の手順を実行して、OpenText eDirectory パターンの下で novell-ndsgrepair を手動で選択します。

1. YaST を開き、[**OES Install and Configuration**] を選択しします。
2. [詳細] をクリックし、左側の [**Novell eDirectory Pattern**] を選択して、右側の [パッケージ] の一番下までスクロールします。
3. [**novell-ndsgrepair**] を選択し、[**受諾**]、[次へ]、[**終了**] の順にクリックします。

13 LDAP Services for OpenText eDirectory について

LDAP (Lightweight Directory Access Protocol) は、クライアントアプリケーションでディレクトリ情報にアクセスするためのインターネット通信プロトコルです。LDAP は、X.500 DAP (Directory Access Protocol) に基づいていますが、従来のクライアントほど複雑ではなく、X.500 標準に基づくその他のディレクトリサービスと同時に使用することができます。

一般に、LDAP は最も単純なディレクトリアクセスプロトコルとして使用されます。

OpenText eDirectory 用の LDAP (Lightweight Directory Access Protocol) サービスは、LDAP クライアントから OpenText eDirectory に保存されている情報にアクセスするためのサーバアプリケーションです。

LDAP サービスには、LDAP を通じて利用できる次のような OpenText eDirectory 機能が含まれます。

- プロビジョニング
- アカウント管理
- 認証
- 権限付与
- 識別情報管理
- 通知
- レポーティング
- 認定
- セグメンテーション

クライアントごとに異なるディレクトリアクセスレベルを設定して、ディレクトリにアクセスするための安全な接続を確立できます。このセキュリティメカニズムを利用すると、一般に公開するディレクトリ情報、組織内で利用する情報、および特定のグループまたは個人だけが利用できる情報を区別して管理できます。

各 LDAP クライアントで利用できるディレクトリ機能は、LDAP クライアントおよび LDAP サーバに組み込まれた機能により異なります。たとえば、LDAP Services for OpenText eDirectory を利用すると、LDAP クライアントは OpenText eDirectory データベース内のデータを読み書きできます。ただし、これには LDAP クライアントに必要な許可が与えられている必要があります。クライアントは、ディレクトリデータの読み書きが許可される場合と、読み込みしか許可されない場合があります。

一般的なクライアント機能を利用すると、クライアントから次のような処理を実行できます。

- 電子メールアドレスや電話番号など、特定の個人についての情報を検索する。

- 特定の姓または特定の文字で始まる姓を持つすべての個人の情報を検索する。
- 任意の OpenText eDirectory オブジェクトまたはエントリについての情報を検索する。
- 氏名、電子メールアドレス、勤務先電話番号、および自宅電話番号を取得する。
- 会社名および市町村名を取得する。

以下のセクションで、LDAP Services for OpenText eDirectory について説明します。

- 374 ページの 「LDAP サービスの主な用語」
- 377 ページの 「LDAP と OpenText eDirectory の連携について」
- 386 ページの 「Linux での LDAP ツールの使用」
- 399 ページの 「拡張可能一致検索フィルタ」
- 401 ページの 「LDAP トランザクション」

LDAP の詳細については、次の Web サイトを参照してください。

- OpenLDAP (http://www.openldap.org/)
- LDAP Roadmap & FAQ (http://www.kingsmountain.com/ldapRoadmap.shtml)

LDAP サービスの主な用語

- 374 ページの 「クライアントとサーバ」
- 374 ページの 「オブジェクト」
- 375 ページの 「参照」

クライアントとサーバ

LDAP クライアント — Internet Explorer、OpenText インポート / エクスポート変換ユーティリティなどのような、1 つのアプリケーションです。

LDAP サーバ — nldap.dlm (Windows 用) または libnldap.so (Linux 用) が動作しているサーバです。

オブジェクト

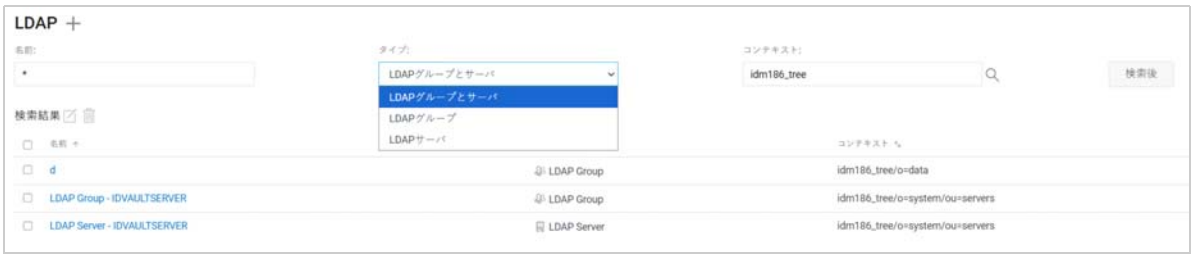
LDAP グループオブジェクト — LDAP サーバで OpenText LDAP プロパティの設定と管理を行います。

このオブジェクトは OpenText eDirectory のインストール時に作成されます。LDAP グループオブジェクトには、複数の LDAP サーバ間で共有できる便利な設定情報が含まれています。

LDAP サーバオブジェクト — LDAP クライアントによる OpenText LDAP サーバ上の情報へのアクセスおよび使用方法の設定と管理を行います。

このオブジェクトは OpenText eDirectory のインストール時に作成されます。LDAP サーバオブジェクトとは、サーバ固有の設定データのことです。

次の図は、OpenText Identity Console の LDAP サーバオブジェクトを表したものです。



参照

参照 — LDAP サーバが LDAP クライアントに送信するメッセージです。このサーバからは結果がすべて提供されず、他の LDAP サーバにまだデータがある可能性をクライアントに通知します。

参照には、操作を続行するのに必要な情報がすべて含まれます。

シナリオ :LDAP クライアントが LDAP サーバに要求を送信しますが、サーバは操作のターゲットエントリをローカルで見つけることができません。その場合、LDAP サーバはパーティションおよび他のサーバに関して所有する知識参照を使用して、そのエントリについてより多くの知識を持つ別のサーバを特定します。LDAP サーバは参照情報をクライアントに送信します。

クライアントは識別されたサーバに対する新しい LDAP 接続を確立し、操作を再試行します。

参照には次の利点があります。

- LDAP クライアントが操作を制御し続けます。

  常に状況を把握することにより、クライアントはより的確な判断ができ、ユーザにフィードバックを返すことができます。また、参照を検索しない選択をしたり、検索の前にユーザに確認メッセージを表示することもできます。
- 多くの場合、参照の方がチェーンよりもリソースを効果的に使用できます。

  チェーンでは、エントリの多い検索操作が要求されると、ネットワーク全体に 2 度送信される場合があります。1 度目はデータのあるサーバからチェーンを実行するサーバへの送信です。2 度目はチェーンを実行するサーバからクライアントへの送信です。

  参照では、クライアントはデータのあるサーバから、1 回の送信で直接データを受け取ることができます。
- エントリの格納場所がわかっていれば、クライアントは直接データのあるサーバにアクセスすることができます。

  チェーンでは、クライアントは詳細を見ることができません。データの出所がわからない場合、そのデータを保持しているサーバにクライアントが直接アクセスすることはありません。

参照には次の欠点があります。

- クライアントが参照とその検索方法を認識できる能力が必要です。

- LDAPv2 クライアントでは結果が認識されないか、認識に古い非標準の方法が使用されます。
- すべてのOpenText eDirectoryパーティションにLDAPサーバのサービスが適用されていなくてはなりません。

  適用されていない場合、参照結果をパーティションのデータに送信できません。

上方参照 ― 通信中のサーバのデータよりもツリーの高い位置にあるデータを持つサーバを参照することです。詳細については、442 ページの 「上方参照を設定する」を参照してください。

上方参照では、マルチベンダーツリーで、上方または隣接した非 OpenText eDirectory パーティションにあるオブジェクトに関する要求が処理されます。

OpenText eDirectory サーバがこのタイプのツリーに含まれるようにするには、eDirectory は「非信頼」とマークされたパーティション内で、階層データを上方に保持するようにします。非信頼領域のオブジェクトは、正しい DN 階層を構築するのに必要なエントリのみから構成されます。これらのエントリは、X.500 の “Glue” エントリに類似しています。

OpenText eDirectory では、知識情報を LDAP 参照データの形式で信頼されていない領域に配置できます。この情報は、LDAP クライアントに参照を返すのに使用します。

LDAP 操作を OpenText eDirectory ツリーの信頼されていない領域で実行すると、LDAP サーバは正しい参照データを検索し、クライアントに参照を返します。

チェーン ― サーバベースのネームレゾリューションプロトコル。

LDAP クライアントが LDAP サーバに要求を送信しますが、サーバは操作のターゲットエントリをローカルで見つけることができません。LDAP サーバ (サーバ A) は OpenText eDirectory ツリーのパーティションおよび他のサーバに関して所有する知識参照を使用して、DN についてより多くの知識を持つ、別の LDAP サーバ (サーバ B) を特定します。LDAP サーバ A は、特定された LDAP サーバ B と通信します。

必要に応じ、サーバ A がエントリのレプリカを持つサーバに接続するまでこの処理が続けられます。その後、OpenText eDirectory は詳細をすべて処理し、操作を完了します。サーバ間の処理はクライアントには表示されないため、クライアントは最初のサーバ A が要求を処理したものと判断します。

LDAP サーバにチェーンを使用した場合、次の利点があります。

- ネームレゾリューションの詳細をすべてクライアントから見えなくします。
- 自動で再認証を行います。
- クライアントのプロキシの役割を果たします。
- OpenText eDirectory ツリーに LDAP サービスをサポートしないサーバがあっても、シームレスに実行されます。

一方、チェーンには次の欠点があります。

- チェーンを使用して名前解決中には、サーバからのフィードバックがなくクライアントが待機する必要がある場合があります。

- LDAP サーバが、WAN リンク経由で多くのエントリを送信するよう要求された場合、処理に長時間かかることがあります。
- ほぼ同じ処理能力を持つサーバがいくつかある場合、別々のサーバが 2 つの要求を同じエントリ上で処理してしまうことがあります。

  OpenText eDirectory は、サーバを接続コスト順にソートしようとします。負荷分散のため、OpenText eDirectory は一番コストの低いサーバの中からランダムに選択を行います。

LDAP と OpenText eDirectory の連携について

このセクションでは次について説明します。

- 377 ページの 「LDAP から OpenText eDirectory に接続する」
- 381 ページの 「クラスと属性のマッピング」
- 384 ページの 「構文の相違」
- 386 ページの 「サポートされる OpenText LDAP コントロールと拡張」

LDAP から OpenText eDirectory に接続する

すべての LDAP クライアントが、次のいずれかのユーザタイプで OpenText eDirectory にバインド (接続) されます。

- ［Public］ユーザ (匿名バインド)
- プロキシユーザ (プロキシユーザ匿名バインド)
- NDS または OpenText eDirectory ユーザ (NDS ユーザバインド)

ユーザの認証に使用されるバインドタイプにより、LDAP クライアントがアクセスできる内容が決定されます。LDAP クライアントは、作成した要求をディレクトリに送信することにより、ディレクトリにアクセスします。LDAP クライアントが LDAP Services for OpenText eDirectory を通じて要求を送信した場合、OpenText eDirectory は、その中から LDAP クライアントが適切なアクセス権を持つ属性の要求だけを処理します。

たとえば LDAP クライアントが、読み込み権が必要なある属性値を要求したものの、その属性についてユーザに許可されているのが比較権だけである場合、この要求は拒否されます。

標準ログイン制限とパスワード制限は引き続き適用されます。ただし、制限はすべて LDAP の実行場所と関係します。時刻およびアドレス制限も適用されますが、アドレス制限は OpenText eDirectory ログインが実行された場所 (この場合は LDAP サーバ) を基準に決定されます。

LDAP クライアントからの書き込みブロック接続のタイムアウトは、NDSD_NLDAP_WRITEBLOCK_TIMEOUT 環境変数を使用して設定できます。この変数は、LDAP クライアントが OpenText eDirectory サーバからデータを読み込むために、クライアントとサーバ間の接続が開いたままになる時間間隔を定義します。接続がブロックされ、LDAP ク

ライアントが指定時間内にデータを読み込めない場合、接続は無効とみなされ、その後閉じられます。この値には、次の例に示すように、この変数の値を秒で指定する必要があります。

NDSD_NLDAP_WRITEBLOCK_TIMEOUT=<seconds>NDSD_NLDAP_WRITEBLOCK_TIMEOUT=60

デフォルト値は 60 秒です。

注 : OpenText eDirectory 9.2.6 以降では、NDSD_NLDAP_WRITEBLOCK_TIMEOUT 環境変数は LDAP クライアントからの読み込みブロック接続と書き込みブロック接続の両方に適用できます。以前のバージョン (OpenText eDirectory 9.2.5) では、この変数は書き込みブロック接続にのみ適用されました。

［Public］ユーザとして接続する

匿名バインドは、ユーザ名またはパスワードを使用しない接続です。サービスで代理ユーザの使用が設定されていない場合、名前とパスワードが定義されていない LDAP クライアントで LDAP Services for OpenText eDirectory にバインドすると、ユーザは OpenText eDirectory に [Public] ユーザとして認証されます。

[Public] ユーザとは、非認証の OpenText eDirectory ユーザのことです。デフォルトでは、［Public］ユーザには OpenText eDirectory ツリー内のオブジェクトのブラウズ権が割り当てられます。［Public］ユーザ用のデフォルトブラウズ権では、OpenText eDirectory オブジェクトをブラウズすることはできますが、ほとんどのオブジェクト属性にアクセスすることはできません。

多くの場合、LDAP クライアントは、デフォルトの［Public］権だけでは不十分です。［Public］の権利は変更できますが、変更した権利はすべてのユーザに対して許可されることになります。この問題を解決するために、プロキシユーザ匿名バインドを使用することをお勧めします。詳細については、378 ページの 「プロキシユーザとして接続する」を参照してください。

［Public］ユーザによるオブジェクト属性へのアクセスを許可するには、[Public] ユーザを該当する (1 つまたは複数の) コンテナのトラスティに設定し、適切なオブジェクト権および属性権を割り当てる必要があります。

プロキシユーザとして接続する

代理ユーザ匿名バインドは、OpenText eDirectory ユーザ名にリンクされた匿名接続です。プロトコルで代理ユーザの使用が設定されている場合、LDAP クライアントが LDAP for OpenText eDirectory に匿名でバインドすると、ユーザは OpenText eDirectory により代理ユーザとして認証されます。LDAP Services for OpenText eDirectory と OpenText eDirectory の両方でユーザ名が設定されます。

通常、匿名バインドには LDAP のポート 389 が使用されます。ただし、ポートはインストール時に手動で変更することができます。

次に、プロキシユーザ匿名バインドの概念について説明します。

- 匿名バインドを経由する LDAP クライアントアクセスは、すべてプロキシユーザオブジェクトを通して割り当てられます。
- 匿名バインドでは LDAP クライアントからパスワードが提供されないので、プロキシユーザに null パスワードを適用し、パスワード変更間隔などのパスワード制限を設定しないようにします。パスワードを強制的に有効期限切れにしたり、プロキシユーザにパスワードの変更を許可したりしないでください。
- プロキシユーザオブジェクトのアドレス制限を設定すると、ユーザがログインできるロケーションを制限できます。
- OpenText eDirectory 内に代理ユーザオブジェクトを作成し、公開する OpenText eDirectory オブジェクトに対する権利を割り当てる必要があります。デフォルトのユーザ権では、特定のオブジェクトと属性だけに対する読み込みアクセス権が許可されます。アクセスする必要がある各サブツリー内のすべてのオブジェクトと属性に対するプロキシユーザ読み込み権および検索権を割り当てます。
- LDAP Services for OpenText eDirectory を設定する LDAP グループオブジェクトの［一般］ページで、代理ユーザオブジェクトを有効化する必要があります。このため、1 つの LDAP グループ内のすべてのサーバに対し、プロキシユーザオブジェクトは 1 つしか作成できません。詳細については、407 ページの 「LDAP オブジェクトを環境設定する」を参照してください。
- プロキシユーザオブジェクトに対し、すべてのプロパティ (デフォルト) か、選択したプロパティの権利を許可できます。

プロキシユーザに対して選択したプロパティの権利だけを許可するには、次を実行します。

1 OpenText Identity Console のホームページで、［**権利の管理**］タイルをクリックします。

2 ［**トラスティ**］をクリックします。

3 トラスティリストを表示するオブジェクトの名前を指定するか、［オブジェクトの検索］( ) を使用してオブジェクトを検索し、［**OK**］をクリックします。

選択したオブジェクトに割り当てられているトラスティのリストが表示されます。

［トラスティの追加 ］をクリックして、オブジェクトに新しいトラスティを割り当てます。リストからトラスティを選択し、 をクリックすると、トラスティを割り当てから削除できます。

4 必要に応じて、詳細を変更するトラスティを選択し、［**割り当てられた権利**］をクリックします。

5 ［**トラスティの変更**］画面で、必要に応じて権利を割り当てて、［**完了**］をクリックします。

6 ［**トラスティの変更**］画面で、［**プロパティの追加** ］をクリックし、［**スキーマ内のすべてのプロパティを表示する**］チェックボックスをオンにします。

7 代理ユーザに継承可能な権利を選択し、必要に応じて権利を割り当てます。

さらに継承可能な権利を追加したい場合は、ステップ 6 とステップ 7 を繰り返します。

リストからプロパティ名を選択して、 をクリックすると、割り当てからプロパティ名を削除できます。

8 ［**完了**］をクリックし、［**適用**］をクリックします。

「リソースが正常に変更されました !」というメッセージが表示されます。

代理ユーザ匿名バインドを実装するには、OpenText eDirectory 内に代理ユーザオブジェクトを作成し、そのユーザに適切な権利を割り当てる必要があります。アクセスする必要がある各サブツリー内のすべてのオブジェクトと属性に対するプロキシユーザ読み込み権および検索権を割り当てます。同じ代理ユーザ名を指定して、LDAP Services for OpenText eDirectory 内で代理ユーザ名を有効化する必要もあります。

1 OpenText Identity Console のホームページで、［**LDAP 環境設定**］タイルをクリックします。

2 ［**オブジェクトの作成** ＋］をクリックします。

3 ［LDAP オブジェクトの作成］ページで、［**名前**］を指定し、［**タイプ**］を選択して、eDirectory ユーザオブジェクトの［**コンテキスト**］を検索します。

4 ［**作成**］をクリックし、［**OK**］をクリックします。

Linux で ldapconfig ユーティリティを使用する

たとえば、LDAP Search Referral Usage で LDAP サーバによる LDAP 参照の処理方法を指定します。

1 システムプロンプトで、次のコマンドを入力します。

```
ldapconfig -s "LDAP:otherReferralUsage=1"
```

2 ユーザ FDN(完全識別 eDirectory ユーザ名) とパスワードを入力します。

NDS または OpenText eDirectory ユーザとして接続する

OpenText eDirectory ユーザバインドは、LDAP クライアントが完全な OpenText eDirectory ユーザ名とパスワードを使用して確立する接続です。OpenText eDirectory ユーザバインドは OpenText eDirectory により認証されます。LDAP クライアントは、その OpenText eDirectory ユーザにアクセスが許可されているすべての情報にアクセスできます。

次に、OpenText eDirectory ユーザバインドの重要な概念について説明します。

- OpenText eDirectory ユーザバインドは、LDAP クライアントに入力されたユーザ名とパスワードを使用して OpenText eDirectory への認証が行われます。
- LDAP クライアントアクセスに使用する OpenText eDirectory ユーザ名とパスワードは、Novell Client が OpenText eDirectory にアクセスする場合も使用できます。
- 非 TLS 接続では、OpenText eDirectory パスワードは LDAP クライアントと LDAP Services for eDirectory の間の経路をクリアテキストデータとして転送されます。

- クリアテキストパスワードが無効にされている場合は、非 TLS 接続上で転送されたユーザ名またはパスワードを含む OpenText eDirectory バインド要求は、すべて拒否されます。
- OpenText eDirectory ユーザパスワードの有効期限が切れた場合、そのユーザの OpenText eDirectory バインド要求は拒否されます。

LDAP クライアントに OpenText eDirectory 権を割り当てる

1. LDAP クライアントが OpenText eDirectory にアクセスするときに使用するユーザ名のタイプを決定します。
   - ［Public］ユーザ (匿名バインド)
   - プロキシユーザ (プロキシユーザ匿名バインド)
   - NDS ユーザ (NDS ユーザバインド)

   詳細については、377 ページの 「LDAP から OpenText eDirectory に接続する」を参照してください。
2. ユーザが 1 つの代理ユーザ、または複数の OpenText eDirectory ユーザ名を使用して LDAP にアクセスする場合は、LDAP を使用してこれらのユーザ名を OpenText eDirectory に作成します。
3. LDAP クライアントが使用するユーザ名に、適切な OpenText eDirectory 権を割り当てます。

ほとんどのユーザに割り当てられるデフォルトの権利では、ユーザ自身が持つオブジェクト以外にはアクセスできません。別のオブジェクトやその属性にアクセスするには、OpenText eDirectory で割り当てられた権利を変更する必要があります。

LDAP クライアントから OpenText eDirectory オブジェクトおよび属性へのアクセスが要求されると、OpenText eDirectory は、LDAP クライアントの OpenText eDirectory 識別情報に基づいて要求を受諾または拒否します。識別情報はバインド時に設定されます。

クラスと属性のマッピング

クラスとは、ディレクトリ内のオブジェクトのタイプ (ユーザ、サーバ、グループなど) です。属性とは、特定のオブジェクトについての追加情報を定義するディレクトリ要素です。たとえば、ユーザオブジェクト属性にはユーザの姓、電話番号などがあります。

スキーマとは、ディレクトリで使用できるクラスと属性、およびディレクトリ構造 (クラス間の相互関係) を定義する一連の規則です。LDAP ディレクトリと OpenText eDirectory ディレクトリのスキーマが異なる場合は、LDAP クラスと属性を、適切な OpenText eDirectory オブジェクトと属性へマッピングしなければならない場合があります。これらのマッピングで、LDAP スキーマから OpenText eDirectory スキーマへの名前の変換を定義します。

LDAP Services for OpenText eDirectory にはデフォルトマッピングがあります。LDAP クラスおよび属性と、OpenText eDirectory オブジェクトタイプおよびプロパティとの対応関係は多くの場合論理的で、直観的に理解できます。ただし、実装時の条件によっては、クラスと属性のマッピングを再設定する必要が生じることもあります。

ほとんどの場合、LDAP クラスと OpenText eDirectory オブジェクトタイプとの間のマッピングは、一対一の対応関係です。ただし、LDAP スキーマでは、同じ属性を意味する CN および共通名のような別名もサポートされています。

LDAP グループ属性をマッピングする

デフォルトの LDAP Services for OpenText eDirectory 環境設定には、事前定義されたクラスと属性のマッピングがあります。これは、LDAP 属性のサブセットから OpenText eDirectory 属性のサブセットへのマッピングです。デフォルト環境設定でマッピングされていない属性には、自動生成されたマッピングが割り当てられます。スキーマ名がスペースまたはコロンを含まない有効な LDAP 名である場合は、マッピングは必要ありません。クラスおよび属性のマッピングを調べて、必要に応じて再設定します。

1 OpenText Identity Console のホームページで、[**LDAP 環境設定**] タイルをクリックします。

2 [**LDAP グループ**] をクリックします。

3 [LDAP グループの変更] ページで、[**属性マップ**] ドロップダウンをクリックします。

4 必要に応じて属性を追加、削除、または変更します。

LDAP 属性の種類によっては別名 (CN および共通名など) が存在する場合があるため、複数の LDAP 属性を対応する 1 つの OpenText eDirectory 属性名にマッピングすることが必要な場合もあります。LDAP Services for OpenText eDirectory が LDAP 属性情報を返す場合、リスト内で検出された最初の一致する属性の値が返されます。

複数の LDAP 属性を 1 つの OpenText eDirectory 属性にマッピングする場合は、属性の順序に意味があるため、リストの順序を変更して属性の優先順位を決定してください。

5 [**保存**] をクリックします。

LDAP グループのクラスマッピング

LDAP クライアントが LDAP サーバに LDAP クラス情報を要求すると、サーバは対応する OpenText eDirectory クラス情報を返します。デフォルトの LDAP Services for OpenText eDirectory 環境設定には、事前定義されたクラスと属性のマッピングがあります。

注 : OpenText eDirectory では、LDAP グループオブジェクトのクラスマッピングは LDAP サーバ間では伝搬しません。複数のサーバ上で同じクラスマッピングを使用するには、環境内のすべての LDAP グループオブジェクトに、マッピングを手動で追加します。

---

1 OpenText Identity Console のホームページで、[**LDAP 環境設定**] タイルをクリックします。

2 [**LDAP グループ**] をクリックします。

3 [LDAP グループの変更] ページで、[**クラスマップ**] ドロップダウンをクリックします。

4 必要に応じてクラスを追加、削除、または変更します。

デフォルトの LDAP Services for OpenText eDirectory 環境設定には、事前定義されたクラスと属性のマッピングがあります。これは、LDAP クラスと属性のサブセットから OpenText eDirectory クラスと属性のサブセットへのマッピングです。デフォルト環境設定でマッピングされていない属性またはクラスには、自動生成されたマッピングが割り当てられます。

スキーマ名がスペースまたはコロンを含まない有効な LDAP 名である場合は、マッピングは必要ありません。クラスおよび属性のマッピングを調べて、必要に応じて再設定します。

5 [**保存**] をクリックします。

LDAP クラスと属性をマッピングする

LDAP ディレクトリと OpenText eDirectory ディレクトリのスキーマは異なるため、LDAP クラスと属性を、適切な OpenText eDirectory オブジェクトと属性へマッピングする必要があります。これらのマッピングで、LDAP スキーマから OpenText eDirectory スキーマへの名前の変換を定義します。

有効な LDAP スキーマ名であれば、スキーマエントリに対する LDAP スキーママッピングは必要ありません。LDAP では、スキーマ名で使用できる文字は英数字とハイフン記号 (-) だけです。LDAP スキーマ名ではスペースは使用できません。

スキーマを LDAP の外部に拡張する場合、.sch ファイルなど LDAP の外部にスキーマを拡張した後でオブジェクト ID による検索を確実に実行するには、LDAP サーバ環境設定をリフレッシュする必要があります。

多対一マッピング

OpenText eDirectory から LDAP をサポートするために、LDAP Services は、(ディレクトリサービスレベルではなく) プロトコルレベルのマッピングを使用して、LDAP と OpenText eDirectory の間で属性とクラスを変換します。したがって、2 つの LDAP クラスまたは属性を同じ OpenText eDirectory クラスまたは属性にマッピングできます。

たとえば、LDAP を使用して Cn を作成し、CommonName=Value を検索すると、Cn と属性値が同じ可能性のある commonName が返されます。

すべての属性を要求すると、そのクラスのマッピングリストの最初にある属性が返されます。名前で属性を要求すると、正しい名前が返されます。

多対一クラスマッピング

LDAP クラス名	eDirectory クラス名
alias aliasObject	別名
groupOfNames groupOfUniqueNames group	グループ
mailGroup rfc822mailgroup	NSCP:mailGroup1

多対一属性マッピング

LDAP 属性名	eDirectory 属性名
c countryName	C
cn commonName	CN
uid userId	uniqueID (固有 ID)
description multiLineDescription	説明
l localityname	L
member uniqueMember	Member (メンバー)
o organizationname	O
ou organizationalUnitName	OU
sn surname	Surname (名字)
st stateOrProvinceName	S
certificateRevocationList;binary certificateRevocationList	ndspkiCertificateRevocationList
authorityRevocationList;binary authorityRevocationList	authorityRevocationList
deltaRevocationList;binary deltaRevocationList	deltaRevocationList
cACertificate;binary cACertificate	cACertificate
crossCertificatePair;binary crossCertificatePair	crossCertificatePair
userCertificate;binary userCertificate	userCertificate

注 : ;binary の付いた属性はセキュリティに関連しています。これらは、アプリケーションが ;binary を付けて取得された名前を必要とする場合のために、マッピングテーブル内に存在します。;binary を付けずに取得した名前が必要な場合は、マッピングの順序を変更できます。

構文の相違

LDAP と OpenText eDirectory では使用される構文が異なります。次のような重要な相違点があります。

- 385 ページの 「コンマ」
- 385 ページの 「タイプ付きの名前」
- 385 ページの 「エスケープ文字」
- 385 ページの 「複数のネーミング属性」

コンマ

LDAP では、区切り記号としてピリオドではなくコンマを使用します。たとえば、OpenText eDirectory の識別名 (完全名) は次のように記述します。

CN=JANEB.OU=MKTG.O=EMA

LDAP 構文を使用すると、同じ識別名は次のようになります。

CN=JANEB,OU=MKTG,O=EMA

また、次は別の LDAP 識別名の例です。

CN=Bill Williams,OU=PR,O=Bella Notte Corp
CN=Susan Jones,OU=Humanities,O=University College London,C=GB

タイプ付きの名前

OpenText eDirectory では、タイプなしの名前 (.JOHN.MARKETING.ABCCORP) とタイプ付きの名前 (CN=JOHN.OU=MARKETING.O=ABCCORP) の両方を使用します。LDAP では、区切り記号としてコンマを使用したタイプ付きの名前 (CN=JOHN,OU=MARKETING,O=ABCCORP) だけを使用します。

エスケープ文字

LDAP 識別名では、エスケープ文字として円記号 (\) を使用します。1 つの円記号とプラス記号 (+) またはコンマ (,) を指定すると、識別名を拡張できます。

次に例を示します。

CN=Pralines\+Cream,OU=Flavors,O=MFG (CN is Pralines+Cream)

CN=D. Cardinal,O=Lionel\,Turner and Kaye,C=US (O は Lionel、Turner、および Kaye)

詳細については、Internet Engineering Task Force の RFC 2253 (http://www.ietf.org/rfc/rfc2253.txt?number=2253) を参照してください。

複数のネーミング属性

オブジェクトは、スキーマ内の複数のネーミング属性を使用して定義できます。LDAP と OpenText eDirectory のユーザオブジェクトには、いずれも CN と UID の 2 つのネーミング属性があります。識別名の中のネーミング属性は、プラス記号 (+) で区切ります。属性に明示的なラベルが付いていない場合は、スキーマによりそれぞれの文字列に対応する属性が決定されます (eDirectory と LDAP の両方で、最初の文字列は CN、次の文字列は UID になります)。識別名の中の各部分に手動でラベルを付けると、ネーミング属性の順序を変更できます。

2 つの相対識別名の例を次に示します。

Smith (CN は Smith CN=Smith)

Smith+Lisa (CN は Smith、UID は Lisa CN=Smith UID=Lisa)

2 つの相対識別名 (Smith と Smith+Lisa) は、2 つの異なる相対識別名によって参照されるため、同じコンテキスト内に共存することができます。

サポートされる OpenText LDAP コントロールと拡張

LDAP クライアントと LDAP サーバは、LDAP 3 プロトコルを使用することにより、コントロールと拡張を適用して LDAP 操作を拡張できます。コントロールと拡張を使用することによって、要求や応答の一部として追加情報を指定できます。拡張された各操作は、自身の属性または objectclass を LDAP サーバに追加するのに必要な、オクテット数値の文字列であるオブジェクト識別子 (OID) により識別されます。LDAP クライアントは、実行するべき拡張操作の OID、およびその拡張操作に固有なデータを指定して、拡張操作要求を送信できます。LDAP サーバはこの要求を受信すると、拡張操作を実行し、OID と追加データが設定された応答をクライアントに送信します。

たとえば、クライアントがサーバに検索要求を送信するとき、ソートを指定するコントロールをこの要求に入れることができます。サーバはこの検索要求を受け取ると、検索結果をソートしてから、その結果をクライアントに戻します。コントロールはサーバからクライアントに送ることもできます。たとえば、サーバは、クライアントにパスワード期限切れを通知する認証要求のコントロールを送ることができます。

デフォルトでは、起動直後の OpenText eDirectory LDAP サーバは、すべてのシステム拡張ならびに選択されたオプション拡張およびコントロールをロードできる状態にあります。オプション拡張に対応する LDAP サーバオブジェクトの extensionInfo 属性により、システム管理者は、オプション拡張およびコントロールの選択と選択解除ができます。

拡張操作を有効にするため、LDAP 3 プロトコルはルート DSE 内の supportedControl 属性および supportedExtension 属性に含まれる、サポートされているコントロールと拡張のリストをサーバに要求します。ルート DSE (DSE は DSA (Directory System Agent) Specific (固有) Entry(エントリ) の略) とは、ディレクトリ情報ツリー (DIT) のルートにあるエントリです。詳細については、449 ページの 「LDAP サーバの情報を取得する」を参照してください。

サポートされている LDAP コントロールと拡張のリストについては、『LDAP and OpenText eDirectory Integration NDK』の「LDAP Controls (https://www.novell.com/documentation/developer/ldapover/ldap_enu/data/cchbehhc.html) 」と「LDAP Extensions (https://www.novell.com/documentation/developer/ldapover/ldap_enu/data/a6ik7oi.html)」を参照してください。

Linux での LDAP ツールの使用

OpenText eDirectory には、LDAP ディレクトリサーバを管理するのに役立つ次の LDAP ツールが含まれています。これらのツールは、/opt/novell/eDirectory/bin に格納されています。

注 : OpenText eDirectory 9.0 以降、.PEM 証明書は特定の TLS 変数経由で渡されます。これらの変数は /etc/opt/novell/eDirectory/conf/openldap/ldap.conf ファイルで定義するか、個別にエクスポートできます。詳細については、OpenLdap マニュアルの Web サイトおよびマニュアルページを参照してください。

ツール	説明
ice	エントリをファイルから LDAP ディレクトリにインポートし、ファイルのディレクトリ内のエントリを変更し、エントリをファイルにエクスポートし、ファイルの属性とクラス定義を追加します。
ldapadd	LDAP ディレクトリに新しいエントリを追加します。
ldapdelete	LDAP ディレクトリサーバからエントリを削除します。ldapdelete ツールは、LDAP サーバとの接続を開始し、エントリのバインドと削除を行います。
ldapmodify	LDAP サーバとの接続を開始し、エントリのバインド、変更、追加を行います。
ldapmodrdn	LDAP ディレクトリサーバ内のエントリの相対識別名 (RDN) を変更します。LDAP サーバとの接続を開始し、エントリの RDN のバインドと変更を行います。
ldapsearch	LDAP ディレクトリサーバでエントリを検索します。LDAP サーバとの接続を開始し、バインドを行い、指定されたフィルタを使用して検索を実行します。フィルタは、RFC 2254 (http://www.ietf.org/rfc/rfc2254.txt) で定義された LDAP フィルタの文字列表現に準拠している必要があります。
ndsindex	インデックスの作成、一覧表示、一時停止、再開、削除を行います。

詳細については、『*LDAP Libraries for C Doc*』の「LDAP Tools (https://www.novell.com/documentation/developer/cldap/)」を参照してください。

LDAP ツールを安全に実行するには、「Linux コンピュータ上の OpenText eDirectory 操作をセキュリティで保護する」を参照し、安全に OpenText eDirectory と LDAP を接続するためのコマンドラインによる LDAP 操作に PEM ファイルを指定します。

LDAP ツール

LDAP ユーティリティは、エントリの削除、変更、追加、スキーマの拡張、相対識別名の変更、エントリの新規コンテナへの異動、検索インデックスの作成、検索の実行に使用できます。

注 : RFC 2256 に準拠するため、OpenText eDirectory の LDAP インタフェースでは、パスワードの長さが 128 文字以下の場合にのみ、バインドが許可されます。また、パスワードを LDAP から設定する場合、最大 128 文字のパスワードのみ設定できます。

ldapadd

ldapadd ユーティリティを使用して、新しいエントリを追加します。構文は次のとおりです。

ldapadd [-c] [-C] [-l] [-M] [-P] [-r] [-n] [-v] [-F] [-l *limit*] [-M[M]] [-d *debuglevel*] [-D *binddn*] [[-W]| [-w *passwd*]] [-h *ldaphost*] [-p *ldapport*] [-P *version*] [-Z[Z]] [-f *file*]

-f オプションが指定されていると、ldpadd は変更をファイルから読み出します。-f オプションが指定されていないと、ldpadd は変更を stdin から読み出します。

ヒント : LDAP ユーティリティからの出力は stdout に送られます。出力を見る前にユーティリティが終了する場合は、出力をファイルにリダイレクトしてください。例 : ldapadd [options] > out.txt

オプション	説明
-a	新しいエントリを追加します。ldapmodify のデフォルトは、既存エントリの変更です。ldapadd として呼び出されると、このフラグは常にオンになります。
-r	デフォルトでは既存の値を置換します。
-c	連続操作モード。エラーが通知されても、ldapmodify は変更動作を継続します。デフォルトでは、エラーが通知されると終了します。
-f *file*	エントリ情報を標準入力ではなく LDIF ファイルから読み出します。レコードの最大長は 4096 行です。
-F	replica: で始まる入力行の内容に関わりなく、すべての変更を強制的に適用します。デフォルトでは、replica: 行が使用中の LDAP サーバホストおよびポートと照らして比較され、replog レコードを実際に適用するべきかどうかが判断されます。

すべての LDAP ツールの共通オプション

すべての LDAP ツールに共通するオプションがいくつかあります。次の表は、これらのオプションを表示したものです。

オプション	説明
-C	後に続く参照 (匿名バインド) を有効にします。
-d *debuglevel*	LDAP デバッグレベルを debuglevel に設定します。このオプションを有効にするには、ldapmodify をコンパイルするときに LDAP_DEBUG の定義が必要です。
-D *binddn*	binddn を使用して LDAP ディレクトリにバインドします。*binddn* には、RFC 1779 に定義されている文字列表現の DN を指定します。
-f *file*	ファイルから行を読み取り、1 行ごとに LDAP 検索を実行します。この場合、コマンドラインで指定したフィルタは、%s が最初に出現した個所がファイルの行に置換されるというパターンとして機能します。1 つのハイフン (-) 文字がファイルとして指定された場合には、標準入力から行が読み取られます。

オプション	説明
-h *ldaphost*	LDAP サーバの実行場所となっている代替ホストを指定します。
-l *limit*	接続タイムアウト (秒) を指定します。
-M	Manage DSA IT コントロール (非クリティカル) を有効にします。
-MM	Manage DSA IT コントロール (クリティカル) を有効にします。
-n	完了した場合の結果を表示しますが、実際にはエントリを変更しません。-v と組み合わせて使用すると、デバッグ時に便利です。
-p *ldapport*	LDAP サーバがリスンする代替 TCP™ ポートを指定します。
-P *version*	LDAP のバージョン (2 または 3) を指定します。
-v	冗長モードが設定され、多くの診断メッセージが標準出力に書き込まれます。
-w *passwd*	簡易認証のパスワードとして、*passwd* を使用します。
-W	簡易認証のプロンプトです。コマンドラインにパスワードを指定するかわりに、このオプションが使用されます。
-Z	操作をバインドして実行する前に、TLS を開始します。TLS を開始する操作の途中でエラーが発生すると、エラーは無視され操作は続行されます。エラーが発生した場合に操作を中止するには、このオプションではなく -ZZ オプションの使用をお勧めします。 このオプションでポートが指定されている場合、そのポートはクリアテキスト接続を受信する必要があります。 サーバの識別情報を確認するには、このオプションを -e オプションと組み合わせて使用し、サーバ証明書ファイルを指定する必要があります。TLS を開始すると、これによりサーバのルート認証局証明書が確認されます。-e オプションが指定されていない場合、サーバからのすべての証明書が許可されます。
-ZZ	操作をバインドして実行する前に、TLS を開始します。TLS を開始する操作の途中でエラーが発生すると、操作は中止されます。 このオプションでポートが指定されている場合、そのポートはクリアテキスト接続を受信する必要があります。 サーバの識別情報を確認するには、このオプションを -e オプションと組み合わせて使用し、サーバ証明書ファイルを指定する必要があります。TLS を開始すると、これによりサーバのルート認証局証明書が確認されます。-e オプションが指定されていない場合、サーバからのすべての証明書が許可されます。

例

/tmp/entrymods ファイルが存在し、その内容が次のようであると仮定します。

```
dn: cn=Modify Me, o=University of Michigan, c=US
changetype: modify
```

```
replace: mail
mail: modme@terminator.rs.itd.umich.edu
-
add: title
title: Manager
-
add: jpegPhoto
jpegPhoto: /tmp/modme.jpeg
-
delete: description
-
```

この場合、コマンド ldapmodify -b -r -f /tmp/entrymods は、Modify Me エントリの mail 属性の中身を modme@terminator.rs.itd.umich.edu の値に置き換え、Manager というタイトルを追加し、/tmp/modme.jpeg ファイルの内容を jpegPhoto として追加し、description 属性を完全に削除します。

このような変更は、以下のように ldapmodify の古い入力規則を使用して実行することもできます。

```
cn=Modify Me, o=University of Michigan, c=US
mail=modme@terminator.rs.itd.umich.edu
+title=Manager
+jpegPhoto=/tmp/modme.jpeg
-description
```

コマンドは次のようになります。

```
ldapmodify -b -r -f /tmp/entrymods
```

/tmp/newentry ファイルが存在し、その内容が次のようであると仮定します。

```
dn: cn=Barbara Jensen, o=University of Michigan, c=US
objectClass: person
cn: Barbara Jensen
cn: B Jensen
sn: Jensen
title: Manager
mail: bjensen@terminator.rs.itd.umich.edu
uid: bjensen
```

この場合、/tmp/newentry ファイルの値を使用して、コマンド ldapadd -f /tmp/entrymods は B Jensen の新しいエントリを追加します。

/tmp/newentry ファイルが存在し、その内容が次のようであると仮定します。

```
dn: cn=Barbara Jensen, o=University of Michigan, c=US
changetype: delete
```

この場合、コマンド ldapmodify -f /tmp/entrymods は B Jensen のエントリを削除します。

ldapdelete

ldapdelete ユーティリティは、指定したインデックスを削除します。LDAP サーバとの接続を開始し、バインドしてから削除します。構文は次のとおりです。

```
ldapdelete [-n] [-v] [-c] [-r] [-l] [-C] [-M] [-d debuglevel] [-f file] [-
D binddn] [[-W]| [-w passwd]] [-h ldaphost] [-p ldapport] [-Z[Z]] [dn]...
```

dn パラメータは、削除するエントリの識別名のリストです。

これは、-f オプションと次の方法でやり取りします。

- コマンドラインに -f オプションがなく、コマンドラインで DN が指定されている場合、ユーティリティにより指定したエントリが削除されます。
- コマンドラインに dn と -f の両方がある場合、ユーティリティは削除する DN をファイルから読み込んで、コマンドラインの DN は無視します。
- コマンドラインに dn と -f オプションがない場合、ユーティリティは stdin から dn を読み込みます。

ヒント : LDAP ユーティリティからの出力は stdout に送られます。出力を見る前にユーティリティが終了する場合は、出力をファイルにリダイレクトしてください。例 : ldapdelete [options] > out.txt

オプション	説明
-c	連続操作モード。エラーが通知されても、ldapdelete は削除動作を継続します。デフォルトでは、エラーが通知されると終了します。
-f *file*	ファイルから行を読み取り、1 行ごとに LDAP 検索を実行します。この場合、コマンドラインで指定したフィルタは、%s が最初に出現した個所がファイルの行に置換されるというパターンとして機能します。
-r	再起的に削除します。

注 : 共通オプションについての詳細は、388 ページの 「すべての LDAP ツールの共通オプション」を参照してください。

例

ldapdelete のコマンド「cn=Delete Me, o=University of Michigan, c=US」では、「University of Michigan」組織のエントリの真下にある commonName「Delete Me」で指定されたエントリを削除します。この場合、削除が許可されるためには binddn および passwd を指定する必要があります (-D オプションおよび -w オプションを参照)。

ldapmodify

ldapmodify ユーティリティを使用すると、既存エントリの属性を変更したり、新規エントリを追加することができます。構文は次のとおりです。

```
ldapmodify [-a] [-c] [-C] [-M] [-P] [-r] [-n] [-v] [-F] [-l limit] [-M[M]]
[-d debuglevel] [-D binddn] [[-W]|[-w passwd]] [-h ldaphost] [-p ldap-port]
[-P version] [-Z[Z]] [-f file]
```

-f オプションを指定すると、ldapmodify により変更がファイルから読み出されます。-f オプションを指定しない場合、変更は stdin から読み出されます。

ヒント : LDAP ユーティリティからの出力は stdout に送られます。出力を見る前にユーティリティが終了する場合は、出力をファイルにリダイレクトしてください。例 : ldapmodify [options] > out.txt

オプション	説明
-a	新しいエントリを追加します。ldapmodify のデフォルトは、既存エントリの変更です。ldapadd として呼び出されると、このフラグは常にオンになります。
-r	デフォルトでは既存の値を置換します。
-c	連続操作モード。エラーが通知されても、ldapmodify は変更動作を継続します。デフォルトでは、エラーが通知されると終了します。
-f *file*	エントリ情報を標準入力ではなく LDIF ファイルから読み出します。レコードの最大長は 4096 行です。
-F	replica: で始まる入力行の内容に関わりなく、すべての変更を強制的に適用します。デフォルトでは、replica: 行が使用中の LDAP サーバホストおよびポートと照らして比較され、replog レコードを実際に適用するべきかどうかが判断されます。

注 : 共通オプションについての詳細は、「388 ページの 「すべての LDAP ツールの共通オプション」」を参照してください。

ldapmodrdn

ldapmodrdn を使用すると、エントリの相対識別名を変更できます。また、エントリを新しいコンテナに移動することもできます。構文は次のとおりです。

```
ldapmodrdn [-r] [-n] [-v] [-c] [-C] [-l] [-M] [-s newsuperior] [-d
debuglevel] [-D binddn] [[-W]|[-w  passwd]]  [-h ldaphost] [-p ldapport] [-
Z[Z]] [-f file] [dn newrdn]
```

注 : LDAP ユーティリティからの出力は stdout に送られます。出力を見る前にユーティリティが終了する場合は、出力をファイルにリダイレクトしてください。例 : ldapmodrdn [options] > out.txt

オプション	説明
-c	連続操作モード。エラーが通知されても、ldapmodify は変更動作を継続します。デフォルトでは、エラーが通知されると終了します。
-f *file*	エントリ変更情報を標準入力やコマンドラインではなくファイルから読み出します。古い RDN と新規 RDN の間に空白行がないことを確認します。空白行があると、-f オプションは失敗します。
-r	エントリから旧 RDN 値を削除します。デフォルトでは、以前の値が保持されます。
-s *newsuperior*	エントリの移動先のコンテナの識別名を指定します。

注 : 共通オプションについての詳細は、(388 ページ) すべての LDAP ツールの共通オプションを参照してください。

例

/tmp/entrymods ファイルが存在すると仮定すると、次のような内容になります。

```
cn=Modify Me, o=University of Michigan, c=US

cn=The New Me
```

ldapsearch

ldapsearch ユーティリティは、指定された属性とオブジェクトクラスのディレクトリを検索します。構文は次のとおりです。

```
ldapsearch [-n] [-u] [-v] [-t] [-A] [-T] [-C] [-V] [-M] [-P] [-L] [-d
debuglevel] [-f file] [-D binddn] [[-W]| [-w bindpasswd]] [-h ldaphost] [-
p ldapport] [-b searchbase] [-s scope] [-a deref] [-l time limit] [-z size
limit] [-Z[Z]] filter [attrs....]
```

ldapsearch ツールは LDAP サーバとの接続を開始し、バインドを行い、フィルタを使用して検索を実行します。フィルタは、RFC 2254 (http://www.ietf.org/rfc/rfc2254.txt) で定義された LDAP フィルタの文字列表現に準拠している必要があります。

ldapsearch が 1 つ以上のエントリを検出すると、attrs で指定された属性が取得され、エントリと値が標準出力に書き込まれます。属性がリストされない場合、すべての属性が戻ります。

ヒント : LDAP ユーティリティからの出力は stdout に送られます。出力を見る前にユーティリティが終了する場合は、出力をファイルにリダイレクトしてください。例 : ldapsearch [options] filter [attribute list] > out.txt

オプション	説明
-a *deref*	別名の逆参照の処理方法を指定します。次の値を使用します。 ◆［Never］: ベースオブジェクトを探すときと検索するときの両方で、別名の逆参照は行われません。 ◆［Always］: ベースオブジェクトを探すときと検索するときの両方で、別名の逆参照を常に行います。 ◆［Search］: ベースオブジェクトの従属を検索するときは別名の逆参照を行いますが、ベースオブジェクトを探すときには行いません。 ◆［Find］: ベースオブジェクトを探すときは別名の逆参照を行いますが、ベースオブジェクトのサブオーディネートを従属するときには行いません。
-A	値ではなく、属性のみ取り込まれます。エントリに属性が存在するかどうかを確認して、具体的な属性値を知る必要がない場合に便利です。
-CC	後に続く参照 (同じバインド DN とパスワードで認証されたバインド) を有効にします。
-b *searchbase*	*searchbase* を検索の開始ポイントとして使用します。
-L	エントリを LDIF 形式で出力します。
-LL	エントリを LDIF 形式で出力します。コメントは出力されません。
-LLL	エントリを LDIF 形式で出力します。コメントおよびバージョンは出力されません。
-s *scope*	検索のスコープを指定します。スコープとして、ベースオブジェクトを示す「base」、1 レベルを示す「one」、またはサブツリー検索を示す「sub」を指定します。デフォルトは「sub」です。
-S *attribute*	戻されたエントリを属性に基づいてソートします。デフォルトでは、戻ったエントリのソートを行いません。属性が長さ 0 の文字列 ("") の場合、エントリはその識別名のコンポーネントによりソートされます。詳細については、ldap_sort を参照してください。通常は、ldapsearch はエントリを受け取った順に出力します。-S オプションを指定するとデフォルトが無効になり、すべてのエントリが取得され、ソートされてから出力されます。
-t	検索されたバイナリ値が一時ファイルに書き込まれます。これは、jpeg の写真やオーディオなど ASCII 以外の値を扱うときに便利です。
-tt	すべての値が一時ファイルに書き込まれます。

オプション	説明
-T *path*	ファイルをパス (デフォルト : /tmp) で指定されたディレクトリに書き出します。
-u	識別名 (DN) をユーザにわかりやすい形式で出力します。
-V	ファイルの URL プリフィックスです。
-V *prefix*	ファイルの URL プレフィックスを指定します (デフォルト : file:// tmp/)。
-z *sizelimit*	検索が終了するまで最大 *sizelimit* エントリだけ待機します。

注 : 共通オプションについての詳細は、「388 ページの 「すべての LDAP ツールの共通オプション」」を参照してください。

例

次のコマンドを実行します。

```
ldapsearch "cn=mark smith" cn telephoneNumber
```

commonName「mark smith」のエントリのサブツリー検索 (デフォルトの検索ベースを使用) を実行します。commonName の値および telephoneNumber の値が取得され、標準出力に表示されます。2 つのエントリが検出された場合、次のように出力されます。

```
cn=Mark D Smith, ou="College of Literature, Science, and the Arts",
ou=Students, ou=People, o=University of Michigan, c=US

cn=Mark Smith

cn=Mark David Smith

cn=Mark D Smith 1

cn=Mark D Smith

telephoneNumber=+1 313 930-9489

cn=Mark C Smith, ou=Information Technology Division, ou=Faculty and Staff,
ou=People,o=University of Michigan, c=US

cn=Mark Smith

cn=Mark C Smith 1

cn=Mark C Smith

telephoneNumber=+1 313 764-2277
```

コマンド :

```
ldapsearch -u -t "uid=mcs" jpegPhoto audio
```

デフォルトの検索ベースを使用して、ユーザ ID「mcs」のエントリのサブツリー検索を実行します。エントリの DN は、DN 自体を含む行の後にわかりやすい形式で出力されます。また、Jpeg の写真の値およびオーディオの値が取得され、一時ファイルに書き込まれます。要求された属性がそれぞれ 1 つの値をもつエントリが 1 つ検出された場合、次のように出力されます。

```
cn=Mark C Smith, ou=Information Technology Division, ou=Faculty and Staff,
ou=People, o=University of Michigan, c=US

Mark C Smith, Information Technology Division, Faculty and Staff, People,
University of Michigan, US

audio=/tmp/ldapsearch-audio-a19924

jpegPhoto=/tmp/ldapsearch-jpegPhoto-a19924
```

次のコマンドは、organizationName が「university」で始まるすべての組織の c=US レベルで、1 レベルの検索を実行します。

```
ldapsearch -L -s one -b "c=US" "o=university*" o description
```

検索結果は LDIF 形式で表示されます。organizationName の値および記述属性の値が取得され、標準出力に書き込まれます。出力結果の例は次のようになります。

```
dn: o=University of Alaska Fairbanks, c=US

o: University of Alaska Fairbanks

description: Preparing Alaska for a brave new yesterday.

description: leaf node only

dn: o=University of Colorado at Boulder, c=US

o: University of Colorado at Boulder

description: No personnel information

description: Institution of education and research

dn: o=University of Colorado at Denver, c=US

o: University of Colorado at D
```

ndsindex

ndsindex ユーティリティは、インデックスと複合インデックスを作成、一覧表示、一時停止、再開、または削除します。複合インデックスについては、ndsindex ユーティリティで $ の符号で区切られた複数の属性を指定できます。構文は次のとおりです。

注

- 複合インデックスの場合は複数の属性を指定できます。OpenText では、パフォーマンス向上のため、入力する属性は最大 3 つにすることをお勧めします。値タイプの複合インデックスの場合、最大 5 つの属性を追加できます。
- ndsindex ユーティリティの接続先は、インデックスを追加したのと同じサーバにするようお勧めします。

```
ndsindex list [-h <hostname>] [-p <port>] -D <bind DN> -W|[-w <password>]
[-l limit] -s <eDirectory Server DN> [-Z[Z]] [<indexName1>,
<indexName2>.....]
```

```
ndsindex add -a [-h <hostname>] [-p <port>] -D <bind DN> -W|[-w <password>]
[-l limit] -s <eDirectory Server DN> [-Z[Z]] <indexDefinintion1>
[<indexDefinintion2>.....]
```

注

- -a オプションを使用すると、新しいインデックスの作成時に渡される属性のリストに先祖 ID 属性のプレフィックスを付けることができます。先祖 ID を持つインデックスは value インデックスタイプでのみ作成できます。Presence および Substring のインデックスタイプで先祖 ID を使用することはできません。
- 先祖 ID を持つインデックスを作成すると、データベースサイズが大きくなります。

```
ndsindex delete [-h <hostname>] [-p <port>] -D <bind DN> -W|[-w <password>]
[-l limit] -s <eDirectory Server DN> [-Z[Z]] <indexName1>
[<indexName2>.....]
```

```
ndsindex resume [-h <hostname>] [-p <port>] -D <bind DN> -W|[-w <password>]
[-l limit] -s <eDirectory Server DN> [-Z[Z]] <indexName1>
[<indexName2>.....]
```

```
ndsindex suspend [-h <hostname>] [-p <port>] -D <bind DN> -W|[-w
<password>] [-l limit] -s <eDirectory Server DN> [-Z[Z]] <indexName1>
[<indexName2>.....]
```

オプション	説明
リスト	指定したインデックスを表示します。インデックスが指定されていない場合、ndsindex はサーバ上のすべての既存のインデックスを表示します。
add	新しいインデックスを作成します。
削除	指定したインデックスを削除します。
再開	指定したインデックスをオフラインの状態から再開します。
一時停止	指定したインデックスを一時停止してオフラインの状態にします。
-s *eDirectory Server DN*	eDirectory サーバの DN を指定します。

注 : 共通オプションについての詳細は、「388 ページの 「すべての LDAP ツールの共通オプション」」を参照してください。

例

サーバ MyHost 上のインデックスを表示するには、次のコマンドを入力します。

```
ndsindex list -h MyHost -D cn=admin,o=mycompany -w password -s
cn=MyHost,o=novell
```

電子メールの属性に MyIndex という名前の下位文字列インデックスを作成するには、次のコマンドを入力します。

```
ndsindex add -h myhost -D cn=admin, o=mycompany -w password -s cn=myhost,
o=novell "MyIndex;email address;substring"
```

市町村の属性に、先祖 ID を指定して MyIndex という名前の値インデックスを作成するには、次のコマンドを入力します。

```
ndsindex add -a -h myhost -D cn=admin,o=mycompany -w password -s
cn=myhost,o=novell "MyIndex;city;value"
```

自宅電話番号の属性に MyIndex という名前の存在インデックスを作成するには、次のコマンドを入力します。

```
ndsindex add -h myhost -D cn=admin,o=mycompany -w password -s
cn=myhost,o=novell "MyIndex;homephone;presence"
```

MyIndex という名前のインデックスを削除するには、次のコマンドを入力します。

```
ndsindex delete -h myhost -D cn=admin,o=mycompany -w password -s
cn=myhost,o=novell MyIndex
```

MyIndex という名前のインデックスを一時停止するには、次のコマンドを入力します。

```
ndsindex suspend -h myhost -D cn=admin,o=mycompany -w password -s
cn=myhost,o=novell MyIndex
```

MyIndex という名前のインデックスを再開するには、次のコマンドを入力します。

```
ndsindex resume -h myhost -D cn=admin,o=mycompany -w password -s
cn=myhost,o=novell MyIndex
```

複合インデックスの例

email address と surname の属性に MyIndex という名前の値インデックスを作成するには、次のコマンドを入力します。

```
ndsindex add –h myhost –D cn=admin, o=mycompany –w password –s cn=myhost,
o=netiq 'MyIndex;email address$surname;value'
```

注 : タイプが Presence および Substring の複合インデックスを作成することはできません。

拡張可能一致検索フィルタ

RFC 2251 (http://www.ietf.org/rfc/rfc2251.txt) で定義される LDAP 3 のコアプロトコルを指定するには、LDAP サーバが拡張可能一致検索フィルタ機能を認識できなくてはなりません。拡張可能一致検索では、LDAP クライアントは検索フィルタに次の項目を指定することができます。

- オプションの属性名
- オプションの一致ルール
- DN 属性がエントリの一部として考慮されるべきかを示すフラグ
- 一致検索に使用される値

拡張可能一致検索フィルタの文字列を次に示します。

```
extensible = attr [":dn"] [":" matchingrule] ":=" value /
             [:dn"] ":" matchingrule ":=" value
```

次の表は、拡張可能検索フィルタパラメータを表示したものです。

パラメータ	説明
attr	適合する属性を指定します。
[":*dn*"]	一致ルールが比較一致に含まれるていることを示します。
[":" *matchingrule*]	使用する一致ルールを指定します。
":="	一致ルールを指定しないと、完全一致とみなされます。
value	比較値

extensibleMatch は、LDAP 3 から導入された新しいフィルタです。matchingRule フィールドがない場合は、属性フィールドが必ず必要です。完全一致検索はその属性に対して実行されます。attribute フィールドがなく、matchingRule が存在する場合、その matchingRule をサポートするエントリ内のすべての属性が matchValue と比較され、matchingRule によりアサーション値の構文が決定されます。

フィルタ項目は次のように評価されます。

- TRUE: エントリに 1 件以上の一致があることを表します。
- FALSE: エントリに一致する属性がないことを表します。
- matchingRule を認識できなかったり、assertionValue を解析できない場合は「未定義」とされます。

matchingRule の他に type フィールドがある場合、matchingRule はその type で使用できるものでなくてはなりません。使用できない場合、フィルタ項目は定義されません。検索フィルタに :dn が指定されている場合、エントリの識別名に含まれるすべての属性に対して一致検索が適用されます。また、フィルタ項目の評価が TRUE の識別名が 1 つ以上属性を

持っている場合も、評価は TRUE になります。dnAttributes フィールドがあるので、単語の一致検索などで、1 つのルールをエントリに適用し、別のルールをエントリと DN 属性に適用するというように、一般的な一致ルールを複数設定する必要がなくなります。

拡張可能一致検索フィルタにより、LDAP クライアントでは次の 2 つのことが可能になります。

- 同じタイプのデータに対し、複数の一致ルールをサポートできます。
- 検索条件に DN 要素を含めることができます。

  DN 指定により、DN の特定要素の一致検索を実行できます。

OpenText eDirectory 8.7.3 以降のバージョンは、DN 属性の照合用に拡張可能な一致フィルタをサポートします。拡張可能一致検索フィルタのもう一つの要素である一致ルールは未定義とみなされ、無視されます。DN 一致検索を使用すると、LDAP クライアントで OpenText eDirectory ツリーからオブジェクトを簡単に検索できます。たとえば、

```
(&(ou:dn:=sales)(objectclass=user))
```

のような複雑な LDAP 検索フィルタにより、セールスコンテナの下のセールスファンクションにあるすべてのユーザオブジェクトをリストすることができます。

使用例

次は、OpenText eDirectory 8.7.3 以降でサポートされている、拡張可能な一致検索フィルタの文字列の例です。

```
(o:dn:=Ace Industry)
```

これは :dn の使用例です。一致を評価するとき、エントリの識別名の属性はエントリの一部とみなされることを表しています。これは、完全一致であることを意味します。

```
(:dn:2.4.8.10:=Dino)
```

これはエントリの属性に適用するフィルタの例です。一致ルールが 2.4.8.10 の DN の属性も考慮されます。

次は、OpenText eDirectory 8.7.3 以降でサポートされて*いない*拡張可能一致検索検索フィルタの文字列の例です。

```
(cn:1.2.3.4.5:=John Smith)
```

この例は、属性タイプ cn と値 John Smith を指定するフィルタを表しています。一致ルール oid 1.2.3.4.5 により、ディレクトリサーバにより一致検索が実行されます。

```
(sn:dn:2.4.6.8.10:=Barbara Jones)
```

これは :dn の使用例です。比較するときは一致ルール 2.4.6.8.10 を使用する必要があることと、一致を評価するときにはエントリの識別名の属性をエントリの一部とみなす必要があることを示します。

LDAP トランザクション

OpenText eDirectory LDAP サーバは、複数の更新操作を 1 つのアトミックな操作 (トランザクションともいう) にまとめることをサポートします。OpenText eDirectory の LDAP によるトランザクションのサポートは “LDAP Transactions” (http://www.watersprings.org/pub/id/draft-zeilenga-ldap-txn-05.txt) および “LDAP: Grouping of Related Operations” (http://www.watersprings.org/pub/id/draft-zeilenga-ldap-grouping-05.txt) という 2 つのインターネット仕様に基づいています。

LDAP トランザクションを使用することで、LDAP アプリケーションは複数の LDAP 更新操作 (追加、変更、削除、名前変更) をグループとして送信し、この操作のグループ全体をコミットまたは中止することができます。

次のように、LDAP トランザクションのコンテキストで出現するエンティティがいくつかあります。

- CreateGroupingRequest (2.16.840.1.113719.1.27.103.1) – これは関連する操作をグループ化する LDAP 拡張操作です。拡張操作は要求されたグループ化のタイプを識別する createGroupType という値を保持します。LDAP トランザクションの場合、グループ化タイプは transactionGroupingType です。(2.16.840.1.113719.1.27.103.8)
- CreateGroupingResponse (2.16.840.1.113719.1.27.103.1) – これは createGroupingRequest への LDAP サーバの応答であり、groupCookie とオプションの createGroupValue という 2 つの応答フィールドが含まれています。
- GroupingControl (2.16.840.1.113719.1.27.103.7) - これは、このコントロールによって保持される値である groupCookie により、グループ化に対する操作の関連付けを示すために使用されます。
- EndGroupingRequest (2.16.840.1.113719.1.27.103.2) – これは、グループ化要求の終了を示すために使用する、もう 1 つの LDAP 拡張操作です。LDAP トランザクションの場合、これはトランザクションの終了を示し、最終的にトランザクションのコミットまたは中止になります。
- EndGroupingResponse (2.16.840.1.113719.1.27.103.2) – これは、endGroupingResponse に対する LDAP サーバの応答であり、成功または失敗を LDAP クライアントに示します。

LDAP トランザクションで、LDAP サーバと LDAP クライアントがやり取りする要求と応答のシーケンスは次のとおりです。

- サーバでアトミックな操作 (つまりトランザクション) として処理する複数の LDAP 操作をクライアントが送信する場合、最初に createGroupingRequest を送信する必要があります。その際、createGroupType は transactionGroupingType とし、createGroupValue は付けません。
- OpenText eDirectory サーバがトランザクションを処理できる場合、クライアントから要求されたグループ化を固有に識別する groupingCookie とともに、成功の結果コードを返します。処理できない場合、サーバは失敗の理由を示す非成功の結果コードをクライアントに返します。
- クライアントがサーバから成功の結果コードを受け取ると、サーバから返される groupingCookie を含む GroupingControl をその後の更新操作に添付し、1 つのトランザクションの一部として処理することを示します。サーバがトランザクションの一部と

して更新操作を処理できる場合、サーバは成功を返し、この要求をキューに入れます。サーバがトランザクションの一部として更新操作を処理できない場合、サーバは失敗の理由を示す非成功の結果コードをクライアントに返します。

- クライアントがグループ化コントロールを添付してすべての更新操作をサーバに送信した後、クライアントは groupingCookie とともに endGroupingRequest をサーバに送信し、トランザクションを終了することを通知します。endGroupValue がない場合はコミット要求を示し、一方、空の endGroupValue は中止要求を示します。
- サーバは、1 つのトランザクション内の保留中のすべての操作を適用します。成功すると、成功を返します。成功しなかった場合、非成功の結果コードを返します。
- クライアントとサーバ間の上記のやり取りの間のいずれかの時点で、サーバがトランザクションの指定の処理を続行できない場合、サーバは endGroupingNotice (2.16.840.1.113719.1.27.103.4) を発行します。その後、クライアントが cookie を使用し続けると、非成功の結果コードを含む応答を受け取ります。

rootDSE エントリの supportedGroupingTypes 属性に transactionGroupingType が存在すれば、LDAP トランザクションがサポートされます。

OpenText eDirectory の LDAP トランザクションの実装は、LDAP トランザクション仕様の古いバージョンに基づいています。本書の執筆時点で、LDAP トランザクションの最新リビジョンのドラフトは Lightweight Directory Access Protocol (LDAP) Transactions (http://tools.ietf.org/html/rfc5805) で参照できます。

制限

LDAP トランザクションの機能には、次の制限があります。

- トランザクションとしてグループ化された操作の影響を受けるすべてのオブジェクトは、サーバにローカルにホストされている必要があります。これらの操作のいずれにおいても、LDAP サーバが別のサーバにチェーンする必要はありません。
- スキーマの変更およびDN変更の操作(サブツリーの移動?)は､LDAPトランザクション内にグループ化できません。
- ストリーム構文のパスワードおよび属性は、LDAP トランザクションの一部として追加できません。
- 1つのトランザクションを別のトランザクションにネストすることはサポートされていません。

14 LDAP Services for OpenText eDirectory の環境設定

OpenText インストールプログラムにより、LDAP Services for OpenText eDirectory が自動的にインストールされます。OpenText eDirectory のインストールの詳細については、『*OpenText™ eDirectory CE 24.4 (v9.3) インストールガイド*』を参照してください。

この章では、以下の説明を行います。

- 403 ページの 「LDAP Services for OpenText eDirectory のロードおよびアンロード」
- 404 ページの 「LDAP サーバがロードされているか確認する」
- 405 ページの 「LDAP サーバが実行されているか確認する」
- 407 ページの 「SSLv3 無効化による POODLE 攻撃の防止」
- 407 ページの 「LDAP オブジェクトを環境設定する」
- 422 ページの 「LDAP サーバをリフレッシュする」
- 423 ページの 「認証とセキュリティ」
- 433 ページの 「LDAP サーバを使ってディレクトリを検索する」
- 442 ページの 「上方参照を設定する」
- 447 ページの 「持続的検索 : OpenText eDirectory イベントの設定」
- 449 ページの 「LDAP サーバの情報を取得する」
- 451 ページの 「汎用タイムサポートの設定」
- 452 ページの 「許容変更の設定」
- 452 ページの 「プロキシ承認コントロール」
- 453 ページの 「LDAP ページ検索コントロール」
- 453 ページの 「LDAP 拡張 DN コントロール」
- 457 ページの 「LDAP イベントの監査」

LDAP ツールについては、「LDAP ツール NDK (http://developer.novell.com/documentation/cldap/ltoolenu/data/hevgtl7k.html)」を参照してください。

LDAP Services for OpenText eDirectory のロードおよびアンロード

LDAP Services for OpenText eDirectory をロードするには、次のコマンドを入力します。

サーバ	コマンド
Windows	［DHOST (NDSCONS)］画面で［**nldap.dlm**］を選択し、［**開始**］をクリックします。
Linux	Linux コマンドプロンプトで、次のように入力します。 /opt/novell/eDirectory/sbin/nldap -l

LDAP Services for OpenText eDirectory をアンロードするには、次のコマンドを入力します。

サーバ	コマンド
Windows	［DHOST (NDSCONS)］画面で［**nldap.dlm**］を選択し、［**停止**］をクリックします。
Linux	LDAP をアンロードするには、［DHost リモート管理 (DHost remote management)］ページで［*LDAP v3 for NetIQ eDirectory*］アクションアイコンをクリックして停止します。 または Linux コマンドプロンプトで、次のように入力します。 /opt/novell/eDirectory/sbin/nldap -u

LDAP サーバがロードされているか確認する

LDAP オブジェクトの設定を行う前に、LDAP サーバがロードされ、動作していることを確認します。この画面では、LDAP サーバがロードされているか確認する方法を説明します。サーバが動作しているか確認するには、「405 ページの 「LDAP サーバが実行されているか確認する」」を参照してください。

Windows の場合

1 Windows サーバで、ndscons.exe を開きます。

［**スタート**］>［**設定**］>［**コントロールパネル**］>［**OpenText eDirectory サービス**］の順にクリックします。

2 ［**サービス**］タブで、［**nldap.dlm**］までスクロールし、［**ステータス**］カラムを表示します。

カラムに［稼働中］と表示されます。

Linux の場合

LDAP サーバが実行されていることを確認するには、次のコマンドを実行します。

```
ndstrace -c modules | grep nldap
```

LDAP サーバがロードされていないか、実行されていない場合、nldap モジュールがロードされていないことを示すエラーが表示されます。

次のオプションを使用することもできます。

- LDAP サーバが実行されており SSL ポートをリスンしているかどうかを確認するには、nldap -s コマンドを実行します。
- LDAP サーバが実行されており TCL ポートをリスンしているかどうかを確認するには、nldap -c コマンドを実行します。

これらはエラーなしで実行されている、OpenText eDirectory のすべてのインスタンスを表示します。LDAP サーバがロードされておらず、いずれのポートもリスンしていない場合、上記のコマンドはエラー -255(LDAP サーバが実行されていることを確認する) を表示します。

LDAP サーバが実行されているか確認する

LDAP サーバをロードした後で、それが実行されているか確認します。その後、デバイスが監視しているか確認します。

- 405 ページの 「シナリオ」
- 406 ページの 「LDAP サーバが実行されているか確認する」
- 406 ページの 「デバイスが受信待機していることを確認する」

シナリオ

通常、LDAP サーバはロードされるとすぐに実行されます。ただし、次の 2 つのシナリオでは、サーバが正しく実行されないことがあります。

シナリオ : サーバがゾンビ状態にある。LDAP サーバは DHost ローダが外部依存関係を解決できる限り、ロードを続けます。ただし、LDAP サーバは有効な設定を 2 つの設定オブジェクト (LDAP サーバオブジェクトと LDAP グループオブジェクト) から取得するまで、正しく動作しません。

LDAP サーバが、「ロードはされても、実行されない (ゾンビ) 状態」の場合、LDAP サーバは定期的に設定オブジェクトを探そうとします。オブジェクトの設定が正しくなかったり、オブジェクトが破損したりしている場合、LDAP サーバ (nldap.nlm、nldap.dlm、libnldap.so、または libnldap.sl) は、アンロードされるかダウンするまで、ゾンビ状態になります。

ローダは LDAP サーバがロードされていると示しているのに、nldap.nlm(または nldap.dlm、libnldap.so、libnldap.sl) により LDAP ポート (389、636) が開かれていません。また、LDAP クライアントの要求はどれも実行されていません。

定期的な試行の記録と、サーバが稼動状態にならない原因を示す DS T r a c e メッセージが表示されます。

シナリオ : サービス拒否 。Digital Airlines 社のサーバは現在、長時間 (20 分以上) かかる検索を処理しています。この処理では、膨大なデータの中から検索が行われています。

この検索の実行中に、あるユーザが次のいずれかを実行します。

- 環境設定パラメータを変更し、設定オブジェクトを更新する。
- ［Refresh Server Now］をクリックする。
- LDAP サーバ (nldap.nlm、nldap.dlm、libnldap.so、libnldap.sl のいずれか) をアンロードする。
- サーバ全体を終了しようとする。

LDAP サーバは、現在の処理が完了してから更新を適用します。この更新が完了するまで、新しい操作は実行されません。この遅れにより、サーバの検索が完了し、更新が適用されるまでの間、サーバが新しい要求に応答しなくなったように見えることがあります。アンロードの実行中にサーバが停止したように見えることもあります。

検索要求が長く、多くの一致項目がある場合、LDAP サーバをアンロードしようとすると検索は中止され、次の一致項目がクライアントに返される時にアンロードが実行されます。ただし、検索要求の結果、20 分間に 1 件以下しか一致する項目が見つからなかった場合、LDAP サーバは実行中の NDS® または OpenText eDirectory 要求を中止できません。

リフレッシュまたは更新の場合、クライアントに返される多くの一致レコードがあっても検索は中止されません。

LDAP サーバが実行されているか確認する

LDAP サービスが実行されていることを確認するには、OpenText インポート / エクスポート変換ユーティリティ (ICE) を使用します。ワークステーションで、ice.exe を実行するか、OpenText Identity Console を使用します。

IP アドレスとポート番号を入力して接続が確立された場合は、サーバは機能しています。その他の場合は、エラーメッセージが表示されます。ログファイルまたはエクスポートファイルをダウンロード (表示) してください。

デバイスが受信待機していることを確認する

デバイスがポート 389 で受信待機していることを確認します。

1 コマンドラインで次を入力します。

```
netstat -a
```

2 ローカルアドレスが *servername*:389 で、状態が LISTENING の行を探します。

次のいずれかの状況が発生した場合は、OpenText iMonitor を実行します。

- ICE ユーティリティから情報を取得できない
- LDAP サーバが LDAP 要求をハンドルしているか確認できない

OpenText iMonitor の詳細については、249 ページの 「環境設定ファイル」および 258 ページの 「トレースを環境設定する」を参照してください。

LDAP 要求の詳細については、『*OpenText™ eDirectory CE 24.4 (v9.3) インストールガイド*』の「「LDAP による OpenText eDirectory との通信」」を参照してください。

SSLv3 無効化による POODLE 攻撃の防止

セキュアな通信のために OpenText eDirectory が LDAPS プロトコルを SSLv3 で使用する場合、CVE-2014--3566 のとおり、SSLv3 が POODLE 攻撃に対して脆弱であることに注意してください。

デフォルトでは、OpenText eDirectory は FIPS モードで実行され、SSLv3 経由の通信を許可しません。詳細については、「FIPS モードで eDirectory を設定する」を参照してください。OpenText eDirectory サーバで TLS の FIPS モードを無効にする場合は、次の手順を使用して、LDAP の SSLv3 を無効にすることをお勧めします。

回避策 :

1. ソフトウェアライセンスおよびダウンロードポータルから、OpenText eDirectory 用の最新の OpenText Identity Console をダウンロードしてインストールします。
2. OpenText Identity Console を起動します。
3. OpenText Identity Console のホームページで、［**LDAP 環境設定**］タイルをクリックします。
4. ［**LDAP サーバ**］を選択します。
5. ［**LDAP サーバの変更**］ページで、［**環境設定**］ドロップダウンを選択し、［**SSLv3**］チェックボックスをオンにします。
6. ［**保存**］をクリックします。

   注 : 英語以外の環境では［**SSLv3**］オプションを使用できません。このオプションを使用するには、優先する表示言語を英語に変更します。

7. LDAP Services for OpenText eDirectory をアンロードして、ロードします。

   詳細については、403 ページの 「LDAP Services for OpenText eDirectory のロードおよびアンロード」を参照してください。

LDAP オブジェクトを環境設定する

OpenText eDirectory のインストール時に、LDAP サーバオブジェクトと LDAP グループオブジェクトが作成されます。LDAP サービスのデフォルト設定は、これらの 2 つのオブジェクト上のディレクトリにあります。OpenText Identity Console で LDAP 管理タスクを使用することで、デフォルトの設定を変更できます。

LDAP サーバオブジェクトとは、サーバ固有の設定データのことです。

LDAP グループオブジェクトには、複数の LDAP サーバ間で共有できる便利な設定情報が含まれています。このオブジェクトは、共通の設定データと LDAP サーバグループを提供します。サーバは共通データを持っています。

複数の LDAP サーバオブジェクトを、1 つの LDAP グループオブジェクトと関連させることができます。関連するすべての LDAP サーバは、サーバ固有の設定をそれぞれの LDAP サーバオブジェクトから取得しますが、共通する情報や共有情報は LDAP グループオブジェクトから取得します。

デフォルトで、OpenText eDirectory インストールプログラムは、各 nldap.nlm または nldap.dlm に LDAP グループオブジェクトと LDAP サーバオブジェクトを 1 つずつインストールします。その後、複数の LDAP サーバオブジェクトを、1 つの LDAP グループオブジェクトに関連付けることができます。

重要 : 新しいバージョンの LDAP サーバオブジェクトを古いバージョンの LDAP グループオブジェクトに関連付けることもできますが、異なるバージョン間での関連付けはお勧めしていません。たとえば、eDirectory 8.7.3 SP9 の LDAP グループオブジェクトと OpenText eDirectory·9.0 以降の LDAP サーバオブジェクトとの関連付けは避けるようにしてください。

LDAP グループオブジェクトが保持する、共通情報の量は制限されています。属性に含まれるデータはほとんど共通しているため、LDAP は多くの属性を読み込む必要がありません。多くの LDAP サーバは同じデータを使用する必要があります。共通の、または共有グループオブジェクトがない場合は、各 LDAP サーバにそのデータを複製する必要があります。

LDAP サーバオブジェクトでは、LDAP グループオブジェクトよりも多くのサーバ固有の設定オプションとデータが許可されています。

どちらのオブジェクトも、相互にポイントされた DN 構文属性を持っています。

LDAP サーバがその設定データを探せるようにするには、関連付けを追加する必要があります。関連付けは、通常の OpenText eDirectory 設定データを保持する NCP™ サーバを通じて行われます。この関連付けは、OpenText eDirectory インストールプログラムにより自動で行われます。

各 OpenText eDirectory サーバは、NCP サーバオブジェクトを持っています。次の図は、OpenText Identity Console に表示されるオブジェクトを示しています。



このオブジェクトは、特定のホスト OpenText eDirectory サーバの LDAP サーバオブジェクトを指す LDAP サーバ属性を持っています。次の図は、この属性を示しています。



通常、LDAP サーバオブジェクト、LDAP グループオブジェクト、NCP サーバオブジェクトは同じコンテナ内にあります。OpenText eDirectory のインストールで、サーバおよび管理者コンテキストを指定するときに、このコンテナを指定します。

LDAP サーバオブジェクトを移動するときは、それを書き込み可能なレプリカに置く必要があります。

Linux 上で LDAP サーバオブジェクトおよび LDAP グループオブジェクトを環境設定する

LDAP 環境設定ユーティリティは ldapconfig です。Linux システムで LDAP サーバオブジェクトおよび LDAP グループオブジェクトの属性を変更、表示、リフレッシュするには、ldapconfig を使用します。

Linux システム上で LDAP 属性値を表示するときには次の構文を使用します。

```
ldapconfig get [...] | set attribute-value-list  [-t treename | -p
hostname[:port]] [-w password] [-a user FDN] [-f]
```

```
ldapconfig [-t tree_name | -p host_name[:port]] [-w password] [-a user FDN]
[-V] [-R] [-H] [-f] -v attribute,attribute2...
```

Linux システム上で LDAP 属性値を変更するときには次の構文を使用します。

```
ldapconfig [-t tree_name | -p host_name[:port]] [-w password] [-a
admin_FDN] -s attribute=value,...
```

パラメータ	説明
-t *treename*	コンポーネントのインストール先となる OpenText eDirectory ツリーの名前。
-p *hostname*	ホストの名前です。DNS 名または IP アドレスを指定することもできます。
-w	管理権を持つユーザのパスワード。

パラメータ	説明
-a	管理権を持つユーザの完全識別名。次に例を示します。 cn=user.o=org1
get | -V	すべての LDAP サーバと LDAP グループの属性を表示します。
get | -v *attribute list*	属性リストにある属性の現在の値を表示します。
set | -s *attribute-value pairs*	属性を指定した値で設定します。
-v	LDAP 属性値を表示します。
-s	インストールされたコンポーネントの属性値を設定します。
-R	LDAP サーバをリフレッシュします。
-V	現在の LDAP 環境設定を表示します。
-H	使用方法とヘルプを表示します。
-f	フィルタ済みレプリカ上での操作を許可します。
attribute	設定可能な LDAP サーバ属性名またはグループ属性名。詳細については、410 ページの 「LDAP サーバオブジェクトの属性」および 419 ページの 「LDAP グループオブジェクトの属性」を参照してください。

例

属性リストの属性の値を表示するには、次のコマンドを入力します。

```
ldapconfig [-t tree_name | -p host_name[:port]]
[-w password] [-a user_FDN] -v "Require TLS for simple binds with
password","searchTimeLimit"
```

LDAP TCP ポート番号と検索サイズの制限を 1000 に設定するには、次のコマンドを入力します。

```
ldapconfig [-t tree_name | -p host_name[:port]]
[-w password] [-a admin_FDN] -s "LDAP TCP Port=389","searchSizeLimit=1000"
```

LDAP サーバオブジェクトの属性

OpenText LDAP サーバプロパティを設定および管理するために、LDAP サーバオブジェクトを使用します。

次の表に、LDAP サーバ属性の説明を示します。

属性	説明
LDAP サーバ	OpenText eDirectory の LDAP サーバオブジェクトの完全識別名。

属性	説明
LDAP Host Server	LDAP サーバの実行場所となるホスト OpenText eDirectory サーバの完全識別名。
LDAP グループ	OpenText eDirectory の中で、この LDAP サーバがメンバーとして属する LDAP グループオブジェクト。
LDAP Server Bind Limit	LDAP サーバに同時にバインドできるクライアントの数。0 を指定すると、無制限になります。
LDAP Server Idle Timeout	あるクライアントと LDAP サーバの間で、ここで指定した期間無活動状態が継続すると、このクライアントと LDAP サーバの接続が切断されます。0 を指定すると、無制限になります。
LDAP Enable TCP	このオプションは廃止される予定です。ldapInterfaces から使用可能です。 詳細については、415 ページの 「ldapInterfaces」を参照してください。
LDAP Enable TLS	このオプションは廃止されました。ただし、ldapInterfaces から使用可能です。 詳細については、415 ページの 「ldapInterfaces」を参照してください。
LDAP TCP Port	このオプションは廃止されました。ただし、ldapInterfaces から使用可能です。 詳細については、415 ページの 「ldapInterfaces」を参照してください。
LDAP TLS ポート	このオプションは廃止されました。ただし、ldapInterfaces から使用可能です。 詳細については、415 ページの 「ldapInterfaces」を参照してください。
keyMaterialName	この LDAP サーバに関連付けられた、SSL LDAP 接続に使用する OpenText eDirectory の証明書オブジェクトの名前。
searchSizeLimit	LDAP サーバが検索要求への応答として LDAP クライアントに返すエントリの最大数。0 を指定すると、無制限になります。 ユーザが LDAP サーバオブジェクト上で管理者権限を持っている場合、searchSizeLimit 値は考慮されません。管理権はキャッシュに保存されるので、ユーザの管理権に対する変更は直ちに有効にはなりません。管理権に対する変更は、次回の LDAP サーバ更新で有効になります。デフォルトでは、LDAP サーバは 30 分ごとに更新されます。
searchTimeLimit	LDAP サーバによる LDAP 検索がタイムアウトになるまでの最大秒数。0 を指定すると、無制限になります。 ユーザが LDAP サーバオブジェクト上で管理者権限を持っている場合、searchTimeLimit 値は考慮されません。管理権はキャッシュに保存されるので、ユーザの管理権に対する変更は直ちに有効にはなりません。管理権に対する変更は、次回の LDAP サーバ更新で有効になります。デフォルトでは、LDAP サーバは 30 分ごとに更新されます。

属性	説明
filteredReplicaUsage	LDAP サーバが LDAP 検索のために、フィルタ処理されたレプリカを使用するかどうかを指定します。 値は 1(フィルタ済みレプリカを使用) か 0(フィルタ済みレプリカ不使用) です。
sslEnableMutualAuthentication	LDAP サーバにおいて、SSL ベースの相互認証 (証明書に基づくクライアント認証) を有効にするかどうかを指定します。
ldapTLSVerifyClientCertificate	LDAP による TLS 操作のクライアント認証の確認を有効または無効にします。
ldapNonStdAllUserAttrsMode	非標準のすべてのユーザとオペレーショナル属性を有効または無効にします。

属性	説明
ldapBindRestrictions	LDAP クライアント接続で LDAP バインド制限とサイファレベルを有効にします。この属性はクライアント接続の制御に使用できます。OpenText Identity Console を使用して、次の 7 つの LDAP バインド制限のいずれかを設定できます。 • **なし :** デフォルトでは、このオプションは有効になっています。このオプションは匿名単純認証と非匿名単純認証の両方を有効にします。このオプションの値は、0 です。 • **匿名単純バインドを不許可にする :** 匿名単純認証を無効にするには、値を 1 に設定します。非匿名単純認証が有効になります。 • **非匿名単純認証を不許可にする :** 非匿名単純認証を無効にするには、値を 2 に設定します。 • **匿名単純認証と非匿名単純認証を許可しない :** 匿名単純認証と非匿名単純認証の両方を無効にするには、値を 3 に設定します。 **注 :** 非匿名単純認証を無効にすると、適切な猶予ログインの制限が強制的に適用されます。 • **非認証バインドを許可しない :** パスワードを使用しない単純認証を無効にするには、値を 4 に設定します。 • **匿名の非認証バインドを許可しない :** 匿名単純認証と非認証バインドを無効にするには、値を 5 に設定します。 • **非匿名単純認証と非認証バインドを許可しない :** 非匿名単純認証と非認証バインドを無効にするには、値を 6 に設定します。このシナリオでは、匿名単純認証が有効になります。 • **匿名単純認証、非匿名単純認証、および非認証バインドを許可しない :** 匿名単純認証、非匿名単純認証、および非認証バインドを無効にするには、値を 7 に設定します。 **注 :** 4 から 7 までの値は、ldapconfig ユーティリティで設定できます。OpenText Identity Console では、この値を設定できません。詳細については、表 14-1 を参照してください。 RSA と楕円曲線デジタル署名 (ECDSA) アルゴリズムでは、OpenText eDirectory は次の値を使用してサイファの使用を制限できます。 • **RSA:** 次の値を使用します。 ◦ **高度暗号化 (128 ビットを超える):** 128 ビット暗号化よりも上位の暗号レベルと 128 ビットキーの一部の暗号スイートの使用を指定する場合には、値を 48 に設定します。 ◦ **中度暗号化 :** 128 ビット暗号化の暗号レベルの使用を指定するには、値を 32 に設定します。 ◦ **低度暗号化 :** エクスポート暗号スイートを除く、64 または 56 ビット暗号化の使用を指定する場合、値を 16 に設定します。 ◦ **エクスポート :** 40 ビットおよび 56 ビット暗号化を含むサイファレベルの使用を指定します。値 0。 デフォルトは［高］で、128 ビット暗号化よりも大きいサイファレベルとなります。この値が［0］に設定されている場合、OpenText eDirectory 9.1 SP4 にアップグレードされた後、値が自動的に［高］に変更されます。 **注 :** TLS の FIPS モードを有効にすると、OpenText eDirectory は暗号

属性	説明
	Suite B モード : 次の値を使用します。 ◆ **Suite B Cipher(128 ビット):** 128 ビットレベルのセキュリティを使用して、Suite B モードの操作を有効にするには、値を 64 に設定します。このオプションを選択すると、OpenText eDirectory は、ピア (任意の LDAP クライアント) による 128 ビットレベルと 192 ビットレベルの両方のセキュリティの使用を許可します。このオプションでは ECDSA 256 または ECDSA 384 のいずれかの証明書を使用できます。 ◆ **Suite B Cipher(128 ビットのみ) を使用する :** 128 ビットレベルのセキュリティを使用して、Suite B モードの操作を有効にするには、値を 80 に設定します。このオプションを選択すると、OpenText eDirectory は、ピア (任意の LDAP クライアント) による 192 ビットレベルのセキュリティの使用を許可しなくなります。このオプションでは ECDSA 256 証明書のみを使用できます。 ◆ **Suite B Cipher(192 ビット) を使用する :** 192 ビットレベルのセキュリティを使用して、Suite B モードの操作を有効にするには、値を 96 に設定します。このオプションを選択すると、OpenText eDirectory は、ピア (任意の LDAP クライアント) による 192 ビットレベルのセキュリティの使用のみを許可します。このオプションでは ECDSA 384 証明書のみを使用できます。 OpenText eDirectory では、ldapbindrestrictions とサイファレベルの値を組み合わせて使用できます。詳細については、表 14-1 を参照してください。
ldapChainSecureRequired	これはブール属性です。有効な場合、他の OpenText eDirectory へのチェーンはセキュアな NCP で行われます。デフォルトでは、ldapChainSecureRequired は無効です。

属性	説明

ldapInterfaces	LDAP サーバが (クリアテキストおよびセキュアなポートの両方を) リスンする LDAP URL を保存するために使用される複数値の SYN_CI_STRING 属性。この属性は、OpenText eDirectory サーバでそれぞれのインスタンスが特定のインタフェースについてリスンすることが必要な、複数のインスタンスを構成する際に役立ちます。これは、LDAP URL 形式の IP アドレスとポート番号を使用して設定できます。LDAP サーバはこれらの IP アドレスとポートをリスンします。
	
	IPv4 と IPv6 リスナの例を次に示します。
	
	`ldap://192.168.1.1:389 - To specify for IPv4 specific address on clear text port`
	
	`ldaps://192.168.2.1:636 - To specify for IPv4 specific address on secure port`
	
	`ldap://[2015::3]:389 - To specify for IPv6 specific address on clear text port`
	
	`ldaps://[2015::3]:636 - To specify for IPv6 specific address on secure port`
	
	`ldap://[::]:389 - To specify for IPv6 unspecified address on clear text port`
	
	`ldaps://[::]:636 - To specify for IPv6 unspecified address on secure port`
	
	新しいサーバが OpenText eDirectory 9.1 以降で設定されている場合は、LDAP Enable TCP、LDAP Enable TLS、LDAP TCP Port、および LDAP TLS Port の属性は自動入力されません。設定中に ldap および ldaps に対して選択されたポートに対応する ldapInterface 属性値は自動入力されます。例 : ldap://:389、ldaps://:636 デフォルトでは、IPv4 インタフェース値のみが ldapInterfaces 属性に追加されます。
	
	アップグレード中に、OpenText eDirectory は、LDAP Enable TCP、LDAP Enable TLS、LDAP TCP Port、LDAP TLS Port 属性を削除するためにトリガされます。これらの属性に対応する値が ldapInterface に入力されます。ldapconfig set コマンドは、コンマ区切りの値を取得して、既存のすべての値を新しい値に置き換えます。

ldapStdCompliance	OpenText eDirectory LDAP サーバは、デフォルトでは、1 つのレベルの検索でサブオーディネート参照を返しません。これを有効にするには、ldapStdCompliance を値 1 にしてオンにする必要があります。この値を設定すると、LDAP サーバは 1 つのレベルの検索でサブオーディネート参照を返すようになります。
ldapChainSecureRequired	これはブール属性です。有効な場合、他の OpenText eDirectory へのチェーンはセキュアな NCP で行われます。デフォルトでは、この属性は無効です。
ldapEnablePSearch	LDAP サーバで持続的検索機能を有効にするかどうかを指定します。 Values= yes, no

属性	説明
ldapMaximumPSearchOperations	同時に実行できる持続的検索操作の数を制限するための整数値です。0 を指定すると、検索操作は無制限になります。
ldapIgnorePSearchLimitsForEvents	持続的検索要求によって最初の結果が返された後で、サイズと時間の制限を無視するかどうかを指定します。 Values= yes, no この属性が FALSE に設定されている場合、すべての持続的検索操作は検索制限の制約を受けます。サイズと時間のいずれかの制限に達した場合、検索操作は失敗し、該当するエラーメッセージが返されます。
ldapGeneralizedTime	時刻を *YYYYMMDDHHmmSS.0Z* 形式で表示するために汎用タイムを有効にします。 Values= yes, no
ldapPermissiveModify	LDAP 変更操作を拡張するために Permissive Modify Control を有効にします。存在しない属性を削除したり、属性にすでにある属性値を追加しようとすると、エラーメッセージが表示されることなく操作が完了します。 Values= yes, no
ldapSSLConfig	この属性では、LDAP サーバオブジェクトで TLS プロトコルおよび暗号を定義することができます。デフォルトの設定では、この属性は無効です。この環境設定属性は、次の優先順位に従います。 ◆ LDAP サーバオブジェクトの ldapSSLConfig 属性値 ◆ LDAP グループオブジェクトの ldapSSLConfig 属性値 この属性でプロトコルと暗号が定義されていない場合、ldapBindRestrictions で指定されているデフォルトの構成に従います。詳細については、419 ページの 「ldapSSLConfig 属性を使用してプロトコルと暗号を構成する」を参照してください。 **注 :** ldapSSLConfig 属性は、eDirectory 9.0 SP2 以降で使用できます。
ldapGroupSSLConfig	この属性では、LDAP グループオブジェクトで TLS プロトコルおよび暗号を定義することができます。デフォルトの設定では、この属性は無効です。この環境設定属性は、次の優先順位に従います。 ◆ LDAP サーバオブジェクトの ldapSSLConfig 属性値 ◆ LDAP グループオブジェクトの ldapSSLConfig 属性値 この属性でプロトコルと暗号が定義されていない場合、ldapBindRestrictions で指定されているデフォルトの構成に従います。詳細については、419 ページの 「ldapSSLConfig 属性を使用してプロトコルと暗号を構成する」を参照してください。 **注 :** この属性を ldapconfig get/set コマンドで設定する場合、ldapGroupSSLConfig を使用します。ldif ファイルで設定する場合には、ldapSSLConfig に LDAP グループオブジェクト DN を指定して使用します。

表 14-1 ldapbindrestrictions とサイファレベルの組み合わせ値

ldapbindrestriction	証明書	サイファレベル	組み合わせ値
なし	RSA	エクスポート	0
	RSA	高	48
	RSA	中	32
	RSA	低い	16
	ECDSA 256/384	SUITEB128	64
	ECDSA 256	SUITEB128ONLY	80
	ECDSA 384	SUITEB192	96
匿名単純認証を許可しない	RSA	エクスポート	1
	RSA	高	49
	RSA	中	33
	RSA	低い	17
	ECDSA 256/384	SUITEB128	65
	ECDSA 256	SUITEB128ONLY	81
	ECDSA 384	SUITEB192	97
ローカルバインドを許可しない	RSA	エクスポート	2
	RSA	高	50
	RSA	中	34
	RSA	低い	18
	ECDSA 256/384	SUITEB128	66
	ECDSA 256	SUITEB128ONLY	82
	ECDSA 384	SUITEB192	98
匿名単純認証とバインド解除を許可しない	RSA	エクスポート	3
	RSA	高	51
	RSA	中	35
	RSA	低い	19
	ECDSA 256/384	SUITEB128	67
	ECDSA 256	SUITEB128ONLY	83
	ECDSA 384	SUITEB192	99

ldapbindrestriction	証明書	サイファレベル	組み合わせ値
非認証バインドを許可しない	RSA	エクスポート	4
	RSA	高	52
	RSA	中	36
	RSA	低い	20
	ECDSA 256/384	SUITEB128	68
	ECDSA 256	SUITEB128ONLY	84
	ECDSA 384	SUITEB192	100
匿名の非認証バインドを許可しない	RSA	エクスポート	5
	RSA	高	53
	RSA	中	37
	RSA	低い	21
	ECDSA 256/384	SUITEB128	69
	ECDSA 256	SUITEB128ONLY	85
	ECDSA 384	SUITEB192	101
非匿名単純認証と非認証バインドを許可しない	RSA	エクスポート	6
	RSA	高	54
	RSA	中	38
	RSA	低い	22
	ECDSA 256/384	SUITEB128	70
	ECDSA 256	SUITEB128ONLY	86
	ECDSA 384	SUITEB192	102
匿名単純認証、非匿名単純認証、および非認証バインドを許可しない	RSA	エクスポート	7
	RSA	高	55
	RSA	中	39
	RSA	低い	23
	ECDSA 256/384	SUITEB128	71
	ECDSA 256	SUITEB128ONLY	87
	ECDSA 384	SUITEB192	103

LDAP グループオブジェクトの属性

LDAP クライアントの OpenText LDAP サーバに対するアクセス方法と、サーバ上の情報の使用方法を設定および管理するには、LDAP グループオブジェクトを使用します。

単純認証に TLS が必要な場合は、「424 ページの 「パスワードとの単純バインドに TLS を要求する」」を参照してください。この属性は、LDAP サーバが LDAP クライアントからパスワードをクリアテキストで送信することを許可するかどうかを指定します。値は 1(はい) または 1(いいえ) です。

デフォルトの参照、referralIncludeFilter、referralExludeFilter、および LDAP サーバによる LDAP 参照の処理方法を指定するには、「434 ページの 「参照を使用する」」を参照してください。

TLS プロトコルと暗号を指定するには、ldapSSLConfig 属性を使用できます。詳細については、419 ページの 「ldapSSLConfig 属性を使用してプロトコルと暗号を構成する」を参照してください。

ldapSSLConfig 属性を使用してプロトコルと暗号を構成する

OpenText eDirectory を使用すると、LDAP サーバの TLS 通信に必要な各種 TLS パラメータと暗号を定義できます。9.30 以降では、TLSv1.2 および TLSv1.3 プロトコルのみがサポートされています。TLSv1.1 プロトコルより下のバージョンはサポートされていません。

LDAP サーバオブジェクトとグループオブジェクトの両方に関して、ldapSSLConfig 属性でプロトコルと暗号を JSON 形式で指定できます。たとえば、以下に示されているように JSON 形式でプロトコルと暗号を定義できます。

```
{
    "Version": 1,
    "Info": {
        "Protocol": "+ALL",
        "Ciphers": "ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-
SHA384", "Ciphersuites" : "TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384"
    }
}
```

注 : ldapSSLConfig 属性で間違った情報を指定すると、ldapBindRestrictions で指定されているデフォルト構成が使用されます。

サポートされている暗号

OpenText eDirectory CE 24.4 (v9.3) では、次の暗号がサポートされています。

TLSv1.3 でサポートされている暗号 -

- TLS_AES_256_GCM_SHA384
- TLS_CHACHA20_POLY1305_SHA256 [非 FIPS のみ]
- TLS_AES_128_GCM_SHA256

TLSv1.2 でサポートされている暗号

- ECDHE-ECDSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-ECDSA-CHACHA20-POLY1305
- ECDHE-RSA-CHACHA20-POLY1305 [非 FIPS のみ]
- ECDHE-ECDSA-AES128-GCM-SHA256
- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-ECDSA-AES256-SHA384
- ECDHE-RSA-AES256-SHA384
- ECDHE-ECDSA-AES128-SHA256
- ECDHE-RSA-AES128-SHA256
- AES256-GCM-SHA384
- AES128-GCM-SHA256
- AES256-SHA256
- AES128-SHA256

暗号を設定する

OpenSSL 暗号リスト形式を使用して、独自の暗号リストを構成できます。次の例では、LDAP サーバの TLS 通信時に使用される暗号リスト形式を示します。

- RSA 証明書 : !CAMELLIA:!DH:!SRP:!MD5:HIGH + aRSA
- ECDSA 証明書 : HIGH+aECDSA
- Suite B 128 ビット準拠の暗号スイート (ECDSA 証明書使用): ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256
- Suite B 192 ビット準拠の暗号スイート (ECDSA 証明書使用): ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES-GCM-SHA

注 : 次のコマンドを実行して、LDAP サーバでサポートされている TLS 暗号スイートのリストを検索します。

```
bash$ nmap --script /usr/share/nmap/scripts/ssl-enum-ciphers.nse -p 636
<eDirectory-server-IP>
```

例 :

```
bash$ nmap --script /usr/share/nmap/scripts/ssl-enum-ciphers.nse -p 636
192.168.1.1
```

暗号リスト形式について詳しくは、OpenSSL ciphers (http://www.openssl.org/docs/man1.0.2/apps/ciphers.html) マニュアルを参照してください。

プロトコルを設定する

OpenText eDirectory では、TLS 通信中に必要なプロトコルのリストを柔軟に設定できます。プロトコルのリストを制御するには、ldapSSLConfig 属性で必要なプロトコルを JSON 形式で定義します。次のプロトコル文字列を設定できます。

- SSLv3、TLSv1.0 および TLSv1.1 - OpenText eDirectory 9.30 以降では、これらのバージョンのプロトコルはサポートされていません。
- TLSv1.2
- TLSv1.3
- ALL

各プロトコル文字列の前に「+」または「-」記号を入力する必要があります。「+」記号はそのプロトコル文字列が OpenText eDirectory で許可されていることを、「-」記号はそのプロトコル文字列が許可されていないことを示します。次の表に、TLS プロトコル構成のいくつかを示します。

プロトコル構成	説明
+TLSv1.2	TLSv1.2 のみを許可します
+TLSv1.3	TLSv1.3 のみを許可します
+ALL-TLSv1.2	TLSv1.2 以外のすべてを許可します
+ALL-TLSv1.3	TLSv1.3 以外のすべてを許可します
+ALL	TLSv1.2、TLSv1.3 を許可します

注 : +ALL を指定する場合、プロトコルの前に付けることができるのは「-」記号だけです。

例 :

Suite B 準拠モードでプロトコルと暗号を構成する (TLSv1.2 の場合)

```
{
    "Version": 1,
    "Info": {
        "Protocol": "+TLSv1.2",
        "Ciphers": "ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-
SHA384"
    }
}
```

上の例では、プロトコルが JSON 形式で + TLSv1.2 として定義されています。

Suite B 準拠モードでプロトコルと暗号を構成する (TLSv1.3 の場合)

```
{
    "Version": 1,
    "Info": {
        "Protocol": "+TLSv1.3",
        "Ciphersuites": "TLS_AES_256_GCM_SHA384:TLS_AES_128_GCM_SHA256"
    }
}
```

上の例では、プロトコルが JSON 形式で +TLSv1.3 として定義されています。

非 Suite B 準拠モードでプロトコルおよび暗号を設定する

例 1:

```
{
    "Version": 1,
    "Info": {
        "Protocol": "+ALL",
        "Ciphers": "HIGH+aECDSA"
    }
}
```

例 2:

```
{
    "Version": 1,
    "Info": {
        "Protocol": "+ALL",
        "Ciphers": "ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-
SHA384","Ciphersuites" : "TLS_AES_256_GCM_SHA384"
    }
}
```

上の例では、プロトコルが JSON 形式で +ALL と定義されています。つまり、TLS 通信時にはすべてのサポート対象プロトコルが許可されます。

LDAP サーバをリフレッシュする

LDAP サーバの環境設定オプションや LDAP サーバの設定を変更した場合、変更を有効にするにはサーバをリフレッシュする必要があります。

ただし、LDAP 要求のサービスの実行中はサーバをリフレッシュできません。たとえば、OpenText eDirectory ツリーの処理に 15 分かかる場合には、この処理が完了するまでリフレッシュは実行されません。

同様に、LDAP サーバスレッドの実行中は、LDAP サーバを終了することはできません。

リフレッシュの実行が予定されている場合は、LDAP サーバはリフレッシュが実行されるまで新しい LDAP 要求の開始を遅らせます。

デフォルトでは、LDAP サーバは 30 分間隔で LDAP サーバオブジェクトと LDAP グループオブジェクトのタイムスタンプをチェックし、設定に変更がなかったか確認します。設定が変更されている場合、サーバはその変更を適用します。

設定のタイムスタンプが前回と変わらない場合には、リフレッシュは実行されません。強制的にリフレッシュを実行すると、サーバはタイムスタンプを無視して変更を適用します。

LDAP サーバをリフレッシュするには、次のいずれかを実行します。

- サーバが次のリフレッシュ間隔で再設定されるまで待つ。
- nldap.nlm をアンロードしてから再ロードする。

  nldap.nlm をアンロードする前に前提条件の NLM プログラムをアンロードする必要はありません。

  nldap.nlm がアンロードされると、従属する NLM プログラムが再ロードされます。
- コマンドラインで、リフレッシュ間隔を変更する。

  このオプションは、WAN リンクが継続して実行されていない場合に便利です。必要に応じ、一時的にサーバのハートビート処理の長さを変更できます。

  この変更は持続しません。nldap.nlm をロードするたびに、コマンドを再入力する必要があります。

  サーバコンソールで次を入力します。

  ldap refresh [=] [*date*][*time*][*interval*]

  - 日付変数の形式は、mm:dd:yyyy です。すべての日付フィールドに 0 と入力すると、現在の日付が使用されます。
  - 時間変数の形式は、hh:mm:ss です。すべての時刻フィールドに 0 と入力すると、現在の時刻が使用されます。
  - 間隔変数の形式は 0 または 1 ～ 2147483647 分の間です。0 と入力すると、デフォルトの 30 分が使用されます。

  このコマンドは、sys:\system ディレクトリの autoexec.ncf ファイルに追加できます。nldap.nlm をロードした行の後に、このコマンドを配置します。

認証とセキュリティ

このセクションでは、次の情報について説明します。

- 424 ページの 「パスワードとの単純バインドに TLS を要求する」
- 424 ページの 「TLS を開始 / 停止する」
- 425 ページの 「TLS のサーバを環境設定する」
- 427 ページの 「TLS のクライアントを環境設定する」
- 427 ページの 「ルート認証局をエクスポートする」
- 428 ページの 「クライアント証明書で認証を受ける」
- 428 ページの 「サードパーティプロバイダの証明書を使用する」

- 429 ページの 「LDAP プロキシユーザを作成および使用する」
- 430 ページの 「SASL を使用する」
- 432 ページの 「NMAS ベースのログインを使用した LDAP 認証」

パスワードとの単純バインドに TLS を要求する

SSL(Secure Socket Layer) 3.1 は Netscape でリリースされました。IETF は、TLS (Transport Layer Security) 1.0 を実装することで、その標準の所有権を持ちます。TLS 1.0 は SSLv2 および v3 と後方互換性があります。

TLS を使用すると、接続をセッション層で暗号化することができます。TLS 接続のために、暗号化されたポートを使用する必要はありません。方法はもう 1 つあります。ポート 636 は暗黙的な TLS ポートであり、クライアントがセキュアポートに接続すると、LDAP サーバは自動的に TLS セッションを開始します。

クライアントは、まずクリアテキストポートに接続し、後で TLS を使用して暗号化された接続にアップグレードすることもできます。

パスワードとの単純バインドに TLS を要求するには、次を実行します。

1 OpenText Identity Console のホームページで、[**LDAP 環境設定**] タイルをクリックします。
2 [LDAP グループ] オブジェクトをクリックし、[**情報**] ドロップダウンメニューをクリックします。
3 [**パスワードとの単純バインドに TLS を必要とする**] チェックボックスをオンにします。



4 [**保存**] > [**OK**] の順にクリックします。

TLS を開始 / 停止する

LDAP 拡張オペレーション STARTTLS により、クリア接続から暗号化された接続にアップグレードすることができます。このアップグレードは、OpenText eDirectory8.7 の新機能です。

暗号化された接続を使用すると、パケット全体が暗号化されます。このため、ネットワーク経由で送信されたデータが第三者によって診断されることはありません。

シナリオ :STARTTLS を使用する ― ポート 389 にクリア接続し、匿名検索を行います。ただし、セキュリティ保護されたデータを扱う場合には TLS セッションに切り換えます。拡張オペレーション STARTTLS を実行し、クリア接続から暗号化された接続にアップグレードします。これでデータの安全が確保されます。

暗号化されたセッションをクリア接続に切り替えるには、TLS を停止します。クリア接続では、クライアントが送受信するデータは暗号化および解読されないので、負荷は少なくなります。そのため、クリア接続の使用時の方が、データの通信速度が速くなります。この時点で、接続は匿名にダウングレードされています。

認証を受けるには LDAP バインド操作を使用します。バインドは、ユーザの資格情報に基づいて ID を確立します。TLS を停止するときに、LDAP サービスは以前に確立された認証をすべて削除します。認証ステータスが匿名に変わります。匿名以外の状態に切り替える場合は、再認証を受ける必要があります。

シナリオ : 再認証 ― あるユーザが STOPTLS を実行します。すると、そのユーザの状態が匿名に変わります。このユーザのファイルにネット上でアクセスするには、Bind コマンドを実行し、ログイン認証情報を入力します。ユーザが認証され、インターネット上でクリアテキストで作業を続行できます。

TLS のサーバを環境設定する

TLS セッションがインスタンス化されると、ハンドシェークが行われます。サーバとクライアントがデータを交換します。ハンドシェークの方法はサーバが決定します。サーバの正当性を証明するため、サーバは常にサーバの証明書をクライアントに送信します。このハンドシェークにより、そのサーバがクライアントに指定されたサーバであることが証明されます。

クライアントにも正当性の証明を要求するには、サーバに値を設定します。これは ldapTLSVerifyClientCertificate という属性です。

値	説明
0	オフ . ハンドシェーク時に、サーバがクライアントに証明書を提供します。サーバがクライアントに証明書の送信を要求することはありません。クライアントは証明書を使用することも、無視することもできます。セキュリティ保護されたセッションが確立されます。
1	ハンドシェークの間、サーバは証明書をクライアントに送信し、クライアントからも証明書を要求します。クライアントはサーバに証明書を送信できます。クライアントの証明書が検証されます。サーバがクライアントの証明書を確認できない場合、接続は終了します。 クライアントが証明書を送信しない場合、サーバは接続を維持します。
2	ハンドシェークの間、サーバはクライアントから証明書を要求します。クライアントが証明書を提供しない、または証明書が確認できない場合には、接続は終了します。

サーバが TLS をサポートするよりも前に、サーバにその正当性の証明に使用する X.509 証明書を提供する必要があります。

この証明書は、OpenText eDirectory のインストール時に自動的に提供されます。インストール時に、PKI (公開鍵インフラストラクチャ) と NMAS (NetIQ モジュラー認証サービス) の一部として、キーマテリアルオブジェクトが作成されます。次の図は、OpenText Identity Console のこれらのオブジェクトを示しています。

図 14-1 *OpenText Identity Console のオブジェクト*



インストール中、これらの証明書の 1 つが LDAP サーバと自動的に関連付けられます。OpenText Identity Console の LDAP サーバオブジェクトの［接続］タブに DN が表示されます。この DN は、X.509 認定を表しています。次の図のサーバ証明書フィールドは、この DN を示しています。

図 14-2 *サーバ証明書フィールド*



OpenText Identity Console で、暗号化キーオブジェクト (KMO) 証明書をブラウズできます。また、ドロップダウンリストから、別の証明書に変更することもできます。DNS または IP 証明書のいずれかを使用します。

検証の際には、サーバは証明書にある名前 (ハード IP アドレスまたは DN) を確認します。

TLS 接続を確立するには、次の条件を確認します。

- LDAP サーバは、サーバの KMO を知る必要があります。
- クリアポートに接続してから、セキュアポートに接続するか TLS を開始します。

LDAP サーバを再設定し、サーバをリフレッシュします。「422 ページの 「LDAP サーバをリフレッシュする」」を参照してください。

TLS のクライアントを環境設定する

LDAP クライアントとは、たとえば、Internet Explorer、ICE のようなアプリケーションのことです。クライアントは、LDAP サーバが使用する認証局を認知している必要があります。

重要 : OpenText eDirectory 9.1 以降、ndsindex および ice を含むすべての LDAP ユーティリティで受け入れられるのは .PEM 形式の証明書のみになりました。.PEM 証明書を LDAP 操作で使用する方法について詳しくは、386 ページの 「Linux での LDAP ツールの使用」を参照してください。

OpenText eDirectory ツリーが構成されている場合、デフォルトでは構成によって次のものが作成されます。

- ツリーの認証局 (ツリー CA)
- ツリー CA からの KMO

LDAP サーバはこの認証プロバイダを使用します。

クライアントは、LDAP サーバが使用していると主張するツリー認証局を確認できるよう、信頼する証明書をインポートする必要があります。この証明書をサーバからインポートしておくと、サーバがその証明書を送信してきたときに、クライアントはそれを確認し正当なサーバであるかどうか確かめることができます。

クライアントが安全に接続できるよう、接続前にクライアントの環境設定をしておく必要があります。

クライアントによる証明書のインポート方法は、使用しているアプリケーションの種類によって異なります。各アプリケーションには、何らかの証明書をインポートする方法があります。IE と ICE では、方法が異なります。これらは異なる LDAP クライアントです。各クライアントは、それぞれの方法で信頼する証明書を探します。

ルート認証局をエクスポートする

ルート認証局は、証明書サーバを受け入れるときに自動的にエクスポートできます。

ルート認証局を手動でエクスポートするには、「736 ページの 「ルート認証局または公開鍵証明書のエクスポート」」を参照してください。

エクスポート機能により、指定したファイルが作成されます。ファイル名は変更できますが、マテリアルオブジェクトのタイプを認識できるよう、ファイル名に「DNS」または「IP」を残しておくことをお勧めします。また、サーバ名も残しておきます。

OpenText eDirectory との安全な LDAP 接続を確立するブラウザに、自己割り当て認証局をインストールします。

InternetExplorer など、Microsoft 社の製品で証明書を使用する場合は、.der 拡張子を残しておくようにします。

アプリケーションまたは SDK が証明書を要求する場合は、証明書をデータベースにインポートします。

Internet Explorer 5 の場合は、ルート証明書が自動的にエクスポートされ、レジストリが更新されます。これには、Microsoft が通常使用している .x509 拡張子が必要です。

クライアント証明書で認証を受ける

相互認証には、TLS セッションとクライアント証明書が必要です。サーバとクライアントの両方が、それぞれが自分の主張するオブジェクトであることを証明する必要があります。クライアントの証明書がトランスポート層で確認されます。しかし、LDAP プロトコル層では、LDAP バインド要求を出すまでクライアントは匿名になります。

この時点で、クライアントはその正当性をサーバに証明しましたが、LDAP にはまだ証明されていません。クライアント証明書に含まれた ID で認証を受けたい場合は、クライアントは SASL EXTERNAL メカニズムを使ってバインドされます。

1 OpenText Identity Console のホームページで、［**LDAP 環境設定**］タイルをクリックします。
2 ［**LDAP サーバ**］をクリックし、LDAP サーバオブジェクトの名前をクリックします。
3 ［**情報**］ドロップダウンメニューをクリックします。
4 ［**クライアント証明書**］ドロップダウンをクリックし、［**必須 (Required)**］を選択します。
これにより、相互認証が可能になります。
5 ［**保存**］>［**OK**］の順にクリックします。

サードパーティプロバイダの証明書を使用する

OpenText eDirectory のインストール時に、LDAP サーバにツリー認証局 (CA) が提供されます。LDAP キーオブジェクトは、その認証局に基づいています。クライアントが LDAP サーバに送信する証明書は、このツリー認証局から検証できます。

LDAP Services for OpenText eDirectory は、複数の認証局をサポートしています。OpenText のツリー認証局は 1 つの認証局にすぎません。LDAP サーバが、他の CA を持っている場合があります (例 : 外部団体の VeriSign* など) この追加 CA もルート認証局です。

LDAP サーバが複数の認証局を使用するように設定するには、LDAP サーバオブジェクトで ldapTLSTrustedRootContainer 属性を設定します。LDAP サーバが複数の認証局を参照することにより、クライアントは外部の認証局を使用できます。

LDAP プロキシユーザを作成および使用する

OpenText eDirectory は、認証されていないユーザに [Public] 識別子を割り当てます。LDAP プロトコルでは、認証されていないユーザは匿名ユーザになります。デフォルトでは、LDAP サーバは匿名ユーザに [Public] 識別子の権利を与えます。これらの権利により、非承認の OpenText eDirectory および匿名 LDAP ユーザは、[Public] 権を使用して OpenText eDirectory をブラウズすることができます。

また、LDAP サーバは匿名ユーザによる別のプロキシユーザの権利の使用を許可します。この値は LDAP グループオブジェクトにあります。OpenText Identity Console では、この値は、［代理ユーザ］フィールドで指定します。次の図は、OpenText Identity Console のこのフィールドを示しています。

***図 14-3** LDAP グループ*



プロキシユーザは識別名です。このプロキシ ID に、Public 識別子とは別の権利を与えることができます。代理ユーザを使用すると、OpenText eDirectory ツリー内の特定コンテナへの LDAP 匿名アクセスを制御できます。

注 : 代理ユーザのログイン制限は、すべての匿名 LDAP ユーザに適用する場合以外は設定しないでください。

シナリオ : NLDAP 代理ユーザを設定する — Digital Airlines 社はリサーチ会社の DataSure 社と契約を締結しています。DataSure 社は LDAP を使用して、Digital Airlines 社の Linux サーバである DigitalAir43 にアクセスし、リサーチの内容を保存します。あなたは DataSure 社に DigitalAir43 のディレクトリに対するパブリック権利を与えたくありません。

そのために、LDAP プロキシユーザを作成し、そのユーザに DataSure ディレクトリに対する特定の権利を割り当てます。LDAP グループオブジェクトにプロキシ識別名を作成し、サーバをリフレッシュします。サーバは自動的に、すべての新規または既存の匿名ユーザについて、プロキシユーザの権利の使用を開始します。

1. OpenText Identity Console のホームページで、［**ユーザ管理**］>［ ＋ **ユーザの追加**］の順にクリックし、パスワードなしで代理ユーザを作成します。
2. ユーザに NULL パスワードを割り当てます。
3. (オプション) 指定したディレクトリにプロキシユーザの権利を割り当てます。

4 Identity Console のホームページで、[LDAP 環境設定] > フォームの [タイプ] ドロップダウンの順にクリックし、[LDAP グループ] を選択し、[検索] をクリックして、[LDAP グループ] オブジェクトを選択します。

5 [情報] をクリックします。[代理ユーザ] フィールドで、[] をクリックして、すでに作成されている [LDAPProxy] ユーザをブラウズして選択し、[保存] をクリックします。

6 [OK] をクリックします。

SASL を使用する

SASL (Simple Authentication and Security Layer) は、異なるメカニズムを介して、接続ベースのプロトコルに認証のサポートやデータのセキュリティサービスを追加するためのメカニズムです。プロトコルとメカニズム間の整形式のインタフェースを提供します。さらに、データ整合性、データ機密性、その他のサービスとともに、データセキュリティ層内での後続のプロトコル交換のセキュリティを保護するためのプロトコルを提供します。

SASL は、メカニズムを再設計せずに新しいプロトコルが既存のメカニズムを再使用できるように設計されており、既存のプロトコルもプロトコルの設計を変更せずに新しいメカニズムを使用できるようにします。SASL を使用するために、各プロトコルは、どのメカニズムが使用されるかを識別する方法、メカニズム固有のサーバチャレンジとクライアント応答を交換する方法、および認証交換の結果を通信する方法を提供します。

SASL メカニズムには、大文字、数字、ハイフン、およびアンダースコアで構成される、文字列によって名前が付けられます。SASL メカニズム名は、IANA (Internet Assigned Numbers Authority) に登録する必要があります。

サーバが要求されたメカニズムをサポートする場合、認証プロトコル交換を開始します。これは要求されたメカニズムに固有の一連のサーバチャレンジとクライアント応答で構成されます。認証プロトコル交換時に、メカニズムは認証を実行し、クライアントからサーバへ承認 ID を送信して、メカニズム固有のセキュリティ層の使用をネゴシエートします。セキュリティ層の使用について合意すると、メカニズムは、それぞれの側で受信可能な最大の暗号テキストバッファサイズの定義つまりネゴシエートする必要もあります。

LDAP サーバでは次のメカニズムがサポートされます。

- DIGEST-MD5
- EXTERNAL
- NMAS_LOGIN
- GSSAPI

これらのメカニズムは、OpenText eDirectory のインストールまたはアップグレード時に、サーバにインストールされます。ただし、Linux では、nmasinst ユーティリティを使用して、NMAS メソッドをインストールする必要があります。

前述したとおり、LDAP サーバは、SASL に問い合わせて環境設定時にインストールしたメカニズムを検索し、インストールされたメカニズムを自動でサポートします。また、supportedSASLMechanisms 属性を使って rootDSE で現在サポートされている SASL メカニズムをレポートします。これらは登録されているメカニズムなので、利用には正しい命名規則を使用する必要があります。

LDAP バインドプロトコルでは、クライアントは認証に様々な SASL メカニズムを使用することができます。アプリケーションが LDAP バインド API を使用している場合は、単純認証を選択し、DN とパスワードを入力するか、SASL バインドを選択し、SASL メカニズム名と、そのメカニズムが要求する、関連する SASL 資格情報を提供する必要があります。

DIGEST-MD5

LDAP はバインド要求で DIGEST-MD5 メカニズムをサポートします。LDAP 単純認証 (DN およびクリアテキストパスワード) を要求する代わりに、DN と MD5 資格情報を提供することで、LDAP SASL バインドを要求します。DIGEST-MD5 メカニズムには TLS は必要ありません。LDAP サーバは、クリア接続およびセキュア接続の両方で DIGEST-MD5 をサポートします。

MD5 は、暗号化されたパスワードのハッシュを提供します。パスワードは、クリア接続でも暗号化されます。そのため LDAP サーバは、ポートがクリアテキストポートか暗号化されたポートかに関係なく、MD5 を使ったパスワードを受け入れます。他のユーザがこの接続を傍受しようとしても、パスワードは検出できません。ただし、接続全体に対してなりすましやハイジャックがなされる可能性はあります。

このメカニズムは LDAP SASL バインドです (単純認証ではありません)。そのため、インストール時に［**パスワードとの単純バインドに TLS を必要とする**］チェックボックスがオンになっていても、LDAP サーバはこの要求を受け入れます。

EXTERNAL

EXTERNAL メカニズムにより、ユーザ DN および資格情報がサーバに提供されたことが LDAP サーバに通知されます。そのため、バインド要求時には DN と資格情報は必要ありません。

LDAP バインド要求は、SASL EXTERNAL メカニズムを使用して、サーバに次を実行するよう指示します。

- EXTERNAL 層に資格情報を問い合わせる
- ユーザをその資格情報とユーザで認証する

その後、セキュアなハンドシェークが実行されます。LDAP サーバはクライアントから資格情報を要求しクライアントがサーバに資格情報を渡すと、サーバはクライアントから渡された証明書を受信し、その証明書を NMAS モジュールに渡して、証明書の中にどんな DN が指定されていてもユーザを認証します。

使用できる DN の証明書を使用するには、クライアントを設定する必要があります。証明書の設定に関する詳細は、「NMAS オンラインヘルプ (https://www.netiq.com/documentation/edir88/nmas88/data/bookinfo.html)」を参照してください。

クライアントが EXTERNAL メカニズムを送信する場合でも、LDAP サーバが要求の処理に失敗する場合があります。失敗の原因として可能性があるものを次に示します。

- 接続がセキュア接続ではない。
- 接続はセキュア接続だが、クライアントがハンドシェーク時に要求された証明書を提供しなかった。
- SASL モジュールを使用できない。

NMAS_LOGIN

NMAS は、さまざまなログインと認証方法を使用してネットワークに認証されるアプリケーションを作成することができる開発フレームワークです。NMAS フレームワークでは、NICI と OpenText eDirectory Services (OpenText eDirectory) を活用する、モジュール式のプラグイン方法を使用して、柔軟かつ拡張可能なログインと認証システムを設計することができます。

NMAS_LOGIN メカニズムにより、LDAP サーバに NMAS のバイオメトリック機能が提供されます。詳細については、「NetIQ モジュラー式認証サービス NDK (http://www.novell.com/documentation/developer/nmas/)」を参照してください。

GSSAPI

GSSAPI メカニズムにより、Kerberos ユーザがチケットを使用して OpenText eDirectory サーバへの認証を受けられるようになります。その際に、個別の LDAP ユーザパスワードの入力は不要です。この機能は、Kerberos インフラストラクチャがすでに配置された環境がある LDAP アプリケーションユーザ向けのものです。このようなユーザは、個別の LDAP ユーザパスワードを入力することなく、Kerberos サーバで発行されたチケットを使用して LDAP サーバへの認証を行うことができます。

GSSAPI の設定については、867 ページの付録 E「OpenText eDirectory での GSSAPI の設定」を参照してください。

NMAS ベースのログインを使用した LDAP 認証

OpenText eDirectory では、NMAS ログインがデフォルトで有効になっています。NMAS ログインを無効にするには、NDSD_TRY_NMASLOGIN_FIRST を false に設定します。

注 : OpenText eDirectory サービスを RHEL 7.x および SLES 12.x プラットフォーム上の pre_ndsd_start_custom スクリプトで実行するために必要なすべての環境変数を追加する必要があります。

LDAP サーバを使ってディレクトリを検索する

このセクションでは、次の情報について説明します。

- 433 ページの 「検索制限を設定する」
- 434 ページの 「参照を使用する」

検索制限を設定する

LDAP サーバオブジェクトの次の属性により、LDAP サーバのディレクトリ検索方法を制御することができます。

- 検索エントリの制限

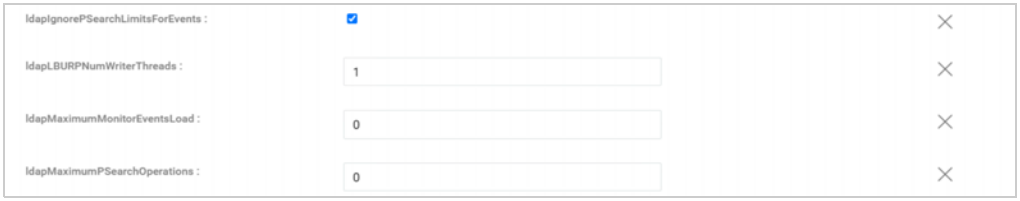
  検索のサイズを制限します。デフォルトは 0 で、サイズの制限はありません。LDAP サーバの負荷が大きくなりすぎないように、検索要求に対して LDAP サーバが返すエントリ数を制限できます。

  シナリオ : 検索のサイズを制限する – ユーザは、検索結果が何千件にもなりそうな、あるオブジェクトの検索を要求します。ただし、検索結果は 10 件に制限してあります。LDAP サーバは 10 件の検索結果を返すと検索を中止します。一致するデータがまだ存在しているが、検索が終了されたことを告げるシステムメッセージが表示されます。

- 結果送信時間

  サーバが検索を行う時間を制限します。デフォルトは 0 秒です。これは時間制限がないことを表します。

次の図は、OpenText Identity Console のこれらの属性を示しています。



1. OpenText Identity Console のホームページで、[**LDAP 環境設定**] タイルをクリックします。
2. [**LDAP サーバ**] をクリックします。
3. [**その他**] ドロップダウンメニューをクリックします。
4. 制限セクションをスクロールして値を入力し、[**保存**] をクリックします。

クライアントも、検索要求に制限を設定することもできます (たとえば検索を 2 秒に制限するなど)。クライアントの制限がサーバの制限と競合する場合、LDAP サーバは値が小さい方の要求を採用します。

検索は、アクセス制御リスト (ACL) に基づいて実行されます。このため、匿名検索の場合は、ディレクトリに何千というエントリが存在していても、Public 権で見ることのできるごく一部のエントリしか返されないことがあります。

参照を使用する

参照は、名前を解決するためのクライアント中心の方法です。LDAP クライアントが LDAP サーバに要求を送信すると、LDAP サーバは要求操作のターゲットエントリをローカルに見つけようとします。LDAP サーバはターゲットエントリを見つけられないと、所有する知識参照を使用して、そのエントリについてのより多くの知識を持つ第 2 の LDAP サーバへの参照を生成します。第 1 のサーバは、参照情報を LDAP クライアントに送信します。

次に、LDAP クライアントは第 2 の LDAP サーバへ接続し、操作を再実行します。第 2 の LDAP サーバが操作のターゲットエントリを保持している場合は、そのサーバが操作を実行します。エントリを保持していない場合は、第 2 の LDAP サーバもまた参照をクライアントに送信します。この操作は、次のいずれかの状況になるまで続けられます。

- クライアントがターゲットエントリを保持するサーバと接続し、要求する操作が実行できる。
- LDAP サーバが、エントリは存在しないとのエラーを返す。
- LDAP サーバがこれ以上参照先がないことを通知する。

LDAP for OpenText eDirectory 8.7 で導入された機能により、参照の動作が以前のバージョンの OpenText eDirectory および NDS から少し変更されました。これにより、LDAP サービスの環境設定方法も変更されました。

デフォルトの参照

通常、デフォルトの参照 URL には、ツリーのルートを保持するサーバを指す LDAP URL が含まれています。LDAP URL の形式は ldap:// *ホスト*: *ポート*です。

［デフォルト参照 URL］フィールドに、デフォルトの参照先を入力します。



これまで、OpenText eDirectory LDAP サーバは、多くのフェイルオーバー時にデフォルトの参照先を送信していました。これが、この動作を予期しないユーザを混乱させる結果になっています。そこで、LDAP Services for OpenText eDirectory では、サブオーディネート参照で、どんな場合にデフォルトの参照先が送信されるかを指定できるようになりました。

この新しいオプションは、LDAP サーバオブジェクトと LDAP グループオブジェクトの ldapDefaultReferralBehavior 属性の値 (設定) で指定します。値は次のビットのビットマスクである整数です。

ビット	値
0x00000001	ベース DN が見つかりません
0x00000002	ベース DN は、利用できない OpenText eDirectory サーバ上にあります
0x00000004	検索スコープのエントリは、利用できない OpenText eDirectory サーバ上にあります

LDAP サーバがその操作に対して「常に参照する」に設定されており、リストされたいずれかの条件と一致し、対応する値が設定されている場合、デフォルトの参照先が返されます。

検索操作の参照先を設定する

LDAP for eDirectory 8.7 で導入された機能により、参照の動作が以前のバージョンの OpenText eDirectory および NDS から少し変更されました。これにより、LDAP サービスの環境設定方法も変更されました。

OpenText eDirectory ツリー内の他の OpenText eDirectory サーバに参照先を返すように eDirectory LDAP サーバを設定することができます。デフォルトでは、LDAP サーバはユーザに代わってすべての操作を他の OpenText eDirectory サーバにチェーンし、参照先は返しません。

eDirectory 8.7 より以前は、参照先オプションの設定は LDAP グループオブジェクトでだけしか使用できませんでした。OpenText eDirectory 9.0 以降では、LDAP サーバオブジェクトにもこれらのオプションを設定できるようになりました。LDAP サーバオブジェクトの設定により、LDAP グループオブジェクトの設定は上書きされます。

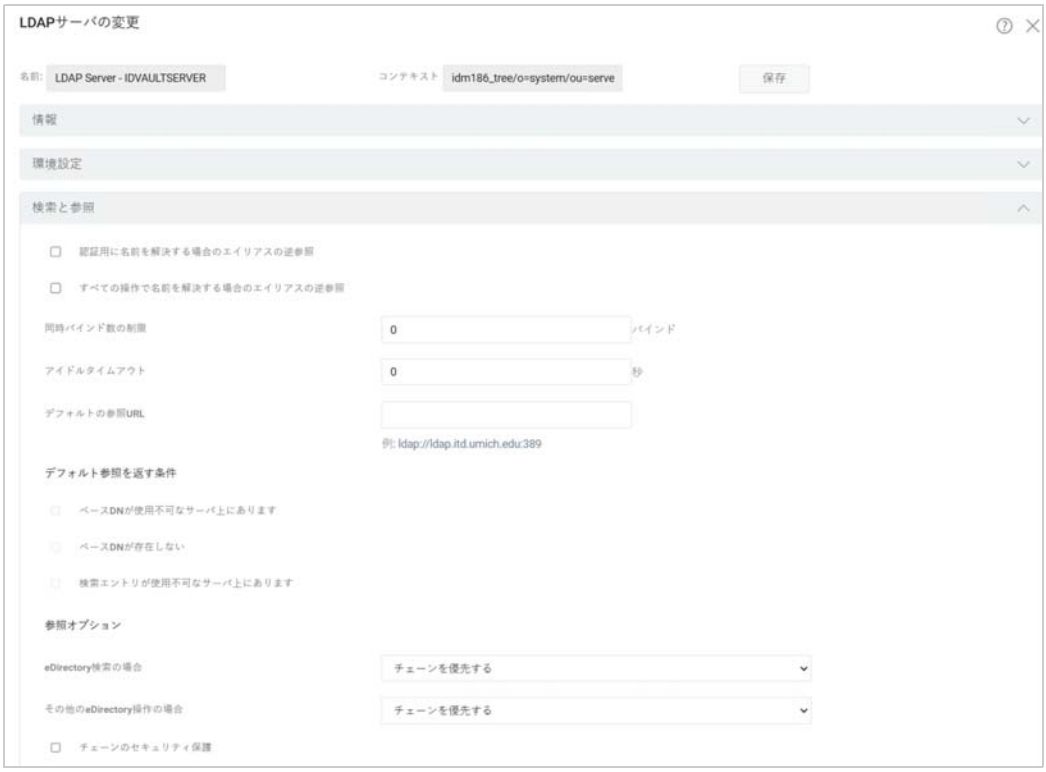
dapSearchReferralOption 属性を操作することにより、照会先オプションを設定することができます。LDAP Services for eDirectory 8.7 より以前は、この属性を次のオプションに設定することができました。

- 437 ページの 「チェーンを優先する」(デフォルトオプション)
- 438 ページの 「参照を優先する」
- 438 ページの 「常に参照する」

これらの参照オプションは、OpenText eDirectory ツリー内の他の OpenText eDirectory サーバの参照およびチェーンでだけ使用できます。この設定は、信頼されていないパーティションからの参照は制御しません。そのため、[照会先オプション] ドロップダウンリストでオプション ([常にチェーンする] など) を選択しても、他のサーバの信頼されていないパーティションからは参照先が送信されます。

LDAP Services for eDirectory 8.7.a では［常にチェーン］オプションにより、OpenText eDirectory DSA 以外の上方参照がサポートされています。詳細については、436 ページの「常にチェーンする」を参照してください。

次の図は、検索およびその他の操作に使用する［LDAP 参照］ドロップダウンリストを示しています。



OpenText eDirectory 操作にはこの他に、「追加」、「削除」、「編集」、「バインド」の各操作の参照があります。

常にチェーンする

［常にチェーンする］オプションは、「まったく参照しない」ように設定するオプションです。このオプションを選択すると、OpenText eDirectory LDAP サーバは、OpenText eDirectory ツリー内にある他の OpenText eDirectory サーバに参照先を返しません。LDAP サーバは、要求を出したクライアントに代わって他の LDAP サーバをチェックし、クライアントに参照先を返します。

［常にチェーン］オプションは、OpenText eDirectory をグローバル連結ツリーのサブオーディネートサーバとして使用している場合に適しています。

この参照先オプションは、eDirectory ツリー内の参照先の処理設定にのみ使用します。このオプションが OpenText eDirectory サーバ以外のサーバの参照の動作に影響することはありません。

他の OpenText eDirectory サーバへの参照をブロックすることにあまり意味はありませんが、これが重要になる場合もあります。eDirectory 8.7 以降のサーバ上の信頼されていないデータを古いバージョンの OpenText eDirectory サーバ上で複製すると、古いサーバを参照したときにクライアントアプリケーションのグローバルツリーが歪んで表示されることがあります。

たとえば、LDAP クライアントは LDAP サーバの参照をキャッシュし、最後に通信したサーバに要求を送信するとします。クライアントが上方参照をサポートする OpenText eDirectory サーバに要求を送信するよう設定されている場合、クライアントのグローバルツリーは正常に表示されます。

しかし、eDirectory 8.7 以前の LDAP サーバは、信頼されていない領域と上方参照を認識しません。このため、クライアントが OpenText eDirectory ツリー内の以前のバージョンの OpenText eDirectory サーバの参照に従い、要求をそのサーバに送信し続けると、以前のバージョンの LDAP サーバにより、信頼されていないデータが実際のディレクトリツリーデータであるかのように表示されてしまいます。

ただし、クライアントによっては、RootDSE の supportedFeatures 属性を取得し、サーバが上方参照をサポートしているか確認するものもあります。

チェーンを優先する

［チェーンを優先する］オプションを選択すると、通常、検索操作で参照先は返されません。LDAP サーバはその代わり、すべての OpenText eDirectory DSA に対する検索操作を実行します。

ただし、持続的検索制御を設定して検索を実行する場合は例外となります。OpenText が実装する持続的検索ではチェーンがサポートされていないため、検索スコープがローカルに限られていなければ参照が送信されます。

LDAP サーバが検索操作を受信します。ツリーのエントリがローカルに格納されていない場合、サーバは自動的に他のサーバにチェーンします。エントリの検出後、LDAP サーバは LDAP クライアントのプロキシとして機能します。LDAP サーバは LDAP クライアントがバインドされたものと同じ識別情報を使用してリモートサーバの認証を受け、そこで検索操作を続行します。

最初に要求を受信した LDAP サーバが、LDAP クライアントにすべての検索エントリと検索結果を送信します。この LDAP サーバが要求をすべて処理するため、LDAP クライアントからは他のサーバが関与していることはわかりません。

OpenText eDirectory でチェーンを使用すると、ある LDAP サーバに多くのデータがない場合でも、そのサーバがツリー全体のデータを保持しているかのように見えます。

［チェーンを優先する］は、パーティションに深くかかわるオプションです。

シナリオ : 他のパーティションで情報を探す — Digital Airlines 社で、ユーザが LDAP サーバ DAir43 に［チェーンを優先する］オプションを選択しました。DAir43 はパーティション A にあります。パーティション B は A のサブパーティションで、LDAP サーバ DAir44 はこのパーティションにあります。

ある LDAP クライアントが検索を要求します。DAir43 は、エントリをローカルで検索しますが、データが一部しか見つかりません。DAir43 は、要求されたエントリを持つ DigitalAir44 に自動的にチェーンします。DAir44 は、DAir43 にデータを送信し、DAir43 は、LDAP クライアントにエントリを送信します。

［チェーンを優先する］オプションを使用すると、操作が持続的検索である場合を除き、LDAP サーバは必要に応じて検索を他のサーバにチェーンします。持続的検索の詳細については、「447 ページの 「持続的検索 : OpenText eDirectory イベントの設定」」を参照してください。

参照を優先する

［参照を優先する］オプションを選択すると、必要に応じ、参照の検索結果が OpenText eDirectory ツリー内の他の eDirectory サーバに返されます。この参照は、データを持つサーバが動作可能であり、LDAP サービスが稼動していることをローカルサーバが確認した場合のみ送信されます。それ以外の場合、操作は他のサーバにチェーンされるか、他のサーバが動作していない場合は処理に失敗します。

パーティションが 2 つあり、サブツリー検索を実行するとします。ローカルサーバから検索エントリがすべて検出されるまで検索が実行されます。そこで、今度は他のサーバの検索を実行します。データのレプリカ (そのパーティション) を持つサーバが nldap.nlm も実行している場合、LDAP サーバは LDAP 参照を確立し、それを LDAP クライアントに返します。

レプリカのあるサーバが nldap.nlm を実行していない場合、LDAP サーバは要求を他のサーバにチェーンし、そこで検索を完了します。

nldap.nlm が起動されると、LDAP サーバはその LDAP サーバが参照先となっている OpenText eDirectory と通信します。クライアントが参照を受信したのに、その参照が停止した場合は、LDAP サーバが実行されていません。

常に参照する

［参照を優先する］オプションは、デフォルト参照がさまざまなフェイルオーバー (たとえば、オブジェクトが見つからなかったり、サーバがダウンしているなど) の状況で送信される場合を除き、［参照を優先する］と同じロジックに従います。

残りのデータのある他のサーバで LDAP サービスが実行されていない場合、最初の LDAP サーバは要求を第 2 のサーバへチェーンしません。

［常に参照する］オプションを設定している場合には、デフォルト参照を指定することができます。［デフォルトの照会先］フィールドで 2 つの異なるベンダの LDAP サーバを結合し、独自のディレクトリツリーを構築することができます。

シナリオ : デフォルトのサーバを使用する ― 1 つの LDAP ツリーがあるとします。ツリーの一部には OpenText eDirectory のサービスが適用されています。サブオーディネイトパーティションには iPlanet のサービスが実行されています。［デフォルトの照会先］フィールドに、iPlanet サーバの URL を入力します。ある LDAP クライアントが検索を要求します。

ベース DN を解決できないため、LDAP サーバは［デフォルトの照会先］フィールドに入力された文字列をクライントに送ります。LDAP クライアントはこの参照の URL で指定された場所を参照して iPlanet サーバに接続し、ここで検索は完了します。

デフォルト参照が設定されており、サーバが探しているベース DN を見つけられない場合、クライアントはデフォルト参照を受け取ります。

参照の形式は、LDAP URL です。例 : LDAP://123.23.45.6:389

LDAP サーバがデフォルト参照をクライアントに送信するとき (ベース DN が利用できない場合) は、サーバはこれにスラッシュ (/) とクライアントが検索中の DN を追加します。デフォルト参照と追加された情報がクライアントに送信されます。クライアントはデフォルト参照で指定したサーバに検索要求を送信します。

LDAP グループオブジェクトには、デフォルト参照の文字列フィールドがあります。LDAP サーバは、そのデータを文字列として扱います。このとき、確認は行われません。入力された文字列が、参照の先頭に追加されます。また、なんらかのデータが参照に追加されます。LDAP サーバが受け入れる文字列の形式は、URL のような形式になります。

LDAP が実行されている他の OpenText eDirectory サーバの参照がクライアントに返されるとき、クライアントは 1 つのサーバにつき 2 つの参照を受信します。

- クライアントをクリアテキストポートに導く参照
- クライアントをセキュアポートに導く参照

2 つの参照を区別するために、クリアテキスト参照には ldap://、セキュアポートの表示には ldaps:// が付きます。

サーバからの参照の場合は、ポート番号を追加します。

他の操作の参照を設定する

履歴参照オプション設定は通常、検索操作にのみ使用します。他の処理に比較オプションを適用する場合は、ldapOtherReferralOption 属性が使用されます。この属性により、同じ値を使って検索以外の操作の動作を制御できます (参照を送信しないバインドは除きます)。

参照フィルタリング

ツリーで実行されている複数のレプリカサーバがあり［参照を優先する / 常に照会］オプションを使用して参照を返すように LDAP サーバを設定してある場合、要求された操作内の DN によって識別されるオブジェクトがローカルに存在しないと、LDAP サーバは参照を返します。そのような場合、LDAP クライアントはサーバに要求を送信し、サーバはそのオブジェクトを保持しているすべての LDAP サーバの参照リストを返します。この参照リストを使用して、LDAP クライアントはこれらの参照のいずれかに従って操作を実行します。クライアントがリソース不足のサーバまたは低速リンクを経由するサーバへの参照に従うことを選択した場合、クライアントはサーバから応答が遅いと認識する場合があります。この場合、LDAP クライアントのパフォーマンスが影響を受けます。LDAP アプリケーション開発者はサーバとネットワークの構成に関する完全な知識を持っていないため、この問題の解決策は、LDAP サーバに参照フィルタリングメカニズムを提供し、特定のサーバの参

照を返すことです。管理者は、ネットワークおける LDAP サーバの性質やネットワークリンクの速度など、必要な知識を持ち、参照フィルタリングの適切な設定を行うことができる場合があります。

属性「referralIncludeFilter」と「referralExcludeFilter」を使用して、LDAP グループオブジェクトに参照フィルタを設定します。これらの属性でこれらのフィルタを設定すると、この LDAP グループオブジェクトに属するすべての LDAP サーバにこの設定が適用されます。LDAP サーバは、referralIncludeList フィルタに一致するすべての LDAP 参照を返し、referralExcludeFilter フィルタに一致するものをドロップします。

referralIncludeFilter のみを指定した場合、referralIncludeFilter 値に一致する LDAP 参照が LDAP クライアントに返され、それ以外のすべての参照は参照リストから除外されます。同様に、referralExcludeFilter のみを指定した場合、referralExcludeFilter 値に一致しない LDAP 参照が LDAP クライアントに返されます。両方のフィルタが存在し、参照がこれらのフィルタのどちらにも一致しない場合、参照は除外されます。

使用可能なすべての参照がフィルタによって許可されていない場合、サーバは使用可能な参照がないかのように動作し、LDAP_OTHER (80) を返します。一部のクライアントツールはこれを「不明なエラー」としてレポートします。これらのフィルタ属性を追加または変更した後、LDAP サーバがリフレッシュされなかった場合、変更内容は次の自動リフレッシュ後に有効になります。

現時点では、これらのフィルタ属性の追加または変更は、OpenText Identity Console のタブでのみ実行できます。

LDAP 参照フィルタリングを指定する形式 —LDAP 参照フィルタの形式は、次の単純な IP アドレス形式です。

[ldap://] | [ldaps://] *IPAdress*[:port]

ここで、クリアテキストポートまたは TLS ポートを指定することは、先頭に ldap:// または ldaps:// 文字列を付加することと同じです。ldap または ldaps のどちらも指定しない場合、一致検索フィルタはクリアテキスト参照と TLS 参照の両方に適用されます。

例 :

例	説明
1.2.3.4	# 任意のポートで LDAP と LDAPS の両方の参照に一致
1.2.	# 1.2.X.Y というすべての IP アドレスに一致
1.2.3.	# 1.2.3.Y というすべての IP アドレスに一致
ldap:// または ldap://*	# すべてのクリアテキストポート LDAP 参照に一致
ldaps:// または ldaps://*	# すべての SSL ポート LDAP 参照に一致
*	# すべてに一致
ldaps://5.6.7.8:636	# 5.6.7.8 という IP アドレスの SSL ポート 636 に一致

これらのフィルタ属性 (referralIncludeFilter および referralExcludeFilter) は複数値です。必要な数の一致フィルタを選択できます。

サンプルシナリオ

- LDAPサーバが1.2.X.YというIPアドレス(X = {0～255}およびY = {0～255})の参照のみを返し、他のすべてを除外するには、次のように入力します。

  referralIncludeFilter = { 1.2 }

- LDAPサーバが164.99.X.YというIPアドレス(Xは100以外)に一致するすべての参照を除外し、164.99.100.Y に一致する参照を返すには、次のように入力します。

  referralIncludeFilter = { 164.99.100., "*"}

  referralExcludeFilter = { 164.99. }

  ここで、IP アドレス 164.99.100.Y は referralExcludeFilter に一致しますが、これらの IP アドレスは referralIncludeFilter とより多くのフィールドで一致するため、これらの参照は LDAP クライアントに返されます。

  注 : 部分的な IP アドレスを指定する際、末尾の「.」は省略できます。

- LDAP サーバがクリアテキストポート参照のみを返し、SSL ポート参照をドロップするには、次のように入力します。

  referralIncludeFilter = { "ldap://" }

  または

  referralExcludeFilter = { "ldaps://" }

- LDAP サーバが IP アドレスのセットから返し、それ以外のすべての IP アドレス参照をドロップするには、次のように入力します。

  referralIncludeFilter = { 1.2.3.4, 2.3.4.5:389, 3.4.5.6:636, ldaps://4.5.6.7 }

  referralExcludeFilter = { "*" }

  注 : ここで、referralExcludeFilter は必要ありません。いずれかの入力された referralIncludeFilter は他のすべてを除外することを意味します。

- 次のように 2 つのフィルタがあるとします。

  referralIncludeFilter = { 1.2.3.4 }

  referralExcludeFilter = { 2.3.4.5 }

  IP アドレス 3.4.5.6 の参照は、referralExcludeFilter に一致しないとしても、referralInclude フィルタとも一致しないので除外されます。

無効なフィルタ — 次のフィルタはサポートされていません。

「.2.3.4」または「*.2.3.4」は、IP アドレス *X*.2.3.4 に一致しません。

「2.3.4*」は 2.3.41 または 2.3.42 のような IP アドレスに一致しません。

sever1.mydomain.com または *.mydomain.com のような DNS 名はサポートされていません。最初のポートから最後のポートまでを指定した参照 IP アドレスの許可など、フィルタにポート範囲を追加することはできません。これらの属性にこれらのフィルタの値を追加する前に、検証チェックは行われません。ただし、無効なフィルタの場合、LDAP サーバはそれらのフィルタを無視し、ログを ndsd.log ファイルに記録します。

既知の問題 —LDAP rootDSE 検索では、LDAP URL 形式のレプリカサーバがある場合、altServers が返されます。これらの URL は、このメカニズムを使用してフィルタ処理されません。

ManageDsaIT の非サポート

LDAP Services for OpenText eDirectory では、OpenText eDirectory ツリー内の OpenText eDirectory サーバの分散関係は、ManageDsaIT 制御以外の方法で管理されます。ManageDsaIT 制御によって、LDAP クライアントが OpenText eDirectory サブオーディネートまたは相互参照の問い合わせや更新を実行することはできません。

サポートされていない機能

LDAP Services for OpenText eDirectory は、サブオーディネートリファレンスをサポートしていません。信頼されたパーティションのサブオーディネートパーティションとして信頼されていないパーティションを作成したり、そのパーティションから参照を送信させると失敗する場合があります。これを行う場合、参照は操作のベース DN を解決するときのみ送信されます。SearchResultReferences は送信されません。

信頼されていない領域のデータの分散更新はサポートされていません。ルートサーバで名前の変更があった場合、名前の変更を信頼されていない領域に同じデータを持っている OpenText eDirectory サーバにコピーするような組み込みのメカニズムはありません。

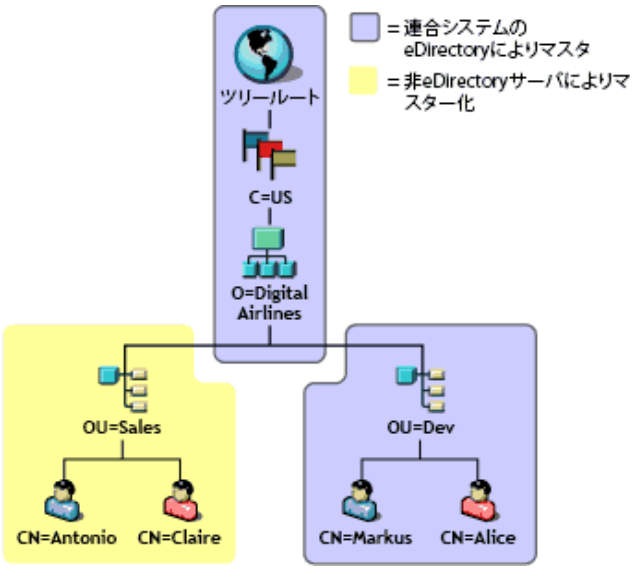
上方参照を設定する

組織の規模が大きくなると、さまざまなベンダの LDAP サーバソフトウェアを使用したディレクトリツリーが必要になります。このようなツリーを、グローバル連結ツリーといいます。LDAP Services for OpenText eDirectory には、参照を連結ツリーの上方の DSA に返す機能があります。

シナリオ : 連結ツリーでの上方参照

Digital Airlines 社には、あるネットワーク担当者がいます。Digital Airlines 社のディレクトリツリーのルート (ツリーのルートから O=Digital Airlines まで) のマスタは、OpenLDAP サーバ上にあります。ある組織 (OU=Sales) のマスタは OpenText eDirectory サーバにあり、別の組織 (OU=Dev) は iPlanet サーバ上にあります。

次の図は、このツリーを示します。



OpenText eDirectory にマスタがあるのは、OU=Sales のパーティション内のデータだけです。他の領域のデータのマスタは eDirectory 以外の DSA 上にあります。ネットワーク担当者は、操作の対象が O=Digital Airlines より上の領域、または OU=Sales 階層に属さない O=Digital Airlines より下の領域である場合、上方参照を返すように LDAP サービスを設定します。

ベース DN が「OU=Dev,O=Digital Airlines,C=US」の OpenText eDirectory LDAP サーバに操作が送信されます。そのエントリを保持するサーバ、またはそのエントリを保持するサーバを認知しているサーバを指す参照が返されます。

同様に、O=Digital Airlines,C=US が対象のサブツリー検索でも、ルート DSA の参照が返されます。するとルート DSA が、OU=Sales および OU=Dev のマスタである DSA の照会を返します。

LDAP サービスにより、OpenText eDirectory サーバがこのツリーに参加するのにデータ階層データを信頼されていないパーティションに持つことができます。信頼されていない領域のオブジェクトに含まれるのは、正しい DN 階層を構築するのに必要なエントリだけです。これらのエントリは、X.500 の “Glue” エントリに類似しています。

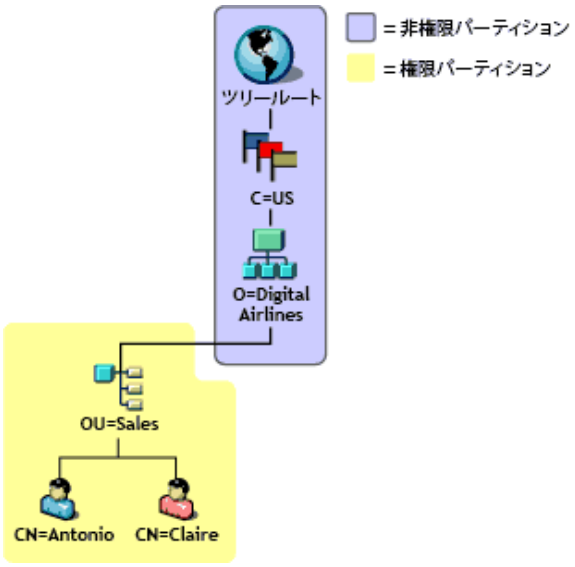
このシナリオでは、ルート、C=US、O=Digital Airlines オブジェクトは OpenText eDirectory サーバの信頼されていない領域にあります。

OpenText eDirectory では、知識情報 (参照データ) を信頼されていない領域に置くことができます。この情報は、LDAP クライアントに参照を返すのに使用されます。

LDAP 操作を OpenText eDirectory ツリーの信頼されていない領域で実行すると、LDAP サーバは正しい参照データを検索し、クライアントに参照を返します。

信頼されていない領域を作成する

次の図は、OpenText eDirectory サーバの連結ツリーが保持する実際のデータを示しています (442 ページの 「シナリオ : 連結ツリーでの上方参照」 を参照)。



エントリは、マスタが他の DSA 上にあっても、OU=Sale s 上に配置されます。これは、OpenText eDirectory サーバが持つエントリに適切な DN を提供できるようにするためです。

信頼されていない領域を作成するには、次を実行します。

1 信頼されていないデータは信頼されたデータとは切り離します。

信頼された領域の最上部に、パーティション境界を作成します。他に指定がない場合、OpenText eDirectory サーバはすべてのデータに対して信頼されたサーバであるとみなされます。

2 ルートパーティションに信頼されていないパーティションとマークします。

2a 信頼属性を、パーティションの最もルートに近い属性に追加します。

2b 値が 0 の信頼属性を作成します。

3 信頼されていない領域の最下部に境界線をひきます。

このサーバの信頼されたサブツリー領域にパーティションルートを作成します。たとえば上の図の場合は、パーティションルートは OU=Sale エントリにあります。新しいパーティションには、0 に設定された信頼された属性はありません。そのため、サーバはパーティションに対し信頼されたサーバであることになります。

4 LDAP サーバをリフレッシュします。

LDAP サーバは、その設定がリフレッシュされるたび、信頼された領域および信頼されていない領域の境界をキャッシュします。手動でサーバの設定をリフレッシュしない場合、サーバは 30 分ごとのバックグラウンドタスクで自動的にリフレッシュします。

複数のパーティションがある場合は、信頼された領域のチェーンに重ねることができます。ただし、LDAP Services for eDirectory では、すべての信頼されていないパーティションは連続していなければならず、ローカルレプリカがそれを保持している必要があります

参照データを指定する

操作が信頼されていない領域で実行されていることが検出されると、LDAP サーバは参照をクライアントに返すのに使用する情報を探します。この参照情報は、次のいずれかになります。

- 信頼されていない領域のいずれか、またはすべてのエントリにある情報
- サーバの環境設定データを持つ LDAP サーバまたは LDAP グループオブジェクト上の、デフォルト参照として指定された情報

信頼されていない領域のエントリが持つ参照情報は、即時上方参照です。このような参照情報は、複数の値をとる ref 属性から構成されています。この属性の詳細については、RFC 3296 (http://www.ietf.org/rfc/rfc3296.txt) を参照してください。

デフォルト参照設定が持つ参照情報は上方参照で、値を 1 つとります。X.501 の immSuper および suprDSE タイプを参照してください。

参照データは LDAP URL の形式で保持されますが、これにはホストと (オプションで) 参照先の DSA のポートだけしか指定されていません。この参照データの例は次のようになります。

```
ldap://ldap.digital_airlines.com:389
```

LDAP サーバは操作のベース DN(見つからない場合は一致した DN) を参照します。ベース DN に参照情報が含まれる場合、LDAP サーバはその情報を参照先として返します。

参照情報が見つからない場合、LDAP サーバはツリーの上方向に向かって参照情報を探します。すべてのエントリを検索しても参照情報が見つからない場合、LDAP サーバは上方参照を返します。この参照先は、LDAP グループまたは LDAP サーバオブジェクト上のデフォルト参照設定にあります。

即時上方参照の追加

immediateSuperiorReference と呼ばれる補助オブジェクトクラスを信頼されていない領域のエントリに追加することができます。この補助クラスは、1 つ以上の LDAP URL とともに作成される ref 属性を追加します。それぞれの URL は DSA のホスト名と (オプションで) ポートを指します。

上方参照を追加する

これまで、LDAP グループオブジェクトは ldapReferral 属性を持っていました。この属性は、OpenText eDirectory ツリーの他の OpenText eDirectory サーバに参照を返すときに発生する、さまざまなフェイルオーバーの状況で使用されるデフォルトの参照先を保持していました。LDAP Services for OpenText eDirectory では、この属性は、連結ツリーの上方 DSA のデフォルト参照を 1 つ指定するのに使用します。

また、ldapReferral 属性が LDAP サーバオブジェクトに追加されました。ldapReferral 属性に LDAP サーバオブジェクトの値が含まれる場合、この設定により LDAP グループオブジェクトの同じ属性の値は上書きされます。この動作により、グループに属するすべての LDAP サーバに特定のデフォルト参照を設定し、1 つか 2 つのサーバのデフォルト参照を別のデフォルトで上書きすることができます。

ldapReferral 属性の値は、LDAP URL です。URL には、ホストと参照先の DSA のポート (オプション) が含まれます。

LDAP で参照情報を更新する

上記のステップに従い、LDAP を使ってこのタスクを実行しても、即時上方参照を追加することはできないことが多くありました。これは、ルートパーティションがすでに信頼されていないとマークされており、LDAP はパーティション内のデータに対するどの操作についても参照を送信していたためです。

信頼されていない領域の情報の更新または問い合わせを行うには、LDAP 要求に ManageDsaIT 制御を設定する必要があります。この制御の詳細については、RFC 3296 (http://www.ietf.org/rfc/rfc3296.txt) を参照してください。この制御により、LDAP サーバは信頼されていない領域全体を信頼された領域であるかのように扱うことができます。

注 : 上方参照機能は、LDAP でのみ利用できます。他のプロトコル (NDAP など) には、信頼属性が存在することによる影響はありません。そのため、OpenText Identity Console を使用した信頼されていない領域のデータの参照や更新が妨害されることはありません。

影響を受ける操作

信頼されていない領域と上方参照は、次の LDAP 操作に影響します。

- 検索と比較
- 編集と追加

  DN 構文属性値はチェックされません。そのため、グループメンバー属性は、信頼されていない領域のエントリを指す DN を含むことができます。
- 削除
- リネーム (moddn)
- 移動 (moddn)

親 DN が信頼されていない領域にある場合、affectsMultipleDSAs エラーが返されます。

- 拡張

上方参照のサポートの有無を確認する

上方参照は、Novell LDAP Services for OpenText eDirectory 8.7 以降でのみサポートされています。ルート DSE の supportedFeatures 属性により、OpenText eDirectory サーバがこの機能をサポートしているかどうかを確認できます。supportedFeatures 属性の値が OID 2.16.840.1.113719.1.27.99.1 である場合は、この機能はサポートされています。その他のルート DSE オブジェクトに対する確認方法では、次が変更されています。

- namingContexts

  この属性は、サーバが信頼されているローカル DSA に保持されるパーティションルートだけを表示します。信頼されていないパーティションルートは表示されません。

- altServer

  この属性は、ローカルサーバと信頼されていないパーティションだけを共有する他の OpenText eDirectory サーバをリストしません。

- superiorReference

  この属性は、DSA の上方参照を通知します。この値は、LDAP サーバまたは LDAP グループオブジェクト上の ldapReferral 属性を更新することにより管理されます。

持続的検索 : OpenText eDirectory イベントの設定

OpenText eDirectory では、ディレクトリ内で重大なイベントが発生したときにアプリケーションに通知するためのイベントサービスが用意されています。これには、ディレクトリサービスに関する一般イベントも含まれます。それ以外は OpenText eDirectory とその機能に固有のイベントです。

OpenText eDirectory イベントは、LDAP プロトコルに対して、次の 2 つの異なる拡張を通してアプリケーションに通知されます。

- 持続的検索制御の実装

  OpenText eDirectory の持続的検索機能は、最初の一致するエントリが返された後も続行される検索操作です。持続的検索は LDAP v 3 の検索操作が拡張されたもので、クライアントからサーバへの検索結果内で更新をチェックする作業が不要になります。持続的検索制御により、クライアントは、ベース DN、検索スコープ、検索フィルタなどを指定する通常の LDAP 検索操作を実行することができます。その後、サーバは最後に SearchResultDone メッセージを返すのではなく、操作による接続が維持されます。このため、クライアントは検索結果に含まれるのエントリが変更されるたびに、最新のエントリを受け取ることができます。これにより、更新が発生するたびにクライアントは目的のエントリのキャッシュを維持したり、何らかのロジックをトリガすることができます。

記事「Persistent Search: A Simple LDAP Change Notification Mechanism (http://www.ietf.org/proceedings/01mar/I-D/ldapext-psearch-03.txt)」では、この拡張についてさらに詳しく説明しています。

- イベントの監視 (OpenText eDirectory 独自の拡張 LDAP 操作)

OpenText eDirectory イベントサービスを使用するアプリケーションは、ディレクトリに対する大きな計算負荷となることがあります。そこで、さまざまな管理パラメータを使用して、個々の OpenText eDirectory サーバにおけるイベントサービスの使用を制御することができます。これらのパラメータは LDAP サーバオブジェクト上に格納されます。OpenText Identity Console を使用して、これらのパラメータを設定します。

イベントサービスを使用する特定のアプリケーションにより、これらのパラメータを特定の値に指定するよう要求される場合があります。特定のアプリケーション独自の要求は、そのアプリケーションのマニュアルに示されています。

詳細については、「Understanding and Using Persistent Search in eDirectory (http://support.novell.com/techcenter/articles/dnd20030204.html)」を参照してください。

持続的検索の管理

OpenText Identity Console を使用すると、持続的検索を表示または編集することができます。

1 OpenText Identity Console のホームページで、[**LDAP 環境設定**] タイルをクリックします。

2 [**LDAP サーバ**] オブジェクトをクリックします。

3 [**その他**] ドロップダウンメニューをクリックします。

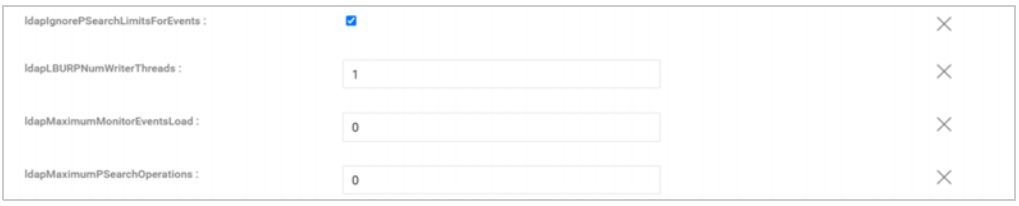
4 デフォルトで、[LdapEnablePSearch] チェックボックスがオンになっています。このサーバの持続的検索を無効にして禁止するには、チェックボックスをオフにします。

注 : 以前に確立された持続的検索操作を無効にすると、このオプションを無効にしてサーバをリフレッシュした後も操作が継続する場合があります。

---

5 このサーバ上の同時持続的検索の数を制御します。

[LdapMaximumPSearchOperations] フィールドに値を指定します。0 を指定すると、同時持続的検索数は無制限になります。



6 サイズおよび時間の制限を無視するかどうかを制御します。

持続的検索要求が最初の検索結果セットを送信した後にサイズと時間の制限を無視するかどうかを制御するには、[LdapIgnorePSearchLimitsForEvents] チェックボックスを選択します。

このオプションを選択しない場合、すべての持続的検索操作は検索制限の制約を受けます。サイズと時間のいずれかの制限に達した場合、検索操作は失敗し、該当するエラーメッセージが返されます。

7 ［保存］>［OK］の順にクリックします。

イベントの監視拡張操作の使用を制御する

1 OpenText Identity Console のホームページで、［LDAP 環境設定］タイルをクリックします。

2 ［LDAP サーバ］オブジェクトをクリックします。

3 ［その他］ドロップダウンメニューをクリックします。

4 クライアントアプリケーションがこの LDAP サーバ上でイベントを監視できるかどうかを制御します。

クライアントアプリケーションがこの LDAP サーバ上でイベントを監視できるようにするには、［LdapEnableMonitorEvents］チェックボックスをオンにします。

イベントの監視を無効にするには、チェックボックスをオフにします。



5 イベント監視アプリケーションがサーバ上に置くことのできる最大負荷を制御します。

［LdapMaximumMonitorEventsLoad］フィールドに値を入力します。

イベントデータの処理と、監視対象アプリケーションへのイベント通知の送信は、LDAP サーバに対して計算負荷となります。あるイベントによるサーバへの正確な負荷は、監視されるイベントの頻度、イベントに関係したデータ、およびそのイベントを監視しているクライアントアプリケーションの数によって決まります。

［最大動作監視負荷］は、イベント監視拡張がサーバにかけることができる負荷の大きさを示す相対値です。0 を指定すると、無制限になります。この属性の適切な値を見つけるには、実際にいろいろな値を試してみてください。

6 ［保存］>［OK］の順にクリックします。

LDAP サーバの情報を取得する

LDAP サーバについての情報を取得するには、ICE か LDAP 検索を使用します。これらのユーティリティは rootDSE(ディレクトリサービスエージェント、固有エントリ) から情報を要求します。

rootDSE は、ディレクトリツリーの擬似オブジェクトです。このオブジェクトは、ツリーのルートにある名前のないエントリです。rootDSE は接続しているサーバに固有の情報を持っています。たとえば、rootDSE はスキーマと、スキーマがサポートする拡張およびコントロールの場所の情報を持っています。

rootDSE はツリー内の名前のないエントリであるため、通常、LDAP サーバは検索操作で rootDSE をクライアントに返しません。

次の表は、rootDSE から得られる情報を表示したものです。

情報と説明	引用
スキーマの場所 : LDAP サーバまたはツリーのスキーマの場所は、subschemaSubentry を読み込むことによって検索できます。eDirectory では、cn=schema が検索のベースになります。	subschemaSubentry: cn=schema
サポートされている拡張 : 拡張により、コンテキストの作成、マージ、新しいレプリカの追加、LDAP サーバのリフレッシュ、レプリカの削除、レプリカタイプのマスタから読み込み / 書き込みまたは読み込み専用への変更などのサーバの管理、および識別情報の管理ができます。 拡張の形式は ASN.1OID です。拡張の詳細については、「LDAP Extensions (https://www.novell.com/documentation/developer/ldapover/ldap_enu/data/a6ik7oi.html)」を参照してください。	supportedExtension: 2.16.840.1.113719.1.27.100.12 supportedExtension: 2.16.840.1.113719.1.27.100.7 supportedExtension: 2.16.840.1.113719.1.27.100.8
LDAP サーバを提供しているベンダ	vendorName: OpenText Corporation
LDAP サーバがサポートしているディレクトリバージョン	vendorVersion: eDirectory v8.7.0 (10410.29)
OpenText eDirectory が実行しているバージョン	vendorVersion: eDirectory v8.7.0 (10410.29)
ディレクトリサーバ名とディレクトリツリー名	dsaName: cn=WestWindNDS,o=westwind directoryTreeName: t=WESTWINDTREE
サポートされている SASL メカニズム	supported SASLMechanisms: EXTERNAL supported SASLMechanisms: DIGEST-MD5 supported SASLMechanisms: NMAS LOGIN
サポートされている LDAP サーバのバージョン。	supportedLDAPVersion: 2 supportedLDAPVersion: 3
サーバ統計情報 : rootDSE は LDAP サーバに関するさまざまな統計情報を提供します (強力な認証バインド数など)。	errors: 0 securityErrors: 0 chainings: 3 referralsReturned: 6 extendedOps: 0 abandonOps: 0 wholeSubtreeSearchOps: 1

rootDSE の情報は、アプリケーション開発に活用することができます。

シナリオ : アプリケーションの開発 — あるユーザが新しいレプリカを作成するアプリケーションを作成しています。rootDSE を読み込むと、リストに supportedExtension: 2.16.840.1.113719.1.27.100.7 と記述されています。これにより、サーバが新しいレプリカを作成するコールをサポートすることがわかります。

また、OpenText Identity Console は rootDSE で利用できる機能をチェックし、その情報に従って動作します。

rootDSE を検索するには、ワークステーションで次を入力します。

構文 :

```
ldapsearch -H ldap://hostname:port -b "" -s base "objectclass=*"
```

例 :

```
ldapsearch -H ldap://10.10.10.10:389 -b "" -s base "objectclass=*"
```

この検索は、ldap_search API を使用したどのアプリケーションでも実行することができます。

検索の鍵はスコープがベース (-s base) であることです。また、ベースは null であり、フィルタが objectclass = * に設定されていることにも注意してください。このクライアントの場合、ベースは -b です。

rootDSE の読み取り方法の詳細については、次のいずれかを参照してください。

- LDAP Libraries for C (https://www.novell.com/documentation/developer/cldap/ldaplibc/data/a2etgcm.html)
- LDAP Classes for Java (https://www.novell.com/documentation/developer/jldap/jldapenu/data/bktitle.html)

LDAP 検索フィルタの詳細については、LDAP Search Filters (https://www.novell.com/documentation/developer/ldapover/ldap_enu/data/a3saoeg.html) を参照してください。このセクションは、NDK マニュアルの「LDAP and NDS Integration」にあります。

汎用タイムサポートの設定

汎用タイムサポートのオプションによって、時間を YYYYMMDDHHmmSS.0Z 形式で表示できます。ldapconfig ユーティリティまたは OpenText Identity Console の LDAP を使用して、LDAP 汎用タイムサポートを有効または無効にすることができます。

汎用タイムサポートは、次のいずれかの方法を使用して有効にすることができます。

OpenText Identity Console での LDAP 設定

1. OpenText Identity Console のホームページで、[**LDAP 環境設定**] タイルをクリックします。
2. [**LDAP サーバ**] をクリックします。
3. [**その他**] ドロップダウンメニュー > [**LdapGeneralizedTime**] の順にクリックし、[**保存**] > [**OK**] の順にクリックします。

ldapconfig ユーティリティ

```
ldapconfig set "ldapGeneralizedTime=yes/no" -a <admin-FDN> -w <admin-password">
```

許容変更の設定

ldapPermissiveModify オプションを TRUE に設定することで、現在の LDAP 変更操作を拡張できます。存在しない属性を削除したり、既存の属性に属性値を追加しようとすると、エラーメッセージを表示しないで操作が完了します。

OpenText Identity Console での LDAP 設定

1. OpenText Identity Console で、[**LDAP 環境設定**] タイルをクリックします。
2. [**LDAP サーバ**] をクリックします。
3. [**その他**] ドロップダウンメニューをクリックして、[**LdapPermissiveModify**] チェックボックスをオンにし、[**保存**] > [**OK**] の順にクリックします。

ldapconfig ユーティリティ

```
ldapconfig set "ldapPermissiveModify=yes/no" -a <admin-FDN> -w <admin-
password">
```

プロキシ承認コントロール

OpenText eDirectory には、RFC 4370 で規定されているように、LDAP プロトコルを使用してプロキシ承認を制御する柔軟性があります。プロキシ承認コントロールにより、クライアントは、接続に関連付けられている現在の承認 ID の代わりに、指定された承認 ID を使用して操作を処理することを要求できます。この機能は、各操作の承認 ID を指定するためのメカニズムを提供し、複数のユーザの代わりにいくつかの操作を実行する必要があるクライアントで役に立ちます。

OpenText eDirectory サーバで認証するには、管理者がクライアント要求でプロキシ承認コントロール OID、2.16.840.1.113730.3.4.18 を指定する必要があります。プロキシ承認コントロールを使用するには、認証済みユーザが偽装ユーザに対してスーパバイザ権を持っている必要があります。

1. OpenText eDirectory ツリーを作成し、そこにユーザオブジェクトを追加します。
2. OpenText Identity Console のホームページで、[**権利の管理**] タイルをクリックします。
3. [**トラスティ**] をクリックします。
4. [**オブジェクトの検索** ] をクリックして、リストからユーザを選択します。
5. [**コンテキストブラウザ**] 画面でユーザを選択し、[**OK**] をクリックします。
6. [**割り当てられた権利**] をクリックして、選択したユーザに割り当てられている権利を選択します。例 : 読み込み、書き込み、またはネスト。
7. [**完了**] をクリックし、[**適用**] をクリックします。

ldapsearch のプロキシ承認を実行するには、次のコマンドを使用します。

```
ldapsearch -x -H ldap://[ldap host ip]:[ldap-port] -D <Admin DN> -w
<Password> -e '!authzid=dn:<Impersonate user> -b o=novell -s one
```

プロキシ承認コントロールを使用して他の LDAP 操作を実行するには、LDAP 要求に 2.16.840.1.113730.3.4.18 OID を指定します。

プロキシ承認操作の監査

プロキシ承認操作を監査するために、OpenText eDirectory は、DSE_IMPERSONATE という新しいイベントを提供します。

LDAP ページ検索コントロール

LDAP ページ検索コントロールは LDAP 検索操作と一緒に使用され、クライアントはページサイズによって制限された方法で検索結果を受信できます。ページのサイズは、使用可能なリソースとインフラストラクチャに基づいて管理者が決定できます。LDAP ページ検索コントロールの詳細については、『RFC 2696』を参照してください。

ページ検索コントロールを使用するには、管理者が検索要求でページ検索コントロール OID、1.2.840.113556.1.4.319 を指定する必要があります。ページ検索コントロールを使用するには、管理者は、すべてのパーティションが同じサーバに存在することを確認する必要があります。

すべてのパーティションが同じサーバに存在しない場合、管理者がページコントロールと重大性レベルを設定して LDAP 検索を実行すると、次のエラーメッセージがクライアント側に表示されます。

```
Server is unwilling to perform (53)
```

管理者が上記のエラーメッセージを受け取りたくない場合は、NDSD_NLDAP_IGNORE_CRITICALITY 環境変数を true に設定します。

注 : 9.2 SP2 より前のバージョンの OpenText eDirectory では、ページ検索コントロールで重大性レベルが設定されている場合でも、クライアント側でこのようなエラーメッセージを表示できませんでした。

---

ldapsearch に対してページ検索コントロールを実行するには、次のコマンドを実行します。

```
ldapsearch -x -H ldap://[ldap host ip]:[ldap-port] -D <Admin DN> -w
<Password> -E \!pr=<page size>/<prompt|no prompt> -b o=novell -s one
```

LDAP 拡張 DN コントロール

OpenText eDirectory には LDAP 拡張 DN コントロールが備わっています。このコントロールは、識別名オブジェクトの拡張フォームを要求するために、拡張 LDAP 検索で使用します。この拡張フォームには、オブジェクト GUID の文字列表現と、オブジェクトの識別名が含まれます。

eDirectory サーバで LDAP 拡張 DN コントロール機能を使用するには、管理者は拡張 LDAP 検索要求で LDAP 拡張 DN コントロールの OID1.2.840.113556.1.4.529 を指定する必要があります。

拡張 DN コントロールによって、クライアントは、このコントロールを使用する LDAP 検索によって返される結果に、オブジェクトの GUID データと、distinguishedName オブジェクトが含まれるように要求できます。次のように返されます。

```
<GUID=xxxxxxxx>;distinguishedName
```

xxxxxxxx は GUID が含まれる文字列、distinguishedName は DN(例 : cn=users,dc=fabrikam,dc=com) です。

LDAP 拡張 DN コントロールは、整数のフラグ値も渡すことができます。渡されるフラグ値によって、返される GUID 値の文字列形式が決まり、次の BER エンコードシーケンスに設定されます。

```
Sequence {
  Flag    INTEGER
}
```

フラグ値 0 の場合、GUID 値は 16 進数文字列形式で返されます。例 :
<GUID=3BC72D2DEC5A704BBDC21F4EF97B7870>

フラグ値 1 の場合、標準文字列形式で GUID 値が返されます。例 : <GUID=098f2470-bae0-11cd-b579-08002b30bfeb>

一部として DN が含まれる OpenText eDirectory のデータタイプには複雑なものもいくつかあります。OpenText eDirectory が LDAP 拡張 DN コントロールで処理する複雑なデータタイプは以下のみです。

- SYN_PATH (volumeDN の GUID が返されます)
- SYN_DN
- SYN_TYPED_NAME

注 : 前述の複雑なデータタイプが含まれる拡張 DN コントロールを使用する場合、LDAP 検索のパフォーマンスが影響を受けます。

例 :

次の C++ コード例は、シーケンスデータを手動でフォーマットする方法を示しています。ber_printf 関数を使用してシーケンスデータを作成します。フラグ部分には、GUID 文字列の書式指定子が含まれます。

```
LDAPControl *FormatExtDNFlags(int iFlagValue)
{
  BerElement *pber = NULL;
  LDAPControl *pLControl = NULL;
  berval *pldctrl_value = NULL;
  int success = -1;

  // Ensure that iFlagValue is either 0 or 1. Convert TRUE (-1) to a legal
value.
  if(iFlagValue != 0)
         iFlagValue = 1;

  // Format and encode the SEQUENCE data in a BerElement.
  pber = ber_alloc_t(LBER_USE_DER);
  if(pber==NULL) return NULL;
  pLControl = new LDAPControl;
  if(pLControl==NULL) { ber_free(pber,1); return NULL; }
  ber_printf(pber,"{i}",iFlagValue);

  // Transfer encoded data into a BERVAL.
  success = ber_flatten(pber,&pldctrl_value);
  ber_free(pber,1);
  if(success != 0) {return NULL;}

  // Copy the BERVAL data to the LDAPControl structure.
  pLControl->ldctl_oid = LDAP_SERVER_EXTENDED_DN_OID;
  pLControl->ldctl_iscritical = true;
  pLControl->ldctl_value.bv_val = new char[pldctrl_value->bv_len];
  memcpy(pLControl->ldctl_value.bv_val,
         pldctrl_value->bv_val, pldctrl_value->bv_len);
  pLControl->ldctl_value.bv_len = pldctrl_value->bv_len;

  // Cleanup temporary berval.
  ber_bvfree(pldctrl_value);

  // Return formatted LDAPControl data.
  return pLControl;
}
```

次の C++ コード例は、拡張 DN コントロールと ldap_search_ext_s 関数を併用する方法について示しています。

```
        int err;
        LDAP *ldapConnection = NULL;
        LDAPControl *pExtDNControl;
        LDAPControl *controlArray[2];
        LDAPMessage *results = NULL;
        LDAPMessage *message = NULL;
        char *dn = NULL;

        // Connect to the default LDAP server.
        ldapConnection = ldap_open( NULL, 0 );
        if ( ldapConnection == NULL ) go to FatalExit0;

        // Bind to the server using default credentials.
        err = ldap_simple_bind_s( ldapConnection, NULL, NULL);
        if (LDAP_SUCCESS != err) go to FatalExit0;

        // Setup the extended DN control, requesting 'standard string'
format.
        pExtDNControl = FormatExtDNFlags(1);
        if (pExtDNControl == NULL) go to FatalExit0;
        controlArray[0] = pExtDNControl;
        controlArray[1] = NULL;

        // Perform a synchronous search.
        err   = ldap_search_ext_s( ldapConnection,
                        "cn=users,dc=Fabrikam,dc=com",
                        LDAP_SCOPE_SUBTREE,
                        "objectClass=*",
                        NULL,          // Retrieve all attributes.
                        0,             // Retrieve attributes and values.
                        (LDAPControl **) &controlArray,
                        NULL,          // Client controls.
                        0,             // Timeout.
                        0,             // Sizelimit.
                        &results       // Receives identifier for results.
                        );
        if (LDAP_SUCCESS != err) go to FatalExit0;

        // Process the search results.
        message = ldap_first_entry( ldapConnection, results );
        while (message != NULL)
        {
```

```
                // Print the distinguished name of the object.
                dn = ldap_get_dn( ldapConnection, message );
                if (!dn) go to FatalExit0;
                printf( "  Distinguished Name is : %s\n", dn );
                ldap_memfree(dn);
                message = ldap_next_entry( ldapConnection, message );
        }

FatalExit0:
        if (ldapConnection)
                ldap_unbind( ldapConnection );
        if (results)
                ldap_msgfree( results );
}
```

LDAP イベントの監査

LDAP 監査は、アプリケーションが追加、変更、検索などの LDAP の処理を監視 / 監査し、接続情報や LDAP 処理が発生する際にサーバが接続するクライアント IP、メッセージ ID、処理に対する結果コードなどの有用な情報を LDAP サーバからフェッチします。

LDAP イベントの監査の詳細については、LDAP Event Services (https://www.novell.com/documentation/developer/ldapover/ldap_enu/data/ag7bleo.html) を参照してください。

15 OpenText eDirectory のバックアップと復元

OpenText eDirectory は、レプリケーションにより耐障害性が保たれるようになっています。OpenText eDirectory ツリーに属するサーバに障害があっても、別のサーバがサービスを続行できます。この意味でレプリカ作成機能は、最も重要な保護機能と位置づけられています。

ただし 1 台だけのサーバで運用している環境では、レプリカ作成は不可能です。また、レプリカを作成していても、完全に復元するのは困難な場合もあります。ハードウェア障害など、機器が破損したときや、火事や水害で何台ものサーバが動かなくなってしまった場合などです。各サーバの OpenText eDirectory をバックアップしておけば、ネットワークの耐障害性を高めることになります。

eDirectory バックアップツールでは、サーバ単位で OpenText eDirectory データベースをバックアップすることができます。これには次のような利点があります。

- **どのプラットフォームでも同じツールで操作可能。**
- **稼動中でもバックアップ可能。**OpenText eDirectory データベースを停止することなく、そのままで完全なバックアップを取ることできます。
- **個々のサーバ単位に、迅速な復元処理が可能。**ハードウェア障害からの復元には特に有用です。
- **高い拡張性。**数千万から数億単位のオブジェクトを保持した OpenText eDirectory データベースを持つサーバでもバックアップ可能です。バックアップの処理速度は主として I/O チャネルの帯域幅で決まります。
- **レプリカ作成機能と DSMASTER サーバを組み合わせて運用していれば、ツリー全体の復元も容易。**DSMASTER サーバを設定していない場合でも、かなりの程度まで復元できます。詳細については、473 ページの 「DSMASTER サーバによる災害対策」を参照してください。
- **関連ファイルもバックアップ可能。**サーバにある、データベース以外の関連ファイルもバックアップできます。NICI セキュリティファイル、ストリームファイル、インクルードファイルで指定したファイル (autoexec.ncf など) が対象になります。
- **サーバの停止直前の状態に OpenText eDirectory を復元可能。**ただしロールフォワードログを継続的に保存していることが条件です。詳細については、475 ページの 「ロールフォワードログを使用する」を参照してください。
- **ハードウェアのアップグレードを単純化。**サーバの識別情報を新しいマシンに移行する簡単な方法は、OpenText eDirectory データベースのコールドバックアップを取り、それを復元するやり方です。RAM のアップグレードなど、変更を加える際の安全措置としても有効です。「596 ページの 「ハードウェアのアップグレードやサーバの交換」」を参照してください。

- **OpenText eDirectory の分散環境での運用を考慮。**ロールフォワードログがあれば、ツリー内の他のサーバと完全に同期した状態にまで復元できます。
- **無人でのバックアップが可能。**バッチファイルを作成し、DSBK Client を使用して無人でバックアップを実行できます。

OpenText eDirectory バックアップツールには、個々のサーバ単位でデータベースおよび関連ファイルのバックアップを取り、復元するために必要な機能がすべて揃っています。個々のオブジェクトやツリーの一部を単位としてバックアップ / 復元することはできません。

システムバックアップ機能と組み合わせれば、eDirectory バックアップファイルをテープに保存して安全を期すことができます。

OES 2 Linux の場合、OpenText Storage Management Services を使用して OpenText eDirectory をバックアップすることができます。SMS には、OpenText eDirectory のバックアップと復元のためのターゲットサービスエージェント (TSA) が用意されています。TSANDS サービスは、ディレクトリツリーの SMS API の実装を提供します。アプリケーションはこの機能を使用して OpenText eDirectory オブジェクトのバックアップと復元を行います。

TSANDS は、バックアップアプリケーションを最大限活用するために、次の機能をサポートします。

- OpenText eDirectory オブジェクトに適用できるフィルタ。
- バックアップデータから OpenText eDirectory オブジェクトを選択的に復元。
- 特定のリソースセットをリネームする機能。
- OpenText eDirectory 変更日に基づく、インクリメンタルバックアップと差分バックアップのサポート。
- SIDF 準拠のソフトウェアによるデータ解釈を可能とする SIDF フォーマットのデータ。

TSANDS の使用法に関する詳細については、TSANDS マニュアルページを参照してください。

この章では、次のトピックについて説明します。

- eDirectory のバックアップ処理に関する確認事項 (461 ページ)
- バックアップサービスおよび復元サービスについて (463 ページ)
- ロールフォワードログを使用する (475 ページ)
- 復元処理の準備 (480 ページ)
- DSBK の使用 (483 ページ)
- NICI のバックアップと復元 (501 ページ)
- 復元後の検証処理に失敗した場合の対処方法 (503 ページ)
- バックアップ / 復元の運用例 (507 ページ)
- DSBK を使用した障害復旧計画 (514 ページ)
- LDAP ベースのバックアップ (517 ページ)

- SMS による eDirectory バックアップ (518 ページ)
- ndsbackup ユーティリティ (852 ページ)

OpenText eDirectory のバックアップ処理に関する確認事項

複数サーバ構成のツリーで、サーバが停止していてもオブジェクトにアクセスできるようにする

❒ 複数サーバ構成のツリーでは、耐障害性を得るため、すべての OpenText eDirectory パーティションについて、複数台のサーバにレプリカが作成されていることを確認します。

レプリカの作成については、「164 ページの 「レプリカを追加する」」を参照してください。

個々のサーバについて、ハードウェア障害などの場合に、迅速に完全復元できるようにするための準備

❒ 定期的に(週1回など)OpenText eDirectoryデータベースのフルバックアップを取ってください。

❒ 定期的に (毎晩など) インクリメンタルバックアップを取ってください。

❒ EBA対応サーバのフルバックアップを取得する間はNICIおよびストリーム属性を必ず選択してください。そうでない場合、バックアップ操作が失敗します。

❒ OpenText eDirectory データベースのフル / インクリメンタルバックアップ終了後、すぐにファイルシステムをテープにフル / インクリメンタルバックアップしてください。

バックアップツールは、サーバ上の指定したディレクトリにバックアップファイルを作成しますが、これを直接テープに保存する機能はありません。したがって、OpenText eDirectory のバックアップ処理後すぐにファイルシステムのバックアップを行い、安全な記録媒体であるテープに保存する必要があります。

❒ 必要に応じて、ロールフォワードログ記録を残すよう設定してください。

レプリカリングに属するサーバは、ロールフォワードログ機能を有効にしておく必要があります。バックアップファイルがあっても、ロールフォワードログがなければ復元後の検証処理に失敗し、データベースを開けないことになります。他のサーバとレプリカを共有するデータベースは、停止直前の状態にまで復元しない限りデフォルトではオープンされません。

単一サーバ環境では、ロールフォワードログがなくても復元後の検証に失敗することはありませんが、最後にバックアップを取った時の状態にしか戻りません。ロールフォワードログがあれば、システム停止直前の状態まで OpenText eDirectory を復元できます。

ロールフォワードログ機能を使う際の主な注意点は次のとおりです。詳細については、475 ページの 「ロールフォワードログを使用する」を参照してください。

- ロールフォワードログの新しい場所を指定してください (デフォルトの使用はお勧めしません)。

ログはサーバ上のローカルファイルとして保存する必要があります。障害対策上、OpenText eDirectory と同じディスクパーティション / ボリューム、同じ記憶デバイスは避けてください。ロールフォワードログ専用の、独立したパーティション / ボリュームを用意するとよいでしょう。

- ロールフォワードログの保存先を文書に記録しておき、障害時にはすぐにわかるようにしてください。

  サーバが正常に機能するときにこの場所を見つけるには、477 ページの 「ロールフォワードログの保存先」を参照してください。ただしハードウェア障害など OpenText eDirectory に影響する障害がサーバで発生すると、ロールフォワードログの場所を調べることができなくなります。

- ロールフォワードログを保存しているディスクパーティション/ボリュームの空き容量を監視し、容量不足にならないようにしてください。

  容量不足のためロールフォワードログが作成できなくなると、OpenText eDirectory はそのサーバに対して応答しなくなります。

- ロールフォワードログの保存先にアクセスできるユーザを制限し、権利のないユーザがログを参照できないようにしてください。

- 復元が必要となったときは、復元処理の終了後、そのサーバのロールフォワードログ設定をやり直してください。復元処理の過程で、設定が初期状態に戻るためです。ロールフォワードログを有効にしてから、改めてフルバックアップも取る必要があります。

❒ NICI を使用する場合、eDirectory が DIB を開き、暗号化されたデータを読み込むのに同じ NICI ファイルが必要となるため、OpenText eDirectory のバックアップに NICI セキュリティファイルが含まれていることを確認します。

詳細については、『NICI Administration Guide』の「Backing Up and Restoring NICI」を参照してください。

❒ 複数サーバ構成のツリーの場合、バックアップ処理にバックアップツールを使うためには、レプリカを共有するサーバすべてを eDirectory 8.5 以降にアップグレードする必要があります。

8.5 より前の eDirectory とは、復元後の検証処理の互換性がないためです。復元検証処理の詳細については、466 ページの 「バックアップツールによる復元作業の概要」を参照してください。

❒ 定期的にバックアップログを調べて、無人でのバックアップが正常に実行されていることを確認してください。

❒ ロールフォワードログディレクトリから古いログファイルを削除します。

❒ サーバをアップグレードする際は、「596 ページの 「ハードウェアのアップグレードやサーバの交換」」を参照してコールドバックアップを取ってください。

❒ 複数サーバ構成のツリーでは、耐障害性を得るため、すべての OpenText eDirectory パーティションについて、複数台のサーバにレプリカが作成されていることを確認します。

パーティションのレプリカを作成しておけば、保守作業などのためサーバを停止している間もオブジェクトにアクセスできるばかりでなく、ハードウェア障害などでサーバが使えなくなった場合に備える障害対策としても役立ちます。逆に複数サーバ構成のツリーで複製されていないパーティションを保持するサーバがあると、障害時にそのパーティションを復元できない恐れがあります。一番良いのは、すべてのパーティションが複製されることを確認することです。複数サーバ構成のツリー内にレプリカ

を作成していないパーティションがある場合の問題点の詳細については、「バックアップツールによる復元作業の概要 , ロールフォワードログを使用する」および「復元後の検証処理に失敗した場合の対処方法」を参照してください。

レプリカの作成については、「62 ページの 「レプリカ」」および「159 ページの第 6 章 「パーティションおよびレプリカの管理」」を参照してください。

- ❒ OpenText eDirectory およびファイルシステムのバックアップを収めたバックアップテープは、安全な場所に保管するようにしてください。
- ❒ バックアップ計画が適切であるか、定期的に検証するようにしてください。
- ❒ (オプション) コールドバックアップ (データベースをクローズしてフルバックアップを取る作業) や高度なバックアップ / 復元操作をリモート操作で実行する場合は、リモート側マシンに DSBK をインストールしてください。また、VPN を使用するなど、ファイアウォール越しにアクセスできるように設定する必要があります。

  DSBK は、サーバに OpenText eDirectory をインストールする際、同時にインストールされます。Sun JVM 1.3.1 が動作するワークステーション上でも使えます。DSBK のインストールや設定の手順については、483 ページの 「DSBK の使用」を参照してください。
- ❒ (オプション) コールドバックアップ (データベースをクローズしてフルバックアップを取る作業) や高度なバックアップ / 復元操作をリモート操作で実行する場合は、リモート側マシンに eMBox をインストールしてください。また、VPN を使用するなど、ファイアウォール越しにアクセスできるように設定する必要があります。

  eMBox は、サーバに OpenText eDirectory をインストールする際、同時にインストールされます。Sun JVM 1.3.1 が動作するワークステーション上でも使えます。eMBox のインストールや設定の手順については、629 ページの 「eMBox クライアントを使ったバックアップ / 復元作業」を参照してください。

災害により何台ものサーバが被害を受けた場合に備える準備

- ❒ 上述の対策はすべて実施してください。
- ❒ 複数サーバ構成のツリーであれば、障害対策のために DSMASTER サーバを用意するようお勧めします。

  詳細については、473 ページの 「DSMASTER サーバによる災害対策」を参照してください。
- ❒ 災害からの復旧計画が適切であるか、定期的に検証するようにしてください。

バックアップサービスおよび復元サービスについて

- 464 ページの 「OpenText eDirectory バックアップツールについて」
- 465 ページの 「DSBK のバックアップおよび復元と TSA for NDS バックアップの違い」
- 466 ページの 「バックアップツールによる復元作業の概要」
- 468 ページの 「バックアップファイルのヘッダ書式」
- 472 ページの 「バックアップログファイルの書式」

- 473 ページの 「DSMASTER サーバによる災害対策」
- 474 ページの 「遷移ベクトルと復元後の検証処理」

OpenText eDirectory バックアップツールについて

バックアップツールには、個々のサーバ単位で、稼動したままの状態で継続的に eDirectory データベースのバックアップを取る機能があります。データベースを停止することなく、処理を始めた時点の OpenText eDirectory のバックアップをサーバで取ることができます。つまり、バックアップ作業はいつでも可能であり、その間も OpenText eDirectory を使い続けることができます。

注 : 特に指示しなければこの「ホット」バックアップになりますが、必要であればデータベースを停止して「コールド」バックアップを取ることもできます。

また、ロールフォワードログを有効にすれば、最後にバックアップを取った時点以降のトランザクションをすべて記録しておけます。これを使えば、サーバが停止する直前の状態に復元することもできます。さらに、レプリカリングに属するサーバすべてについてロールフォワードログを取る必要があります。これにより、他のサーバとの同期状態も復元できます。バックアップファイルがあっても、ロールフォワードログがなければ復元後の検証処理に失敗し、データベースを開けないことになります。ロールフォワードログの機能は、デフォルトでは無効になっています。詳細については、475 ページの 「ロールフォワードログを使用する」を参照してください。

バックアップツールは OpenText eDirectory のオブジェクトをすべて一度にバックアップできるわけではありません。各サーバのパーティションをバックアップするだけです。この方式は、特定のサーバのみを復元する場合に優れているばかりでなく、TSA for NDS® バックアップを使う旧式の方法に比べて高速です。ただし、eDirectory 8.6 のマニュアルにも記載されているように、TSA for NDS は今でも使用できます。必要に応じて TSA for NDS と新しいバックアップを使い分けられます。動作の違いについては、「465 ページの 「DSBK のバックアップおよび復元と TSA for NDS バックアップの違い」」を参照してください。

OpenText eDirectory バックアップツールは、ファイルシステムバックアップ機能と組み合わせて使用して、OpenText eDirectory バックアップファイルをテープに安全に保存する必要があります。OpenText はバックアップ用製品を開発している主要企業と提携しています。詳細については、OpenText eDirectory Partner Products (http://www.novell.com/partnerguide/section/466.html) を参照してください。

バックアップツールで作成されるバックアップファイルやログファイルの形式については、「472 ページの 「バックアップログファイルの書式」」および「468 ページの 「バックアップファイルのヘッダ書式」」を参照してください。

DSBK のバックアップおよび復元と TSA for NDS バックアップの違い

OpenText eDirectory の以前のバージョンでは、バックアップ / 復元ツールは、ツリーをオブジェクト単位でバックアップする方式を使用していました。

eDirectory 8.7 で組み込まれたバックアップツールは、方式やアーキテクチャが一新されています。ツリーではなくサーバを基準とし、個々のサーバの OpenText eDirectory データベースをバックアップします。この変更により、以前の TSA for NDS バックアップよりも大幅に速くなりました。

TSA for NDS は今でも使用できますが、今後は新しいバックアップツールを使うようお勧めします。

新旧バックアップツールの違いを次の表に示します。

項目	TSA for NDS による以前のバックアップ	バックアップツールによる「ホットバックアップ」
バックアップの対象	ツリー全体をオブジェクト単位でバックアップ。	サーバ単位で OpenText eDirectory データベースをバックアップ。 ツリー全体の耐障害性は主としてレプリカ作成機能で確保していますが、サーバ単位のバックアップ機能によりさらに強固になります。 災害などで何台ものサーバが停止した場合でもツリーを復元できるようにするためには、DSMASTER サーバを導入し、それに応じたレプリカ作成方針を立てる必要があります。「473 ページの「DSMASTER サーバによる災害対策」」を参照してください。
Speed	N/A	大幅に改善されています。新しいバックアップの機能では処理性能を最重視しました。
バックアップファイルの保存先	直接テープに書き出し可。	ファイルシステム上のバックアップファイルとして書き出し。 別途ファイルシステムのバックアップツールを使って、テープに書き出す必要があります。
プラットフォーム間の違い	プラットフォームによって使い方が別々。	どのプラットフォームでも同じ使い方。

項目	TSA for NDS による以前のバックアップ	バックアップツールによる「ホットバックアップ」
個々のサーバ単位での復元	考慮されていません。	ハードディスクが障害した後にサーバ単位で復元することができます。サーバを別のマシンに移行する際にもバックアップを利用できます。 ロールフォワードログを使えば、停止直前の状態に復元することも可能です。これにより、レプリカリングに属する他のサーバと同期状態を揃えることができます。 個々のサーバの OpenText eDirectory 関連ファイルもバックアップの対象とすることができます。たとえば NICI ファイルのバックアップと復元ができます。独自の関連ファイルリストを作成して、対象ファイルをバックアップに追加することもできます。
NICI ファイルのバックアップ / 復元	考慮されていません。	NICI ファイルのバックアップ / 復元も可能です。これにより、復元後、暗号化データにアクセスできます。復元作業の時間を大幅に短縮できるでしょう。
個々のサーバのロールフォワードログ	考慮されていません。	最後にバックアップを取った時点以降のトランザクションを、ロールフォワードログとしてすべて記録しておけます。これを使うと、停止直前の状態にまで復元することができます。複数サーバ環境では、これを使用して他のサーバと同期状態を揃えることができます。ロールフォワードログの機能は、デフォルトでは無効になっています。詳細については、475 ページの 「ロールフォワードログを使用する」を参照してください。

バックアップツールによる復元作業の概要

復元作業に先立ち、バックアップファイルをすべて揃えておく必要があります。その手順については「480 ページの 「復元処理の準備」」を参照してください。DSBK からバックアップツールの復元機能を呼び出すと、バックアップツールで次のような処理が行われます。

1. DS エージェントをクローズします。
2. アクティブな DIB(Data Information Base) セットを、NDS から RST に切り替えます。

注 : 既存の NDS データベースはそのままサーバに残り、復元後の検証に失敗した場合は、再びこれがアクティブな DIB セットになります。

3. 復元処理が始まります。新たに RST という DIB セットを作り、そこに復元します。
4. DIB セットをいったん無効にします。

   擬似サーバのログイン無効属性をオンにします。これは、この DIB セットを使って DS エージェントがオープンされるのを避けるための措置です。
5. ロールフォワードログに関する設定をデフォルトに戻します。これを防ぐには、-s スイッチを使用します。

   したがって、復元後はロールフォワードログを書き出さない設定になります。書き出し先ファイル名の設定もデフォルトに戻ります。

注 : このサーバでロールフォワードログ記録を使うためには、復元後に改めて有効に切り替え、障害対策のための書き出し先も設定し直して、ロールフォワードログ記録の環境設定を再作成する必要があります。ロールフォワードログを有効にしてから、改めてフルバックアップも取る必要があります。

---

6. 復元された RST データベースの検証処理を行います。

   復元されたデータの整合性を確認します。この確認は、レプリカを共有しているすべてのサーバにアクセスし、遷移ベクトルを比較して実行されます。

   検証結果はログファイルに出力されます。

   リモートサーバの遷移ベクトルの方がローカルベクトルより後の時間に作成されたものである場合は、復元されなかったデータがあるということなので、検証処理は失敗します。

   いずれかのレプリカが検証に失敗した場合、ログファイルに記録される情報の例を示します。比較された遷移ベクトルがわかります。

```
Server: \T=LONE_RANGER\O=novell\CN=CHIP
   Replica: \T=LONE_RANGER\O=novell
      Status: ERROR = -6034
         Local TV             Remote TV
         s3D35F377 r02 e002   s3D35F3C4 r02 e002
         s3D35F370 r01 e001   s3D35F370 r01 e001
         s3D35F363 r03 e001   s3D35F363 r03 e001
         s3D35F31E r04 e004   s3D35F372 r04 e002
         s3D35F2EE r05 e001   s3D35F2EE r05 e001
         s3D35F365 r06 e003   s3D35F365 r06 e003
```

   詳細については、474 ページの 「遷移ベクトルと復元後の検証処理」を参照してください。
7. 検証に成功した場合は、RST を NDS と改名し、ログイン無効属性をオフにします。したがってこれがサーバでアクティブな OpenText eDirectory データベースになります。失敗した場合は改名しないので、元の NDS が再びアクティブな DIB セットになります。

   検証に失敗した場合の復旧方法については、「503 ページの 「復元後の検証処理に失敗した場合の対処方法」」を参照してください。

注 :「高度な復元オプション」を使えば、強制的に RST データベースをアクティブにし、ロックを解除することもできますが、OpenText サポート部門の指示がない場合はお勧めできません。

バックアップファイルのヘッダ書式

バックアップファイルのヘッダには、次のような重要な情報が記録されています。

- バックアップファイルが作成された時点のファイル名。

  バックアップ作成後にファイル名を変更した場合にはこの情報が有用になります。
- バックアップ時点のロールフォワードログ名。

  1 回分のフルバックアップと 3 回分のインクリメンタルバックアップがあって、最後のインクリメンタルバックアップから復元しているような場合に役立ちます。完全に復元するために必要な最初のロールフォワードログを知ることができます。
- このサーバが保持しているレプリカのリスト。

  各レプリカをどのサーバに配置しているか記録していない場合に役立ちます。災害で何台ものサーバが停止した場合、バックアップファイルのヘッダに表示されたこの情報を見れば、最初にどのサーバを復元するべきか判断できます。
- 同時にバックアップするようユーザのインクルードファイルに指定された関連ファイル名。
- 複数のファイルに分割してバックアップする場合のファイル数。

各バックアップファイルのヘッダは XML 形式で記述されます。ヘッダ部に続き、データベース内のデータを、バイナリ形式で記録します

注 : ファイルの末尾にバイナリデータがあると構文解析でエラーが発生しますが、XML ヘッダは XML 標準に適合しています。

バックアップが複数のファイルに分かれる場合、各ファイルに同じヘッダ情報を記録します。

警告 : バックアップファイルを開くときは、ヘッダを確認するだけにしてください。ファイルを保存または変更しようとすると、ファイルの一部が切り捨てられる場合があります。ほとんどのアプリケーションがバイナリデータを正しく保存することができません。

XML ヘッダの DTD を次に示しますこの DTD は参照のため、バックアップファイルのヘッダの一部として書き込まれます。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE backup [
<!ELEMENT backup (file|replica)*>
<!ELEMENT file (#PCDATA)>
<!ELEMENT replica EMPTY>
<!ATTLIST backup version CDATA #REQUIRED
    backup_type (full|incremental) #REQUIRED
    idtag CDATA #REQUIRED
    time CDATA #REQUIRED
    srvname CDATA #REQUIRED
    dsversion CDATA #REQUIRED
    compression CDATA "none"
    os CDATA #REQUIRED
    current_log CDATA #REQUIRED
    number_of_files CDATA #IMPLIED
```

```
    backup_file CDATA #REQUIRED
    incremental_file_ID CDATA #IMPLIED
    next_inc_file_ID CDATA #IMPLIED>
<!ATTLIST file size CDATA #REQUIRED
    name CDATA #REQUIRED
    encoding CDATA "base64"
    type (user|nici) #REQUIRED>
<!ATTLIST replica partition_DN CDATA #REQUIRED
    modification_time CDATA #REQUIRED
    replica_type (MASTER|SECONDARY|READONLY|SUBREF|
    SPARSE_WRITE|SPARSE_READ|Unknown) #REQUIRED
    replica_state (ON|NEW_REPLICA|DYING_REPLICA|LOCKED|
    CRT_0|CRT_1|TRANSITION_ON|DEAD_REPLICA|
    BEGIN_ADD|MASTER_START|MASTER_DONE|
    FEDERATED|SS_0|SS_1|JS_0|JS_1|MS_0|MS_1|
    Unknown) #REQUIRED>
]>
```

DTD に含まれる属性について次の表で説明します。

属性	説明
backup version	バックアップツールのバージョン。
backup backup_type	フルバックアップかインクリメンタルバックアップかの別コールドバックアップはフルバックアップとして扱います。
backup idtag	バックアップ時刻に基づいて生成された GUID。バックアップファイル名が変わっていても、これを使えばバックアップを識別できます。
backup time	バックアップ処理の開始日時。
backup srvname	バックアップ対象サーバの識別名。
backup dsversion	サーバ上で稼働している OpenText eDirectory のバージョン。
backup compression	バックアップツールがバックアップデータを圧縮したかどうか。これはバックアップデータのみに適用されます。ヘッダ自体が圧縮されることはありません。
backup os	バックアップ処理を実行したオペレーティングシステム。復元はこれと同じオペレーティングシステムのみで行うようお勧めします。
backup current_log	復元に必要な最初のロールフォワードログ。これは復元用の複数の正しいファイルを収集するのに役立ちます。
backup number_of_files	分割してバックアップする場合のファイル数。ひとつ目のバックアップファイルにのみ記録されます。
backup backup_file	このバックアップファイル名。 複数のファイルに分けてバックアップする場合、各ファイル名には順序番号がつきますが、それも含むファイル名が記録されます。複数のバックアップファイルのファイル名の例は、「-s *file_size*」を参照してください。

属性	説明
backup incremental_file_ID	インクリメンタルバックアップの場合、そのファイルの ID。
backup next_inc_file_ID	これに続くインクリメンタルバックアップに与える ID。正しく復元するために必要な情報です。
ファイルサイズ	このファイルの <file> タグ間にあるデータ長。
file name	バックアップファイル名およびその保存先。
file encoding	ファイルのエンコードに使ったアルゴリズム。
file type	これが NICI ファイルか、それ以外にユーザが指定したファイルかの別。
replica partition_DN	パーティションの識別名。 各レプリカをどのサーバに配置しているか記録していない場合に役立ちます。災害で何台ものサーバが停止した場合、バックアップファイルのヘッダに表示されたこの情報を見れば、最初にどのサーバを復元するべきか判断できます。
replica modification_time	バックアップ時点の、このレプリカの遷移ベクトル。
replica replica_type	レプリカの種別。マスタ、読み込み専用など。
replica_state	バックアップ時点でのレプリカの状態。「On」、「New Replica」など。

バックアップファイルのヘッダ例を次に示します。これは Windows サーバで作成したもので、NICI セキュリティファイルもバックアップ対象になっています。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE backup [
<!ELEMENT backup (file|replica)*>
<!ELEMENT file (#PCDATA)>
<!ELEMENT replica EMPTY>
<!ATTLIST backup version CDATA #REQUIRED
    backup_type (full|incremental) #REQUIRED
    idtag CDATA #REQUIRED
    time CDATA #REQUIRED
    srvname CDATA #REQUIRED
    dsversion CDATA #REQUIRED
    compression CDATA "none"
    os CDATA #REQUIRED
    current_log CDATA #REQUIRED
    number_of_files CDATA #IMPLIED
    backup_file CDATA #REQUIRED
    incremental_file_ID CDATA #IMPLIED
    next_inc_file_ID CDATA #IMPLIED>
<!ATTLIST file size CDATA #REQUIRED
    name CDATA #REQUIRED
    encoding CDATA "base64"
    type (user|nici) #REQUIRED>
<!ATTLIST replica partition_DN CDATA #REQUIRED
    modification_time CDATA #REQUIRED
```

```
    replica_type (MASTER|SECONDARY|READONLY|SUBREF|
    SPARSE_WRITE|SPARSE_READ|Unknown) #REQUIRED
    replica_state (ON|NEW_REPLICA|DYING_REPLICA|LOCKED|
    CRT_0|CRT_1|TRANSITION_ON|DEAD_REPLICA|
    BEGIN_ADD|MASTER_START|MASTER_DONE|
    FEDERATED|SS_0|SS_1|JS_0|JS_1|MS_0|MS_1|
    Unknown) #REQUIRED>
]>

<backup version="2" backup_type="full" idtag="3D611DA2" time="2002-8-
19'T10:32:35" srvname="\T=MY_TREE\O=novell\CN=DSUTIL-DELL-NDS"
dsversion="1041081" compression="none" os="windows"
current_log="00000003.log" next_inc_file_ID="2" number_of_files="0000001"
backup_file="c:\backup\header.bak"><replica partition_DN="\T=MY_TREE"
modification_time="s3D611D95_r1_e2" replica_type="MASTER"
replica_state="ON" /><replica partition_DN="\T=MY_TREE\O=part1"
modification_time="s3D611D95_r1_e2" replica_type="MASTER"
replica_state="ON" /><replica partition_DN="\T=MY_TREE\O=part2"
modification_time="s3D611D95_r1_e2" replica_type="MASTER"
replica_state="ON" /><replica partition_DN="\T=MY_TREE\O=part3"
modification_time="s3D611D96_r1_e2" replica_type="MASTER"
replica_state="ON" /><file size="190"
name="C:\WINDOWS\system32\novell\nici\bhawkins\XARCHIVE.001"
encoding="base64" type="nici">the data is included here</file>

<file size="4228"
name="C:\WINDOWS\system32\novell\nici\bhawkins\XMGRCFG.KS2"
encoding="base64" type="nici">the data is included here</file>

<file size="168"
name="C:\WINDOWS\system32\novell\nici\bhawkins\XMGRCFG.KS3"
encoding="base64" type="nici">the data is included here</file>

<file size="aaac" name="C:\WINDOWS\system32\novell\nici\nicintacl.exe"
encoding="base64" type="nici">the data is included here</file>

<file size="150" name="C:\WINDOWS\system32\novell\nici\NICISDI.KEY"
encoding="base64" type="nici">the data is included here
</file>

<file size="4228"
name="C:\WINDOWS\system32\novell\nici\system\Xmgrcfg.ks2"
encoding="base64" type="nici">the data is included here
</file>

<file size="168" name="C:\WINDOWS\system32\novell\nici\system\Xmgrcfg.ks3"
encoding="base64" type="nici">the data is included here
</file>

<file size="1414" name="C:\WINDOWS\system32\novell\nici\xmgrcfg.wks"
encoding="base64" type="nici">the data is included here
</file>

</backup>
```

ヘッダ部に続き、データベースのバックアップデータをバイナリ形式で格納します。

バックアップログファイルの書式

OpenText eDirectory バックアップツールは、前回までのバックアップを含め、処理内容を細かく記録したログを残すようになっています。このログファイルにはすべてのバックアップ履歴、バックアップの開始および終了時刻、およびバックアッププロセス中に発生した考えられるエラーについての情報が含まれます。前回までのログに追記する形で記録します。書き出し先を別途指定することもできます。

無人バックアップが正常に実行されているか確認するためにも、このログは重要です。最終行を見ると、成功か失敗かの区別およびエラーコードがわかります。

バックアップツールのログファイルには、過去のバックアップ処理の ID も記録されています。これは復元に必要なフルバックアップ、インクリメンタルバックアップのファイルを間違いなく揃えるために役立ちます。先頭の 4 行は、バックアップファイルのヘッダ情報をそのまま複写したものです。

また、同時にバックアップしたファイル名も記録します。NICI ファイルや、インクルードファイルでユーザが指定したファイルがこれに当たります。

復元処理の際は、実際に復元されたファイルを記録します。

ログファイルの出力例を次に 2 つ示します。

```
|==================DSBackup Log: Backup================|
Backup type: Full
Log file name: sys:/backup/backup.log
Backup started: 2002-6-21'T19:53:5GMT
Backup file name: sys:/backup/backup.bak
Server name: \T=VIRTUALNW_TREE\O=novell\CN=VIRTUALNW
Current Roll Forward Log: 00000001.log
DS Version: 1041072
Backup ID: 3D138421
Backing up security file: sys:/system/nici/INITNICI.LOG
Backing up security file: sys:/system/nici/NICISDI.KEY
Backing up security file: sys:/system/nici/XARCHIVE.000
Backing up security file: sys:/system/nici/XARCHIVE.001
Backing up security file: sys:/system/nici/XMGRCFG.KS2
Backing up security file: sys:/system/nici/XMGRCFG.KS3
Backing up security file: sys:/system/nici/XMGRCFG.NIF
Starting database backup...
Database backup finished
Completion time 00:00:03
Backup completed successfully

|==================DSBackup Log: Restore================|
Log file name: sys:/save/doc.log
```

```
Restore started: 2002-7-19'T19:1:34GMT
Restore file name: sys:/backup/backup.bak
Starting database restore...
Restoring file sys:/backup/backup.bak
Restoring file sys:/system/nici/INITNICI.LOG
Restoring file sys:/system/nici/NICISDI.KEY
Restoring file sys:/system/nici/XARCHIVE.000
Restoring file sys:/system/nici/XARCHIVE.001
Restoring file sys:/system/nici/XMGRCFG.KS2
Restoring file sys:/system/nici/XMGRCFG.KS3
Restoring file sys:/system/nici/XMGRCFG.NIF
Database restore finished
Completion time 00:00:15
Restore completed successfully
```

DSMASTER サーバによる災害対策

複数サーバ環境で、サーバがすべて失われてしまうような災害にも備えるためには、ツリーの一部として DSMASTER サーバを導入することを検討してください。

バックアップツールは、各サーバを個別にバックアップするために使用されます。これはツリーではなくサーバを単位とします。ただし、DSMASTER サーバを作成しておけば、バックアップツールでも、ツリー構造全体をバックアップできるようになります。運用例の概要については、「512 ページの 「シナリオ : 複数サーバ構成のネットワークで、すべてのサーバが使えなくなった場合」」を参照してください。

災害からの復旧で問題となるのは、同じパーティションのレプリカが複数あって、互いに整合性が取れていない場合の対処方法です。災害によってロールフォワードログも失われている場合、すべてのサーバを同じ時点の状態に復元することはできません。バックアップされたレプリカはサーバごとに異なる時点のものなので、ロールフォワードログを使わずに単純に復元しただけでは、全体をツリーとしてまとめる際に問題が起こるためです。

注 : 復元検証プロセスは、これらの問題を防ぐのに役立ちます。復元した OpenText eDirectory データベースが他のレプリカと整合が取れていない場合、デフォルトでは、そのデータベースはオープンできません。

DSMASTER サーバを導入すればこのような事態に備えることができます。ツリー全体のマスタコピーを作成しておき、これを復元の基準点として使用します。

DSMASTER サーバを導入する手順を次に示します。

- ある 1 台のサーバに、ツリーに属するパーティションすべてのレプリカを保持するよう設定してください。これにより、ツリー全体のコピーが、特定のサーバの OpenText eDirectory データベース中に作られるようになります。ただし、ツリーの容量が大きい場合は、複数台のキーサーバに分けても構いません。これを DSMASTER サーバと呼びます。DSMASTER サーバに作るレプリカは、マスタまたは読み書き用と設定してください。

注 : 複数台のキー DSMASTER サーバに分割する場合は、同じパーティションのレプリカが 2 台以上に作られることのないようにしてください。重複のない構成にしておけば、災害後の復元作業において、レプリカ間に不整合が生じることはありません。

災害から復旧する際に、最新のロールフォワードログは使えません。このログは各サーバのローカルファイルとして保存されるためです。したがって、複数台の DSMASTER サーバを、すべて同じ時点の状態に戻すことはできない場合があります。2 台の DSMASTER サーバに同じレプリカがあれば、これが一致せず、ツリーに不整合が生じる恐れがあります。そのため、障害復旧としては、同じパーティションのレプリカを複数の DSMASTER サーバに作成することは避ける必要があります。

レプリカ全般については、「62 ページの 「レプリカ」」を参照してください。

- DSMASTER サーバを定期的にバックアップして、ツリー全体のバックアップコピーを作成してください。災害対策としては、DSMASTER サーバのバックアップ以外にも、充分な予防措置を講じておくとよいでしょう。

以上のようにツリーを設計しておくと、災害が起こっても、迅速にツリー構造を再構築し、稼動させることができます。1 台だけのサーバ (あるいは少数のキーサーバ) のみを復元し、これをマスタレプリカとして扱うようにすればよいのです。

このツリーが稼動し始めてから、フル / インクリメンタルバックアップファイルを使用して、DSMASTER 以外のサーバも順次復元していきます。ロールフォワードログがないため、これらの他のサーバで復元処理の検証が失敗します。そこで、レプリカリングからいったん外し、DSRepair を使って、すべてのレプリカ情報を外部参照に変更します。その後、DSMASTER サーバ上のコピーからレプリカを作成して、改めてサーバにレプリカを追加します。この手順については、「503 ページの 「復元後の検証処理に失敗した場合の対処方法」」に記載されています。

災害により一部のサーバが失われた場合、操作手順はやや複雑になりますので、OpenText サポート部門に連絡してください。

遷移ベクトルと復元後の検証処理

遷移ベクトルとは、レプリカのタイムスタンプのことです。レプリカ作成時刻を 1970 年 1 月 1 日からの経過秒数で表したものと、レプリカ番号、および現在のイベント番号を組にして表示されます。ここで例を示します。

```
s3D35F377 r02 e002
```

バックアップと復元のコンテキストでは、復元されたサーバが参加しているレプリカリング内で正しく同期していることを確認するために遷移ベクトルが使用されます。

同じパーティションのレプリカを保持するサーバは、レプリカの同期を取るため、互いにデータをやり取りしています。サーバはレプリカリング内の他のサーバと通信するたびに、他のサーバの遷移ベクトルを記録しています。遷移ベクトルを使用すると、レプリカリング内の各レプリカが同期を保つためには、どのデータを送ればよいか、サーバが常に把握できます。サーバが停止するとこの通信が止まり、再び通信できるようになるまでの間、他のサーバはそのサーバに関して遷移ベクトルの更新や変更を記録しますが、送信しません。

あるサーバの eDirectory を復元した後、検証処理として、復元された遷移ベクトルをレプリカリングに属する他のサーバと比較します。これは、復元されたレプリカが他のサーバと同期の取れた状態であるかどうか確認するために実行されます。

リモートサーバの遷移ベクトルの方がローカルベクトルより後の時間に作成されたものである場合は、復元されなかったデータがあるということなので、検証処理は失敗します。これは、前回のフルバックアップまたはインクリメンタルバックアップの前に継続的なロールフォワードログ記録を有効にしていなかった、ロールフォワードログを復元に含めなかった、または復元で指定したロールフォワードログがすべて揃っていなかった、などの理由でデータが不足していることが考えられます。

デフォルトでは、復元した OpenText eDirectory データベースが他のレプリカと整合が取れていない場合、そのままではオープンできません。

遷移ベクトルに不整合があるログファイルエントリの例については、「466 ページの 「バックアップツールによる復元作業の概要」」を参照してください。

検証に失敗した場合の対処方法については、「503 ページの 「復元後の検証処理に失敗した場合の対処方法」」を参照してください。

ロールフォワードログを使用する

ロールフォワードログとは、他のデータベース製品でいう「ジャーナル」に相当する機能です。ロールフォワードログ (RFL) は、データベースの変更をすべて記録したものです。

ロールフォワードログを使用する利点は、最後のフル / インクリメンタルバックアップ以降の変更履歴が得られるため、障害で停止する直前の状態にまで OpenText eDirectory を復元できることです。ロールフォワードログを使用しないと、最後のフル / インクリメンタルバックアップを取った時点までしか OpenText eDirectory を復元できません。

OpenText eDirectory は、トランザクションをデータベースに反映する前に、その操作内容をログファイルに記録するようになっています。デフォルトでは、ログファイルはディスク容量の節約のために次々に重ね書きされるようになっているため、OpenText eDirectory データベースの変更履歴は残りません。

継続的にロールフォワードログを取る設定にすると、変更履歴が連続したロールフォワードログファイルに保存されます。ロールフォワードログはサーバの性能には影響がありません。OpenText eDirectory がすでに作成しているログファイルのエントリを単に保存するだけです。

レプリカリングに属するサーバは、ロールフォワードログ機能を有効にしておく必要があります。バックアップファイルがあっても、ロールフォワードログがなければ復元後の検証処理に失敗し、データベースを開けないことになります。他のサーバとレプリカを共有するデータベースは、停止直前の状態にまで復元しない限りデフォルトではオープンされません。ロールフォワードログがない場合は、「503 ページの 「復元後の検証処理に失敗した場合の対処方法」」で説明されている別の手順で復旧してください。

ロールフォワードログの機能は、デフォルトでは無効になっています。サーバで必要に応じて有効に切り替えてください。ロールフォワードログは、サーバの復元作業を実行すると再び無効になり、設定がデフォルトに戻ります。このため復元後に再び有効にし、設定を再作成した上で、改めてフルバックアップを取ってください。

注 : フルバックアップが改めて必要となるのは、スケジュールに従って次に無人でのフルバックアップが取られるまでに、再び障害が起こる可能性があるためです。

単一サーバ環境ではロールフォワードログがなくても構いません。しかしロールフォワードログがあれば、最後のバックアップ時ではなくシステム停止直前の状態まで OpenText eDirectory を復元できます。

ロールフォワードログをオンにする場合、ディスクの空き容量は常に監視している必要があります。詳細については、479 ページの 「ロールフォワードログのバックアップと削除」を参照してください。

この節では、次の項目について説明します。

- 476 ページの 「ロールフォワードログ機能を使用する上での注意事項」
- 477 ページの 「ロールフォワードログの保存先」
- 479 ページの 「ロールフォワードログのバックアップと削除」
- 480 ページの 「注意 : OpenText eDirectory を削除するとロールフォワードログも削除される問題」

ロールフォワードログの切り替えや設定には、DSBK を使います。または 488 ページの「DSBK によるロールフォワードログの設定」を参照してください。

ロールフォワードログ機能を使用する上での注意事項

継続的にロールフォワードログ機能を使用する場合、次のような点に注意してください。

- **バックアップ処理の実行前にロールフォワードログ機能を有効にしておかないと、**

  データベースの復元に利用することはできません。
- **障害に備えるため、eDirectory とは別の記憶デバイスにロールフォワードログを保存するようにしてください。**セキュリティを考慮すれば、ログへのアクセス権も制限する必要があります。詳細については、477 ページの 「ロールフォワードログの保存先」を参照してください。
- **ロールフォワードログの保存先を文書に記録しておいてください。**詳細については、477 ページの 「ロールフォワードログの保存先」を参照してください。
- **ログの保存先のディスクの空き容量を常に監視している必要があります。**詳細については、479 ページの 「ロールフォワードログのバックアップと削除」を参照してください。
- **ロールフォワードログ機能が無効になっていたり、ログファイルを損失した場合は、有効に切り替えた後、改めてフルバックアップを取ってください。**そうしなければ完全に復元できなくなる恐れがあります。次のような状況の場合に必要です。
  - 復元処理の直後。復元処理の過程で、ロールフォワードログ機能は無効になり、設定もデフォルト値に戻ってしまいます。
  - デバイス障害などにより、ロールフォワードログを保存しているディレクトリを損失した場合。
  - 意図せずにロールフォワードログ機能を無効にしてしまった場合。

- **ストリームファイルのログ機能を有効にすると、ディスクの空き容量が急速に減少します。**ストリームファイル (ログインスクリプトなど) のログ出力を有効にすると、変更があるたびに、ストリームファイル全体がロールフォワードログに複写されるためです。ストリームファイルのログ出力を無効にし、フル / インクリメンタルバックアップの際にのみストリームファイルをバックアップすると、ログファイルが大きくなるのを遅らせられます。
- **データベースの復元で最も時間を要するのは、ロールフォワードログを参照する処理です。**ロールフォワードログの容量は、ツリー構造に対して施された更新の回数に応じて増え、ストリームファイル (ログインスクリプトなど) のログ出力を有効にするとさらに増えます。

  データベースが頻繁に更新されるようであれば、OpenText eDirectory のバックアップ頻度を上げることも検討するとよいでしょう。こうすると、復元処理の過程でロールフォワードログを参照する処理が少なくなります。
- **ロールフォワードログファイルの名前を変更しないでください。**ログが作成されたときとファイル名が異なる場合、そのログファイルを復元に使用することはできません。
- **OpenText eDirectory を削除するとロールフォワードログもすべて削除されます。**復元にログを使用する予定がある場合は、OpenText eDirectory を削除する前に、ロールフォワードログを別の場所にコピーする必要があります。
- **復元が必要な場合は、復元処理の終了後にそのサーバのロールフォワードログ設定を再作成してください。**この機能を有効にし、ログの保存先を安全な場所に設定します。ロールフォワードログを有効にしてから、改めてフルバックアップも取る必要があります。

  この手順が必要となるのは、復元処理の過程で、ロールフォワードログに関する設定はデフォルトに戻るためです。つまり、ロールフォワードログ機能は無効となり、保存先もデフォルトの場所になるからです。フルバックアップが改めて必要となるのは、スケジュールに従って次に無人でのフルバックアップが取られるまでに、再び障害が起こる可能性があるためです。

ロールフォワードログの保存先

ロールフォワードログ機能を有効にした場合、ロールフォワードログディレクトリの場所を、OpenText eDirectory とは別の記憶デバイスに変更する必要があります。

保存先を設定する上で、次の点に注意してください。

- **デフォルトの場所のままにせず、必ず OpenText eDirectory とは別の記憶デバイス上に配置してください。**こうしておけば、記憶デバイスの障害のために OpenText eDirectory が失われても、ロールフォワードログにアクセスして OpenText eDirectory を復元できます。

  サーバに記憶デバイスが 1 つしかない場合、記憶デバイスの障害が起こると、ロールフォワードログでは OpenText eDirectory の耐障害性は保たれません。この機能はロールフォワードログを使用しないでおく方法もあります。

  ロールフォワードログの保存先を変更するには、setconfig DSBK のバックアップ環境設定を使用します。ロールフォワードログはサーバ上のローカルファイルとして保存する必要があります。

- **保存先を記録してください。**ロールフォワードログの保存先を記録して、サーバのデータベースの復元が必要なときに見つけられるようにしてください。これはサーバが正常で障害が発生する前に実行することが重要です。

  サーバが正常に動作していれば、DSBK で backup getconfig オプションを使って保存先を調べることができます。ただし、ハードウェア障害などで eDirectory が使えない状態になると、この方法でロールフォワードログの場所を調べることはできません。

  サーバに障害が発生し、それを復元する場合は、OpenText eDirectory を新たにインストールすると、ロールフォワードログの保存先設定はデフォルトの場所に戻ります。このため、復元作業のために OpenText eDirectory を再インストールしたとしても、サーバの停止前にロールフォワードログをどこに保存していたか、eDirectory で調べることはできません。その場合は記録を参照して位置を調べる必要があります。

  ロールフォワードログの保存先設定は、_ndsdb.ini ファイルにも記録されています。ただし、OpenText eDirectory と同じディスクパーティション / ボリュームにあるため、OpenText eDirectory がある記憶デバイスに障害が起これば、ログの保存先を調べるために _ndsdb.ini ファイルを使用することはできません。

- **ロールフォワードログの保存先へのアクセス権を制限してください。**これはセキュリティ上の問題です。見ただけで中身がわかるような形式にはなっていませんが、デコードは可能なため、重要なデータが漏洩する恐れがあります。
- **ディスクの空き容量が充分かどうか、常に監視している必要があります。**詳細については、479 ページの 「ロールフォワードログのバックアップと削除」を参照してください。
- **ロールフォワードログ専用のディスクパーティション/ボリュームを用意するのが最善です。**こうしておけば、ディスク容量やアクセス権を監視しやすくなります。
- **ログの保存先パスのうち、一番深い階層のディレクトリ名は OpenText eDirectory によって作成されます。**この名前は現在の OpenText eDirectory データベース名に基づいて決まります。

  たとえば、ログの保存先を d\Novell\NDS\DIBFiles と指定した場合、現在の OpenText eDirectory データベース名が「NDS」であれば、ロールフォワードログの実際の保存先ファイルは d:\Novell\NDS\DIBFiles\nds.rfl となります。データベースの名前を NDS から ND1 に変更した場合、ロールフォワードログのディレクトリは d:\Novell\NDS\DIBFiles\nd1.rfl に変更されます。

  保存先の設定を変えるとその時点で新しいディレクトリができますが、ロールフォワードログはデータベースでトランザクションが発生するまで作成されません。

- **復元の際は、必要なロールフォワードログをすべて同じディレクトリに集めます。**詳細については、480 ページの 「復元処理の準備」を参照してください。

ロールフォワードログのバックアップと削除

放置しておけばロールフォワードログは次々に蓄積され、ディスクパーティション / ボリュームがいっぱいになります。ディスク容量が不足してロールフォワードログを作成できない場合は、OpenText eDirectory はそのサーバに対して応答しなくなります。定期的にログファイルをバックアップし、サーバからは削除するようにして、常に充分なディスク容量を確保するようお勧めします。

削除しても構わないロールフォワードログを判別し、バックアップを取った上で削除するには、次の手順に従います。

1 「最後に使用済みになった」ロールフォワードログ名を調べてください。

DSBK Client で「最後に使用済みになった」ロールフォワードログの名前を確認し、getconfig backup コマンドを入力します。手順については、488 ページの 「DSBK によるロールフォワードログの設定」を参照してください。

「最後に使用済みになった」ロールフォワードログとは、トランザクション履歴の記録が終わり、今は書き出しをしていないログファイルのうち、最新のものを表します。データベースはこのファイルに対する書き込みを終えて、今は新しいログファイルで始めているため、オープンしておく必要はもうありません。そのため、「最後に使用済みになった」ロールフォワードログと呼ばれます。一方、データベースがトランザクションを記録している現在のロールフォワードログは「使用中」で、データベースに必要なものです。

2 ファイルシステムのバックアップ機能を使って、ロールフォワードログをテープに保存してください。

3 「最後に使用済みになった」ものよりも古いロールフォワードログを削除してください。

警告 : ロールフォワードログを削除する際にはより注意を払い、削除しようとするファイルが確実にバックアップされているか、繰り返し確認してください。

「最後に使用済みになった」ロールフォワードログとは、データベースが記録を完了しクローズにしたファイルを指します。サーバからこのファイルを削除しても構わないという意味ではありません。まだテープにバックアップしていないファイルは削除しないようにしてください。

テープに保存してあるロールフォワードログを復元のために使う場合は、次の点に注意してください。

- 復元に使用する他のロールフォワードログと同様に、ファイルシステムをバックアップしたテープから取得したログファイルは、他のログと合わせてひとつのフォルダに集めてください。このフォルダは、サーバからローカルにアクセスできる必要があります。
- テープおよびサーバに複製されたファイルのタイムスタンプを比較する必要があります。タイムスタンプに違いがある場合は、最新のサーバ上のファイルを使います。たとえば、ファイルシステムのバックアップ中にデータベースで使用されていたロールフォワードログファイルは、テープでは不完全になります。最新の完全なファイルはサーバに格納されています。

注意 : OpenText eDirectory を削除するとロールフォワードログも削除される問題

サーバから OpenText eDirectory を削除すると、ロールフォワードログのディレクトリおよびその中身もすべて削除されます。サーバの復元にログを使用する予定がある場合は、OpenText eDirectory を削除する前に、ロールフォワードログを別の場所にコピーする必要があります。

復元処理の準備

OpenText eDirectory データベースの復元作業で最も大切なのは、復元が完全に行われたかどうか確認することです。作業に先立ち、480 ページの 「復元作業の前提条件」の説明に従って、必要な準備を行います。必要なバックアップファイルを揃える手順については、「481 ページの 「復元に必要なバックアップファイルの収集」」を参照してください。

復元作業の前提条件

- ❒ 復元するサーバとレプリカを共有しているサーバはすべて、稼動状態で、通信できるようにしておかなければなりません。これは、復元後の検証処理で、同じレプリカリングに属するサーバ間の整合性を確認するために必要です。
- ❒ 必要な次のバックアップファイルをすべて収集してください。
  - フルバックアップおよびそれ以降のインクリメンタルバックアップのファイルを、復元するサーバの、1 つのディレクトリ内に集めます。
  - 最後にバックアップを取って以降のロールフォワードログをすべて、同じサーバ上のもう 1 つのディレクトリにまとめておきます。

    このサーバがレプリカリングに属している場合、最後にバックアップを取った時点以降のロールフォワードログをすべて、ひとつのディレクトリ内に集めておきます。ファイル名はログ生成時と同じにしておかなければなりません。

  詳細については、481 ページの 「復元に必要なバックアップファイルの収集」を参照してください。

注 : サーバのバックアップファイルがない場合は、Xbrowse を使用して、サーバ情報の回復に役立つ情報を OpenText eDirectory に問い合わせてください。この作業は、サーバオブジェクトやその関連オブジェクトをツリーから削除する前に実行する必要があります。

Xbrowse の詳細については、OpenText サポートの Web サイトを参照してください。

- ❒ OpenText eDirectory を再インストールし、仮のツリーで稼働させておきます。

  障害が起きる前と同じ名前のサーバを作成するので、まずサーバを新しいツリーで稼動させます。復元でサーバーの完全な識別情報を再作成する前に元のツリーに新しくインストールしたサーバを置くことで混乱が生じないようにするためです。データベースの復元処理が完了してから、本来のツリーにサーバを組み入れることになります。

- (状況によって実行) このサーバでロールフォワードログ機能を使うためには、復元後に改めて有効に切り替え、障害対策のための書き出し先も設定し直して、ロールフォワードログの環境設定を再作成する必要があります。ロールフォワードログを有効にしてから、改めてフルバックアップも取る必要があります。

  復元処理の過程で、ロールフォワードログの機能は無効になり、ログ保存先の設定もデフォルトに戻ってしまいます。

  フルバックアップが改めて必要となるのは、スケジュールに従って次に無人でのフルバックアップが取られるまでに、再び障害が起こる可能性があるためです。
- (状況によって実行)IPアドレスを指定してこのサーバにアクセスするアプリケーションやオブジェクトがある場合は、元と同じ IP アドレスを設定してください。

復元の過程で、OpenText eDirectory バックアップツールはまず、フルバックアップファイルからの復元を試みます。それが済むと、バックアップツールでインクリメンタルバックアップファイルの名前を入力するよう求められます。その際、次に適用するべきファイルの ID が提示されます。インクリメンタルバックアップファイルからの復元が終わると、今度はロールフォワードログを参照しての復元処理が始まります。466 ページの 「バックアップツールによる復元作業の概要」も参照してください。

必要なファイルをすべて揃えた後、DSBK Client から復元処理を起動します。「489 ページの「DSBK によるバックアップファイルの復元作業」」を参照してください。

復元に必要なバックアップファイルの収集

1. ファイルシステムをバックアップしたテープから、OpenText eDirectory フルバックアップファイルを、サーバ上の適当なディレクトリにコピーしてください。

   最後に取ったフルバックアップの ID は、バックアップツールのログファイルで確認できます。
2. 同様に、一連のインクリメンタルバックアップファイルを、サーバ上の適当なディレクトリにコピーしてください。

   必要なインクリメンタルバックアップファイルは、フルバックアップファイルのヘッダ部で確認できます。「next_inc_file_ID」属性として、次のインクリメンタルバックアップファイルの ID が記述されています。next_inc_file_ID は、インクリメンタルバックアップファイルのヘッダ部にある「incremental_file_number」属性の ID と同じです。ヘッダの説明は、「468 ページの 「バックアップファイルのヘッダ書式」」を参照してください。

   警告 : バックアップファイルを開くときは、ヘッダを確認するだけにしてください。ファイルを保存または変更しようとすると、ファイルの一部が切り捨てられる場合があります。ほとんどのアプリケーションがバイナリデータを正しく保存することができません。

   インクリメンタルバックアップファイルにはそれぞれ、次のインクリメンタルバックアップファイルの ID が記載されています。

   この ID もバックアップツールのログファイルで確認できます。

同じ名前のファイルがいくつもあって、ひとつのディレクトリにまとめるためにファイル名を変更しているような場合、ID はその識別に不可欠です。たとえば無人でのバックアップにいつも同じバッチファイルを使っていて、バックアップファイル名が常に同じであるような場合です。ヘッダ部の ID を見れば、ファイル名が変わっていても適切なファイルを判別できます。

3 (状況によって実行) ロールフォワードログ機能を有効にしていた場合は、最後のバックアップ以降のロールフォワードログを、生成時のファイル名のまま、サーバ上の適当なディレクトリに集めてください。

このサーバがレプリカリングに属している場合は、ロールフォワードログを使った復元処理が必須です。ロールフォワードログがすべて揃っていない場合、復元検証は失敗してしまいます。リング内の他のレプリカと比較して遷移ベクトルが一致しないからです。デフォルトでは、復元した OpenText eDirectory データベースが他のレプリカと整合が取れていない場合、そのままではオープンできません。

テキストエディタで最新のバックアップファイルを開き、ヘッダの「current_log」属性を読んで、最初に必要なロールフォワードログを特定します。この作業を繰り返して、続くすべてのロールフォワードログを集めます。

警告 : バックアップファイルを開くときは、ヘッダを確認するだけにしてください。ファイルを保存または変更しようとすると、ファイルの一部が切り捨てられる場合があります。ほとんどのアプリケーションがバイナリデータを正しく保存することができません。

必要なロールフォワードログがすべて 1ヶ所にまとまっているとは限りません。よく確認して、すべて同じディレクトリに揃えてください。ロールフォワードログは、次の理由から複数の場所に格納されている場合があります。

- 最後に eDirectory のフル / インクリメンタルバックアップを実行してから、ロールフォワードログの保存先を変更した場合。
- ファイルシステムのバックアップを使用して、ロールフォワードログをテープにバックアップした後で、空きディスク容量を確保するためにそれらのファイルを削除した場合。

  テープにバックアップされたロールフォワードログを取得する場合は、データが最新のセットであることを確認してください。テープおよびサーバに複製されたファイルのタイムスタンプを比較する必要があります。ファイルシステムのバックアップ中にデータベースで使用されていたロールフォワードログファイルは、テープでは不完全になります。最新の完全なファイルはサーバに格納されています。
- 最後にバックアップを実行してから OpenText eDirectory データベースの名前を変更した (NDS から ND1 に変更した場合など)。この変更により、ロールフォワードログのパスの最後のディレクトリ名が変更されます。

  たとえばログの保存先を d:\novell\nds\dibfiles\ と指定した場合、OpenText eDirectory データベース名が「NDS」であれば、ロールフォワードログの実際の保存先ファイルはディレクトリ d:\novell\nds\dibfiles\nds.rfl\ となります。データベースの名前を NDS から ND1 に変更した場合、ロールフォワードログのディレクトリは D:\Novell\nds\dibfiles\nd1.rfl\ に変更されます。

重要 : 必要なロールフォワードログがすべてそろっていることを確認してください。バックアップツールでは、ロールフォワードログがすべてそろっているかどうか確認できません。ロールフォワードログは順番に開かれて使用されます。指定したディレクトリ内に次のロールフォワードログが見つからない場合は、復元プロセスが中止されます。必要なロールフォワードログがすべてそろっていなければ復元は完了しません。

DSBK の使用

DSBK は、OpenText eDirectory バックアップを実行する簡易なコマンドラインパーサです。ただし、DSBK では最初にログインしたり役割ベースサービスを設定することなく、サーバコンソールからバックアップを実行できます。これは、Linux ではスクリプトとして実行され、Windows ではコンソールユーティリティとして実行されます。

DSBK 操作が完了すると、操作の結果がファイル (Linux では dsbk.err) に書き込まれ、プログラムを使用して開き、その内容を表示することができます。操作時にエラーが発生した場合は、このファイルの最初の 4 バイトにエラーコードが記録されます。エラーが発生しなかった場合、このファイルの最初の 4 バイトには 0 が記録されます。

注 : バックアップまたは復元の設定を完了する前に、OpenText によって指定されたすべてのガイドラインを済ませていることを確認します。

eDirectory のバックアップ / 復元作業に先立ち、「OpenText eDirectory のバックアップ処理に関する確認事項」を参照して問題点を確認し、効率的に作業できるようにしてください。

このセクションでは、次の点を説明します。

- 483 ページの 「前提条件」
- 484 ページの 「さまざまなプラットフォームで DSBK を使用する」
- 487 ページの 「DSBK による手動バックアップ」
- 488 ページの 「OpenText eDirectory のバックアップの自動化」
- 488 ページの 「DSBK によるロールフォワードログの設定」
- 489 ページの 「DSBK によるバックアップファイルの復元作業」
- 491 ページの 「バックアップ / 復元のコマンドラインオプション」
- 501 ページの 「cron ジョブとしての DSBK の実行」

前提条件

❒ ロールフォワードログを作成するのであれば、バックアップを行う前にこの機能を有効にしてください。

レプリカリングに属するサーバは、ロールフォワードログ機能を有効にしておく必要があります。バックアップファイルがあっても、ロールフォワードログがなければ復元後の検証処理に失敗し、データベースを開けないことになります。

ロールフォワードログの詳細については、「ロールフォワードログを使用する」を参照してください。また、この機能を有効にする手順については、DSBK によるロールフォワードログの設定を参照してください。

- OpenText eDirectory 以外にも追加でバックアップしたいファイルがあれば、それを列挙したインクルードファイルを作っておいてください。

  スイッチを使用して、ストリームファイルをバックアップすることができます。NICI ファイルは常にバックアップするようお勧めします。NICI のバックアップ方法の詳細については「NICI のバックアップと復元」を参照してください。

  それ以外に autoexec.ncf ファイルなどをバックアップしたい場合は、そのパスとファイル名をインクルードファイルに列挙します。複数のファイルがある場合はセミコロンで区切ります。改行 (ハードリターン) や空白を含めないようにしてください。例 : sys:\system\autoexec.ncf;sys:\etc\hosts;

- eDirectory のバックアップ後すぐに、ファイルシステムのバックアップ作業を行い、テープに保存できるよう準備してください。バックアップツールによる処理では、サーバ上にバックアップファイルができるだけです。

ヒント : コピー先ストレージデバイスに容量の制約がある場合は、あらかじめ eDirectory バックアップファイルの最大サイズを設定すると便利です。その場合、backup コマンドの「-s」オプションを使い、バイト単位で指定します。また、バックアップファイルの作成後、サードパーティ製ファイル圧縮ツールを使う方法もあります。80% 程度は圧縮できます。

- コマンドラインオプションについては、「バックアップ / 復元のコマンドラインオプション」を参照してください。

さまざまなプラットフォームで DSBK を使用する

- 484 ページの 「Linux 上で DSBK を使用する」
- 486 ページの 「Windows で DSBK を使用する」

Linux 上で DSBK を使用する

DSBK コマンドは、OpenText eDirectory がインストールされている Linux サーバのシェルで直接実行できます。

コマンドの出力は OpenText eDirectory インスタンス固有のログファイルに書き込まれます (デフォルトのインスタンス : /var/opt/novell/eDirectory/log/ndsd.log)。

```
DSBK HELP
To get help on a specific function type "help <function name>"
 Current functions:
  backup
  restore
  restadv
  getconfig
  setconfig
  cancel
```

DSBK コマンドを crontab に入力して、dsbk getconfig および dsbk backup コマンドを定期的に実行し、週 1 回フルバックアップを行い他の日にインクリメンタルバックアップを行ったり、必要に応じて任意の組み合わせを行ったりすることができます。

DSBK で RFL を使用する

- 次のコマンドを使用して RFL をオンにします。

  ```
  dsbk setconfig -L
  ```

  -L オプションにより、新しいロールフォワードログ記録セッションを開始します。
- 次のコマンドを使用して、作成されるロールフォワードログの場所を設定します。

  ```
  dsbk setconfig -L -r <roll forward log directory>
  ```
- 次のコマンドを使用して、作成されるロールフォワードログの場所を取得します。

  ```
  dsbk getconfig
  ```
- -a オプションを使用して、バックアップ時に、ロールフォワードログのディレクトリから古いログファイルを削除します。

  ```
  backup -f <file name> -l <file name> [-s <size>] [-u <file name>] [-e
  <password>] [-t] [-w] [-a]
  [-b|-i|-c] [-o] [-d] [--config-file <configuration file>]
  ```

ヒント : DSBK ユーティリティを対話形式で使用する場合は、2 つ目の端末ウィンドウを開いて、tail -f <instance specific ndsd.log> を実行します。これにより、入力されたコマンドに対する出力をすぐに読み込むことができます。

バックアップが完了したら、標準のファイルシステムバックアップユーティリティを使用してバックアップします。

注 : DSBK コマンドラインオプションに関する詳細については、「バックアップ / 復元のコマンドラインオプション」を参照してください。

Windows で DSBK を使用する

このセクションでは、Windows プラットフォームでの DSBK ユーティリティの基本的な操作について説明します。

OpenText eDirectory をホストする Windows サーバ上で DSBK を使用する場合、次の手順を実行します。

1. **OpenText eDirectory サービス**コンソールからユーティリティを起動します。**dsbk.dlm** は［サービス］タブのサービスリストで使用可能なオプションの 1 つです。**dsbk** サブコマンドとそのサブコマンドのパラメータは［**起動パラメータ**］フィールドで指定します。
2. getconfig スイッチを使用して、バックアップの現在の設定を表示します。すべての DSBK コマンドの出力は、Windows の OpenText eDirectory インストールフォルダにある backup.out ファイルに追加されます。
3. 次のコマンドを使用して、作成されるロールフォワードログの場所を設定します。

   ```
   setconfig -r <roll forward log directory> -L
   ```

   -L オプションにより、新しいロールフォワードログ記録セッションを開始します。
4. 次のコマンドで、ツリー上のバックアップを開始します。

   ```
   backup -f <backup file> -l <logfile> -t -w -b -e <password>
   ```

   次のオプションを使用します。

   - -t: ストリームファイルのバックアップを取ります。
   - -w: 同じ名前の既存のバックアップファイルを上書きします。
   - -b: フルバックアップを実行します。
   - -e *<password>*: 指定したパスワードを使用して NICI バックアップを実行します。
   - -a: 「ホット」バックアップ中にロールフォワードログのディレクトリから古いログファイルを削除します。

   たとえば、次のようにバックアップを起動します。

   ```
   backup -f c:\dsbk.bak -l c:\backup.log -t -w -b -e novell
   ```

   完了したバックアップのステータスは backup.out ファイルで確認できます。

注 : DSBK コマンドラインオプションに関する詳細については、「バックアップ / 復元のコマンドラインオプション」を参照してください。

次のコマンドを使用して RFL をオンにすることができます。

```
setconfig -r <roll forward log directory> -L
```

DSBK による手動バックアップ

DSBK を使って、OpenText eDirectory データベースの中身を、指定したファイルにバックアップすることができます。バックアップファイルには、OpenText eDirectory をバックアップ時点の状態に復元するために必要な情報がすべて含まれています。バックアップの処理結果は所定のログファイルに記録されます。

DSBK を使うと次のような作業ができます。

- データベースを開いたままで、フル / インクリメンタルバックアップ (ホットバックアップ)。

  「ホット」バックアップの場合、処理中も OpenText eDirectory データベースは開いたままでありながら、バックアップ開始時点のスナップショットである完全なバックアップを作成できます。
- コールドバックアップ (データベースをいったん停止してフルバックアップ)。

  この機能は、ハードウェアをアップグレードする、あるいはサーバを新規マシン (同じオペレーティングシステムが動作するもの) に移行する場合に有用です。詳しくは、ハードウェアのアップグレードやサーバの交換 を参照してください。
- バックアップ後、データベースを閉じたままにしてロックする設定。
- バックアップファイルの最大サイズの設定。

手順

DSBK を使って OpenText eDirectory データベースをバックアップする手順を次に示します。

1 次の一般的な形式に従って、dsbk backup コマンドを入力します。

dsbk backup -b *-f backup_filepath_and_backup_filename* -l *backup_log_filename_and_path* -u *include_file_filename_and_path* -t -w

各スイッチの間にはスペースが必要です。スイッチの順序は重要ではありません。

たとえば、Windows では、次のコマンドを入力します。

dsbk backup -b -f c:\backups\8_20_2001.bak -l c:\backups\backup.log -u c:\backups\myincludefile.txt -t -w

この例では、フルバックアップを取ること (-b)、バックアップファイルを c:\backups\8_20_2001.bak とすること、処理結果を c:\backups\backup.log に出力すること、さらに、データベースとともに、次に示す他のファイルもバックアップすることを指定しています。

- 管理者があらかじめ作成したインクルードファイル (-u c:\backups\myincludefile.txt) に列挙されたファイル。
- ストリームファイル (-t)。

さらにこの例では「-w」オプションが指定されているため、同じ名前のバックアップファイルがあれば上書きされます。

出力は dsbackup.out ファイルに書き込まれ、バックアップが成功したかどうかを示します。

OpenText eDirectory のバックアップ処理が終了したら、すぐにファイルシステムのバックアップ作業を行い、テープに保存します。バックアップツールによる処理では、サーバ上にバックアップファイルができるだけです。

OpenText eDirectory のバックアップの自動化

OpenText eDirectory のバックアップを自動化するには、次のコマンドをバッチに記述します。

```
dhostcon.exe 192.168.1.1 load dsbk backup -b -f <Backup File> -l <Log File>
-t -w
```

次に例を示します。

```
c:\novell\nds\dhostcon.exe 192.168.1.1 load dsbk backup -b -f
edirbackup.bak -l c:\novell\edir-backup.log -t -w
```

このファイルを eDirectory がインストールされている場所に保存します。

DSBK によるロールフォワードログの設定

DSBK を使用して、ロールフォワードログの設定を変更します。次のような設定ができます。

- 現在の設定の確認
- ロールフォワードログ機能の有効 / 無効の切り替え

  レプリカリングに属するサーバは、ロールフォワードログ機能を有効にしておく必要があります。バックアップファイルがあっても、ロールフォワードログがなければ復元後の検証処理に失敗し、データベースを開けないことになります。
- ロールフォワードログの保存先ディレクトリの変更
- ロールフォワードログのファイルサイズの最小値、最大値の設定
- 現在使用中のログ、既に書き出しを終えた最新のログの判別
- ストリームファイルをロールフォワードログに含めるかどうかの切り替え

ロールフォワードログの詳細については、「475 ページの 「ロールフォワードログを使用する」」を参照してください。

手順

1 次のように入力して、現在の設定を確認します。

dsbk getconfig

オプション指定は必要ありません。

たとえば次のように表示されます。

```
Roll forward log status OFF
Stream file logging status OFF
Current roll forward log directory C:\rfl\nds.rfl
Minimum roll forward log size (bytes) 104857600
Maximum roll forward log size (bytes) 4294705152
Last roll forward log not used 00000000.log
Current roll forward log 00000001.log
*** END ***
```

2 setconfig コマンドで設定を変更します。次のような形式で入力してください。

dsbk setconfig [-L|-l] [-T|-t] -r *path_to_roll-forward_logs* -n *minimum_file_size* -m *maximum_file_size*

各スイッチの間にはスペースが必要です。スイッチの順序は重要ではありません。

ロールフォワードログ専用のディスクパーティション / ボリュームを用意するのが最善です。こうしておけば、ディスク容量やアクセス権を監視しやすくなります。

警告 : ロールフォワードログ記録を有効にしたら、デフォルトの保存先は使用しないでください。障害対策上、OpenText eDirectory とは別のディスクパーティション / ボリューム、別の記憶デバイスを指定してください。ロールフォワードログディレクトリは、バックアップ環境設定を変更するサーバ上である必要があります。

重要 : ロールフォワードログ機能を有効にする場合、ログを保存するボリュームのディスク容量を常に監視してください。これを怠ると、ログの容量は増える一方なので、ディスクパーティション / ボリュームがあふれてしまう恐れがあります。ディスク容量が不足してロールフォワードログを作成できない場合は、OpenText eDirectory はそのサーバに対して応答しなくなります。書き出しが終わったロールフォワードログは、定期的にバックアップし、サーバから削除するようお勧めします。詳細については、479 ページの 「ロールフォワードログのバックアップと削除」を参照してください。

DSBK によるバックアップファイルの復元作業

DSBK を使用して、手動で作成したバックアップファイルに保存されたデータから、OpenText eDirectory データベースを復元します。復元の処理結果は所定のログファイルに記録されます。

DSBK では、高度な復元オプションも使用できます。これらは、「491 ページの 「バックアップ / 復元のコマンドラインオプション」」の restore および restadv の項に記載されています。

追加の前提条件

❒ 復元対象サーバに OpenText eDirectory をインストールし、稼働させておいてください。

たとえば記憶デバイスの障害の場合、デバイスを交換し、改めて OpenText eDirectory をインストールすることになります。故障したサーバごと交換する、あるいは単に新しいサーバに移行する場合は、新しいサーバにオペレーティングシステムをインストールした上で、OpenText eDirectory も準備します。

- コマンドラインオプションについては、「491 ページの 「バックアップ / 復元のコマンドラインオプション」」を参照してください。
- 復元処理の詳細については、「466 ページの 「バックアップツールによる復元作業の概要」」を参照してください。

手順

DSBK を使って OpenText eDirectory データベースを復元する手順を次に示します。

1. 必要なバックアップファイルを集めておきます。詳しくは「480 ページの 「復元処理の準備」」を参照してください。
2. 次の一般的な形式に従って、dsbk restore コマンドを入力します。

   dsbk restore -r -a -o -f *full_backup_path_and_filename* -d *roll-forward_log_location* -l *restore_log_path_and_filename*

   各スイッチの間にはスペースが必要です。スイッチの順序は重要ではありません。-r スイッチを使用して、OpenText eDirectory データベース自体を復元してください。そうしないと、他の種類のファイルだけが復元されます。復元処理の終了後にデータベースをアクティブにして開くには、-a および -o を指定してください。

   ロールフォワードログを使って復元する場合は、ログのフルパスを指定しなければなりません。フルパスには、OpenText eDirectory により自動的に作成されたディレクトリ (通常は \nds.rfl) も含まれます。このディレクトリの詳細については、477 ページの「ロールフォワードログの保存先」を参照してください。

   次に例を示します。

   dsbk restore -r -a -o -f $HOME/backup/nds.bak -d $HOME/backup/rfldir/nds.rfl -l $HOME/backup/backup.log

   この例では、データベースそのものを復元し (-r)、復元の検証が正常終了してから、そのデータベースをアクティブにして (-a)、開く (-o) よう指定しています。-f スイッチでフルバックアップファイルの場所を、-d でロールフォワードログの場所を指定しています。また、復元処理の結果を記録するログファイルを、-l で指定しています。

   DSBK はフルバックアップを復元します。復元が正常に終了したかどうかを示す出力が ndsd.log に書き込まれます。
3. (状況によって実行) 復元処理に失敗した場合は、ログファイルでエラーの原因を確認してください。

   復元後の検証に失敗した場合の対処については、「503 ページの 「復元後の検証処理に失敗した場合の対処方法」」を参照してください。

   注 : レプリカリング中に eDirectory 8.5 より前のバージョンが稼動しているサーバがある場合、復元処理は失敗します。エラーコードは -666、すなわち「DS バージョンの不整合」となります。

   ---
4. (状況によって実行)NICI セキュリティファイルを復元した場合は、復元の完了後、サーバを再起動して NICI を再初期化し、その後で DIB を復元します。
5. ここでサーバが通常どおり要求に応答することを確認しておきます。

6 (状況によって実行) このサーバでロールフォワードログ機能を使うためには、改めて有効に切り替え、障害対策のための書き出し先も設定し直して、ロールフォワードログの環境設定を再作成する必要があります。ロールフォワードログを有効にしてから、改めてフルバックアップも取る必要があります。

この手順が必要となるのは、復元処理の過程で、ロールフォワードログに関する設定はデフォルトに戻るためです。つまり、ロールフォワードログ機能は無効となり、保存先もデフォルトの場所になるからです。フルバックアップが改めて必要となるのは、スケジュールに従って次に無人でのフルバックアップが取られるまでに、再び障害が起こる可能性があるためです。

ロールフォワードログの詳細については、「475 ページの 「ロールフォワードログを使用する」」を参照してください。

以上で復元作業が終了しました。NICI の再初期化も済んでいるので、暗号化された情報にもアクセスできます。ロールフォワードログ機能を使用する場合は、今後の障害に備えるため、再びこの機能を有効にし、フルバックアップを取っておいてください。

バックアップ / 復元のコマンドラインオプション

OpenText eDirectory バックアップツールのコマンドラインオプションは、backup、restore、restadv、getconfig、setconfig、および cancel の 6 つの関数に分類されます。

オプションはどのような順序で指定しても構いません。各オプション間は空白で区切ってください。

オプション	説明
backup	データベースおよび関連ファイルのバックアップ
-f *file_name*	(必須) バックアップファイルの名前とパス。 バックアップツールで作成するバックアップファイルのファイル名と場所を指定します。バックアップ対象サーバ上のローカルファイルを指定してください。たとえば、backup -f C:\backup\ndsbak.bak と指定すると、データベースが C:\backup\ndsbak.bak にバックアップされます。
-l *file_name*	(必須) ログファイルの名前とパス バックアップ処理の結果を出力するログファイルを指定します。
-b	(オプション) フルバックアップを実行 . OpenText eDirectory データベースのフルバックアップを取ります。これがデフォルトの動作で、「-i」も「-c」も指定しなければフルバックアップになります。■分節拡張■ -i と -c のどちらも指定しなかった場合は、フルバックアップが実行されます。
-i	(オプション) インクリメンタルバックアップを実行 . OpenText eDirectory データベースのインクリメンタルバックアップを取ります。最後に実施したフル / インクリメンタルバックアップ以降、変化した部分のみをバックアップします。

オプション	説明
-t	(オプション) ストリームファイルもバックアップ . OpenText eDirectory データベースをバックアップする際、ストリームファイルも含める指定です。
-u *file_name*	(オプション) インクルードファイル名およびパス . バックアップ対象に追加するファイル名を列挙した、インクルードファイルを指定します。この環境設定ファイルは、サーバの OpenText eDirectory データベースを復元する際に必要な他のファイルをバックアップに含めるように作成できます。 インクルードファイルには各ファイルのフルパスを記述し、末尾にセミコロン (;) を置いてください。 ファイルのリストにスペースやハードリターンを含めないでください。 指定どおりバックアップされたことは、ログファイルを見るか、バックアップファイルのヘッダを見れば確認できます。472 ページの「バックアップログファイルの書式」および 468 ページの 「バックアップファイルのヘッダ書式」を参照してください。 **警告 :** バックアップファイルを開くときは、ヘッダを確認するだけにしてください。ファイルを保存または変更しようとすると、ファイルの一部が切り捨てられる場合があります。ほとんどのアプリケーションがバイナリデータを正しく保存することができません。

オプション	説明
-s *file_size*	(オプション) バックアップファイルの最大容量 (MB 単位)。 バックアップファイルの最大容量 (MB) を指定します。バックアップファイルを保存する記憶媒体に容量制限がある場合、このオプションで最大容量を指定するとよいでしょう。 最大容量に達すると新しいバックアップファイルが生成されます。ファイル名の末尾に、5 桁の 16 進数値を追加した名前になります。この拡張の数字は、新規ファイルが作成されるたびに 1 ずつ増加します。 たとえば、コマンドの一部として「backup -f 10:/backup/mydib.bak -s 10」というスイッチを使用した場合は、バックアップファイルの最大容量を 10MB に設定できます。データベースが 35MB あったとすれば、最終的に以下のバックアップファイルができます。 C:\backup\mydib.bak、サイズは 9.6MB C:\backup\mydib.bak.00001、サイズは 9.6MB C:\backup\mydib.bak.00002、サイズは 9.6MB C:\backup\mydib.bak.00003、サイズは 5.6MB 最小可能サイズは約 1MB です。バックアップで作成されるファイル数によって、最初のファイルが大きくなる場合があります。 最初のファイルには、number_of_files というバックアップタグの下に属性が含まれています。これはバックアップファイルの総数を表します。上記の例では 4 となります。また、各バックアップファイルのヘッダに、backup_file という属性が含まれています。これは本来のバックアップファイル名を表します。詳細については、468 ページの「バックアップファイルのヘッダ書式」を参照してください。 上記の 4 つのバックアップファイルを使って復元する場合、コマンドは次のようになります。 restore -f C:\backup\mydib.bak -l *log_file_path_and_filename* ファイルが複数に分かれていることはバックアップツールによって自動的に認識され、同じディレクトリ内にある、上記の名前のファイルが検索されます。 **ヒント :** サードパーティ製のファイル圧縮ツールを使えば、バックアップファイルの容量を小さくすることができます。80% 程度は圧縮できます。

オプション	説明
-w	(オプション) 同名のバックアップファイルがあれば上書き -f スイッチで指定されたものと同じ名前のバックアップファイルがあれば、上書きします。この指定がない場合で同名のファイルが存在すると、対話式モードであれば、バックアップツールは上書きしてよいかどうか確認を求めます。バッチモードでは、同じ名前のファイルが存在する場合に -w が指定されていないと、デフォルトの動作として、ファイルは上書きされず、バックアップは作成されません。 OpenText eDirectory のフル / インクリメンタルバックアップの都度、すぐにファイルシステムのバックアップを取っているのであれば、前回のバックアップファイルはテープに保存されているはずです。したがって上書きしても問題ありません。 **重要 :** バッチファイルを使って無人バックアップを行う場合、このオプションを指定してください。同じバッチファイルを繰り返し使用するなど、同じ名前のバックアップファイルが存在する場合、バックアップが正常に行われるように、-w オプションを使用して、既存のバックアップファイルが上書きされるようにしてください。 バッチモードでは、同じ名前のファイルが存在する場合に -w が指定されていないと、デフォルトの動作として、ファイルは上書きされず、バックアップは作成されません。なお、対話式モードの場合は、-w が指定されていないと、ファイルを上書きしてよいかどうか問い合わせます。
-c	(オプション) コールドバックアップを実行 フルバックアップと同様ですが、いったんデータベースを停止してから実行します。-o または -o -d が指定されている場合を除き、バックアップ終了後、データベースは再びオープンされます。
-o	(オプション) コールドバックアップ後、データベースを停止したままにする -c スイッチを使用した場合にのみ使用できます。コールドバックアップの終了後、データベースを停止したままにします。この機能は、ハードウェアをアップグレードする、あるいは新規サーバ (同じオペレーティングシステムが動作するもの) に移行する場合に有用です。詳しくは「596 ページの 「ハードウェアのアップグレードやサーバの交換」」を参照してください。

オプション	説明
-d	(オプション) コールドバックアップ後、DS エージェントを無効にする -c スイッチと -o スイッチの両方を指定した場合にのみ使用できます。コールドバックアップ後、DS エージェントを無効にします。この機能は、ハードウェアをアップグレードする、あるいは新規サーバ (同じオペレーティングシステムが動作するもの) に移行する場合に有用です。詳しくは「596 ページの 「ハードウェアのアップグレードやサーバの交換」」を参照してください。 擬似サーバの「login disabled」属性を設定することにより、DS エージェントを無効にします。その結果、OpenText eDirectory を起動しようとすると「-663」エラーが発生します。
-e *password*	NICI バックアップの実行 *password* は、NICI バックアップパスワードを指定します。この同じパスワードを NICI ファイルの復元でも指定する必要があります。
--config-file *configuration file*	(オプション) バックアップする OpenText eDirectory のインスタンスを指定できます。 *configuration file* は、バックアップする eDirectory インスタンスの環境設定ファイルへの絶対パスを指定します。次に例を示します。 --config-file /etc/opt/novell/eDirectory/conf/nds.conf このスイッチは、Linux 環境にのみ適用されます。
restore	データベースおよび関連ファイルの復元
-f *file_name*	(必須) バックアップファイルの名前とパス。 復元に使うフルバックアップファイルを指定します。このファイルは復元対象サーバ上に置いておかなければなりません。たとえば、restore -f :/backup/ndsbak.bak を指定すると、ファイル C:/backup/ndsbak.bak から復元されます。 複数のファイルに分かれている場合は、すべて同じディレクトリ内に集めておいてください。
-l *file_name*	(必須) ログファイルの名前とパス 復元処理の結果を出力するログファイルを指定します。
-r	(オプション)DIB セットも復元。 OpenText eDirectory データベースを復元する旨の指定です。 **警告 :** このオプションを指定しなかった場合、OpenText eDirectory データベース自身は復元されません。指定した種類以外のファイルのみが復元の対象になります。

オプション	説明
-d *dir_name*	(オプション) ロールフォワードログのあるディレクトリ ロールフォワードログを集めたディレクトリを指定します。復元対象サーバ上のフルパスで指定してください。必要なロールフォワードログをすべて、作成時と同じファイル名にして、ひとつのディレクトリに集めておかなければなりません。 バックアップファイルからの復元後、ロールフォワードログを使って、バックアップ時点以降の変更を反映させます。-d スイッチが使用されない場合、バックアップ時にロールフォワードログ記録を有効にしていても、バックアップツールはログファイルを参照しません。 最初に適用するべきロールフォワードログは、最新のバックアップファイルをテキストエディタで開き、backup タグの current_log 属性を見れば確認できます。ここでいう最新のバックアップファイルとは、-f オプションで指定するフルバックアップファイルか、または復元処理で適用することになる最後のインクリメンタルバックアップファイルです。ヘッダに記述される属性について詳しくは、「468 ページの 「バックアップファイルのヘッダ書式」」を参照してください。 **警告 :** バックアップファイルを開くときは、ヘッダを確認するだけにしてください。ファイルを保存または変更しようとすると、ファイルの一部が切り捨てられる場合があります。ほとんどのアプリケーションがバイナリデータを正しく保存することができません。
-u	(オプション) インクルードファイルに列挙されたファイルも復元 データベースに追加する形でバックアップしていたファイルも復元します。 バックアップの過程で、データベース以外にもバックアップが必要なファイルを列挙したファイルを作成し、インクルードファイルとして指定することもできます。しかしその場合でも、「-u」オプションで指定しなければ復元されません。
-a	(オプション) 検証後、DIB をアクティブにする指定。 復元後の検証処理が正常終了したら、データベース名を RST から NDS に変更します。この処理の概要については、「466 ページの 「バックアップツールによる復元作業の概要」」を参照してください。
-o	(オプション) 処理終了後、データベースをオープンする 処理の終了後にデータベースをオープンするようにバックアップツールに指示します。検証処理が正常終了すれば、データベースが自動的に開きます。失敗した場合は、復元前のデータベースが開きます。この処理の概要については、「466 ページの 「バックアップツールによる復元作業の概要」」を参照してください。
-s	復元処理後にロールフォワードログをリセットしないようにバックアップツールに指示します。主に、デフォルトの RFL の場所のインスタンスで使用されます。

オプション	説明
-n	(オプション) 復元後にデータベースを検証しない 検証せずにデータベースを復元するようにバックアップツールに指示します。このサーバの遷移ベクトルをレプリカリングに属する他のサーバと比較する、という検証処理を行いません。遷移ベクトルの詳細については、「474 ページの 「遷移ベクトルと復元後の検証処理」」を参照してください。他のオプションで明示的に指定されていない限り、RST から NDS への改名もしません。 **重要 :** OpenText サポートから提案されない限り、このオプションの使用はお勧めできません。
-v	(オプション) 上書きして復元 検証処理を行うことなく、データベース名を RST から NDS に変更します。 **重要 :** OpenText サポートから提案されない限り、このオプションの使用はお勧めできません。
-k	(オプション) データベースのロックを解除 NDS データベースのロックを解除します。
-i	順番になったインクリメンタルファイルのカンマ区切りリスト。
-e *password*	バックアップした NICI ファイルの復元 *password* は、NICI ファイルのバックアップで使用された NICI バックアップパスワードを指定します。NICI ファイルの復元時に誤ったパスワードが指定された場合は、エラーメッセージが表示されます。
--config-file *configuration file*	(オプション) 復元する OpenText eDirectory のインスタンスを指定できます。 *configuration file* は、復元する OpenText eDirectory インスタンスの環境設定ファイルへの絶対パスを指定します。次に例を示します。 --config-file /etc/opt/novell/eDirectory/conf/nds.conf このスイッチは、Linux 環境にのみ適用されます。
restadv	高度な復元機能。 **注 :** すべての高度な復元オプションで、DS エージェントがクローズされます。
-l *file_name*	(必須) ログファイルの名前とパス 復元処理の結果を出力するログファイルを指定します。

オプション	説明
-o	(オプション) 処理終了後、データベースをオープンする 処理の終了後にデータベースをオープンするようにバックアップツールに指示します。検証処理が正常終了すれば、データベースが自動的に開きます。失敗した場合は、復元前のデータベースが開きます。 この処理の概要については、「466 ページの 「バックアップツールによる復元作業の概要」」を参照してください。
-n	(オプション) 前に失敗した復元の検証処理を起動 前に復元して検証に失敗した RST データベースを再度検証します。
-m	(オプション) 復元された DIB ファイルの削除 RST データベースが存在すれば削除します。
-v	(オプション) 上書きして復元 検証処理を行うことなく、データベース名を RST から NDS に変更します。 **重要 :** OpenText サポートから提案されない限り、このオプションの使用はお勧めできません。
-k	(オプション) データベースのロックを解除 NDS データベースのロックを解除します。
-i	順番になったインクリメンタルファイルのカンマ区切りリスト。 **重要 :** このオプションは、DSBK にのみ適用されます。

getconfig	ロールフォワードログに関する現在の設定を表示。
	
	指定できるオプションはありません。
	
	現在の設定を表示します。たとえば、ロールフォワードログ記録が無効になっている場合、getconfig コマンドでは次のような情報が表示されます。
	
	`  Roll forward log status OFF`
	`Stream file logging status OFF`
	`Current roll forward log directory C:\rfl\nds.rfl`
	`Minimum roll forward log size (bytes) 104857600`
	`Maximum roll forward log size (bytes) 4294705152`
	`Last roll forward log not used 00000000.log`
	`Current roll forward log 00000001.log`
	`*** END ***`

setconfig	ロールフォワードログに関する設定の変更。

オプション	説明
-L	(オプション) ロールフォワードログ記録の有効化。 ロールフォワードログ機能を有効にします。(デフォルトでは無効)。この記録を有効にしておけば、停止する直前の状態にまでサーバを復元できるようになります。無効のままであれば、最後のフル / インクリメンタルバックアップ時点までしか復元できません。 レプリカリングに属するサーバについては、ロールフォワードログ記録を有効にして、他のサーバとの同期状態も復元できるようにしてください。 ただし管理者にとっては、監視しなければならない対象が増えます。これを怠ると、ログの容量は増える一方なので、ディスクパーティション / ボリュームがあふれてしまう恐れがあります。ディスク容量が不足してロールフォワードログを作成できない場合は、OpenText eDirectory はそのサーバに対して応答しなくなります。定期的にバックアップを取り、使わなくなったログは削除する必要があります。詳細については、479 ページの 「ロールフォワードログのバックアップと削除」を参照してください。 詳細については、475 ページの 「ロールフォワードログを使用する」を参照してください。
-l	(オプション) ロールフォワードログ機能の無効化。 ロールフォワードログ機能を無効にします (デフォルトでは無効) データベースでは連続したログを保存していくのをやめ、現在のロールフォワードログに上書きしていくようになります。ロールフォワードログ作成がオフの場合、最後にフル / インクリメンタルバックアップを実行した時点までしか OpenText eDirectory を復元できません。 誤って無効にしてしまった場合、ただちに有効にすると同時に、今障害が起こっても復元できるよう、改めてデータベースのバックアップを取ってください。 詳細については、475 ページの 「ロールフォワードログを使用する」を参照してください。
-T	(オプション) ストリームファイルのログ出力開始 (ロールフォワードログ機能が有効な場合のみ) ストリームファイルが更新された場合、その全体をロールフォワードログにコピーするようになります。ストリームファイルとは、ログインスクリプトなど、データベースに関係する追加の情報ファイルのことです。 ただしストリームファイルを記録すると、ディスクの空き容量が急速に減少します。ログの出力先ディスクパーティション / ボリュームの空き容量を、常に監視するようにしてください。ディスク容量が不足してロールフォワードログを作成できない場合は、OpenText eDirectory はそのサーバに対して応答しなくなります。

オプション	説明
-t	(オプション) ストリームファイルのログ出力停止 ストリームファイルが更新されても、その全体をロールフォワードログにコピーしないようになります。この場合でも、フル / インクリメンタルバックアップの際には、ストリームファイルもバックアップできます。ストリームファイルを頻繁に更新しないのであれば、この方法で充分でしょう。 ストリームファイルを記録しないと、ログファイルの容量が急速に増えるのを抑えることにもなります。
-r *dir_name*	(オプション) ロールフォワードログの出力先ディレクトリの設定。 ロールフォワードログの出力先ディレクトリを指定します。たとえば、setconfig -r vol2:\rfl というコマンドを実行すると、vol2:\rfl 以下にディレクトリが作成され、その下にログファイルができるようになります。 このディレクトリ名は現在の OpenText eDirectory データベース名に基づいて決まります。通常のインストールでは「NDS」なので、最終ディレクトリ名は vol2:\rfl\nds.rfl\ となります。ここで OpenText eDirectory データベース名を「ND1」に変更すると、保存先もこれに合わせて vol2:\rfl\nd1.rfl\ に変わります。 現在の保存先設定は getconfig コマンドで確認できます。 保存先の設定を変えるとその時点で新しいディレクトリができますが、ログファイルは実際にトランザクションが発生するまで作成されません。 **重要 :** バックアップツールでは、ロールフォワードログの保存先ディレクトリが変わったことを認識できません。データベースを復元する際には、最後のバックアップ以降のロールフォワードログをすべて、ひとつのディレクトリに集めておく必要があります。 詳細については、475 ページの 「ロールフォワードログを使用する」を参照してください。
-n *file_size*	(オプション) ロールフォワードログの最小容量の設定 ロールフォワードログの最小容量をバイト単位で指定します。この容量に達した後、実行中のトランザクションが終了すると、ログ出力先が新しいファイルに切り替わります。
-m *file_size*	(オプション) ロールフォワードログの最大容量の設定 ロールフォワードログの最大容量をバイト単位で指定します。この上限に達してもトランザクションが進行中の場合は、トランザクションは次のファイルに続けて記録されます。この設定は最小サイズの設定より常に大きくする必要があります。
-s	(オプション) ログ出力先ファイルの強制切り替え 実行中のトランザクションが終了した時点で、ログ出力先を新しいファイルに切り替えます。次のトランザクション開始時に新しいファイルが作成されます。

オプション	説明
キャンセル	バックアップ / 復元処理を取り消します。指定できるオプションはありません。 **注 :** このオプションは、DSBK に適用されません。
--config-file *configuration file*	(オプション) ロールフォワードログ設定を構成する OpenText eDirectory のインスタンスを指定できるようにします。 *configuration file* は、ロールフォワードログ設定を構成する OpenText eDirectory インスタンスの環境設定ファイルへの絶対パスを指定します。次に例を示します。 --config-file /etc/opt/novell/eDirectory/conf/nds.conf このスイッチは、Linux 環境にのみ適用されます。

cron ジョブとしての DSBK の実行

dsbk スクリプトには、DSTrace バイナリへのフルパスが含まれていません。そのため、デフォルト設定を使用して cron ジョブとしてスクリプトを実行すると、スクリプトが失敗します。ただし、パスを追加するように /opt/novell/eDirectory/bin/dsbk スクリプトを変更しないでください。これは、今後の OpenText eDirectory パッチによって、このファイルが上書きされ、スクリプトに対して行われたすべてのカスタマイズが元に戻される可能性があるためです。

代わりに、dsbk を cron ジョブとして実行する前に、crontab ファイル内の PATH 環境変数に ndstrace が配置されているディレクトリを追加します。そうすれば、cron ジョブが ndstrace アプリケーションを探して実行します。

NICI のバックアップと復元

NICI は、ファイルシステム内と、システムおよびユーザ固有のディレクトリやファイルに、キーとユーザデータを保存します。これらのディレクトリとファイルは、オペレーティングシステムによって提供されるメカニズムを使用して適切なアクセス権を設定することによって保護されます。この設定は、NICI インストールプログラムによって行われます。NICI のバックアップと復元は、ルートユーザに対してのみサポートされ、非ルートユーザに対してはサポートされません。

システムから NICI をアンインストールしても、システムまたはユーザ固有のディレクトリとファイルは削除されません。したがって、これらのファイルを以前の状態に復元することが必要になるのは、重大なシステム障害や人為的エラーから回復する場合のみです。既存の NICI ユーザディレクトリおよびファイルを上書きすると、既存のアプリケーションで問題が発生する可能性があることを理解しておくことが重要です。

DIB を開くためのデータベースキーは NICI キーでラップします。したがって、NICI のバックアップから独立して行った OpenText eDirectory のバックアップは役に立ちません。OpenText eDirectory のバックアップソリューション (DSBK と eMBox Backup) では、以下を可能にするスイッチ (-e) が使用されます。

1. OpenText eDirectory バックアップを実行中に NICI キーをバックアップします。
2. OpenText eDirectory 復元を実行中に NICI キーを復元します。

OpenText eDirectory のバックアップソリューションの詳細については、483 ページの「DSBK の使用」を参照してください。

NICI のバックアップ

NICI バックアップは、フル OpenText eDirectory バックアップまたはインクリメンタル OpenText eDirectory バックアップと一緒に実行できます。

NICI バックアップを実行するコマンドは次のとおりです。

dsbk backup -f file_name -l log_file_name -e password

-f と -l は、backup コマンドと一緒に使用すべき必須オプションです。

-e は、NICI ファイルをバックアップするためのスイッチです。

file_name は、バックアップツールで作成するバックアップファイルのファイル名と場所を指定します。

log_file_name は、バックアップ操作の結果を記録するために作成されるログファイルのファイル名と場所を指定します。

password は、NICI バックアップパスワードを指定します。パスワードは平文として指定できます。Linux では、パスワードをファイルとして渡すこともできます。この同じパスワードを NICI ファイルの復元でも指定する必要があります。

注 : NICI バックアップパスワードを -e スイッチと一緒に指定しなかった場合は、次のエラーメッセージが表示されます。

DSBK では :

```
Enter password along with the (-e) option!
DSBK error! 4
```

NICI の復元

1 NICI ファイルだけを復元します (DIB は復元しません)。

dsbk restore -f file_name -l log_file_name -e password

-f と -l は、restore コマンドと一緒に使用すべき必須オプションです。

-e は、NICI ファイルを復元するためのスイッチです。

file_name は、復元する情報を含むバックアップファイルのファイル名と場所を指定します。log_file_name は、復元操作の結果を記録するために作成されるログファイルのファイル名と場所を指定します。password は、NICI ファイルのバックアップで使用された NICI バックアップパスワードを指定します。NICI ファイルの復元時に誤ったパスワードが指定された場合は、エラーメッセージが表示されます。

2 ndsd サーバを再起動します。

3 DIB を復元します。

dsbk restore -f file_name -l log_file_name -a -r -o

-f と -l は、restore コマンドと一緒に使用すべき必須オプションです。

-a は検証後に DIB をアクティブにし、-r は DIB セットを復元し、-o は終了後にデータベースを開きます。

NICI バックアップがフルバックアップ中とインクリメンタルバックアップ中に実行され、フルバックアップとインクリメンタルバックアップで別々の NICI バックアップパスワードが使用された場合は、NICI ファイルを復元するときに、フルバックアップで使用されたパスワードを使用して NICI ファイルを復元する必要があります。

注 : パスワードを -e スイッチと一緒に指定しなかった場合は、次のエラーメッセージが表示されます。

DSBK では :

```
Enter password along with the (-e) option!
DSBK error! 4
```

NICI の復元中に誤ったパスワードが指定された場合は、次のエラーが表示されます。

```
NICI RESTORE: "NICI Files has not been restored(Check your parameters)"
Error!: -32
```

復元後の検証処理に失敗した場合の対処方法

復元処理の一環として検証を行います。復元された OpenText eDirectory データベースと、レプリカリングに属する他のサーバとの間で、遷移ベクトルを比較するというものです。復元プロセスの詳細については、「466 ページの 「バックアップツールによる復元作業の概要」」と「474 ページの 「遷移ベクトルと復元後の検証処理」」を参照してください。

遷移ベクトルが合致しなければ、検証に失敗したことになります。これは一般に、復元処理に使ったファイルのデータが不足していたことを表します。たとえば次のような状況が考えられます。

- 最後にバックアップを実施した後、ロールフォワードログ機能を有効にしていなかった場合。
- ロールフォワードログを取っていたのに、復元の際これを使わなかった場合。
- ロールフォワードログが不足していた場合。

デフォルトでは、復元した OpenText eDirectory データベースが他のレプリカと整合が取れていない場合、そのままではオープンできません。

バックアップファイルやロールフォワードログの指定を単に忘れただけであれば、正しく指定してもう一度復元処理を実行すればよいはずです。次回に復元が成功すれば、検証に成功し、データベースがオープンされるでしょう。

必要なバックアップファイルやロールフォワードログが揃っていない場合は、以下の手順でサーバを復旧してください。検証に失敗したときの復旧手順の概要を説明します。

- サーバの識別情報やファイルシステム権利は、バックアップファイルなどが不足していても復旧できるはずです。
- バックアップからレプリカを復元できない場合でも、サーバとしては動作します。新しいレプリカが追加されたものとして扱われ、他のサーバにも影響を及ぼしてしまうのです。したがって、レプリカリングからいったんサーバを外し、高度な復元機能や DSRepair ツールを使って元の状態に復旧してから、改めてレプリカリングに追加してください。
- このサーバにしかデータがない、すなわち他のサーバにレプリカが作られていないパーティションについては、残念ながら復元できません。

検証に失敗した後、このセクションの説明に従ってサーバの識別情報やファイルシステム権利を復旧し、レプリカリングからいったん外し、再び追加します。ｋ この手順でレプリケーションが終了すれば、サーバは元どおりに機能するようになるはずです。ただしレプリカを作っていなかったために復元できなかったパーティションを除きます。

まず、「504 ページの 「レプリカリングをクリーンアップする」」に従って作業してください。それが終了したら 506 ページの 「サーバの復旧とレプリカの再追加」に進みます。

レプリカリングをクリーンアップする

この手順では、次の方法について説明します。

- **マスタレプリカの再割り当て。**障害が発生したサーバが、あるパーティションのマスタレプリカを保持していた場合は、DSRepair を使って、レプリカリストに属する他のサーバ上のレプリカを、マスタとして扱うよう指定します。
- **障害が発生したサーバに対するレプリカリストの参照の削除。**障害が発生したサーバを含むレプリカリングのメンバーである各サーバに対して、障害が発生したサーバが利用できなくなったことを通知する必要があります。

前提条件

- ❒ OpenText eDirectory そのものは、当該サーバに正常にインストールされているものとします。
- ❒ 復元を試み、検証処理で失敗したことが前提です。
- ❒ OpenText eDirectory データベースは稼働しており、(復元処理により作成された)RST データベースもマシンに残っているものとします。
- ❒ どのパーティションのレプリカがこのサーバに保持されていたか、は分かっているものとします。このサーバのレプリカはバックアップファイルのヘッダ部に記録されています。

手順

レプリカリングをクリーンアップするには、次の操作を行います。

1 DSRepair を起動します。復元対象サーバとレプリカを共有しているサーバのコンソールから、次のオプションを指定して起動してください。

- **Windows:** -a スイッチを使用します。
- **Linux:** -Ad スイッチを使用します。

-a または -Ad スイッチを使用した詳細オプションで DSRepair を実行する方法については、「364 ページの 「DSRepair オプション」」を参照してください。

警告 : -a または -Ad を指定して DSRepair を使用する場合は、詳細オプションによってはツリーが破損することがあります。

---

2 [**レプリカ操作とパーティション操作**] を選択します。

3 編集したいパーティションを選択します。このパーティションが属するレプリカリングから、障害の起こったサーバを外すことになります。

4 [**レプリカリングの表示**] を選択して、このパーティションに関するレプリカを持つサーバのリストを表示します。

5 (状況によって実行) 当該サーバにマスタレプリカがあった場合、[**このサーバを新しいマスタレプリカに設定**] コマンドで、他のサーバにマスタを切り替えます。

この時点で、対象のレプリカリングは新しいマスタレプリカを保持しています。リングを構成するすべてのレプリカに対して、新しいマスタが存在することが通知されます。

6 そのまましばらく待ちます。上記のレプリカがマスタになったことが他のサーバに認識されるまで、しばらく待ちます。

7 [**レプリカリングの表示**] に戻ります。障害が発生したサーバの名前を選択して、[**レプリカリングからのサーバの削除**] を選択してください。

DSRepair を起動する際に詳細オプションで -a または -Ad(プラットフォームにより選択) を指定していなかった場合、このコマンドは表示されません。

警告 : 障害の起こったサーバをマスタレプリカに設定したままで、このコマンドを実行しないでください。リング内のサーバリストを見れば確認できます。マスタレプリカであれば、「ステップ 5」を参照して、他のサーバにマスタを切り替えてから、当該サーバをレプリカリングから外します。

---

8 管理者としてログインします。

9 説明メッセージを読み、それに対する同意を入力して処理を続行します。

10 DSRepair を終了します。

レプリカリングを構成するすべてのサーバに通知が行われます。

11 障害が発生したサーバを含んでいる各レプリカリングごとに、1 つのサーバ上で、この手順を繰り返します。

続いて、障害が生じたサーバ上に、新たにレプリカを構築します。「506 ページの 「サーバの復旧とレプリカの再追加」」に進んでください。

サーバの復旧とレプリカの再追加

レプリカ設定を「外部参照」側に書き替え、自分自身はレプリカリングに属していないものとして動作するようにします。その上でサーバからレプリカを削除すると、データベースのロックを解除できるようになります。

レプリカを削除すれば、レプリカをサーバに再追加する作業は終わりです。あとは自動的に、他のサーバから各レプリカの最新版を参照し、再追加していきます。各レプリカが再追加されたら、サーバは元と同じように機能するはずです。

DSRepair を使用してレプリカを削除し、レプリケーション機能により再追加する手順を次に示します。

1 504 ページの 「レプリカリングをクリーンアップする」が完了していることを確認します。

2 上書き復元コマンドを、次のように実行します。併せてログファイル名も指定してください。

dsbk restadv -v -l *logfilename*

この詳細復元オプションにより、RST データベース (復元したが検証に失敗したもの) の名前を NDS に変更します。ただしロックは解除しません。

3 サーバコンソールで、DSRepair の高度な復元機能により、レプリカ設定をすべて外部参照に切り替えます。

- **Windows:** [スタート] > [設定] > [コントロールパネル] > [OpenText eDirectory サービス] の順にクリックします。[dsrepair.dlm] を選択します。[起動パラメータ] フィールドに、「-XK2 -rd」と入力します。[Start] をクリックします。
- **Linux:** 次のコマンドを入力します。

```
ndsrepair -P -Ad
```

-rd または -P スイッチは、ローカルデータベースとレプリカを修復します。

警告 : DSRepair による高度な復元機能は、正しく使用しないとツリーが破損することがあります。

---

4 修復が終了したら、ロックを解除してデータベースを開きます。次のように実行してください。

```
dsbk restadv -o -k -l logfilename
```

-o はデータベースのオープン、-k はロック解除を表します。

5 OpenText Identity Console を使って、修復されたサーバをレプリカリングに再追加します。

5a OpenText Identity Console のホームページで、[パーティション管理] タイルをクリックします。

5b [パーティション] ページで、[タイプ] ドロップダウンメニューをクリックし、[サーバ] を選択して、[検索] をクリックします。

5c 複製するパーティションを選択します。

5d [レプリカビュー] ページで、[レプリカの追加 +] をクリックします。

5e［パーティション名 ］をクリックして、復元したサーバを選択します。

5f 必要なレプリカのタイプを選択して、［OK］をクリックしてから、［完了］をクリックします。

5g このサーバが属していた各レプリカリングについて、上記の操作を繰り返します。

6 レプリケーション処理が終わるまでしばらく待ちます。

レプリカの状態が［新規］から［オン］に変われば、レプリケーション処理は終了です。状態は OpenText Identity Console で確認できます。詳細については、172 ページの「レプリカに関する情報を表示する」を参照してください。

7 NICI セキュリティファイルを復元するには、最初に、NICI ファイルだけを復元してから、NDSD サーバを再起動して、DIB を復元します。

8 (状況によって実行) このサーバでロールフォワードログ機能を使うためには、改めて有効に切り替え、障害対策のための書き出し先も設定し直して、ロールフォワードログの環境設定を再作成する必要があります。ロールフォワードログを有効にしてから、改めてフルバックアップも取る必要があります。

この手順が必要となるのは、復元処理の過程で、ロールフォワードログに関する設定はデフォルトに戻るためです。つまり、ロールフォワードログ機能は無効となり、保存先もデフォルトの場所になるからです。フルバックアップが改めて必要となるのは、スケジュールに従って次に無人でのフルバックアップが取られるまでに、再び障害が起こる可能性があるためです。

ロールフォワードログの詳細については、「475 ページの 「ロールフォワードログを使用する」」を参照してください。

バックアップ / 復元の運用例

- 508 ページの 「シナリオ : 単一サーバ構成のネットワークで、OpenText eDirectory を格納しているハードディスクが故障した場合」
- 509 ページの 「シナリオ : 複数サーバ構成のネットワークで、OpenText eDirectory を格納しているハードディスクが故障した場合」
- 511 ページの 「シナリオ : 複数サーバ構成のネットワークで、1 台のサーバが完全に使えなくなった場合」
- 512 ページの 「シナリオ : 複数サーバ構成のネットワークで、数台のサーバが使えなくなった場合」
- 512 ページの 「シナリオ : 複数サーバ構成のネットワークで、すべてのサーバが使えなくなった場合」

シナリオ : 単一サーバ構成のネットワークで、OpenText eDirectory を格納しているハードディスクが故障した場合

あるユーザは Stationery Supply 社で単一サーバ構成のネットワークを管理しています。サーバは 1 台しかないので、耐障害性を得るためにレプリケーションに頼ることはできません。その場合は、バックアップツール機能を使えば、OpenText eDirectory のバックアップ / 復元処理が容易になります。この機能は、サーバに特化されているため、高速です。

eDirectory 8.7.3 以降のバージョンで、バックアップツールを実行するためのバッチファイルを使用してサーバの無人バックアップをセットアップしました。

毎週日曜の夜にフルバックアップを取り、さらに平日は毎晩、インクリメンタルバックアップを取るようにしました。また、eDirectory の無人バックアップのすぐ後に、ファイルシステムのフル / インクリメンタルバックアップを取るようにします。バックアップテープには、ファイルシステムデータのほか、OpenText eDirectory の最新バックアップも保存されることになります。さらに、データ保存サービス会社と契約して、バックアップテープを社外に保管することにしました。

毎週月曜日の朝、バックアップログを調べて、正常にフルバックアップが取れていることを確認します。また、普段から時々、インクリメンタルバックアップの状況をログで調べています。

ロールフォワードログの機能は無効にしています。それは次のような理由によります。

- サーバには独立した記憶デバイスがついていないので、ロールフォワードログを有効にしても OpenText eDirectory のバックアップとしては役に立ちません。記憶デバイスが故障すれば、OpenText eDirectory ばかりでなくログも消えてしまうので、いずれにしても復元には使えません。
- ツリー構成が変わることはあまりありません。また、障害が発生しても、前の晩の状態にまで復元できれば充分だと考えています。停止直前の状態まで OpenText eDirectory を復元できなくても構いません。
- サーバはレプリカリングに属して他のサーバと連携しているわけではないため、ロールフォワードログがなくても、復元後の検証処理には成功するはずです。

Stationery Supply 社では、人事異動などによりツリー構成を大きく変える場合は、その直前と直後に、手動でバックアップすることにしています。日曜日以外でも、必要に応じて臨時のバックアップを取ろうというわけです。これがロールフォワードログに代わる措置です。

必要になればいつでもバックアップ作業ができるよう、時々テストして確かめています。テスト用にもう 1 台サーバを購入する予算はないので、市内にあるサービス会社と契約し、必要なときだけサーバを使わせてもらえるようにしました。実機と似た構成のサーバにオペレーティングシステムをインストールし、OpenText eDirectory データベースの環境もできるだけ同じように構築します。このサーバに、実機から取ったバックアップファイルを使って復元し、OpenText eDirectory が想定どおりに復元されていることを確認するのです。

ある水曜日の朝、OpenText eDirectory が格納されたハードディスクの故障が見つかりました。そこで新しいハードディスクを手配したほか、日曜日に取ったフルバックアップファイル、月曜日および火曜日に取ったインクリメンタルバックアップファイルを用意しまし

た。次に、ハードディスクを交換し、OpenText eDirectory をインストールしました。その上に、フル / インクリメンタルバックアップファイルから復元しました。水曜日の朝、故障が起こる前にツリー構成を変更しましたが、これは復元できませんでした。ロールフォワードログを残しておかなかったからです。ただし、昨夜のバックアップに復元できれば十分です。ロールフォワードログを実行することによって管理オーバーヘッドが賄えるとは思っていません。

シナリオ : 複数サーバ構成のネットワークで、OpenText eDirectory を格納しているハードディスクが故障した場合

あるユーザが Outdoor Recreation 社で OpenText eDirectory が稼働する 10 台のサーバを管理しています。毎週日曜の夜にフルバックアップを取り、さらに平日は毎晩、インクリメンタルバックアップを取っています。OpenText eDirectory のバックアップ後すぐに、ファイルシステムバックアップによりテープに保存しています。

すべてのサーバがレプリカリングに参加しています。ロールフォワードログ機能は全サーバで有効です。また、その保存先は、OpenText eDirectory とは別の記憶デバイスに割り当てています。ディスクの空き領域やアクセス権は随時監視して、ストレージデバイスがロールフォワードログでいっぱいにならないよう注意しています。使用済みのロールフォワードログは、時々テープにバックアップして OpenText eDirectory で使用しているもの以外を削除し、空き容量を増やすようにしています。

もちろんロールフォワードログ記録を有効にすると管理の手間が増えますが、それを上回る利点があると考えています。サーバがレプリカリングに属している場合、いつでも最新の状態をバックアップしておく必要があるからです。こうしておけば、障害が発生しても、他のサーバとの同期状態も含めて元どおりに復元できます。

さらに、テスト環境で定期的にバックアップファイルからの復元を試み、想定どおりに動作することを確認しています。

ある木曜日の午後 2 時、Inventory_DB1 という Linux サーバの、OpenText eDirectory を格納しているハードディスクが故障しました。

そこでまず、最新のフルバックアップファイルとそれ以降のインクリメンタルバックアップファイルを集め、昨晩午前 1 時の状態にまでデータベースを復元します。ロールフォワードログは昨夜のバックアップ以降の変更を記録しています。そこで次に、午前 1 時以降のロールフォワードログを使って、故障が起こる直前の状態に戻さなければなりません。

次のような手順で作業を進めることにしました。

1. まず、サーバのハードディスクを交換します。
2. 日曜の夜に取ったフルバックアップのテープを用意しました。

   フルバックアップ用のバッチファイルは、毎週日曜の夜にファイル /adminfiles/backup/backupfull.bk にバックアップするようになっています。

   ファイルの容量制限を 200MB と設定していたので、次の 2 つのファイルがありました。

   backupfull.bk.00001 (250MB)

   backupfull.bk.00002 (32MB)

3. さらに、月曜、火曜、水曜に取ったインクリメンタルバックアップのテープも用意しました。

   インクリメンタルバックアップ用のバッチファイルは、平日に毎晩、ファイル /adminfiles/backup/backupincr.bk にバックアップするようになっています。

   平日の晩は毎日、OpenText eDirectory のインクリメンタルバックアップに同じバッチファイルを使っているので、ファイル名は常に同じです。しかし、テープからサーバにコピーする時に、ファイル名を変更しなければなりません。復元処理の際は、全ファイルを同じディレクトリに集めておく必要があるからです。

4. まず、ハードディスクを交換しました。

   幸い Linux オペレーティングシステムを格納したハードディスクは故障しなかったので、Linux の再インストールは必要ありませんでした。

5. バックアップテープを使って、障害を受けたディスクパーティションを復元しました。
6. 次に eDirectory を再インストールし、仮のツリーを作りました。復元処理では、いったんこのツリー上にデータを復元してから実動ツリーに切り替えることになります。
7. サーバ上に、バックアップファイルを集めておくためのディレクトリ /adminfiles/restore を作りました。
8. フルバックアップファイル (2 つに分割されているもの) を、このディレクトリにコピーしました。
9. さらに、月曜、火曜、水曜のインクリメンタルバックアップファイルもコピーしました。

   どれも同じ backupincr.bk というファイル名なので、次のように名前を変更します。

   backupincr.mon.bk

   backupincr.tues.bk

   backupincr.wed.bk

   注 : インクリメンタルバックアップファイルは、必ずしもフルバックアップファイルと同じディレクトリにコピーしなくても構いません。ただし各インクリメンタルバックアップファイルはすべて同じディレクトリに置いておく必要があります。

   ---

10. EmBox を使って、OpenText eDirectory を復元します。
    a. 次のコマンドでサーバにログインします : edirutil -i
    b. 次のコマンドを実行します : login -s 10.71.39.10 -u admin.novell -p 8028 -w novell -n
    c. 次のコマンドを実行します : restore -r -a -o -f /embox/backup5000.bak -d /var/opt/novell/eDirectory/data/dib/nds.rfl -l /embox/backupNew.log
    d. ［復元ウィザード - オプション］画面で次のように操作しました。

       ［**データベースを復元**］チェックボックスをオン。

       ［**ロールフォワードログの復元**］チェックボックスをオン。

       ロールフォワードログの保存先を入力。

       これは普段ロールフォワードログを保存している場所です。OpenText eDirectory とは別のハードディスクなので、今までのロールフォワードログが残っているはずです。

       ［**セキュリティファイルの復元**］チェックボックスをオン。

［**復元されたデータベースを検証後にアクティブにします**］チェックボックスをオン。

［**復元の完了後にデータベースを開く**］チェックボックスをオン。

これは、復元後の検証に成功したら OpenText eDirectory をオープンするための指定です。

11. 復元処理を起動し、インクリメンタルバックアップファイル名を問い合わせるメッセージが表示されたので入力しました。
12. 復元後の検証処理にも成功し、自動的にデータベースがオープンされ、従来どおりのツリーで動作するようになりました。

    ロールフォワードログはそのまま残っており、その場所を正しく指定したため、停止直前の状態に復元でき、検証も成功しました。
13. 復元後、ロールフォワードログに関する設定をやり直し、改めてフルバックアップを取っておきました。

    ロールフォワードログの機能は、復元処理の過程で無効に戻ってしまうため、ここで有効にする必要があります。フルバックアップが改めて必要となるのは、スケジュールに従って次に無人でのフルバックアップが取られるまでに、再び障害が起こる可能性があるためです。

サーバの稼動状況を調べ、正常に動作していることを確認しました。

シナリオ : 複数サーバ構成のネットワークで、1 台のサーバが完全に使えなくなった場合

ユーザは GK Designs 社で 15 台のサーバを管理しています。毎週土曜の夜にフルバックアップを取り、さらに平日は毎晩、インクリメンタルバックアップを取っています。OpenText eDirectory のバックアップ後すぐに、ファイルシステムバックアップによりテープに保存しています。

すべてのサーバがレプリカリングに参加しています。ロールフォワードログ機能は全サーバで有効です。

ある日、漏電による火事のため、ある支店のサーバが 1 台、完全に使えなくなってしまいました。幸い、このサーバのパーティションは、ひとつを除いてすべて、他のサーバにレプリカが作成されていました。ロールフォワードログ機能は有効にしていましたが、それも使えなくなってしまいました。したがって、停止直前の状態にまで OpenText eDirectory データベースを復元することはできません。

しかしバックアップファイルがあるので、サーバの OpenText eDirectory 識別情報は再作成できます。復元にロールフォワードログは使えないので、他のサーバが想定している同期状態と合致しません (「474 ページの 「遷移ベクトルと復元後の検証処理」」を参照)。そのため復元の検証処理が失敗します。つまり、デフォルトで、復元処理が終わっても eDirectory データベースが開きません。

そこで、レプリカリングからいったんこのサーバを外し、DSRepair を使って、サーバの古くなったレプリカ情報をすべて外部参照に変え、それから最新のレプリカを保持しているサーバから複製することによって、このサーバに各パーティションの新しいコピーを追加しなおしました。具体的な手順は「503 ページの 「復元後の検証処理に失敗した場合の対処方法」」を参照してください。

レプリカが作られていなかったパーティションがひとつありました。これは、支店内にあるファックス / プリンタ複合機や大判カラープリンタなど、この支店内のネットワーク印刷オブジェクトを管理しているコンテナです。他のサーバにレプリカがないため、このパーティションは上述の手順では復元できません。このパーティションのオブジェクトは一から作り直さざるをえなかったので、将来に備え、これも他のサーバにレプリカを作っておくことにしました。

そこでロールフォワードログに関する設定をやり直し、改めてフルバックアップを取っておきました。ロールフォワードログの機能は、復元処理の過程で無効に戻ってしまうためです。

シナリオ : 複数サーバ構成のネットワークで、数台のサーバが使えなくなった場合

ユーザは 3ヶ所に分けて設置された 20 台のサーバを管理しています。その 1ヶ所で、水道管破裂による水漏れ事故のため、8 台のうち 5 台が使えなくなってしまいました。

幸い、どのサーバについても、OpenText eDirectory のバックアップを取っていました。しかし、すべてのサーバがレプリカリングに属しており、ロールフォワードログはなくなってしまったので、これを使わずにすべてをツリーに戻さなければなりません。最初にどのサーバから着手し、レプリカ間の不整合をどのように解消すればよいか、判断できませんでした。状況が複雑であるため、復元方法を決めるのに OpenText のサポート部門に相談することにしました。

シナリオ : 複数サーバ構成のネットワークで、すべてのサーバが使えなくなった場合

ユーザとそのチームは、Human Resource Consulting 社で 1ヶ所に設置された 50 台のサーバを管理しています。

普段から障害対策のため、ツリーの各パーティションについて 3 つずつレプリカを作成しています。したがって、サーバが 1 台停止しても、そのパーティションにあるオブジェクトは他のサーバからアクセスできます。さらに、サーバを個別に復元できるよう、バックアップツールで定期的にすべてのサーバをバックアップし、ロールフォワードログ記録を有効にし、そしてバックアップテープは別の建物に保管するように計画しました。

災害に備えるため、チームでは 2 台のサーバを DSMASTER として割り当てています。2 台を割り当てているのは、ツリーが大きすぎて、1 台では全パーティションのレプリカを保持できないからです。ツリーに属するどのパーティションも、いずれかの DSMASTER サーバにレプリカが作成されています。逆に、同じパーティションのレプリカが 2 台の DSMASTER サーバに作成されることはないようにして、重複を避けています。これが災害対策として重要な点です。

さらにチームでは、テスト環境で定期的にバックアップファイルからの復元を試み、想定どおりに動作することを確認しています。

ある日、台風で社屋が倒壊し、データセンタにあったサーバもすべて壊れてしまいました。

台風が去った後、チームではまず、パーティションすべてのレプリカを保持する、2 台の DSMASTER サーバを復元する作業に取りかかりました。最新のフルバックアップファイルおよびそれ以降のインクリメンタルバックアップファイルを使いましたが、ロールフォワードログは使いませんでした。サーバが壊れたときに、一緒になくなってしまったからです。DSMASTER サーバを設定する際、この 2 台が同じレプリカを共有することはないようにしていました。そのため、ロールフォワードログを適用しなくても、復元後の検証処理は問題なく成功したのです。DSMASTER サーバが復元されたことにより、Human Resources Consulting 社のツリーに属するオブジェクトはすべてアクセスできるようになりました。

DSMASTER が復元できれば、整合性を損なうことなくツリー全体を再作成できるので、その役割は非常に重要です。

このチームは、ロールフォワードログの機能も有効にしていました。障害が発生する直前の状態にまでサーバを復元でき、レプリカリングに属する他のサーバとの同期状態に不整合が生じないからです。サーバが復旧し、他のサーバと通信できるようになると、停止していた間に行われた更新情報を自動的に受け取り、同期を取ることができます。

しかし今回はそのロールフォワードログを使えません。その場合、レプリカリングに属するサーバのうち、最初に復元処理を実施したものしか正常に復元できないことになります。それ以外のサーバは、他のサーバと同期状態が一致しないということで、復元後の検証処理に失敗してしまうのです (「474 ページの 「遷移ベクトルと復元後の検証処理」」を参照)。検証処理に失敗すると、この OpenText eDirectory データベースはアクティブになりません。

しかしこのチームは、このような状況もきちんと考慮していました。2 台の DSMASTER サーバを設定し、重複して保持するパーティションがないようにして、これを復元作業の起点としたのです。重複がないので検証処理に失敗することはありません。これをマスタとして順次他のサーバにコピーしていけば、レプリカリング全体が復元できることになります。

DSMASTER サーバの復元後、他のサーバを復元するために、いくつか必要な作業があります。このチームは次のような手順で、他のサーバを順次復元していきました。

- DSMASTER サーバ上のレプリカを、マスタレプリカとして割り当てます。
- DSMASTER 以外のサーバをレプリカリングから外します。
- フル / インクリメンタルバックアップファイルを使って、DSMASTER 以外のサーバを復元します。

  ロールフォワードログがないため、復元後の検証処理に失敗することは分かっています。したがって、データベースは復元されても、まだアクティブになっていません。
- 高度な復元機能を使って、データベースをアクティブにしました。ロックはまだ解除しません。
- DSRepair を使って、他のサーバを参照してレプリカを作る設定に変更しました。
- データベースのロックを解除しました。

  この時点で、各サーバの識別情報は元どおりに戻っていますが、レプリカ情報を同期させようとはしません。他のサーバからレプリカのデータを受け取り、改めて構築できる状態になっています。

- 各サーバは DSMASTER サーバからデータを受け取り、自動的にレプリカを復元してレプリカリングに復帰します。

  各サーバにどのレプリカが置かれていたか、チームではきちんと把握していました。もっとも、最終バックアップ時点の状況は、バックアップファイルのヘッダを見れば確認できます。
- ロールフォワードログに関する設定をやり直し、改めてフルバックアップを取っておきました。ロールフォワードログの機能は、復元処理の過程で無効に戻ってしまうので、ここで有効にしたのです。改めてフルバックアップを取る必要があるのは、スケジュールに従って次に無人でのフルバックアップが取られるまでに、再び障害が起こる可能性があるためです。

具体的な手順は「503 ページの 「復元後の検証処理に失敗した場合の対処方法」」を参照してください。

かなりの作業量でしたが、ツリー自身は比較的早期に使用できるようになり、サーバ全体の OpenText eDirectory 識別情報を復旧する目処も立っていました。

DSBK を使用した障害復旧計画

障害復旧計画を使用すれば、ディスクを破損した時点の環境設定に戻すことができます。オペレーティングシステムが破損した場合でもサーバを回復できるように、サーバのディスクをリモートの場所にバックアップしておく必要があります。

このセクションでは、OpenText eDirectory サーバの障害復旧計画の例を示します。

- 514 ページの 「Linux 上での障害復旧計画」
- 516 ページの 「Windows 上での障害復旧計画」

Linux 上での障害復旧計画

サーバのディスクのバックアップを取得するには :

1 DSBK を次のように設定します。

  1a /etc でファイル dsbk.conf を作成します。

  1b 一時ファイルを作成します。例 : /tmp/dsbk.tmp

  1c 前のステップで作成した一時ファイルの場所を /etc/dsbk.conf ファイルの中で指定します。

2 サーバのディスクを読み書きモードでリモートマシンにマウントして、リモートマシンのディスク上のすべてのバックアップファイルを保存します。

  たとえば、「eDirServer# mount <remote machine IP>:/home/backup/ /mnt/dsbkBkp」と入力します。

3 次のコマンドを使用してカスタムバックアップの場所を設定します。

```
dsbk setconfig -L -T -r /mnt/dsbkBkp
```

注 : サーバ上の /opt/novell/eDirectory/bin から、DSBK が実行することを確認します。

4 NICI と一緒にフルバックアップをリモートの場所のファイルシステムに取得します。

```
dsbk backup -f <backup file location> -l <log file location> -e
<password for NICI backup> -t -b
```

たとえば、「dsbk backup -f /mnt/dsbkBkp/fb1.bak -l /mnt/dsbkBkp/fb1.log -e novell -t -b」と入力します。

注 : -e オプションは、NICI のバックアップに使用します。例では、novell が NICI バックアップ用のパスワードです。独自のパスワードを選択した場合は、NICI の復元でも同じパスワードを使用する必要があります。

---

5 次のコマンドを使用して、インクリメンタルバックアップを取得します。

```
dsbk backup -f <incremental backup file location> -l <incremental log
file location> -t -i
```

次に例を示します。

1 日目 : dsbk backup -f /mnt/dsbkBkp/ib1.bak -l /mnt/dsbkBkp/ib1.log -t -i

2 日目 : dsbk backup -f /mnt/dsbkBkp/ib2.bak -l /mnt/dsbkBkp/ib2.log -t -i

---

注 : インクリメンタルバックアップを取得中に、NICI をバックアップする必要はありません。

OpenText eDirectory サーバが破損した場合は、次の手順を実行して、リモートの場所のバックアップを使用して OpenText eDirectory サーバを復旧します。

1 オペレーティングシステムが破損した場合は、オペレーティングシステムをインストールし直します。

2 OpenText eDirectory だけが破損した場合は、OpenText eDirectory RPM を削除することによって、eDirectory 用のシステムのクリーンアップを実行します。

3 前と同じ OpenText eDirectory をインストールして、1 つのサーバダミーツリーを設定します。次に例を示します。

ndsconfig new -t dummy_bkp_tree -n novell -a admin.novell -w novell

4 フルバックアップファイルから NICI を復元します (-d、-r、-a、-o オプションは使用しません)。

dsbk restore -f *<backup file location>* -l *<log file location>* -e *<password used to NICI backup>*

たとえば、「dsbk restore -f /mnt/dsbkBkp/fb1.bak -l /mnt/dsbkBkp/restore1.log -e novell」と入力します。

5 NICI の復元後に、OpenText eDirectory サーバを再起動します。

6 フルバックアップファイルとインクリメンタルバックアップファイルの両方を復元します。次に例を示します。

dsbk restore -f /mnt/dsbkBkp/fb1.bak -l /mnt/dsbkBkp/restore2.log -d /mnt/dsbkBkp/nds.rfl/ -r -a -e novell -o -i /mnt/dsbkBkp/ib1.bak, /mnt/dsbkBkp/ib2.bak

backup コマンドと restore コマンドの詳細については、「484 ページの 「Linux 上で DSBK を使用する」」を参照してください。

Windows 上での障害復旧計画

サーバのディスクのバックアップを取得するには :

1 サーバのディスクを読み書きモードでリモートマシンにマップします。たとえば、「O:\dsbkBkp」と入力します。

2 DSBK コマンドを実行するには :

2a NDScons.exe を実行して、OpenText eDirectory サーバコンソールを開きます。

2b ［**サービス**］タブで、［**dsbk.dlm**］をクリックします。

2c ［**起動パラメータ**］フィールドに、DSBK コマンドを入力します。

3 次のコマンドを使用してカスタムバックアップの場所を設定します。

setconfig -L -T -r O:\dsbkBkp

4 NICI と一緒にフルバックアップをリモートの場所のファイルシステムに取得します。

backup -f *<backup file location>* -l *<log file location>* -e *<password for NICI backup>* -t -b

---

注 : -e オプションは、NICI のバックアップに使用します。例では、novell が NICI バックアップ用のパスワードです。独自のパスワードを選択した場合は、NICI の復元でも同じパスワードを使用する必要があります。

---

5 次のコマンドを使用して、インクリメンタルバックアップを取得します。

backup -f *<incremental backup file location>* -l *<incremental log file location>* -t -i

次に例を示します。

1 日目 : backup -f O:\dsbkBkp\ib1.bak -l O:\dsbkBkp\ib1.log -t -i

2 日目 : backup -f O:\dsbkBkp\ib2.bak -l O:\dsbkBkp\ib2.log -t -i

---

注 : インクリメンタルバックアップを取得中に、NICI をバックアップする必要はありません。

OpenText eDirectory サーバが破損した場合は、次の手順を実行して、リモートの場所のバックアップを使用して OpenText eDirectory サーバを復旧します。

1 オペレーティングシステムが破損した場合は、オペレーティングシステムをインストールし直します。

2 OpenText eDirectory だけが破損した場合は、OpenText eDirectory 用のシステムのクリーンアップを実行します。

3 前と同じ OpenText eDirectory をインストールして、1 つのサーバダミーツリーを設定します。

4 フルバックアップファイルから NICI を復元します (-d、-r、-a、-o オプションは使用しません)。

次に例を示します。

restore -f *<backup file location>* -l *<log file location>* -e *<password used for NICI backup>*

たとえば、「restore -f O:\dsbkBkp\fb1.bak -l O:\dsbkBkp restore1.log -e novell」と入力します。

5 NICI の復元後に、OpenText eDirectory サーバを再起動します。

6 フルバックアップファイルとインクリメンタルバックアップファイルの両方を復元します。

次に例を示します。

restore -f O:\dsbkBkp\fb1.bak -l O:\dsbkBkp\restore2.log -d O:\dsbkBkp\nds.rfl\ -r -a -e novell -o -i O:\dsbkBkp\ib1.bak, O:\dsbkBkp\ib2.bak

backup コマンドと restore コマンドの詳細については、「486 ページの 「Windows で DSBK を使用する」」を参照してください。

LDAP ベースのバックアップ

LDAP ベースのバックアップ機能を使用すると、1 回につき 1 つのオブジェクトの属性と属性値がバックアップされます。

次の表に、この機能をサポートするプラットフォームを示します。

機能	Linux	Windows
LDAP ベースのバックアップ	✓	✓

この機能を使用すれば、変更が加えられている場合にだけオブジェクトをバックアップする、インクリメンタルバックアップを実行できます。

LDAP ベースのバックアップでは、LDAP 拡張オペレーションを通じて、LDAP Libraries for C によって提供される OpenText eDirectory オブジェクトのバックアップ / 復元用インタフェースを使用できます。

LDAP Libraries for C SDK の詳細については、LDAP Libraries for C のマニュアル (https://www.microfocus.com/documentation/edirectory-developer-documentation/ldap-libraries-for-c/#:~:text=LDAP%20Libraries%20for%20C%20enables,draft%20proposed%20to%20the%20IETF) を参照してください。

LDAP を使用して OpenText eDirectory オブジェクトのバックアップと復元を行う方法の例については、backup.c のサンプルコード (https://www.novell.com/documentation/developer/samplecode/cldap_sample/cldap_sample/extensions/backup.c.html) を参照してください。

LDAP ベースのバックアップの必要性

LDAP ベースのバックアップは、現在のバックアップと復元を使用して問題の解決を試みます。

この機能で解決される問題には次のようなものがあります。

- サードパーティのバックアップアプリケーションまたは開発者が使用して、サポートされるすべてのプラットフォームで OpenText eDirectory をバックアップできるような、一貫性のあるインタフェースを提供する。
- オブジェクトのインクリメンタルバックアップを行うバックアップソリューションを提供する。

その他の情報

この機能の詳細については、次を参照してください。

- LDAP Libraries for C (https://www.novell.com/documentation/developer/cldap/ldaplibc/data/a2etgcm.html)
- サンプルコード : backup.c (https://www.novell.com/documentation/developer/samplecode/cldap_sample/cldap_sample/extensions/backup.c.html)

SMS による OpenText eDirectory のバックアップ

Storage Management Services(SMS) は、バックアップアプリケーションが完全なバックアップソリューションを提供するために使用する API フレームワークです。SMS は、2 つの主要なコンポーネントで構成されています。

- Storage Management Data Requester (SMDR)
- ターゲットサービスエージェント (TSA:Target Service Agent)

OpenText eDirectory 用の TSA(tsands) は、OpenText eDirectory のターゲットにサービスを提供し、ディレクトリツリーに Storage Management Services API を実装します。アプリケーションは SMS API のトップに構築し、完結したバックアップサービスが提供されます。

注 : OpenText eDirectory 用の TSA は、OES 環境でのみサポートされます。

---

16 Suite B モードでの OpenText eDirectory の設定

Suite B は、秘密または極秘レベルに分類されたトラフィックを市販品で保護できるようにするために国家安全保障局 (NSA) によって標準化された暗号化アルゴリズムのセットです。Suite B アルゴリズムは、パブリックネットワーク経由で渡される機密情報と非機密情報のセキュリティを確保する手段として機能します。

注 : Suite B 標準は変更される場合があります。NSA が今後推奨事項を変更する可能性があることを覚えておいてください。OpenText eDirectory における Suite B のサポートは、NSA の推奨事項の弊社の解釈に基づきます。

Suite B には、次の暗号化アルゴリズムが含まれています。

- 128 ビット鍵または 256 ビット鍵を使用した Advanced Encryption Standard (AES) に基づく暗号化
- P-256 曲線と P-384 曲線上の楕円曲線デジタル署名アルゴリズム (ECDSA) を使用したデジタル署名
- P-256 曲線と P-384 曲線上の楕円曲線 Diffie Hellman (ECDH) 方式を使用した、事前に共有されたまたは動的な鍵交換
- セキュアハッシュアルゴリズム 2 (SHA-256 と SHA-384) に基づくハッシュ法 (デジタル指紋法)

Suite B の詳細については、「Suite B Cryptography (https://en.wikipedia.org/wiki/NSA_Suite_B_Cryptography)」を参照してください。

OpenText eDirectory では、Suite B モードで次のモジュールを個別に設定することができます。

モジュール	説明
NPKI (OpenText Certificate Server)	Certificate Server は、OpenText eDirectory にネイティブに統合された非対称暗号化サービスを提供します。このサービスを使用すれば、ユーザとサーバの両方の証明書を作成、発行、および管理することができます。これらのサービスにより、インターネットなどのパブリック通信チャネルを介した機密データの伝送を保護できます。 Suite B モードで Certificate Server を設定すると、Certificate Server は RFC 5759 に準拠します。この標準では、Suite B 証明書と証明書取り消しリスト (CRL) の基本プロファイルが規定されています。詳細については、521 ページの 「Certificate Server 上での Suite B の有効化」を参照してください。

モジュール	説明
LDAP サービスと HTTP サービス	LDAP サービスは、LDAP クライアントが OpenText eDirectory に保存された情報にアクセスするためのサーバアプリケーションです。これは、クロスプラットフォームの監視および診断機能を、HTTP サービスを利用している eDirectory ツリー内のすべてのサーバに提供します。 Suite B モードでこれらのサービスを設定すると、ECDSA 証明書のサポートが追加され、RFC 6460 で規定された TLS 1.2 暗号と Suite B Cipher の使用が強制されます。詳細については、522 ページの 「ECDSA 証明書と Suite B Cipher を使用するための LDAP サービスと HTTP サービスの設定」を参照してください。
NICI	NICI とは、鍵、アルゴリズム、さまざまな鍵保存 / 使用メカニズム、および大規模な鍵管理システムを提供する暗号化モジュールです。アプリケーションでのデータと鍵の安全な保存と転送を支援するために、NICI では、3 種類の鍵 (鍵保管鍵、NICI Security Domain Infrastructure (SDI) 鍵、およびセッション鍵) を提供しています。 Suite B モードでサーバを設定すると、NICI は、256 ビット AES 鍵を使用してツリー内の機密データを保護します。たとえば、パスワードやチャレンジ / レスポンスデータを保護します。NICI 3.0 にアップグレードすると、自動的に、鍵保管鍵とセッション鍵が Suite B に準拠するように再作成されます。 OpenText eDirectory は、ツリー鍵とも呼ばれる NICI SDI 鍵を使用して、ローカルまたはリモートストレージのデータの暗号化に使用される鍵を安全にラップします。この鍵は、ツリー内のサーバがラップ解除できます。データは、OpenText eDirectory の権限と組み合わせることによってセキュリティが確保されます。ツリー鍵は、ツリー内のすべてのサーバが使用できます。同じデータにアクセスするために、複数のサーバが同じ NICI SDI 鍵を使用します。そのため、この鍵は NICI 3.0 のインストールでは自動的に作成されません。この鍵を手動で作成する必要があります。詳細については、525 ページの 「AES 256 ビット SDI 鍵の作成」を参照してください。
バックグラウンド認証メカニズム	TLS 1.2 に基づく標準ベースのバックグラウンド認証メカニズムを、OpenText eDirectory を使用したシングルサインオン認証に提供します。詳細については、526 ページの 「バックグラウンド認証の有効化」を参照してください。

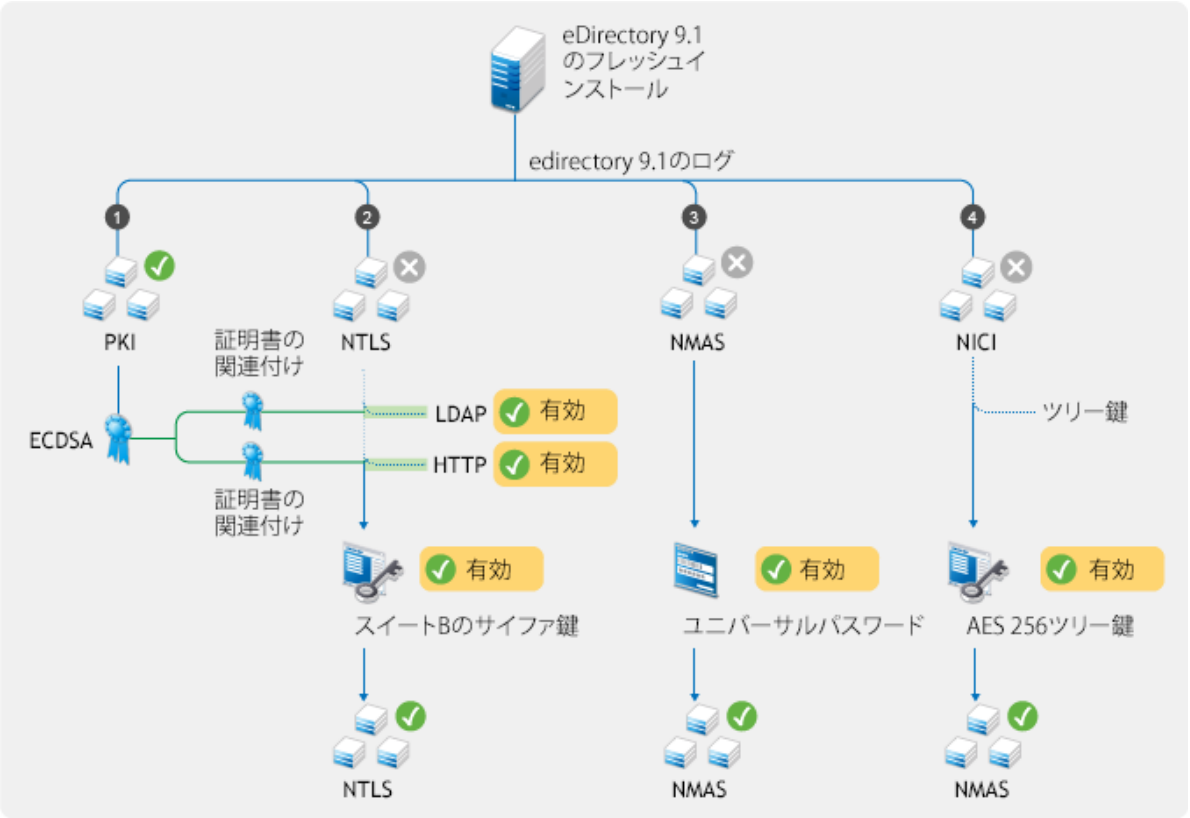
以下の各セクションで、Suite B モードでの OpenText eDirectory モジュールの設定に関する情報について説明します。

- 520 ページの 「新規インストールでの Suite B の有効化」
- 526 ページの 「既存のサーバ上での Suite B の設定」

新規インストールでの Suite B の有効化

図 16-1 に、新しいインストールで、OpenText eDirectory コンポーネントに対して Suite B を有効にするための一連のタスクを示します。

図 16-1 新規インストールでの Suite B の有効化



Certificate Server 上での Suite B の有効化

新しいツリーを設定すると、Certificate Server が、従来の RSA 証明書に加えて、ツリー認証局 (CA) 用の P-384 曲線上の自己署名 ECDSA 証明書を作成します。新しいサーバをツリーに追加するか、古いサーバを OpenText eDirectory 9.2 にアップグレードすると、Certificate Server がこれらのサーバに ECDSA 証明書を発行します。

注 : デフォルトで、OpenText Certificate Server は P-384 曲線上の ECDSA 証明書を作成します。ただし、P-256 曲線上のサーバ証明書を作成することもできます。

Suite B を有効にせずに ECDSA 証明書だけを使用することができます。Suite B の有効化は、RFC 5759 に準拠するための追加のステップです。

CA Certificate Server を Suite B モードで動作するように設定するには、次の手順を実行します。

1. CA Certificate Server の Enhanced Background Authentication を設定します。

   詳細については、526 ページの 「バックグラウンド認証の有効化」を参照してください。
2. OpenText Identity Console を起動し、[**証明書管理**] タイル > [**CA 管理**] の順にクリックします。

3 ［CA 環境設定  ］をクリックします。

4 ［CERTIFICATE SELF PROVISIONING］ページで、［Suite B モードを有効にする (Enable Suite B Mode)］チェックボックスをオンにします。

5 ［適用］>［OK］の順にクリックします。

CA Certificate Server が Suite B モードの場合は、認証局が RSA 証明書の作成を許可しません。また、サーバの自動プロビジョニングで RSA 証明書がこれ以上生成されません。新しいサーバを追加する場合は、サーバが Enhanced Background Authentication を実行するように設定されていることを確認してください。

ツリー内のすべてのサーバまたはツリーに接続している任意の外部サービスが ECDSA サーバ証明書の使用を開始したら、不要な RSA 証明書を無効にしたり、削除したりできます。

注 : OpenText eDirectory 8.8.8 Patch 6 で導入された Follow CA のアルゴリズム機能は、OpenText eDirectory 9.0 以降では使用できなくなりました。代わりに、OpenText eDirectory 9.2 サーバでは RSA 証明書には SHA-256 アルゴリズムを、ECDSA 証明書には SHA-384 をデフォルトで使用します。

ECDSA 証明書と Suite B Cipher を使用するための LDAP サービスと HTTP サービスの設定

NTLS (OpenText Transport Layer Security) は、FIPS 準拠 OpenSSL モジュールを通して TLS 1.2 と Suite B 暗号アルゴリズムをサポートします。FIPS 準拠 OpenSSL モジュールは、LDAP、httpstk (iMonitor)、NCP エンジンなどの OpenText eDirectory コンポーネントで使用されます。詳細については、『*OpenText™ eDirectory CE 24.4 (v9.3) インストールガイド*』の「FIPS モードでの OpenText™ eDirectory の運用」を参照してください。

サーバ上で Suite B モードを有効にする前に、サーバに ECDSA 証明書がインストールされており、OpenText eDirectory 環境内の LDAP クライアント、LDAP ブラウザ、および Web ブラウザが TLS 1.2、ECDSA 証明書、および Suite B Cipher をサポートしていることを確認します。Suite B モードで LDAP インタフェースと HTTPS インタフェースを設定するには、必要な Suite B モードでインタフェースを有効にして、該当する ECDSA サーバ証明書をそれらに関連付けます。ツリー内の eDirectory サーバごとにこの手順を繰り返します。サーバの暗号レベルと ECDSA サーバ証明書を表示するには、サーバの LDAP 設定オブジェクトと httpstk 設定オブジェクト (ldapServer と httpServer) を使用します。

Suite B モードで LDAP サーバを設定するには :

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory にログインします。

3 Suite B モードで設定したい［LDAP サーバ］オブジェクトを選択します。

4 ［情報］をクリックします。

5 ［サーバ証明書］ドロップダウンで、LDAP サーバオブジェクトで使用する［SSL 楕円曲線証明書 DNS (SSL Elliptic Curve CertificateDNS)］を選択します。

6 ［環境設定］タイルをクリックします。

7 LDAP サーバオブジェクトに対して有効にする Suite B モードに応じて、［暗号化に際してのバインド制限］ドロップダウンリストから値を選択します。

サイファに関するバインド制限	サイファスイート	説明
SuiteB Cipher (128 ビット) を使用する	◆ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 ◆ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	128 ビットレベルのセキュリティを使用することで、Suite B モードの操作を有効にします。このオプションを選択すると、OpenText eDirectory は、ピア (任意の LDAP クライアント) による 128 ビットレベルと 192 ビットレベルの両方のセキュリティの使用を許可します。このオプションでは ECDSA 256 または ECDSA 384 のいずれかの証明書を使用できます。
SuiteB Cipher (128 ビットのみ) を使用する	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	128 ビットレベルのセキュリティを使用することで、Suite B モードの操作を有効にします。このオプションを選択すると、OpenText eDirectory は、ピア (任意の LDAP クライアント) による 192 ビットレベルのセキュリティの使用を許可しなくなります。 証明書チェーン内のすべての証明書で P-256 曲線上の ECDSA 鍵を使用する必要があります。これは、サーバでは必須であり、クライアントではクライアント証明書の検証が有効になっている場合に適用されます。
SuiteB Cipher (192 ビット) を使用する	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	192 ビットレベルのセキュリティを使用することで、Suite B モードの操作を有効にします。このオプションを選択すると、OpenText eDirectory は、ピア (任意の LDAP クライアント) による 192 ビットレベルのセキュリティの使用のみを許可します。 証明書チェーン内のすべての証明書で P-384 曲線上の ECDSA 鍵を使用する必要があります。これは、サーバでは必須であり、クライアントではクライアント証明書の検証が有効になっている場合に適用されます。

OpenText eDirectory では、ldapbindrestrictions とサイファレベルの値を組み合わせて使用できます。詳細については、417 ページの 表 14-1 を参照してください。

8 ［保存］をクリックして、［OK］をクリックします。

9 変更を有効にするには、次のいずれかの操作を実行します。

- OpenText eDirectory を再起動します。
- LDAP サーバをアンロードしてロードします。

Suite B モードで HTTPS インタフェースを設定するには :

1 適切な権利を持った管理者として OpenText eDirectory ツリーを含む OpenText Identity Console にログインします。

2 [**オブジェクト管理**] に移動します。

3 [**タイプ**] ドロップダウンから、変更する httpServer オブジェクトを選択し、[**検索**] をクリックします。

4 [Http サーバ] をクリックし、[**値がある属性**] リストから [httpBindRestrictions] を選択します。

5 httpserver オブジェクトに対して有効にする Suite B モードに応じて、表示されたダイアログで値を 4、5、または 6 に変更します。

サイファに関するバインド制限	サイファスイート	説明
4 - SuiteB Cipher (128 ビット) を使用する	◆ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 ◆ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	128 ビットのセキュリティレベル (Suite B Cipher 128 ビット) を使用して、Suite B モード操作を有効にします。このオプションを選択すると、OpenText eDirectory はクライアント (Web ブラウザ) による 128 ビットと 192 ビットの両方のセキュリティレベルを許可します。このオプションでは ECDSA 256 または ECDSA 384 のいずれかの証明書を使用できます。
5 - SuiteB Cipher (128 ビットのみ) を使用する	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	128 ビットのセキュリティレベル (Suite B Cipher 128 ビットのみ) を使用して、Suite B モード操作を有効にします。このオプションを選択すると、OpenText eDirectory はクライアント (Web ブラウザ) による 192 ビットのセキュリティレベルを許可しません。 証明書チェーン内のすべての証明書で P-256 曲線上の ECDSA 鍵を使用する必要があります。これは、サーバでは必須であり、クライアントではクライアント証明書の検証が有効になっている場合に適用されます。

サイファに関するバインド制限	サイファスイート	説明
6 - SuiteB Cipher (192 ビット) を使用する	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	192 ビットのセキュリティレベル (Suite B Cipher 192 ビット) を使用して、Suite B モード操作を有効にします。このオプションを選択すると、OpenText eDirectory はクライアント (Web ブラウザ) による 192 ビットのセキュリティレベルのみを許可します。 証明書チェーン内のすべての証明書で P-384 曲線上の ECDSA 鍵を使用する必要があります。これは、サーバでは必須であり、クライアントではクライアント証明書の検証が有効になっている場合に適用されます。

6 [値がある属性] リストから httpKeyMaterialObject を選択して、[編集] をクリックします。

7 ブラウズして HTTPS インタフェースで使用する楕円曲線証明書を選択し、[OK] をクリックします。

8 [保存] をクリックします。

9 [OK] をクリックします。

10 変更を有効にするために eDirectory を再起動します。

AES 256 ビット SDI 鍵の作成

デフォルトで、NICI SDI 鍵は 3DES 鍵です。ただし、Suite B モードをサポートするには、AES 256 ビット NICI SDI 鍵を手動で作成する必要があります。ツリー内のすべてのサーバが OpenText eDirectory 9.0 以降の場合にのみ、この鍵を作成します。

KAP.Security コンテナの書き込み可能なレプリカが保存されているサーバが OpenText eDirectory 9.2 にアップグレードされた場合は、PKI ヘルスチェックでこのコンテナに W1 オブジェクトが作成されます。ツリー内のすべてのサーバが eDirectory 9.2 にアップグレードされた場合は、ツリー管理者が AES 256 ビット NICI SDI キーを作成できます。

AES 256 ビット NICI SDI 鍵を作成するには、『NICI Administration Guide (NICI 管理ガイド)』の「Creating an AES 256-Bit Tree Key (AES 256 ビットツリーキーを作成する」の手順に従ってください。

AES 256 ビット NICI SDI 鍵を使用したデータの再暗号化

NMAS では、NICI SDI 鍵を使用して、パスワードとチャレンジ / レスポンス設定 (質問と回答) が安全に保存されます。また、NMAS には、NICI SDI 鍵を使用するユーザ固有の設定とメソッド固有の設定用の秘密の場所があります。大規模な展開で複数のユーザのパスワードを再暗号化するには、diagpwd ユーティリティを使用します。詳細については、823 ページの 「ユニバーサルパスワードの診断ユーティリティ」を参照してください。

重要 : OpenText eDirectory 9.0 より前のバージョンのサーバが OpenText eDirectory 環境に含まれている場合、それらのサーバは AES 256 ビットのツリーキーを使用して暗号化されたパスワードや秘密データを復号化することができず、これらのサーバへのログインは失敗します。

バックグラウンド認証の有効化

OpenText eDirectory は、アクセスを要求しているユーザの識別情報を検証する強力な認証メカニズムを提供します。Enhanced Background Authentication の詳細については、「529 ページの第 17 章「Enhanced Background Authentication の有効化」」を参照してください。

既存のサーバ上での Suite B の設定

OpenText eDirectory ツリー内の既存のサーバ上で Suite B を有効にするには、次の操作を実行します。

1. 認証局として機能するサーバを OpenText eDirectory 9.3 にアップグレードします。

   認証局サーバをアップグレードすると、ECDSA 自己署名認証局証明書が作成されます。他のサーバを OpenText eDirectory 9.3 にアップグレードすると、新しい認証局がこれらのサーバに ECDSA 証明書を発行します。

2. ツリー内の必要なサーバを OpenText eDirectory 9.3 にアップグレードします。

   アップグレードプロセスで、アップグレードされたサーバ用の ECDSA 証明書が生成されます。これらの証明書を使用して、LDAP および HTTP プロトコルスタックインタフェースを Suite B モードにする必要があります。詳細については、522 ページの「ECDSA 証明書と Suite B Cipher を使用するための LDAP サービスと HTTP サービスの設定」を参照してください。

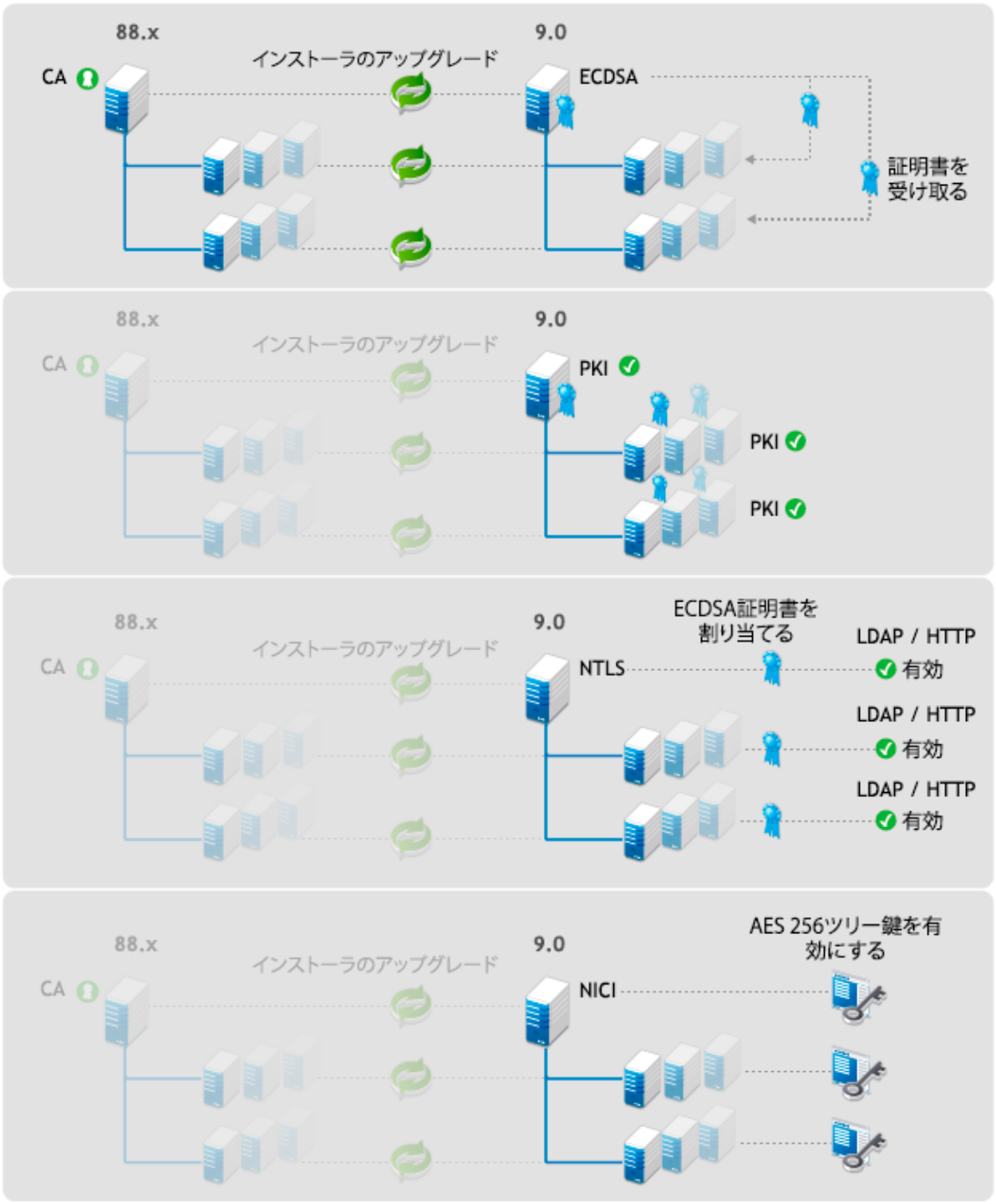
3. AES 256 ビット SDI 鍵を作成します。詳細については、525 ページの 「AES 256 ビット SDI 鍵の作成」を参照してください。

4. AES 256 ビット NICI SDI 鍵を使用してデータを再暗号化します。詳細については、525 ページの 「AES 256 ビット SDI 鍵の作成」を参照してください。

5. バックグラウンド認証を設定します。詳細については、529 ページの 「Enhanced Background Authentication の有効化」を参照してください。

図 16-2 に、OpenText eDirectory のアップグレード時に Suite B を有効にするため一連のタスクを示します。

図 16-2 OpenText eDirectory のアップグレード時の Suite B の有効化



17 Enhanced Background Authentication の有効化

OpenText eDirectory は、アクセスを要求しているユーザの識別情報を検証する強力な認証メカニズムを提供します。認証は 2 つのフェーズで構成されます。

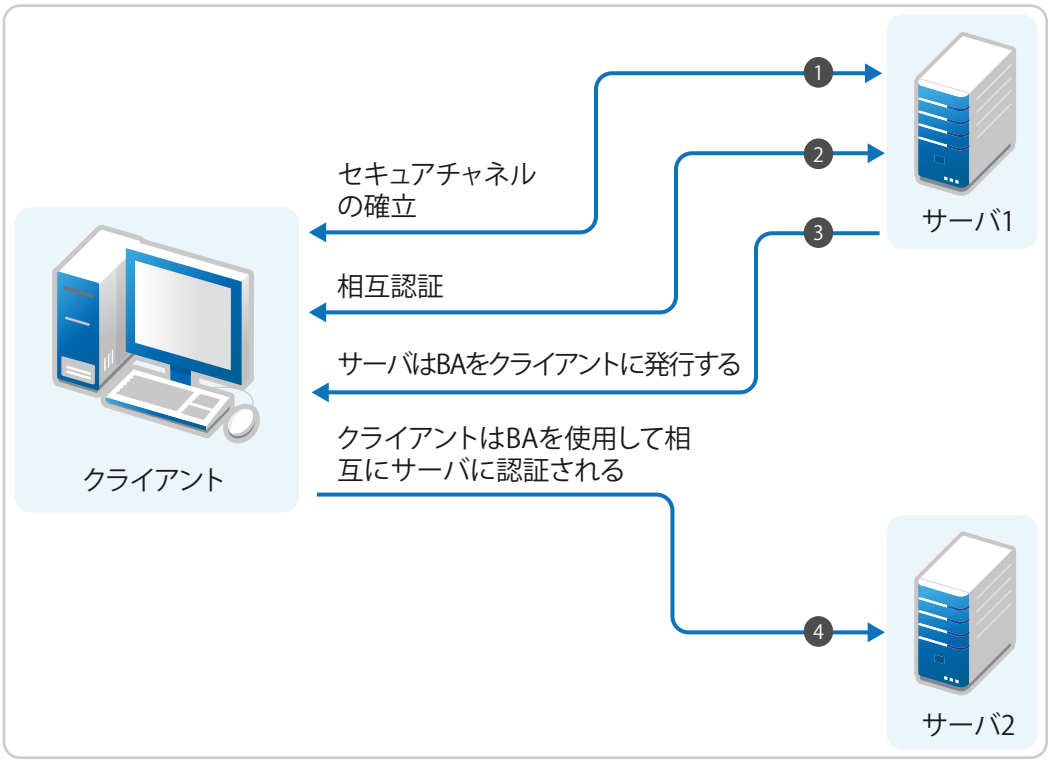
- ログイン
- バックグラウンド認証 (BA)

ユーザがログインすると、NMAS (NetIQ モジュラー認証サービス) が、ユーザの長期資格情報 (パスワードなど) を検証して、BA マテリアルをユーザに発行します。

ツリー内の他のサーバに対する認証中に、ユーザがこの BA マテリアルを使用します。この OpenText eDirectory のシングルサインオン機能を使用すれば、長期資格情報を再度入力しなくても、ツリー内の任意のサーバに対して認証できます。

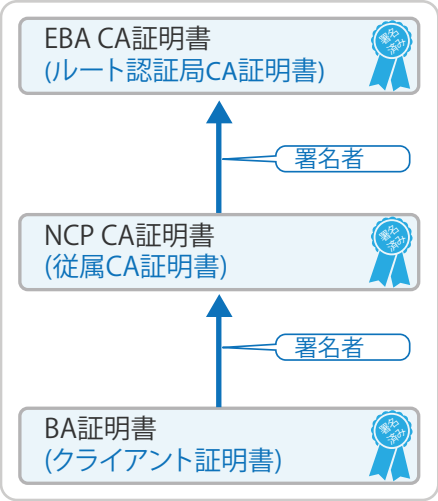
OpenText eDirectory 9.0 では、専有の BA プロトコルの制限を解決できる標準ベースの BA プロトコルを導入しています。このプロトコルは、Enhanced Background Authentication (EBA) と呼ばれています。EBA が使用されている場合は、NMAS がユーザに BA マテリアルとして X.509 認定を発行し、BA プロトコルが相互認証に TLS バージョン 1.2 を使用します。

図 17-1 EBA プロセス



EBA 対応 OpenText eDirectory ツリーでは、EBA CA が EBA 用の信頼されたルート認証局です。EBA CA では自己署名証明書が使用されます。ツリールートパーティションの書き込み可能なレプリカを EBA CA として使用して、ツリー内のサーバのいずれかを設定できます。通常は、ツリールートパーティションの書き込み可能なレプリカをホストするツリー内で設定され、EBA を使用して設定された最初のサーバが EBA CA として機能します。また、ツリールートパーティションの書き込み可能なレプリカを含むツリー内の任意の OpenText eDirectory 9.2 サーバを EBA CA として機能するように設定することもできます。

***図 17-2** EBA 認証局*



ツリー内の各 EBA 対応サーバが、EBA CA に従属する認証局になり、NCP CA と呼ばれます。ログイン後に、NMAS が NCP CA によって発行された BA 証明書をログインしているユーザに返します。

注 : OpenText eDirectory にログインするどのオブジェクトについても、オブジェクト DN のすべてのネーミング属性の OID がスキーマに設定されている必要があります。

EBA を使用して OpenText eDirectory サーバに対して認証するには、クライアントにツリーの EBA CA 証明書が必要です。EBA CA 証明書を取得するには、ebaclientinit ユーティリティを使用します。これは、OpenText eDirectory 9.0 以降にパッケージ化されている新しいコマンドラインユーティリティです。このユーティリティは、ツリーの EBA CA 証明書をダウンロードし、. eba.p12 という名前のファイルに保存します。このファイルは、Linux 上のユーザのホームディレクトリ ($HOME) と Windows 上のユーザのプロファイルディレクトリ (%USERPROFILE%) に存在します。

注 : EBA を正しく機能させるためには、OpenText eDirectory 環境内のすべての EBA 対応サーバとクライアント上の時刻を同期させます。

---

ebaclientinit ユーティリティを複数のツリーに対して実行すると、OpenText eDirectory ツリーに関連付けられた EBA CA 証明書が .eba.p12 ファイルに追加されます。各 OpenText eDirectory ツリー用の EBA CA 証明書を取得するには、ツリーごとに 1 回ずつ ebaclientinit ユーティリティを実行します。

EBA の有効化

このセクションでは、OpenText eDirectory 上で EBA を有効にします。インストールのタイプに応じて、次のセクションのいずれかの手順に従います。

- 531 ページの 「OpenText eDirectory ツリー上での EBA の有効化」
- 532 ページの 「OpenText eDirectory サーバ上での EBA の有効化」
- 533 ページの 「OpenText eDirectory サーバ上での EBA の無効化」

OpenText eDirectory ツリー上での EBA の有効化

- 531 ページの 「新しい OpenText eDirectory ツリー上での EBA の有効化」
- 531 ページの 「既存のツリー上での EBA の有効化」

新しい OpenText eDirectory ツリー上での EBA の有効化

新しいツリー上で EBA を有効にするには、プラットフォームに応じて、次の操作のいずれかを実行します。

- **Linux:** 新しい OpenText eDirectory ツリーを設定するときに EBA を有効にするには、コマンドラインで —configure–eba-now 引数を指定して ndsconfig コマンドを実行します。

  例 : ndsconfig new —configure–eba-now *yes*

  この引数がコマンドに渡されなかった場合は、EBA を有効にするよう求めるプロンプトが表示されます。初期設定に基づいて、プロンプトで「**yes**」または「**no**」と入力します。
- **Windows:** インストールプログラムには、OpenText eDirectory の環境設定中に EBA を有効にするオプションが用意されています。EBA を有効にするには、環境設定時に［**EBA を有効にする**］オプションを選択します。

注 : OpenText eDirectory では、EBA 対応サーバ上で ebaext.dlm モジュールと ebassl_srv.dlm モジュールの自動起動を設定することができません。これは、OpenText eDirectory サーバ上で EBA が有効になっていると、DS モジュールが自動的にこれらのモジュールをロードするためです。

非 EBA 対応サーバ上で ebaext.dlm モジュールと ebassl_srv.dlm モジュールをロードしようとすると、モジュールは正常にロードされますが、EBA 機能は使用できません。

既存のツリー上での EBA の有効化

既存の OpenText eDirectory ツリー上で EBA を有効にするには、プラットフォームに応じて、次の操作のいずれかを実行します。

- **Linux:** ツリールートパーティションの書き込み可能なレプリカが保存されているサーバのいずれかで、—configure–eba-now 引数を指定して、ndsconfig upgrade コマンドを実行します。

例 : ndsconfig upgrade —configure–eba-now *yes*

- **Windows:** ツリールートパーティションの書き込み可能なレプリカが保存されているサーバのいずれかの OpenText eDirectory 9.2 インストールフォルダから eDirectory_910_Windows_x86__64.exe を実行し、OpenText eDirectory の環境設定時に［**EBA を有効にする**］オプションを選択します。

OpenText eDirectory サーバ上での EBA の有効化

OpenText eDirectory サーバ上で EBA を有効にすると、証明書署名要求 (CSR) が EBA CA に送信されます。EBA CA は、CSR を検証して、アクセス制御チェックを実行してから、NCP CA 証明書をサーバに発行します。サーバ上で EBA を有効にする前に、次のことを確認します。

- (必須)管理者DNを含むパーティションの書き込み可能なレプリカがツリー内のEBA対応サーバ上に存在する。
- (オプション) サーバオブジェクトを含むパーティションの書き込み可能なレプリカがツリー内の EBA 対応サーバ上に存在する。サーバがこの条件を満たしていない場合、EBA CA は CSR を保存し、NCP CA 証明書を発行しません。これにより、OpenText eDirectory の環境設定が失敗し、管理者が OpenText Identity Console の EBA プラグインを使用して CSR を承認する必要があります。詳細については、535 ページの「OpenText Identity Console を使用した EBA CA の管理」を参照してください。CSR を承認したら、ndsconfig upgrade コマンドを実行することによって EBA を設定します。たとえば、「ndsconfig upgrade --configure-eba-now *yes*」と入力します。
- 532 ページの 「新しいサーバの追加時の EBA の有効化」
- 532 ページの 「設定済みのサーバ上での EBA の有効化」

新しいサーバの追加時の EBA の有効化

新しいサーバをツリーに追加するときに EBA を有効にするには、プラットフォームに応じて、次の操作のいずれかを実行します。

- **Linux:** —configure–eba-now yes 引数を指定して、*ndsconfig add* コマンドを実行します。

  例 : ndsconfig add —configure–eba-now *yes*
- **Windows:** eDirectory 9.3 のインストールフォルダから eDirectory_910_Windows_x86__64.exe を実行して、OpenText eDirectory の環境設定時に［**EBA を有効にする**］オプションを選択します。

設定済みのサーバ上での EBA の有効化

設定済みのサーバ上で EBA を有効にするには、プラットフォームに応じて、次の操作のいずれかを実行します。

- **Linux:** —configure–eba-now yes 引数を指定して、*ndsconfig upgrade* コマンドを実行します。

  例 : ndsconfig upgrade —configure–eba-now *yes*

- **Windows:** OpenText eDirectory 9.3 のインストールフォルダから eDirectory_910_Windows_x86__64.exe を実行して、OpenText eDirectory の環境設定時に［**EBA を有効にする**］オプションを選択します。

重要 : EBA CA として機能するサーバに加えて、ツリールートパーティションの読み書き可能レプリカを含む EBA 対応サーバを 1 台以上用意することをお勧めします。EBA CA として機能するサーバがダウンした場合は、他の EBA 対応サーバを EBA CA として機能するように設定することができます。詳細については、537 ページの 「新しいサーバへの EBA CA の役割の移動」を参照してください。

OpenText eDirectory サーバ上での EBA の無効化

設定済みのサーバ上で EBA を無効にするには、プラットフォームに応じて、次の操作のいずれかを実行します。

- **Linux:**
  - EBA を無効にして OpenText eDirectory サーバを再起動するには、次のコマンドを実行します。

    ```
    ndsmanage stopall
    export DISABLE_EBA=true
    ndsmanage startall
    ```

  - EBA を有効にして OpenText eDirectory サーバを再起動するには、次のコマンドを実行します。

    ```
    ndsmanage stopall
    unset DISABLE_EBA
    ndsmanage startall
    ```

  注 : RHEL 7.x プラットフォームと SLES 12.x プラットフォームの /etc/opt/novell/eDirectory/conf ディレクトリに配置された env ファイルに OpenText eDirectory サービスに必要なすべての環境変数を追加する必要があります。

- **Windows:** ［**コントロールパネル**］> ［**システム**］> ［**システムの詳細設定**］> ［**環境変数**］> ［**システム変数**］> ［**新規**］の順に移動します。DISABLE_EBA という新しい変数を追加して、値 1 を設定し、サーバを再起動します。

重要 : EBA を無効にするのはトラブルシューティングの場合だけにしてください。EBA CA として機能する OpenText eDirectory サーバで 7 日以上 EBA を無効にすると、OpenText eDirectory ツリーの EBA 機能が切断されます。詳細については、TID 7017232 を参照してください。

EBA に関する情報の表示

EBA 対応サーバのさまざまな側面に関する情報を提供するツールとユーティリティを以下に示します。

ユーティリティ	説明
ndstrace	EBA 要求の処理や NCP CA 証明書の発行などの EBA 操作に関する情報を提供します。この情報を表示するには、次の手順を実行します。 Linux 上では、ndstrace の EBA タグを有効にします。 Windows 上では、NDSCons.exe で EBA タグが有効になっている dstrace モジュールをロードします。
ndsd.log、dhost.log	EBA 開始メッセージと停止メッセージに関する情報を提供します。この情報を表示するには、次の手順を実行します。 Linux 上では、EBA を設定したサーバの ndsd.log ファイルを参照します。 Windows 上では、この情報が dhost.log ファイルに記録されます。
ndscheck	サーバ上の EBA ステータス (サーバが EBA 対応かどうか) を提供します。 サーバが EBA 対応の場合は、コマンドの出力に NCP CA 証明書の有効性やサーバが EBA CA として機能しているかどうかなどの情報が表示されます。 ndscheck コマンドをリモートで実行する場合は、ツリーの EBA CA 証明書がローカルコンピュータにダウンロードされていることを確認してください。
ndslogin	EBA の環境設定に関するトラブルシューティング情報を提供します。 EBA の環境設定をトラブルシューティングするには、-c 引数を指定した ndslogin コマンドを使用して OpenText eDirectory にログインします。 例 : ndslogin <admin DN> –p <password> –c ログインに成功するためには、ツリーの EBA CA 証明書がローカルコンピュータにダウンロードされていることを確認してください。
schema.log	EBA が設定されているサーバの schema.log ファイルに、EBA のスキーマ拡張に関する情報が書き込まれます。
nioutput.log	OpenText eDirectory の環境設定時に EBA が選択されたかどうかを指定します。

ユーティリティ	説明
iMonitor	OpenText iMonitor を使用すれば、EBA 対応サーバで次の情報を監視することができます。 ◆ ［エージェント環境設定］ページの［接続情報］タブで EBA 対応パラメータを確認することによって、サーバが EBA 対応かどうかを判断します。このパラメータの値が true の場合は、サーバが EBA 対応です。そうでない場合は、このパラメータの値が false になります。 ◆ Linux上のndstraceで表示可能なものと同じデバッグ情報を表示します。トレース設定ページに、この情報を表示するための EBA 用のタグが含まれています。 ◆ ［エージェントヘルス］ページで、EBA CA 証明書と NCP CA 証明書の有効性、およびサーバの EBA 対応ステータスに関する情報を表示します。これらの項目は、証明書が有効で、EBA 属性が影響を受けていなければ、緑色で表示されます。 ◆ ［Verb 統計情報］ページで、EBA 対応サーバ上の EBA 要求動詞を表示します。 ◆ ［iMonitor Health Check Agent］ページで以下を確認します。 ◆ サーバが EBA 対応であるかどうか ◆ サーバが EBA CA をホストしているかどうか ◆ EBA CA 証明書が有効かどうか ◆ NCP CA 証明書が有効かどうか

OpenText Identity Console を使用した EBA CA の管理

OpenText Identity Console の［**EBA**］タイルから OpenText eDirectory にアクセスするには、EBA CA 証明書を OpenText Identity Console の信頼された EBA 証明書ストアに配置する必要があります。

［EBA CA 管理］ページを開くには、OpenText Identity Console にログインし、ホームページで［**EBA**］タイルをクリックします。［**EBA CA 管理**］が開きます。

［**EBA CA 管理**］ページには、EBA CA のさまざまな側面を管理するための以下のタブが表示されます。

- **一般 :** EBA CA の IP アドレスとその証明書を表示します。
- **発行された証明書 :** NCP CA 証明書を IP アドレスおよびポートとともに表示します。

  証明書を取り消すには、証明書を選択して [**取り消し**] をクリックします。証明書を無効にすると、NCP CA 証明書を所有しているサーバが機能しなくなるため、このオプションは異常な状況以外は使用しないでください。証明書を無効にする必要があるのはサーバが侵害された場合などです。
- **CSR:** 管理者の承認が保留になっている証明書署名要求を列挙します。証明書署名要求を承認するには、リストから証明書を選択して、［**承認**］をクリックします。

EBA が有効な場合の OpenText eDirectory 操作の制限

パーティションのマスタ読み書き可能レプリカが保存されているサーバが EBA 対応の場合は、そのパーティションも EBA 対応と見なされます。OpenText eDirectory では、パーティションを非 EBA 対応にする操作が禁止されます。EBA 対応サーバでは、パーティションとレプリカに対する操作に次の制限が加えられます。

- 536 ページの 「レプリカタイプの変更に関する制限」
- 536 ページの 「パーティションのマスタの変更に関する制限」
- 536 ページの 「パーティションのマージに関する制限」
- 537 ページの 「EBA が有効になっているサーバの再設定に関する制限」

レプリカタイプの変更に関する制限

- EBA CA をホストしているサーバにツリールートパーティションのマスタレプリカが保存されている場合、OpenText eDirectory はそのレプリカタイプの変更を許可しません。
- EBA CA がツリールートパーティションの読み書き可能レプリカが保存されているサーバによってホストされている場合は、レプリカタイプをマスタ以外に変更しないでください。レプリカタイプを［読み込み専用］/［フィルタ済み読み込みのみ］に変更すると、ツリー全体で EBA 機能が停止する可能性があります。OpenText eDirectory は、ツリールートパーティションのマスタレプリカが保存されている OpenText eDirectory サーバが EBA 対応の場合にこの制限を適用します。

注 : 従来のサーバと OpenText eDirectory 9.2 サーバが混在する環境では、レプリカタイプの変更に成功する場合があります。ただし、これを行うと、EBA 機能が停止する可能性があります。

パーティションのマスタの変更に関する制限

パーティションのマスタレプリカが EBA 対応サーバ上に存在する場合は、次の操作が失敗します。

- マスタの役割を非 EBA 対応サーバに移動する。
- サーバが EBA CA として機能している場合に、マスタの役割を他のサーバに移動する。

パーティションのマージに関する制限

親パーティションが非 EBA 対応で、子パーティションが EBA 対応の場合は、この 2 つのパーティションをマージしないでください。これを行うと、EBA 機能が停止する可能性があります。

EBA が有効になっているサーバの再設定に関する制限

OpenText eDirectory が EBA 対応モードのサーバ上で設定されている場合は、この設定が nds.conf ファイルの n4u.server.eba_enabled パラメータに保存されます。OpenText eDirectory がこのサーバ上で設定解除されてから再び設定された場合は、デフォルトで EBA がオンになります。非 EBA モードでサーバを設定するには、サーバ上で OpenText eDirectory を設定する前に、nds.conf ファイルからこのパラメータを削除します。

EBA 対応サーバのバックアップ

EBA 対応 OpenText eDirectory サーバをバックアップする場合は、「Backing Up and Restoring OpenText eDirectory」の手順に従ってください。EBA 対応サーバのインクリメンタルバックアップまたはフルバックアップを取得するときに、NICI 属性とストリーム属性が選択されていることを確認します。そうでない場合は、サーバを復元することができません。

新しいサーバへの EBA CA の役割の移動

EBA CA として機能しているサーバがダウンした場合は、OpenText eDirectory がツリー内の別の EBA 対応サーバに EBA CA の役割を移動できる柔軟性を提供します。EBA CA の役割を新しいサーバに移動する前に、新しいサーバが以下の状態になっていることを確認します。

- EBA 対応である。
- EBA CA がダウンした時点ですでにレプリカが作成されていたツリールートパーティションの書き込み可能なレプリカが保存されている。

Linux オペレーティングシステム上の新しいサーバに EBA CA の役割を移動するには、新しいサーバ上の bash シェルから次のコマンドを実行します。

```
ndstrace -c "config ebassl_srv seize_ebaca"
```

OpenText eDirectory には、EBA CA の役割が新しいサーバに移動されたことを示す成功メッセージが表示されます。オリジナルのサーバがまだ機能しているときにこの操作を試すと、失敗します。

Windows では、次の手順を実行します。

1. ndscons.exe を開きます。
2. **［スタート］** > **［設定］** > **［コントロールパネル］** > **［OpenText eDirectory サービス］** の順にクリックします。
3. **［サービス］** タブで、ebassl_srv.dlm までスクロールしてから、**［起動パラメータ］** フィールドに「seize_ebaca」と入力します。
4. **［設定］** をクリックします。

EBA CA の役割の転送に関するメッセージを表示するには、EBA CA 役割の転送操作実行時に、EBA タグを有効にして dstrace.dlm を実行します。DSTrace に、操作の成功または失敗に応じて適切なメッセージが表示されます。オリジナルのサーバがまだ機能しているときにこの操作を試すと、失敗します。

注

- EBA CA の役割が正常に取得されたかどうかを判断するには、新しいサーバ上で ndscheck を実行します。ndscheck の出力に「EBACA=true」と表示された場合は、新しいサーバがツリーの EBA CA になっています。
- EBA CA をホストしているサーバがダウンした場合は、ツリールートパーティションのレプリカリング内の他のサーバを EBA CA として指定します。ダウンしたサーバにツリールートパーティションのマスタレプリカが保存されていた場合は、EBA CA として機能している新しいサーバにマスタの役割を移動することをお勧めします。マスタの役割を移動するには、353 ページの 「レプリカの修復」の手順に従います。

---

18 OpenText eDirectory の SNMP サポート

SNMP(Simple Network Management Protocol) は、インターネットを介してデバイスを操作および保守するための標準的なプロトコルです。管理コンソールアプリケーションと管理対象デバイスは、このプロトコルに従って管理情報をやり取りします。管理コンソールアプリケーションとは、IBM Tivoli NetView や Solstice SunNet Manager などのアプリケーションを指します。管理対象デバイスには、ホスト、ルータ、ブリッジ、ハブのほかに、OpenText eDirectory などのネットワークアプリケーションも含まれます。

この章では、OpenText eDirectory の SNMP サービスについて説明します。この付録には以下のトピックがあります。

- 539 ページの 「SNMP に関する用語の定義」
- 540 ページの 「SNMP サービスについて」
- 542 ページの 「OpenText eDirectory と SNMP」
- 545 ページの 「OpenText eDirectory の SNMP サービスのインストールと設定」
- 552 ページの 「SNMP による OpenText eDirectory の監視」
- 582 ページの 「トラブルシューティング」

SNMP に関する用語の定義

この章で使われる用語の定義を次の表に示します。

用語集	定義
EMANATE	SNMP Research International, Inc. の製品。Enhanced MANagement Agent Through Extensions の略称です。
［SNMP］	Simple Network Management Protocol の略。ネットワークの稼動状況に関するデータをやり取りするためのプロトコルです。
NAA	Native Agent Adapter の略。ネイティブエージェントアダプタ。
NMS	ネットワーク管理ステーション
MA	Management Agent
SA	サブエージェント
MIB	Management Information Base
NCP	NetWare/Novell Core Protocol
NMA	Network Management Application の略。ネットワーク管理アプリケーション。

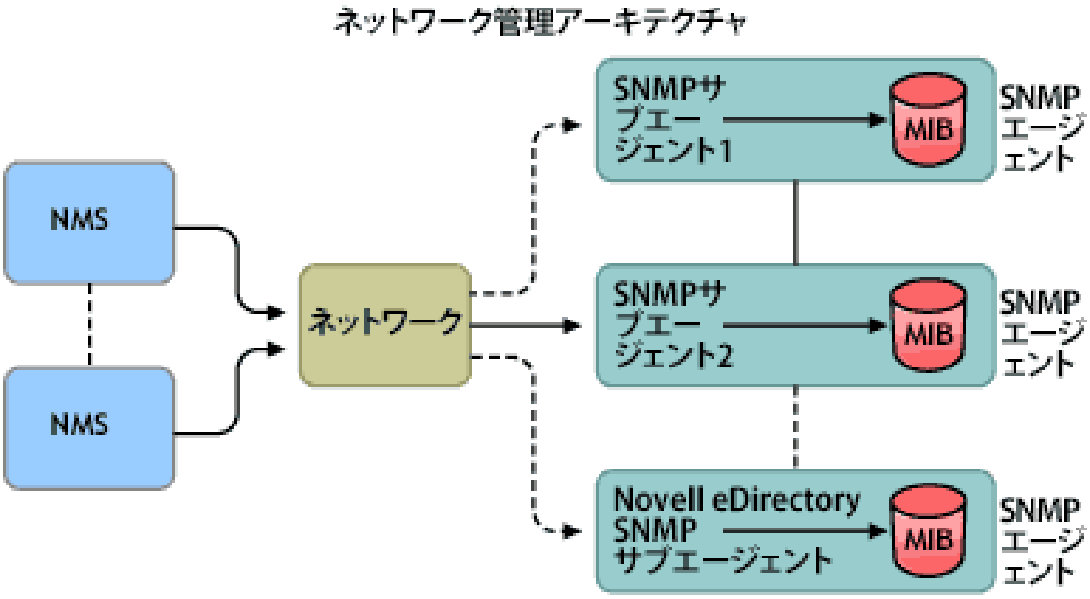
用語集	定義
edir.mib	OpenText eDirectory サーバの監視に使う MIB のこと。OpenText eDirectory に関する MIB オブジェクトおよびトラップが設定されています。
トラップ	OpenText eDirectory イベントがサーバ上で発生したときに、管理対象デバイス上のエージェントが発する警告のこと。警告を発する条件は、OpenText が提供する MIB(管理情報ベース) で定義されています。

SNMP サービスについて

SNMP は「マネージャ / エージェント」アーキテクチャにもとづくプロトコルです。SNMP を使用して行われるネットワーク管理のアーキテクチャは、次のような要素から成ります。

- NMS(Network Management Station)
- 管理対象デバイス
- マスタエージェント
- サブエージェント
- MIB(Management Information Base)
- ネットワーク管理プロトコル

図 18-1 ネットワーク管理アーキテクチャ



ネットワーク管理ステーション

ネットワーク管理ステーション (NMS) とは、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ワークステーションのことです。管理対象デバイスに関する情報をグラフィック表示します。

NMS には次のような機能があります。

- ネットワーク管理システム全体のユーザインタフェースを提供します。この機能により、ネットワーク管理の強固さ、柔軟性、および使いやすさが決まります。
- SNMP Get、Get Next、SNMP GetResponse、および Set の操作はここから実行します。また、ネットワーク上の管理対象デバイスから送られてくる SNMP トラップを捕捉するのも、やはり NMS の役割です。
- 1 つ以上のネットワーク管理アプリケーション (NMA) を同時に監視します。NMS は、管理対象デバイス、テーブルの表示、およびログ記録に関する情報を図を使って表示する機能を備えています。
- NMS に組み込まれている MIB コンパイラで、MIB ファイルをコンパイルすることができます。

管理対象デバイス

SNMP がインストールされているデバイスは、すべて管理対象デバイスとして扱うことができます。ホスト、ルータ、ブリッジ、ハブなどが管理対象デバイスになります。NMS は管理対象デバイスを監視し、またデバイスと通信します。

NMS と管理対象デバイスとの間では、サブエージェントおよびマスタエージェントという 2 種類のエージェントを介して情報をやり取りします。

サブエージェント

サブエージェントには、管理対象デバイスに関する情報を集め、マスタエージェントに渡す役割があります。

マスタエージェント

マスタエージェントには、さまざまなサブエージェントと NMS の間で情報を交換する役割があります。マスタエージェントは、通信相手のサブエージェントと同じホスト上で動作します。

Management Information Base

SNMP では、プロトコルデータ単位 (PDU: Protocol Data Unit) という形でネットワーク情報を交換します。PDU には、管理対象デバイスに保存されている変数に関する情報が含まれています。この変数のことを管理オブジェクトと言い、その値とオブジェクト名が NMS に渡されます。管理オブジェクトはすべて管理情報ベース (MIB) に定義されています。MIB はツリー状の階層構造で表される仮想データベースです。

SNMP のネットワーク管理プロトコル

SNMP の基本関数を次の表に示します。

機能	説明
取得	マネージャがエージェントに情報を要求するために使用します。
Get Next	配列や表から情報を取得する際にマネージャが使用します。
Get Response	マネージャから問い合わせを受けたエージェントが、それに応答するために使用します。
設定	エージェント側の MIB にある変数の値を変更するために、マネージャが使用します。
Trap	あるイベントが発生した際、エージェントがマネージャに通知するために使用します。

SNMP の詳細については、次の Web サイトを参照してください。

- NET-SNMP のホームページ (http://net-snmp.sourceforge.net)
- SNMP FAQ (http://www.faqs.org/faqs/snmp-faq/part1)
- RFC 1157 (http://www.ietf.org/rfc/rfc1157.txt)
- SNMPInfo (https://metacpan.org/pod/SNMP::Info)
- SNMP RFC Standard MIBs and Informative Links (https://www.snmp.com/protocol/snmp_rfcs.shtml)
- RFC 2605 (http://www.ietf.org/rfc/rfc2605.txt?number=2605)

OpenText eDirectory と SNMP

OpenText eDirectory には、ユーザ、アプリケーション、ネットワークデバイス、データなど、数多くのオブジェクトを格納し、管理することができます。OpenText eDirectory で管理するオブジェクトが増えてくると、オブジェクトの追加や変更を追跡する必要性も増してきます。この問題を解決する手段として SNMP を使用することで、ユーザは OpenText eDirectory サーバを監視し、変更を追跡できます。

OpenText eDirectory の管理に SNMP を使う利点

- OpenText eDirectory サーバをリアルタイムに監視
- サードパーティ製 SNMP MIB ブラウザから OpenText eDirectory を監視
- OpenText eDirectory が正常に稼働しているか、ステータスを追跡可能
- 起こりうる問題を検知時に特定し、対処が可能
- トラップや統計に関する設定により、対象を選択して監視可能
- OpenText eDirectory に対するアクセス状況をグラフ表示
- SNMP で収集した履歴データを格納し、分析可能

- SNMP の Get 要求、GetNext 要求を使った統計機能にも対応
- プラットフォームを選ばず SNMP ネイティブマスタエージェントを使用可能

SNMP と OpenText eDirectory の連携について

SNMP を OpenText eDirectory に実装すると、アクセス状況、稼働状況、エラー、キャッシュ性能に関する OpenText eDirectory の統計情報を取得できます。また、イベントが発生すると SNMP からトラップが送信されます。トラップや統計情報は MIB に定義されています。

注 : これらの属性へのアクセスには常にセキュリティ保護されたチャネルを使用すると指定している場合は、暗号化された属性へアクセスする際にセキュリティ保護されたチャネル以外は使用できない可能性があります。詳細については、325 ページの 「暗号化属性」を参照してください。

ディレクトリサービス監視 MIB

OpenText eDirectory の MIB には、OpenText eDirectory を監視するための統計情報やトラップが定義されています。この MIB には次の id が割り当てられています。

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).novell(23).mibDoc(2).ndsMIB(98)

[Statistics (統計情報)]

OpenText eDirectory MIB は、管理対象オブジェクトを次の 4 つの異なるテーブルに分けて格納しています。

- **キャッシュデータベース統計テーブル - ndsDbCacheTable:** ディレクトリサーバに関する記述と、サーバにキャッシュされたエントリに関する統計情報の要約を格納します。
- **設定データベース統計テーブル - ndsDbConfigTable:** ディレクトリサーバに関する記述と、サーバで設定されたエントリに関する統計情報の要約を格納します。
- **プロトコル統計テーブル - ndsProtoIfOpsTable** ディレクトリサーバのアプリケーションプロトコルインタフェースごとに、アクセス状況、稼動状況、エラーに関する統計情報の要約を格納します。
- **相互通信統計テーブル - ndsServerIntTable** 監視対象ディレクトリが通信した、あるいは通信を試みたディレクトリサーバを、最新の「N」回分記録しておきます。「N」はローカルで定義する定数です。

注 : 統計情報の詳細については、577 ページの 「[Statistics (統計情報)]」を参照してください。

トラップ - ndsTrapVariables

OpenText eDirectory MIB には 119 種類のトラップが定義されています。そのうち 117 種類は OpenText eDirectory のイベントにマップされています。ほかに ndsServerStart および ndsServerStop というトラップがあり、これは SNMP サブエージェントが直接生成します。この 2 つのトラップは設定できません。

注 : トラップの詳細については、「552 ページの 「トラップ」」を参照してください。

統計情報とトラップの詳細については、「edir.mib」を参照してください。

edir.mib は次のディレクトリにあります。

Windows: *install_directory*\SNMP
Linux: /etc/opt/novell/eDirectory/conf/ndssnmp/

SNMP グループオブジェクト

SNMP グループオブジェクトは、OpenText eDirectory SNMP トラップの設定や管理に使います。インストールの過程で、「SNMP Group - *server_name*」という名前の SNMP グループオブジェクトが作られます (ここで *server_name* は、OpenText eDirectory の SNMP サービスをインストールしたサーバ名を表します)。この SNMP グループオブジェクトが作られるのは、サーバオブジェクトと同じコンテナ内です。SNMP 設定ユーティリティは、SNMP トラップの設定に使います。

Windows の場合

SNMP グループオブジェクトを作るには、次のコマンドを実行してください。

rundll32 snmpinst, snmpinst -c <createobj> -a *<userFDN>* -p *<password>* -h *<hostname or IP address>*

パラメータ	説明
-c *<createobj>*	オブジェクトの作成を示すトラップコマンド。
-a *<userFDN>*	管理者権利を持つユーザの完全識別名。
-p <password>	認証に使う userFDN パスワード
-h *<hostname or IP address>*	DNS ホスト名または IP アドレス。

例 :

rundll32 snmpinst, snmpinst -c createobj -a admin.mycontext -p mypassword -h 160.98.146.26

SNMP グループオブジェクトを削除するには、次のコマンドを実行してください。

rundll32 snmpinst, snmpinst -c *<deleteobj>* -a *<userFDN>* -p *<password>* -h *<hostname or IP address>*

パラメータについてはオブジェクトを作る場合と同様です。

例 :

rundll32 snmpinst, snmpinst -c deleteobj -a admin.mycontext -p mypassword -h 160.98.146.26

Linux の場合

SNMP グループオブジェクトを作るには、次のコマンドを実行してください。

ndsconfig add -m *<modulename>* -a *<userFDN>*

例 :

ndsconfig add -m snmp -a admin.mycontext

OpenText eDirectory の SNMP サービスのインストールと設定

SNMP service for OpenText eDirectory は、OpenText eDirectory をインストールする際にインストールされます。OpenText eDirectory の SNMP サービスのデフォルト設定の変更には OpenText Identity Console を使います。詳細については、548 ページの 「ダイナミック設定」を参照してください。

SNMP Group-Object という名前の新しいオブジェクトが、OpenText eDirectory のインストール時に、ディレクトリツリーに追加されます。このオブジェクトは OpenText eDirectory SNMP トラップの設定や管理に使います。詳細については、544 ページの 「SNMP グループオブジェクト」を参照してください。

OpenText eDirectory がインストール済みの Windows に SNMP をインストールする手順

OpenText eDirectory のインストール時に SNMP サービスを除外した場合は、SNMP サブエージェント用ファイルがコピーされるだけで、レジストリは元のままになっています。

あとになって SNMP サービスが必要になった場合は、次のコマンドで SNMP サービスをインストールしてレジストリを更新してください。

rundll32 snmpinst, snmpinst -c createreg

SNMP サーバモジュールのロードとアンロード

SNMP サーバモジュールは手動でロード、アンロードできます。デフォルトでは、どのプラットフォームでも、自動でロードされるようになっています。ただし、Windows と Linux では手動でサーバモジュールをロードできます。

SNMP サーバモジュールをロードするには、次のコマンドを入力します。

サーバ	コマンド
Windows	[DHOST (NDSCONS)] 画面で、[**ndssnmp.dlm**] を選択し、[**開始**] をクリックします。
Linux	DHOST のリモート管理ページで、OpenText eDirectory SNMP トラップサーバの開始用操作アイコンをクリックすると、SNMP トラップサーバがロードされます。 または プロンプトで、次のコマンドを入力します。 `/opt/novell/eDirectory/bin/ndssnmp -l`

SNMP サーバモジュールをアンロードするには、次のコマンドを入力します。

サーバ	コマンド
Windows	[DHOST (NDSCONS)] 画面で、[**ndssnmp.dlm**] を選択し、[**停止**] をクリックします。
Linux	DHOST のリモート管理ページで、**OpenText eDirectory 9.2 SNMP トラップサーバ**の停止用操作アイコンをクリックすると、SNMP トラップサーバがアンロードされます。 または プロンプトで、次のコマンドを入力します。 `/opt/novell/eDirectory/bin/ndssnmp -u`

サブエージェントの設定

- 546 ページの 「スタティック設定」
- 548 ページの 「ダイナミック設定」

スタティック設定

スタティック設定は、サブエージェントを実際に稼動させる前に行います。Windows または Linux 上では、ndssnmp.cfg ファイルを編集することによって手動で設定できます。ndssnmp.cfg ファイルは次のディレクトリに配置されています。

Windows: *install_directory*\SNMP\

Linux: /etc/opt/novell/eDirectory/conf/ndssnmp/

注 : ndssnmp.cfg ファイルに変更を加えた場合は、サブエージェントを再起動する必要があります。

サブエージェントの設定は、次のような書式で記述してください。

- INTERACTIVE *status*

  ここで *status* には、「on」または「off」を指定します。「on」を指定すると、サブエージェント起動時にユーザ名とパスワードを要求されるようになります。ステータスが「off」の場合は、ユーザ名とパスワードがセキュリティで保護されたストアから取得されます。デフォルト = オフ .

  例 :

  INTERACTIVE on

  INTERACTIVE off

- INTERACTION *value*

  ここで *value* は、インタラクション表のエントリ数を表します。範囲は 1 ～ 10 です。デフォルトは 4 です。

  例 :

  INTERACTION 4

  INTERACTION 2

- MONITOR *status*

  ここで *status* には、「on」または「off」を指定します。デフォルト = オン .

  例 :

  MONITOR on

  MONITOR off

- SSLKEY *certificate_file*

  ここで *certificate_file* は、証明書のエクスポート先パスを表します。このとき、エクスポートした証明書が実際に存在しているパスを指定してください。

  例 :

  SSLKEY /home/guest/snmp-cert.der (Linux)

  SSLKEY c:\home\guest\snmp-cert.der (Windows)

  注 : 共通の証明書を受け入れない複数のインスタンスを監視している場合は、このオプションがサポートされません。

- SERVER *hostname/IP_address:NCP_port*

  ここで *hostname* は、OpenText eDirectory サーバをインストールし、設定したホスト名を表します。指定できるのは、ローカルにインストールしたサーバに限ります。この指定は必須です。指定がなければどのサーバも監視の対象になりません。デフォルト : ローカルサーバのホスト名。

  例 :

  SERVER myserver

  SERVER myserver:1524

Linux 上で OpenText eDirectory のインスタンスが複数存在する場合、監視するすべての eDirectory サーバを次のように指定できます。

SERVER myserver:1524

SERVER myserver:2524

SERVER myserver:6524

注 : このコマンドで、コロン (:) の前後にはスペースを入れないでください。

ダイナミック設定

ダイナミック設定は、ディレクトリサービスの稼動中、いつでも次のいずれかの方法で実行できます。

コマンドライン

トラップ設定用のコマンドラインユーティリティを使って、OpenText eDirectory の SNMP トラップを設定できます。

次のような操作が可能です。

- トラップの有効化と無効化
- トラップインターバルの設定
- エラートラップの有効化と無効化
- 有効なトラップ、無効なトラップ、またはすべてのトラップの一覧表示

注 : 詳細については、569 ページの 「トラップに関する設定」を参照してください。

OpenText Identity Console アプリケーション

トラップは、OpenText Identity Console を使用して設定することもできます。OpenText Identity Console はブラウザベースのツールで、OpenText eDirectory オブジェクトを運用、管理、設定するために使用します。OpenText Identity Console を使用すると、ユーザに特定のタスクや責任を割り当てたり、それらのタスクを実行するために必要なツールおよびそれに伴う権利だけを付与したりすることができます。

1. OpenText Identity Console で、[**SNMP**] タイルをクリックします。
2. 設定する [SNMP グループ] オブジェクトをクリックします。
3. [**一般**] および [**トラップ**] ドロップダウンメニューで、設定可能なパラメータを指定します。
4. [**保存**] をクリックし、[**OK**] をクリックすると、今設定した内容が保存されます。

注 : 詳細については、「SNMP グループオブジェクトの管理」を参照してください。

OpenText eDirectory の SNMP サービスの設定

このセクションでは、次のプラットフォーム上で OpenText eDirectory の SNMP サービスをセットアップする方法について説明します。

- 549 ページの 「Windows」
- 550 ページの 「Linux」

OpenText eDirectory の SNMP サービスの設定は、次の手順で行います。

1. マスタエージェントの設定
2. マスタエージェントの起動
3. サブエージェントの設定
4. サブエージェントの起動

Windows

- 549 ページの 「マスタエージェントの設定」
- 550 ページの 「マスタエージェントの起動」
- 550 ページの 「マスタエージェントの停止」
- 550 ページの 「サブエージェントの起動」

マスタエージェントの設定

注 : SNMP マスタエージェントは、OpenText eDirectory のインストールに先立ってインストールしておく必要があります。詳細については、「Microsoft SNMP Services (http://technet.microsoft.com/en-us/library/bb726977.aspx)」を参照してください。

---

1 ［Microsoft SNMP Properties］ダイアログボックスを開き、［**エージェント**］タブをクリックします。

2 接続先および場所に関する情報を入力してください。

3 ［トラップ］タブをクリックし、コミュニティ名およびトラップの送り先に関する情報を入力します。

   3a コミュニティ名を入力し、［**追加**］をクリックします。

   3b トラップの送り先のコンピュータの IP アドレスまたはホスト名を入力してください。

   3c ［**追加**］ボタンを押すと、ここで入力した IP アドレスまたはホスト名が追加されます。

4 ［**デスクトップとの対話をサービスに許可する**］オプションを有効にします。

   このオプションが無効のままだと、Windows 上の SNMP には接続できません。

   Windows プラットフォーム上 : ［**スタート**］> ［**設定**］> ［**コントロール パネル**］> ［**管理ツール**］> ［**サービス**］の順にクリックします。［**SNMP**］を右クリックし、［**プロパティ**］を選択します。［**ログオン**］タブで、［**デスクトップとの対話をサービスに許可する**］を選択します。

マスタエージェントの起動

1 マスタエージェントを起動するには、次の操作を実行します。

［スタート］>［設定］>［コントロール パネル］>［管理ツール］>［サービス］>［SNMP］>［開始］の順にクリックします。

2 コマンドプロンプトで次のコマンドを入力します。

Net start SNMP

マスタエージェントの停止

マスタエージェントを停止するには、次のいずれかの操作をしてください。

1 ［スタート］>［設定］>［コントロール パネル］>［管理ツール］>［サービス］>［SNMP］>［停止］の順にクリックします。

2 コマンドプロンプトで次のコマンドを入力します。

Net stop SNMP

サブエージェントの起動

Windows の場合、マスタエージェントを起動すると、サブエージェントも自動的に起動されます。

重要 : 最新の Service Pack は、SNMP サービスのインストール後にインストールする必要があります。

Linux

Linux の場合は、net-snmp をインストールしておく必要があります。デフォルトで、ほとんどの Linux システムにインストールされています。

Linux 上での SNMP サービスのセットアップ

- 550 ページの 「マスタエージェントの設定」
- 551 ページの 「マスタエージェントの起動」
- 551 ページの 「サブエージェントの起動」
- 552 ページの 「サブエージェントの停止」

マスタエージェントの設定

Linux 上でマスタエージェントを設定するには、「551 ページの 「snmpd.conf の変更」」の説明に従って snmpd.conf ファイルを変更します。

snmpd.conf ファイルは、SLES 上の /etc/snmp ディレクトリとその他の Linux プラットフォーム上の /etc ディレクトリに配置されています。

snmpd.conf の変更

snmpd.conf ファイルに、次の行を入力します。

```
trapsink myserver public
```

ここで *myserver* は、トラップの送り先ホスト名を表します。

snmpd.conf ファイルに、次の行を追加します。

```
master agentx
```

また、次のように変更してください。

元の記述	変更後の記述
com2sec notConfigUser default public	com2sec demouser default public
group notConfigGroup v1 notConfigUser	group demogroup v1 demouser
view systemview included system	view all included .1
access notConfigGroup "" any noauth exact systemview none none	access demogroup "" any noauth exact all all all

上記の内容が snmpd.conf ファイルに存在しない場合は追加してください。

重要 : 設定ファイルを書き替えた場合、マスタエージェントとサブエージェントを再起動する必要があります。

マスタエージェントの起動

マスタエージェントを起動するには、次のコマンドを実行してください。

```
/usr/sbin/snmpd -C -c /etc/snmpd.conf
```

注 : SLES 12 以上でマスタエージェントを開始するには、/etc/init.d/snmpd start コマンドを実行します。

サブエージェントの起動

サブエージェントを起動するには、次のコマンドを実行してください。

```
/etc/init.d/ndssnmpsa start
```

プロンプトが表示されたら、ユーザ名とパスワードを入力します。正常に認証されれば、/etc/opt/novell/eDirectory/conf/ndssnmp/ndssnmp.cfg ファイルで INTERACTION = ON という設定になっている場合、次のようなメッセージが表示されます。

```
Do you want to remember password? (Y/N)
```

ここで「Y」と入力すると、パスワードが保存されます。次回からは、パスワードを入力することなくサブエージェントを起動できます。

「N」と入力した場合は、次回からもパスワードを求められます。

注 : サーバがダウンすると、マスタエージェントとサブエージェントもダウンします。したがって、サーバのリブート時にマスタエージェントとサブエージェントを開始するには、次のコマンドを実行します。

```
chkconfig snmpd on
chkconfig ndssnmpsa on
```

サブエージェントの停止

サブエージェントを停止するには、次のコマンドを実行してください。

/etc/init.d/ndssnmpsa stop

SNMP による OpenText eDirectory の監視

OpenText eDirectory の動作を監視するために、SNMP の機能であるトラップや統計を使うことができます。

そのためには、NCP サーバ、LDAP グループ、LDAP サーバオブジェクトに対して、次のような権利が必要です。

- NCP サーバオブジェクトに対するスーパバイザ権
- LDAP グループオブジェクトの「LDAP クリアテキストパスワードを許可する」属性に対する読み出し権利
- LDAP サーバオブジェクトの LDAP TCP Port 属性、LDAP SSL Port 属性に対する読み出し権利

通常、管理者としてログインしたユーザならば、SNMP により OpenText eDirectory サーバを監視する上で問題が生じることはありません。

トラップ

ndsServerStart (2001) および ndsServerStop (2002) を設定できなかったトラップの中から、119 個のトラップが SNMP コンポーネントによって生成されます。これらのトラップはデフォルトで有効です。

トラップの生成状況は MIB ブラウザで確認できます。

トラップ番号	トラップ名	生成される条件
1	ndsCreateEntry	新規オブジェクトがディレクトリに追加されたとき。 例 : LDAP ツール、ICE、または OpenText Identity Console を使って、オブジェクトを作成したとき。

トラップ番号	トラップ名	生成される条件
2	ndsDeleteEntry	オブジェクトが削除されたとき。 例 : LDAP ツール、ICE、または OpenText Identity Console を使って、オブジェクトを作成したとき。
3	ndsRenameEntry	オブジェクト名が変更されたとき。 例 : LDAP ツール、ICE、または OpenText Identity Console を使って、オブジェクトの名前を変更したとき。
4	ndsMoveSourceEntry	オブジェクトのコンテキストが変わったとき。このトラップにより、変更前のコンテキストが通知されます。 例 : ldapmodrdn や ldapsdk でオブジェクトを移動したとき。
5	ndsAddValue	オブジェクトの属性値が追加されたとき。 例 : LDAP ツール、ICE、または OpenText Identity Console を使って、属性に新しい値を追加したとき。 **注 :** 返される値が NULL の場合は、セキュリティ保護されたチャネルを経由してディレクトリにアクセスする必要があります。詳細については、568 ページの 「暗号化属性にアクセスする」を参照してください。
6	ndsDeleteValue	オブジェクトの属性値が削除されたとき。 例 : LDAP ツール、ICE、または OpenText Identity Console を使って、属性の値を削除したとき。 **注 :** 返される値が NULL の場合は、セキュリティ保護されたチャネルを経由してディレクトリにアクセスする必要があります。詳細については、568 ページの 「暗号化属性にアクセスする」を参照してください。
7	ndsCloseStream	ストリーム属性が変更されたとき。

トラップ番号	トラップ名	生成される条件
8	ndsDeleteAttribute	オブジェクトの属性値 (単一値と定義されているもの) が削除されたとき。 例 : LDAP ツール、ICE、または OpenText Identity Console を使って、属性を削除したとき。 **注 :** 返される値が NULL の場合は、セキュリティ保護されたチャネルを経由してディレクトリにアクセスする必要があります。詳細については、568 ページの 「暗号化属性にアクセスする」を参照してください。
9	ndsCheckSecurityEquiv	あるエントリの同等セキュリティベクトルが検査されたとき。 例 : LDAP ツール、ICE、または OpenText Identity Console を使って、同等セキュリティ属性を変更したとき。
10	ndsUpdateSecurityEquiv	あるエントリの同等セキュリティベクトルが変更されたとき。 例 : LDAP ツール、ICE、または OpenText Identity Console を使って、同等セキュリティ属性を変更したとき。
11	ndsMoveDestEntry	オブジェクトのコンテキストが変わったとき。このトラップにより、変更後のコンテキストが通知されます。 例 : ldapmodrdn や ldapsdk でオブジェクトを移動したとき。
12	ndsDeleteUnusedExtref	バックリンクオブジェクトが削除されたとき。
13	ndsAgentOpenLocal	ローカルディレクトリエージェントがオープンされたとき。 例 : 標準修復を実行したとき。
14	ndsAgentCloseLocal	ローカルディレクトリエージェントがクローズされたとき。 例 : 標準修復を実行したとき。
15	ndsDSABadVerb	DSAgent 要求に関連づけられたバーブ番号が正しくないとき。 例 : DClient 呼び出しを使って、OpenText eDirectory に不正なバーブ要求を送ったとき。

トラップ番号	トラップ名	生成される条件
16	ndsMoveSubtree	コンテナオブジェクトがそれに含まれるオブジェクトと共に移動されたとき。 例 : LDAP ツール、ICE、または OpenText Identity Console を使って、パーティションを他のコンテキストに移動したとき。
17	ndsNoReplicaPointer	レプリカにレプリカポインタが関連づけられていないとき。
18	ndsSyncInEnd	インバウンド同期が終了したとき。
19	ndsBacklinkSecurEquiv	バックリンク操作により、オブジェクトの同等セキュリティベクトルが更新されたとき。 例 : LDAP ツール、ICE、または OpenText Identity Console を使って、同等セキュリティ属性を変更したとき。
20	ndsBacklinkOperPrivChg	バックリンク操作により、オブジェクトのコンソールオペレータ権利が変更されたとき。
21	ndsDeleteSubtree	コンテナオブジェクトがそれに含まれるオブジェクトと共に削除されたとき。
22	ndsReferral	参照が作成されたとき。
23	ndsUpdateClassDef	スキーマクラス定義が更新されたとき。 例 : 新規クラスまたは属性をプライマリサーバに追加し、セカンダリサーバ側で、LDAP ツール、ICE、または OpenText Identity Console を使って同期を取ったとき。
24	ndsUpdateAttributeDef	スキーマ属性定義が更新されたとき。 例 : 新規属性をプライマリサーバに追加し、セカンダリサーバ側で、LDAP ツール、ICE、または OpenText Identity Console を使って同期を取ったとき。
25	ndsLostEntry	OpenText eDirectory のローカルサーバには存在しないはずのエントリに対する更新要求があったとき。■分節拡張■■分節統合■
26	ndsPurgeEntryFail	パージ処理に失敗したとき。
27	ndsPurgeStart	パージ処理を開始したとき。 例 : DSTrace を実行し、ndstrace=*j を設定したとき。

トラップ番号	トラップ名	生成される条件
28	ndsPurgeEnd	パージ処理が終了したとき。 例 : DSTrace を実行し、ndstrace=*j を設定したとき。
29	ndsLimberDone	limber 処理が終了したとき。 例 : 一定時間の経過後にリンバを起動するように DSTrace を設定したとき。
30	ndsPartitionSplitDone	パーティション分割処理が終了したとき。 例 : OpenText Identity Console を使ってパーティションを作成したとき。
31	ndsSyncServerOutStart	ある特定のサーバに同期するアウトバウンド同期処理が起動されたとき。 例 : 一定時間の経過後にアウトバウンド同期処理を開始するように DSTrace を設定したとき。
32	ndsSyncServerOutEnd	ある特定のサーバに同期するアウトバウンド同期処理が終了したとき。 例 : 一定時間の経過後にアウトバウンド同期処理を停止するように DSTrace を設定したとき。
33	ndsSyncPartitionStart	パーティション同期処理が起動されたとき。 例 : あるコンテナをパーティション分割したとき。
34	ndsSyncPartitionEnd	パーティション同期処理が終了したとき。 例 : あるコンテナをパーティション分割したとき。
35	ndsMoveTreeStart	サブツリーの移動処理が起動されたとき。 パーティションの移動に伴い、サブツリーも移動します。 例 : OpenText Identity Console を使ってパーティションを作成し、それを他のコンテナに移動したとき。

トラップ番号	トラップ名	生成される条件
36	ndsMoveTreeEnd	サブツリーの移動処理が終了したとき。 パーティションの結合処理に伴い、サブツリーも移動します。 例 : OpenText Identity Console を使ってパーティションを作成し、それを他のコンテナに移動したとき。
37	ndsJoinPartitionDone	パーティションの結合処理が終了したとき。 例 : OpenText Identity Console を使ってパーティションを作成し、それを他のパーティションとマージしたとき。
38	ndsPartitionLocked	パーティションがロックされたとき (結合処理の際など)。 例 : OpenText Identity Console を使ってパーティションを作成したとき。
39	ndsPartitionUnlocked	パーティションのロックが解除されたとき (結合処理が終了したときなど)。 例 : OpenText Identity Console を使ってパーティションを作成したとき。
40	ndsSchemaSync	スキーマの同期処理が起こったとき。 例 : ldapsdk schsync を使用してスキーマの同期処理をスケジュールしたとき。
41	ndsNameCollision	他のサーバにある別のオブジェクトと名前が重複 (衝突) したとき。 例 : iMonitor を使い、プライマリサーバとセカンダリサーバのアウトバウンド同期処理を無効にした状態で、LDAP ツールを使って両サーバにユーザオブジェクトを作成したとします。こうしておいて、iMonitor でアウトバウンド同期処理を実行すると、衝突が起こります。
43	ndsChangeModuleState	eDirectory モジュール (NLM/DLM) がロードまたはアンロードされたとき。 例 : nldap モジュールをロードまたはアンロードしたとき。

トラップ番号	トラップ名	生成される条件
44	ndsLumberDone	バックグラウンドでの limber 処理が始まったとき。
45	ndsBacklinkProcDone	バックリンク処理が終了したとき。 例 : 一定時間の経過後にバックリンクを起動するように DSTrace を設定したとき。
46	ndsServerRename	サーバ名が変更されたとき。 例 : ldapmodrdn または ldapsdk を使って、サーバ名を変更したとき。
47	ndsSyntheticTime	将来のタイムスタンプを付与したオブジェクトが作成されたとき。eDirectory サーバの同期に際しては、合成時刻として扱うことになります。 例 : ndsconfig を使って、セカンダリサーバをツリーに追加したとき。
48	ndsServerAddressChange	limber 処理によりサーバ参照が変更されたとき。 例 : サーバの IP アドレスを変更して ndsd を再起動したとき。
49	ndsDSARead	エントリが読み出されたとき。 OpenText eDirectory を対象とする操作があれば、必ずこのトラップが生成されます。 例 : ldapsearch を実行したとき。
50	ndsLogin	OpenText eDirectory へのログインがあったとき。 例 : ndslogin を使ってツリーにログインしたとき。
51	ndsChangePassword	パスワードが変更されたとき。 例 : ldapmodify を使ってユーザオブジェクトのパスワードを変更したとき。

トラップ番号	トラップ名	生成される条件
52	ndsLogout	OpenText eDirectory からログアウトされたとき。 例 : Novell Client からツリーへの接続を切ったとき。
53	ndsAddReplica	サーバパーティションにレプリカが追加されたとき。 例 : ndsconfig を使ってツリーに新規レプリカを追加したとき。
54	ndsRemoveReplica	レプリカが削除されたとき。 例 : OpenText Identity Console を使って、あるサーバからレプリカを削除したとき。
55	ndsSplitPartition	パーティションが分割されたとき。 例 : OpenText Identity Console を使ってパーティションを作成したとき。
56	ndsJoinPartition	ペアレントパーティションにチャイルドパーティションが結合されたとき。 例 : OpenText Identity Console を使ってパーティションを作成し、そのパーティションを結合したとき。
57	ndsChangeReplicaType	パーティションレプリカのタイプが変更されたとき。 例 : レプリカタイプをマスタから読み書き用に変更したとき。
58	ndsAddEntry	新規オブジェクトが追加されたとき。 例 : OpenText Identity Console を使ってユーザオブジェクトを追加したとき。
59	ndsAbortPartitionOp	パーティションに関する処理が中断されたとき。 例 : コンテナのパーティション分割処理を実行し、途中で中断したとき。

トラップ番号	トラップ名	生成される条件
60	ndsRecvReplicaUpdates	同期処理中に、レプリカ側で更新通知を受け取ったとき。 例 : 複数のサーバツリーから成る構成の OpenText eDirectory サーバが、保持しているレプリカの更新を要求したとき。この操作は、OpenText Identity Console を使って実行できます。
61	ndsRepairTimeStamps	レプリカのタイムスタンプが修復されたとき。 例 : DSRepair (Linux の場合は ndsrepair、Wiindows の場合は NSDCons) を使って、タイムスタンプの DIB 修復処理を実行したとき。
62	ndsSendReplicaUpdates	同期処理中にレプリカが更新されたとき。 例 : 複数のサーバツリーから成る構成の OpenText eDirectory サーバが、保持しているレプリカの更新要求を送ったとき。この操作は、OpenText Identity Console を使って実行できます。
63	ndsVerifyPass	パスワードが正しいと確認されたとき。 例 : 無効になったパスワードを変更する際、確認のために入力させた旧パスワードが正しいと確認されたとき。
64	ndsBackupEntry	エントリがバックアップされたとき。 例 : バックアップユーティリティ (Linux の場合は ndsbackup、Windows の場合は NDSCons) を使って、Directory オブジェクトをバックアップしたとき。
65	ndsRestoreEntry	エントリが復元されたとき。 例 : バックアップユーティリティ (Linux の場合は ndsbackup、Windows の場合は NDSCons) を使って、バックアップされていた Directory オブジェクトを復元したとき。

トラップ番号	トラップ名	生成される条件
66	ndsDefineAttributeDef	スキーマに属性定義が追加されたとき。 例 : OpenText eDirectory ツリースキーマに新しい属性定義を追加して拡張したとき。ZENWorks® や NMAS™ などの OpenText eDirectory 依存アプリケーションをインストールすると、スキーマを拡張できます。また、OpenText Identity Console または Linux 上のスキーマ拡張ユーティリティ ndssch を使って拡張することもできます。
67	ndsRemoveAttributeDef	スキーマから属性定義が削除されたとき。 例 : OpenText eDirectory ツリースキーマから属性定義を削除したとき。属性は、OpenText Identity Console または Linux 上のスキーマ拡張ユーティリティ ndssch を使って削除できます。
68	ndsRemoveClassDef	スキーマからクラス定義が削除されたとき。 例 : OpenText eDirectory ツリースキーマからオブジェクトクラス定義を削除したとき。これは、OpenText Identity Console または Linux 上のスキーマ拡張ユーティリティ ndssch を使って削除できます。
69	ndsDefineClassDef	スキーマにクラス定義が追加されたとき。 例 : OpenText eDirectory ツリースキーマに新しいクラスを追加して拡張したとき。ZENWorks や NMAS などの OpenText eDirectory 依存アプリケーションをインストールすると、スキーマを拡張できます。また、OpenText Identity Console または Linux 上のスキーマ拡張ユーティリティ ndssch を使って拡張することもできます。
70	ndsModifyClassDef	クラス定義が変更されたとき。 例 : 既存のオブジェクトクラスや属性定義を変更したとき。
71	ndsResetDSCounters	OpenText eDirectory に内蔵されたカウンタがリセットされたとき。
72	ndsRemoveEntryDir	エントリに関連づけられたディレクトリが削除されたとき。

トラップ番号	トラップ名	生成される条件
73	ndsCompAttributeValue	属性値が比較されたとき。 例 : 属性値を他のオブジェクトの属性値と比較したとき。ユーザオブジェクトを対象とする LDAP 検索により、入力された値と電話番号が一致するかどうか検査するような場合。
74	ndsOpenStream	ストリーム属性がオープンまたはクローズされたとき。 例 : ストリームを読み込みまたは書き込み操作用に作成またはオープンしたとき。ユーザオブジェクト用のログインスクリプトを作成したとき。DIB ディレクトリ以下にファイルが生成される結果、このトラップが発生します。
75	ndsListSubordinates	コンテナオブジェクトに対して、それに含まれるエントリのリストを取得する処理が実行されたとき。この処理では、コンテナオブジェクトの直下にあるエントリのみが検索の対象となります。 例 : OpenText Identity Console を使って、コンテナオブジェクトをクリックすることにより、これに含まれるオブジェクトを一覧表示しようとしたとき。
76	ndsListContainerClasses	エントリに対して、これを含めることができるクラスのリストを取得する処理が実行されたとき。 例 : あるオブジェクトについて、これを含めることができるクラス (コンテナクラス) を一覧表示しようとしたとき。 ユーザオブジェクトを含めることができるコンテナクラスを検索すれば、Organization(組織)、Organization Unit(部署)、Domain(ドメイン) などといったクラスが表示されるはずです。
77	ndsInspectEntry	エントリの検査処理が実行されたとき。 例 : あるエントリの情報を取得し、そのエントリでこれまでにエラーが発生したことがあるかどうかを検査しようとしたとき。バックグラウンドで OpenText eDirectory のフラットクリーナ処理を実行する際にこのイベントが発生する結果、トラップが送られることになります。

トラップ番号	トラップ名	生成される条件
78	ndsResendEntry	エントリの再送信処理が実行されたとき。 例 : レプリカの作成処理中に、エントリの再送信が起こったとき。サーバ間の接続に問題があり、オブジェクトの送信に失敗したような場合に起こります。
79	ndsMutateEntry	エントリの変換処理が実行されたとき。 例 : バインダリオブジェクトクラスからユーザオブジェクトクラスに変換したとき。
80	ndsMergeEntries	2 つのエントリがマージされたとき。 例 : 2 つのユーザオブジェクトをマージしたとき。Entry2 (ndsEntryName2) を Entry (ndsEntryName) にマージしたとき。
81	ndsMergeTree	2 つの OpenText eDirectory ツリーがマージされたとき。 例 : DSMerge (Linux の場合は ndsmerge、Windows の場合は NDSCons) を使って、2 つの eDirectory ツリーをマージしたとき。
82	ndsCreateSubref	サブオーディネートリファレンスが生成されたとき。 例 : チャイルドパーティションのレプリカをサーバから削除したとき。自動的にサブオーディネートリファレンスのレプリカが生成され、その結果トラップが発生します。
83	ndsListPartitions	パーティションのリスト取得処理が実行されたとき。 例 : OpenText Identity Console を使って、パーティションビューやスキーマビューで OpenText eDirectory サーバオブジェクトをクリックすることにより、サーバ上のパーティションを一覧表示したとき。
84	ndsReadAttribute	属性値が読み出されたとき。 例 : ツリーの検索処理を実行したとき。
85	ndsReadReferences	エントリの参照が読み出されたとき。

トラップ番号	トラップ名	生成される条件
86	ndsUpdateReplica	パーティションレプリカに対して、レプリカの更新処理が実行されたとき。 例 : あるサーバからユーザを削除したとき。この削除操作に対して他のレプリカが更新されます。
87	ndsStartUpdateReplica	パーティションレプリカに対して、レプリカの更新開始処理が開始されたとき。 例 : あるサーバからユーザを削除したとき。この削除操作に対して他のレプリカが更新されます。
88	ndsEndUpdateReplica	パーティションレプリカに対して、レプリカの更新処理が実行され、終了したとき。 例 : あるサーバからユーザを削除したとき。この削除操作に対して他のレプリカが更新されます。
89	ndsSyncPartition	パーティションレプリカに対して、パーティションの同期処理が実行されたとき。 例 : あるパーティションからユーザを削除したとき。DSTrace を使うと同期処理の状況を見ることができます。
90	ndsSyncSchema	ルートのマスタレプリカが、そのスキーマをサーバと同期させる要求を受け取ったとき。 例 : OpenText Identity Console、LDAP ツール、または ndssch ユーティリティを使って、新規クラスを追加したとき。
91	ndsCreateBackLink	バックリンクが生成されたとき。ローカルに存在しないオブジェクトが参照された場合に、バックリンクが作成されます。 例 : マルチサーバシナリオで、複数のユーザを含むパーティションを作成したとき。サーバの 1 つからこのパーティションを削除すると、サブオーディネートリファレンスが生成されます。このとき、削除されたパーティションに存在していたユーザを参照するバックリンクが生成されます。

トラップ番号	トラップ名	生成される条件
93	ndsChangeTreeName	ツリー名が変更されたとき。 例 : マージユーティリティ DSMerge/ndsmerge を使ってツリー名を変更したとき。
94	ndsStartJoinPartition	パーティションの結合処理が始まったとき。 例 : LDAP ツールを使って、パーティションをマージまたは結合したとき。
95	ndsAbortJoinPartition	パーティションの結合処理が中断されたとき。 例 : LDAP ツールを使って、パーティションをマージまたは結合したとき。
96	ndsUpdateSchema	スキーマ更新処理が実行されたとき。 例 : OpenText Identity Console、LDAP ツール、または ndssch を使って、新規クラスを追加したとき。
97	ndsStartUpdateSchema	スキーマ更新処理が開始されたとき。 例 : OpenText Identity Console、LDAP ツール、または ndssch を使って、新規クラスを追加したとき。
98	ndsEndUpdateSchema	スキーマ更新処理が終了したとき。 例 : OpenText Identity Console、LDAP ツール、または ndssch を使って、新規クラスを追加したとき。
99	ndsMoveTree	ツリー移動処理が実行されたとき。 例 : パーティションをあるコンテナから別のコンテナに移動したとき。
101	ndsConnectToAddress	特定のアドレスとの間で接続が確立されたとき。 例 : OpenText Identity Console を使ってツリーをブラウズしたとき。

トラップ番号	トラップ名	生成される条件
102	ndsSearch	検索処理が実行されたとき。 例 : LDAP ツールを使ってツリーに対する ldapsearch を実行したとき。
103	ndsPartitionStateChange	パーティションが作成または削除されたとき。 例 : 新規パーティションを作成する .
104	ndsRemoveBacklink	使われていない外部参照が削除され、該当するオブジェクトを保持しているサーバに対して、バックリンク削除要求が送られたとき。
105	ndsLowLevelJoinPartition	パーティションの結合処理中に、低レベルの結合処理が実行されたとき。 例 : OpenText Identity Console、または LDAP ツールを使って、パーティションをマージまたは結合したとき。
106	ndsCreateNameBase	OpenText eDirectory ネームベースが作成されたとき。
107	ndsChangeSecurityEquals	同等セキュリティ属性が変更されたとき。 例 : OpenText Identity Console を使って、任意のユーザの同等セキュリティを変更し、admin と同じにしたとき。
108	ndsRemoveEntry	OpenText eDirectory からエントリが削除されたとき。 例 : OpenText Identity Console を使ってユーザを削除したとき。
109	ndsCRCFailure	断片化した NCP 要求を構成し直す際に、CRC(冗長巡回検査) エラーが発生したとき。
110	ndsModifyEntry	OpenText eDirectory エントリが変更されたとき。 例 : OpenText Identity Console を使って、任意のユーザの属性を変更したとき。
111	ndsNewSchemaEpoch	スキーマが DSRepair でリセットされたとき。 例 : Linux 上で、ndsrepair -S-Ad を使って新規スキーマエポックを作成したとき。

トラップ番号	トラップ名	生成される条件
112	ndsLowLevelSplitPartition	パーティションを作成する際に、低レベル分割処理が実行されたとき。 例 : OpenText Identity Console、または LDAP ツールを使って、パーティションを作成したとき。
113	ndsReplicaInTransition	レプリカが追加または削除されたとき。
114	ndsAclModify	オブジェクトのトラスティが変更された、すなわち ACL(アクセス制御リスト) オブジェクトが変更されたとき。 例 : LDAP ツール、ICE、または OpenText Identity Console を使って、オブジェクトのトラスティを追加、変更、削除したとき。
115	ndsLoginEnable	ユーザアカウントを有効にする要求をサーバから受け取ったとき。 例 : LDAP ツール、ICE、または OpenText Identity Console を使って、アカウント無効属性を有効にしたとき。
116	ndsLoginDisable	ユーザアカウントを無効にする要求をサーバから受け取ったとき。 例 : LDAP ツール、ICE、または OpenText Identity Console を使って、アカウント無効属性を無効にしたとき。
117	ndsDetectIntruder	不正侵入を検出したため、ユーザアカウントがロックされたとき。 例 : LDAP ツール、ICE、または OpenText Identity Console を使って、不正侵入者属性によってロックされたとき。

トラップ番号	トラップ名	生成される条件
2001	ndsServerStart	サブエージェントが OpenText eDirectory サーバに正常に再接続できたとき。このトラップには 2 つの変数が含まれています。 ◆ ndsTrapTime: この変数は、サブエージェントが OpenText eDirectory サーバに正常に再接続した時刻を、1970 年 1 月 1 日午前 0 時 (万国標準時) からの経過秒数で表します。 ◆ ndsServerName: サブエージェントが正常に再接続した OpenText eDirectory サーバを表します。 例 : サブエージェントが稼働したままの状態でいったん OpenText eDirectory サーバを停止し、再び起動したとき。
2002	ndsServerStop	サブエージェントと OpenText eDirectory サーバとの接続が失われたとき。このトラップには 2 つの変数が含まれています。 ◆ ndsTrapTime: サブエージェントが eDirectory サーバと切断された時刻を、1970 年 1 月 1 日午前 0 時 (万国標準時) からの経過秒数で表します。 ◆ ndsServerName: サブエージェントがそれまで接続されていた eDirectory サーバを表します。 例 : サブエージェントが稼働中に OpenText eDirectory サーバを停止したとき。

暗号化属性にアクセスする

OpenText eDirectory では、特定の機密データをディスクに保存する場合や、ネットワーク上からそのデータにアクセスする場合に、データを暗号化して保護できます。暗号化属性へのアクセスにセキュリティ保護されたチャネルを使用する場合は、事前にそれを指定できます。詳細については、332 ページの 「暗号化属性にアクセスする」を参照してください。

暗号化属性へのアクセスにセキュリティ保護されたチャネルのみを使用すると指定している場合は、NDS 値イベントがブロックされます。値イベントに関連するトラップの値データは NULL になり、暗号化属性の値を取得するにはセキュリティ保護されたチャネルが必要であるということを示すエラー「-6089」が返されます。値データが NULL になるトラップは次のとおりです。

- ndsAddValue
- ndsDeleteValue
- ndsDeleteAttribute

トラップに関する設定

トラップに関する設定の手順はプラットフォームによって異なります。

Platform	ユーティリティ
Windows	ndssnmpcfg
Linux	ndssnmpconfig

Windows

Windows では、ndssnmpcfg を使ってトラップに関する設定を行います。このユーティリティは *install_path*\ ディレクトリにあります。このユーティリティの機能としては、トラップの有効 / 無効の切り替え、各トラップ間の時間間隔の設定、デフォルトの時間間隔の設定、操作に失敗した場合のトラップの有効化、すべてのトラップのリスト表示などがあります。

使用率 :

ndssnmpcfg -h [*hostname*[:*port*]] -p *password* -a *userFDN* -c *command*

パラメータ	説明
-h	DNS ホスト名または IP アドレス。
-p	認証に使う userFDN パスワード
-a	管理者権限を持つユーザの完全識別名
-c	トラップコマンド (「570 ページの 「Windows のトラップコマンド」」を参照)

Windows のトラップコマンド

トラップコマンド	説明	使用法
DISABLE	トラップを無効にするコマンド。NMS は、トラップが送られてきても受け取らないようになります。	特定のトラップ (次の例では 10 番、11 番、100 番) を無効にしたい場合 : ndssnmpcfg "DISABLE 10, 11, 100" 特定のトラップ (次の例では 10 番、11 番、100 番) 以外をすべて無効にしたい場合 : ndssnmpcfg "DISABLE ID != 10, 11, 100" ある範囲の番号のトラップ (次の例では 20 ～ 29 番) を無効にしたい場合 : ndssnmpcfg "DISABLE 20-29" トラップをすべて無効にしたい場合 : ndssnmpcfg "DISABLE ALL"
ENABLE	トラップを有効にするコマンド。NMS は、送られてきたトラップを受け取るようになります。	ndssnmpcfg "ENABLE *trapSpec*" *trapSpec* は次のいずれかの形式で指定してください。 特定のトラップ (次の例では 10 番、11 番、100 番) を有効にしたい場合 : ndssnmpcfg "ENABLE 10, 11, 100" 特定のトラップ (次の例では 10 番、11 番、100 番) 以外をすべて有効にしたい場合 : ndssnmpcfg "ENABLE ID != 10, 11, 100" ある範囲の番号のトラップ (次の例では 20 ～ 29 番) を有効にしたい場合 : ndssnmpcfg "ENABLE 20-29" トラップをすべて有効にしたい場合 : ndssnmpcfg "ENABLE ALL"

トラップコマンド	説明	使用法
INTERVAL	時間間隔を設定する、または表示するためのコマンド。 ここでいう時間間隔とは、同じトラップを繰り返し送る場合に、何秒間の間隔をおくかを表すものです。 0 ～ 2,592,000 の範囲 (秒単位) で設定してください。 この範囲外の値を指定した場合、デフォルトの時間間隔が指定されたものとみなします。 設定値を 0 とすれば、トラップがすべて送られるようになります。	時間間隔の設定値を表示する場合 : ndssnmpcfg "213,240,79 INTERVAL" 複数のトラップについて時間間隔を設定する場合 (次の例では 12 番、17 番、101 番トラップについて 5 秒と設定): ndssnmpcfg "12 17 101 INTERVAL 5" デフォルトの時間間隔を表示する場合 : ndssnmpcfg "DEFAULT INTERVAL" デフォルトの時間間隔を設定する場合 : ndssnmpcfg "DEFAULT INTERVAL=10"

トラップコマンド	説明	使用法
LIST	ある条件を満たすトラップ番号を一覧表示するコマンド。	ndssnmpcfg LIST *trapSpec* *trapSpec* には、トラップ番号のほか、以下に述べるキーワードで条件を指定できます。 ALL、ENABLED、DISABLED、FAILED、または論理式 例 : 有効なトラップをすべて、名称を添えて表示 : ndssnmpcfg LIST ENABLED 無効なトラップをすべて、名称を添えて表示 : ndssnmpcfg LIST DISABLED 117 種類すべてのトラップについて、名称を添えて表示 : ndssnmpcfg LIST ALL 特定の番号 (次の例では 12 番、224 番、300 番) のトラップについて、名称を添えて表示 : ndssnmpcfg LIST ID = 12,224,300 特定の番号 (次の例では 12 番、224 番、300 番) 以外のトラップをすべて、名称を添えて表示 : ndssnmpcfg LIST ID != 12,224,300 有効なエラートラップをすべて、名称を添えて表示 : ndssnmpcfg LIST FAILED

トラップコマンド	説明	使用法
READ_CFG	環境設定ファイル ndstrap.cfg を参照して、ディレクトリ構成を再設定するコマンド。 環境設定ファイルの設定に加えられた変更はすべて有効になります。いくつものコマンドを ndstrap.cfg に記述しておき、ひとまとめにして実行する、という使い方を主として想定したユーティリティです。 ndstrap.cfg は*インストールディレクトリ*\SNMP にあります。 ndstrap.cfg ファイルでは、トラップの環境設定に使用されるオペレーショナルパラメータを設定し、SNMP トラップの操作を設定する方法を指定します。トラップの環境設定ユーティリティ ndssnmpcfg が READ_CFG コマンドで実行される場合には、必ずこのファイルが読み込まれます。	ndssnmpcfg "READ_CFG"
FAILURE	エラートラップをすべて表示するコマンド。 エラートラップとは、イベントの失敗時に生成されるトラップのことです。 **注 :** エラートラップをいったん無効にし、「enable trapid」コマンドで再び有効にすると、エラートラップではなく、正常に処理されたことを表すトラップとして扱われるようになります。	ndssnmpcfg "FAILURE *trapSpec*" *trapSpec* には、トラップ番号をコンマまたはスペースで区切って指定するほか、キーワード「ALL」や論理式を指定できます。例 : 複数のトラップをエラートラップと設定 : ndssnmpcfg "FAILURE 10,11,100" 指定した番号以外のすべてのトラップをエラートラップと設定 : ndssnmpcfg "FAILURE ID != 24,30" すべてのトラップをエラートラップと設定 : ndssnmpcfg "FAILURE ALL"

Linux

Linux では、ndssnmpconfig を使ってトラップに関する設定を行います。これは /etc/ndssnmp/ ディレクトリにあります。このユーティリティの機能としては、トラップの有効 / 無効の切り替え、各トラップ間の時間間隔の設定、デフォルトの時間間隔の設定、操作に失敗した場合のトラップの有効化、すべてのトラップのリスト表示などがあります。

使用率 :

ndssnmpconfig -h [*hostname*[*:port*]] -p *password* -a *userFDN* -c *command*

パラメータ	説明
-h	DNS ホスト名または IP アドレス。
-p	認証に使う userFDN パスワード
-a	管理者権利を持つユーザの完全識別名。
-c	トラップコマンド (「574 ページの 「Linux トラップコマンド」」を参照)

Linux トラップコマンド

トラップコマンド	説明	使用法
DISABLE	トラップを無効にするコマンド。NMS は、トラップが送られてきても受け取らないようになります。	特定のトラップ (次の例では 10 番、11 番、100 番) を無効にしたい場合 : ndssnmpconfig "DISABLE 10, 11, 100" 特定のトラップ (次の例では 10 番、11 番、100 番) 以外をすべて無効にしたい場合 : ndssnmpconfig "DISABLE ID != 10, 11, 100" ある範囲の番号のトラップ (次の例では 20 ～ 29 番) を無効にしたい場合 : ndssnmpconfig "DISABLE 20-29" トラップをすべて無効にしたい場合 : ndssnmpconfig "DISABLE ALL"

トラップコマンド	説明	使用法
ENABLE	トラップを有効にするコマンド。NMS は、送られてきたトラップを受け取るようになります。	ndssnmpconfig "ENABLE *trapSpec*" *trapSpec* は次のいずれかの形式で指定してください。 特定のトラップ (次の例では 10 番、11 番、100 番) を有効にしたい場合 : ndssnmpconfig "ENABLE 10, 11, 100" 特定のトラップ (次の例では 10 番、11 番、100 番) 以外をすべて有効にしたい場合 : ndssnmpconfig "ENABLE ID != 10, 11, 100" ある範囲の番号のトラップ (次の例では 20 〜 29 番) を有効にしたい場合 : ndssnmpconfig "ENABLE 20-29" トラップをすべて有効にしたい場合 : ndssnmpconfig "ENABLE ALL"
INTERVAL	時間間隔を設定する、または表示するためのコマンド。 ここでいう時間間隔とは、同じトラップを繰り返し送る場合に、何秒間の間隔をおくかを表すものです。 0 〜 2,592,000 の範囲 (秒単位) で設定してください。 この範囲外の値を指定した場合、デフォルトの時間間隔が指定されたものとみなします。 設定値を 0 とすれば、トラップがすべて送られるようになります。	時間間隔の設定値を表示する場合 : ndssnmpconfig "213,240,79 INTERVAL" 複数のトラップについて時間間隔を設定する場合 (次の例では 12 番、17 番、101 番トラップについて 5 秒と設定): ndssnmpconfig "12 17 101 INTERVAL 5" デフォルトの時間間隔を表示する場合 : ndssnmpconfig "DEFAULT INTERVAL" デフォルトの時間間隔を設定する場合 : ndssnmpconfig "DEFAULT INTERVAL=10"

トラップコマンド	説明	使用法
LIST	ある条件を満たすトラップ番号を一覧表示するコマンド。	ndssnmpconfig LIST <*trapSpec*> *trapSpec* には、トラップ番号のほか、以下に述べるキーワードで条件を指定できます。 ALL、ENABLED、DISABLED、FAILED、または論理式 例 : 有効なトラップをすべて、名称を添えて表示 : ndssnmpconfig LIST ENABLED 無効なトラップをすべて、名称を添えて表示 : ndssnmpconfig LIST DISABLED 117 種類すべてのトラップについて、名称を添えて表示 : ndssnmpconfig LIST ALL 特定の番号 (次の例では 12 番、224 番、300 番) のトラップについて、名称を添えて表示 : ndssnmpconfig LIST ID = 12,224,300 特定の番号 (次の例では 12 番、224 番、300 番) 以外のトラップをすべて、名称を添えて表示 : ndssnmpconfig LIST ID != 12,224,300 有効なエラートラップをすべて、名称を添えて表示 : ndssnmpconfig LIST FAILED

トラップコマンド	説明	使用法
READ_CFG	環境設定ファイル ndstrap.cfg を参照して、ディレクトリ構成を再設定するコマンド。 環境設定ファイルの設定に加えられた変更はすべて有効になります。いくつものコマンドを ndstrap.cfg に記述しておき、ひとまとめにして実行する、という使い方を主として想定したユーティリティです。 ndstrap.cfg ファイルは /etc/ndssnmp/ に配置されています。 ndstrap.cfg ファイルでは、トラップの環境設定に使用されるオペレーショナルパラメータを設定し、SNMP トラップの操作を設定する方法を指定します。トラップの環境設定ユーティリティ ndssnmpcfg が READ_CFG コマンドで実行される場合には、必ずこのファイルが読み込まれます。	ndssnmpconfig "READ_CFG"
FAILURE	エラートラップをすべて表示するコマンド。 エラートラップとは、イベントの失敗時に生成されるトラップのことです。 **注 :** エラートラップをいったん無効にし、「enable trapid」コマンドで再び有効にすると、エラートラップではなく、正常に処理されたことを表すトラップとして扱われるようになります。	ndssnmpconfig "FAILURE *trapSpec*" *trapSpec* には、トラップ番号をコンマまたはスペースで区切って指定するほか、キーワード ALL や論理式を指定できます。 例 : 複数のトラップをエラートラップと設定 : ndssnmpconfig "FAILURE 10,11,100" 指定した番号以外のすべてのトラップをエラートラップと設定 : ndssnmpconfig "FAILURE ID != 24,30" すべてのトラップをエラートラップと設定 : ndssnmpconfig "FAILURE ALL"

[Statistics (統計情報)]

- 578 ページの 「ndsDbCache」
- 579 ページの 「ndsDbConfig」

- 579 ページの 「ndsProtoIfOps」
- 581 ページの 「ndsServerInt」

ndsDbCache

ディレクトリ以下にある管理対象オブジェクト	説明
ndsDbSrvApplIndex	OpenText eDirectory サーバアプリケーションを固有に識別するインデックス。
ndsDbDibSize	OpenText eDirectory データベースの現在のサイズ (KB 単位)。
ndsDbBlockSize	OpenText eDirectory データベースのブロックサイズ (KB 単位)。
ndsDbEntryCacheMaxSize	エントリキャッシュの最大容量 (KB 単位)。
ndsDbBlockCacheMaxSize	ブロックキャッシュの最大容量 (KB 単位)。
ndsDbEntryCacheCurrentSize	現在のエントリキャッシュ容量。
ndsDbBlockCacheCurrentSize	現在のブロックキャッシュ容量。
ndsDbEntryCacheCount	キャッシュ内のエントリ数。
ndsDbBlockCacheCount	キャッシュ内のブロック数。
ndsDbEntryCacheOldVerCount	キャッシュ内に残っている旧バージョンのエントリ数。
ndsDbBlockCacheOldVerCount	キャッシュ内に残っている旧バージョンのブロック数。
ndsDbEntryCacheOldVerSize	旧バージョンのエントリキャッシュ容量。
ndsDbBlockCacheOldVerSize	旧バージョンのブロックキャッシュ容量。
ndsDbEntryCacheHits	キャッシュ内のエントリがヒットした回数。
ndsDbBlockCacheHits	キャッシュ内のブロックがヒットした回数。
ndsDbEntryCacheHitLooks	キャッシュ内のエントリがヒットするかどうか試みられた回数。
ndsDbBlockCacheHitLooks	キャッシュ内のブロックがヒットするかどうか試みられた回数。
ndsDbEntryCacheFaults	キャッシュ内のエントリがヒットしなかった回数。
ndsDbBlockCacheFaults	キャッシュ内のブロックがヒットしなかった回数。
ndsDbEntryCacheFaultLooks	キャッシュ内のエントリがヒットしないかどうか試みられた回数。
ndsDbBlockCacheFaultLooks	キャッシュ内のブロックがヒットしないかどうか試みられた回数。

ndsDbConfig

ディレクトリ以下にある管理対象オブジェクト	説明
ndsDbCfgSrvApplIndex	OpenText eDirectory サーバアプリケーションを固有に識別するインデックス。
ndsDbCfgDynamicCacheAdjust	動的キャッシュ調整が有効かどうか。0 = off 1 = on
ndsDbCfgDynamicCacheAdjustPercent	動的キャッシュ調整に、空きメモリの何 % を割り当てるか。
ndsDbCfgDynamicCacheAdjustMin	動的キャッシュ調整に使う最小容量。キャッシュ容量制限を KB 単位で表したもの。
ndsDbCfgDynamicCacheAdjustMinToLeave	動的キャッシュ調整に使う最小容量のうち、利用可能なメモリから差し引く容量 (KB 単位)。
ndsDbCfgHardLimitCacheAdjust	キャッシュ調整に割り当てるメモリ容量にハード制限を設定するかどうか。0 = off 1 = on
ndsDbCfgHardLimitCacheAdjustMax	キャッシュの最大容量 (KB 単位)。これはハードメモリ制限を表します。
ndsDbCfgBlockCachePercent	ブロックキャッシュに割り当てる比率。
ndsDbCfgCacheAdjustInterval	キャッシュ調整を行う時間間隔 (秒単位)。
ndsDbCfgCacheCleanupInterval	キャッシュのクリーンアップを行う時間間隔 (秒単位)。
ndsDbCfgPermanentSettings	常時接続の設定が有効かどうか。0 = off 1 = on

ndsProtoIfOps

ディレクトリ以下にある管理対象オブジェクト	説明
ndsProtoIfSrvApplIndex	OpenText eDirectory サーバアプリケーションを固有に識別するインデックス。
ndsProtoIfIndex	OpenText eDirectory サーバのプロトコルインタフェースに対応するエントリを固有に識別するインデックス。
ndsProtoIfDescription	DS プロトコルインタフェースに使うポート番号
ndsProtoIfUnauthBinds	認証を省略した匿名バインド要求を受け取った回数。
ndsProtoIfSimpleAuthBinds	バインド要求のうち、簡易認証手続きにより認証に成功したものの回数。簡易認証手続きとは、パスワードを暗号化して、または平文のまま送ることにより行うものです。

ディレクトリ以下にある管理対象オブジェクト	説明
ndsProtoIfStrongAuthBinds	バインド要求のうち、強度の高い認証手続きである SASL および X.500 の認証に成功したものの回数。外部認証手続きによるものも数に含みます。
ndsProtoIfBindSecurityErrors	バインド要求のうち、認証手続きが適切でない、あるいは資格情報が無効であるために拒否したものの回数。
ndsProtoIfInOps	DUA(ディレクトリユーザエージェント) または他の OpenText eDirectory サーバから受け取った要求の回数。
ndsProtoIfReadOps	受け取った読み出し要求の数。
ndsProtoIfCompareOps	受け取った比較要求の数。
ndsProtoIfAddEntryOps	受け取ったエントリ追加要求の数。
ndsProtoIfRemoveEntryOps	受け取ったエントリ削除要求の数。
ndsProtoIfModifyEntryOps	受け取ったエントリ変更要求の数。
ndsProtoIfModifyRDNOps	受け取った RDN(相対識別名) 変更要求の数。
ndsProtoIfListOps	受け取ったリスト要求の数。
ndsProtoIfSearchOps	受け取った検索要求 (ベースオブジェクト検索、1 レベル検索、サブツリー全体の検索) の数。
ndsProtoIfOneLevelSearchOps	受け取った 1 レベル検索要求の数。
ndsProtoIfWholeSubtreeSearchOps	受け取ったサブツリー全体の検索要求の数。
ndsProtoIfExtendedOps	拡張処理の回数。
ndsProtoIfReferrals	処理要求に応じて返した参照の個数。
ndsProtoIfChainings	この OpenText eDirectory サーバから他の OpenText eDirectory サーバに転送された処理の数。
ndsProtoIfSecurityErrors	受け取った要求のうち、セキュリティ保護方針に合致しなかったものの数。
ndsProtoIfErrors	受け取った要求のうち、エラーのため応じなかったものの数。ただしセキュリティ保護エラー、参照エラーを除きます。一部でも処理ができたものは数に含めません。たとえば、名前づけや更新、属性、サービスに関連するエラーがあります。
ndsProtoIfReplicationUpdatesIn	OpenText eDirectory サーバから取得した、または受け取ったレプリケーション更新の数。
ndsProtoIfReplicationUpdatesOut	OpenText eDirectory サーバに送った、または検知されたレプリケーション更新の数。
ndsProtoIfInBytes	インタフェース上の受信トラフィック (バイト単位)。DUA からの要求と、他の OpenText eDirectory サーバからの応答がこれに含まれます。

ディレクトリ以下にある管理対象オブジェクト	説明
ndsProtoIfOutBytes	インタフェース上の送信トラフィック (バイト単位)。DUA や OpenText eDirectory サーバへの応答と、他の OpenText eDirectory サーバへの要求がこれに含まれます。

ndsServerInt

ディレクトリ以下にある管理対象オブジェクト	説明
ndsSrvIntSrvApplIndex	OpenText eDirectory サーバアプリケーションを固有に識別するインデックス。
ndsSrvIntProtoIfIndex	OpenText eDirectory サーバのプロトコルインタフェースに対応するエントリを固有に識別するインデックス。
ndsSrvIntIndex	このオブジェクトは ndsSrvIntSrvApplIndex および ndsSrvIntProtoIfIndex と組み合わせて使います。applIndex で示される OpenText eDirectory サーバと、特別なプロトコルを使うピア OpenText eDirectory サーバとの間でやり取りされる有用な情報を格納した、仮想的な行を識別するための固有のキーとなります。
ndsSrvIntURL	ピア OpenText eDirectory サーバの URL。
ndsSrvIntTimeOfCreation	この行が作成された時刻。1970 年 1 月 1 日午前 0 時 (万国標準時) からの経過秒数で表します。
ndsSrvIntTimeOfLastAttempt	ピア OpenText eDirectory サーバとの接続を試みた直近の時刻。1970 年 1 月 1 日午前 0 時 (万国標準時) からの経過秒数で表します。
ndsSrvIntTimeOfLastSuccess	ピア OpenText eDirectory サーバと正常に接続できた直近の時刻。1970 年 1 月 1 日午前 0 時 (万国標準時) からの経過秒数で表します。
ndsSrvIntFailuresSinceLastSuccess	ピア OpenText eDirectory サーバと正常に接続した直近の時刻以降に発生したエラーの数。まだ一度も正常に接続できていない場合は、このエントリが作成されて以来のエラー数。
ndsSrvIntFailures	このエントリが作成されて以来、ピア OpenText eDirectory サーバとの接続に失敗した回数。
ndsSrvIntSuccesses	このエントリが作成されて以来、ピア OpenText eDirectory サーバとの接続に成功した回数。

トラブルシューティング

問題を迅速に解決できるよう、実際に発生したエラー状況、その解決に役立つ情報が、ログファイルに記録されています。詳細については、899 ページの 「SNMP のトラブルシューティング」を参照してください。

表 18-1 に、Linux プラットフォーム用のサーバログファイルのデフォルトの場所を示します。ndsd.log ファイルの場所を特定するには、OpenText eDirectory インスタンスに対して ndsconfig get n4u.server.log-file コマンドを実行します。

表 18-1 *[ログファイルの場所]*

Platform	サブエージェント	サーバ	マスタ
Windows	*install_directory*\nds\snmp\dssnmpsa.log	*install_directory*\nds\snmp\dssnmpsrv.log	NA
Linux	/var/opt/novell/eDirectory/log/ndssnmpsa.log	/var/opt/novell/eDirectory/log/ndsd.log	/var/log/messages

19 OpenText eDirectory のメンテナンス

OpenText eDirectory のパフォーマンスを最適にするには、定期的なヘルスチェック手順を実行し、必要に応じてハードウェアのアップグレードや交換を行ってディレクトリをメンテナンスする必要があります。

この章では、次のメンテナンスに関するトピックについて説明します。

パフォーマンス

- 583 ページの 「詳細参照コスト」

ヘルスチェック

- 593 ページの 「OpenText eDirectory の正常動作の維持」
- 596 ページの 「監視のためのリソース」

ハードウェアの交換

- 596 ページの 「ハードウェアのアップグレードやサーバの交換」

OpenText eDirectory の回復

- 604 ページの 「ハードウェア障害後の OpenText eDirectory の復元」

詳細参照コスト

サーバアプリケーションの多くは、組み込みクライアント (Dclient) 経由で他のサーバと通信します。これは、1 つのサーバに、アプリケーションが動作するのに必要なすべての OpenText eDirectory データが保存されているわけではないためです。1 つの例が、要求をチェーンするように設定された NLDAP です。

サーバアプリケーションがローカルサーバに保存されていないデータを要求した場合、サーバは要求されたデータが保存されている別のサーバを特定してから、クライアントに関するデータを取得します。このプロセスは「ツリーウォーキング」と呼ばれます。通常、サーバがツリーウォーキング経由で要求を満たすまでに時間がかかります。OpenText eDirectory ツリー設計に関するベストプラクティスのガイドラインを使用すればツリーウォーキングの必要性が最小限に抑えられますが、それでも必要な場合があります。

図 19-1 詳細参照コスト

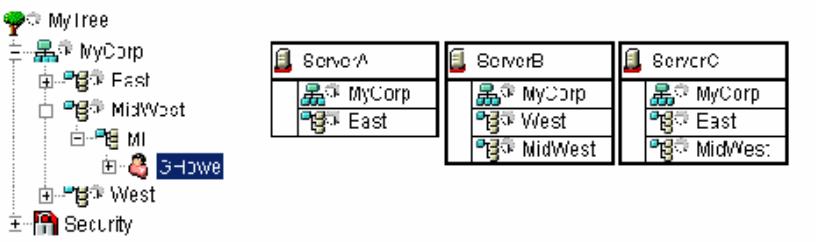


図 19-1 は、O=MyCorp から始まる、サーバ A に対する cn=GHowe の LDAP サブツリー検索を示しています。ただし、cn=GHowe オブジェクトは、サーバ A 上に存在しない ou=MidWest パーティションに配置されています。

クライアントの要求を満たすために必要なデータが保存されたサーバを特定するには、サーバ A がサーバ B またはサーバ C からデータを取得する必要があります。これを行うには、サーバ A がサーバ B とサーバ C のどちらかに要求を送信する必要があります。サーバ A はサーバ B を選択しました。サーバ選択プロセスは予測不能であることに注意してください。サーバ B はネットワーク上で使用可能で、要求を受け入れますが、要求を迅速に処理できないため、サーバ C も必要なデータを提供できたとしても、サーバ A はサーバ B を待機することになります。サーバ B が要求を満たすか、ネットワーク上で使用できなくなるまで、サーバ A からの要求が待たされることになります。

以下のセクションで、OpenText eDirectory サーバのパフォーマンスを改善する方法について説明します。

- 584 ページの 「サーバ間接続の向上」
- 586 ページの 「参照コストのメリット」
- 587 ページの 「ARC の展開」
- 588 ページの 「詳細参照コストの有効化」
- 589 ページの 「詳細参照コストの調整」
- 590 ページの 「詳細参照コストの監視」

サーバ間接続の向上

詳細参照コスト (ARC) は高度なコストアルゴリズムです。ARC の主な目的は、サーバの機能停止を回避することです。ARC のメリットを以下に示します。

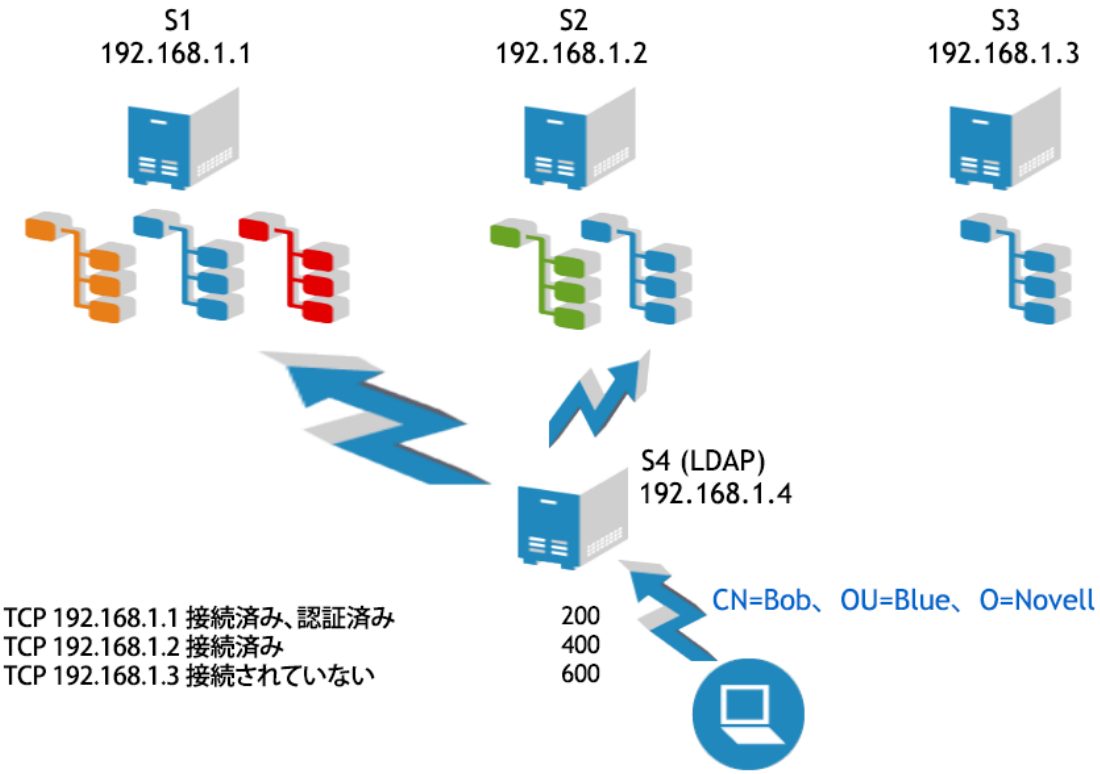
- サーバのパフォーマンスと耐障害性の向上
- サーバ間通信の改善
- 負荷分散
- リモートサーバヘルスモニタリング
- 通信問題の分離と特定の簡素化

ARC を使用すべきなのは誰か？

オブジェクトまたはサービスのローカルコピーが保存されていないサーバは、他のサーバと頻繁に通信するため、ツリーで ARC からもたらされた情報を探索する必要があります。ARC は、LDAP 環境、特に、優先チェーン処理中に非常に効率的に動作します。

たとえば、図 19-2 に示すように、サーバは要求を送り続ける別のサーバから悩まされる場合があります。

図 19-2　ワンストップサーバの影響

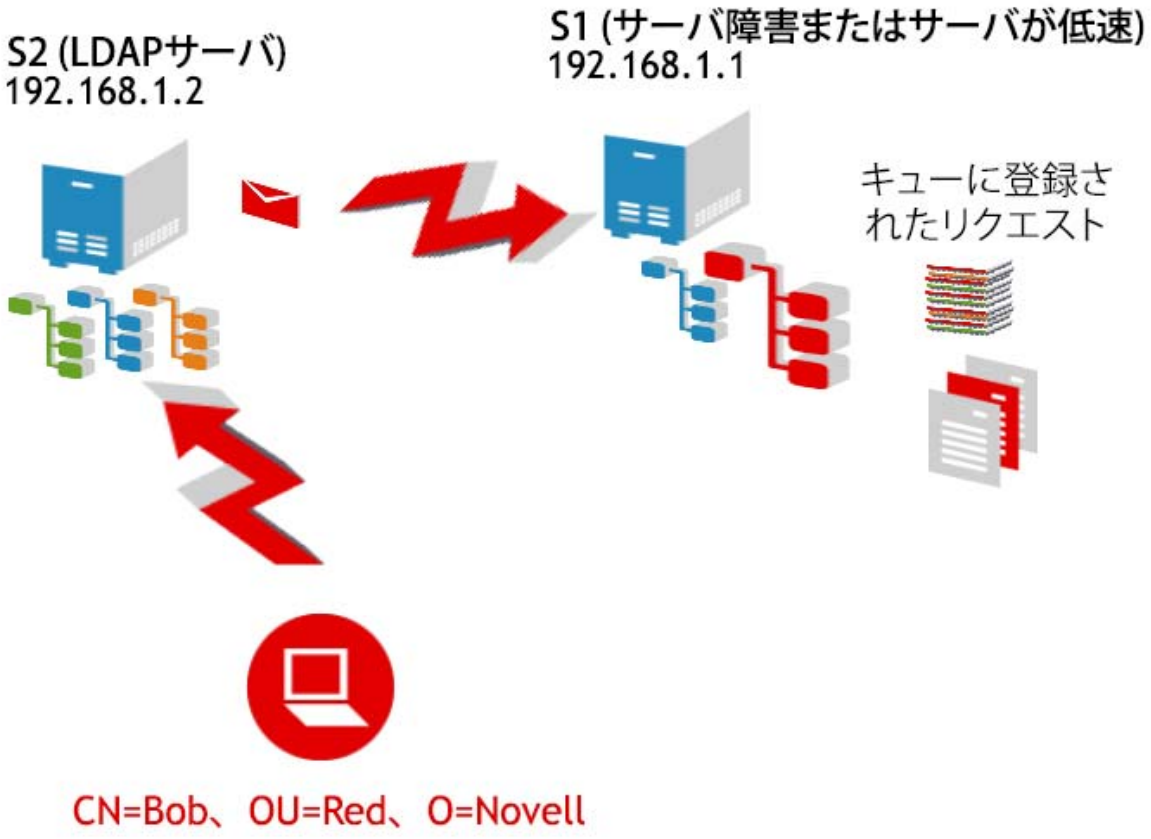


必要なオブジェクトのレプリカが保存された使用可能なサーバが他にあったとしても、このサーバが優先されると思われます。これは、サービスまたはレプリカを要求しているサーバがすでにこのサーバに接続されており、処理可能なすべての要求がこのサーバに集中する傾向があるためです。図 19-2 は、S4 からのすべての要求が S1 に送信される様子を示しています。S4 はすでに S1 に接続されており、S1 に対して認証されているため、S2 と S3 がその要求に応えることができたとしても、S4 は青色のパーティションに対するすべての要求を S1 に送信し続けます。ARC を使用すれば、より迅速に対応可能なサーバに負荷を分散することによって、このような状況を回避できます。このサーバを必要とするリモートサーバ (S4) 上で ARC を有効にする必要があります。または、すべてのサーバ上で ARC を有効にすることができます。

図 19-3 は、「カスケーディングサーバ」の影響を示すもう 1 つのシナリオを示しています。ここで、サーバ S1 は頻繁に応答を返しませんが、ダウンしているわけではありません。S1 がダウンすると、要求がタイムアウトして、通信が停止します。サーバがまだトランスポートレベルでアップしているが、データベースが低速またはビジーの場合は、他のサーバからの新しい要求を受け付けてキューに入れ続けます。そのため、最終的に、追加の

サーバ (S2) がスレッドを使い果たすことになります。未処理の要求がそれぞれリモートサーバでスレッドを取得し、スレッドが使い果たされた段階で、サーバが応答を返さなくなります。ARC は最も高速なサーバに要求を分散することによってこの問題を解決します。これは、低速なまたは不具合のあるサーバは要求の処理に高いコストがかかるためです。

図 19-3 カスケーディングサーバの影響



加えて、ARC は、耐障害性の向上にも役立ちます。サーバの通信問題を簡単に特定する機能を備えています。

参照コストのメリット

- これは、リモートサーバへのほとんどの名前解決要求を計時 / ルーティングします。
- アドレスごとの名前解決要求時間をミリ秒単位で平均化します。これにより、ARC は、より細かいレベルでより積極的に参照のコストを調整できます。タイミングが秒単位ではなくミリ秒単位で追跡されるため、低速なサーバをすばやく検出することもできます。

- 未処理要求を追跡することによって、要求に時間がかかりすぎているかどうかをすばやく判断できます。サーバで時間がかかっていることを確認するために、要求が完了するまで待つ必要はありません。
- アドレス単位で応答時間を追跡します。1 台のサーバが同じアドレスに何回も接続するのが普通です。接続単位ではなくアドレス単位で追跡することによって、1 つの接続で他の接続から収集された統計情報を参考にできます。

注 : LDAP 要求を理解するために、ARC はプライベート接続の応答性も考慮します。

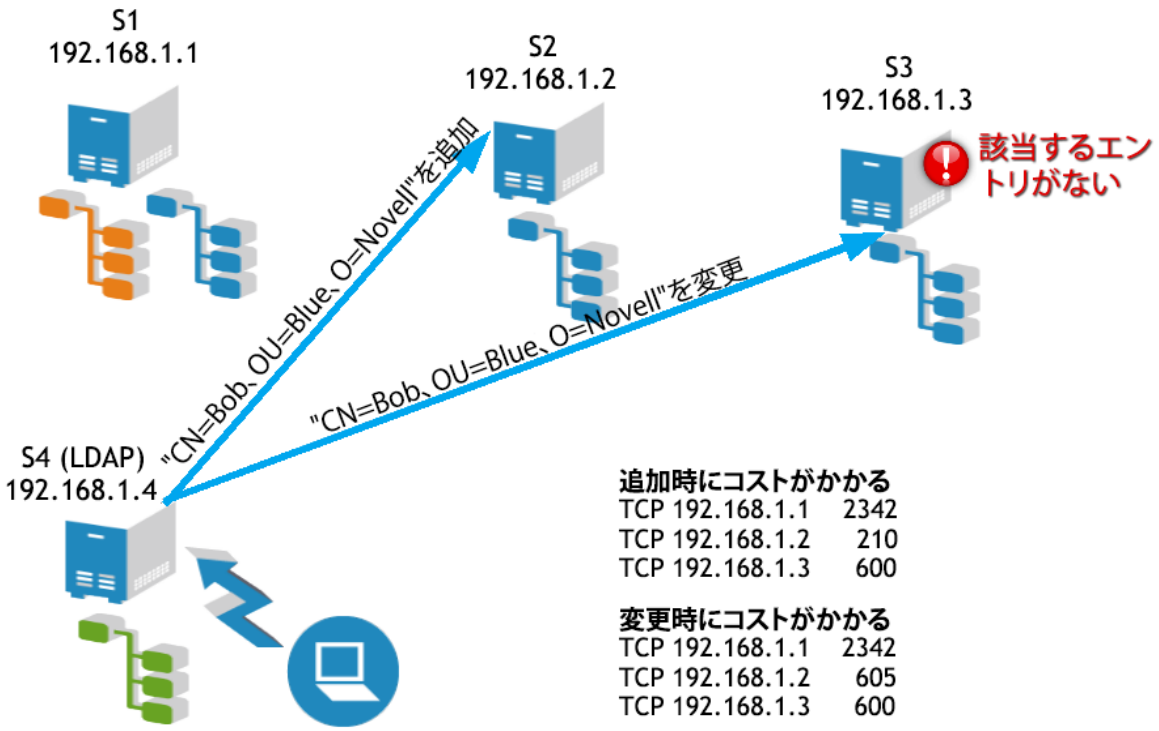
ARC の展開

ARC は、通常、サーバ単位で展開されます。ARC 対応のこれらのサーバは新しいコスト情報を入手することができます。環境内のすべてのサーバで ARC を有効にする必要があります。

展開に関する考慮事項

すべてのサーバ上で ARC を有効にすることが役に立つとは限りません。図 19-4 は、LDAP サーバの有効性に影響を与える状況を示しています。図では、S4 に青色のパーティションではなく緑色のパーティションのコピーが保存されています。青色のパーティションに関する情報が必要なチェーン LDAP 要求は、S1、S2、または S3 サーバに送信され、処理される必要があります。これはほとんどのケースで動作します。ARC は、まさにこのような状況に合わせて設計されています。

図 19-4 *ARC 展開に関する考慮事項*



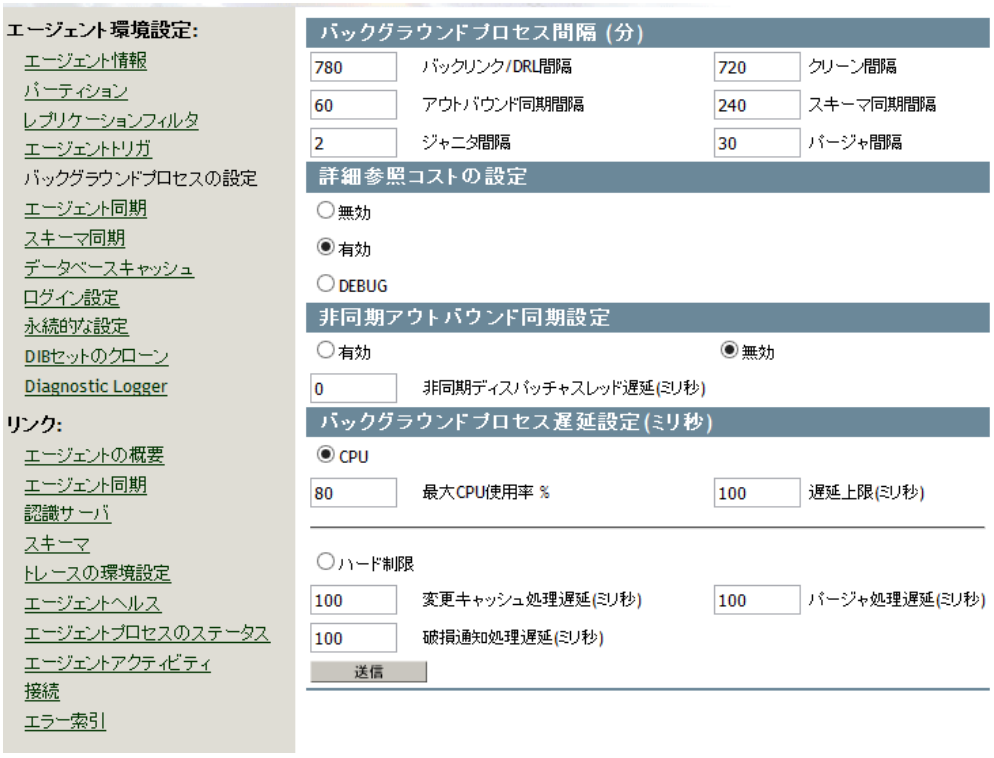
ただし、特定の LDAP 操作の実行には困難が伴います。Bob.Blue.Novell などのユーザを追加することはできますが、Bob を変更するために急いで戻ろうとすると操作が失敗します。図は S2 上に Bob が追加されたことを示していますが、S3 上で Bob を変更しようとしても、S3 がまだ S2 と同期していないために操作が失敗して、S3 は Bob を受け取ることができません。ARC はユーザを別のサーバに誘導する機能を備えています。これは、ARC の方がオリジナルのコスト計算手法より動的なためです。

この設定は、サーバコストがあまり変化せず、正常に同期しているシナリオでうまく動作します。S4 上で ARC を無効にするとこの問題が解決されます。

詳細参照コストの有効化

ARC は、デフォルトで、OpenText eDirectory に対して有効になっています。NDS OpenText iMonitor を使用して ARC を設定するには、[**エージェント環境設定**] > [**バックグラウンドプロセスの設定**] の順にクリックします。加えて、[**有効**]、[**無効**]、および [**デバッグ**] オプションを使用できます。

図 19-5 *NDS OpenText iMonitor エージェントの環境設定画面*



NDSTrace

すべての UNIX プラットフォームで、NDSTrace ツールを使用して ARC を有効にします。

表 19-1 UNIX プラットフォーム上での ARC の有効化

set NDSTRACE =!ARC	デバッグ用の gv_ResolveTimesTable を表示します。
set NDSTRACE =!ARC0	詳細参照コストを無効にします。
set NDSTRACE =!ARC1	詳細参照コストを有効にします。
set NDSTRACE =!ARC2	詳細参照コストをデバッグモードで有効にして、コストが決定されるたびに、名前解決 DSTrace フラグに対する参照のコストを表示します。

詳細参照コストの調整

デフォルトで、ARC は調整する必要がありません。ただし、ARC の機能を変更したり、特定の機能を有効 / 無効にしたりするための調整可能なパラメータが ARC 内に存在します。ARC にとって重要なコンポーネントが 3 つあります。

詳細コスト

特定のアドレスのコストを見積もるように要求されたら、ARC は、接続に関する既知の情報を使用して特定の参照のコストを計算します。ARC がオンになっている場合は、参照のコスト計算に常に詳細コストが使用されます。

バックグラウンド監視

バックグラウンドスレッドは、定期的にタイマ情報をチェックして誤差がないことを確認します。サーバが低速の場合は、そのコストが上昇して、通信が停止する可能性が高くなります。バックグラウンドスレッドは、定期的 (デフォルトでは 1 分ごと) に、テーブル内のサーバが更新されていないかどうかをチェックします。過去 3 分以内にサーバが更新されていなかった場合は、サーバがそれ自体に名前解決要求を発行してサーバのヘルスをチェックします。これにより、最新のサーバのコストが計算され、サーバがビジー状態を脱したのかどうか、つまり、正常に戻ったのかどうかも検出されるため、クライアントはサーバのヘルスをチェックする必要がありません。バックグラウンドスレッド用の変更可能な永続的な環境設定パラメータが 2 つあります。

- **ARC_MAX_WAIT:** サーバのヘルスをチェックするためのサーバへの要求を発行するまでのタイマ値 (デフォルトは 180 秒)
- **ARC_BG_INTERVAL:** バックグラウンドスレッドの実行間隔 (デフォルトは 60 秒。0 は無効になっていることを意味し、スレッドは実行されません)。

詳細については、セクション 8.4.24 の永続的な環境設定パラメータの設定に関する説明を参照してください。

リモートヘルス情報

ARC を使用したサーバは、定期的に、リモートサーバにヘルス情報を要求します。これらは、ネットワーク上に追加で発行される要求ではなく、サーバが頻繁に発行する標準の名前解決要求で返される追加のヘルス情報です。この情報は、高負荷状態にあるサーバへの

対応を迅速化するコストアルゴリズムで使用されます。名前解決要求がリモートサーバに発行されると、前回の更新から 15 秒以上経過していた場合は、ヘルス情報がリモートサーバに要求され、名前解決要求の応答に追加されます。

リモートヘルスモニタリングには調整可能なパラメータが 1 つあります。

- **ARC_DS_INFO_INTERVAL:** これは、ARC 内のロック (ヘルス) 情報の要求間隔です (デフォルトは 15 秒)。

詳細参照コストの監視

ResolveTimes テーブルを出力して、操作における詳細参照コストを確認することができます。

ResolveTimes テーブルは次のコマンドを使用して出力します。

- set DSTRACE = +DBG
- set DSTRACE = !ARC

このコマンドは、Resolve Times テーブルとサーバごとの現在保存されている情報を出力します。トランスポートアドレス、アドレスが最後に使用された以降の経過時間 (ミリ秒)、参照決定で使用された最後のコスト、およびそのアドレスに対する未処理要求の数が表示されます。

未処理要求の数が多くても、必ずしも問題ではありません。単に、そのサーバの使用頻度が高いこと示しているにすぎない場合もあります。

トラブルシューティングでの ARC の使用

ARC の最も有用な機能の 1 つがサーバに伴う通信問題を迅速に特定する機能です。

ResolveTimes テーブルのプリントアウトの例を以下に示します。

ARC は現在有効になっています。

***表 19-2** 解決時間コスト*

Slot	トランスポートアドレス	Cost	最後の使用	オン	要求数	待機数	ロック時間
1	tcp:151.155.134.27:524	214	14	14	0	0	0
2	tcp:151.155.134.11:524	0	0	0	0	0	0
3	udp:151.155.134.11:524	0	0	0	0	0	0
4	cp:151.155.134.13:524	554759	280	0	0	27	582
5	tcp:151.155.134.59:524	0	179	179	0	0	0
6	udp:151.155.134.59:524	0	119	119	0	0	0
7	tcp:151.155.134.28:524	1543	119	119	0	0	0
8	tcp:151.155.134.15:524	124	14	14	0	0	0

このプリントアウトは、このサーバの視点から 151.155.134.13 で問題が発生していることを示しています。問題がトランスポートではなくサーバにある可能性が高いこともわかります。サーバでは、27 件の要求がデータベースへのアクセス待ちであり、それらがデータベースロックを取得するのに時間がかかっています。このサーバでは、2 件の要求がリモートサーバからの応答を受け取っていません。

151.155.134.11 と 151.155.134.59 は、非常に高速なサーバか、あまりビジーではないかのどちらかまたはその両方であることもわかります。151.155.134.59 と 151.155.134.11 の両方で TCP 経由の通信問題が 1 回発生したが、UDP 接続が使用できるため現在は正常に戻っていることもわかります。サーバへの UDP 接続は、TCP 経由でサーバに通知できない場合にのみ、試行されます。

それぞれの数字の意味を以下に示します。

トランスポートアドレス : リモートサーバのアドレス。

Cost: リモートサーバの現在のコスト。

最後の使用 : サーバとの最後の通信以降の経過時間 (秒)。

オン : リモートサーバからの最後のヘルス情報以降の経過時間 (秒)。

要求数 : リモートサーバへの未処理要求数。

待機数 : データベースのロックを待っているリモートサーバへの要求数。

ロック時間 : プロセスがリモートサーバ上でデータベースのロックを維持していた時間。

次のプリントアウトは通信の問題を迅速に特定するためのもう 1 つの例を示しています。サーバは TCP 経由で 151.155.134.13 に通信できないことがわかります。

ARC は現在有効になっています。

表 19-3 *解決時間コスト*

Slot	トランスポートアドレス	Cost	最後の使用	オン	要求数	待機数	ロック時間
1	tcp:151.155.134.27:524	394	92	14	0	0	0
2	tcp:151.155.134.11:524	0	0	0	0	0	0
3	udp:151.155.134.11:524	0	0	0	0	0	0
4	tcp:151.155.134.13:524	5000000	180	180 が不正なアドレスキャッシュ内にあります			

これらの表を参照する際は、以下の点にご留意ください。

- 未処理の要求があっても問題とは限りません。単にサーバが多くの要求に対応している場合もあるからです。コストが高いサーバ上の未処理の要求は問題です。
- サーバのヘルスを示す第一の指標は現在のコストであり、問題が発生しているサーバを特定するのに役立ちます。

注 : すべての要求の往復時間と未処理期間が測定されます。これは、トランスポート時間もコストの一部であることを意味します。この表でサーバに問題があるように見えても、他のサーバからは問題がないように見える場合、それはトランスポートの問題を示唆している可能性があります。

バックグラウンドスレッドトレース

ARCBackgroundResolveTimerThread の動作を示すトレースを以下に示します。

```
ARCBackGroundResolveTimerThread started Interval = 60 MaxWait = 180000

Updating timer info for tcp:151.155.134.11:524

Updating timer info for udp:151.155.134.11:524

Updating timer info for tcp:151.155.134.13:524
ARCBackGroundResolveTimerThread error -635 in DCConnectToAddress for
tcp:151.155.134.59:524

ARCBackGroundResolveTimerThread completed in 0 seconds

8-total timers 4-stale timers 3-timers updated
```

上記のメッセージから、次のことが分かります。

- TCP:151.155.134.11 は 3 分を超える期間使用されていません。
- UDP:151.155.134.11 は 3 分を超える期間使用されていません。
- TCP: 151.155.134.13 は 3 分を超える期間使用されていません。

上記のすべてのサーバでタイマー情報が更新され、次のような結果になりました。

- TCP: 151.155.134.59 は依然としてこのサーバから到達できません。

新しいコストは非常に動的でとても頻繁に変化します。この動きを監視するには、詳細参照コストパラメータをデバッグモードに設定します。

注 : 監視が終わったら、set NDSTRACE =!ARC1 コマンドを実行して、ARC を非デバッグモードにリセットしてください。印刷が必要なければ、オーバーヘッドの印刷コストはないに越したことはありません。

DSTrace または NDSTrace では、詳細参照コストと +RSLV がオンになっている場合、個々の参照コストが表示されます。残りのタグは set NDSTrace = nodebug コマンドを使用してオフにします。

```
Sorted results from DCAdjustCostAndSort follow:

137.65.10.3 cost of 217

137.65.10.9 cost of 222

137.65.10.10 cost of 400
```

リモートサーバが低速または過負荷になると、数値がすぐに変化します。ExRef サーバのコストは毎秒動的に調整されるため、一定期間のコストを監視するには、ログファイルをトレースする必要があります。

OpenText eDirectory の正常動作の維持

ディレクトリサービスのヘルスはあらゆる組織にとって極めて重要です。OpenText iMonitor を使用して定期的にヘルスチェックすることで、ディレクトリの円滑な機能が維持され、アップグレードやトラブルシューティングがずっと容易になります。

ヘルスチェックを実行する時期

一般に、ネットワークを頻繁に変更しない場合 (2 ～ 3ヶ月程度の頻度でしかサーバとパーティションを追加せず、単純な変更だけを頻繁に行う場合) は、ヘルスチェックは月に 1 度実行します。

ネットワークの変更が頻繁に発生する場合 (パーティションやサーバが毎週追加される、あるいは組織の再編成を行っている場合)、ヘルスチェックは週に 1 度実行します。

環境の変更に応じて、ヘルスチェックの頻度を調整します。ヘルスチェックの実行頻度に影響する要素を次に示します。

- パーティションとレプリカの数
- サーバを保持しているレプリカの安定度
- OpenText eDirectory パーティション内の情報量
- オブジェクトのサイズと複雑さ
- 以前の DSRepairs 内のエラー数

ヘルスチェックを実行すると、所有する権利に基づき、OpenText iMonitor がすべてのサーバから情報を集めます。なお、ヘルスチェックレポートを実行すると、ネットワークトラフィックが発生し、ディスク容量を消費する場合があることに注意してください。

ヘルスチェックの概要

完全なヘルスチェックでは、次の情報がチェックされます。

- OpenText eDirectory のバージョン

  同じサーバ上で異なるバージョンの NDS や OpenText eDirectory を実行していると、同期の問題が発生する可能性があります。NDS または OpenText eDirectory のバージョンが古い場合は、最新のソフトウェアパッチをソフトウェアライセンスおよびダウンロードポータルからダウンロードします。
- 時刻同期

  すべての OpenText eDirectory サーバは、正確な時刻を維持する必要があります。タイムスタンプが各オブジェクトおよびプロパティに割り当てられ、これによりオブジェクトおよびプロパティの更新が正しい順序で行われます。OpenText eDirectory では、タイムスタンプを使用して、同期が必要なレプリカを判別します。
- 同期の許容範囲

  インバウンドやアウトバウンドのデータ変更により同期を行ってから経過した期間で、どれだけのデータが未処理となっているかなどをチェックします。

- バックグラウンド処理

  プロセスはさまざまなタスクを実行しますが、その中には変更の複製およびシステム情報の保守があります。

  - 外部参照
  - 破損通知
  - OpenText eDirectory スキーマ

これらのチェックを実行するための詳細な手順については、次のセクション (594 ページの「OpenText iMonitor を使用した OpenText eDirectory のヘルスチェック」) を参照してください。

OpenText iMonitor を使用した OpenText eDirectory のヘルスチェック

環境設定によっては、OpenText eDirectory サーバのヘルスチェックを OpenText iMonitor の次の 2 つの方法のいずれかを使用して実行できます。

- ナビゲータフレームを使用する
- アシスタントフレームを使用する

ナビゲータフレームを使用する

1 OpenText iMonitor にアクセスします。

「245 ページの 「OpenText iMonitor へのアクセス」」を参照してください。

2 ナビゲータフレームで、［レポート］アイコンをクリックします。

3 アシスタントフレームで、［**レポート設定**］リンクをクリックします。

データフレームに、実行可能レポートリストが表示されます。

4 必要なサーバ情報の［レポートの設定］アイコンをクリックします。

データフレームに、サーバ情報レポートが表示されます。このレポートを使用して、レポートに必要なオプションを選択します。

5 ［**ヘルスのサブレポート**］チェックボックスをオンにします。

6 指定した間隔でレポートを実行するには、データフレームの［レポートのスケジュール］セクションで、必要なオプションを選択します。

重要：スケジュールされたレポートを実行する場合は、Public ユーザとして実行され、認証済みユーザとして実行する場合よりも得られる情報が少なくなる可能性があります。

---

7 ［**レポートの実行**］をクリックして、レポートを処理します。

アシスタントフレームを使用する

1 OpenText iMonitor にアクセスします。

「245 ページの 「OpenText iMonitor へのアクセス」」を参照してください。

2 アシスタントフレームで、［エージェントヘルス］をクリックします。

OpenText iMonitor が情報を取得するサーバ (接続先のサーバとは限りません) のヘルスチェック情報が、データフレームに表示されます。

レポート情報の検討

レポートが生成されたら、データフレームにレポート結果が表示されます。ツリー内に正常でないサーバがある場合、レポートは次の 3 つのカテゴリに分けられます (グループ化はヘルス状態が悪いサーバから始まります)。

- 警告のあるサーバ
- 疑わしいサーバ
- 正常なサーバ

警告のあるサーバや疑わしいサーバがない場合は、これらのカテゴリは表示されません。

正常に動作していないサーバがある場合は、そのサーバの横の［エージェントヘルスサブレポート］リンク をクリックできます。オンラインの文脈依存型ヘルプを使用して、問題を解決します。このヘルプは、個々のオプションの意味、それが重要である理由、問題の解決方法、範囲の調整方法、およびヘルスチェックに追加するオプションがあるかどうかを確認するのに役立ちます。

重要 : 警告のあるサーバがある場合、その問題を解決することを強くお勧めします。疑わしいサーバについても、評価することをお勧めします。

その他の情報

OpenText eDirectory の正常な動作を維持するために使用するツールとテクニックについては、『OpenText eDirectory Tools & Diagnostics Course 3007』を参照してください。このコースでは、次の方法について学習します。

- OpenText eDirectory のヘルスチェックの実行方法
- OpenText eDirectory の正しい操作方法 .
- OpenText eDirectory の問題の適切な診断、トラブルシューティング、および解決の方法
- OpenText eDirectory のトラブルシューティングツールおよびユーティリティの使用方法

このコースの詳細については、OpenText トレーニングサービス Web サイト (https://www.netiq.com/training/) を参照してください。

監視のためのリソース

OpenText DSTrace ユーティリティは Windows と Linux で動作します。このツールは、OpenText eDirectory の膨大なリソースを監視するのに役立ちます。DSTrace の詳細については、次を参照してください。

- 258 ページの 「トレースを環境設定する」
- “Looking Into the Directory Services Trace (DSTrace) Options” (http://support.novell.com/techcenter/articles/anp20010801.html)
- “More on Using the DSTrace Command” (http://support.novell.com/techcenter/articles/anp20010901.html)

また、OpenText eDirectory 環境用の他の管理ソリューションを提供するサードパーティの製品も使用できます。詳細については、次の Web サイトを参照してください。

- Symantec 社 (http://www.symantec.com/compliance/)
- Blue Lance (http://www.bluelance.com)
- Quest (http://www.quest.com/active-directory/)

弊社のパートナーが提供していない OpenText eDirectory の特性を監視または監査する必要がある場合には、OpenText コンサルティングサービスが、OpenText Event System を使用してカスタマイズされた評価と監査を行うためのお手伝いをいたします。

ハードウェアのアップグレードやサーバの交換

このセクションでは、ハードウェアをアップグレードまたは交換する際に、特定のサーバ上の OpenText eDirectory を移す、または保護するための情報について説明します。ここでの説明は、「459 ページの 「OpenText eDirectory のバックアップと復元」」の情報に基づいています。

Backup OpenText eDirectory Management Tool を使用すれば、次の操作を行うための OpenText eDirectory 情報を準備できます。

- 596 ページの 「サーバを交換しないでハードウェアまたはストレージデバイスを計画的にアップグレードする」
- 600 ページの 「サーバの計画的な交換」

サーバを交換しないでハードウェアまたはストレージデバイスを計画的にアップグレードする

記憶デバイスや RAM などのハードウェアのアップグレードを計画している場合、Backup eMtool を使用して OpenText eDirectory のコールドバックアップを行い、ファイルシステムのバックアップも行います。これにより、サーバの OpenText eDirectory 識別情報とファイルシステムデータが保護されます。このバックアップには、次の利点があります。

- 記憶デバイスを交換する場合、バックアップによって古い記憶デバイスから新しい記憶デバイスに情報を移すことができます。

- OpenText eDirectory が格納されたディスクパーティションまたはディスクボリュームを含む記憶デバイスを交換する場合、このバックアップ情報を用いれば、復元プロセスを使用して OpenText eDirectory データベースを新しい記憶デバイス上に再構築できます。
- OpenText eDirectory のコールドバックアップを実行し、その後でデータベースをクローズしておけば、バックアップ後のデータベースの変更を心配することなくハードウェアをアップグレードし、データベースを移すことができます。
- 何か問題が発生した場合、バックアップを使用して復元できます。

OpenText eDirectory のコールドバックアップを実行する場合、オプションを使用してサーバ上の OpenText eDirectory をロックし、使用不可にする必要があります。これにより、バックアップ後のデータ変更を防げます。このサーバと通信している他のサーバからは、このサーバは停止しているように見えます。通常サーバに送信される OpenText eDirectory 情報は、そのサーバと再び通信できるようになるまで、ツリー内の別のサーバに保存されます。保存された情報は、サーバがオンライン状態に戻ったときに、サーバを同期するために使用されます。

注 : OpenText eDirectory ツリー内の他のサーバは、このサーバがすぐにオンライン状態に戻ると期待しているため、アップグレードをすばやく完了させ、できるだけ早くサーバ上の OpenText eDirectory データベースをオープンする必要があります。

ハードウェアのアップグレードを計画的に実行するには、次の手順に従います。

1. アップグレードによりサーバに問題が発生するかもしれないと心配なら、必要に応じて、使用する別のコンピュータを準備するとよいでしょう。

   詳細については、600 ページの 「1. サーバ交換の準備」を参照してください。

2. OpenText eDirectory データベースのコールドバックアップを実行し、その後でデータベースをクローズし、ロックしたままにしておくには、クライアントコマンドを次のように使用します。NICI を使用する場合は、セキュリティファイルもバックアップします。

   ```
   backup -f backup_filename_and_path
   -l log_filename_and_path -t -c -o -d
   ```

   NICI を使用する場合は、NICI ファイルをバックアップします。クライアントとスイッチの使用についての詳細は、「487 ページの 「DSBK による手動バックアップ」」と「491 ページの 「バックアップ / 復元のコマンドラインオプション」」を参照してください。

   これで、OpenText eDirectory データベースはロックされました。手順を完了するまでは、サーバ上で新たなデータ変更が実行されないように、データベースをロックしたままにしておく必要があります。

   サーバが使用できない時間を最小限に抑えるために、以降の手順をすばやく完了させます。

3. お好みのバックアップツールを使用して、ファイルシステムをバックアップします。

   データベースをバックアップした*後*で、ファイルシステムのバックアップを行うのは重要です。これにより、OpenText eDirectory バックアップファイルが、他のファイルシステムと一緒にテープに保存されます。

4 サーバを停止させ、ハードウェアを交換します。

5 ハードウェアを交換した後で、ハードウェア変更の種類に応じた以下の手順を実行します。

ハードウェア変更の種類 ...	実行する手順
記憶デバイスに変更がない場合	サーバを起動し、データベースのロック解除を行います。
記憶デバイスの交換を行ったが、OpenText eDirectory が格納されたディスクパーティション / ボリュームに変更はない場合	1. サーバと OpenText eDirectory を起動します。 2. 交換した記憶デバイス上にあったディスクパーティション / ボリュームのファイルシステムだけを復元します。 3. OpenText eDirectory データベースのロックを解除します。

ハードウェア変更の種類 ...	実行する手順
OpenText eDirectory が保存された記憶デバイスを交換した場合	1. 必要に応じてオペレーティングシステムをインストールします。 2. 記憶デバイスの変更により影響を受けたディスクパーティション上に、ファイルシステムを復元します。 3. 新しい記憶デバイス上の、新しい一時ツリー内に、OpenText eDirectory をインストールします。 4. バックアップから OpenText eDirectory を復元します (元のツリー内に配置されます)。このとき、復元した後で OpenText eDirectory がクローズされ、ロックされたままになるようにオプションを指定します。restore-r-f *backup_filename_and_path* -l *log_filename_and_path* などのコマンドを使用します。インクルードファイルに列挙されたファイルをバックアップしてから、NICI ファイルを別個に復元する場合は、-u オプションを追加します。 5. eDirectory データベースのロックを解除します。 6. NICI セキュリティファイルを復元した場合は、復元を完了した後で、サーバを再起動してセキュリティシステムを再初期化します。 7. サーバが通常通り応答するかどうかをチェックします。OpenText iMonitor を使用してサーバとその同期をチェックします。 8. このサーバでロールフォワードログを使用していた場合、復元を完了した後で、ロールフォワードログ設定を再作成します。ロールフォワードログ記録を有効にしてから、改めてフルバックアップも取る必要があります。復元した後は、設定がデフォルトの状態にリセットされます。つまり、ロールフォワードログ記録がオフになります。フルバックアップが改めて必要となるのは、スケジュールに従って次に無人でのフルバックアップが取られるまでに、再び障害が起こる可能性があるためです。

サーバが通常どおりに応答しない場合、次にいずれかの方法によって回復する必要がある場合があります。

- 変更前のハードウェア環境設定が機能していたことから、これを再作成します。
- 作成したファイルシステムと OpenText eDirectory のバックアップを使用して、このサーバの識別情報を別のマシンに移します。詳細については、600 ページの 「サーバの計画的な交換」を参照してください。

サーバの計画的な交換

次に示す手順は、サーバの OpenText eDirectory 資格情報とファイルシステムデータを別のマシンに移して、実際にサーバを置き換えた場合です。ここでは、古いサーバをサーバ A とし、それに置き換わるサーバをサーバ B としています。

Backup eMTool を使用して OpenText eDirectory のコールドバックアップ (データベースをクローズした状態でのバックアップ) を実行し、さらにお好みのツールを使用してファイルシステムのバックアップをして、サーバの交換に備えます。このバックアップ情報を用いれば、復元プロセスを使用して、新しいコンピュータ上にサーバを再構築できます。

OpenText eDirectory のコールドバックアップを実行する場合、オプションを使用してサーバ A 上の OpenText eDirectory をロックし、使用不可にする必要があります。これにより、バックアップ後のデータ変更を防げます。このサーバと通信している他のサーバからは、このサーバは停止しているように見えます。通常サーバに送信される OpenText eDirectory 情報は、そのサーバと再び通信できるようになるまで、ツリー内の別のサーバに保存されます。保存された情報は、新しいコンピュータであるサーバ B 上で、サーバがオンライン状態に戻ったときに同期するために使用されます。

注 : OpenText eDirectory ツリー内の他のサーバは、このサーバがすぐにオンライン状態に戻ると期待しているため、できるだけ早く交換してサーバに OpenText eDirectory 情報を復元する必要があります。

サーバを置き換えるための手順の概要を次に示します。

1. 交換する際のサーバ A の停止時間を短くするには、「600 ページの 「1. サーバ交換の準備」」で説明しているようにサーバ B にオペレーティングシステムをインストールするなどして、交換前にできるだけサーバ B の準備を整えておきます。
2. 601 ページの 「2. OpenText eDirectory のバックアップの作成」の説明に従って、サーバ A の OpenText eDirectory およびファイルシステムのバックアップを行います。
3. 「602 ページの 「3. サーバ交換のための OpenText eDirectory 情報の復元」」の説明に従って、サーバ B に情報を移します。

1. サーバ交換の準備

次に示すサーバ A とサーバ B のチェックリストを使用して、サーバ A を交換する準備ができているかを確認します。開始する前にサーバ B の準備をしておけば、あるコンピュータから別のコンピュータへ転送する間のサーバの停止時間を減らすことができます。

サーバ A の準備

- ❒ サーバ A に最新のバージョンのオペレーティングシステムがインストールされていることを確認します。
- ❒ Tree パーティションのマスタを保持しているサーバで DSRepair を実行し、さらに時刻同期を実行して、サーバ A のツリーが正常に機能していることを確認します。
- ❒ サーバ A のデータベースで DSRepair を実行します。サーバ A が完全に同期されていることを確認します。

サーバ B の準備

- ❒ 最新バージョンのオペレーティングシステムをインストールします。このオペレーティングシステムは、サーバ A のものと同じである必要があります。
- ❒ サーバ B を新しい一時ツリーに配置し、OpenText eDirectory をインストールします。

  (602 ページの 「3. サーバ交換のための OpenText eDirectory 情報の復元」の過程で OpenText eDirectory を復元するには、サーバ B をサーバ A が配置されていた元のツリー内に配置します)。

次のセクション (601 ページの 「2. OpenText eDirectory のバックアップの作成」) の手順に進みます。

2. OpenText eDirectory のバックアップの作成

サーバ交換の前に、OpenText eDirectory のバックアップを作成する必要があります。600 ページの 「1. サーバ交換の準備」の手順を完了したら、Client を使用し、バックアップの後でデータベースを使用不可にしてロックする詳細オプションを設定して、サーバ A 上の OpenText eDirectory データベースのコールドバックアップを実行します。

OpenText eDirectory のコールドバックアップ (データベースがクローズ中のバックアップ) を作成し、その後でデータベースをクローズのままにしておくには :

1. 600 ページの 「1. サーバ交換の準備」が完了していることを確認します。
2. クライアントで、-c、-o、および -d スイッチを指定した次のような backup コマンドを用いてサーバ A 上の OpenText eDirectory データベースのコールドバックアップを実行し、完了後もデータベースをクローズしてロックしたままにします。

   backup -f *backup_filename_and_path* -l *log_filename_and_path* -t -c -o -d

   NICI を使用する場合は、NICI ファイルをバックアップします。クライアントとスイッチの使用についての詳細は、「487 ページの 「DSBK による手動バックアップ」」と「491 ページの 「バックアップ / 復元のコマンドラインオプション」」を参照してください。

   サーバ A の OpenText eDirectory データベースは現在ロックされています。データベースをサーバ B 上に復元しツリー内に戻すまでは、サーバ上で新たなデータ変更が実行されないように、データベースをロックしたままにしておく必要があります。

   サーバアップグレードまたはサーバ交換の残りの手順を迅速に完了させ、サーバが使用できない時間を最小限に抑えます。
3. サーバ A のファイルシステムのフルバックアップを作成します。

   データベースをバックアップした *後* で、ファイルシステムのバックアップを行うのは重要です。これにより、OpenText eDirectory バックアップファイルが、残りのファイルシステムと一緒にテープに保存されます。

   SMS の使用の詳細については、「*SMS を使用したデータのバックアップ* (https://www.novell.com/documentation/open-enterprise-server-2018/bkup_sms_lx/data/hhc3nq5m.html)」を参照してください。

4 サーバ A 上の OpenText eDirectory データベースをロックし、サーバ A をネットワークから外します。

続いて「602 ページの 「3. サーバ交換のための OpenText eDirectory 情報の復元」」の手順を実行します。

3. サーバ交換のための OpenText eDirectory 情報の復元

サーバ A の OpenText eDirectory 識別情報およびファイルシステムをサーバ B に移すには :

1 「600 ページの 「1. サーバ交換の準備」」および「601 ページの 「2. OpenText eDirectory のバックアップの作成」」が完了していることを確認します。

2 サーバ B が起動し、eDirectory が実行されていることを確認します。

3 restore を次のように使用して、サーバ A の OpenText eDirectory 識別情報とファイルシステムをサーバ B に移します。

3a サーバ A の OpenText eDirectory コールドバックアップファイルをサーバ B にコピーします。

サードパーティのファイル圧縮ツールは圧縮性能が良いので、そのようなツールを使用した場合、バックアップファイルはとても小さくなることがあります。これにより、ファイルのコピーを早くできる場合があります。

3b 複製した OpenText eDirectory のバックアップファイルを使用して、サーバ A の OpenText eDirectory データベースをサーバ B 上に復元します。それには、コマンドラインクライアントで、次のようなコマンドを使用します。

restore -r -f *backup_filename_and_path* -l *log_filename_and_path*

NICI を使用する場合は、NICI ファイルを復元します。インクルードファイルに列挙されたファイルをバックアップしてあった場合は、-u オプションを追加します。クライアントとスイッチの使用についての詳細は、「489 ページの 「DSBK によるバックアップファイルの復元作業」」と「491 ページの 「バックアップ / 復元のコマンドラインオプション」」を参照してください。

復元にはロールフォワードログを含める必要はありません。なぜなら、コールドバックアップを実行し、その後でデータベースをクローズしてあるからです。データベースではどのようなトランザクションも実行されていません。データベースはクローズされており、バックアップ以降、ロールフォワードログは作成されていません。

3c バックアップされたサーバ A のファイルシステムデータをサーバ B に移します。

4 NICI を使用している場合は、サーバを再起動して NICI を再初期化し、復元された NICI セキュリティファイルが使用されるようにします。

5 OpenText eDirectory データベースのロックを解除します。

6 復元が完了した後は、サーバ B がサーバ A の識別情報を正しく引き継ぎ、通常どおりに応答しているかチェックします。OpenText iMonitor を使用してサーバとその同期をチェックします。

サーバの応答が通常どおりなら、サーバの交換は完了です。これで、サーバ A から eDirectory をアンインストールして OpenText eDirectory 識別情報を削除し、このコンピュータを別の目的に使用できます。サーバ A をネットワークに戻すのは、OpenText

eDirectory を削除した後にしてください。そうしないと、OpenText eDirectory の同期でネットワークが混乱してしまいます。なぜなら、サーバ A とサーバ B の同じ識別情報により、競合が発生するためです。

7 (特定条件における処理) このサーバでロールフォワードログを使用していた場合、復元を完了した後で、ロールフォワードログ設定を作成し直します。ロールフォワードログを有効にしてから、改めてフルバックアップも取る必要があります。

復元した後は、設定がデフォルトの状態にリセットされます。つまり、ロールフォワードログがオフになっています。フルバックアップが改めて必要となるのは、スケジュールに従って次に無人でのフルバックアップが取られるまでに、再び障害が起こる可能性があるためです。

サーバ B が正常に動作せず、サーバ A の識別情報およびファイルシステムを直ちに使用できるようにする必要がある場合は、次を実行します。

1 サーバ B のネットワークケーブルを外すか、サーバ B を停止します。

2 サーバ A をネットワークに再接続し、起動してから、OpenText eDirectory データベースをオープンします。

DSRepair の実行を要求するシステムメッセージを無視します。

3 サーバ B から OpenText eDirectory を削除し、再度アップグレードを試みます。

サーバの IP アドレスの変更

サーバの IP アドレスが変更されることはあまりありません。IP アドレスが変更された場合は、すべての OpenText eDirectory インスタンスの nds.conf ファイルを新しい IP アドレスで更新する必要があります。IP アドレスが頻繁に変更される場合は、nds.conf で IP アドレスの代わりにインタフェース名を使用する必要があります。

例 : n4u.server.interfaces=eth0@1524

IP アドレスが変更されても、サーバの IP ベースのキーマテリアルオブジェクト (KMO) は自動的に更新されません。古い KMO(名前に IP が含まれている) を削除する必要はありませんが、削除すればツリーが見やすくなります。ndsconfig upgrade コマンドを実行して、KMO を再作成し、それらを NCP サーバオブジェクトと LDAP サーバオブジェクトにリンクさせます。

注 : ndsconfig upgrade を実行すると、OpenText eDirectory インスタンスが再起動されます。

これで、サーバは新しいアドレスをリスンするようになります。ツリー内に複数のサーバが存在する場合は、DSRepair ネットワーク修復オプションを実行します。

ndsrepair -N

修復オプションの実行後に、OpenText eDirectory サーバを再起動します。

サーバの IP アドレスの変更に関する詳細については、TID# 3201067 を参照してください。

ハードウェア障害後の OpenText eDirectory の復元

OpenText eDirectory が配置されたディスクパーティションまたはディスクボリュームを含むハードディスクの障害は、サーバから OpenText eDirectory が削除された状態と同じです。幸い、複数サーバ環境では、1 つのサーバがダウンしても、そのレプリカリング内の残りのサーバが正常に稼働していれば問題はありません。

OpenText eDirectory が格納されたディスクパーティションまたはディスクボリュームの障害の後で OpenText eDirectory を復元するには、480 ページの 「復元処理の準備」および489 ページの 「DSBK によるバックアップファイルの復元作業」で説明されている手順に従って、バックアップファイルから復元します。

ハードディスクの新規インストールでは、製造元から提供されている指示に従って、サーバのハードディスクが動作することを検証します。新しいハードディスクには、少なくとも置き換えられる元のドライブと同じ記憶容量が必要です。ローカルサーバ情報のファイルを使用して、環境設定情報を確認します。

注

- ウイルス対策ソフトウェアやバックアップソフトウェアのプロセスから、OpenText eDirectory サーバ上の DIB ディレクトリを除外することをお勧めします。DIB ディレクトリのバックアップには、OpenText eDirectory バックアップツールを使用します。OpenText eDirectory のバックアップの詳細については、459 ページの 「OpenText eDirectory のバックアップと復元」を参照してください。
- サーバのバックアップファイルがない場合は、Xbrowse ツールを使用して、サーバ情報の回復に役立つ情報を OpenText eDirectory に問い合わせてください。この作業は、サーバオブジェクトやその関連オブジェクトをツリーから削除する前に実行する必要があります。Xbrowse の詳細については、OpenText サポートの Web サイトを参照してください。

サブツリー検索のパフォーマンスの向上

OpenText eDirectory では、深い入れ子構造を持つ大規模なツリーに対してサブツリー検索を行う場合、パフォーマンスは検索のベース DN に関係なくフラットな状態であり続けます。この問題は、AncestorID 属性を使用することにより解決されています。AncestorID 属性はすべての祖先の entryID のリストであり、各エントリに関連付けられています。この AncestorID 属性は、サブツリー検索の間に内部で使用されます。したがって、AncestorID は検索のスコープを制限します。

この属性は、DIB のエントリを追加している間やすべてのエントリをアップグレードした後に表示されます。また、サブツリーが移動されると、サブツリーのすべてのエントリに対する属性が再表示されます。ただし、アップグレードやサブツリーの移動を行った後で属性を作成する際は、サブツリー検索時に AncestorID 属性は使用されません。したがって、サブツリーのパフォーマンスは OpenText eDirectory 以前のサブツリー検索のものと同様になります。

AncestorID がアップグレード後に更新されているかどうかを確認するには。

AncestorID が一度作成されると、NDS オブジェクトのアップグレードバージョンが 6 以上に変更されます。エージェント情報の **DIB 履歴**セクションで iMonitor を使用して、このバージョンを表示できます。

AncestorID がサブツリーの移動操作の後に更新されているかどうかを確認するには。

AncestorID が作成されている間、擬似サーバオブジェクトの属性 UpdateInProgress は、サブツリーのパーティションルートのエントリ ID のリストを保持します。AncestorID が一度表示されると、擬似サーバに属性は存在しなくなります。

AncestorID 属性が無効の場合、DSRepair は AncestorID 属性を更新します。

コンテナの準備状況

エントリキャッシュの使用率を最適化し、属性検索操作のパフォーマンスを向上させるため、FLAIM は、値のサイズが大きい属性または値の個数が多い属性を、属性コンテナと呼ばれる別の場所に格納します。デフォルトでは、次の条件を満たす属性がコンテナに自動的に移されます :

- 値の数が 25 個を超えている
- サイズが 2048 バイトを超える値を持っている

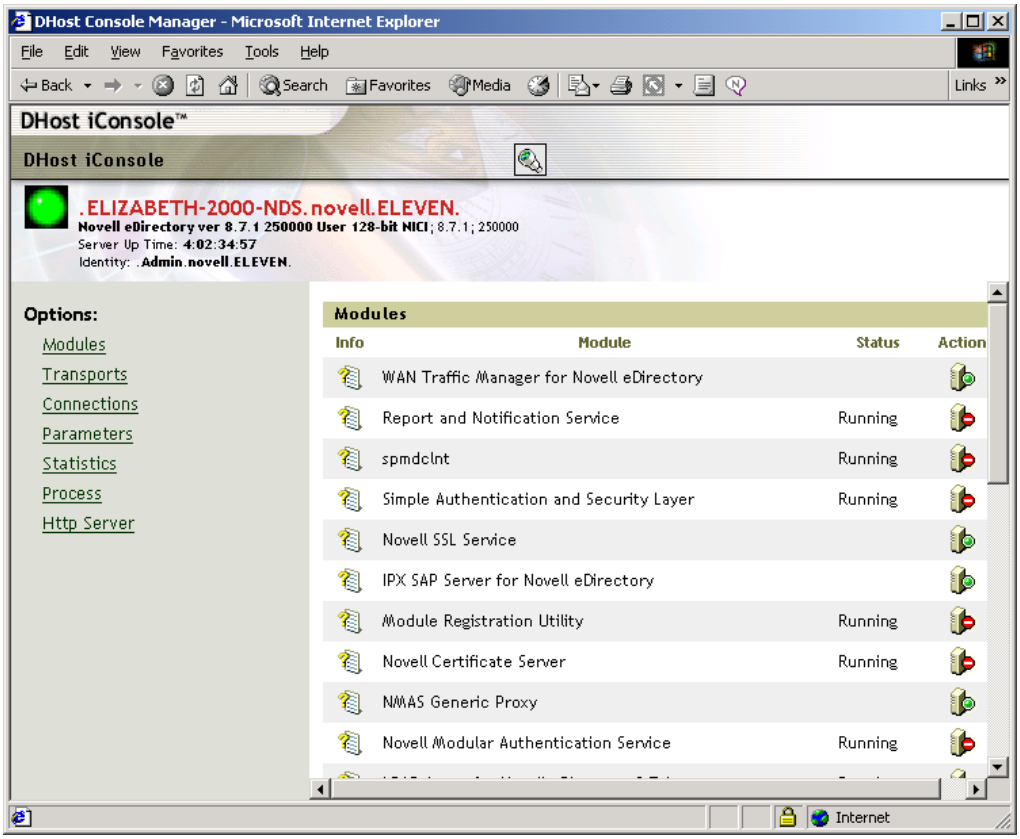
OpenText eDirectory には、別の属性コンテナへの属性の移動を制御するための柔軟性が備えられています。管理者は、必要に応じて、属性の移動をスケジュールできます。詳細については、「FLAIM Attribute Containerization」を参照してください。

20 DHost OpenText iConsole Manager

DHost OpenText iConsole Manager は Web ベースのブラウザを使用した管理ツールで、次のことが行えます。

- DHost モジュールの管理
- DHost 環境設定パラメータの照会
- DHost 接続情報の表示
- スレッドプール統計情報の表示
- DHostプロトコルスタックマネージャにより登録されたプロトコルに関する詳細情報の表示

図 20-1 *DHost OpenText iConsole Manager*



この章では、次の情報について説明します。

- 608 ページの 「DHost について」
- 609 ページの 「DHost OpenText iConsole の実行」
- 610 ページの 「OpenText eDirectory モジュールの管理」

- 611 ページの 「DHost 情報の照会」
- 613 ページの 「プロセススタック」

DHost について

Windows と Linux 向けの OpenText eDirectory ソフトウェアはすべて同じコアコードに基づいてビルドされています。Windows および Linux 版の OpenText eDirectory が別のバージョンの OpenText eDirectory と適切にやり取りするには、NCP (NetWare Core Protocol) サービスのサブセットをサポートしている必要があります。このサブセットを操作するプログラムは DHost と呼ばれます。DHost は、OpenText eDirectory 下で動作し、NCP がネイティブで備えている機能を提供します。

DHost は次のサービスを提供します。

サービス	説明
NCP エンジン	クライアントが OpenText eDirectory サーバに要求を送信したり、OpenText eDirectory サーバから応答を受信したりできるようにするパケットベースのプロトコルです。 詳細については、「NetWare Core Protocols NDK (https://www.microfocus.com/documentation/open-enterprise-server-developer-documentation/ncp/)」を参照してください。
ウォッチドッグ	ワークステーションが OpenText eDirectory サーバに接続された状態であることを確認するために使用するパケットです。 詳細については、「Watchdog Packet Spoofing」を参照してください。
接続テーブル	OpenText eDirectory サーバに付随するあらゆるプロセス、プリントサーバ、アプリケーション、ワークステーション、またはその他のエンティティに割り当てられる固有の番号です。この番号は、接続が行われるごとに異なる可能性があります。接続番号は、ネットワークセキュリティの実装やネットワークアカウントに使用されます。この番号は、ファイルサーバ接続テーブル内でのオブジェクトの場所を反映しています。さらにこの番号を使用すると、ネットワークにログインしたオブジェクトに関する情報の識別や取得が容易になります。
イベントシステム	個々のサーバのアクティビティを監視する手段をアプリケーションに提供します。
スレッドプール	独立したエンティティとして実行され、システムソフトウェアによってスケジュールされる一連の命令です。
NCP 拡張	サーバアプリケーション開発者が、NCP として実装する NLM™ ソフトウェアを作成できるようにします。 詳細については、NCP NDK の「NCP Extension (https://www.microfocus.com/documentation/open-enterprise-server-developer-documentation/ncp/ncpdocs/ncpext.htm)」を参照してください。

サービス	説明
メッセージ処理	ドキュメントを圧縮または凝縮した形式、またはドキュメントの要約で、大きなドキュメントの電子指紋として機能します。メッセージ処理は、個々のドキュメントに固有のデジタル署名を作成するために使用されます。

DHost OpenText iConsole の実行

- 609 ページの 「Windows で DHost OpenText iConsole を実行する」
- 609 ページの 「Linux で DHost OpenText iConsole を実行する」

Windows で DHost OpenText iConsole を実行する

1 Web ブラウザを開きます。

2 アドレス (URL) フィールドに、次の形式で入力します。

- http: http://*server.name*:*port*/dhost の場合

  例 : http://MyServer:8028/dhost

- https: https://*server.name*:*port*/dhost の場合

  例 : https://MyServer:8030/dhost

DHost OpenText iConsole へのアクセスに、サーバの IP アドレスを使用することもできます。例 :

```
http://137.65.135.150:8028/dhost
```

3 ユーザ名、コンテキスト、パスワードを指定します。

Linux で DHost OpenText iConsole を実行する

1 Web ブラウザを開きます。

2 アドレス (URL) フィールドに、次の形式で入力します。

- http: http://*server.name*:*port*/dhost の場合

  例 : http://MyServer:8028/dhost

- https: https://*server.name*:*port*/dhost の場合

  例 : https://MyServer:8030/dhost

DHost OpenText iConsole へのアクセスに、サーバの IP アドレスを使用することもできます。例 :

```
http://137.65.135.150:8028/dhost
```

3 ユーザ名、コンテキスト、パスワードを指定します。

OpenText eDirectory モジュールの管理

DHost OpenText iConsole のモジュールページでは、使用可能な OpenText eDirectory サービスとその状態についての情報が表示されます。また、このモジュールページからこれらのサービスを開始または停止 (ロードまたはアンロード) できます。

ロードやアンロードができるのは、LDAP、SNMP、HTTPSTK などの非対話型モジュールだけです。

モジュールページには、次に示す属性があります。

属性	説明
Info	 をクリックして、選択したモジュールのモジュールの説明、ファイル名、モジュールハンドル、属性、および共有オブジェクト名を表示します。
モジュール	モジュール名を表示します。
ステータス	モジュールが実行されているかどうかを表示します。
アクション	モジュールが実行可能かどうかを示します。モジュールのステータスは、次の 3 つのうちのどれかになります。 •  は、モジュールがシステムモジュールで、アンロードできないことを示します。 •  は、モジュールがロード可能で、ロードの準備ができていることを示します。 •  は、モジュールが実行中であることを示します。

- 610 ページの 「Windows でモジュールをロードまたはアンロードする」
- 611 ページの 「Linux でモジュールをロードまたはアンロードする」

Windows でモジュールをロードまたはアンロードする

1 Web ブラウザを開きます。

2 アドレス (URL) フィールドに、次の形式で入力します。

http://*server.name*:*port*/dhost

たとえば、次のように入力します。

```
http://MyServer:80/dhost
```

DHost OpenText iConsole へのアクセスに、サーバの IP アドレスを使用することもできます。例 :

```
http://137.65.135.150:80/dhost
```

3 ユーザ名、コンテキスト、パスワードを指定します。

4 ［モジュール］をクリックします。

5 をクリックしてモジュールをロードするか、をクリックしてモジュールをアンロードします。

Linux でモジュールをロードまたはアンロードする

1 Web ブラウザを開きます。

2 アドレス (URL) フィールドに、次の形式で入力します。

http://*server.name*:*port*/dhost

たとえば、次のように入力します。

```
http://MyServer:80/dhost
```

DHost OpenText iConsole へのアクセスに、サーバの IP アドレスを使用することもできます。例 :

```
http://137.65.135.150:80/dhost
```

3 ユーザ名、コンテキスト、パスワードを指定します。

4 ［モジュール］をクリックします。

5 をクリックしてモジュールをロードするか、をクリックしてモジュールをアンロードします。

DHost 情報の照会

DHost OpenText iConsole Manager を使用すれば、次に示す情報を照会できます。

- 環境設定パラメータ
- PSTACK マネージャを使用して登録されたプロトコル
- 接続プロパティ
- スレッドプールの概要

環境設定パラメータを表示する

環境設定パラメータは、Linux に特有のものです。

DHost OpenText iConsole Manager で、［パラメータ］をクリックします。詳細については、609 ページの 「Linux で DHost OpenText iConsole を実行する」を参照してください。

環境設定パラメータには、次の情報が表示されます。

オプション	説明
パラメータ名	環境設定パラメータの名前を表示します。
［デフォルト値］	環境設定パラメータのデフォルト値を表示します。

オプション	説明
設定する値	現在の設定値を表示します。
Minimum value	パラメータに設定できる最小値を表示します。
Maximum value	パラメータに設定できる最大値を表示します。
タイプ	パラメータに設定できる値のタイプを表示します。

詳細については、『*OpenText™ eDirectory CE 24.4 (v9.3) インストールガイド*』の「「環境設定パラメータ」」を参照してください。

プロトコル情報を表示する

DHost OpenText iConsole Manager で、［**トランスポート**］をクリックします。

次のプロトコル情報が表示されます。

- ID
- プロトコル
- トランスポート

接続プロパティを表示する

DHost OpenText iConsole Manager で、［**接続**］をクリックします。

次の接続プロパティが表示されます。

- 接続
- フラグ
- ID
- 表示名
- トランスポート
- 認証名
- SEV 回数
- 最終アクセス
- ロック状態

スレッドプールの統計情報を表示する

DHost OpenText iConsole Manager で、［**統計情報**］をクリックします。

次のスレッドプール統計情報が表示されます。

- Spawned Threads(生成スレッド)
- Dead Threads(停止スレッド)

- Idle Threads(アイドルスレッド)
- Worker Thread(動作スレッド)
- Peak Worker Thread(ピーク動作スレッド)
- Ready for Work Thread(動作待機スレッド)
- Ready Queue Peak Worker Threads(待機キューピーク動作スレッド)
- Ready Queue Max Wait Time(待機キュー最大待ち時間)
- Schedule Delay Minimum Time(最小スケジュール遅延時間)
- Schedule Delay Maximum Time(最大スケジュール遅延時間)
- Schedule Delay Average Time(平均スケジュール遅延時間)
- Waiting For Work(動作待ち)
- Peaking Waiting For Work(ピーク動作待ち)

プロセススタック

プロセススタックには、DHost のプロセス空間で現在実行されているすべてのスレッドのリストが含まれます。スレッドに関する詳細な情報は、スレッド ID をクリックして取得できます。この機能は、主に OpenText エンジニアやサポート担当者により、ローレベルのデバッグツールとして使用されます。

このオプションは、Windows でのみ使用できます。

1. Web ブラウザを開きます。
2. アドレス (URL) フィールドに、次の形式で入力します。

   http://*server.name*:*port*/dhost

   たとえば、次のように入力します。

   `http://MyServer:80/dhost`

   DHost OpenText iConsole へのアクセスに、サーバの IP アドレスを使用することもできます。例 :

   `http://137.65.135.150:80/dhost`
3. ユーザ名、コンテキスト、パスワードを指定します。
4. [**プロセス**] をクリックします。
5. スレッドのコールスタックを表示するには、スレッド ID をクリックします。

21 sadmin パスワードの設定

OpenText eDirectory がロードされていない場合に HTTPSTK(HTTP プロトコルスタック) へのアクセスに使用できる、事前に設定された管理者ユーザをセットアップしておくことができます。事前に設定された管理者ユーザ (sadmin) には、OpenText eDirectory 管理者ユーザオブジェクトと同等の権利があります。サーバが、OpenText eDirectory が適切に機能していない状態の場合、このユーザとしてサーバにログインし、OpenText eDirectory を使用せずに実行できる必要なすべての診断およびデバッギング作業を実行します。

注 : sadmin ユーザ名は大文字と小文字が区別されます。

---

ndspassstore ユーティリティを使用して、Windows システムと Linux システム上で sadmin パスワードを設定します。

サーバコンソールで次のコマンドを入力します。

ndspassstore -a sadmin -w *<password>*

ここで、sadmin(管理者コンテキスト) は管理権限を持つユーザの完全識別名で、password は認証用のパスワードです。複数インスタンスのシナリオでは、該当するインスタンスを選択します。

例 : ndspassstore -a sadmin -w pass

デフォルトで ndspassstore は、Windows では **C:\Novell\NDS** に、UNIX では **/opt/novell/eDirectory/bin** にあります。

22 OpenText eDirectory 管理ツールボックス

OpenText eDirectory 管理ツールボックス (eMBox) を使用すると、サーバ上でもリモートでも OpenText eDirectory のバックエンドユーティリティすべてにアクセスできます。

eMBox を OpenText Identity Console とあわせて使用すると、DSRepair、DSMerge、バックアップと復元、サービスマネージャなどの OpenText eDirectory ユーティリティに Web ベースでアクセスできます。

OpenText Identity Console で OpenText eDirectory のメンテナンスに関係する次のタスクを実行するには、RAC を設定する必要があります。

- バックアップ環境設定
- ツリーの結合
- OpenText eDirectory の修復
- サーバの修復
- 同期の修復
- レプリカの修復
- レプリカリングの修復
- 復元
- スキーマの保守
- サービスマネージャ
- ツリーのマージ
- ツリー名の変更

すべての機能は、ローカルサーバまたはリモートのいずれからでもコマンドラインクライアントを通じて使用できます。クライアントを使用して、1 つのサーバまたはワークステーションから複数のサーバに対するタスクを実行できます。

バックアップ、DSRepair、DSMerge、スキーマの操作、および OpenText eDirectory サービスマネージャなどのすべての eMTool (OpenText eDirectory Management Tool) を実行するには、OpenText eDirectory サーバに eMBox をロードして実行する必要があります。

注 : eMTool の使用方法については、925 ページの 「Linux でのユーティリティのトラブルシューティング」を参照してください。

この節では、次の項目について説明します。

- 618 ページの 「コマンドラインクライアントの使用」
- 628 ページの 「ログの記録の使用」
- 629 ページの 「eMBox クライアントを使ったバックアップ / 復元作業」

コマンドラインクライアントの使用

にアクセスする方法の 1 つは、eMBox の Java コマンドラインクライアントを使用することです。このコマンドラインクライアントには、対話式モードとバッチモードの 2 つのモードがあります。対話式モードでは、コマンドを一度に 1 つづつ実行します。バッチモードでは、コマンドのグループを自動で実行できます。コマンドラインクライアントにはログサービスがあり、いずれのモードでも使用できます。

コマンドラインクライアントは、Java アプリケーションです。これを実行するには、Azul ZuluOpenJDK の最新バージョン (1.8 以上) をインストールする必要があります。また、古いバージョンの Java すべてで、使用可能なパッチアップグレードをインストールしてアップグレードしてください。最新バージョンの Java をインストールしたら、次の環境変数をエクスポートします。

- EDIR_JAVA_HOME
- JAVA_HOME
- JRE_HOME

注

- Linux で前述の環境変数が何も検出されないと、コマンドラインクライアントはデフォルト PATH 環境変数で Java バイナリを検索します。
- OpenText eDirectory 9.1 SP2 以降では、Azul ZuluOpenJDK 1.8.0_192 がサポートされています。

例

環境変数の例のいくつかを以下に記します。

- **Linux**
  - EDIR_JAVA_HOME=/usr/java/java1.8.0_131
  - JAVA_HOME= /usr/java/java1.8.0_131
  - JRE_HOME= /usr/java/java1.8.0_131/jre
- **Windows**
  - EDIR_JAVA_HOME= C:\Program Files\Java\jdk1.8.0_131
  - JAVA_HOME= C:\Program Files\Java\jdk1.8.0_131
  - JRE_HOME= C:\Program Files\Java\jdk1.8.0_131\jre

また、ファイアウォール越しに管理対象のサーバにアクセスできる必要があります。1 つのサーバまたはワークステーションから、複数のサーバに対してタスクを実行できます。

注 : OpenText eDirectory Management Toolbox は、コマンドラインクライアントとコマンドラインヘルプの両方とも、英語にのみ対応しています。

この節では、次の項目について説明します。

- 619 ページの 「コマンドラインヘルプを表示する」
- 619 ページの 「コマンドラインクライアントを対話式モードで実行する」
- 624 ページの 「コマンドラインクライアントをバッチモードで実行する」
- 626 ページの 「eMBox コマンドラインクライアントのオプション」
- 627 ページの 「クライアントを使用してセキュア接続を確立する」
- 627 ページの 「OpenText eDirectory のポート番号を確認する」

コマンドラインヘルプを表示する

クライアントを実行する前に、の一般的なコマンドラインヘルプを表示するには、次の操作を実行します。

- Linux: コマンドラインで、「edirutil -?」と入力します。
- Windows: *drive*\novell\nds\\edirutil.exe -? を実行します。

インタラクティブモードでのインタラクティブなコマンドラインヘルプを表示するには、クライアントのプロンプトで疑問符 (?) を入力します。たとえば、「Client> ?」のように入力します。

ヘルプには、626 ページの 「eMBox コマンドラインクライアントのオプション」で示すようなコマンドラインオプションに関する情報が表示されます。

コマンドラインクライアントを対話式モードで実行する

対話式モードでは、コマンドを一度に 1 つづつ実行します。

この節では、次の項目について説明します。

- 620 ページの 「OpenText eDirectory サーバでクライアントを実行する」
- 620 ページの 「ワークステーションでクライアントを実行する」
- 620 ページの 「クライアント用にパスおよびクラスパスをセットアップする」
- 621 ページの 「サーバにログインする」
- 622 ページの 「使用言語、タイムアウト、およびログファイルを設定する」
- 622 ページの 「eMTool とそのサービスを表示する」
- 623 ページの 「特定のサービスを実行する」
- 623 ページの 「現在のサーバからログアウトする」
- 623 ページの 「eMBox クライアントを終了する」

OpenText eDirectory サーバでクライアントを実行する

クライアントおよび Sun JVM 1.3.1 は、OpenText eDirectory と同時にインストールされています。OpenText eDirectory サーバ上でクライアントをインタラクティブモードで開くには、次の操作を実行します。

- Linux: コマンドラインで、「edirutil -i」と入力します。
- Windows: *drive*\novell\nds\edirutil.exe -i を実行します。

edirutil ファイルは、クライアントを実行するためのショートカットです。これは、Java 実行可能ファイルと、クライアントが OpenText eDirectory と一緒にインストールされているデフォルトの場所を指しています。この情報は、620 ページの 「クライアント用にパスおよびクラスパスをセットアップする」に記載されているようにして手動で入力することもできます。

管理するサーバに対してコマンドラインクライアントを使用するには、ファイアウォールの後ろ側へのアクセスが必要になります。したがって、リモートからの操作には VPN アクセスが必要となります。

ワークステーションでクライアントを実行する

OpenText eDirectory サーバ以外のマシン上でクライアントを使用するには :

- eMBoxClient.jar ファイルを OpenText eDirectory サーバから自分のマシンにコピーします。
  - Windows: \novell\nds\\eMBoxClient.jar
  - Linux: /opt/novell/eDirectory/lib/nds-modules//eMBoxClient.jar
- Sun JVM 1.3.1 がインストールされていることを確認します。
- 管理するサーバに対してコマンドラインクライアントを使用するためには、ファイアウォール越しにアクセスできることを確認します。

ワークステーション上では、サーバ上のようにクライアントをインタラクティブモードで開くためのショートカットとして edirutil コマンドを使用することはできません。パスおよびクラスパス内で環境をセットアップするか、パスをその都度手動で入力します。詳細については、620 ページの 「クライアント用にパスおよびクラスパスをセットアップする」を参照してください。

クライアント用にパスおよびクラスパスをセットアップする

OpenText eDirectory サーバ上でクライアントを実行している場合、Java ファイルまたは eMBoxClient.jar ファイルの場所を変更していなければ、クライアントの実行へのショートカットとして edirutil を使用することができます。詳細については、620 ページの 「OpenText eDirectory サーバでクライアントを実行する」を参照してください。

ただし、デフォルトの場所を変更した場合、または eMBoxClient.jar ファイルをサーバでないマシン上で実行している場合、あるいはクラスパスを手動で入力したい場合は、クライアントのパスおよびクラスパスをこのセクションの説明のようにセットアップする必要があります。

次の操作を実行すれば、クライアントをコンピュータ上のどの場所からでも実行できます。

- Java 実行ファイル (Java.exe など) があるディレクトリをパスに追加するか、または Java がすでに実行されていることを確認します。

  サーバの場合、ほとんどはすでに実行されています。Windows サーバ、Linux サーバ、および UNIX サーバでは、実行ファイルのディレクトリをパスに追加する必要があります。

  ワークステーションの場合、手動セットアップが必要となる場合があります。たとえば、Windwos では、[**スタート**] > [**設定**] > [**コントロール パネル**] > [**システム**] の順にクリックします。[**詳細設定**] タブで、[**環境変数**] をクリックし、[**パス**] 変数にパスを追加します。

  パスを手動で入力するには、次の操作を実行します。Java 実行ファイルへのパスが追加されていない場合は、まずコマンドラインで Java 実行ファイルが含まれたディレクトリへ移動してから、実行する必要があります。たとえば、Windows では、「cd c:\novell\nds\\jre\bin」と入力します。

- eMBoxClient.jar ファイルへのパスを、クラスパスに追加します。

  Windows サーバまたはワークステーション : set CLASSPATH=*path*\eMBoxClient.jar

  Linux サーバまたはワークステーション : export CLASSPATH=*path*/eMBoxClient.jar

  パスを手動で入力するには、次の操作を実行します。クラスパスを指定する他の方法としては、次のように、実行するときにその都度 Java の -cp フラグを使用する方法もあります。

  ```
  java -cp path/eMBoxClient.jar -i
  ```

  たとえば、Windows では、「java -cp c:\novell\nds\\eMBoxClient.jar -i」と入力します。

これらの手順を実行した後は、次のコマンドを使用して、コンピュータ上のどの場所からでも対話式モードによる eMBox クライアントを実行できます。

```
java  -i
```

Java コマンドに関する詳細については、Oracle の Web サイト (http://www.oracle.com/technetwork/java/) で Java のマニュアルを参照してください。

サーバにログインする

サーバにログインするには、サーバ名または IP アドレス、および特定のサーバへ接続するためのポート番号を指定する必要があります。パブリックログインの場合はユーザ名とパスワードは不要です。

たとえば、クライアントを対話式モードでオープンした後で、次のように入力します。

```
login -s 137.65.123.244 -p 8030 -u admin.mycompany
-w mypassword
```

次の例に示すように、-n を使用して、クライアントとの安全でない接続を確立します。

```
login -s 137.65.123.244 -p 8028 -u admin.mycompany
-w mypassword -n
```

ポート番号についての詳細は、627 ページの 「OpenText eDirectory のポート番号を確認する」を参照してください。

使用言語、タイムアウト、およびログファイルを設定する

デフォルトの言語は、クライアントシステムの言語です。そのため、ほとんどの場合、特別に言語を設定する必要はありません。同様に、タイムアウトもほとんどの場合、デフォルトの設定で問題ありません。ログファイルを設定するには、ファイル名とファイルを開くモード (追加または上書き) を指定します。

次の表に、コマンド例を示します。

コマンド	説明
set -L en,de	使用言語を英語、ドイツ語の優先順で設定します。
set -T 100	タイムアウトを 100 秒に設定します。タイムアウトは、サーバからの応答を待つ時間を設定するものです。
set -l mylog.txt -o	ログファイルとして mylog.txt を使用し、上書きモードで開きます。 デフォルトの設定は「追加」です。

eMTool とそのサービスを表示する

サーバにログインしたら、list コマンドを使用して、そのサーバ上で使用できるサービスを表示できます。

list コマンドを使用すると、次に示す eMTool とそのサービスが動的に表示されます。

eMTool	説明
バックアップ	OpenText eDirectory Backup eMTool
DSMerge	OpenText eDirectory Merge eMTool
DSRepair	OpenText eDirectory Repair eMTool
DSSchema	OpenText eDirectory Schema Operations eMTool
service	OpenText eDirectory Service Manager eMTool

リストを強制的にリフレッシュするには、-r を使用します。サービスの詳細を表示するには、-t を使用します。コマンド形式のみを表示するには、-f を使用します。

次の表に、コマンド例を示します。

コマンド	説明
リスト	サーバ上で使用できる eMTool を表示します。
list -r	eMTool リストをリフレッシュします。
list -t backup	backup サービスの詳細を表示します。
list -t dsrepair	DSRepair サービスの詳細を表示します。
list -t dsmerge -f	DSMerge サービスのコマンド形式だけを表示します。

特定のサービスを実行する

サーバにログインした後は、各 eMTool サービスを使用してタスクを実行できます。次に例を示します。

コマンド	説明
dsrepair.rld	ローカルデータベースを修復します。
backup.getconfig	バックアップ設定情報を取得します。

詳細については、次を参照してください。

- 629 ページの 「eMBox クライアントを使ったバックアップ / 復元作業」
- 322 ページの 「クライアントを使用したツリーのマージ」
- 369 ページの 「クライアントを使用したデータベースの修復」
- 212 ページの 「クライアントのサービスマネージャ eMTool を使用する」

現在のサーバからログアウトする

現在のセッションからログアウトするには、次のコマンドを使用します。

logout

別のサーバにログインする場合は、このコマンドを使用する必要がありません。現在のサーバから自動的にログアウトされます。

eMBox クライアントを終了する

eMBox クライアントを終了するには、次のいずれかのコマンドを使用します。

exit

または

quit

コマンドラインクライアントをバッチモードで実行する

クライアントをバッチモードで実行するには、次の 3 つの方法があります。

- 624 ページの 「単一タスク」
- 624 ページの 「内部バッチファイル」
- 625 ページの 「システムバッチファイル」

システムバッチファイルと内部バッチファイルを組み合わせて使用することで、コマンドをより自由に実行でき、頻繁に実行するコマンドの編成や再使用が可能です。

単一タスク

コマンドラインから単一のタスクをバッチモードで実行するには、コマンドに -t オプションを使用してツールとタスクを指定し、-i オプション (-i は対話式モードを指定するオプションです) を省くだけです。次に例を示します。

```
java -s 137.65.123.244 -p 8008 -u admin.mycompany
-w mypassword -l mylog.txt -t dsrepair.rld -n
```

異なるサーバ上で複数のタスクを実行する場合や、頻繁に実行するタスクの場合は、内部バッチファイルを使用する方が便利です。詳細については、次のセクション (624 ページの「内部バッチファイル」) を参照してください。

内部バッチファイル

クライアント内部バッチファイルを使用してクライアントをバッチモードで実行するには、インタラクティブモードで実行するコマンドのグループを記述したファイルを作成する必要があります。

クライアント内部バッチファイルを使用すると、バッチファイルに記述したすべてのコマンドを自動的に実行できます。複数のツールを使用した複数のタスクを、同一のサーバ上でタスクごとにログインとログアウトを繰り返すことなく実行できます。また 1 つのサーバから、複数のサーバに対して複数のツールを使用したタスクを実行できます。

内部バッチファイルを使用すれば、頻繁に実行するコマンドをまとめ、再利用することができます。それらのコマンドは、実行に際してその都度コマンドラインから手動で入力する必要はありません。

内部バッチファイルを実行するには、コマンドラインでクライアントのコマンドを使用します。たとえば、次のコマンドは、サーバにログインし、mybatch.mbx ファイル内に列挙されたコマンドを実行します。

```
java -s 137.65.123.244 -p 8008 -u admin.mycompany -w mypassword -l
mylog.txt -o -b mybatch.mbx -n
```

もう 1 つの方法は、同様のコマンドをシステムバッチファイル内に記述し、そのファイルがサーバ上で自動実行されるようにスケジュールすることです。詳細については、625 ページの 「システムバッチファイル」を参照してください。

次に、内部バッチファイルの例を示します。このファイルには、実行するコマンドの例および別のサーバへログインする例が記述されています。この例では、クライアントを開いたときに、サーバにログインしているものと仮定しています。それぞれのコマンドは別々の行に入力する必要があります。# で始まる行はコメントです。

```
# This file is named mybatch.mbx.
# This is an example of commands you could use in
# an internal command batch file.

# Backup commands
backup.getconfig
backup.backup -b -f mybackup.bak -l backup.log -t -w

# DSRepair commands
dsrepair.rld

# Log in to a different server
login -s 137.65.123.255 -p 8008 -u admin.mycompany -w mypassword -n

# DSMerge commands
dsmerge.pr -u admin.mycompany -p admin.mycompany -n mypassword # Schema
Operations
dsschema.rst
dsschema.dse
dsschema.rls
dsschema.gsu
dsschema.scc
dsschema.irs -n LocalTree

# DSService commands
service.serviceList

# End of example.
```

システムバッチファイル

他のコマンドラインツールと同様に、クライアントコマンドが含まれたシステムバッチファイルを作成し、それらをコマンドラインから手動で実行したり、サーバ上で自動で実行されるようにスケジュールしたりできます。たとえば、633 ページの 「バッチファイルと eMBox クライアントによる無人バックアップ」で示す例のように、システムバッチファイルを使用して、自動でバックアップを実行できます。

1 つのサーバから複数のサーバに対して、複数のツールを使用したタスクを実行できます。

システムバッチファイルでは、クライアントの単一コマンドと内部バッチファイルを組み合わせて使用できます。これにより、コマンドをより自由に実行でき、頻繁に実行するコマンドの編成や再使用ができます。詳細については、624 ページの 「内部バッチファイル」を参照してください。

バッチファイルを自動で実行する方法については、ご使用のオペレーティングシステムのマニュアルまたはサードパーティ製スケジューリングソフトウェアのマニュアルを参照してください。

eMBox コマンドラインクライアントのオプション

オプション	説明
-? または -h	ヘルプ情報を表示します。
-i	コマンドを一度に 1 つづつ、対話的に実行します。
-s *server*	サーバの名前または IP アドレスを指定します。 Default=127.0.0.1
-p *port*	サーバのポート番号を指定します。 Default=8008
-u *user*	ユーザ DN. たとえば、admin.mycompany と指定します。 Default=anonymous
-w *password*	-u で指定したユーザのパスワードを指定します。
-m *mode*	ログインモードを指定します。 Default=dclient
-n	安全な SSL 接続を試行しません。保護されていない接続を使用します。 このオプションを使用しない場合、クライアントでは SSL 接続を確立しようとします。そのため、クラスパスには JSSE ファイルが必要となり、これがない場合はエラーが返されます。詳細については、627 ページの「クライアントを使用してセキュア接続を確立する」を参照してください。
-l *log file*	ログファイルの名前を指定します。
-o	ログファイルを開いた後は、上書きします。
-T *timeout*	サーバから応答を待つ時間 (秒) を指定します。
-L *language*	使用する言語を優先順にカンマで区切って指定した一覧です。たとえば、「en-US,de_DE」のように指定します。このオプションのデフォルトは、クライアントシステムの言語です。
-t [*tool.*]*task options*	この接続での単一のサービスを実行します。-t に続く文字列は、有効なコマンドである必要があります。
-b *batch file*	バッチファイルで指定した一連のサービスを実行します。バッチファイル内のコマンドは、行を分けて記述する必要があります。# で始まる行はコメントです。

クライアントを使用してセキュア接続を確立する

非セキュア接続を使用している場合、ユーザ名やパスワードなどの入力したすべての情報は、クリアテキストでネットワーク上に送信されます。

SSL を使用したセキュア接続を確立するには、次の操作を実行します。

- サーバへログインする際は、コマンドで -n オプションを使用しないでください。このオプションは、非セキュア接続を指定するものです。セキュア接続がデフォルトの設定です。
- クラスパスに、次に示す JSSE(Java Secure Socket Extension) ファイルがあることを確認します。
  - jsse.jar
  - jnet.jar
  - jcert.jar

  これらのファイルがない場合、クライアントは、セキュア接続が確立できないことを示すエラーを返します。

  これらのファイルや JSSE についての情報は、Oracle の Web サイト (http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136007.html) から入手できます。

OpenText eDirectory のポート番号を確認する

クライアントでサーバにログインするには、ポート番号を指定する必要があります。

OpenText eDirectory をインストール時にポート番号を指定した場合は、その番号を使用します。

すべてのプラットフォームで、デフォルトの非セキュアポートは 8008、セキュアポートは 8030 です。

次のセクションでは、OpenText eDirectory に割り当てられているポートを確認するのに役立つヒントを説明します。

- 627 ページの 「Windows の場合」
- 628 ページの 「Linux の場合」

Windows の場合

1 [スタート] > [設定] > [コントロールパネル] の順にクリックします。

2 [OpenText eDirectory サービス] アイコンをダブルクリックしてから、[トランスポート] タブをクリックします。

3 セキュアポート、または非セキュアポートを確認します。
  - 非セキュアポートを確認するには、HTTP の横のプラス記号をクリックします。
  - セキュアポートを確認するには、HTTPS の横のプラス記号をクリックします。

  [Bound Transports] の横のプラス記号をクリックすると、ポート番号が表示されます。

Linux の場合

次のコマンドを使用して、ポートを表示できます。

ndsconfig get | grep http

http.server.interface、その次にポート番号という表現で記述された行を探します。

ログの記録の使用

Logger は、DSBackup、DSMerge、DSRepair など、すべての OpenText eDirectory 管理ツール (eMTool) に関するすべてのイベントを記録するインフラストラクチャモジュールです。このリリースで提供されているログファイルは 1 つだけです。このログファイルに、すべての eMTool の操作が記録されます。

ログの記録は、クライアントを実行するときに指定するログファイルを通して提供されるクライアントログ記録サービスとは異なります。このログファイルとは、たとえば、クライアントコマンドで -l mylogfile.txt を指定するときや、OpenText Identity Console でログファイル名として「mylogfile.txt」と入力するときに指定されるもののことです。現行のログ記録では、eMTool によって実行されるタスクに関するすべてのサーバメッセージが記録されるので、記録されるログ情報は、クライアントログ記録サービスのものより詳細です。これに対して、クライアントログサービスではクライアントメッセージおよびクライアントに送信されたメッセージが記録され、進捗状況の概要がレポートされます。

ログの記録は非同期で実行され、デフォルトではすべての操作が記録されます。

このリリースのログの記録には、次の機能があります。

- ログファイルの名前と場所を変更できます。

  デフォルトでは、ログファイルは OpenText eDirectory のインストール先と同じディレクトリにある \log ディレクトリに作成されます。
- 最大ファイルサイズが変更できます。変更後、ログファイルはリセットされます。

  最大ファイルサイズは 8MB です。
- ログモードを変更できます。

  すべての新しいメッセージをログファイルに追加するか、または既存のログファイルを上書きするかを選択できます。デフォルトでは、追加のオプションが指定されています。
- ログの記録を開始または停止できます。

  デフォルトでは、が起動するとログの記録は開始モードになります。停止モードの間は、メッセージは記録されません。
- ログファイルの内容をリセットできます。
- クライアントコンピュータからログファイルを読み込むことができます。

詳細については、次を参照してください。

- 629 ページの 「ログの記録コマンドラインクライアントを使用する」

ログの記録コマンドラインクライアントを使用する

次の表に、ログの記録コマンドラインクライアントのオプションを示します。

オプション	説明
logstart	ログの記録を開始します。
logstop	ログの記録を停止します。
readlog	現在のログファイルを表示します。
getlogstate	現在のログの記録の状態 (開始または停止) を表示します。
getloginfo	ログファイルの名前、ログ記録モード (追加または上書き)、最大サイズと現在のサイズを表示します。
setloginfo [-f *filename*] [-s *size in Kilo bytes*] [-a | -o]	次のパラメータを使用して、ログファイルの名前、サイズ、ログモード (追加または上書き) を設定します。 ◆ -f *filename* ログファイルの名前。 ◆ -s *size in KB* ログファイルの最大サイズを設定します。 ◆ -a 新しいログメッセージを現在のログメッセージに追加します。 ◆ -o ログファイルを上書きします。
emptylog	サーバログファイルの内容をクリアします。

eMBox クライアントを使ったバックアップ / 復元作業

eMBox クライアントはコマンドライン Java クライアントで、これを使用すると OpenText eDirectory Backup eMTool などの eMBox ツールにアクセスできます。複数サーバ環境でも、ファイアウォール越しのアクセスができれば、1 台のコンピュータからバックアップ、復元、ロールフォワードログの設定ができます。高度な処理は、コマンドラインから実行する Java クライアントである、eMBox Client を使い、リモート操作で実行できます。ファイアウォール越しにでも、あるいは VPN 経由でも操作できます。

OpenText eDirectory バックアップツールは、eMBox ツールセットの一部です。eMBox は、OpenText eDirectory の一部としてサーバにインストールされるサービスです。

バックアップツールは次のファイルで構成されます。

ファイル名	説明
backupcr	バックアップ / 復元機能をすべて含むコアライブラリ。 このライブラリにはユーザインタフェースはなく、backuptl プログラムから動的にロード、リンクされる形で動作します。
backuptl	backupcr ライブラリへのツールインタフェース。DSBK アーキテクチャを介してバックアップ / 復元機能を提供するプログラムです。 これには、OpenText Identity Console、DSBK、Java コマンドラインクライアントを介してアクセスできます。
dsbackup_en.xlf	バックアップツールから返されるメッセージを含む言語ファイルです。

重要 : 復元後の検証処理については、8.5 より前の OpenText eDirectory とは互換性がありません。レプリカリングに属するサーバで新しいバックアップ / 復元ツールを使う場合は、これに属するサーバすべてを OpenText eDirectory 8.5 以降にアップグレードする必要があります。

eMBox クライアントはバッチモードで実行できるため、OpenText eDirectory Backup eMTool を使用して無人バックアップを行うことができます。

eMBoxClient.jar ファイルは、OpenText eDirectory の一部としてサーバにインストールされます。それ以外にも、Sun JVM 1.3.1 が動作する環境であれば、eMBoxClient.jar をコピーして動かすことができます。詳細については、617 ページの 「OpenText eDirectory 管理ツールボックス」および 620 ページの 「ワークステーションでクライアントを実行する」を参照してください。

OpenText eDirectory のバックアップ / 復元作業に先立ち、461 ページの 「OpenText eDirectory のバックアップ処理に関する確認事項」を参照して問題点を確認し、効率的に作業できるようにしてください。

- 630 ページの 「前提条件」
- 631 ページの 「eMBox クライアントによる手動バックアップ」
- 633 ページの 「バッチファイルと eMBox クライアントによる無人バックアップ」
- 634 ページの 「eMBox クライアントによるロールフォワードログの設定」
- 636 ページの 「eMBox クライアントによるバックアップファイルの復元作業」

前提条件

- ❒ バックアップ処理を起動するマシンに、ファイル eMBoxClient.jar があることを確認してください。

このファイルは、OpenText eDirectory インストールの一部としてサーバにインストールされます。それ以外にも、Sun JVM 1.3.1 が動作する環境であれば、eMBoxClient.jar をコピーして実行することができます。複数サーバ環境でも、ファイアウォール越しのアクセスが可能であれば、1 台のコンピュータからバックアップを実行できます。詳細については、618 ページの 「コマンドラインクライアントの使用」を参照してください。

- ロールフォワードログを作成するのであれば、バックアップを行う前にこの機能を有効にしてください。

  レプリカリングに属するサーバは、ロールフォワードログ機能を有効にしておく必要があります。バックアップファイルがあっても、ロールフォワードログがなければ復元後の検証処理に失敗し、データベースを開けないことになります。

  ロールフォワードログの詳細については、「475 ページの 「ロールフォワードログを使用する」」を参照してください。また、この機能を有効にする手順については、634 ページの 「eMBox クライアントによるロールフォワードログの設定」を参照してください。
- コマンドラインオプションについては、「491 ページの 「バックアップ / 復元のコマンドラインオプション」」を参照してください。
- 複数サーバ環境のツリーの場合、このサーバとレプリカを共有するサーバすべてについて、OpenText eDirectory 8.5 以降にアップグレードする必要があります。

eMBox クライアントによる手動バックアップ

eMBox クライアントを使うと、OpenText eDirectory データベースの中身を、バックアップが実行されているサーバの指定したファイルにバックアップできます。バックアップファイルには、OpenText eDirectory をバックアップ時点の状態に復元するために必要な情報がすべて含まれています。バックアップの処理結果は所定のログファイルに記録されます。

eMBox クライアントを使って OpenText eDirectory データベースをバックアップする手順を次に示します。

1. eMBox クライアントを対話式モードで起動します。
   - Linux: コマンドラインで、「edirutil -i」と入力します。
   - Windows: *drive*\novell\nds\edirutil.exe -i を実行します。

   edirutil ファイルは、eMBox クライアントを実行するためのショートカットです。Java 実行可能ファイルと、eDirectory で eMBox クライアントがインストールされるデフォルトの場所を指します。この情報は、620 ページの 「クライアント用にパスおよびクラスパスをセットアップする」に記載されているようにして手動で入力することもできます。

   eMBox Client が開くと、「eMBox Client>」という eMBox Client プロンプトが現れます。
2. バックアップの対象サーバにログインします。次のように入力してください。

   login -s *server_name_or_IP_address* -p *port_number* -u *username.context* -w *password*

   たとえば、Windows では、次のように入力します。

   login -s 151.155.111.1 -p 8009 -u admin.mycompany -w mypassword

セキュア接続が確立できないというエラーが表示される場合は、「627 ページの 「クライアントを使用してセキュア接続を確立する」」に表示されている JSSE ファイルがシステム上にない可能性があります。

指定するポート番号が分からない場合は、627 ページの 「OpenText eDirectory のポート番号を確認する」を参照してください。

eMBox クライアントはログインが成功したかどうかを表示します。

3 eMBox クライアントのプロンプトが出たら、次のような形式でバックアップコマンドを入力します。

backup -b -f *backup_filename_and_path* -l *backup_log_filename_and_path* -u *include_file_filename_and_path* -t -w -a

各スイッチの間にはスペースが必要です。スイッチの順序は重要ではありません。

たとえば、Windows では、次のように入力します。

backup -b -f c:\backups\8_20_2001.bak -l c:\backups\backup.log -u c:\backups\myincludefile.txt -t -w -a

この例では、フルバックアップを取ること (-b)、バックアップファイルを c:\backups\8_20_2001.bak とすること、処理結果を c:\backups\backup.log に出力すること、さらに、データベースとともに、次に示す他のファイルもバックアップすることを指定しています。

- 管理者があらかじめ作成したインクルードファイル (-u c:\backups\myincludefile.txt) に列挙されたファイル。
- ストリームファイル (-t)

  さらにこの例ではバックアップファイルの上書き (-w) が指定されているため、同じ名前のバックアップファイルがあれば Backup eMTool により上書きされます。
- -a オプションは、ホット連続バックアップ中に、ロールフォワードログディレクトリから古いログファイルを削除します。

eMBox クライアントはバックアップが成功したかどうかを表示します。

4 サーバからログアウトするには、次のコマンドを入力します。

logout

5 eMBox クライアントを終了するには、次のコマンドを入力します。

exit

6 OpenText eDirectory のバックアップ処理が終了したら、すぐにファイルシステムのバックアップ作業を行い、テープに保存します。Backup eMTool による処理では、サーバ上にバックアップファイルができるだけです。

手動バックアップの詳細については、487 ページの 「DSBK による手動バックアップ」を参照してください。

バッチファイルと eMBox クライアントによる無人バックアップ

バッチファイルを使用して、eMBox クライアントによる OpenText eDirectory の無人バックアップを実行します。たとえば、OpenText eDirectory のフルバックアップを週 1 回取り、インクリメンタルバックアップを毎晩取る、といった運用が可能です。

バッチモードで eMBox クライアントを実行するには、システムバッチファイルを使う、eMBox クライアントの内蔵バッチファイルを使う、両者を組み合わせて使う、という方法があります。詳細については、624 ページの 「コマンドラインクライアントをバッチモードで実行する」を参照してください。

ここではシステムバッチファイルを使う方法を解説します。

1 サーバをバックアップするためのシステムバッチファイルを作成します。次のような書式で、1 行に 1 サーバ分のコマンドを記述してください。

   Windows 環境と Linux 環境で一般的なパターンは次のとおりです。

   ```
   java -cp path/eMBoxClient.jar embox -s server_name -p port_number -u
   username.context -w password -t backup.backup -b -f
   backup_filename_and_path -l backup_log_filename_and_path -u
   include_file_filename_and_path -t -w
   ```

   具体例とその解説については、「633 ページの 「無人バックアップ用システムバッチファイルの例」」を参照してください。

   毎晩実行するインクリメンタルバックアップにも、フルバックアップに使うのと同じバッチファイルが使えますが、-b スイッチを -i に変更して、フルバックアップではなくインクリメンタルバックアップを実行します。フルバックアップとインクリメンタルバックアップで、保存先バックアップファイル名を異なるものにしておく方がよいでしょう。

   指定するポート番号が分からない場合は、「627 ページの 「OpenText eDirectory のポート番号を確認する」」を参照してください。セキュア接続を使用する場合は、「627 ページの 「クライアントを使用してセキュア接続を確立する」」を参照してください。

   eMBox クライアントの内蔵バッチファイルの使い方については、「624 ページの 「コマンドラインクライアントをバッチモードで実行する」」を参照してください。

2 このバッチファイルを定期的に起動するよう設定します。具体的な設定方法については、オペレーティングシステムまたはサードパーティ製ソフトウェアの資料を参照してください。

3 OpenText eDirectory のバックアップ後すぐに、ファイルシステムのバックアップ作業を行い、OpenText eDirectory バックアップファイルをテープに保存できるよう準備してください。

   Backup eMTool による処理では、サーバ上にバックアップファイルができるだけです。

4 バックアップが正常に実行されているか、ログファイルで定期的に確認してください。

無人バックアップ用システムバッチファイルの例

システムバッチファイルの例を以下に示します。

Windows 用のバッチファイル例

```
java -cp c:\novell\nds\embox\eMBoxClient.jar embox -s myserver -p 8008 -u
admin.myorg -w mypassword -n -t backup.backup -b -f c:\backup\backup.bak -
u c:\backup\includes\includefile.txt -l c:\backup\backup.log -t -w
```

この例には次のようなオプションが指定されています。

- フルバックアップを取る指定 (-b)。
- インクルードファイルの指定 (-u)。これはオプションです。データベース以外にもファイルをバックアップしたい場合に使います。インクルードファイルはあらかじめ用意しておいてください。
- ストリームファイル (-t) もバックアップされます。
- 同じ名前のバックアップファイルがあれば上書きする指定 (-w)。

  重要 : 同じバッチファイルを繰り返し使用するなど、同じ名前のバックアップファイルが存在する場合、バックアップが正常に行われるように、-w オプションを使用して、既存のバックアップファイルが上書きされるようにしてください。

  バッチモードでは、同じ名前のファイルが存在する場合に -w が指定されていないと、デフォルトの動作として、ファイルは上書きされず、バックアップは作成されません。対話式モードでは、-w が指定されていないと、ファイルを上書きしてよいかどうかを尋ねられます。

  OpenText eDirectory のフル / インクリメンタルバックアップの都度、すぐにファイルシステムのバックアップを取っているのであれば、前回のバックアップファイルはテープに保存されているはずです。したがって上書きしても問題ありません。
- この例では非セキュアポートが使用されているため (-p 8008)、非セキュア接続が指定されています (-n)。

eMBox クライアントによるロールフォワードログの設定

eMBox クライアントを使って、ロールフォワードログに関する設定を変更することができます。次のような設定ができます。

- 現在の設定の確認
- ロールフォワードログ機能の有効 / 無効の切り替え

  レプリカリングに属するサーバは、ロールフォワードログ機能を有効にしておく必要があります。バックアップファイルがあっても、ロールフォワードログがなければ復元後の検証処理に失敗し、データベースを開けないことになります。
- ロールフォワードログの保存先ディレクトリの変更
- ロールフォワードログのファイルサイズの最小値、最大値の設定
- 現在使用中のログ、既に書き出しを終えた最新のログの判別
- ストリームファイルをロールフォワードログに含めるかどうかの切り替え

ロールフォワードログの詳細については、「475 ページの 「ロールフォワードログを使用する」」を参照してください。

1 eMBox クライアントを対話式モードで起動します。

- Linux: コマンドラインで、「edirutil -i」と入力します。
- Windows: *drive*\novell\nds\edirutil.exe -i を実行します。

edirutil ファイルは、eMBox クライアントを実行するためのショートカットです。これは、Java 実行可能ファイルと、eMBox クライアントが OpenText eDirectory と一緒にインストールされているデフォルトの場所を指しています。必要な -ns オプションがこれには含まれています。これらのオプションは、620 ページの 「ワークステーションでクライアントを実行する」に記載されているようにして手動で入力することもできます。

正常に起動されると、「eMBox Client>」というプロンプトが現れます。

2 ロールフォワードログの設定を行うサーバにログインします。次のように入力してください。

login -s *server_name_or_IP_address* -p *port_number* -u *username.context* -w *password*

たとえば、Windows では、次のように入力します。

```
login -s 151.155.111.1 -p 8009 -u admin.mycompany -w mypassword
```

セキュア接続が確立できないというエラーが表示される場合は、「627 ページの 「クライアントを使用してセキュア接続を確立する」」に表示されている JSSE ファイルがシステム上にない可能性があります。

指定するポート番号が分からない場合は、627 ページの 「OpenText eDirectory のポート番号を確認する」を参照してください。

eMBox クライアントはログインが成功したかどうかを表示します。

3 (オプション) 次のコマンドを入力して、現在の設定を検索します。

getconfig

オプション指定は必要ありません。

たとえば次のように表示されます。

```
Roll forward log status OFF
Stream file logging status OFF
Current roll forward log directory C:\rfl\nds.rfl
Minimum roll forward log size (bytes) 104857600
Maximum roll forward log size (bytes) 4294705152
Last roll forward log not used 00000000.log
Current roll forward log 00000001.log
*** END ***
```

4 setconfig コマンドで設定を変更します。次のような形式で入力してください。

setconfig [-L|-l] [-T|-t] -r *path_to_roll-forward_logs* -n *minimum_file_size* -m *maximum_file_size*

各スイッチの間にはスペースが必要です。スイッチの順序は重要ではありません。

ロールフォワードログ専用のディスクパーティション / ボリュームを用意するのが最善です。こうしておけば、ディスク容量やアクセス権を監視しやすくなります。

警告 : ロールフォワードログ記録を有効にしたら、デフォルトの保存先は使用しないでください。障害対策上、OpenText eDirectory とは別のディスクパーティション / ボリューム、別の記憶デバイスを指定してください。ロールフォワードログディレクトリは、バックアップ環境設定を変更するサーバ上である必要があります。

重要 : ロールフォワードログ機能を有効にする場合、ログを保存するボリュームのディスク容量を常に監視してください。これを怠ると、ログの容量は増える一方なので、ディスクパーティション / ボリュームがあふれてしまう恐れがあります。ディスク容量が不足してロールフォワードログを作成できない場合は、OpenText eDirectory はそのサーバに対して応答しなくなります。書き出しが終わったロールフォワードログは、定期的にバックアップし、サーバから削除するようお勧めします。詳細については、479 ページの 「ロールフォワードログのバックアップと削除」を参照してください。

5 サーバからログアウトするには、次のコマンドを入力します。

```
logout
```

6 eMBox クライアントを終了するには、次のコマンドを入力します。

```
exit
```

eMBox クライアントによるバックアップファイルの復元作業

eMBox クライアントを使ってバックアップファイルに保存されたデータから OpenText eDirectory データベースを復元する手順を解説します。手動あるいはバッチファイルで残しておいたバックアップファイルから、データを復元できます。復元の処理結果は所定のログファイルに記録されます。

eMBox クライアントを使えば、OpenText Identity Console では使用できない高度な復元オプションも使用できます。これらは、「491 ページの 「バックアップ / 復元のコマンドラインオプション」」の restore および restadv の項に記載されています。

eMBox クライアントを使って OpenText eDirectory データベースを復元する手順を示します。

1 必要なバックアップファイルを集めておきます。詳しくは「480 ページの 「復元処理の準備」」を参照してください。

2 eMBox クライアントを対話式モードで起動します。

- Linux: コマンドラインで、「edirutil -i」と入力します。
- Windows: *drive*\novell\nds\edirutil.exe -i を実行します。

edirutil ファイルは、eMBox クライアントを実行するためのショートカットです。Java 実行可能ファイルと、OpenText eDirectory で eMBox クライアントがインストールされるデフォルトの場所が指定されているほか、必要な -ns オプションも含まれています。この情報は、620 ページの 「ワークステーションでクライアントを実行する」に記載されているようにして手動で入力することもできます。

正常に起動されると、「eMBox Client>」というプロンプトが現れます。

3 復元の対象サーバにログインします。次のように入力してください。

login -s *server_name_or_IP_address* -p *port_number* -u *username.context* -w *password*

たとえば、Windows では、次のように入力します。

```
login -s 151.155.111.1 -p 8009 -u admin.mycompany -w mypassword
```

セキュア接続が確立できないというエラーが表示される場合は、「627 ページの 「クライアントを使用してセキュア接続を確立する」」に表示されている JSSE ファイルがシステム上にない可能性があります。

指定するポート番号が分からない場合は、627 ページの 「OpenText eDirectory のポート番号を確認する」を参照してください。

eMBox クライアントはログインが成功したかどうかを表示します。

4 eMBox クライアントのプロンプトで、次のような一般的な形式で restore コマンドを入力します。

restore -r -a -o -f *full_backup_path_and_filename* -d *roll-forward_log_location* -l *restore_log_path_and_filename*

各スイッチの間にはスペースが必要です。スイッチの順序は重要ではありません。-r スイッチを使用して、OpenText eDirectory データベース自体を復元してください。そうしないと、他の種類のファイルだけが復元されます。復元処理の終了後にデータベースをアクティブにして開くには、-a および -o を指定してください。

ロールフォワードログを使って復元する場合は、ログのフルパスを指定しなければなりません。フルパスには、OpenText eDirectory により自動的に作成されたディレクトリ (通常は \nds.rfl) も含まれます。このディレクトリの詳細については、477 ページの「ロールフォワードログの保存先」を参照してください。

次に例を示します。

```
restore -r -a -o -f sys:/backup/nds.bak -d $HOME/rfldir/nds.rfl -l $HOME/backups/backup.log
```

この例では、データベースそのものを復元し (-r)、復元の検証が正常終了してから、そのデータベースをアクティブにして (-a)、開く (-o) よう指定しています。-f スイッチでフルバックアップファイルの場所を、-d でロールフォワードログの場所を指定しています。また、復元処理の結果を記録するログファイルを、-l で指定しています。

これによりフルバックアップファイルからの復元処理が実行され、次にインクリメンタルバックアップファイルの指定を求めるプロンプトが現れます。

5 (状況によって実行) インクリメンタルバックアップファイルから復元する場合は、プロンプトに応じて順次、そのパスとファイル名を入力します。

プロンプトには次に指定するべきファイルの ID が表示されます。これはインクリメンタルバックアップファイルのヘッダに記載されているものです。

バックアップ処理が正常終了すれば、その旨の表示が現れます。

6 (状況によって実行) 復元処理に失敗した場合は、ログファイルでエラーの原因を確認してください。

復元後の検証に失敗した場合の対処については、「503 ページの 「復元後の検証処理に失敗した場合の対処方法」」を参照してください。

注 : レプリカリング中に OpenText eDirectory 8.5 より前のバージョンが稼動しているサーバがある場合、復元処理は失敗します。エラーコードは -666、すなわち「DS バージョンの不整合」となります。

7 サーバからログアウトするには、次のコマンドを入力します。

```
logout
```

8 eMBox クライアントを終了するには、次のコマンドを入力します。

```
exit
```

9 (状況によって実行)NICI セキュリティファイルを復元した場合は、復元の完了後、サーバを再起動して NICI を再初期化し、その後で DIB を復元します。

10 ここでサーバが通常どおり要求に応答することを確認しておきます。

11 (状況によって実行) このサーバでロールフォワードログ機能を使うためには、改めて有効に切り替え、障害対策のための書き出し先も設定し直して、ロールフォワードログの環境設定を再作成する必要があります。ロールフォワードログを有効にしてから、改めてフルバックアップも取る必要があります。

この手順が必要となるのは、復元処理の過程で、ロールフォワードログに関する設定はデフォルトに戻るためです。つまり、ロールフォワードログ機能は無効となり、保存先もデフォルトの場所になるからです。フルバックアップが改めて必要となるのは、スケジュールに従って次に無人でのフルバックアップが取られるまでに、再び障害が起こる可能性があるためです。

ロールフォワードログの詳細については、「475 ページの 「ロールフォワードログを使用する」」を参照してください。

以上で復元作業が終了しました。NICI の再初期化も済んでいるので、暗号化された情報にもアクセスできます。ロールフォワードログ機能を使用する場合は、今後の障害に備えるため、再びこの機能を有効にし、フルバックアップを取っておいてください。

23 OpenText eDirectory イベントの監査

以下のいずれかの方法で、OpenText eDirectory イベントを監査することができます。

- 639 ページの 「CEF による監査」
- 662 ページの 「ジャーナルイベントのキャッシング」
- 664 ページの 「LDAP 監査」

CEF による監査

共通イベント形式 (CEF) は、分散システムレベルで複数のコンポーネントからの監査情報をマージしたり分析したりする操作を容易にする標準のイベント形式です。CEF 形式では、転送メカニズムとして Syslog メッセージ形式が使用されます。

CEF は、テキストベースの拡張可能な形式で、オンプレミスのデバイスやクラウドベースのサービスなど、複数のデバイスタイプをサポートするよう設計されています。CEF イベントは、異なるアプリケーションの監査記録を理解しやすくするのに役立ちます。

CEF を使用するように OpenText eDirectory を設定すると、次のようなメリットがあります。

- セキュリティで保護された同一の監査サービスを分散システムに提供します。
- CEF では、ログ管理を簡素化する標準メッセージ形式を使用します。
- 新しいイベントの形式は、Sentinel とシームレスに統合します。

この次のセクションでは、OpenText eDirectory で CEF を設定する方法について説明します。

- 639 ページの 「CEF の設定」

CEF の設定

OpenText eDirectory インストールキットには、ダウンロードパッケージの一部として Linux 版と Windows 版の両方の CEF クライアントが付属しています。OpenText eDirectory のインストールプログラムは、オペレーティングシステム上に CEF パッケージをインストールします。CEF パッケージには、次のファイルが含まれています。

- Linux
  - novell-edirectory-xdaslog

- novell-edirectory-xdaslog-conf
- novell-edirectory-cefinstrument-9.2.0-0.x86_64.rpm

- Windows
  - cefauditds.dlm
  - xdaslog.dll

このセクションでは、次のトピックについて説明します。

- 640 ページの 「システム要件」
- 640 ページの 「CEF 用の OpenText Identity Console のインストール」
- 641 ページの 「CEF プロパティファイルの設定」
- 648 ページの 「監査用の CEF の設定」
- 650 ページの 「モジュールのロードとアンロード」
- 651 ページの 「CEF イベントキャッシングの有効化」
- 653 ページの 「CEF イベントタイプについて」
- 655 ページの 「CEF イベントのコレクタの使用」
- 655 ページの 「CEF 監査イベントのフィルタ処理について」
- 657 ページの 「CEF 実装スキーマ」
- 662 ページの 「CEF イベント」

システム要件

Audit OpenText Identity Console アプリケーションをインストールして使用するには、OpenText Identity Console 1.5 以上が必要です。要件やダウンロード手順については、「OpenText Identity Console のインストールの計画 」を参照してください。

CEF 用の OpenText Identity Console のインストール

OpenText Identity Console アプリケーションをインストールするには :

1 次の URL を使用して、Web ブラウザから OpenText Identity Console を開きます。

`https://ip_address_or_DNS:Port/identityconsole/`

ここで、*ip_address_or_DNS* は、OpenText Identity Console サーバの IP アドレスまたは DNS 名およびポート番号です。

例 :

`http://111.111.1.1:9000/identityconsole.html`

2 ツリーユーザ名とパスワードを使用して、OpenText Identity Console にログインします。

OpenText Identity Console のすべての機能を利用するには、管理者としてツリーにログインします。管理者ユーザだけが、すべての機能への完全アクセス権を持っています。管理者ユーザ以外は、権利が割り当てられている役割にのみアクセスできます。

詳細については、『OpenText Identity Console 管理ガイド (https://www.netiq.com/documentation/identity-console/identity_console-admin/data/bookinfo.html)』を参照してください。

CEF プロパティファイルの設定

OpenText eDirectory メディアでは、サンプルプロパティファイルである auditlogconfig.properties.template ファイルが configdir (n4u.server.configdir) ディレクトリに格納されています。

次の表に、Linux オペレーティングシステムと Windows オペレーティングシステム上の auditlogconfig.properties ファイルのデフォルトの場所を示します。

表 23-1 *CEF 環境設定ファイル*

オペレーティングシステム	プロパティファイルの場所
Linux	`/etc/opt/novell/eDirectory/conf/auditlogconfig.properties` 非ルートインストールでは、CEF プロパティファイルが conf ディレクトリに配置されます。
Windows	`<Install Path>/novell/nds/auditlogconfig.properties` 通常、このプロパティファイルは OpenText eDirectory のインストールディレクトリにあります。

プロパティファイルを設定してから、環境を OpenText eDirectory 9.2 から最新バージョンにアップグレードする場合、インストーラはプロパティファイルを置き換えません。代わりに、アップグレードプロセスがこのファイル (auditlogconfig.properties) を更新してカスタマイズ内容が保持されるようにします。

OpenText Identity Console のインストール後に、CEF を設定できます。CEF 環境設定は、単純なテキストベースの auditlogconfig.properties 環境設定ファイルに保存されます。実際の要件に応じてこのファイルをカスタマイズできます。

CEF auditlogconfig.properties ファイルには次の情報が含まれています。

Linux

```
# Set the level of the root logger to DEBUG and attach appenders.
#log4j.rootLogger=info, S, R

# Defines appender S to be a SyslogAppender.
#log4j.appender.S=org.apache.log4j.net.SyslogAppender
```

```
# Defines location of Syslog server.
#log4j.appender.S.Host=localhost
#log4j.appender.S.Port=port

# Specify protocol to be used (UDP/TCP/SSL)
#log4j.appender.S.Protocol=TCP

# Specify SSL certificate file for SSL connection.
# File path should be given with double backslash.
#log4j.appender.S.SSLCertFile=/etc/opt/novell/mycert.pem

# Minimum log-level allowed in syslog.
#log4j.appender.S.Threshold=INFO

# Defines the type of facility.
#log4j.appender.S.Facility=USER

# Defines caching for SyslogAppender.
# Inputs should be yes/no
#log4j.appender.S.CacheEnabled=no

# Cache location directory
# Directory should be available for creating cache files
#log4j.appender.S.CacheDir=/var/opt/novell/eDirectory

# Cache File Size
# Cache File size should be in the range of 50MB to 4000MB in limited
growth mode
# Cache File size should be set as 0MB to enable infinite growth of cache
file
#log4j.appender.S.CacheMaxFileSize=500MB

# Layout definition for appender Syslog S.
#log4j.appender.S.layout=org.apache.log4j.PatternLayout
#log4j.appender.S.layout.ConversionPattern=%c: %m%n

# Defines appender R to be a Rolling File Appender.
#log4j.appender.R=org.apache.log4j.RollingFileAppender

# Log file for appender R.
#log4j.appender.R.File=/var/opt/novell/eDirectory/log/cef-events.log

# Max size of log file for appender R.
#log4j.appender.R.MaxFileSize=100MB

# Set the maximum number of backup files to keep for appender R.
# Max can be 13. If set to zero, then there will be no backup files.
#log4j.appender.R.MaxBackupIndex=10

# Layout definition for appender Rolling log file R.
#log4j.appender.R.layout=org.apache.log4j.PatternLayout
#log4j.appender.R.layout.ConversionPattern=%d{MMM dd HH:mm:ss} %c %m%n
```

Windows

```
# Brief description for appenders and their options are provided.
# For detailed decriptions refer to log4cxx documentation.

# Set the level of the root logger to DEBUG and attach appenders.
#log4j.rootLogger=info, S, R

# Defines appender S to be a SyslogAppender.
#log4j.appender.S=org.apache.log4j.net.SyslogAppender

# Defines location of Syslog server.
#log4j.appender.S.Host=localhost
#log4j.appender.S.Port=port

# Specify protocol to be used (UDP/TCP/SSL).
#log4j.appender.S.Protocol=SSL

# Specify SSL certificate file for SSL connection.
# File path should be given with double backslash.
#log4j.appender.S.SSLCertFile=C:\\Novell\\mycert.pem

# Minimum log-level allowed in syslog.
#log4j.appender.S.Threshold=INFO

# Defines the type of facility.
#log4j.appender.S.Facility=USER

# Defines caching for SyslogAppender.
# Inputs should be yes/no
#log4j.appender.S.CacheEnabled=yes

# Cache location directory
# Directory should be available for creating cache files
#log4j.appender.S.CacheDir=C:\\NetIQ\\eDirectory

# Cache File Size
# Cache File size should be in the range of 50MB to 4000MB in limited
growth mode
# Cache File size should be set as 0MB to enable infinite growth of cache
file
#log4j.appender.S.CacheMaxFileSize=500MB

# Layout definition for appender Syslog S.
#log4j.appender.S.layout=org.apache.log4j.PatternLayout
#log4j.appender.S.layout.ConversionPattern=%c: %m%n

# Defines appender R to be a Rolling File Appender.
#log4j.appender.R=org.apache.log4j.RollingFileAppender
```

```
# Log file for appender R.
# File path should be given with double backslash.
#log4j.appender.R.File=C:\\cef-events.log

# Max size of log file for appender R.
#log4j.appender.R.MaxFileSize=100MB

# Set the maximum number of backup files to keep for appender R.
# Max can be 13. If set to zero, then there will be no backup files.
#log4j.appender.R.MaxBackupIndex=10

# Layout definition for appender Rolling log file R.
#log4j.appender.R.layout=org.apache.log4j.PatternLayout
#log4j.appender.R.layout.ConversionPattern=%d{MMM dd HH:mm:ss} %c %m%n
```

auditlogconfig.properties ファイルの内容を検討する前に、以下の考慮事項を確認することをお勧めします。

- 文字の S と R は、それぞれ Syslog アペンダとローリングファイルアペンダを意味します。
- エントリは大文字と小文字が区別されません。
- エントリは任意の順序で表示できます。
- ファイル内の空白行は有効です。
- ハッシュ (#) で始まるすべての行はコメントアウトされた行です。

次の表に、auditlogconfig.properties ファイル内設定に関する情報を示します。

重要 : 環境設定の変更後は、OpenText eDirectory を再起動する必要があります。

設定	説明
log4j.rootLogger=**info, S, R**	ルートロガーのレベルをデバッグに設定して、S または R という名前のアペンダをアタッチします。S は Syslog アペンダを、R はローリングファイルアペンダを意味します。ルートロガーの使用可能なレベルは、優先度順に、**trace**、**debug**、**info**、**warn**、**error**、および **fatal** です。デフォルトでは **info** が指定され、これには **trace** と **debug** を除く他のすべてのレベルのイベントが含まれます。すべてのイベントをログに送信する場合は、このフィールドに［**TRACE**］または［**ALL**］を指定します。
log4j.appender.S=**org.apache.log4j.net.SyslogAppender**	アペンダ S を Syslog アペンダとして指定します。
log4j.appender.S.Host=**localhost**	CEF イベントの記録先の Syslog サーバの場所を指定します。 IFor example,log4j.appender.S.Host=192.168.0.1

設定	説明
log4j.appender.S.Port=**port**	CEF が Syslog サーバに接続するポートです。 このポートは 1 ～ 65535 の値をサポートします。無効な値が指定された場合は、ポートがデフォルトで 514 に設定します。 CEF と Syslog サーバ間の接続が失われた場合は、接続が回復するまで Identity Manager はイベントを記録できません。
log4j.appender.S.Protocol=TCP	使用するプロトコルを指定します。たとえば、UDP、TCP、または SSL などです。
log4j.appender.S.SSLCertFile=**/etc/opt/novell/mycert.pem**	SSL 接続用の SSL 証明書ファイルを指定します。ファイルのパスを指定する場合は二重の円記号を使用します。この設定の指定は任意です。
log4j.appender.S.Threshold=**INFO**	Syslog で許可される最小ログレベルを指定します。サポートされているログレベルは、優先度順に、TRACE、DEBUG、INFO、WARN、および ERROR です。デフォルトの最小ログレベルは INFO で、これには、TRACE と DEBUG を除く他のすべてのログレベルが含まれます。つまり、最小ログレベルとして INFO を指定すると、TRACE イベントと DEBUG イベントは Syslog サーバに送信されません。すべてのイベントを Syslog サーバに送信する場合は、このフィールドに［TRACE］または［ALL］を指定します。
log4j.appender.S.Facility=**USER**	ファシリティのタイプを指定します。ファシリティは、メッセージを分類するために使用されます。現在、USER ファシリティがサポートされています。これらの値は大文字または小文字で指定することができます。
log4j.appender.S.layout=**org.apache.log4j.PatternLayout**	Syslog アペンダ用のレイアウト設定です。
log4j.appender.S.layout.ConversionPattern**=%c : %p%m%n**	Syslog アペンダ用のレイアウト設定です。変換パターンとその説明については、「logging.apache.org」を参照してください。
log4j.appender.R=**org.apache.log4j.RollingFileAppender**	アペンダ R をローリングファイルアペンダとして指定します。
log4j.appender.R.File=**/var/opt/novell/eDirectory/log/cef-events.log**	ローリングファイルアペンダ用のログファイルの場所です。
log4j.appender.R.MaxFileSize=**100MB**	ローリングファイルアペンダ用のログファイルの最大サイズ (MB 単位) です。この値はクライアントで許可される最大サイズに設定します。

設定	説明
log4j.appender.R.MaxBackupIndex=10	ローリングファイルアペンダ用のバックアップファイルの最大数を指定します。バックアップファイルの最大数は 10 にすることができます。0 の値はバックアップファイルなしを意味します。
log4j.appender.R.layout=org.apache.log4j.PatternLayout	ローリングファイルアペンダ用のレイアウト設定です。
log4j.appender.R.layout.ConversionPattern=%d{MMM dd HH:mm:ss} %c : %p%m%n	ローリングファイルアペンダ用のレイアウト設定です。 変換パターンとその説明については、logging.apache.org を参照してください。

次の表に、米国で解釈される日付と時刻のパターンの例を示します。指定されている日付と時刻は、米国太平洋タイムゾーンの 2012 年 7 月 4 日 12 時 8 分 56 秒です。

***表 23-2** 日付と時刻のパターンの例*

日付と時刻のパターン	結果
"yyyy.MM.dd G 'at' HH:mm:ss z"	2012.07.04 AD at 12:08:56 PDT
"EEE, MMM d, ''yy"	Wed, Jul 4, '01
"h:mm a"	12:08 PM
"hh 'o''clock' a, zzzz"	12 o'clock PM, Pacific Daylight Time
"K:mm a, z"	0:08 PM, PDT
"yyyyy.MMMMM.dd GGG hh:mm aaa"	02012.July.24 AD 12:08 PM
"EEE, d MMM yyyy HH:mm:ss Z"	Wed, 24 Jul 2012 12:08:56 -0700
"yyMMddHHmmssZ"	120724120856-0700
"yyyy-MM-dd'T'HH:mm:ss.SSSZ"	2012-07-04T12:08:56.235-0700

Syslog アペンダの有効化

Syslog アペンダを使用して、リアルタイムイベントを表示できます。加えて、Syslog サーバは災害発生時に優れたバックアップサポートを提供します。

Syslog アペンダを有効にするには、auditlogconfig.properties ファイルに次の変更を加えます。

1. 次のエントリを S に変更して、Syslog アペンダをアタッチします。

   log4j.rootLogger=info, S

2. 次のエントリをコメントアウトします。

   ```
   log4j.appender.S=org.apache.log4j.net.SyslogAppender
   log4j.appender.S.Host=localhost
   ```

```
log4j.appender.S.Port=port

log4j.appender.S.Protocol=SSL

log4j.appender.S.SSLCertFile=/etc/opt/novell/mycert.pem

#log4j.appender.S.Threshold=INFO

#log4j.appender.S.Facility=USER

#log4j.appender.S.layout=org.apache.log4j.PatternLayout

#log4j.appender.S.layout.ConversionPattern=%c: %m%n
```

3 OpenText Identity Console にログインし、ログイベントを変更します。CEF イベントの設定方法については、649 ページの 「監査する CEF イベントの設定」を参照してください。

注 : SyslogAppender に対して UDP プロトコルを使用する CEF キャッシングは機能しません。

Syslog SSL 接続用の証明書の生成

Syslog 接続用の証明書を生成するには :

1. 次の OpenSSL コマンドを使用して証明書を作成します。

   ```
   openssl s_client -host LOG_SERVER  -port 1443 -showcerts
   ```

2. /etc/opt/novell/eDirectory/conf/auditlogconfig.properties ファイルで作成した証明書ファイルの場所を指定します。

ローリングファイルアペンダの有効化

監査ソリューションが個別のサーバに制限されている場合は、このファイルアペンダをお勧めします。ローリングファイルアペンダはローカルファイルシステムにイベントを格納し、イベントの損失を防止するため、syslog アペンダより信頼性が向上します。また、このソリューションは、セットアップするコンポーネントの数が少なく立ち上げが容易なため、デモンストレーションに向いています。

注 : CEF にファイルコネクタを使用している場合は、ローリングファイルアペンダの変換パターンが、次のような Syslog アペンダと同様であることを確認します。

```
log4j.appender.R.layout.ConversionPattern=%c: %m%n
```

ローリングファイルアペンダを有効にするには、auditlogconfig.properties ファイルに次の変更を加えます。

1 次のエントリを R に変更して、ローリングファイルアペンダをアタッチします。

log4j.rootLogger=info, R

2 次のエントリをコメントアウトします。

```
log4j.appender.R=org.apache.log4j.RollingFileAppender

log4j.appender.R.File=/var/opt/novell/eDirectory/log/cef-events.log
```

```
log4j.appender.R.MaxFileSize=100MB

log4j.appender.R.MaxBackupIndex=10

log4j.appender.R.layout=org.apache.log4j.PatternLayout

log4j.appender.R.layout.ConversionPattern=%d{MMM dd HH:mm:ss} %c %m%n
```

3 OpenText Identity Console から目的のイベントを選択します。

CEF イベントの設定方法については、649 ページの 「監査する CEF イベントの設定」を参照してください。

監査用の CEF の設定

- 648 ページの 「OpenText Identity Console を使用した CEF の設定」
- 649 ページの 「監査する CEF イベントの設定」

OpenText Identity Console を使用した CEF の設定

1 次の URL を使用して、Web ブラウザから OpenText Identity Console を開きます。

`https://ip_address_or_DNS/identityconsole/`

ここで、*ip_address_or_DNS* は、OpenText Identity Console サーバの IP アドレスまたは DNS 名およびポート番号です。

例 :

`http://111.111.1.1:9000/`

2 ツリーユーザ名とパスワードを使用してログインします。

OpenText Identity Console のすべての機能を利用するには、管理者としてツリーにログインします。管理者ユーザだけが、すべての機能への完全アクセス権を持っています。管理者ユーザ以外は、権利が割り当てられている役割にのみアクセスできます。

詳細については、『*OpenText Identity Console 管理ガイド (https://www.netiq.com/documentation/identity-console/identity_console-admin/data/t48zlpkxbru1.html)*』を参照してください。

3 [**監査**] を選択します。

4 監査を設定したい必要な **NCP サーバ**をドロップダウンから選択します。

5 必要に応じて、[**イベントの環境設定**] で詳細を指定します。

6 [**OK**] をクリックします。

[CEF 監査] ページが表示されます。639 ページの 「CEF の設定」に進みます。

注 : 管理者は、OpenText Audit Identity Console で次の属性を無効にすることをお勧めします。

- 信頼
- DirXML-DriverStorage
- 破損通知

- パーティションステータス
- レプリカ
- Revision (改訂数)
- SAS: ログインシークレット

7 **適用**をクリックします。［**OK**］をクリックします。

監査する CEF イベントの設定

1 ユーザ名とパスワードを使用して、OpenText Identity Console にログインします。

2 ［**監査**］をクリックします。

3 CEF イベントを設定します。

- **Global:** 重複したエントリのグローバル設定を選択またはクリアすることができます。
  - **複製されたイベントを送信しない :** 他のサーバからの複製に起因する重複イベントの受信を停止する場合にこのオプションを選択します。
- **ログイベントの値 :** イベントがテキストファイルに記録されます。サイズが 768 バイトを超えるイベントの値が「大きい値」と見なされます。どんなサイズのイベントでもログに記録できます。
  - **大きい値をログに記録 :** このオプションは、サイズが 768 バイトを超えるイベントをログに記録する場合に選択します。
  - **大きい値をログに記録しない :** このオプションを選択して、サイズが 768 バイトより小さいイベントをログに記録します。
  - **属性値のログ記録 :** 属性値を表示するにはこのオプションを選択します。これは、［**値の追加**］イベントと［**値の削除**］イベントにのみ適用されます。
  - **属性値をログに記録しない :** 属性値を抑止するにはこのオプションを選択します。これは、［**値の追加**］イベントと［**値の削除**］イベントにのみ適用されます。デフォルトでは、このオプションが選択されています。
  - **暗号化属性値のログ記録 :** 暗号化された属性値を表示するにはこのオプションを選択します。これは、［**値の追加**］イベントと［**値の削除**］イベントにのみ適用されます。
  - **暗号化属性値をログに記録しない :** 暗号化された属性値を抑止するにはこのオプションを選択します。これは、［**値の追加**］イベントと［**値の削除**］イベントにのみ適用されます。デフォルトでは、このオプションが選択されています。

注 : イベントサイズがこのサイズを超えている場合は、イベントの値が切り詰められてログファイルに保存されます。

- **基本イベント環境設定 :** 環境に必要なイベントに基づいて、次のオプションの値を指定します。

注 : デフォルトでは、［基本イベント環境設定］セクションの下に個々のイベントカテゴリが折りたたまれます。個々のイベントを選択するには、各カテゴリを展開できます。

オプション	説明
セキュリティイベント	イベントをログに記録するセキュリティイベントを選択します。メンバーの追加または削除、不正侵入者の検出、パスワード変更、ユーザの認証などのイベントをログに記録できます。
オブジェクトイベント	イベントをログに記録するオブジェクトイベントを選択します。オブジェクトの作成、削除、名前変更、移動、検索を行うイベントをログに記録できます。
属性イベント	イベントをログに記録する属性イベントを選択します。属性の読み込みと削除、属性値の追加、削除、比較を行うイベントをログに記録できます。
LDAP イベント	イベントをログに記録する LDAP イベントを選択します。
EBA イベント	イベントをログに記録する EBA イベントを選択します。

対応する CEF イベントにマッピングされた OpenText eDirectory の内部イベントについては、881 ページの 「OpenText eDirectory イベントと CEF イベントのマッピング」を参照してください。

注 : イベント環境設定を変更してから、NCP サーバ上で設定変更が有効になるまで最大 3 分かかります。NCP サーバ上で設定変更をすぐに有効にするには、cefauditds モジュールをアンロードしてからロードする必要があります。

モジュールのロードとアンロード

CEF イベントを設定したら、次のコマンドを実行して、CEF モジュールをロードまたはアンロードします。

ndsd サーバの起動時に自動的に cefauditds モジュールをロードするには :

- **Linux**

  cefauditds を /etc/opt/novell/eDirectory/conf/ndsmodules.conf ファイルに追加します。

- **Windows**

  ndscons.exe を実行して、使用可能なモジュールのリストから［cefauditds］を選択し、［**スタートアップ**］をクリックしてから、起動のタイプに［**自動**］を選択します。

cefauditds モジュールを手動でロードまたはアンロードするには :

- **Linux**

  ロードするには、ndstrace -c "load cefauditds" を実行します。

アンロードするには、ndstrace -c "unload cefauditds" を実行します。

- **Windows**

  ロードするには、ndscons.exe を実行して、使用可能なモジュールのリストから［**cefauditds**］を選択し、［**開始**］をクリックします。

  アンロードするには、ndscons.exe を実行して、使用可能なモジュールのリストから［**cefauditds**］を選択し、［**停止**］をクリックします。

CEF イベントキャッシングの有効化

OpenText eDirectory 9.2 では、必要に応じて、CEF イベントをエージェントの Syslog アペンダキャッシュにローカルに保存することができます。イベントをキャッシュに保存することにより、エージェントが監査サーバと通信できない場合にも、生成された監査イベントが保持され、監査データが消失しないように守られます。エージェントコンピュータが監査サーバと再び通信できるようになると、エージェントは、キャッシュに保存されたイベントの再送信を試みます。

CEF のイベントキャッシングはデフォルトでは無効となっています。イベントキャッシングを有効にするには、次の手順を実行します。

1. エージェントコンピュータで、CEF プロパティファイルの場所に移動します。auditlogconfig.properties ファイルは、デフォルトで、/etc/opt/novell/eDirectory/conf/auditlogconfig.properties に配置されています。非ルートインストールでは、CEF プロパティファイルがデフォルトで conf ディレクトリに配置されます。
2. テキストエディタを使用して auditlogconfig.properties ファイルを開きます。
3. プロパティファイル内で、log4j.appender.S.CacheEnabled プロパティに移動して、このプロパティ値を yes に変更します。
4. (状況によって実行) 特定のディレクトリでイベントをキャッシュに保存する場合は、そのディレクトリパスを指すように log4j.appender.S.CacheDir プロパティの値を変更します。ディレクトリを指定する場合は、そのディレクトリパスがサーバ上の有効な場所であることを確認します。log4j.appender.S.CacheDir プロパティが設定されていないと、Syslog Appender によって、特定のインスタンスの dib ディレクトリにキャッシュイベントのログが記録されます。
5. キャッシュファイルのカスタムファイルサイズは、次の 2 つのモードで指定できます。

   5a **有限成長モード :** 有限成長モードでは、キャッシュファイルサイズの最小値は 50MB、最大値は 4GB に設定する必要があります。この範囲外の値を設定すると、キャッシュファイルサイズの値はデフォルトで 500MB に設定されます。キャッシュファイルサイズの制限が最大レベルに達すると、キャッシュファイルのロールオーバーが開始されます。このモードでは、キャッシュされたすべてのイベントログが Syslog サーバに送信されても、キャッシュファイルは切り捨てられません。

   5b **無限成長モード :** 無限成長モードを有効にするには、キャッシュファイルサイズを 0MB に設定します。このモードでは、キャッシュファイルのロールオーバーは発生しません。キャッシュファイルの切り捨ては、このモードでキャッシュされ

たすべてのイベントが Syslog サーバに送信された後に実行されます。キャッシュファイルの増加による DIB の成長への影響を避けるため、キャッシュファイルを別のパーティションまたはディスクに配置することをお勧めします。

6 auditlogconfig.properties ファイルを保存して閉じます。

7 CEF モジュールを再ロードします。

CEF キャッシングメカニズム

キャッシングメカニズムでは、キャッシュディレクトリの場所にある次のファイルをすべて使用します。

- CEF-S-cache.log
- RolledOver-CEF-S-record
- OffsetWriter-CEF-S-record
- OffsetReader-CEF-S-record

CEF-S-cache.log ファイルは、Cache Writer と Cache Reader によって提供されます。Syslog サーバへの接続が失われた場合、Writer はイベントをキャッシュログに書き込みます。.OffsetWriter-CEF-S-record ファイルは、Writer Offset で更新され、キャッシュファイルに記録された CEF イベントの量を示します。Writer が最大キャッシュファイルサイズに達すると、キャッシュファイルの先頭にロールオーバーが発生し、.RolledOver-CEF-S-record ファイル内のフラグの値を 0 から 1 に変更します。この時点で、Cache Writer がロールバックしたことを示す適切なメッセージが ndsd.log に記録されます。Writer は、キャッシュログにイベントを書き込み続け、Reader のオフセット値に達する前に停止します。これは、まだ Syslog サーバに送信されていないイベントが、Writer によって上書きされないようにするためです。この後に生成されたイベントは失われ、キャッシュファイルが完全に使用されたことを示す適切なメッセージが ndsd.log に記録されます。

Reader は、キャッシュファイルからイベントを読み込み、接続が再確立されるとそれらを Syslog サーバに送信します。.OffsetReader-CEF-S-record ファイルは、Reader Offset で更新され、キャッシュファイルから Syslog サーバに送信された CEF イベントの量を示します。Writer がより多くの新しいイベントをキャッシュに書き込めるようにするため、Reader の位置が前に進みます。Reader は、ファイルの最後に達するとロールバックし、.RolledOver-CEF-S-record ファイルのフラグは 0 に戻されます。この時点で、Cache Reader がロールバックしたことを示す適切なメッセージが ndsd.log に記録されます。読み込みと送信は、Reader オフセットが Writer オフセットと一致するまで継続されます。このプロシージャにより、キャッシュされたすべてのデータが Syslog サーバに送信されます。このプロセスが完了すると、イベントが Syslog サーバに直接送信されるので、キャッシュファイルの更新は発生しなくなります。

注 : ロールオーバーは無限成長モードでは発生しないので、キャッシュされたすべてのイベントが Syslog サーバに送信されると、キャッシュファイルは切り捨てられます。キャッシュファイルの切り捨てが完了すると、すべてのオフセットもリセットされます。

CEF イベントタイプについて

イベントを次のカテゴリに記録するように、CEF を設定できます。

- セキュリティ
- オブジェクト
- 属性
- LDAP
- EBA

次のデフォルトセットのイベントタイプを監査できます。

カテゴリ	イベントタイプ
セキュリティ	• ACL が変更されました • メンバーの追加 • メンバーの削除 • 不正侵入者が検出されました • ログインが無効化されました • ログインが有効化されました • ログイン • 同等セキュリティの変更 • 監査の環境設定 • パスワードの変更 • アカウントのロック解除 • ログアウト • 接続 • なりすまし • 認証 • パスワードの確認 • ログイン環境設定の変更 • 資格情報の照会
オブジェクト	• オブジェクトの作成 • オブジェクトの削除 • オブジェクトのリネーム • オブジェクトの移動 • DSA の読み取り • 検索

カテゴリ	イベントタイプ
属性	◆ 属性の読み込み ◆ 属性の削除 ◆ 値の追加 ◆ 値の削除 ◆ 属性値の比較
LDAP	◆ LDAP バインド ◆ LDAP レスポンスのバインド ◆ LDAP バインド解除 ◆ LDAP 接続 ◆ LDAP 検索 ◆ LDAP 検索の応答 ◆ LDAP 検索エントリの応答 ◆ LDAP 追加 ◆ LDAP 追加の応答 ◆ LDAP 比較 ◆ LDAP 比較の応答 ◆ LDAP 変更 ◆ LDAP レスポンスの変更 ◆ LDAP 削除 ◆ LDAP レスポンスの削除 ◆ LDAP DN の変更 ◆ LDAP DN レスポンスの変更 ◆ LDAP 破棄 ◆ LDAP 拡張操作 ◆ LDAP システム拡張操作 ◆ LDAP 拡張操作の応答 ◆ LDAP サーバの環境設定の変更 ◆ 不明な LDAP 操作 ◆ LDAP パスワードの変更
EBA	◆ サービスの環境設定の変更

注 : *Modify LDAP Server Configuration* イベントは、LDAP サーバが LDAP クライアント要求で更新されると発行されます。

CEF イベントのコレクタの使用

コレクタを使用して CEF イベントを収集する方法について詳しくは、Sentinel Plug-ins ページの OpenText eDirectory コレクタ資料を参照してください。

CEF 監査イベントのフィルタ処理について

CEF は、フィルタとイベント通知を使用して、特定のタイプのイベントが発生したとき、または、発生しなかったときにレポートを作成できます。1 つまたは複数の固有オブジェクトクラスまたは属性のイベントを、イベントタイプに応じてフィルタ処理することもできます。CEF は、生成されたすべてのイベントを OpenText eDirectory サーバで設定済みのフィルタに照らして評価し、それらのフィルタに一致するイベントのみをログに記録します。

このセクションでは、システムフィルタと通知の設定に必要な情報を提供します。

- 655 ページの 「CEF オブジェクトイベントのフィルタリング」
- 656 ページの 「CEF 属性イベントのフィルタリング」
- 656 ページの 「除外フィルタを使用した OpenText eDirectory イベントのフィルタリング」

CEF オブジェクトイベントのフィルタリング

オブジェクト用のフィルタリングを設定して、特定のイベントのみを検索することができます。たとえば、誰かが OpenText eDirectory でユーザアカウントを作成したら通知が届くようにするには、新しいユーザオブジェクトの作成に関するイベントをログに記録するため、ユーザオブジェクトクラスを選択するフィルタを作成できます。

アカウントフィルタリングを設定するには、[オブジェクトイベント] リンクをクリックして、クラスを選択してから、[OK] をクリックしてアプリケーションを終了します。

アカウント管理イベントのフィルタを設定するには :

1. OpenText Identity Console のホームページで、[監査] に移動します。
2. ドロップダウンメニューから監視対象の [NCP サーバ] を選択し、[OK] をクリックします。
3. [オブジェクトイベント] タイルをクリックします。
4. 設定する必要なオブジェクトイベントを選択します。
5. [適用] をクリックして [OK] をクリックすると、監査の詳細が正常に変更されます。

設定済みのフィルタを使用して、CEF 監査モジュールはすべての選択済みオブジェクトクラスと属性に関して生成されたイベントすべてをチェックし、それらのイベントをログに記録します。

CEF 属性イベントのフィルタリング

［**属性イベント**］リンクをクリックして、属性イベントのフィルタを設定します。たとえば、誰かが OpenText eDirectory で新しい属性値を追加したら通知が届くようにするため、新しい値の追加に関するイベントをログに記録するフィルタを作成できます。

トラスト管理イベントのフィルタを設定するには :

1 OpenText Identity Console で、［**eDirectory 監査 (eDirectory Auditing)**］に移動します。

2 監視する NCP サーバを選択してから、［**OK**］をクリックします。

3 ［**イベント属性**］をクリックします。

［**属性フィルタリング**］ウィンドウが表示されます。

4 ［**使用可能なクラス**］リストで、イベントを収集するオブジェクトクラスを選択し、右矢印をクリックしてそれらを［**選択されたクラス**］リストに移動します。デフォルトでは、［**dynamicGroup**］、［**dynamicGroupAux**］、［**グループ**］、［**LDAP グループ**］、［**職種**］オブジェクトクラスが選択されています。

5 ［**使用可能な属性**］リストで、選択したオブジェクトクラスの属性を任意の数だけ選択します。属性を選択し、右矢印をクリックしてその属性を選択された属性のリストに追加します。

注 : オブジェクトクラスを選択すると、そのオブジェクトクラスのすべての属性の属性イベントすべてが選択されます。この場合は、選択したオブジェクトクラスのすべての属性に関するすべての属性イベントを取得することになります。

---

6 ［**OK**］をクリックします。

フィルタを設定すると、CEF 監査モジュールはすべての選択済みオブジェクトクラスと属性に関して生成されたイベントすべてをチェックし、それらのイベントをログに記録します。

除外フィルタを使用した OpenText eDirectory イベントのフィルタリング

［**除外フィルタ**］リンクをクリックして、イベントを生成する必要がないオブジェクトクラスと属性のためのフィルタを設定します。オブジェクトクラスと属性を選択できます。

不要な OpenText eDirectory イベントのフィルタを設定するには :

1 OpenText Identity Console のホームページで、［**監査**］タイルをクリックします。

2 監視する NCP サーバを選択してから、［**OK**］をクリックします。

3 ［**詳細設定**］ドロップダウンから［**除外フィルタ**］をクリックします。

［CEF 除外フィルタリング］ウィンドウが表示されます。

4 ［**使用可能なオブジェクトクラス**］リストで、イベントを収集しないオブジェクトクラスを選択し、右矢印をクリックしてそれらを［**選択されたオブジェクトクラス**］リストに移動します。

5 ［**使用可能な属性**］リストで、任意の数の属性を選択します。° 属性を選択し、右矢印をクリックしてその属性を選択された属性のリストに追加します。

6 ［**OK**］をクリックします。

設定済みのフィルタを使用して、CEF 監査モジュールは、すべての選択済みオブジェクトクラスと属性に関するイベントの生成を停止します。

CEF 実装スキーマ

このドキュメントでは、CEF プロトコルを定義し、標準の実装方法の詳細を記します。ここでは、標準で使用されるヘッダと定義済みの拡張について説明します。

Syslog での CEF の使用

CEF では、syslog をトランスポートメカニズムとして使用します。次に示すように、syslog プレフィックス、ヘッダ、および拡張で構成される形式を使用します。

```
Jan 18 11:07:53 host CEF:Version|Device Vendor|Device Product|Device
Version|Device Event Class ID|Name|Severity|[Extension]
```

メッセージの CEF:Version 部分は、必須のヘッダです。メッセージの残りの部分は、パイプ文字 (「|」) で区切られたフィールドを使用してフォーマットします。残りのフィールドはすべて必要で、657 ページの 「CEF フィールドの定義」に従って定義しなければなりません。

メッセージの拡張部分は他のフィールドのプレースホルダで、必須ではありません。他のフィールドは、キーと値のペアとしてログに記録されます。詳細については、657 ページの 「CEF フィールドの定義」を参照してください。

CEF フィールドの定義

スキーマ内のこれらのフィールドは、CEF 監査イベント用に定義されたフィールドです。これらのフィールドの一部または全部が他のタイプのイベントに関係している場合もありますが、監査サービスにはこの種の情報が不可欠です。

表 23-3 *CEF フィールドの定義*

CEF フィールド	説明
Device Product	Device Product は、送信元のデバイスを固有に識別する文字列です。2 つの製品が、同じデバイス製品のペアを持つことはできません。管理者は、デバイス製品のペアごとに固有の名前を割り当てる必要があります。
Device Version	Device Version は、送信元のデバイスを固有に識別する文字列です。2 つの製品が、同じデバイスバージョンのペアを持つことはできません。管理者は、デバイスバージョンのペアごとに固有の名前を割り当てる必要があります。
Device Vendor	Device Vendor は、送信元のデバイスを固有に識別する文字列です。2 つの製品が、同じデバイスベンダーのペアを持つことはできません。管理者は、デバイスベンダーのペアごとに固有の名前を割り当てる必要があります。

CEF フィールド	説明
Device Event Class ID	Device Event Class ID は、イベントタイプごとの固有の ID です。文字列または整数を指定できます。デバイスイベントクラス ID は、報告されたイベントのタイプを識別します。侵入検知システム (IDS) では、特定の動作を検出する署名または規則ごとに固有のデバイスイベントクラス ID が割り当てられています。これは、他のタイプのデバイスの要件でもあり、イベントを処理する Correlation Engine instances でも役立ちます。署名 ID とも呼ばれます。
Severity	イベントの重要度を示す、文字列または整数です。有効な文字列値は [**不明**]、[**低い**]、[**中**]、[**高い**]、[**非常に高い**] です。有効な整数値は 0-3(低い)、4-6(中)、7-8(高い)、9-10(非常に高い) です。 eDirectory 9.2 SP2 以降では、値と整数の次のマッピングが実行されます。 ◆ 1 - TRACE ◆ 2 - DEBUG ◆ 4 - INFO ◆ 6 - WARN ◆ 8 - ERROR
Version	CEF 形式のバージョンを示す整数です。イベントコンシューマは、この情報を使用して、その後に続くフィールドが表す内容を判別します。現在の CEF バージョンは 0 です。
Device Address	IP ネットワーク内でイベントが参照するデバイスのアドレスを識別します。形式は、IPv4 アドレスです。
c6a1	このディクショナリ内で適用されるフィールドが他にはない場合にフィールドをマップするために利用できる 4 つの IPV6 アドレスフィールドの 1 つです。
dvchost	形式は、ノードが使用可能なときに、デバイスノードに関連付けられている完全修飾ドメイン名 (FQDN) でなければなりません。たとえば、host.domain.com または host です。
rt	アクティビティに関連するイベントの受信時刻です。形式は、MMM dd yyyy HH:mm:ss またはエポック以降のミリ秒です。
dtz	イベントを生成するデバイスのタイムゾーン。
sourceServiceName	このイベントの生成を担当するサービス。
sproc	イベントのソースプロセスの名前。
src	IP ネットワーク内でイベントが参照するソースを識別します。形式は、IPv4 アドレスです。たとえば、192.168.10.1。
spt	これは、ソースポート番号です。有効なポート番号は、0 から 65535 です。

CEF フィールド	説明
shost	IP ネットワーク内でイベントが参照するソースを識別します。形式は、ノードが使用可能なときに、ソースノードに関連付けられている完全修飾ドメイン名 (FQDN) でなければなりません。たとえば、host.domain.com または host。
suser	名前でソースユーザを識別します。電子メールアドレスも、UserName フィールドにマップされます。送信者は、sourceUserName に配置される候補です。
dst	IP ネットワークでイベントが参照するあて先アドレスを識別します。形式は、IPv4 アドレスです。たとえば、192.168.10.1。
duser	名前で宛先ユーザを識別します。これは、イベントの宛先に関連付けられているユーザです。多くの場合、電子メールアドレスが UserName フィールドにマップされます。受信者は、destinationUserName に配置される候補です。
cn1	このディクショナリ内で適用されるフィールドが他にはない場合にフィールドをマップするために利用できる 3 つの数値フィールドの 1 つです。控えめに使用し、可能な場合にはより具体的なディクショナリ提供フィールドを使用するようにします。deviceCustomNumber1 とも呼ばれます。
cn2	このディクショナリ内で適用されるフィールドが他にはない場合にフィールドをマップするために利用できる 3 つの数値フィールドの 1 つです。控えめに使用し、可能な場合にはより具体的なディクショナリ提供フィールドを使用するようにします。deviceCustomNumber2 とも呼ばれます。
cn3	このディクショナリ内で適用されるフィールドが他にはない場合にフィールドをマップするために利用できる 3 つの数値フィールドの 1 つです。控えめに使用し、可能な場合にはより具体的なディクショナリ提供フィールドを使用するようにします。deviceCustomNumber3 とも呼ばれます。
cn1Label	すべてのカスタムフィールドには、フィールド自体を表すことができる対応するラベルフィールドがあります。各フィールドは、そのフィールドの目的を記述する文字列です。deviceCustomNumber1Label とも呼ばれます。
cn2Label	すべてのカスタムフィールドには、フィールド自体を表すことができる対応するラベルフィールドがあります。各フィールドは、そのフィールドの目的を記述する文字列です。deviceCustomNumber2Label とも呼ばれます。
cn3Label	すべてのカスタムフィールドには、フィールド自体を表すことができる対応するラベルフィールドがあります。各フィールドは、そのフィールドの目的を記述する文字列です。deviceCustomNumber3Label とも呼ばれます。
cs1	このディクショナリ内で適用されるフィールドが他にはない場合にフィールドをマップするために利用できる 6 つの文字列の 1 つです。控えめに使用し、可能な場合にはより具体的なディクショナリ提供フィールドを使用するようにします。deviceCustomString1 とも呼ばれます。

CEF フィールド	説明
cs2	このディクショナリ内で適用されるフィールドが他にはない場合にフィールドをマップするために利用できる 6 つの文字列の 1 つです。控えめに使用し、可能な場合にはより具体的なディクショナリ提供フィールドを使用するようにします。deviceCustomString2 とも呼ばれます。
cs3	このディクショナリ内で適用されるフィールドが他にはない場合にフィールドをマップするために利用できる 6 つの文字列の 1 つです。控えめに使用し、可能な場合にはより具体的なディクショナリ提供フィールドを使用するようにします。deviceCustomString3 とも呼ばれます。
cs4	このディクショナリ内で適用されるフィールドが他にはない場合にフィールドをマップするために利用できる 6 つの文字列の 1 つです。控えめに使用し、可能な場合にはより具体的なディクショナリ提供フィールドを使用するようにします。deviceCustomString4 とも呼ばれます。
cs5	このディクショナリ内で適用されるフィールドが他にはない場合にフィールドをマップするために利用できる 6 つの文字列の 1 つです。控えめに使用し、可能な場合にはより具体的なディクショナリ提供フィールドを使用するようにします。deviceCustomString5 とも呼ばれます。
cs6	このディクショナリ内で適用されるフィールドが他にはない場合にフィールドをマップするために利用できる 6 つの文字列の 1 つです。控えめに使用し、可能な場合にはより具体的なディクショナリ提供フィールドを使用するようにします。deviceCustomString6 とも呼ばれます。
cs1Label	すべてのカスタムフィールドには、フィールド自体を表すことができる対応するラベルフィールドがあります。各フィールドは、そのフィールドの目的を記述する文字列です。deviceCustomString1Label とも呼ばれます。
cs2Label	すべてのカスタムフィールドには、フィールド自体を表すことができる対応するラベルフィールドがあります。各フィールドは、そのフィールドの目的を記述する文字列です。deviceCustomString2Label とも呼ばれます。
cs3Label	すべてのカスタムフィールドには、フィールド自体を表すことができる対応するラベルフィールドがあります。各フィールドは、そのフィールドの目的を記述する文字列です。deviceCustomString3Label とも呼ばれます。
cs4Label	すべてのカスタムフィールドには、フィールド自体を表すことができる対応するラベルフィールドがあります。各フィールドは、そのフィールドの目的を記述する文字列です。deviceCustomString4Label とも呼ばれます。
cs5Label	すべてのカスタムフィールドには、フィールド自体を表すことができる対応するラベルフィールドがあります。各フィールドは、そのフィールドの目的を記述する文字列です。deviceCustomString5Label とも呼ばれます。
cs6Label	すべてのカスタムフィールドには、フィールド自体を表すことができる対応するラベルフィールドがあります。各フィールドは、そのフィールドの目的を記述する文字列です。deviceCustomString6Label とも呼ばれます。

CEF フィールド	説明
flexString1	このディクショナリ内で適用されるフィールドが他にはない場合に文字列データをマップするために利用できる 2 つの文字列フィールドの 1 つです。控えめに使用し、可能な場合にはより具体的なディクショナリ提供フィールドを使用するようにします。通常これらのフィールドはお客様が使用するために予約されていて、必要がない限りベンダーが設定すべきではありません。
flexString2	このディクショナリ内で適用されるフィールドが他にはない場合に文字列データをマップするために利用できる 2 つの文字列フィールドの 1 つです。控えめに使用し、可能な場合にはより具体的なディクショナリ提供フィールドを使用するようにします。通常これらのフィールドはお客様が使用するために予約されていて、必要がない限りベンダーが設定すべきではありません。
flexString1Label	このディクショナリ内で適用されるフィールドが他にはない場合に文字列データをマップするために利用できる 2 つの文字列フィールドの 1 つです。控えめに使用し、可能な場合にはより具体的なディクショナリ提供フィールドを使用するようにします。通常これらのフィールドはお客様が使用するために予約されていて、必要がない限りベンダーが設定すべきではありません。
flexString2Label	このディクショナリ内で適用されるフィールドが他にはない場合に文字列データをマップするために利用できる 2 つの文字列フィールドの 1 つです。控えめに使用し、可能な場合にはより具体的なディクショナリ提供フィールドを使用するようにします。通常これらのフィールドはお客様が使用するために予約されていて、必要がない限りベンダーが設定すべきではありません。
flexNumber1	このディクショナリ内で適用されるフィールドが他にはない場合に Long データをマップするために利用できる 2 つの数値フィールドの 1 つです。控えめに使用し、可能な場合にはより具体的なディクショナリ提供フィールドを使用するようにします。通常これらのフィールドはお客様が使用するために予約されていて、必要がない限りベンダーが設定すべきではありません。
flexNumber2	このディクショナリ内で適用されるフィールドが他にはない場合に Long データをマップするために利用できる 2 つの数値フィールドの 1 つです。控えめに使用し、可能な場合にはより具体的なディクショナリ提供フィールドを使用するようにします。通常これらのフィールドはお客様が使用するために予約されていて、必要がない限りベンダーが設定すべきではありません。
flexNumber1Label	このディクショナリ内で適用されるフィールドが他にはない場合に Long データをマップするために利用できる 2 つの数値フィールドの 1 つです。控えめに使用し、可能な場合にはより具体的なディクショナリ提供フィールドを使用するようにします。通常これらのフィールドはお客様が使用するために予約されていて、必要がない限りベンダーが設定すべきではありません。

CEF フィールド	説明
flexNumber2Label	このディクショナリ内で適用されるフィールドが他にはない場合に Long データをマップするために利用できる 2 つの数値フィールドの 1 つです。控えめに使用し、可能な場合にはより具体的なディクショナリ提供フィールドを使用するようにします。通常これらのフィールドはお客様が使用するために予約されていて、必要がない限りベンダーが設定すべきではありません。
cat	元のデバイスが割り当てたカテゴリを表します。多くの場合、デバイスでは独自のカテゴリスキーマを使用してイベントを分類します。deviceEventCategory とも呼ばれます。たとえば、/Monitor/Disk/Read。
reason	監査イベントが生成された理由。たとえば、不正なパスワードまたは不明ユーザ。エラーや、戻りコードのこともあります。
outcome	通常、成功または失敗などの結果を表示します。eventOutcome とも呼ばれます。
msg	イベントの説明

イベントの例

イベントの例を以下に示します。

```
Apr 15 19:43:37 eDirectory
CEF:0|NetIQ|eDirectory|9.2.2|CEF0B035D|AUTHENTICATE|4|dvc=164.99.179.219
dvchost=WIN-ADKHVNQHFCR rt=1586960017190 dtz=India Standard Time
sourceServiceName=CN\=WIN19,O\=novell sproc=eDirectory#DS
src=164.99.179.219 spt=0 suser=CN\=admin,O\=novell
duser=CN\=admin,O\=novell cs2Label=Class Name cs2=User cs3Label=Tree Name
cs3=WIN19_tree cs4Label=Correlation ID cs4=eDirectory#24# cs6Label=Server
Name cs6=CN\=WIN19,O\=novell flexString2Label=SubEvent
flexString2=DSE_AUTHENTICATE flexNumber2Label=Grouping flexNumber2=2131
cat=Security reason=0 outcome=Success msg=CN\=admin,O\=novell (Class:
User) authenticated from  to server CN\=WIN19,O\=novell : Success
```

CEF イベント

CEF イベントの詳細については、881 ページの付録 G「OpenText eDirectory イベントと CEF イベントのマッピング」を参照してください。

ジャーナルイベントのキャッシング

OpenText eDirectory は、イベントコンシューマがイベントに登録して、それらのイベントが発生したらそれをコンシュームすることができるイベントシステムを備えています。イベントハンドラは、ワーカー、インライン、またはジャーナルとして登録することができます。ジャーナルイベントキューは、発生と同じ順序でイベントを報告することになっています。現在のジャーナルイベントシステムでは、ジャーナルイベントキューがメモリ内

に保持されます。イベントのコンシューマが低速の場合やイベントの発生速度が処理速度を上回っている場合は、ジャーナルキューが増加し始めます。その結果、ndsd プロセスのメモリが増大します。

このジャーナルイベントシステムは変更され、メモリとディスクを組み合わせて使用することでイベントを 1 つのキュー内に保持できるようになりました。これにより、ndsd プロセスのメモリ使用量の急激な増加が軽減されます。

イベントが原因でメモリが増加する可能性があるのは、*ndstrace が有効*になっている場合か、*監査が有効*になっている場合です。メモリの増大は、イベントシステムキャッシングを有効にすることによって制御できます。

イベントシステムキャッシングの設定

イベントシステムキャッシングのための次の環境変数を設定する必要があります。

- NDSD_EVENT_DISK_CACHE

  この変数は、新しいイベントシステムの使用を制御します。デフォルトで、新しいイベントシステムは無効になっています。新しいイベントシステムを有効にするには、値を *true* または *1* にしてこの変数をエクスポートします。

- (オプション) NDSD_EVENT_DISK_CACHE_DIR

  この変数は、イベントファイルが作成される一時的な場所を指定します。指定されたディレクトリの下に、別のサブディレクトリ cdir (存在しなければ) が作成されます。起動時に、このサブディレクトリ内のすべてのファイルがクリーンアップされます。キャッシングディレクトリは、別のディスクパーティション内に設定し、DIB と同じパーティション内には設定しないことをお勧めします。

  Linux では、NDSD_EVENT_DISK_CACHE_DIR が指定されていない場合や指定されたディレクトリにアクセスできない場合、ndsd は vardir をキャッシングディレクトリとして使用します。デフォルトで、vardir の値は /var/opt/novell/eDirectory/data/ です。

  Windows では、この変数が指定されていない場合や指定されたディレクトリにアクセスできない場合、dhost は DIBFiles ディレクトリを使用します。

注 : キャッシングディレクトリで十分なディスク容量が使用できることを確認してください。ndsd/dhost は数 GB のディスク容量をすぐに消費する可能性があります。

LDAP 監査

監査機能は、ディレクトリを評価する際に、管理者が興味を持つ主要な機能の 1 つです。OpenText eDirectory のイベントメカニズムは、OpenText eDirectory の監査を容易にします。多くのアプリケーションがディレクトリにアクセスするために LDAP プロトコルを導入しているので、LDAP の処理を監査する必要性が大いに広まっています。

この章では次のセクションについて説明します。

- 664 ページの 「LDAP 監査の必要性」
- 664 ページの 「LDAP 監査の利用」
- 665 ページの 「その他の情報」

LDAP 監査の必要性

このイベントメカニズムは既存の OpenText eDirectory LDAP サーバでは明らかに不足しており、これらのサーバでは十分な LDAP 情報を提供できませんでした。NDS イベントシステムはすべての OpenText eDirectory 操作に対するイベントを生成していましたが、その情報のほとんどはアプリケーションが LDAP サーバを監査するのに不十分または不適切でした。プロトコルやバインドの詳細、ネットワークアドレス、認証方法、認証タイプ、LDAP 検索、トランザクションの詳細などの情報が LDAP サーバの監査に不可欠ですが、NDS イベントでは利用できませんでした。アプリケーション開発者にとって、従来のイベントをベースにしたアプリケーションで監査機能を実現することは困難でした。

LDAP は、OpenText eDirectory の重要なインタフェースです。アプリケーションが OpenText eDirectory LDAP サーバを監査するためのメカニズムを提供するために、OpenText eDirectory には LDAP イベントサブシステムが組み込まれています。このサブシステムは、アプリケーションが LDAP サーバを監査するのに関連のあるすべての情報を含む LDAP 特有のイベントを生成します。これは LDAP 監査として知られています。

LDAP 監査の利用

LDAP 監査は、アプリケーションが追加、変更、検索などの LDAP の処理を監視 / 監査し、接続情報や LDAP 処理の際にサーバが接続するクライアント IP、メッセージ ID、処理に対する結果コードなどの有用な情報を LDAP サーバから取り込みます。

LDAP 監査は、LDAP Libraries for C (https://www.microfocus.com/documentation/edirectory-developer-documentation/ldap-libraries-for-c/) により実行されますが、新 LDAP の構造とイベントを通して監査機能をクライアント側のインタフェースとして提供します。

その他の情報

LDAP 監査イベントの詳細については、次のマニュアルを参照してください。

- NDK LDAP ツール (https://www.microfocus.com/documentation/edirectory-developer-documentation/ldap-classes-for-java/docs/jldapenu.pdf) に関しては、LDAP Libraries for C のマニュアルを参照してください。
- LDAP のツール情報に関しては、『LDAP Libraries for C (https://www.microfocus.com/documentation/edirectory-developer-documentation/ldap-libraries-for-c/)』を参照してください。

24 OpenText eDirectory の認証フレームワークについて

このセクションでは、NMAS の概要を示します。NMAS は、OpenText eDirectory と一緒に自動的にインストールされます。サポートされているプラットフォームとインストール手順の詳細については、『*OpenText™ eDirectory CE 24.4 (v9.3) インストールガイド*』を参照してください。

- 667 ページの 「NMAS の機能」
- 671 ページの 「NMAS ソフトウェア」
- 674 ページの 「ログインとポストログインのメソッドとシーケンスの管理」
- 682 ページの 「NMAS を使用したネットワークへのログイン」
- 684 ページの 「OpenText パスワードの履歴」
- 685 ページの 「NMAS HOTP ベースのログイン」
- 691 ページの 「その他の管理タスク」
- 697 ページの 「セキュリティ上の考慮事項」

NMAS の機能

NMAS は、ネットワーク上の情報の保護を支援するように設計されています。パスワード管理ツールに加えて、NMAS は OpenText eDirectory ネットワークに対して認証する手段も兼ね備えています。これにより、ネットワークリソースにアクセスしている人物が本人かどうかを確認できます。

NMAS は、認証デバイスに対するワークステーション上でのユーザのセッション中に 3 つの操作フェーズを使用します。それらのフェーズは次のとおりです。

1. ユーザ識別フェーズ (人物の特定)
2. 認証 (ログイン) フェーズ (本人であることの証明)
3. デバイス取り外し検出フェーズ (接続の確認)

この 3 つの操作フェーズは完全に独立しています。認証デバイスは各フェーズで使用できますが、毎回同じデバイスを使用する必要はありません。

ユーザ識別フェーズ

これは、ユーザ名を収集するプロセスです。また、このフェーズでは、認証フェーズで使用されるツリー名、ユーザのコンテキスト、サーバ名、および NMAS シーケンスの名前も入力されます。この認証情報は、認証デバイスから取得することも、ユーザが手動で入力することもできます。

認証 (ログイン) フェーズ

- 668 ページの 「パスワード認証」
- 669 ページの 「物理デバイス認証」
- 669 ページの 「バイオメトリック認証」

NMAS は、ログインファクタと呼ばれる、ネットワークにログインするための 3 種類のアプローチを使用します。このログインファクタは、ユーザがネットワークに対する認証に使用できる各種の項目または品質を表します。

- パスワード認証 (ユーザが知っていること)
- 物理デバイス認証 (ユーザが持っているもの)
- バイオメトリック認証 (ユーザそのもの)

ログインファクタの詳細については、670 ページの 「ログインメソッドとポストログインメソッドとシーケンス」を参照してください。

パスワード認証

パスワード (ユーザが知っていること) は、ネットワークに対する認証のための重要なメソッドです。NMAS には、複数のパスワード認証オプションが用意されています。

- **NDS ログインメソッド :** NDS パスワードは非可逆のハッシュ形式で保存され、NDS システムだけがこのパスワードを使用できます。このオプションは、ユニバーサルパスワードが有効にされていて設定済みであればそれを使用します。
- **SCRAM ログインメソッド :** SCRAM (Salted Challenge Response Authentication Mechanism) は、ユーザが OpenText eDirectory サーバにログインしようとしたときに、パスワードの PBKDF2 ハッシュを使用してユーザを認証します (RFC 5802)。詳細については、788 ページの 「非可逆パスワードストレージを理解する」を参照してください。

  注 : OpenText eDirectory 9.2 ツリーを新規作成する際に、SCRAM メソッドが自動的にインストールされます。旧バージョンの OpenText eDirectory ツリーを 9.2 にアップグレードする際には、SCRAM メソッドを手動でインストールする必要があります。詳細については、674 ページの 「ログインメソッドのインストール方法」を参照してください。

- **単純パスワード :** 単純パスワードを使用する場合、管理者は、外部の LDAP ディレクトリからユーザとパスワード (平文とハッシュ済み) をインポートすることができます。このオプションは、ユニバーサルパスワードが有効にされていて設定済みであればそれを使用します。
- **Digest-MD5 SASL:** Digest-MD5 SASL は、LDAP SASL バインドに使用される MD5 アルゴリズムによってハッシュされたパスワードを検証する IETF 標準の DIGEST-MD5 SASL メカニズムを提供します。このオプションは、ユニバーサルパスワードが有効にされていて設定済みであればそれを使用します。
- **チャレンジ / レスポンス方式 :** チャレンジ / レスポンス方式は、ユーザが事前に設定された秘密の質問に 1 回以上回答することによって自分の身元を証明する方法です。

ユニバーサルパスワードは、各種のパスワードと認証のシステムの統合と管理を簡略化して 1 つの首尾一貫したネットワークを構築するための手段です。ユニバーサルパスワードの詳細については、785 ページの第 26 章「パスワードを管理する」を参照してください。

物理デバイス認証

これまで、いくつかのタイプの物理デバイス (ユーザが持っているもの) 向けの NMAS 用認証モジュールが、OpenText とサードパーティの認証開発者により作成されてきました。

注 : NMAS では、すべての物理デバイス認証メソッド (証明書付きスマートカード、ワンタイムパスワード (OTP) デバイス、近接型カードなど) を指す言葉として用いています。

- **スマートカード :** スマートカードは、クレジットカード大のプラスティックカード、または、データの保存や暗号化機能の実行が可能な組み込み型のプログラマブルマイクロチップを含む USB デバイスです。NMAS を使用すれば、OpenText eDirectory に対する認証時にスマートカードで身元確認を行えます。

  OpenText は、スマートカードを使用するための OpenText Enhanced Smart Card ログインメソッドを提供しています。OpenText Enhanced Smart Card ログインメソッドは、Identity Assurance Client の一部として提供されます。詳細については、『*OpenText Enhanced Smart Card Method 3.0 Installation and Administration Guide (OpenText Enhanced Smart Card メソッド 3.0 インストールおよび管理ガイド)*』を参照してください。
- **ワンタイムパスワード (OTP) デバイス :** OTP デバイスは、その所有者を認証するためのワンタイムパスワードを生成するハンドヘルドハードウェアデバイスです。
- **近接型カード :** 近接型カードは、ユーザが装着するカードです。この技術は、ユーザのワークステーションとカードの近接性に基づいてそのワークステーションをロック / ロック解除します。

  OpenText は、RFID 近接型カードをサポートする pcProx ログインメソッドを提供しています。pcProx ログインメソッドは、OpenText SecureLogin 製品の一部として提供されます。詳細については、『*NMAS Login Method and Login ID Snap-In for pcProx (pcProx 用の NMAS ログインメソッドおよびログイン ID スナップイン)*』を参照してください。

バイオメトリック認証

*バイオメトリックス*は、人体の特徴 (ユーザそのもの) を測定して統計的に解析するための科学技術です。バイオメトリック方式は、NMAS で使用するためにサードパーティ企業から提供されます。

バイオメトリック認証には、リーダーまたはスキャニングデバイス、スキャンされた情報をデジタル形式に変換するソフトウェア、および、入力されたバイオメトリックデータと比較するバイオメトリックデータを保存するためのデータベースまたはディレクトリが必要です。

バイオメトリック入力の変換では、ソフトウェアが特定のデータのポイントを一致ポイントとして識別します。この一致ポイントがアルゴリズムを使用して処理され、ユーザがアクセス権を取得しようとするときにスキャンされるバイオメトリックデータと比較することができる値が作成されます。

バイオメトリック認証の例としては、指紋、網膜、虹彩、および顔の特徴のスキャンが挙げられます。バイオメトリックスにはさらに筆跡、タイピングパターン、音声認識なども含まれます。

デバイス取り外し検出フェーズ

ユーザセッションは、ログインの完了後にこのフェーズに入ります。次の 2 つのメソッドがあります。

- OpenText SecureLogin で使用可能な Secure Workstation メソッド。ユーザセッションは、認証デバイス (スマートカードなど) を取り外すと終了できます。このデバイスは他のフェーズで使用する必要はありません。.

  Secure Workstation メソッドの詳細については、『OpenText SecureLogin Administration Guide (OpenText SecureLogin 7.0 SP3 管理ガイド) (https://www.microfocus.com/documentation/securelogin/9.2/administration_guide/front.html)』を参照してください。
- OpenText Enhanced Smart Card ログインメソッドにも、スマートカード取り外し検出機能があります。OpenText Enhanced Smart Card ログインメソッドの詳細については、『*OpenText Enhanced Smart Card Method Installation Guide (OpenText Enhanced Smart Card メソッドインストールガイド)*』を参照してください。

ログインメソッドとポストログインメソッドとシーケンス

ログインメソッドは、ログインファクタの特定の実装です。NMAS には、3 つのログインファクタ (パスワード、物理デバイス、バイオメトリック認証) に基づく複数のログインメソッドの選択肢があります。

*ポストログインメソッド*は、ユーザが OpenText eDirectory に対して認証された後に実行されるセキュリティプロセスです。ポストログインメソッドの 1 つの例が OpenText Secure Workstation メソッド (OpenText SecureLogin で使用可能) です。このメソッドでは、ワークステーションのロック後にコンピュータにアクセスするためにユーザが資格情報を入力する必要があります。

NMAS ソフトウェアには、OpenText とサードパーティ認証開発者による複数のログインメソッドとポストログインメソッドのサポートが含まれます。ログインメソッドによっては、追加のハードウェアが必要な場合があります。詳細については、サードパーティ製品のマニュアルを参照してください。

メソッドを決定してインストールしたら、それを使用するために、ログインシーケンスに割り当てる必要があります。*ログインシーケンス*は、順序付けられた 1 つ以上のメソッドのセットです。ユーザは、これらの定義されたログインシーケンスを使用してネットワークにログインします。シーケンスに複数のログインメソッドが含まれる場合、それらのメソッドは指定された順序でユーザに提示されます。ログインメソッドが最初に提示され、その後にポストログインメソッドが提示されます。

NMAS には、And と Or の両方のログインシーケンスが存在します。And ログインシーケンスでは、シーケンス内のすべてのログインメソッドを正常に完了する必要があります。Or ログインシーケンスでは、シーケンス内のいずれかのログインメソッドのみを正常に完

了する必要があります。Or ログインシーケンスの 1 つの例としては、ユーザが各種の認証デバイスで同じログインシーケンスを使用してワークステーションにログインできるようにしている場合が挙げられます。

セキュリティオブジェクトのキャッシュ

セキュリティコンテナは、ツリーに最初のサーバがインストールされたときにルートパーティションから分かれて作成され、グローバルデータ、セキュリティポリシー、キーなどの情報を保持します。

ユニバーサルパスワードが導入された後は、ユーザが NMAS を介して OpenText eDirectory にログインするたびに、NMAS がセキュリティコンテナ内の情報にアクセスしてログインを認証していました。セキュリティコンテナがあるパーティションがローカルに存在しない場合、NMAS はそのパーティションを持つサーバにアクセスしていました。このとき、NMAS 認証のパフォーマンスに悪影響が及んでいました。セキュリティコンテナがあるパーティションを持つサーバに WAN リンク経由でアクセスする必要がある状況では、この問題はさらに悪化しました。

これを解決するために、セキュリティコンテナデータがローカルサーバ上のキャッシュに保存されます。このため NMAS は、ユーザがログインするたびに、異なるコンピュータに置かれているセキュリティコンテナにアクセスする必要がありません。セキュリティコンテナには、ローカルで容易にアクセスすることができます。これによってパフォーマンスが向上します。セキュリティコンテナがあるパーティションをローカルサーバに追加することでパフォーマンスは向上しますが、サーバの数が多すぎる場合はそうはいかない可能性があります。

セキュリティコンテナ内の実際のデータが、セキュリティコンテナのパーティションを含むサーバ上で変更された場合、ローカルキャッシュはバックリンカと呼ばれるバックグラウンドプロセスによってリフレッシュされます。デフォルトでは、バックリンカが 13 時間ごとに実行され、変更されたデータがリモートサーバから取得されます。データの即時同期が必要な場合は、OpenText iMonitor、Linux 上の ndstrace、または Windows 上の ndscons を用いてバックリンカをローカルサーバにスケジュールできます。詳細については、OpenText iMonitor のオンラインヘルプまたは ndstrace のマニュアルページを参照してください。

セキュリティオブジェクトのキャッシュ機能は、デフォルトで有効になっています。バックリンカによってデータをキャッシュしない場合は、NCP サーバオブジェクトから CachedAttrsOnExtRef を削除します。

NMAS ソフトウェア

NMAS は、OpenText eDirectory のバンドル製品として同梱されています。ソフトウェアイメージには以下が含まれます。

- NMAS サーバソフトウェア
- ログインメソッドソフトウェア
- ログインシーケンスごとの複数のログインメソッドのサポート

- 段階的認証のサポート
- ユニバーサルパスワード

NMAS クライアントソフトウェアは、OpenText Client for Windows と OpenText SecureLogin で使用できます。

- 672 ページの 「サーバソフトウェアとクライアントソフトウェアのインストール」
- 672 ページの 「ログインメソッドソフトウェアとパートナー」
- 673 ページの 「ユニバーサルパスワード」
- 673 ページの 「OpenText Identity Console での管理」

サーバソフトウェアとクライアントソフトウェアのインストール

NMAS サーバ側ソフトウェアは、デフォルトで、OpenText eDirectory と一緒にインストールされます。NMAS クライアント側ソフトウェアは、NMAS ログインメソッドを使用してネットワークにアクセスするすべてのクライアントワークステーションにインストールする必要があります。インストール後は、OpenText Identity Console の［**認証管理**］タイルを使用して NMAS を管理できます。

現在、NMAS クライアントソフトウェアは OpenText クライアントに同梱されています。詳細については、OpenText Client for Windows のマニュアルを参照してください。

インストール中に、NMAS は、OpenText eDirectory スキーマを拡張して、OpenText eDirectory ツリー内のセキュリティコンテナに新しいオブジェクトを作成します。これらの新しいオブジェクトとは、承認済みログインメソッドコンテナ、承認済みポストログインメソッドコンテナ、セキュリティポリシーオブジェクト、およびログインポリシーオブジェクトです。ログインメソッドはすべて承認済みログインメソッドコンテナに保存され、管理されます。ポストログインメソッドはすべて承認済みポストログインメソッドコンテナに保存され、管理されます。

ログインメソッドソフトウェアとパートナー

- 672 ページの 「ソフトウェアとパートナー」
- 673 ページの 「ログインメソッドのインストール」

ソフトウェアとパートナー

現在サポートされているいくつかのログインメソッドが、NMAS ソフトウェアイメージ上に含まれています。

NMAS ソフトウェアには、サードパーティの認証開発者による複数のログインメソッドのサポートが含まれています。OpenText パートナーの一覧については、OpenText パートナー Web サイト (https://www.opentext.com/partners/find-an-opentext-partner) を参照してください。

NMAS 用のログインメソッドを開発しているパートナーは、それぞれ、ユニークな製品の機能と特性を活かしてネットワーク認証を解決しています。そのため、実際のセキュリティ特性はログインメソッドごとに異なります。

OpenText では、これらのパートナー製品のセキュリティ手法を評価していません。したがって、これらの製品に OpenText Yes、Tested & Approved、または OpenText eDirectory Enabled ロゴの表示が認められていても、それらのロゴは一般的な製品の相互運用性を示しているにすぎません。

自社のセキュリティニーズに最もよく合った製品を見極めるためにも、各パートナーの製品の機能を慎重に検討することをお勧めします。また、ログインメソッドによっては、NMAS 製品に同梱されていないハードウェアやソフトウェアが別途必要な場合があることに注意してください。

ログインメソッドのインストール

NMAS ログインメソッドは、次を使用してインストールできます。

- nmasinst (すべての OpenText eDirectory プラットフォームで利用可能)。これを使用するには OpenText eDirectory をインストールする必要があります。
- OpenText Identity Console の［**認証管理**］タイル。

ログインメソッドのインストール方法については、674 ページの 「ログインメソッドのインストール方法」を参照してください。

ユニバーサルパスワード

ユニバーサルパスワードは、各種のパスワードと認証のシステムの統合と管理を簡略化して 1 つの首尾一貫したネットワークを構築するための手段です。これを使用すると、OpenText eDirectory へのすべてのアクセスに対して 1 つのパスワードが提供され、パスワードでの拡張文字の使用、詳細パスワードポリシーの適用、および OpenText eDirectory から他のシステムへのパスワードの同期が可能となります。

ユニバーサルパスワードの詳細については、785 ページの第 26 章「パスワードを管理する」を参照してください。

OpenText Identity Console での管理

NMAS の管理は、OpenText Identity Console の［**認証管理**］タイルを使用して行えます。これは、OpenText eDirectory を管理するための Web ベースのユーティリティです。OpenText Identity Console 内の特定のプロパティページで、ログインメソッド、ログインシーケンス、登録、および段階的認証を管理できます。

デフォルトで、NMAS は標準の NDS パスワードログインメソッドをインストールします。追加のログインメソッドは、OpenText Identity Console の［認証管理］を使用するか、［新規オブジェクトの作成 (Create New Object)］オプションを通して承認済みログインメソッドコンテナから起動されるウィザードを使用してインストールできます。ポストログインメソッドは、新しいオブジェクトの作成オプションを通して承認済みポストログインメソッドコンテナから起動されるウィザードを使用してインストールできます。

ログインメソッドのインストール方法については、674 ページの 「ログインメソッドのインストール方法」を参照してください。

ログインとポストログインのメソッドとシーケンスの管理

このセクションでは、NMAS 用のログインとポストログインのメソッドとシーケンスのインストール、セットアップ、および設定方法について説明します。

NMAS には、3 つのログインファクタ (パスワード、物理デバイス、バイオメトリック認証) に基づく複数のログインメソッドの選択肢があります。

NMAS には、OpenText とサードパーティ認証開発者による複数のログインメソッドとポストログインメソッドのサポートが含まれます。メソッドによっては、追加のハードウェアやソフトウェアが必要な場合があります。使用するメソッドに必要なすべてのハードウェアとソフトウェアが揃っていることを確認してください。

NMAS では、ソフトウェアビルド内にいくつかのログインメソッドが同梱されています。その他のログインメソッドはサードパーティベンダから入手できます。

OpenText eDirectory パートナーの一覧については、OpenText パートナー Web サイトを参照してください。いくつかのパートナーがサードパーティログインメソッドを開発しています。

- 674 ページの 「ログインメソッドのインストール方法」
- 676 ページの 「ログインメソッドとポストログインメソッドの更新」
- 676 ページの 「ログインシーケンスの管理」
- 679 ページの 「ユーザに対するログインシーケンスの承認」
- 680 ページの 「デフォルトログインシーケンスの設定」
- 680 ページの 「ログインメソッドの削除」
- 681 ページの 「ログインシーケンスの削除」

ログインメソッドのインストール方法

OpenText eDirectory で使用するログインメソッドのインストール方法は 3 種類あります。

- nmasinst ユーティリティ (Linux と Windows)。OpenText eDirectory にログインメソッドをインストールできます。
- OpenText Identity Console (Linux と Windows)。OpenText eDirectory にログインメソッドとポストログインメソッドをインストールできます。

- 675 ページの 「nmasinst ユーティリティを使用したログインメソッドのインストール」
- 675 ページの「OpenText Identity Console を使用したログインメソッドまたはポストログインメソッドのインストール」

nmasinst ユーティリティを使用したログインメソッドのインストール

サーバコンソールのコマンドラインから次のように入力します。

nmasinst -addmethod admin.context treename config.txt_path [-h hostname[:port]] [-w password|file:<filename>|env:<environment_variable>] [-checkversion] [-d]

- *admin.context*: 管理者の名前とコンテキスト。
- *treename*: ログインメソッドをインストールする OpenText eDirectory ツリーの名前。
- *config.txt_ path* - ログインメソッドの config.txt ファイルへの完全パスまたは相対パス。config.txt ファイルは各ログインメソッドに付属しています。
- [-h *hostname*[:*port*]]: (オプション) サーバのホスト名とポート。OpenText eDirectory がデフォルトポートで実行していない場合に使用します。IP アドレスを指定することもできます。OpenText eDirectory 9.2 では IPv4 アドレスと IPv6 アドレスの両方がサポートされています。次に例を示します。
  - **IPv4:** -h 127.0.0.1:8443
  - **IPv6:** -h [2001:db8::6]:8443
- [-w *password*|file:<*filename*>|env:<*environment_variable*>]: このオプションを使用すれば、次の方法のいずれかを使用してパスワードを指定することができます。
  - コマンドラインから。例 : -w n
  - ファイルを通して。例 : -w file:/tmp/passwd
  - 環境変数を通して。例 : -w env:PASSWD
- [-checkversion]: このオプションは、インストールするメソッドバージョンがインストール済みのメソッドバージョンと同じかそれよりも新しい場合にエラーを報告します。
- [-d]: サポートされていないプラットフォーム用のメソッドを削除します。

ログインメソッドがすでに存在した場合は、nmasinst がそれを更新します。

OpenText Identity Console を使用したログインメソッドまたはポストログインメソッドのインストール

1. OpenText Identity Console を起動します。
2. 管理者または管理権を持つユーザとして、OpenText eDirectory ツリーに対する認証を受けます。
3. OpenText Identity Console のホームページで、[**認証管理**] > [**ログインメソッドとシーケンス**] の順に選択します。
4. [**ログインメソッド**] タブをクリックします。
5. [**ログインメソッドのインストール (install Login methods)** +] をクリックします。

6 インストールするログインメソッド (.zip) ファイルをブラウズして選択してから、[次へ] をクリックします。

7 インストールウィザードに従って完了します。

ログインメソッドとポストログインメソッドの更新

ログインメソッドベンダがログインメソッドまたはポストログインメソッドの更新を提供にしている場合は、次の手順を実行してメソッドを更新することができます。

- 676 ページの 「nmasinst ユーティリティを使用したログインメソッドの更新」
- 676 ページの 「OpenText Identity Console を使用したログインメソッドの更新」

nmasinst ユーティリティを使用したログインメソッドの更新

nmasinst ユーティリティを使用してログインメソッドをインストールする手順と同じ手順を使用します (676 ページの 「nmasinst ユーティリティを使用したログインメソッドの更新」を参照)。新しい config.txt ファイルへのパスを追加すると、ログインメソッドが更新されます。

OpenText Identity Console を使用したログインメソッドの更新

1 OpenText Identity Console を起動します。

2 管理者または管理権を持つユーザとして、OpenText eDirectory ツリーに対する認証を受けます。

3 [**認証管理**] タイル > [**ログインメソッド**] の順に選択します。

4 更新するログインメソッドを選択します。

5 ログインメソッドのプロパティページで、[**Update Method**] をクリックします。

6 更新ウィザードに従って完了します。

ログインシーケンスの管理

ログインをインストールするときに、インストールするログインメソッドのみを使用するログインシーケンスを作成するかどうかが尋ねられます。[はい] を選択すると、1 つのログインメソッドだけを含むログインシーケンスが作成されます。

ログインシーケンスは手動で作成して管理することもできます。ログインメソッドとポストログインメソッドをインストールしたら、OpenText Identity Console を使用してログインシーケンスを表示、追加、変更、または削除することができます。メソッドが変更または更新されても、ログインシーケンスは作成されません。

NMAS では、シーケンスごとに複数のログインメソッドとポストログインメソッドをセットアップすることができます。ポストログインメソッドを選択するためには、1 つ以上のログインメソッドを選択しておく必要があります。

1 つのシーケンスに対して複数のメソッドが選択されている場合は、それらが表示順に実行されます。ログインメソッドが先に実行され、その後でポストログインメソッドが実行されます。

ログインシーケンスは、AND シーケンスまたは OR シーケンスにすることができます。AND シーケンスは、すべてのログインメソッドによってユーザの ID が正常に検証された場合に成功します。OR シーケンスの場合は、いずれかのログインメソッドによってユーザの ID が検証されれば、ログインが成功します。

ポストログインメソッドは、And/Or 関係とは無関係に、ログインが成功した場合にのみ実行されます。

シーケンスを作成したら、その新しいシーケンスを使用した OpenText eDirectory へのログインをユーザに許可することができます。

- 677 ページの 「OpenText Identity Console を使用した新しいログインシーケンスの作成」
- 678 ページの 「ログインシーケンスの変更」
- 678 ページの 「ログインシーケンスの削除」

OpenText Identity Console を使用した新しいログインシーケンスの作成

1 OpenText Identity Console を起動します。
2 管理者または管理権を持つユーザとして、OpenText eDirectory ツリーに対する認証を受けます。
3 [**認証管理**] > [**ログインメソッドシーケンス**] の順に選択します。
4 [**作成** ＋] をクリックして、新しいログインシーケンスの名前を指定します。

すべての使用可能なメソッドが、[**Available Login Methods**] と [**Available Post-Login Methods**] の下に一覧表示されます。
5 リストから [**シーケンスタイプ**] を選択します。

[*And*] を選択する場合、ユーザは、ログインシーケンスに含まれるすべてのログインメソッドを使用してログインする必要があります。[*Or*] を選択する場合、ユーザは、ログインシーケンスに含まれるログインメソッドのいずれか 1 つを使用するだけでログインできます。
6 矢印を使用して必要なそれぞれのメソッドをシーケンスに追加します。

複数のメソッドを使用する場合、実行順序を変更するには垂直矢印を使用します。

[**Sequence Grade**] フィールドに、ログインシーケンスのグレードが表示されます。And シーケンスの場合、シーケンスグレードは、一連のログインメソッドのグレードの和集合になります。Or シーケンスの場合、シーケンスグレードは、一連のメソッドのグレードの共通部分になります。
7 [**作成**] をクリックしてログインシーケンスを保存します。

ログインシーケンスの変更

1 OpenText Identity Console を起動します。

2 管理者または管理権を持つユーザとして、OpenText eDirectory ツリーに対する認証を受けます。

3 [**認証管理**] タイル > [**ログインメソッドシーケンス**] の順に移動します。

4 [**ログインメソッドシーケンス**] をクリックします。シーケンスについて、[**名前**]、[**グレード**]、[**承認済み**]、[**デフォルト**] のリストが表示され、[**ログインメソッド**] および [**ポストログインメソッド**] が一覧表示されます。

5 アクションの選択

5a [＋] をクリックして、ログインメソッドシーケンスを作成します。作成する [**名前**] および [**シーケンスタイプ**] ドロップダウン値を指定します。使用可能なすべてのメソッドが、[**使用可能な > ログインメソッド**] リストと [**使用可能なポストログインメソッド**] リストに表示されます。

5b [ ] をクリックして、既存のログインメソッドシーケンスを変更します。

注 : ポストログインメソッドを選択するためには、1 つ以上のログインメソッドを選択しておく必要があります。

---

6 [**ログインメソッド**] のシーケンス順序を変更するには、上矢印と下矢印を使用します。

7 [**シーケンスタイプ**] を変更するには、シーケンスタイプの横にあるドロップダウンリストを使用します。

8 [**保存**] をクリックするか、変更を保存しない場合は、[**キャンセル**]、[**終了 (exit)**] の順にクリックします。

重要 : 関連付けられたメソッドが存在しないログインシーケンスは保存されません。

---

9 [**OK**] をクリックします。

ログインシーケンスの削除

1 OpenText Identity Console を起動します。

2 管理者または管理権を持つユーザとして、OpenText eDirectory ツリーに対する認証を受けます。

3 [**認証管理**] > [**ログインメソッドシーケンス**] の順に移動します。

4 削除する [**ログインシーケンス**] を選択し、[**削除**] をクリックしてログインメソッドシーケンスを削除します。

1 つ以上のログインシーケンスでそのログインメソッドが引き続き使用されている場合は、警告メッセージが表示されます。

5 [**OK**] をクリックします。

ユーザに対するログインシーケンスの承認

- 679 ページの 「ログインシーケンスの割り当て」
- 679 ページの 「ログインシーケンスの承認」

ログインシーケンスの割り当て

承認済みログインシーケンスとデフォルトログインシーケンスを、ユーザ、コンテナ、パーティションルート、またはログインポリシーオブジェクトに割り当てることができます。NMAS は、ユーザオブジェクト、ユーザオブジェクトのコンテナ、ユーザオブジェクトのパーティションルート、およびログインポリシーオブジェクトの順にそれらの属性を読み込むことによって、ユーザの承認済みログインシーケンスまたはデフォルトログインシーケンスを検索します。

ユーザオブジェクトで見つかった属性は、コンテナ、パーティションルート、またはログインポリシーオブジェクトで見つかった属性よりも優先されます。ログインシーケンスがパーティションルートに割り当てられている場合は、そのログインシーケンスがそのパーティションルートの下のすべてのユーザに適用されます。ただしそれは、特定のユーザに個別にログインシーケンスが割り当てられていない場合に限られます。

また、コンテナに割り当てられたログインシーケンスは、そのコンテナ内のシーケンスが割り当てられていないユーザにのみ適用され、そのコンテナのサブコンテナ内のユーザには適用されません。

ログインシーケンスの承認

1 OpenText Identity Console を起動します。

2 管理者または管理権を持つユーザとして、OpenText eDirectory ツリーに対する認証を受けます。

3 [**認証管理**] タイル > [**ログインメソッドシーケンス**] の順に移動します。

4 [**ログインシーケンス**] を選択し、[ログインメソッドシーケンスの承認] ( ) をクリックして、選択したログインメソッドシーケンスを承認するか、[不正なログインメソッドシーケンス] ( ) をクリックして承認を拒否します。または、下矢印をクリックしてシーケンスを承認または承認拒否し、上下矢印をクリックしてシーケンスリストを並べ替えます。

[**デフォルト**] リストでは、承認されたログインシーケンスメソッドのみを [デフォルト] に設定できます。トグルアイコン ( ) を使用してこの操作を実行できます。

デフォルトログインシーケンスの設定

ユーザがログイン時にログインシーケンスを指定する必要がないようにデフォルトログインシーケンスを設定するには :

1 OpenText Identity Console を起動します。

2 管理者または管理権を持つユーザとして、OpenText eDirectory ツリーに対する認証を受けます。

3 [**認証管理**] タイル > [**ログインメソッドシーケンス**] の順に移動します。

4 承認されたログインシーケンスを [**デフォルト**] リストで選択し、[] をクリックします。

選択したシーケンスがデフォルトログインシーケンスになります。ユーザがログインシーケンスを使用せずにログインしようとすると、このデフォルトログインシーケンスが使用されます。

注 : ワークステーションがユーザのデフォルトログインシーケンスを実行できない場合は、NDS パスワードログインメソッドが使用されます。

ログインシーケンスの割り当て方法については、679 ページの 「ログインシーケンスの割り当て」を参照してください。

ログインメソッドの削除

OpenText Identity Console の NMAS では、ログインシーケンスの一部になっているログインメソッドを削除できません。ログインメソッドのデフォルトインストールでは、そのメソッドのみを含むログインシーケンスが作成されます。その結果、ほとんどのメソッドが 1 つ以上のシーケンスに組み込まれます。

注 : nmasinst には、NMAS メソッドを削除するオプションがありません。これは、OpenText Identity Console を使用して行う必要があります。

ログインメソッドを削除するには、次の 2 つの手順を実行する必要があります。

- 681 ページの 「すべてのログインシーケンスからログインメソッドを削除する」
- 681 ページの 「ログインメソッドを削除する」

すべてのログインシーケンスからログインメソッドを削除する

OpenText Identity Console を使用して任意のログインシーケンスのログインメソッドを削除するには :

1 OpenText Identity Console のホームページで、[**認証管理**] > [**ログインメソッドシーケンス**] の順にクリックします。

2 [**ログインメソッドシーケンス**] リスト内のシーケンスごとに :

2a シーケンス名をクリックします。

2b 削除するログインメソッドが [**ログインメソッド**] リストにも [**Post-Login Methods**] リストにも列挙されていないことを確認します。

2c 対象のログインメソッドが、選択されたメソッドの 1 つとして列挙されている場合は、それを選択して左矢印をクリックすることによってリストから移動することができます。

すべてのログインシーケンスからログインメソッドを削除したら、そのメソッドを削除することができます。詳細については、681 ページの 「ログインメソッドを削除する」を参照してください。

ログインメソッドを削除する

OpenText Identity Console を使用してログインメソッドを削除するには :

1 OpenText Identity Console のホームページで、[**認証管理**] > [**ログインメソッド**] の順にクリックします。

2 削除するログインメソッドを選択します (複数選択可)。

3 **削除**をクリックしてから **OK** をクリックします。

ログインシーケンスの削除

1 OpenText Identity Console を起動します。

2 管理者または管理権を持つユーザとして、OpenText eDirectory ツリーに対する認証を受けます。

3 [**認証管理**] > [**ログインメソッドシーケンス**] の順に移動します。

4 削除する [**ログインシーケンス**] を選択し、[**削除** 🗑] をクリックしてログインメソッドシーケンスを削除します。

1 つ以上のログインシーケンスでそのログインメソッドが引き続き使用されている場合は、警告メッセージが表示されます。

5 [**OK**] をクリックします。

NMAS を使用したネットワークへのログイン

NMAS をインストールしたら、ユーザがネットワークにログインする準備が整ったことになります。このセクションでは、ネットワークユーザに周知しておく必要があるログインエクスペリエンスの一部の追加機能について説明します。

- 682 ページの 「パスワードフィールド」
- 682 ページの 「カスタムログイン」
- 683 ページの 「ワークステーションのロック解除」
- 683 ページの 「NMAS クライアントトレースの収集」
- 683 ページの 「NMAS クリアランスステータスの表示」

パスワードフィールド

NMAS クライアントソフトウェアのインストール方法に応じて、Novell Client のログインダイアログボックスにパスワードフィールドが表示される場合と表示されない場合があります。ユーザがバイオメトリックまたは物理デバイス () ログインファクタを使用している場合は、ネットワークにログインするためのパスワードが不要な場合があります。

パスワードフィールドの非表示に関する詳細については、Novell Client For Windows のマニュアル (http://www.novell.com/documentation/windows_client/) を参照してください。

カスタムログイン

NMAS ログインメソッドを使用してネットワークにログインしているユーザは、必要なクリアランスとログインシーケンスを選択することによって、ログインをカスタマイズすることができます。そうしない場合は、最後のログインシーケンスとクリアランス (もしあれば) が使用されます。クリアランスまたはログインシーケンスが事前に指定されていない場合は、デフォルトが使用されます。

1. Novell Client のダイアログボックスが表示されたら、[詳細] をクリックします。
2. [NMAS] タブをクリックします。
3. [ログイン] ドロップダウンリストから目的のログインシーケンスを選択するか、OpenText eDirectory ツリーで最新の完全なリストをブラウズします。

   ブラウズできるのは、[OpenText eDirectory] タブで OpenText eDirectory ツリーが指定されている場合だけです。
4. 必要なユーザセッションクリアランスを指定するか、OpenText eDirectory ツリーで最新の完全なリストをブラウズします。

   デフォルトで、[クリアランス] フィールドは無効になっています。[クリアランス] フィールドを有効にするには :

   4a タスクバーで赤色の N を右クリックします。

   4b [Novell Client Properties] > [Location Profiles] の順にクリックします。

   4c 必要なプロファイルを選択して、[プロパティ] をクリックしてから、[プロパティ] をクリックします。

4d［NMAS］タブで、［Display Clearance Field］を選択します。

4e［OK］を 3 回クリックします。

重要 : ユーザには、ログインシーケンスごとに複数のセッションクリアランスを割り当てることができます。［クリアランス］フィールドに、必要なユーザセッションクリアランスが入力されていることを確認します。

5 OK をクリックします。

ワークステーションのロック解除

ユーザのワークステーションに NMAS を追加すると、Windows ワークステーションをロック解除するプロセスが変更されます。通常、ユーザは Windows の［ディスプレイ］コントロール パネルで設定するスクリーンセーバを使用して各自のワークステーションのパスワード保護を有効にできます。NMAS を使用してワークステーションをロック解除するには、ユーザが代わりに最初のログインに使用したのと同じ認証プロセスを通過する必要があります。

たとえば、NMAS を使用してネットワークに対する認証を受けてバイオメトリックログインメソッドを使用した場合、ワークステーションのロックを解除して使用するには、再度同じバイオメトリックログインメソッドを使用する必要があります。

Windows ワークステーションを使用している場合は、ツリーへのログインに使用したログインメソッドを使用してワークステーションをロック解除する必要があります。複数の OpenText eDirectory ツリーに接続している場合は、いずれかの OpenText eDirectory ツリー用のログインシーケンスを使用できます。デフォルトは最初の OpenText eDirectory ツリーです。

NMAS クライアントトレースの収集

NMAS クライアントトレースを収集すれば、NMAS 認証の問題のトラブルシューティングに役立ちます。詳細については、「TID # 3331372」を参照してください。

NMAS クリアランスステータスの表示

1 タスクバーで赤色の N を右クリックします。

2［Novell Connections］をクリックします。

3 スクロールオーバーすると、各接続に関連付けられた NMAS クリアランスが表示されます。

OpenText パスワードの履歴

これまでは、パスワードの制約事項のために、管理者は複数のパスワード (単純パスワード、NDS パスワード、拡張パスワード) を管理する必要がありました。さらに、複数のパスワードを同期させる必要もありました。

- NDS パスワード : 古い NDS パスワードは非可逆のハッシュ形式で保存されます。このパスワードは NDS システム専用であり、他の任意のシステムで使える別のフォームに変換することはできません。
- 単純パスワード : 単純パスワードは、元々、管理者が Active Directory* や iPlanet* などの外部の LDAP ディレクトリからユーザとパスワード (平文とハッシュ済み) をインポートできるようにするために実装されたものです。

  シンプルパスワードの制約は、パスワードポリシー (最小長、満了日など) がいっさい施行されないことです。
- 拡張パスワード : ユニバーサルパスワードの前身である拡張パスワード (現在は未サポート) は、一部のパスワードポリシーを提供しますが、その設計には他のパスワードとの一貫性がありません。このパスワードは、片方向同期を提供し、単純パスワードや NDS パスワードに取って代わるパスワードです。

これらのパスワードの問題を解決するためにユニバーサルパスワードが作成されました。このパスワードには次のような特徴があります。

- OpenText eDirectory へのすべてのアクセスを 1 つのパスワードで行えます。
- パスワード内で拡張文字を使用することができます。
- 詳細なパスワードポリシーを適用できます。
- OpenText eDirectory から他のシステムへパスワードを同期できます。

ユニバーサルパスワードは、NMAS モジュールのコンポーネントである Secure Password Manager によって管理されます。セキュアパスワードマネージャを使用すると、OpenText、Novell、および OpenText パートナーのさまざまな製品におけるパスワードベースの認証スキーマの管理を簡略化できます。この管理ツールは、1 つのパスワードのみを表示し、下位互換性を維持するためのすべてのバックグラウンドの処理は表示しません。

セキュアパスワードマネージャとユニバーサルパスワードを管理または利用する他のコンポーネントは、OpenText eDirectory インストールの一部としてインストールされます。ただし、ユニバーサルパスワードはデフォルトで有効になっていません。認証とパスワード設定用のすべての API がユニバーサルパスワードをサポートするように移行されるので、既存のすべての管理ツールは、これらの新しいライブラリを備えたクライアント上で実行すると、自動的にユニバーサルパスワードと連動します。

注 : [パスワードポリシー] タイルは、[認証管理] から使用できます。

Novell Client はユニバーサルパスワードをサポートします。また、ネットワーク内の既存システムのために、引き続き NDS パスワードもサポートしています。Novell Client は、最初のログイン時に自動的に NDS パスワードをユニバーサルパスワードに移行する機能を備えています。

「既存のパスワードがパスワードポリシーに準拠しているかどうかを確認する (ログイン時に検証が実行される)」パスワードポリシールールが true に設定されていない場合は、NDS パスワードがユニバーサルパスワードに移行されるときにパスワード期限の時刻が更新されません。

ユニバーサルパスワードの展開と管理に関する詳細については、785 ページの第 26 章「パスワードを管理する」を参照してください。

NMAS HOTP ベースのログイン

次の各セクションでは、NMAS HOTP に関する情報を提供します。

- 685 ページの 「概要」
- 686 ページの 「インストール」
- 688 ページの 「カウンタの再同期」
- 689 ページの 「環境設定」
- 690 ページの 「当バージョンの注意事項」
- 691 ページの 「nmashotpconf ユーティリティは、ユーザ再同期ウィンドウを変更できません。」

概要

HOTP は、HMAC ベースのワンタイムパスワード (OTP) アルゴリズムです。OTP は、1 回のログインセッションまたはトランザクションでのみ有効なパスワードです。OTP に関連したセキュリティ攻撃の発生リスクは比較的小さいので、OTP は従来の (静的な) パスワードよりも優れたパフォーマンスを提供します。不正侵入者が、サービスへのログインやトランザクションの実行に使用された OTP を記録したとしても、その OTP はすでに 1 回使用されていて有効ではなくなっているので、その OTP を使用することはできません。すべての OTP ベースの認証には、OTP サーバと OTP クライアント (ハードウェア / ソフトウェア) が必要です。NMAS の OTP ベースの認証の実装は、RFC 4226 標準に基づきます。これまで個別にサーバに提示されていた NDS パスワードが、OTP に付加されるようになり、すべてのクライアントコンポーネントとそのユーザインタフェースを維持したままでパスワードベースの認証が拡張されました。OpenText eDirectory サーバに対する認証は、LDAP ベースのログインを使用することによって、HOTP 機能を通して実行されます。

LDAP ベースのログイン

前提条件

- NDSD_TRY_NMASLOGIN_FIRST 環境変数が true に設定されていることを確認してください。

  詳細については、「パスワードを管理するを参照してください。

注 : これはデフォルトで、OpenText eDirectory 9.0 以降で設定されます。

ログイン方法

HOTP 対応ユーザは、NDS パスワードと HOTP 値を連結させることによって、LDAP バインドを実行できます。

次に例を示します。

```
ldapsearch -D cn=user1,o=novell -w secret40338314 -h 164.99.91.165 -p 389 -
b   "o=novell" -s sub -LLL dn
```

NCP ベースのログイン

HOTP 対応ユーザは、次のユーティリティのいずれかで NDS パスワードと HOTP 値を連結させることによって、NCP ログインを実行できます。

- ndslogin

  次に例を示します。

  ```
  ndslogin user1.org -h org.com -p secret40338314
  ```

- OpenText Identity Console
- iMonitor

注 : LDAP 認証を実行する OpenText Identity Console は、HOTP 対応ユーザが使用するとエラーになります。

インストール

- 686 ページの 「サーバのインストール」
- 687 ページの 「nmashotpconf ユーティリティの取得および使用方法」

サーバのインストール

HOTP サーバモジュールは、NMAS サーバコンポーネントの一部です。このサーバモジュールがクライアントから提示された OTP を検証します。

次の属性を NMAS HOTP サーバ上で使用できます。

- sasOTPCounter (ユーザ属性ごと)
- sasOTPEnabled (ユーザ/直接のペアレントコンテナ/パーティションルート/ログインポリシーオブジェクトごと)
- sasOTPDigits (ユーザ / 直接のペアレントコンテナ / パーティションルート / ログインポリシーオブジェクトごと)
- asOTPLookAheadWindow (ログインポリシーオブジェクトでツリー全体に対して設定されます)
- sasOTPResync (ユーザ属性あたり 9)

nmashotpconf ユーティリティの取得および使用方法

nmashotpconf ユーティリティは、eDirectory サーバ上で OTP 属性を設定する環境設定ユーティリティです。

注 : HOTP ユーティリティは、Linux 64 ビットプラットフォームでのみ使用できます。

nashotpconf ユーティリティを実行するには、次の手順を実行します。

1 nmashotpconf ユーティリティを入手し、NMAS HOTP ユーティリティを解凍したディレクトリを指定します。

注 : nnmashotpconf ユーティリティは、NMAS にバンドルされています。このユーティリティをダウンロードするには、https://sld.microfocus.com を参照してください。

解凍されたファイルには、32 ビットと 64 ビットの Linux マシン用の linux ディレクトリと linux_x64 ディレクトリが含まれています。

Linux ディレクトリと linux_x64 ディレクトリには、nmashotpconf 実行可能ファイルと libnmasext.so ファイルが含まれています。

2 Linux 32 ビットマシン上では linux/final ディレクトリに移動し、Linux 64 ビットマシン上では linux_x64/final ディレクトリに移動します。

3 ルート認証局証明書をダウンロードして、ローカルに保存します。

詳細については、736 ページの 「ルート認証局または公開鍵証明書のエクスポート」を参照してください。

使用するには :

```
 nmashotpconf -h <host_name> [-p <ssl_port>] -D <login_dn> [-w
<password>]
  -e <trusted_cert> -t <cert_type> [-r <resync_window>] [-y
  <user_resync_window>] [-u <hotp_dn> [-o <hotp_options>] [-d digits]
[-c
  <counter>] [-s <secret> -f <secret_format>]]
```

オプション	説明
host_name	LDAP サーバ名またはサーバの IP アドレスを指定します。
ssl_port	LDAP サーバ上の SSL ポートを指定します。デフォルトは 636 です。
login_dn	ユーザの DN を指定します。
パスワード	ユーザ DN のパスワードを指定します。
trusted_cert	ルート認証局証明書ファイルを指定します。
cert_type	ルート認証局証明書のエンコードタイプを指定します。たとえば、DER は der エンコードファイルを意味し、B64 は b64 エンコードファイルを意味します。

オプション	説明
encoded file digits	HOTP 値として使用する桁数を指定します。 **注 :** この設定がツリー内のすべてのユーザに適用されます。
resync_window	カウンタ再同期先読みウィンドウを指定します。
user_resync_window	カウンタユーザ再同期先読みウィンドウを指定します。
hotp_dn	HOTP 属性を設定しているターゲット DN を指定します。ツリーレベルで HOTP を設定するには、ツリーレベルで HOTP を有効 / 無効にするか、ツリーレベルで［桁数］を設定してから、「cn=Login Policy,cn=Security」として DN を指定します。
hotp_options	hotp_dn オプションに対して HOTP を有効または無効にします。HOTP を有効にするには ENABLE を指定し、HOTP を無効にするには DISABLE を指定します。
counter	HOTP カウンタ値を指定します。カウンタ値の有効範囲は 0 ～ 2147483647 です。カウンタ値は hotp_dn オプションを通して設定されます。
hotp_dn secret	OATH HOTP シークレットを指定します。たとえば、16 進形式のシークレットの未加工のバイト値は 31323334353637383930313233343536373839 30 で、対応する ASCII/ 拡張 ASCII 文字列は 12345678901234567890 です。
secret_format	OATH HOTP シークレットの形式を指定します。 ◆ **STRING:** この形式は、ASCII/ 拡張 ASCII 文字列に使用されます。たとえば、「12345678901234567890」と指定します。 ◆ **RAW:** この形式は、16 進形式の未加工のバイト値に使用されます。たとえば、31323334353637383930313233343536373839 30 の場合は、31 が最初の文字の 16 進値で、32 が 2 つ目の文字の値になります。

カウンタの再同期

サーバのカウンタ値は HOTP 認証に成功するとインクリメントされ、クライアントのカウンタは新しい HOTP がユーザから要求されるたびにインクリメントされます。サーバ上のカウンタ値とクライアント上のカウンタは同期していない場合があります。

これを解決するには、ツリー全体の先読みまたは再同期ウィンドウの設定を実施する必要があります。サーバで受信された HOTP がサーバのカウンタ値に対応していないことが検出された場合は、サーバが再同期ウィンドウ内の次のいくつかの HOTP 値を再計算して、それらを受信された HOTP に照らしてチェックすることができます。一致が確認された場合は、認証が成功して、一致した HOTP に対応するカウンタ値が、サーバカウンタとして設定されます。

認証を成功させるために、サーバカウンタは認証が成功する次のカウンタ値に設定されます。

ツリー全体の再同期ウィンドウの設定は、できるだけ小さくして、攻撃者が HOTP 値の再作成を試みる余地を制限する必要があります。クライアントとサーバのカウンタの誤差がツリー全体の再同期ウィンドウの設定を超えている場合は、一時的にユーザ固有の再同期ウィンドウを大きい値に設定して HOTP ベースの認証を試みることにより再同期することができます。

HOTP ベースの認証を設定するには、nmashotpconf ユーティリティを使用する必要があります。詳細については、環境設定のセクションを参照してください。

環境設定

HOTP ベースの認証用に OpenText eDirectory ユーザをプロビジョニングするには、RFC 4226 標準に従って、次の設定手順を実行します。

- ユーザ/コンテナ/パーティションルート/ログインポリシーオブジェクト(この順序は優先順位と同じになります) に対して HOTP を有効にします。
- そのユーザに対して HOTP 共有秘密鍵とカウンタを設定します。この 2 つの設定によって、HOTP 値が決定されます。
- ユーザ / コンテナ / パーティションルート / ログインポリシーオブジェクトに対して HOTP 値の桁数を設定します。有効な桁数の範囲は 6 ～ 9 です。
- 再同期ウィンドウを次のように設定します。
  - ログインポリシーオブジェクトでツリー全体の再同期ウィンドウを設定します。
  - ユーザレベルでユーザ固有の再同期ウィンドウを設定します。これは、クライアントとサーバが同期していない場合にのみ必要です。

例 :

- ユーザオブジェクトに対してシークレットとカウンタを設定するには、次のコマンドを実行します。

  ```
  ./nmashotpconf -h 192.168.1.1 -p 636 -D cn=admin,o=novell -w novell   -
  e   /var/opt/novell/eDirectory/data/SSCert.der  -t  DER  -u
  cn=user1,o=novell  -c 0   -s   31323334353637383930313233343536373839 30
  -f  RAW
  ```

- ユーザオブジェクトに対して OTP を有効にするには、次のコマンドを実行します。

  ```
  ./nmashotpconf -h 192.168.1.1 -p 636 -D cn=admin,o=novell -w novell   -
  e   /var/opt/novell/eDirectory/data/SSCert.der  -t  DER  -u
  cn=user1,o=novell  -o ENABLE
  ```

- ユーザオブジェクトに対して OTP を無効にするには、次のコマンドを実行します。

  ```
  ./nmashotpconf -h 192.168.1.1 -p 636 -D cn=admin,o=novell -w novell   -
  e   /var/opt/novell/eDirectory/data/SSCert.der  -t  DER  -u
  cn=user1,o=novell  -o DISABLE
  ```

  同様に、コンテナ / パーティションまたはルート / ログインポリシーオブジェクトに対して OTP を有効または無効にすることができます。

- ユーザオブジェクトに対して OTP 桁を設定するには、次のコマンドを実行します。

  ```
  ./nmashotpconf -h 192.168.1.1 -p 636 -D cn=admin,o=novell -w novell   -
  e /var/opt/novell/eDirectory/data/SSCert.der  -t  DER  -u
  cn=user1,o=novell  -d 6
  ```

  同様に、ペアレントコンテナ / パーティションルート / ログインポリシーオブジェクトに対しても OTP 桁を設定できます。

- ユーザ再同期ウィンドウを設定するには、次のコマンドを実行します。

  ```
  ./nmashotpconf -h 192.168.1.1 -p 636 -D cn=admin,o=novell -w novell   -
  y 5 -e   /var/opt/novell/eDirectory/data/SSCert.der  -t  DER  -u
  cn=user1,o=novell
  ```

- カウンタ再同期先読みウィンドウを設定するには、次のコマンドを実行します。

  ```
  ./nmashotpconf -h 192.168.1.1 -p 636 -D cn=admin,o=novell -w novell
  -r 6
  ```

注 : 設定をテストするには、HOTP 規格に準拠しているハードウェアやソフトウェアによって生成された HOTP を使用できます。

当バージョンの注意事項

- 690 ページの 「ndsconfig add は、HOTP 対応管理者ユーザの場合は失敗します。」
- 690 ページの 「HOTP 対応ユーザとしての読み込み専用レプリカへのログインが失敗する」

ndsconfig add は、HOTP 対応管理者ユーザの場合は失敗します。

HOTP 対応ユーザの場合は、OTP 桁が認証に使用されます。ndsconfig ユーティリティは、以降の認証に同じ OTP 桁を使用するため、ndsconfig add が失敗します。同様に、ndsconfig upgrade も失敗します。

この問題を回避するため、ndsconfig add/upgrade を実行するユーザに対しては HOTP を有効にしないでください。

HOTP 対応ユーザとしての読み込み専用レプリカへのログインが失敗する

HOTP 対応ユーザとして読み込み専用レプリカに要求を送信することによって LDAP ログインを実行すると、LDAP チェーンが実行されません。読み込み専用レプリカは、実際のユーザが存在するサーバには要求を転送しません。レプリカは失敗し、不正なレプリカタイプのエラーが表示されます。

nmashotpconf ユーティリティは、ユーザ再同期ウィンドウを変更できません。

ユーザ再同期ウィンドウの値がすでに設定 (たとえば、2 に) されている場合、その値が nmashotpconf ユーティリティを使用して変更されると、次のエラーが表示されます。

```
ldap_modify_ext_s on HOTP DN failed: error code=19: Constraint violation
```

エラーの原因の 1 つは、ユーザ再同期値を変更するために、-o (OTP を有効または無効にするオプション)、-d (OTP 桁)、-c (otpcouter)、および -y (user_resync_window) オプションの組み合わせを使用していることです。

その他の管理タスク

このセクションでは、NMAS のその他の管理タスクについて説明します。

- 691 ページの 「ポリシーリフレッシュレートコマンドの使用」
- 692 ページの 「LoginInfo コマンドの使用」
- 695 ページの 「LDAP に対する NMAS ベースのログインの無効化」
- 695 ページの 「NMAS コマンドの呼び出し」
- 696 ページの 「失敗したログイン試行の遅延時間の設定」
- 696 ページの 「DSTrace の使用」
- 696 ページの 「NMAS クライアントの無効化とアンインストール」
- 697 ページの 「OpenText Audit と外部証明書の使用」

ポリシーリフレッシュレートコマンドの使用

ログイン試行のたびにではなく、スケジュールしたインターバルで、セキュリティコンテナに保存された NMAS ログインポリシーからキャッシュに保存された NMAS ログインポリシーをリフレッシュするように NMAS を設定することができます。この設定は NMAS ポリシーリフレッシュレートコマンドを使用してサーバごとに構成します。

注 : サーバは、起動時に 1 回だけセキュリティコンテナにアクセスしてポリシーをキャッシュに保存します。その後は、設定されたインターバルに基づいて、セキュリティコンテナにアクセスしてポリシーをリフレッシュすることを試みます。

ポリシーリフレッシュレートコマンドの構文は次のとおりです。

nmas RefreshRate *minutes*

ここで、*minutes* は、キャッシュに保存された NMAS ログインポリシーを更新する必要があるかどうかをチェックする間隔 (分) です。

各 NMAS サーバプラットフォームでポリシーリフレッシュレートコマンドを呼び出す方法については、695 ページの 「NMAS コマンドの呼び出し」を参照してください。

LoginInfo コマンドの使用

NMAS 3.2 以降では、LoginInfo <numb> コマンドを使用して、特定のユーザオブジェクトログイン属性の自動更新をオフにすることができます。属性を自動更新すると問題が発生する場合には、手動でこれを行うことができます。次のセクションでは、この機能についてさらに詳しく説明します。

- 692 ページの 「LDAP バインド用の NMAS ログイン」
- 692 ページの 「ユーザオブジェクトログイン属性を自動更新することによって発生する問題」
- 693 ページの 「LoginInfo コマンドを使用した属性の更新のタイミングの制御」
- 693 ページの 「sasUpdateLoginInfo 属性と sasUpdateLoginTimeInterval 属性の使用」

LDAP バインド用の NMAS ログイン

OpenText eDirectory 9.2 では、デフォルトで、LDAP バインドのために NMAS ログインが有効になっています。NMAS ログインが有効になっている場合は、OpenText eDirectory がユーザの認証後に自動的にユーザオブジェクトログイン属性を更新します。更新されるログイン属性のリスト (ただし完全に網羅するものではない) を以下に示します。

- ログイン時刻
- ネットワークアドレス
- 最終ログイン時刻

LDAP に対する NMAS ベースのログインを無効にするには、695 ページの 「LDAP に対する NMAS ベースのログインの無効化」を参照してください。

ユーザオブジェクトログイン属性を自動更新することによって発生する問題

ユーザオブジェクトログイン属性の自動更新は、次のような問題を引き起こす可能性があります。

- 高い使用率
- 無応答
- 特に LDAP 環境でビジー状態の認証サーバで見られるクライアントのタイムアウト

これらの問題が発生した場合は、ログイン属性の更新のタイミングを調整する必要があります。この方法については、693 ページの 「LoginInfo コマンドを使用した属性の更新のタイミングの制御」を参照してください。

LoginInfo コマンドを使用した属性の更新のタイミングの制御

ログイン属性の更新タイミングを制御するには、nmas LoginInfo <num> コマンドを実行します。

<num> の値は次のとおりです。

- **0 または off:** ログイン属性を更新しません。
- **1:** 不正侵入者の検出に必要な属性のみを更新します。
- **2:** 未使用のユーザパスワードポリシー属性を除く、すべてのログイン属性を更新します。
- **3 または on:** すべてのログイン属性を更新します。

NMAS サーバプラットフォームごとの LoginInfo コマンドの呼び出し方法については、695 ページの 「NMAS コマンドの呼び出し」を参照してください。

sasUpdateLoginInfo 属性と sasUpdateLoginTimeInterval 属性の使用

sasUpdateLoginInfo 属性は、LoginInfo 属性の更新を制御します。

sasUpdateLoginTimeInterval 属性は、指定されたインターバルでユーザのログイン時刻属性が更新されるように制御します。

SasUpdateLoginInfo 属性は、次の値に設定することができます。

- **0 または off:** ログイン属性を更新しません。
- **1:** 不正侵入者の検出に必要な属性のみを更新します。
- **2:** 未使用のユーザパスワードポリシー属性を除く、すべてのログイン属性を更新します。
- **3 または on:** すべてのログイン属性を更新します。

SasUpdateLoginTimeInterval 属性は、0 ～ 1440 分 (つまり、1 日) の値に設定することができます。

- 値が 0 の場合は、ログインに成功するたびにログイン時刻属性と最終ログイン時刻属性が更新されます。
- 値が 1 ～ 1440 分の場合は、指定されたインターバルでログイン時刻属性が更新されます。最終ログイン時刻属性は更新されません。

注 : ログイン時刻属性は、インターバル中に続いてログインに成功しても更新されません。ただし、インターバル中ログインに失敗してその後ログインに成功した場合は、ログイン時刻属性が更新されます。ログインの成功からインターバル時間がカウントされます。

sasUpdateLoginTimeInterval 属性は、sasUpdateLoginInfo 属性値が 2 または 3 に設定されている場合にのみ有効になります。

これらの属性は、以下のオブジェクトに対して設定できます。オブジェクトは優先順位順に列挙しています (ユーザの優先順位が最も高くなります)。

- User (ユーザ)

- ユーザのコンテナ
- パーティションルート
- ログインポリシー

ログインポリシーオブジェクトに対して sasUpdateLoginInfo と sasUpdateLoginTimeInterval を設定した場合は、その次のポリシーリフレッシュサイクルの後に設定が有効になります。ユーザ、コンテナ、パーティションルート、またはログインポリシーに対してこれらの属性が設定されていない場合は、後方互換性を維持するために、コマンドラインを使用してサーバ上で設定された値が使用されます。

eDirectory サーバ上で属性値を設定する例を以下に示します。

```
#cat nmas.config (The nmas.config file must be in the same directory as the
dib directory.)
nmas LoginInfo 2
nmas UpdateLoginTimeInterval 30
```

パーティションルートで属性値を設定するには :

1. ツリーに属性を追加するには、**Identity Console**、［**スキーマ管理**］タイル > ［**属性**］の順に移動して、［**属性の作成**］をクリックします。
2. 矢印を使用して必要な属性を［**使用可能なオプション属性**］リストから［**オプション属性**］リストに移動します。

パーティションルートで属性の値を設定するには、コマンドラインまたは ldif ファイルを使用して、ldapmodify コマンドと次のコマンドを実行します。

```
dn:T=< tree name>
changetype:modify
add:sasUpdateLoginTimeInterval
sasUpdateLoginTimeInterval:35

dn:T=< tree name>
changetype:modify
add:sasUpdateLoginInfo
sasUpdateLoginInfo: 2
```

ユーザ、コンテナ、およびログインポリシーオブジェクトの sasUpdateLoginInfo 属性値または sasUpdateLoginTimeInterval 属性値は、OpenText Identity Console または ldif ファイルを使用して編集できます。

例 :

```
#cat changesasUpdateLoginInfo.ldif
dn: cn=user1,o=org
change type: modify
replace: sasUpdateLoginInfo
sasUpdateLoginInfo: 1

#cat changesasUpdateLoginTimeInterval.ldif
dn: cn=user1,o=org
changetype: modify
replace: sasUpdateLoginTimeInterval
sasUpdateLoginTimeInterval: 60
```

この設定は、user1 のログイン時刻属性の更新を前回の更新から 60 分間無効にします。

OpenText Identity Console から sasUpdateLoginInfo 属性と sasUpdateLoginTimeInterval 属性を指定するには :

1. OpenText Identity Console で、[**オブジェクト管理**] タイルをクリックします。
2. [**タイプ**] ドロップダウンリストから、[ユーザ] を選択し、[**検索**] をクリックします。
3. ユーザを選択します。
4. [**その他**] タブで、[**その他の属性の追加** +] をクリックし、[**値がない属性 (Unvalued Attributes)**] リストから [**sasUpdateLoginTimeInterval**] を選択します。
5. 矢印ボタンを使用して、[**値がない属性**] リストから [**値がある属性**] リストに [**sasUpdateLoginTimeInterval**] を移動してから、[**適用**] をクリックします。

LDAP に対する NMAS ベースのログインの無効化

OpenText eDirectory 9.2 では、デフォルトで、NMAS ログインが有効になっています。NMAS ログインを無効にするには、NDSD_TRY_NMASLOGIN_FIRST を false に設定します。

Windows 上で LDAP に対する NMAS ベースのログインを無効にするには、[マイコンピュータ] を右クリックして、[プロパティ] を選択します。[詳細設定] タブの [環境変数] をクリックします。[システム環境変数] で、変数を追加して、その値を false に設定します。

注 : RHEL 7.x プラットフォームと SLES 12.x プラットフォームの /etc/opt/novell/eDirectory/conf ディレクトリに配置された env ファイルに OpenText eDirectory サービスに必要なすべての環境変数を追加する必要があります。

NMAS コマンドの呼び出し

NMAS コマンドを呼び出す方法は、実行しているプラットフォームによって異なります。次のプラットフォームがサポートされています。

- 696 ページの 「Windows」
- 696 ページの 「Linux」

Windows

NMAS は、起動時に nmas.cfg ファイル内のコマンドを処理します。nmas.cfg ファイルは、dib ファイルと同じディレクトリ (通常は c:/novell/nds/dibfiles) に配置されている必要があります。

または

NMAS が起動したら、次の手順を使用します。

1 OpenText eDirectory Services コンソールで、nmas.dlm を選択します。
2 [**起動パラメータ**] フィールドにコマンドを入力します。
3 [**Configure (構成)**] をクリックします。

Linux

NMAS は、起動時に nmas.config ファイル内のコマンドを処理します。nmas.config ファイルは、dib ディレクトリと同じディレクトリに配置されている必要があります。たとえば、.dib ディレクトリのパスが /var/opt/novell/eDirectory/data/dib の場合は、nmas.config ファイルのパスは /var/opt/novell/eDirectory/data/nmas.config になります。

失敗したログイン試行の遅延時間の設定

1 OpenText Identity Console のホームページで、[**オブジェクト管理**] タイルをクリックします。
2 [**タイプ**] ドロップダウンメニューをクリックして、[**ユーザ**] を選択し、[**ユーザの変更**] ページで、[**認証**] タブ > [**設定**] ドロップダウンの順にクリックします。
3 失敗したログイン試行から次のログイン試行までの任意のログイン画面表示遅延時間を秒数で入力してから、[**保存**] をクリックします。

DSTrace の使用

DSTrace ユーティリティを使用して、NMAS からトレース情報を取得できます。

NMAS クライアントトレースを収集する方法については、「TID # 3331372」を参照してください。

NMAS サーバトレースを収集する方法については、「TID # 3815371」を参照してください。

NMAS クライアントの無効化とアンインストール

NMAS クライアントを無効にするには :

1 ワークステーションで、赤色の N を右クリックします。
2 [**Novell Client Properties**] をクリックします。
3 [**Advanced Login**] タブをクリックします。

4 ［Parameter Groups］リストから、［NMAS Authentication］を選択します。

5 ［設定］で、［オフ］を選択します。

6 OK をクリックします。

NMAS クライアントをアンインストールするには、Windows の［コントロール パネル］で［プログラムの追加と削除］オプションを使用します。

注 : NMAS を無効にしたり削除したりしても、Novell Client for Windows からのユニバーサル パスワードの変更のサポートは削除されません。

OpenText Audit と外部証明書の使用

NMAS と OpenText Audit で外部証明書を使用するには、まず、証明書を次の名前の 2 つの .pem ファイルに変換する必要があります。

- nmascert.pem: これは証明書を含むファイルです。
- nmaskey.pem: これは秘密鍵を含むファイルです。

これらのファイルは、システム内の各 NMAS サーバで、プラットフォームごとに次のディレクトリにコピーする必要があります。

- Linux: /etc
- Windows: GetWindowsDirectory からの戻り値 (通常は c:\windows)

NMAS は、ログがオープンの場合、nmascert.pem ファイルと nmaskey.pem ファイルが存在すればそれらを OpenText Audit プラットフォームエージェントに提供します。ファイルが存在しない場合は、NMAS が OpenText Audit プラットフォームエージェントに内部証明書と鍵を提供します。

セキュリティ上の考慮事項

このセクションでは、NetIQ モジュラー認証サービスのセキュリティに関連した具体的な情報を提供します。以下のサブセクションで構成されています。

- 698 ページの 「パートナーログインメソッド」
- 698 ページの 「ログインポリシー」
- 699 ページの 「NMASInst」
- 699 ページの 「ユニバーサルパスワード」
- 701 ページの 「SDI キー」

パートナーログインメソッド

OpenText では、パートナーのログインメソッドのセキュリティ手法を評価していません。パートナーの製品に OpenText Yes、Tested & Approved、または OpenText eDirectory Enabled のロゴの表示が認められていても、それらのロゴは一般的な製品の相互運用性を示しているにすぎません。

ログインポリシー

- 承認済みログインシーケンス、デフォルトログインシーケンス、承認済みクリアランス、またはデフォルトクリアランスがパーティションルートではないコンテナに割り当てられている場合、そのポリシーは、そのコンテナ内のユーザオブジェクトに対してのみ有効になり、サブコンテナ内のユーザオブジェクトに対しては有効になりません。
- 承認済みログインシーケンス、デフォルトログインシーケンス、承認済みクリアランス、またはデフォルトクリアランスがパーティションルートであるコンテナに割り当てられている場合、そのポリシーは、それらの値がユーザオブジェクトやオブジェクトのペアレントコンテナに割り当てられていないパーティション内のすべてのユーザに対して有効になります。
- 承認済みログインシーケンス、デフォルトログインシーケンス、承認済みクリアランス、またはデフォルトクリアランスがログインポリシーに割り当てられている場合、そのポリシーは、それらの値がユーザオブジェクト、オブジェクトのペアレントコンテナ、またはオブジェクトのパーティションルートに割り当てられていないツリー内のすべてのユーザに対して有効になります。
- ユーザにパスワードまたはその他の推測可能なログインシークレット (秘密の質問の回答など) が割り当てられている場合は、不正侵入者検出を有効にして、不正侵入者によるログインシークレットの推測を遅らせるかまたは阻止する必要があります。
- デフォルトで、失敗したログイン試行には 3 秒間の遅延が適用されます。この遅延の目的は、不正侵入者がパスワードを推測する試みを遅らせることです。失敗したログインの遅延の長さは設定によって変更できます。デフォルトの 3 秒間を使用することをお勧めします。
- 不正侵入者検出、ネットワークアドレス制限、時間制限などのログインポリシーは、すべてのログインシーケンスに適用されます。たとえば、いくつかの OpenText 製品で提供されているパスワード忘れセルフサービス機能によってチャレンジ / レスポンス方式のログインメソッドが呼び出された場合は、ログインポリシーが適用されます。
- ログイン試行と設定変更を追跡できるように、NMAS™ 監査を有効にする必要があります。
- ポリシーリフレッシュレートコマンドを使用して、キャッシュに保存されたパスワードポリシーをリフレッシュする必要があるかどうかを、ログインのたびにではなく定義したインターバルでチェックしている場合は、ログインポリシーの変更の適用に遅れが生じます。

- LoginInfo コマンドを使用すると、ログイン中のログイン関連属性の更新を無効にすることができます。これらの属性には、不正侵入者検出属性が含まれます。これらのログイン関連属性の更新を無効にすると、ログインのパフォーマンスが向上します。ただし、これらの属性の更新を無効にすると、システムのセキュリティが低下する可能性があります。
- 不正侵入者検出ポリシーは、ユーザオブジェクトの直接のコンテナまたはユーザオブジェクトのパーティションルート上で設定できます。NMAS は、最初に、ペアレントコンテナで不正侵入者検出ポリシーをチェックします。ポリシーが見つからなかった場合は、パーティションルートで不正侵入者検出ポリシーがチェックされます。

NMASInst

ログインメソッドをアップグレードするときに、-checkversion オプションが使用されていなければ、nmasinst が新しいバージョンを古いバージョンに置き換えます。

nmasinst にはコマンドラインでパスワードを指定するオプションがありますが、この方法ではパスワードが漏洩する可能性があるため推奨されていません。OpenText eDirectory 9.0 以降では、nmasinst によって、ファイルまたは環境変数からパスワードを取得できます。

ユニバーサルパスワード

- セキュリティコンテナにはグローバルポリシーが含まれているため、書き込み可能なレプリカを配置する場所に注意する必要があります。一部のサーバは、OpenText eDirectory ツリーで指定されたセキュリティポリシー全体を変更することができます。ユーザが NMAS を使用してログインするためには、ユーザオブジェクトとセキュリティコンテナのレプリカを NMAS サーバ上に配置する必要があります。
- パスワードポリシーがパーティションルートではないコンテナに割り当てられている場合、そのポリシーは、コンテナ内のユーザオブジェクトに対してのみ有効になり、サブコンテナ内のユーザオブジェクトに対しては有効になりません。
- パスワードポリシーがパーティションルートであるコンテナに割り当てられている場合、そのポリシーは、それらの値がユーザオブジェクトにもオブジェクトのペアレントコンテナにも割り当てられていないパーティション内のすべてのユーザに対して有効になります。
- パスワードポリシーがログインポリシーに割り当てられている場合、そのポリシーは、それらの値がユーザオブジェクト、オブジェクトのペアレントコンテナ、またはオブジェクトのパーティションルートのいずれにも割り当てられていないツリー内のすべてのユーザに対して有効になります。
- 「既存のパスワードがパスワードポリシーに準拠しているかどうかを確認する (ログイン時に検証が実行される)」パスワードポリシールールが true に設定されていない場合は、NDS パスワードがユニバーサルパスワードに移行されるときにパスワード期限の時刻が更新されません。
- パスワードポリシーは、ユーザまたはパスワード管理者が文書化された NMAS LDAP 拡張機能を使用してユニバーサルパスワードを読み込むことができるように設定できます。特定のインストールに必要でない限り、これらのオプションは有効にしないでく

ださい。ユーザパスワードを読み込み可能にする必要がある場合は、選択されたユーザだけがパスワードを読み込めるようにパスワードポリシーを設定する必要があります。

- 接続されたシステム間でのパスワードの同期に Identity Manager Password Synchronization が使用されている場合にのみ、配布パスワードに同期するためのパスワードポリシーを設定する必要があります。

  Identity Manager Password Synchronization を使用して接続されたシステム間でパスワードを同期する方法については、『*OpenText Identity Manager 4.5 Password Management Guide (OpenText Identity Manager 4.5 パスワード管理ガイド)*』を参照してください。

- 単純パスワードに同期するようにパスワードポリシーを設定する必要があるのは以下の場合です。
  - ユーザオブジェクトの書き込み可能なレプリカが保存されているサーバがある。
  - ユーザがCIFSやAFPなどのネイティブなファイルアクセスプロトコルを使用してそれらのサーバにアクセスしている。
- パスワードポリシーに対して詳細パスワードルールが有効になっている場合は、ユーザオブジェクトに対する従来のパスワードルールが無視され、ユーザが自分のパスワードを変更するときかログインするときに、パスワードポリシーのルールに合わせて更新されます。
- パスワード除外ルール (パスワード履歴、除外されたパスワード、および許可されていない属性値) は、NMAS がランダムパスワードの生成に使用されている場合は適用されません。
- パスワードルールを選択する場合は、推測するのが難しいパスワードにする必要がありますが、覚えられないものにはならないようにしなければなりません。
- 管理者が NDS パスワードを削除するように指定した場合は、NDS パスワードハッシュが OpenText eDirectory 以外は認識できないランダム値に設定されます。そのランダム値にハッシュ可能なパスワード値は存在することもあれば存在しないこともあります。
- XML パスワードの複雑さ
  - 重複したルールタグがある場合、ポリシーに照らしたパスワードのチェックとランダムパスワードの生成には、最も制限的なルールが使用されます (他のルールは無視されます)。
  - ランダムパスワードの生成では、ViolationsAllowed と NumberOfCharactersToEvaluate のルール設定属性が無視されます。
  - ランダムパスワードの生成には、XML 内の最初のポリシーのみが使用されます。

ユニバーサルパスワードセキュリティの詳細については、「785 ページの第 26 章「パスワードを管理する」」を参照してください。

SDI キー

ツリー鍵とも呼ばれるセキュリティドメインインフラストラクチャ (SDI) 鍵を Triple DES 鍵 (3DES) にする必要があります。SDI 鍵は、SDIDiag ユーティリティを使用してチェックまたはアップグレードすることができます。785 ページの第 26 章「パスワードを管理する」の「SDI ドメインキーサーバ上で NICI 3.0 が実行されていることを検証します。」を参照してください。

OpenText eDirectory 9.0 以降では、AES 256 ツリー鍵もサポートされます。詳細については、「Creating an AES 256-Bit Tree Key (AES 256 ビットツリーキーを作成する)」を参照してください。

25 Certificate Server について

OpenText eDirectory をインストールすると、OpenText Certificate Server が自動的にインストールされます。Certificate Server は、OpenText eDirectory にネイティブに統合された非対称暗号化サービスを提供します。このサービスを使用すれば、ユーザとサーバの両方の証明書を作成、発行、および管理することができます。これらのサービスにより、インターネットなどのパブリック通信チャネルを介した機密データの伝送を保護できます。

注

- 公開鍵暗号の概念については、「773 ページの 「公開鍵暗号化の基本」」を参照してください。
- RSA 暗号化用の MD2 および MD5 署名アルゴリズムは、OpenText eDirectory 9.2 以降ではサポートされません。

- 703 ページの 「OpenText Certificate Server の機能」
- 704 ページの 「OpenText Certificate Server のコンポーネント」
- 711 ページの 「OpenText Certificate Server の設定」
- 722 ページの 「OpenText Certificate Server の管理」
- 773 ページの 「公開鍵暗号化の基本」
- 779 ページの 「タスクを実行するために必要なエントリ権」

OpenText Certificate Server の機能

公開鍵暗号はネットワーク管理者にとっての特有の課題です。OpenText Certificate Server には、そのような課題に対処するのに役立つ次の特徴があります。

- ネットワーク上で公開鍵暗号サービスを提供します

  OpenText eDirectory ツリー内に組織認証局 (CA) を作成することができ、無制限の数のユーザ証明書とサーバ証明書を発行することができます。また、外部認証局のサービスも使用でき、必要に応じて両方を組み合わせて使用することができます。
- 公開鍵証明書の取得と管理に関連したコストを制御します

  組織認証局を作成し、その組織の認証局を通して公開鍵を発行することができます。
- 公開鍵証明書を改ざんから保護しながら公に利用可能にします

  証明書は OpenText eDirectory に保存されるため、eDirectory のレプリケーション機能とアクセス制御機能を利用できます。
- 署名/復号化操作に秘密鍵を使用するソフトウェアルーチンのみが秘密鍵にアクセスできるようにします

秘密鍵は NICI によって暗号化され、署名 / 復号化操作にそれらの秘密鍵を使用するソフトウェアルーチンのみがそれらの秘密鍵を使用できるようにされます。

- 秘密鍵を安全にバックアップします

  秘密鍵は NICI によって暗号化され、OpenText eDirectory に保存され、標準の OpenText eDirectory バックアップユーティリティを使用してバックアップされます。

- OpenText Identity Console を使用して証明書を一元管理できます。

  OpenText Identity Console を使用すると、組織認証局から発行された証明書や、PKCS #10 形式の証明書署名要求をサポートする他の CA から発行された証明書を管理できます。

- ユーザが独自の証明書を管理できます

  ユーザは、Identity Console を使用して、システム管理者の介在なしに、暗号化対応アプリケーションで使用するキーをエクスポートできます。

- 広く使用されている電子メールクライアントとブラウザをサポートします

OpenText Certificate Server のコンポーネント

このセクションでは、OpenText Certificate Server のコンポーネントについて説明します。

- 704 ページの 「OpenText Certificate Server」
- 711 ページの 「NICI」

OpenText Certificate Server

OpenText Certificate Server は、Certificate Server の管理インタフェースである、OpenText Identity Console の PKI サーバコンポーネントで構成されます。

Certificate Server では、次のタスクを実行できます。

- OpenText eDirectory ツリーと組織に固有の組織認証局を設定します。
- 公開鍵証明書とその関連秘密鍵を要求および管理し、OpenText eDirectory ツリーに保存します。

証明書での 8192 ビット RSA キーの使用

証明書サーバを使用すると、証明書作成手順の一環としてキーサイズを選択できます。OpenText eDirectory は、最大 8192 ビットのキーサイズをサポートします。アプリケーションで X.509 認定と 8192 ビット RSA 公開鍵を使用する場合は、アプリケーションで 8192 ビット RSA 鍵がサポートされている必要があります。サポートされていないと、アプリケーションが期待どおりに機能しない可能性があります。

重要 : TLS 接続を確立するために 8K ビット RSA 公開鍵の X.509 証明書を使用すると、OpenText eDirectory サーバのパフォーマンスに影響します。OpenText では、8K ビット鍵の RSA 証明書を使用するように OpenText eDirectory サーバを構成することを推奨していませ

ん。TLS セッションを確立する処理はサーバでの計算量が多くなり、複数の TLS セッションが同時に確立されるとシステムのパフォーマンスが大幅に低下する可能性があるためです。

8192 ビット RSA 公開鍵で認証局証明書を作成する前に、eDirectory ツリー内のすべてのサーバを 9.1 にアップグレードしてください。

注 : 8192 ビット RSA 公開鍵でサーバ証明書を設定する前に、最小でも OpenText eDirectory 、Identity Console 1.7 を使用していることを確認してください。

ECDSA 証明書の使用

Certificate Server は、RSA 証明書のサポートと同様に、楕円曲線デジタル署名アルゴリズム (ECDSA) の証明書とキーの使用および管理をサポートします。

RSA 鍵との互換性のあるセキュリティを持つ ECDSA 鍵ペアは、RSA 鍵よりもはるかに小さく、TLS 接続の確立に使用された場合にパフォーマンスを大幅に向上させます。ECDSA は、楕円曲線暗号システム (ECC) を利用します。ECC は、楕円曲線を使ってキーを生成します。この技術は、RSA や Diffie-Hellman などのほとんどの公開鍵暗号化方式と組み合わせて使用できます。ECC ベースの署名をデジタル証明書とともに使用することには、サイズとパフォーマンスの両面でメリットがあります。

OpenText eDirectory は、次の曲線を使用した ECDSA 証明書とキーをサポートします。

- P-256
- P-384
- P-521

Suite B モードでは、Certificate Server は RFC 5759 に準拠します。この RFC は、すべての Suite B 証明書と CRL には、ECDSA とともに、P-256 か P-384 のどちらかの曲線を使用して生成されたキーを使用して署名する必要があることを規定しています。

証明書に P-256 曲線に基づく鍵が含まれている場合、署名する CA の鍵は、P-256 または P-384 曲線に基づいていなければなりません。証明書に P-384 曲線に基づく鍵が含まれている場合、署名する CA の鍵は、P-384 曲線に基づいていなければなりません。証明書と CRL は、署名する CA のキーのサイズに合わせて、SHA-256 または SHA-384 を使用してハッシュする必要があります。

インストールした OpenText Certificate Server は、OpenText Identity Console を使用して管理します。

OpenText Identity Console を使用して、次のタスクを実行できます。

- 706 ページの 「組織用の組織認証局を作成する」
- 706 ページの 「暗号化対応アプリケーションごとにサーバ証明書オブジェクトを作成する」
- 707 ページの 「ユーザ証明書の作成」
- 708 ページの 「ルート認証局コンテナの作成」

- 708 ページの 「ルート認証局オブジェクトの作成」
- 708 ページの 「外部のユーザとサーバ用の証明書の作成」
- 708 ページの 「証明書の検証」
- 709 ページの 「証明書取り消しリストの管理」
- 710 ページの 「秘密鍵と証明書のエクスポート」
- 710 ページの 「秘密鍵と証明書のインポート」
- 711 ページの 「SAS サービスオブジェクトの作成」

組織用の組織認証局を作成する

OpenText eDirectory ツリー内に組織認証局 (CA) が存在しなければ、インストール中にそれを作成することもできます。インストールの完了後に組織認証局を再度作成することもできます。

組織認証局オブジェクトには、公開鍵、秘密鍵、証明書、証明書チェーン、およびその他の組織認証局に関する設定情報が含まれています。組織認証局オブジェクトは、OpenText eDirectory 内のセキュリティコンテナ内に存在します。

認証局サービスを提供するようにサーバを設定すると、そのサーバは OpenText eDirectory ツリー全体で認証局サービスを実行します。ツリーに従属認証局証明書が存在しており、サブ認証局をホストするサーバを OpenText eDirectory 9.2 にアップグレードする場合は、ECDSA 認証局証明書は生成されません。管理者は、従属認証局の RSA 証明書と同じサブジェクト名を持つ従属認証局の ECDSA 証明書をインポートする必要があります。認証局として機能しているサーバが OpenText eDirectory 9.2 の場合は、OpenText eDirectory が組織認証局用の ECDSA 証明書を作成します。組織 CA に ECDSA 証明書が存在する場合は、eDirectory が自動的にサーバ用の ECDSA 証明書を作成します。

組織認証局の作成方法については、「725 ページの 「組織の認証局オブジェクトを作成する」」を参照してください。

暗号化対応アプリケーションごとにサーバ証明書オブジェクトを作成する

証明書サーバをインストールすると、デフォルトのサーバ証明書オブジェクトが作成されます。

- SSL CertificateDNS - *server_name*
- サーバ上で設定された IP アドレスごとの証明書 (IP AG *xxx.xxx.xxx.xxx - server_name*)
- サーバ上で設定された DNS 名ごとの証明書 (DNS AG *www.example.com - server_name*)
- SSL EC CertificateDNS - *server_name*
- サーバ上で設定された IP アドレスごとの証明書 (IP EC AG *xxx.xxx.xxx.xxx - server_name*)
- サーバ上で設定された DNS 名ごとの証明書 (DNS EC AG *www.example.com - server_name*)

注 : OpenText eDirectory では、SSL CertificateIP は自動的に作成されません。SSL CertificateDNS には、［サブジェクトの代替名］にリストされているすべての IP が含まれます。

インストールの完了後に、他のサーバ証明書オブジェクトを作成できます。

サーバ証明書オブジェクトには、公開鍵、秘密鍵、証明書、およびサーバアプリケーションに対して SSL セキュリティサービスを有効にする証明書チェーンが含まれます。サーバ証明書オブジェクトは、組織認証局と外部認証局のどちらかで署名することができます。

サーバは、多数のサーバ証明書オブジェクトをサーバに関連付けることができます。特定のサーバ上で動作している暗号化対応アプリケーションは、そのサーバのサーバ証明書オブジェクトのいずれかを使用するように設定できます。特定のサーバ上で動作している複数のアプリケーションは同じサーバ証明書オブジェクトを使用できますが、サーバ証明書オブジェクトをサーバ間で共有することはできません。

サーバ証明書オブジェクトは、サーバが存在するコンテナ内にのみ作成できます。サーバオブジェクトを移動する場合は、そのサーバに属しているすべてのサーバ証明書オブジェクトも移動する必要があります。サーバ証明書オブジェクトの名前は変更しないでください。どのサーバ証明書オブジェクトがサーバに属しているかは、サーバ証明書オブジェクト名にそのサーバの名前が含まれているかを確認することによって、あるいは、OpenText Identity Console でサーバ証明書オブジェクトを表示してホストサーバを確認することによって判別できます。

サーバ証明書オブジェクトに保存されている鍵ペアは、その鍵ペアの作成時に入力された名前で参照されます。鍵ペアの名前は、サーバ証明書オブジェクトの名前ではありません。鍵ペアを使用するように暗号化対応アプリケーションを設定する場合、それらのキーは、サーバ証明書オブジェクト名ではなく、鍵ペア名で参照します。

デフォルトサーバ証明書オブジェクトが破損するか無効になった場合は、デフォルト証明書の作成ウィザードを使用してその古いデフォルト証明書を置き換えます。デフォルト証明書の作成ウィザードにアクセスする方法については、「734 ページの 「デフォルトのサーバ証明書オブジェクトを作成する」」を参照してください。

デフォルトでは、組織認証局に ECDSA 証明書が存在する場合は、OpenText eDirectory が ECDSA 証明書を作成します。

ユーザ証明書の作成

ユーザは自分のユーザ証明書と秘密鍵にアクセスして、それらを認証、データ暗号化 / 復号化、デジタル署名、およびセキュリティで保護された電子メールに利用できます。最も一般的な用途の 1 つが、S/MIME 標準を使用してデジタル署名され、暗号化される電子メールの送受信です。

通常は、ユーザ証明書を作成するのに十分な権利を保有するのは、CA 管理者だけです。しかし、秘密鍵を OpenText eDirectory からエクスポートしたりダウンロードしたりする権利は、該当ユーザのみが保有します。すべてのユーザが他のユーザの公開鍵証明書をエクスポートできます。

ユーザ証明書は、ユーザのプロパティページの［セキュリティ］タブで作成され、組織認証局によって署名されます。他の CA によって作成された証明書と秘密鍵は、作成後にインポートできます。

複数の証明書をユーザのオブジェクトに保存することができます。

ユーザ証明書の作成方法については、「720 ページの 「ユーザ証明書の作成」」を参照してください。

ルート認証局コンテナの作成

ルート認証局は、公開鍵暗号の信頼の基盤です。ルート認証局は、他の CA によって署名された証明書の検証に使用されます。ルート認証局は、SSL、セキュリティで保護された電子メール、および証明書ベースの認証のセキュリティを有効にします。

ルート認証局コンテナは、ルート認証局オブジェクトを格納する OpenText eDirectory オブジェクトです。

デフォルトのルート認証局コンテナは、CN=trusted roots.CN=security です。

ルート認証局コンテナの作成方法については、「721 ページの 「ルート認証局コンテナの作成」」を参照してください。

ルート認証局オブジェクトの作成

ルート認証局オブジェクトは、認証され、有効であると認められた認証局のルート認証局証明書を格納する OpenText eDirectory オブジェクトです。ルート認証局証明書は、必要に応じてエクスポートして使用することができます。ルート認証局証明書を使用するように設定されたアプリケーションは、ルート認証局コンテナ内のいずれかの CA によって署名された証明書を有効と見なします。

ルート認証局オブジェクトは、ルート認証局コンテナ内に存在する必要があります。

ルート認証局オブジェクトの作成方法については、「721 ページの 「ルート認証局オブジェクトの作成」」を参照してください。

外部のユーザとサーバ用の証明書の作成

認証局管理者は、組織認証局を使用して、OpenText eDirectory の外部のユーザとサーバの証明書に署名できます。このような証明書は、帯域外の方法で CA 管理者に提供される PKCS #10 証明書署名要求 (CSR) を使用して要求されます。

CSR を受け取った認証局管理者は、OpenText Identity Console の証明書発行ツールを使用して、証明書を発行できます。この証明書は、OpenText eDirectory 内のオブジェクトに保存されません。帯域外の方法で要求元に返す必要があります。

証明書の検証

OpenText Certificate Server では、OpenText eDirectory ツリー内のすべての証明書の有効性をチェックできます。証明書検証プロセスは、証明書チェーン内の各証明書を、ルート認証局証明書までたどってチェックして、有効または無効のステータスを返します。

- 組織認証局の証明書の有効性をチェックするには、「731 ページの 「組織認証局の証明書の検証」」を参照してください。

- サーバの証明書の有効性をチェックするには、「742 ページの 「サーバ証明書の検証」」を参照してください。
- ユーザの証明書の有効性をチェックするには、「747 ページの 「ユーザ証明書の検証」」を参照してください。
- ルート認証局の証明書の有効性をチェックするには、「757 ページの 「ルート認証局オブジェクトの検証」」を参照してください。

現在の時刻が証明書の有効期間内であるか、証明書が取り消されていないか、信頼されている認証局によって署名されているかなど、事前定義された一連の条件を証明書が満たすと、その証明書は有効であると見なされます。

外部認証局によって署名された CN=trusted roots.CN=security 内のユーザ証明書または中間 CA 証明書を検証する場合、証明書検証が成功するためには、その外部認証局の証明書がルート認証局オブジェクト内に保存されている必要があります。

証明書取り消しリストの管理

証明書取り消しリスト (CRL) は、取り消された証明書とその理由の公開されたリストです。

OpenText Certificate Server は、CRL を管理するためのシステムを提供します。これはオプションシステムですが、組織認証局によって作成された証明書を取り消せるようにするには、このシステムを実装する必要があります。CRL の管理方法については、「758 ページの「証明書取り消しリスト (CRL) のタスク」」を参照してください。

NetIQ Certificate Server のインストール時に、ユーザが CRL コンテナの作成に必要な権利を持っている場合は、CRL コンテナが作成されます。そうでない場合は、インストールが完了してから、適切な権利を持つユーザが手動で CRL コンテナを作成できます。

CRL 設定オブジェクトは、CRL コンテナ内に作成できます。オブジェクトには、OpenText eDirectory ツリーで使用可能な CRL オブジェクトに関する設定情報が含まれています。通常、ツリーには CRL 設定オブジェクトが 1 つだけあります。新しい組織認証局を作成またはロールオーバする場合には、複数の CRL 設定オブジェクトが必要になりますが、新しい証明書を作成するために使用できる CRL 設定オブジェクトは 1 つだけです。

CRL オブジェクト (配布ポイントとも呼ばれます) は、OpenText eDirectory ツリー内の任意のコンテナに作成できます。ただし、OpenText CRL オブジェクトは通常、CRL コンテナに存在します。CRL 設定オブジェクトを作成すると、CRL オブジェクトが自動的に作成されます。CRL オブジェクトには、CRL の詳細情報が記載された CRL ファイルが含まれています。OpenText CRL オブジェクトの場合、サーバが新しい証明書を発行するたびに、CRL ファイルは自動的に作成および更新されます。その他の CRL オブジェクトの場合、サードパーティの認証局から CRL ファイルをインポートする必要があります。組織認証局を保持するサーバを OpenText eDirectory 9.2 にアップグレードするとき、アップグレードプロセスが CRL 配布ポイントを自動的に作成します。また、OpenText eDirectory では、RSA 証明書と ECDSA 証明書用の別個の CRL 設定オブジェクトが提供されます。

CRL 設定オブジェクトは削除できますが、推奨されていません。CRL 設定オブジェクトを削除すると、サーバは CRL ファイルを作成しなくなります。CRL オブジェクトで指定された場所に CRL ファイルがすでに存在する場合は、有効期限が切れるまで、証明書の検証でそのファイルが使用し続けられます。証明書が期限切れになると、CRL 配布ポイントでその既存の CRL ファイルを参照しているすべての証明書の検証が失敗します。

CRL オブジェクトを削除すると、サーバが次に CRL ファイルを生成するときに CRL オブジェクトが再作成されます。OpenText Identity Console を使用して作成した CRL オブジェクトを削除しそれをインポートする場合、そのオブジェクトは永続的に失われ、そのオブジェクトを参照するすべての証明書は無効とみなされます。

通常、関連する配布ポイントを含む最後の証明書の有効期限が切れてから 1 日経過するまで、CRL コンテナ、CRL 設定オブジェクト、CRL オブジェクト、CRL ファイルは削除しません。

秘密鍵と証明書のエクスポート

ユーザ、サーバ、および認証局キーを作成するときに、それらをエクスポート可能としてマークできます。鍵がエクスポート可能な場合、その鍵を抽出し、関連する証明書と一緒にファイルに配置できます。ファイルは、他のプラットフォームに転送可能な業界標準の形式 (PFX または PKCS #12) で書き込まれます。秘密鍵を保護するため、ファイルはユーザが指定したパスワードで暗号化されます。

秘密鍵と証明書をエクスポートして、鍵のバックアップコピーを取ったり、鍵を別のサーバに移動したり、またはサーバ間で鍵を共有したりできます。

秘密鍵と証明書をエクスポートする方法の詳細については、747 ページの 「ユーザ証明書と秘密鍵のエクスポート」を参照してください。

秘密鍵と証明書のインポート

サーバ証明書、ユーザ証明書、または認証局オブジェクトの作成時に、新しい鍵を作成するのではなく鍵をインポートすることもできます。鍵とそれに関連付けられた証明書は、PFX または PKCS #12 形式である必要があります。

認証局オブジェクトをサーバの障害から回復するため、1 つのサーバから別のサーバに組織認証局を移動するため、または別の認証局に従属する認証局のために、新しい鍵を作成するのではなく、鍵をインポートすることもできます。

ユーザ証明書または秘密鍵がサードパーティ認証局によって署名されている場合でも、インポートすることができます。

サーバ証明書オブジェクトをサーバの障害から回復するため、鍵と証明書を別のサーバに移動するため、または別のサーバと鍵と証明書を共有するために、新しい鍵を作成するのではなく、鍵をインポートすることもできます。

SAS サービスオブジェクトの作成

SAS サービスオブジェクトは、サーバとサーバ証明書間の通信を容易にします。サーバを OpenText eDirectory ツリーから削除する場合は、サーバに関連付けられた SAS サービスオブジェクトを削除する必要があります。サーバをツリーに戻す場合は、そのサーバに属する SAS Service オブジェクトを作成する必要があります。ツリーに戻さない場合は、新しいサーバ証明書を作成することはできません。

SAS Service オブジェクトは、サーバヘルスチェックの一部として自動的に作成されます。このオブジェクトを手動で作成する必要はありません。

新しい SAS Service オブジェクトを作成できるのは、サーバオブジェクトと同じコンテナ内に、正しく名付けられた SAS Service オブジェクトが存在しない場合だけです。たとえば、「WAKE」と名付けられたサーバでは、「SAS Service- WAKE」と名付けられた SAS Service オブジェクトが作成されます。ユーティリティでは、サーバオブジェクトから SAS Service までの DS ポインタ、および SAS Service からサーバオブジェクトまでの DS ポインタが追加されます。また、SAS サービスオブジェクト上に正しい ACL エントリが設定されます。

SAS サービスオブジェクトがすでに正しい名前で存在する場合、新しい SAS サービスオブジェクトを作成することはできません。古い SAS サービスオブジェクトの DS ポインタは、誤っていたり見つからなかったりする場合があります。または、ACL が適切でない場合もあります。この場合、破損した SAS サービスオブジェクトを削除し、Identity Console を使用して新しい SAS サービスオブジェクトを作成できます。このサーバに属するサーバ証明書がある場合、［**その他**］タブを使用して、サーバ証明書を SAS サービスオブジェクトへ手動でリンクする必要があります。

SAS サービスオブジェクトを作成する方法の詳細については、722 ページの 「SAS サービスオブジェクトの作成」を参照してください。

NICI

NICI は、OpenText Certificate Server、NMAS (NetIQ モジュラー認証サービス)、およびその他のアプリケーションに暗号化を提供する、基盤となる暗号化インフラストラクチャです。

OpenText Certificate Server を正しく機能させるために、サーバに NICI をインストールする必要があります。NICI は、OpenText Certificate Server には付属していません。大抵、OpenText Certificate Server が Open Enterprise Server (OES) や OpenText eDirectory などの別の製品にバンドルされる場合に、NICI の提供およびインストールが行われます。新しいバージョンの NICI が必要な場合は、ソフトウェアライセンスおよびダウンロードポータルからダウンロードできます。

OpenText Certificate Server の設定

OpenText Certificate Server をインストールした後、次のタスクを実行して、ネットワーク上で使用できるように設定する必要があります。

- 712 ページの 「使用する認証局のタイプの決定」
- 713 ページの 「組織の認証局オブジェクトを作成する」

- 715 ページの 「従属認証局」
- 718 ページの 「認証局オブジェクトの作成に関する制限事項」
- 718 ページの 「Suite B モードでの認証局の設定」
- 719 ページの 「サーバ証明書オブジェクトを作成する」
- 720 ページの 「暗号化対応アプリケーションの設定」
- 720 ページの 「追加コンポーネントの設定」

使用する認証局のタイプの決定

OpenText Certificate Server では、サーバとエンドユーザの両方の証明書を作成できます。組織認証局または外部 (サードパーティ) 認証局のいずれかによって、サーバ証明書に署名できます。ユーザ証明書への署名は組織認証局でしかできませんが、PKCS #12 形式でサードパーティ認証局によって署名されたユーザ証明書をインポートすることもできます。

サーバ証明書オブジェクトの作成プロセス中に、サーバ証明書オブジェクトに署名する認証局のタイプを尋ねられます。

組織認証局は組織固有のものであり、署名操作には組織固有の公開鍵を使用します。秘密鍵は、組織認証局を作成するときに作成されます。

サードパーティ認証局は、OpenText eDirectory ツリーの外部でサードパーティによって管理されます。サードパーティ認証局の例として、VeriSign があります。

認証局の両方のタイプを同時に使用できます。認証局の 1 つのタイプを使用しても、もう一方を使用できなくなることはありません。

- 712 ページの 「OpenText Certificate Server で提供される組織認証局を使用することの利点」
- 713 ページの 「外部認証局を使用することの利点」

OpenText Certificate Server で提供される組織認証局を使用することの利点

- **互換性。**組織認証局は、LDAP サービスなどの OpenText のアプリケーションと互換性があります。組織認証局によって発行される証明書は、X.509 v3 に準拠しており、サードパーティ製のアプリケーションでも使用できます。
- **認証局のコスト削減。**組織認証局では、コストをかけずにいくらでも公開鍵証明書を作成できます。外部の認証局で単一の公開鍵証明書を取得しようとすると、かなりの費用がかかる可能性があります。
- **すべて揃った互換性のあるソリューションのコンポーネント。**組織認証局を使用すると、どんな外部サービスにも依存せずに、OpenText eDirectory に組み込まれた完全な暗号化システムを使用できます。さらに、OpenText Certificate Server は、さまざまな OpenText 製品と互換性があります。
- **証明書の属性およびコンテンツ制御。**組織認証局は、ネットワーク管理者によって管理されます。ネットワーク管理者は、証明書の有効期間、キーサイズ、署名アルゴリズムなどの公開鍵証明書の属性を決定します。

- **簡単な管理。**組織認証局は、追加コストや複雑な手順なしに、外部認証局と同様の機能を実行します。

外部認証局を使用することの利点

- **法的責任。**外部認証局の過失により、秘密鍵が漏洩したり、公開鍵証明書が改ざんされたりした場合に、その認証局が賠償責任を負う可能性があります。
- **可用性。**外部認証局の証明書は、OpenText eDirectory の外部アプリケーションでも幅広く利用可能で信頼される可能性があります。

組織の認証局オブジェクトを作成する

デフォルトでは、OpenText Certificate Server のインストールプロセスで、組織認証局 (CA) が作成されます。組織認証局の名前を指定するように求められます。[**完了**] クリックすると、組織認証局がデフォルトのパラメータで作成され、セキュリティコンテナに配置されます。

組織認証局の作成をより細かく制御する場合は、OpenText Identity Console を使用して組織認証局を手動で作成できます。また、組織認証局を削除した場合、それを再作成する必要があります。

作成プロセス時に、組織認証局オブジェクトの名前を付け、組織認証局サービスをホストするサーバ (組織認証局サービスが実行されるサーバ) を選択するよう求められます。組織認証局サービスをホストするサーバを決定する際、次の点を考慮します。

- 物理的に安全なサーバを選択します。

  認証局サーバへの物理的アクセスは、システムのセキュリティ上、重要な部分になります。認証局サーバが侵害されると、その認証局によって発行されたすべての証明書も侵害されます。
- 可用性、安定性、および堅牢性の高いサーバを選択します。

  認証局サービスが使用できない場合、証明書は作成できません。証明書はインストール時に作成される必要があるため、これは新しいサーバのインストールに影響します。
- 信頼できるソフトウェアのみが実行されているサーバを選択します。

  不明なまたは不審なソフトウェアを実行していると、認証局サービスが侵害される可能性があります。
- ツリーから削除されないサーバを選択します。

  サーバがツリーから削除されると、認証局を削除する前に取ったバックアップを使用して認証局オブジェクトを再作成するか、新しい認証局を作成する必要があります。新しい認証局を作成する場合、既存のサーバとユーザ証明書を置換する必要があります。
- ツリー内の他のサーバと互換性があるプロトコルを実行しているサーバを選択します。

  たとえば、IP です。
- ECDSA 証明書で組織認証局を作成します。

組織認証局オブジェクトを作成するには、次の手順を実行します。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**CA 管理**] タイル > [**OpenText Certificate Server**] > [**認証局の設定 (Configure Certificate Authority)**] の順に選択します。

組織の認証局オブジェクトが存在しない場合、[Create an Organizational Certificate Authority Object] ダイアログボックスとオブジェクトを作成するウィザードが開きます。メッセージに従ってオブジェクトを作成します。ダイアログボックスまたはウィザードページの具体的な説明については、[ヘルプ] をクリックしてください。

注 : ここで指定する CRL ファイルパスが、OpenText eDirectory インストールパスと対応していることを確認します。

---

4 認証局の作成を完了したら、認証局の公開鍵 / 秘密鍵のペアのバックアップを作成し、安全な場所に保存することをお勧めします。詳細については、727 ページの 「組織認証局のバックアップ」を参照してください。

注 : OpenText eDirectory ツリーは組織の認証局を 1 つしか格納できません。

OpenText eDirectory では、管理者は、デフォルトのサーバ証明書が生成されるときに使用される RSA 鍵のサイズ、楕円曲線、および証明書を指定できます。認証局オブジェクトの次の 3 つの属性を使用して、これらのパラメータを指定できます。

- **ndspkiDefaultRSAKeySize:** RSA サーバ証明書の鍵サイズを指定します。このフィールドでは、最大で 8192 ビット RSA 暗号化まで指定できます。

  重要 : TLS 接続を確立するために 8K ビット RSA 公開鍵の X.509 証明書を使用すると、OpenText eDirectory サーバのパフォーマンスに影響します。OpenText では、8K ビット鍵の RSA 証明書を使用するように OpenText eDirectory サーバを構成することを推奨していません。TLS セッションを確立する処理はサーバでの計算量が多くなり、複数の TLS セッションが同時に確立されるとシステムのパフォーマンスが大幅に低下する可能性があるためです。

- **ndspkiDefaultECCurve:** サーバ証明書の EC 制限の曲線を指定します。次のいずれかの EC を指定できます。
  - P256
  - P384
  - P521

- **ndspkiDefaultCertificateLife:** デフォルトのサーバ証明書の証明書有効期限を指定します。年単位でサーバ証明書有効期限を指定できます。たとえば、このフィールドに 4 と指定すると、サーバ証明書有効期限は 4 年間に設定されます。証明書サーバは、デフォルトのサーバ証明書の最小の有効期間が 1 年間であることと、最大の有効期間が CA の期限を超えていないことを確認します。

  注

  - ndspkiDefaultCertificateLife 属性は、サーバ証明書にのみ適用できます。
  - 新しい OpenText eDirectory サーバを設定する際に古いデフォルト値を渡すと、上記のパラメータは未設定のままになります。
  - 上記のパラメータは、既存のデフォルト証明書には影響しません。認証局にこれらのパラメータを指定した後に OpenText eDirectory サーバを 9.2 にアップグレードする場合、既存のデフォルトサーバ証明書は再作成されません。

新しい OpenText eDirectory ツリーを設定する際にこれらのパラメータを指定すると、組織認証局証明書もこれらのパラメータを使用して作成されます。詳細については、『OpenText™ eDirectory CE 24.4 (v9.3) インストールガイド』を参照してください。

従属認証局

OpenText Certificate Server に、従属認証局のサポートが追加されました。この機能によって、組織認証局をサードパーティ認証局または別の OpenText eDirectory ツリーにある認証局のいずれかに従属させることができます。OpenText eDirectory ツリーには、組織認証局を 1 つしか格納できません。

従属認証局を使用する次のような理由があります。

- 組織認証局を既存のサードパーティ PKI の一部にすることができます。
- 複数のツリーで、共通の PKI のルート認証局 (または信頼アンカー) を共有することができます。
- 認証局をより安全なシステム上に配置することにより、ルート認証局のセキュリティを向上させることができます。
- より緊密に管理されているツリー内にルート認証局を配置することにより、リスクを低減します (たとえば、不正管理者または不正ユーザから保護されているツリー)。

- 715 ページの 「従属認証局の作成」
- 716 ページの 「従属認証局の PKCS #12 ファイルの作成」

従属認証局の作成

従属認証局を作成するには、既存の組織認証局を最初に削除する必要があります (732 ページの 「組織認証局の削除」を参照してください)。従属認証局の公開鍵 / 秘密鍵と証明書チェーンが格納されている PKCS #12 ファイルが存在している必要があります。サードパーティ認証局から直接このファイルを取得できます。また、このファイルを作成する方法に

ついては、716 ページの 「従属認証局の PKCS #12 ファイルの作成」を参照してください。従属認証局を作成するには、OpenText Identity Console でツリーに接続して認証局の設定タスクを実行し、［インポート］作成方法を使用します。

従属認証局の PKCS #12 ファイルの作成

1 サーバ証明書オブジェクト (または KMO) および ECDSA と RSA キーで PKCS #10 CSR を作成します。

1a OpenText Identity Console を起動します。

1b OpenText Identity Console のホームページで、［**サーバ証明書の管理**］タイル >、［**サーバ証明書の作成** ＋］の順に選択します。

1c 認証局をホストするサーバを選択し、証明書のニックネームを指定します。［カスタム］作成方法を選択して［**次へ**］をクリックします。

1d ［**External Certificate Authority**］を選択して、［**次へ**］をクリックします。

1e アルゴリズムおよびキーサイズを選択し、［**秘密鍵のエクスポートを許可**］が選択されていることを確認して、［**次へ**］をクリックします。

重要 : OpenText eDirectory 環境で RSA と ECDSA の両方の証明書を使用する場合、使用する証明書ごとにこの手順を繰り返します。OpenText では、RSA では 2048 ビット、ECDSA では 384 ビットのキーサイズを使用することをお勧めします。

1f ［**サブジェクト名**］フィールドの右側にある［**編集**］ボタンをクリックして、従属認証局とツリーを表すように［**サブジェクト名**］を編集し、署名アルゴリズムを選択します (OpenText では、SHA-1 より強力なアルゴリズムを使用することをお勧めします)。

1g 概要が正しいことを確認し、［**完了**］をクリックします。

1h ［**証明書署名要求の保存**］をクリックし、プロンプトに従って CSR をファイルに保存します。

2 CSR に署名を行い証明書を作成します。

2a 従属認証局がサードパーティ PKI の一部である場合、サードパーティ認証局に CSR から証明書を作成させます。

または

従属認証局が別の OpenText eDirectory ツリー内の認証局によって署名される場合は、ステップ 2b から続行します。

注 : ECDSA の場合、CSR は ECDSA CA でしか署名できません。

2b OpenText Identity Console を起動します。

2c OpenText Identity Console のホームページで、［**サーバ証明書の管理**］タイル >、［**サーバ証明書の作成** ＋］の順に選択します。

2d CSR を含むファイルを選択します。

2e ［**認証局**］のキータイプを選択し、［**拡張キーの使用を有効にする**］を選択解除します。

2f 認証局の証明書タイプを選択し、[指定なし] または [特定のパス (Specific Path)] 長のいずれかを選択します。

2g [証明書パラメータ] ドロップダウンをクリックして、サブジェクト名を確認し、必要に応じて編集します。有効期間を指定します (5 ～ 10 年を推奨)。

2h [証明書フォーマット] ドロップダウンをクリックして証明書のフォーマットを選択し、[次へ] をクリックします。

2i 完了をクリックします。

3 認証局証明書を取得します。

3a 従属認証局がサードパーティ PKI の一部になる場合、サードパーティから認証局証明書を取得します。

または

従属認証局が別の OpenText eDirectory ツリー内の認証局によって署名される場合は、ステップ 3b から続行します。˚

3b Identity Console を起動します。

3c OpenText Identity Console のホームページで、[サーバ証明書の管理] タイルをクリックして、[CA 管理] をクリックします。

3d [自己署名証明書 ECDSA (Self-Signed Certificate ECDSA)] 選択します。

[秘密鍵のエクスポート] チェックボックスをオンにすると、PKCS12 としてのエクスポート形式が有効になります。

3e [OK] をクリックし、[エクスポートしたファイルを保存する] オプションをクリックしてファイルを必要な場所に保存し、[OK] をクリックします。

4 証明書をサーバ証明書オブジェクト (または KMO) にインポートします。

4a OpenText Identity Console を起動します。

4b OpenText Identity Console のホームページで [CA 管理] タイルを選択します。

4c [認証局の作成] ページで、[サーバ] を選択し、[オブジェクト名] を指定して、[作成方法] を選択します。[次へ] > [OK] の順にクリックします。

4d [拡張キー使用の仕様] ドロップダウンを選択して、必要な認証局を選択します。

4e [基本制約] ドロップダウンをクリックして、[指定] を選択します。

4f [次へ] をクリックして。[OK] をクリックします。

4g ステップ 1 で作成したサーバ証明書オブジェクト (または KMO) を選択し、[インポート] をクリックします。

4h ステップ 2 およびステップ 3 で取得した証明書を格納している 2 つのファイルを選択し、[OK] をクリックします。

重要 : OpenText eDirectory 環境で RSA と ECDSA の両方の証明書を使用する場合、使用する証明書ごとにこの手順を繰り返します。

---

5 PKCS #12 ファイルに公開鍵 / 秘密鍵をエクスポートします。

5a ステップ 4h から操作を続け、[エクスポート] をクリックして、秘密鍵を含めることを選択し、[次へ] をクリックします。

5b [Save the Exported Certificate to a File] をクリックし、プロンプトに従って PKCS#12 ファイルを保存します。

5c このファイルのコピーを作成し、パスワードと一緒に安全な場所に保存します。

6 (オプション) サーバ証明書オブジェクト (または KMO) を削除します。

7 組織認証局を削除します。詳細については、732 ページの 「組織認証局の削除」を参照してください。

8 サブ認証局証明書と秘密鍵を PKCS #12 ファイルからインポートします。詳細については、729 ページの 「組織認証局の復元」を参照してください。

認証局オブジェクトの作成に関する制限事項

OpenText eDirectory 9.0 以降では、認証局の RSA と EC の両方の証明書をサポートします。RSA および EC 証明書で認証局オブジェクトを作成する際に、次の考慮事項が適用されます。

- RSA および EC 証明書のサブジェクト名は同じである必要があります。
- RSA と EC CA は、混合モードにはできません。RSA と EC CA は、ルート認証局またはサブ認証局のいずれかとして使用できます。たとえば、RSA CA をルート認証局として使用し、EC CA をサブ CA として使用することはできません。またその逆も同様です。
- OpenText eDirectory では、OpenText eDirectory 外部のサードパーティアプリケーションで生成された自己署名認証局証明書をインポートすることはできません。

Suite B モードでの認証局の設定

Suite B モードで認証局を設定する前に、認証局がホストされているサーバに、NICI 3.0 がインストールされていること、Enhanced Background Authentication を実行するように設定がなされていることを確認します。

Suite B モードで動作するように認証局を設定するには、次の手順を実行します。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

3 [証明書管理] タイル > [CA 管理] の順に移動します。

4 [CA 環境設定] () をクリックして、[Suite B モードを有効にする (Enable Suite B Mode)] チェックボックスをオンにします。

5 適用をクリックします。

認証局が Suite B モードのとき、証明書サーバは RSA 証明書を作成できません。どの ECDSA 証明書がサポートされているかについては、704 ページの 「OpenText Certificate Server のコンポーネント」を参照してください。

Suite B で設定されている OpenText eDirectory 9.0 以降のツリーに OpenText eDirectory の旧バージョンを使用するサーバを追加する場合、サーバに Suite B 機能がないため、Certificate Server はそのサーバの証明書を作成しません。これらのサーバで Suite B 機能を有効にするには、サーバを OpenText eDirectory 9.0 以降にアップグレードする必要があります。

サーバ証明書オブジェクトを作成する

サーバ証明書オブジェクトは、サーバの OpenText eDirectory オブジェクトを格納するコンテナに作成されます。必要に応じて、サーバ上にある暗号化対応アプリケーションごとに別個のサーバ証明書オブジェクトを作成することも、そのサーバで使用するすべてのアプリケーションに対して 1 つのサーバ証明書オブジェクトを作成することもできます。

注 : サーバ証明書オブジェクトと暗号化キーオブジェクト (KMO) は同じ意味です。OpenText eDirectory オブジェクトのスキーマ名は NDSPKI: 暗号化キーです。

Certificate Server をインストールすると、OpenText eDirectory はデフォルトパラメータで、サーバ証明書オブジェクトを自動的に作成し、ターゲットサーバが存在するコンテナに配置します。デフォルト証明書を上書きまたは新しく作成する必要がある場合は、[デフォルト証明書の作成] ウィザードを使用できます。詳細については、734 ページの 「デフォルトのサーバ証明書オブジェクトを作成する」を参照してください。

サーバ証明書オブジェクトの作成をより細かく制御する場合は、サーバ証明書オブジェクトを手動で作成できます。追加のサーバ証明書オブジェクトを作成することもできます。

- 719 ページの 「サーバ証明書オブジェクトを手動で作成する」
- 720 ページの 「サーバ証明書の作成に関するヒント」

サーバ証明書オブジェクトを手動で作成する

1. OpenText Identity Console を起動します。
2. 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

   このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。
3. [**証明書管理**] に移動して、[**サーバ証明書の管理**] をクリックします。
4. ＋ をクリックして、サーバ証明書オブジェクトを作成します。[**サーバ証明書の作成 (Create a Server Certificate)**] ページが開きます。
5. 作成されるサーバ証明書オブジェクトの [**ニックネーム**] を指定します。
6. [**作成方法**] を選択して、[**次へ**] をクリックします。
7. [**サマリ**] タブで、作成するユーザ証明書に適用されるパラメータを表示します。
8. [**OK**] をクリックします。

サーバ証明書の作成に関するヒント

サーバ証明書オブジェクトの作成プロセス中に、鍵ペアの名前を指定し、鍵ペアを関連付けるサーバを選択するよう求められます。サーバ証明書オブジェクトは、OpenText Certificate Server によって生成され、その名前は選択した鍵ペアの名前に基づいて指定されます。

［カスタム］作成方法を選択する場合も、サーバ証明書オブジェクトを、組織の組織認証局または外部認証局のいずれで署名するか指定するよう求められます。この決定方法の詳細については、712 ページの 「使用する認証局のタイプの決定」を参照してください。

組織の組織認証局を使用する場合、サーバ証明書オブジェクトが関連付けられているサーバが、組織認証局をホストするサーバと通信できるようにするか、またはそのサーバと同じである必要があります。これらのサーバでは、同じプロトコル (IP) を実行している必要があります。

外部認証局を使用して証明書に署名する場合は、サーバ証明書オブジェクトが関連付けられているサーバによって、外部認証局に送信する必要がある証明書署名要求が生成されます。

証明書が署名されて返されたら、外部認証局のルート認証局と一緒に、それをサーバ証明書オブジェクトにインストールする必要があります。

サーバ証明書オブジェクトを作成した後、これを使用するアプリケーションを設定できます。(参照先 720 ページの 「暗号化対応アプリケーションの設定」.) アプリケーションの設定では、鍵は、サーバ証明書オブジェクトを作成したときに入力した鍵ペアの名前で参照されます。

暗号化対応アプリケーションの設定

OpenText Certificate Server を設定した後、個々の暗号化対応アプリケーションを設定して、作成したカスタム証明書を使用できるようにする必要があります。設定手順は個々のアプリケーションに固有のものであるため、具体的な手順についてはアプリケーションのマニュアルを参照することをお勧めします。

追加コンポーネントの設定

OpenText Certificate Server には、追加機能を提供するように設定可能な追加コンポーネントが含まれています。

- 720 ページの 「ユーザ証明書の作成」
- 721 ページの 「ルート認証局コンテナの作成」
- 721 ページの 「ルート認証局オブジェクトの作成」
- 722 ページの 「SAS サービスオブジェクトの作成」

ユーザ証明書の作成

1. OpenText Identity Console を起動します。
2. 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] > [**ユーザ証明書の管理 >**] **ユーザ証明書の管理** > [**ユーザ証明書の作成** ＋] の順にクリックします。

この操作により、ユーザ証明書の作成を支援するウィザードが開きます。メッセージに従ってオブジェクトを作成します。ウィザードの各ページに関する特定の情報については、[**ヘルプ**] をクリックします。

ルート認証局コンテナの作成

OpenText eDirectory ツリーの任意の場所に、ルート認証局コンテナを作成できます。

1 OpenText Identity Console を起動します。

2 適切な権利を持つ管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 [**証明書管理**] に移動して、[**ルート認証局管理**] をクリックします。

4 ＋ をクリックして、新しいルート認証局コンテナを作成します。

5 ルート認証局コンテナの [**名前**] と [**コンテキスト**] を指定します。オブジェクトセレクタ ( ) を使用して、ルート認証局コンテナの [**コンテキスト**] を選択します。

6 [**OK**] をクリックします。

注 : 複数のアプリケーションで、ルート認証局コンテナに特定の名前を指定し、OpenText eDirectory ツリーの特定の場所に配置することが必要になる場合があります。OpenText Certificate Server では、ルート認証局コンテナの名前を「ルート認証局」とし、セキュリティコンテナに配置する必要があります。このコンテナ内の証明書は、外部認証局によって署名されたユーザ証明書と、ルート認証局オブジェクトに保存される中間認証局証明書を検証するために使用されます。サーバ証明書と組織認証局証明書は、独自のオブジェクトに保存されている証明書チェーンを使用します。

ルート認証局オブジェクトの作成

ルート認証局オブジェクトは、ルート認証局コンテナにのみ格納できます。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] > [**ルート認証局管理**] > [**ルート認証局**] タブ > [**ルート認証局の作成** ＋] の順にクリックします。

この操作によって、ルート認証局オブジェクトを作成するための [Create a Trusted Root Object] ウィザードが開きます。メッセージに従ってオブジェクトを作成します。ウィザードの各ページに関する特定の情報については、[**ヘルプ**] をクリックします。

注 : ルート認証局オブジェクトには、すべてのタイプの証明書を保存できます (認証局証明書、中間認証局証明書、またはユーザ証明書)。

SAS サービスオブジェクトの作成

SAS サービスオブジェクトは、サーバヘルスチェックの一部として自動的に作成されます。このオブジェクトを手動で作成する必要はありません。手動で作成する必要がある場合、次の手順を実行します。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] > [**SAS サービスオブジェクト**] > [**SAS サービスオブジェクトの作成** +] の順にクリックします。

この操作によって、SAS サービスオブジェクトを作成するための [SAS サービスオブジェクトの作成] ウィザードが開きます。メッセージに従ってオブジェクトを作成します。ウィザードの各ページに関する特定の情報については、? をクリックします。

OpenText Certificate Server の管理

システム管理者は、OpenText Certificate Server で提供される非対称暗号化サービスを維持するためにいくつかのタスクを実行する必要があります。Identity Console を使用して、これらのタスクを実行します。このセクションでは、各タスクの実行に関する簡単な概略と具体的な情報を提供します。

認証局のタスク :

- 725 ページの 「組織の認証局オブジェクトを作成する」
- 725 ページの 「公開鍵証明書の発行」
- 726 ページの 「組織認証局のプロパティの表示」
- 726 ページの 「組織認証局の公開鍵証明書プロパティの表示」
- 726 ページの 「認証局の自己署名証明書プロパティの表示」
- 727 ページの 「組織認証局の自己署名証明書のエクスポート」
- 727 ページの 「組織認証局のバックアップ」
- 729 ページの 「組織認証局の復元」
- 730 ページの 「組織認証局の別のサーバへの移動」
- 731 ページの 「組織認証局の証明書の検証」
- 732 ページの 「組織認証局の削除」
- 732 ページの 「組織認証局のロールオーバー」

サーバ証明書オブジェクトのタスク：

- 734 ページの 「サーバ証明書オブジェクトを作成する」
- 734 ページの 「デフォルトのサーバ証明書オブジェクトを作成する」
- 735 ページの 「サーバ証明書オブジェクトへの公開鍵証明書のインポート」
- 736 ページの 「ルート認証局または公開鍵証明書のエクスポート」
- 737 ページの 「サーバ証明書オブジェクトの削除」
- 738 ページの 「サーバ証明書オブジェクトのプロパティの表示」
- 738 ページの 「サーバ証明書オブジェクトの公開鍵証明書プロパティの表示」
- 739 ページの 「サーバ証明書オブジェクトのルート認証局証明書プロパティの表示」
- 739 ページの 「サーバ証明書オブジェクトのバックアップ」
- 740 ページの 「サーバ証明書オブジェクトの復元」
- 741 ページの 「サーバ証明書オブジェクトとクラスタリング」
- 742 ページの 「サーバ証明書の検証」
- 742 ページの 「ルート認証局または自己署名証明書の取り消し」
- 743 ページの 「別のサーバへのサーバ証明書オブジェクトの移動」
- 743 ページの 「サーバ証明書オブジェクトのキーマテリアルの置換」

ユーザ証明書のタスク：

- 744 ページの 「ユーザ証明書の作成」
- 744 ページの 「ユーザ証明書の一括作成」
- 745 ページの「ユーザオブジェクトへの公開鍵証明書のインポート (秘密鍵は省略可)」
- 746 ページの 「ユーザ証明書のプロパティの表示」
- 746 ページの 「ユーザ証明書のエクスポート」
- 747 ページの 「ユーザ証明書と秘密鍵のエクスポート」
- 747 ページの 「ユーザ証明書の検証」
- 749 ページの 「ユーザ証明書の取り消し」
- 749 ページの 「ユーザ証明書と秘密鍵の削除」

X.509 証明書の自己プロビジョニング：

- 750 ページの 「概要」
- 751 ページの 「ユーザ自己プロビジョニング」
- 752 ページの 「サーバ自己プロビジョニング」
- 753 ページの 「証明書の自己プロビジョニングと証明書の発行タスク」

外部アプリケーションでの OpenText eDirectory 証明書の使用

- 754 ページの 「PKI ヘルスチェック機能」
- 755 ページの 「eDirectory 証明書をエクスポートするための SAS:Service オブジェクトの設定」

ルート認証局オブジェクトのタスク：

- 708 ページの 「ルート認証局コンテナの作成」
- 708 ページの 「ルート認証局オブジェクトの作成」
- 756 ページの 「ルート認証局オブジェクトのプロパティの表示」
- 757 ページの 「ルート認証局オブジェクトの検証」
- 758 ページの 「ルート認証局証明書の取り消し」

証明書取り消しリスト (CRL) のタスク：

- 759 ページの 「手動による CRL コンテナの作成」
- 759 ページの 「CRL コンテナの削除」
- 760 ページの 「CRL 設定オブジェクトの作成」
- 760 ページの 「CRL 設定オブジェクトのアクティブ化」
- 761 ページの 「CRL 設定オブジェクトのプロパティの表示と変更」
- 763 ページの 「CRL 設定オブジェクトの削除」
- 763 ページの 「CRL オブジェクトの作成」
- 764 ページの 「CRL ファイルのエクスポート」
- 765 ページの 「CRL ファイルの置き換え」
- 766 ページの 「CRL オブジェクトのプロパティの表示」
- 766 ページの 「CRL オブジェクトの削除」

OpenText eDirectory タスク：

- 767 ページの 「複数のセキュリティコンテナ、組織認証局、KAP コンテナ、および W0 オブジェクトの解決」
- 768 ページの 「セキュリティコンテナの復元または再作成」
- 768 ページの 「KAP および W0 の復元または再作成」

アプリケーションのタスク

認証局のタスク

- 725 ページの 「組織の認証局オブジェクトを作成する」
- 725 ページの 「公開鍵証明書の発行」
- 726 ページの 「組織認証局のプロパティの表示」
- 726 ページの 「組織認証局の公開鍵証明書プロパティの表示」

- 726 ページの 「認証局の自己署名証明書プロパティの表示」
- 727 ページの 「組織認証局の自己署名証明書のエクスポート」
- 727 ページの 「組織認証局のバックアップ」
- 729 ページの 「組織認証局の復元」
- 730 ページの 「組織認証局の別のサーバへの移動」
- 731 ページの 「組織認証局の証明書の検証」
- 732 ページの 「組織認証局の削除」
- 732 ページの 「組織認証局のロールオーバー」

組織の認証局オブジェクトを作成する

このタスクについては、713 ページの 「組織の認証局オブジェクトを作成する」で説明されています。

公開鍵証明書の発行

このタスクでは、サーバ証明書オブジェクトを認識しない暗号化対応アプリケーション用の証明書を生成できます。

組織認証局は、外部認証局と同じ方法で機能します。つまり、証明書署名要求 (CSR) から証明書を発行する機能があります。ユーザが署名のために CSR を送信してきたら、組織認証局を使用して証明書を発行できます。証明書を要求したユーザは、発行された証明書を取得し、暗号化対応アプリケーションに直接インポートします。

公開鍵証明書を発行するには、次の手順を実行します。

1. OpenText Identity Console を起動します。
2. 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

   このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。
3. OpenText Identity Console のホームページで、[**証明書管理**] タイルを選択し、[**証明書の発行**] をクリックします。
4. [**ファイルの選択 (Chose file)**] ボタンを使用して CSR ファイルを見つけ、ファイルを開き、[**開く (Open)**] をクリックします。
5. 鍵のタイプ、鍵の使用方法、および拡張鍵の使用法を指定し、[**次へ**] をクリックします。
6. 証明書の基本制約を指定して、[**次へ**] をクリックします。
7. サブジェクト名、有効期間、有効開始日と有効期限日、および任意のカスタム拡張機能を指定し、[**次へ**] をクリックします。

8 パラメータシートを確認します。正しい場合は、[完了] をクリックします。正しくない場合は、変更が必要な箇所まで [戻る] をクリックして戻ります。

[完了] をクリックすると、証明書が作成されたというメッセージがダイアログボックスに表示されます。Base64 形式でシステムクリップボードに、Base64 形式のファイルに、またはバイナリ DER 形式のファイルに証明書を保存できます。[詳細] をクリックして、発行された証明書に関する詳細を表示することもできます。

組織認証局のプロパティの表示

任意の OpenText eDirectory オブジェクトで表示できる OpenText eDirectory の権利とプロパティに加え、組織認証局に固有のプロパティ (それに関連付けられた公開鍵証明書や自己署名証明書のプロパティなど) を表示することもできます。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[CA 管理] タイルを選択し、[RSA 証明書] をクリックします。

この操作により、組織認証局のプロパティページが表示されます。これには、全般ページ、詳細ページ、拡張ページなどが含まれます。

4 表示するタブをクリックします。

組織認証局の公開鍵証明書プロパティの表示

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[証明書管理] タイル > [CA 管理] の順に選択します。

この操作により、組織認証局のプロパティページが表示されます。これには、全般ページ、CRL ページ、証明書ページ、その他の OpenText eDirectory 関連ページが含まれます。

4 [閉じる] をクリックします。

認証局の自己署名証明書プロパティの表示

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 ［証明書管理］タイル > ［CA 管理］ > ［自己署名証明書 ECDSA(Self Signed Certificate ECDSA)］の順に移動します。

この操作により、選択した［自己署名証明書 (Self Signed Certificate)］の［証明書の詳細］が表示されます。これには、［全般］、［詳細］、［拡張 (Extensions)］、［CRL 設定 (CRL Configuration)］および［CRL 情報］タブが含まれます。

4 従属認証局の場合、自己署名証明書はありません。

5 ［閉じる］をクリックします。

組織認証局の自己署名証明書のエクスポート

自己署名証明書は、組織認証局の識別情報と、組織認証局によって署名された証明書の有効性を確認するために使用できます。

組織認証局のプロパティページでは、このオブジェクトに関連付けられた証明書とプロパティを参照できます。［自己署名証明書］プロパティページでは、自己署名証明書を、暗号化対応のアプリケーションで使用するファイルにエクスポートできます。

組織の認証局に存在する自己署名証明書は、組織の認証局によって署名された証明書を持つサーバ証明書オブジェクトのルート認証局証明書と同じものです。組織認証局の自己署名証明書をルート認証局として認識するサービスでは、組織認証局によって署名された有効なユーザやサーバを許可できます。

このタスクは、認証局が従属認証局の場合、適用されません。

組織認証局の自己署名証明書をエクスポートするには、次の手順を実行します。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 ［証明書管理］タイル > ［CA 管理］の順に移動して、［自己署名証明書 (Self Signed Certificate)］を選択します。

4 ［エクスポート  ］をクリックして、証明書をエクスポートします。

5 ［OK］をクリックします。

組織認証局のバックアップ

OpenText では、組織認証局のホストサーバで回復不能な障害が発生する場合に備え、組織認証局の秘密鍵と証明書をバックアップしておくことをお勧めします。障害が発生した場合、バックアップファイルを使用して、組織認証局をツリー内の任意のサーバに復元できます。

注 : 組織認証局をバックアップする機能は、NetIQ Certificate Server バージョン 9.0 以上で作成された組織認証局に対してのみ使用できます。証明書サーバの以前のバージョンでは、組織認証局の秘密鍵は、エクスポートできない方法で作成されていました。

バックアップファイルには、認証局の秘密鍵、自己署名証明書、公開鍵証明書、およびこの操作に必要とされるいくつかの他の証明書が含まれています。この情報は、PKCS #12 形式 (PFX とも呼ばれます) で保存されます。

組織認証局が正常に機能しているときに、組織認証局をバックアップする必要があります。

NetIQ Certificate Server 9.0 以降で認証局を完全にバックアップするには、CRL データベースと発行済み証明書データベースをバックアップする必要があります。

その他のプラットフォームでは、これらのデータベースは両方とも、OpenText eDirectory dib ファイルと同じディレクトリにあります。これらの場所のデフォルト値は次のとおりです。

- Windows の場合 : c:\novell\nds\dibfiles
- Linux の場合 : /var/opt/novell/edirectory/data/dib

これらのデフォルト設定は、OpenText eDirectory のインストール時に変更できます。

CRL データベースのバックアップ対象ファイルは、crl.db、crl.01、および crl.rfl ディレクトリです。発行済み証明書データベースのバックアップ対象ファイルは、cert.db、cert.lck、cert.01、および cert.rfl ディレクトリです。

OpenText eDirectory dib ディレクトリは、標準かつ通常バックアップ計画の一部に含める必要があります。

組織認証局をバックアップするには、次の手順を実行します。

1. OpenText Identity Console を起動します。
2. 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

   このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。
3. OpenText Identity Console のホームページで、[**証明書管理**] > [**CA 管理**] の順にクリックします。
4. [**自己署名証明書**] または [**公開鍵証明書**] のいずれかを選択します。証明書は両方とも、バックアップ操作中にファイルに書き込まれます。

   OpenText では、RSA 証明書と ECDSA 証明書の [**自己署名証明書**] を別個に選択することをお勧めします。
5. [**エクスポート** ] をクリックします。

   この操作により、証明書をファイルにエクスポートするためのウィザードが開きます。
6. 秘密鍵のエクスポートを選択して、6 文字以上の英数字で PFX ファイルの暗号化に使用するパスワードを指定し、[**次へ**] をクリックします。
7. [**Save the Exported Certificate to a File**] リンクをクリックし、ファイル名およびバックアップファイルを保存する場所を指定します。
8. [**保存**] をクリックします。

9 [閉じる] をクリックします。

暗号化されたバックアップファイルは、指定した場所に書き込まれます。これで、緊急用に、ファイルを安全な場所に保存する準備ができました。

重要 : エクスポートしたファイルは、バックアップメディアに配置し、安全な場所に保存する必要があります。ファイルの暗号化に使用するパスワードは記憶するか、または必要なときに確実に使用でき、かつ他のユーザにはアクセスできない安全な場所に保存しておく必要があります。

組織認証局の復元

組織認証局オブジェクトが削除されたまたは破損した場合、あるいは組織認証局のホストサーバに回復不能な障害が発生した場合、727 ページの 「組織認証局のバックアップ」で作成したバックアップファイルを使用して、完全に復元することができます。

注 : 組織認証局のバックアップを作成できなかった場合、NICI 2.*x* がサーバにインストールされ、NICI 設定情報のバックアップが作成されているならば、組織認証局を復元できる可能性があります。

NetIQ Certificate Server 9.0 で認証局を完全に復元するには、CRL データベースと発行済み証明書データベースを復元する必要があります。

これらのデータベースは両方とも、OpenText eDirectory dib ファイルと同じディレクトリにあります。これらの場所のデフォルト値は次のとおりです。

- Windows の場合 : c:\novell\nds\dibfiles
- Linux の場合 : /var/opt/novell/edirectory/data/dib

これらのデフォルト値は、OpenText eDirectory のインストール時に変更できます。

CRL データベースの復元対象ファイルは、crl.db、crl.01、および crl.rfl ディレクトリです。発行済み証明書データベースの復元対象ファイルは、cert.db、cert.lck、cert.01、および cert.rfl ディレクトリです。

OpenText eDirectory dib ディレクトリは、標準および通常バックアップ計画の一部に含める必要があります。

組織認証局を復元するには、次の手順を実行します。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 (状況によって実行) 組織認証局オブジェクトが存在する場合は、それを削除する必要があります。

3a Identity Console のホームページで、［CA 管理］をクリックします

3b 組織認証局オブジェクトをブラウズしてクリックします。

3c ［OK］をクリックします。

4 ［CA 管理］タイル > ［サーバ証明書の管理］の順にクリックします。

この操作により、［Create an Organizational Certificate Authority Object］ダイアログボックスと、オブジェクトを作成する対応するウィザードが開きます。

5 作成ダイアログボックスで、組織認証局をホストするサーバおよび組織認証局オブジェクトの名前を指定する必要があります。

6 ［インポート］オプションを選択します。

RSA と ECDSA の両方の証明書を選択します。証明書サーバでは、両方の証明書のサブジェクト名が同じである必要があります。

重要 : 証明書サーバでは、外部自己署名認証局証明書のインポートをサポートしません。ただし、従属認証局証明書をインポートすることはできます。

7 ［次へ］をクリックします。

8 開くダイアログで、［参照］をクリックし、RSA と ECDSA のファイルの名前を選択します。

9 バックアップ実行時に、ファイルを暗号化するために使用したパスワードを入力します。

10 ［OK］をクリックします。

これで組織認証局の秘密鍵と証明書が復元され、認証局が完全に機能するようになります。将来の使用に備え、ここでファイルをもう一度保存できます。

重要 : 必ず、バックアップメディアを保護してください。

組織認証局の別のサーバへの移動

727 ページの 「組織認証局のバックアップ」および 729 ページの 「組織認証局の復元」で説明されているバックアップと復元の手順を利用して、1 つのサーバから別のサーバに組織認証局を移動できます。

NetIQ Certificate Server 3.2 以降で、認証局を完全に移動するには、CRL データベースと発行済み証明書データベースを移動する必要があります。

その他のプラットフォームでは、これらのデータベースは両方とも、OpenText eDirectory dib ファイルと同じディレクトリにあります。これらの場所のデフォルト値は次のとおりです。

- Windows の場合 : c:\novell\nds\dibfiles
- Linux の場合 : /var/opt/novell/edirectory/data/dib

これらのデフォルト設定は、OpenText eDirectory のインストール時に変更できます。

CRL データベースの移動対象ファイルは、crl.db、crl.01、および crl.rfl ディレクトリです。発行済み証明書データベースの移動対象ファイルは、cert.db、cert.lck、cert.01、および cert.rfl ディレクトリです。

1 組織認証局が機能することを確認します
2 組織認証局をバックアップします
3 認証局キーをエクスポートします
4 組織認証局オブジェクトを削除します。
5 両方のサーバで OpenText eDirectory を停止します
6 ソースの cert ファイルと crl ファイルを、rfl ログが含まれるあて先サーバにコピーします
7 両方のサーバで OpenText eDirectory を開始します
8 あて先サーバで組織認証局を再作成します

重要 : 必ず、バックアップメディアを保護してください。

組織認証局の証明書の検証

証明書に問題があることが疑われる場合、または失効していると思われる場合は、OpenText Identity Console を使用して簡単に証明書を検証できます。外部認証局によって発行された証明書を含め、eDirectory ツリー内のすべての証明書を検証できます。

証明書の検証プロセスには、証明書のデータだけでなく、証明書チェーン内のデータに対する複数のチェックが含まれます。証明書チェーンは、ルート認証局証明書と、必要に応じて 1 つ以上の中間認証局の証明書からなります。

証明書チェーン内のすべての証明書が有効であれば、結果は有効になります。現在の時刻が証明書の有効期間内であるか、証明書が取り消されていないか、信頼されている認証局によって署名されているかなど、事前定義された一連の条件を証明書が満たすと、その証明書は有効であると見なされます。CRL 配布ポイント拡張機能または OCSP AIA 拡張機能を使用している証明書に限り、取り消されているかどうかがチェックされます。

証明書チェーン内の 1 つ以上の証明書が無効であると判明した場合、または有効であると断定できない場合、結果は無効になります。無効であると見なされている証明書およびその理由を示す、これらの証明書に関する追加情報が提供されます。理由に関する詳細については、[ヘルプ] をクリックしてください。

証明書を検証するには、次の手順を実行します。

1 OpenText Identity Console を起動します。
2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

   このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。
3 [**証明書管理**] に移動して、[**CA 管理**] をクリックします。
4 [**証明書**] タブで証明書を選択し、[検証 (validate)] ( ) をクリックします。

5 ［検証結果］に［**証明書**］の［**ステータス**］が表示されます。証明書が有効でない場合は、理由が示されます。

6 ［**OK**］をクリックします。

組織認証局の削除

組織認証局オブジェクトの削除は、絶対に必要な場合、またはバックアップから組織認証局を復元する場合にのみ行います (729 ページの 「組織認証局の復元」を参照してください)。オブジェクトを削除する唯一安全な方法は、後で復元できるようにまずバックアップを実行してから行う方法です。

ただし、組織認証局を削除し、復元されないようにする必要がある場合もあります。たとえば、ツリーをマージするとき、結果のツリーには 1 つの組織認証局だけが存在でき、その他の認証局は削除する必要があります。また、組織認証局のホストサーバが修復できないほど破損しているときに、認証局のバックアップまたは NICI 設定が行われていない場合、残されている唯一のオプションは認証局を削除して初めからやり直すことです。

組織認証局オブジェクトを削除するには、次の手順を実行します。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 秘密鍵なしで自己署名証明書をバックアップします。

4 CN=trusted roots.CN=security container で自己署名証明書を使用して、ルート認証局証明書を作成します。次の情報を参照してください :

5 OpenText Identity Console のホームページで、［**証明書管理**］をクリックし、削除する証明書を選択して、［**オブジェクトの削除** 🗑 ］をクリックします。

6 ［**OK**］をクリックします。

組織認証局のロールオーバー

組織認証局 (CA) 証明書を置き換える際に考慮すべき重要な問題が 2 つあります。

- 管理されている証明書のタイプ
- 認証局を置き換える理由

サーバ証明書オブジェクト (KMO) には、サーバの公開鍵証明書と、公開鍵証明書の署名に使用されたルート認証局証明書の両方が含まれます。ユーザ証明書は、ユーザオブジェクトの属性として保存され、それらに署名したルート認証局とはペアになりません。したがって、ルート認証局証明書が置き換えられても、ルート認証局がまだアクセス可能であるため、サーバ証明書は引き続き有効になります。ただしユーザ証明書は、証明書の検証によって確認できるルート認証局コンテナにルート認証局証明書が配置されていない限り、すぐに無効になります。

認証局を交換する 3 つの理由があります。

- 認証局がその有効性の終わりに達した (認証局の有効期限が切れた)。

- 認証局が侵害された。
- その他の何らかの理由で認証局証明書を置き換えたい (強力な鍵が必要、新しいセキュリティポリシーで必要とされる、外部で署名された認証局を使用したいなど)。

認証局の有効期限が切れた場合、認証局が署名した証明書の有効期限も切れます。認証局を交換した後、新しい認証局を使用してそれぞれの署名済み証明書を再作成する必要があります。

認証局が侵害された場合、認証局を交換することによって、古い認証局によって署名されたユーザ証明書を無効にします。OpenText Identity Console で［デフォルト証明書の作成］タスクを実行することによって、簡単に置き換えることができます。証明書サーバによってデフォルトで作成されたすべての証明書は、新しい認証局で再作成されます。カスタマイズした方法で作成したすべての証明書は、新しい認証局を使用して手動で再作成する必要があります。デフォルトの証明書を作成する方法の詳細については、734 ページの 「デフォルトのサーバ証明書オブジェクトを作成する」を参照してください。

何らかの理由で認証局を再作成する場合は、ルート認証局コンテナにルート認証局証明書を保存することによって、ユーザ証明書を再作成するのに都合の良いときまでユーザ証明書を有効なままにしておきます。

ルート認証局証明書を置き換えるには、次の手順を実行します。

1. 後で復元しなければならない場合に備え、現在の認証局をバックアップします。
2. 証明書の作成に使用されたルート認証局証明書をエクスポートします。古いシステムではほとんどの場合、これは自己署名証明書でした。

   最近では、認証局証明書を外部で署名する機能が追加されました。認証局が外部で署名される場合、公開鍵証明書をエクスポートします。チェーン内のすべての証明書は、ルート認証局コンテナに独自のオブジェクトを持つ必要があります。

   認証局が侵害されていない場合は、ルート認証局コンテナにルート認証局証明書を作成します。そうすることにより、置き換えられるまでユーザ証明書は有効になります。
3. 古い認証局を削除します。組織認証局の削除方法については、732 ページの 「組織認証局の削除」を参照してください。
4. 新しい認証局を作成します。新しい組織認証局の作成方法については、725 ページの「組織の認証局オブジェクトを作成する」を参照してください。
5. 必要に応じて、OpenText Identity Console で［デフォルト証明書の作成］タスクを使用してサーバ証明書を再作成します。OpenText Identity Console を使用してデフォルト証明書を作成する方法については、734 ページの 「デフォルトのサーバ証明書オブジェクトを作成する」を参照してください。

   デフォルトでは生成されないその他のサーバ証明書を再作成します。
6. 必要に応じて、OpenText Identity Console で［ユーザ証明書の作成］タスクを使用するかユーザプロパティを表示して、証明書を表示し、［**新規**］をクリックしてユーザ証明書を再作成します。

サーバ証明書オブジェクトのタスク

- 734 ページの 「サーバ証明書オブジェクトを作成する」
- 734 ページの 「デフォルトのサーバ証明書オブジェクトを作成する」

- 735 ページの 「サーバ証明書オブジェクトへの公開鍵証明書のインポート」
- 736 ページの 「ルート認証局または公開鍵証明書のエクスポート」
- 737 ページの 「サーバ証明書オブジェクトの削除」
- 738 ページの 「サーバ証明書オブジェクトのプロパティの表示」
- 738 ページの 「サーバ証明書オブジェクトの公開鍵証明書プロパティの表示」
- 739 ページの 「サーバ証明書オブジェクトのルート認証局証明書プロパティの表示」
- 739 ページの 「サーバ証明書オブジェクトのバックアップ」
- 740 ページの 「サーバ証明書オブジェクトの復元」
- 741 ページの 「サーバ証明書オブジェクトとクラスタリング」
- 742 ページの 「サーバ証明書の検証」
- 742 ページの 「ルート認証局または自己署名証明書の取り消し」
- 743 ページの 「別のサーバへのサーバ証明書オブジェクトの移動」
- 743 ページの 「サーバ証明書オブジェクトのキーマテリアルの置換」

サーバ証明書オブジェクトを作成する

このタスクについては、719 ページの 「サーバ証明書オブジェクトを作成する」で説明されています。

デフォルトのサーバ証明書オブジェクトを作成する

証明書サーバをインストールすると、デフォルトのサーバ証明書オブジェクトが作成されます。

- SSL CertificateDNS - *server_name*
- サーバで設定した IP アドレスごとの証明書 (IPAG*xxx.xxx.xxx.xxx - server_name*)
- サーバで設定した DNS 名ごとの証明書 (DNSAG*www.example.com - server_name*)

注 : OpenText eDirectory では、SSL CertificateIP は自動的に作成されません。SSL 証明書 DNS には、［件名の代替名］にリストされているすべての IP が含まれます。PKI OpenText Identity Console アプリケーションを使用してデフォルト証明書を作成または修復しようとしても、デフォルトでは SSL CertificateIP 証明書は作成または修復されません。ただし、OpenText Identity Console の［証明書管理］タイルに、デフォルトの動作を上書きして、強制的に SSL CertificateIP 証明書の作成 / 修復を実行することを選択できるチェックボックスがあります。

OpenText eDirectory 9.0 以降では、組織認証局に ECDSA 証明書があれば、ECDSA 証明書が自動的に作成されます。

何らかの理由でこれらの証明書が破損しているまたは無効になった場合、あるいは既存のデフォルト証明書を置換する場合は、次の手順に従って［Create Default Server Certificates］ウィザードを使用します。

1 OpenText Identity Console を起動します。
2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 ［**証明書管理**］タイルに移動して、［**デフォルト証明書**］をクリックします。
4 デフォルト証明書を作成する 1 つまたは複数のサーバを選択し、［**次へ**］をクリックします。
5 既存のデフォルトサーバ証明書を上書きする場合は［**はい**］を選択し、既存のデフォルトサーバ証明書が有効でないときにのみ上書きする場合は［**いいえ**］を選択します。
6 (単一サーバのみ) 既存のデフォルト IP アドレスを使用する場合は、そのオプションを選択します。別の IP アドレスを使用する場合は、そのオプションを選択し新しい IP アドレスを指定します。
7 (単一サーバのみ) 既存の DNS アドレスを使用する場合は、そのオプションを選択します。別の DNS アドレスを使用する場合は、そのオプションを選択し新しい DNS アドレスを指定します。
8 ［**次へ**］をクリックします。
9 デフォルト証明書に適用可能なパラメータサマリを確認し、［**完了**］をクリックします。

サーバ証明書オブジェクトの作成をより細かく制御する場合は、サーバ証明書オブジェクトを手動で作成できます。詳細については、719 ページの 「サーバ証明書オブジェクトを手動で作成する」を参照してください。

サーバ証明書オブジェクトへの公開鍵証明書のインポート

証明書署名要求 (CSR) を作成し、認証局 (CA) が署名済み公開鍵証明書を返してきた後、公開鍵証明書をインポートします。このタスクは、外部認証局の署名オプションとカスタムオプションを使用して、サーバ証明書オブジェクトを作成したときに適用されます。

認証局が証明書を返す方法はいくつかあります。通常、認証局によって、それぞれ 1 つずつ証明書を格納した 1 つ以上のファイルか、複数の証明書を格納した 1 つのファイルが返されます。これらのファイルは、バイナリの DER エンコードファイル (.der、.cer、.crt、.p7b) か、テキストの Base64 エンコードファイル (.cer、.b64) になります。

ファイルに複数の証明書が含まれる場合、サーバ証明書オブジェクトにインポートできるようにするため PKCS #7 形式である必要があります。さらに、このファイルには、オブジェクトにインポートされるすべての証明書 (ルートレベルの認証局証明書、すべての中間証明書、サーバ証明書) を含める必要があります。

証明書に署名した結果として認証局が複数のファイルを返す場合、各ファイルにはサーバ証明書オブジェクトにインポートする必要がある別個の証明書が格納されます。3 つ以上のファイル (1 つのルートレベルの認証局、1 つ以上の中間認証局、1 つのサーバ証明書) がある場合、サーバ証明書オブジェクトにインポートするため、これらのファイルを PKCS #7 ファイルにまとめる必要があります。

PKCS #7 ファイルを作成する方法はいくつかあります。1 つの方法は、Internet Explorer にすべての証明書をインポートする方法です。それらの証明書をインポートした後、Internet Explorer を使用して、サーバ証明書と証明書チェーン内のすべての証明書を PKCS #7 形式でエクスポートすることができます。この方法の詳細については、921 ページの 「外部認証局」を参照してください。

一部の認証局は、ルートレベルの認証局証明書をサーバ証明書と一緒に返しません。それらのルートレベルの認証局証明書を取得するには、認証局プロバイダに直接問い合わせるか、テクニカルサポートに連絡します。

サーバ証明書オブジェクトに証明書をインポートするには、次の手順を実行します。

1. OpenText Identity Console を起動します。
2. 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

   このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。
3. OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**サーバ証明書の管理**] の順にクリックします。
4. 変更するサーバ証明書オブジェクトの横にある [**インポート**] をクリックします。
5. 証明書データファイルをブラウズして選択します。
6. ルート認証局データファイルをブラウズして選択します。

   1 つのファイルにすべての証明書が含まれている場合は、このフィールドを空白のままにします。
7. [**OK**] をクリックします。

ルート認証局または公開鍵証明書のエクスポート

次の理由から、証明書をファイルにエクスポートすることがあります。

- クライアント (インターネットブラウザなど) がそのファイルを使用して、暗号化対応アプリケーションによって送信された証明書チェーンを検証できるようにするため。
- ファイルのバックアップコピーを提供するため。

DER エンコード (.der) と Base64 エンコード (.b64) の 2 つのファイル形式で証明書をエクスポートできます。.crt 拡張子も、DER エンコード証明書に使用できます。暗号化対応アプリケーションに証明書を直接貼り付けることができるように、システムクリップボードに Base64 形式でエクスポートすることもできます。

ルート認証局または公開鍵証明書をエクスポートするには、次の手順を実行します。

1. OpenText Identity Console を起動します。
2. 適切な権利を持ったユーザとして OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**サーバ証明書**] の順にクリックします。

4 特定のアプリケーションが使用するように設定されているサーバ証明書オブジェクトを選択します。

5 [**サーバ証明書のエクスポート**] をクリックします。

この操作により、証明書をファイルにエクスポートするためのウィザードが開きます。

6 ドロップダウンリストを使用して、エクスポートする証明書を指定します。

7 秘密鍵をエクスポートするかどうかを選択し、パスワードを入力し、パスワードを再入力します。

8 エクスポート形式 (バイナリの DER またはテキストエンコードの Base64) を選択し、[**OK**] をクリックします。

9 [**Save the Exported Certificate to a File**] をクリックし、ファイルを任意の場所に保存します。

10 [**OK**] をクリックします。

11 必要に応じてファイルを使用します。

たとえば、Internet Explorer ブラウザにルート認証局証明書をインストールする場合は、ファイルをダブルクリックします。この操作により、ルート認証局として認証局を受諾するウィザードが開始します。ルート認証局として認証局を受諾すると、ブラウザは、この認証局によって発行された証明書を使用するサービスで SSL 接続を自動的に受け入れるようになります。

サーバ証明書オブジェクトの削除

秘密鍵が侵害されていると思われる場合、鍵ペアを使用する必要がなくなった場合、またはサーバ証明書オブジェクトのルート認証局を信頼できなくなった場合、サーバ証明書オブジェクトを削除する必要があります。

重要 : サーバ証明書オブジェクトを削除すると、以前にバックアップが作成されていない限り復元できません。オブジェクトを削除する前に、どの暗号化対応アプリケーションでもそのオブジェクトを使用する必要がないことを確認します。サーバ証明書オブジェクトを再作成することはできますが、古いオブジェクトを参照するすべてのアプリケーションを再設定する必要があります。

サーバ証明書オブジェクトを削除するには、次の手順を実行します。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**サーバ証明書**] の順にクリックします。

4 削除するサーバ証明書オブジェクトを選択します。

5 [削除 🗑] > [OK] の順にクリックします。

サーバ証明書オブジェクトのプロパティの表示

任意の OpenText eDirectory オブジェクトで表示できる OpenText eDirectory の権利とプロパティに加え、サーバ証明書オブジェクトに固有のプロパティ (存在する場合、それに関連付けられた公開鍵証明書やルート認証局証明書のプロパティも含みます) を表示することもできます。

サーバ証明書オブジェクトのプロパティを表示するには、次の手順を実行します。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[証明書管理] タイル > [サーバ証明書の管理] の順にクリックします。

4 表示するサーバ証明書オブジェクトのニックネームをクリックします。

5 全般、詳細、拡張などのサーバ証明書オブジェクトのプロパティを表示できます。

6 [キャンセル] をクリックします。

サーバ証明書オブジェクトの公開鍵証明書プロパティの表示

サーバ証明書オブジェクトの公開鍵証明書プロパティを表示するには、次の手順を実行します。

1 OpenText Identity Console を起動します。

2 適切な権利を持ったユーザとして OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[証明書管理] > [サーバ証明書の管理] の順にクリックします。

4 表示するサーバ証明書オブジェクトを選択して、[チェーンの表示] をクリックします。

- 公開鍵証明書がインストールされている場合、プロパティページには、サブジェクト名の完全識別名、発行者の完全識別名、および公開鍵証明書の有効期限が表示されます。
- 公開鍵証明書がまだインストールされていない場合、プロパティページにそのことが示されます。

5 公開鍵証明書に関する追加情報を表示するには、証明書のニックネームをクリックして [詳細] ページを表示します。

[詳細] ページには、公開鍵証明書に含まれている情報が表示されます。

6 閉じるをクリックします。

サーバ証明書オブジェクトのルート認証局証明書プロパティの表示

サーバ証明書オブジェクトのルート認証局証明書プロパティを表示するには、次の手順を実行します。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、［**証明書管理**］タイル > ［**ルート認証局管理**］の順にクリックします。

4 表示するサーバ証明書オブジェクトをブラウズして選択します。

5 ［**OK**］をクリックします。

6 ［**ルート認証局の証明書**］をクリックします。

- ルート認証局証明書がインストールされている場合、プロパティページには、サブジェクト名の完全識別名、発行者の完全識別名、およびルート認証局証明書の有効期限が表示されます。
- ルート認証局証明書がまだインストールされていない場合、プロパティページにそのことが示されます。

7 証明書チェーンを表示するには、証明書のニックネームの前にあるプラス記号 (+) をクリックしてビューを展開します。

8 ルート認証局証明書に関する追加情報を表示するには、証明書のニックネームをクリックして［詳細］ページを表示します。

［詳細］ページには、ルート認証局証明書に含まれている情報が表示されます。

9 ［**閉じる**］> ［**キャンセル**］をクリックします。

サーバ証明書オブジェクトのバックアップ

OpenText Certificate Server では、サードパーティ認証局によって署名された証明書をサーバ証明書オブジェクトに保存することができます。多くの場合、これらの証明書にはかなりの費用がかかります。残念ながら、証明書を保有するサーバで回復不能なエラーが発生すると、サーバ証明書オブジェクトは使用できなくなります。そのような障害から守るために、外部認証局よって署名されたサーバ証明書とそれに関連付けられている秘密鍵をバックアップできます。障害が発生した場合、バックアップファイルを使用して、サーバ証明書オブジェクトをツリー内の任意のサーバに復元できます。

バックアップファイルには、サーバの秘密鍵、公開鍵証明書、ルート認証局証明書、および保存されているすべての中間認証局証明書が含められます。この情報は、PKCS #12 形式 (PFX とも呼ばれます) で保存されます。

サーバ証明書オブジェクトが正常に機能しているときに、サーバ証明書オブジェクトをバックアップする必要があります。

サーバ証明書オブジェクトをバックアップするには、次の手順を実行します。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**サーバ証明書**] の順にクリックします。

4 ルート認証局証明書または公開鍵証明書のいずれかをクリックします。証明書は両方とも、バックアップ操作中にファイルに書き込まれます。

5 バックアップするサーバ証明書オブジェクトのチェックボックスをオンにして、[**エクスポート**] をクリックします。

この操作により、証明書をファイルにエクスポートするためのウィザードが開きます。

6 チェックボックスで秘密鍵をエクスポートするかどうかを選択します。

7 6 文字以上の英数字で PFX ファイルの暗号化で使用するパスワードを指定します。パスワードを再入力します。

8 [**エクスポート形式**] を選択し、[**OK**] をクリックします。

9 [**エクスポートされた証明書ファイルの保存 (Save the Exported Certificate File)**] をクリックします。ファイル名とバックアップファイルの場所を選択します。

10 [**閉じる**] をクリックします。

暗号化されたバックアップファイルは、指定した場所に書き込まれます。これで、緊急用に、ファイルを安全な場所に保存する準備ができました。

重要 : エクスポートしたファイルは、バックアップメディアに配置し、安全な場所に保存する必要があります。ファイルの暗号化に使用するパスワードは記憶するか、または必要なときに確実に使用でき、かつ他のユーザにはアクセスできないボールトに保存しておく必要があります。

サーバ証明書オブジェクトの復元

サーバ証明書オブジェクトが削除されたまたは破損した場合、あるいはサーバ証明書オブジェクトを所有するサーバで回復不能な障害が発生した場合、739 ページの 「サーバ証明書オブジェクトのバックアップ」で作成したバックアップファイルを使用して、完全に復元することができます。

サーバ証明書オブジェクトのバックアップを作成できなかった場合、NICI 2.x がサーバにインストールされていて、NICI 設定情報のバックアップが作成されていれば、そのサーバ証明書オブジェクトを使用できる可能性があります。NICI 設定ファイルのバックアップおよび復元方法については、『*NICI Administration Guide*』の「Backing Up and Restoring NICI (https://www.netiq.com/documentation/nici27x/nici_admin_guide/data/bwf6d4c.html)」セクションを参照してください。

サーバ証明書オブジェクトを復元するには、次の手順を実行します。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 古いサーバ証明書オブジェクトを削除します。

4 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**サーバ証明書**] の順にクリックします。

この操作によって、オブジェクトを作成する [Create a Server Certificate] ウィザードが開きます。

5 このウィザードで、サーバ証明書オブジェクトを所有するサーバを指定し、サーバ証明書の証明書ニックネームを指定します。サーバには、NetIQ Certificate Server 2.21 以降のバージョンがインストールされ、実行されている必要があります。

6 [**インポート**] オプションを選択し、[**次へ**] をクリックします。

7 バックアップファイルをブラウズして選択し、バックアップファイルのパスワードを入力して、[**次へ**] をクリックします。

これで、サーバの秘密鍵と証明書が復元され、サーバ証明書オブジェクトは完全に機能するようになります。必要に応じて、将来の使用に備え、再度バックアップファイルを保存できます。

重要 : 必ず、バックアップメディアを保護してください。

サーバ証明書オブジェクトとクラスタリング

クラスタ化された環境でサーバ証明書オブジェクトを設定し、サーバ証明書オブジェクトを使用する暗号化対応アプリケーションがそれらに常にアクセスできるようにできます。サーバ証明書オブジェクトのバックアップと復元機能を使用して、オブジェクトのキーマテリアルをクラスタ内の 1 つのノードからすべてのノードに複製できます。外部認証局によって署名されたキーマテリアルにこのプロセスを使用することによって、クラスタ内のすべてのノードのために新しいキーマテリアルを要求するのではなく 1 つのサーバ証明書のキーマテリアルを複製できるため、コストを削減できます。

クラスタ環境で機能するようにサーバ証明書を設定するには、次の手順を実行します。

1 組織認証局または任意の外部認証局を使用して、クラスタ内のサーバにサーバ証明書を作成します。詳細については、719 ページの 「サーバ証明書オブジェクトを作成する」を参照してください。

サーバ証明書オブジェクトを作成する場合、証明書のサブジェクト名の一般名 (CN) 部分はサービス固有の IP または DNS 名にする必要があります。そうしないと、URL の IP または DNS 名が証明書のものと一致していないことを示すブラウザ警告メッセージが表示されます。

複数のサービスにそれぞれ異なる IP または DNS アドレスがある場合、各サービスに対してサーバ証明書を作成する必要があります。

2 クラスタ内の残りのすべてのサーバで、ステップ 1 で作成したのと同じ鍵ペア名でサーバ証明書オブジェクトを作成することによって、このサーバ証明書オブジェクトのキーマテリアルをバックアップおよび復元します。

詳細については、739 ページの 「サーバ証明書オブジェクトのバックアップ」を参照してください。

サーバ証明書の検証

証明書に問題があることが疑われる場合、または失効していると思われる場合は、OpenText Identity Console を使用して簡単に証明書を検証できます。外部認証局によって発行された証明書を含め、OpenText eDirectory ツリー内のすべての証明書を検証できます。

証明書の検証プロセスには、証明書のデータだけでなく、証明書チェーン内のデータに対する複数のチェックが含まれます。証明書チェーンは、ルート認証局証明書と、必要に応じて 1 つ以上の中間認証局の証明書からなります。

証明書チェーン内のすべての証明書が有効であれば、結果は有効になります。現在の時刻が証明書の有効期間内であるか、証明書が取り消されていないか、信頼されている認証局によって署名されているかなど、事前定義された一連の条件を証明書が満たすと、その証明書は有効であると見なされます。CRL 配布ポイント拡張機能または OCSP AIA 拡張機能を使用している証明書に限り、取り消されているかどうかがチェックされます。

証明書チェーン内の 1 つ以上の証明書が無効であると判明した場合、または有効であると断定できない場合、結果は無効になります。無効であると見なされている証明書およびその理由を示す、これらの証明書に関する追加情報が提供されます。理由に関する詳細については、[ヘルプ] をクリックしてください。

証明書を検証するには、次の手順を実行します。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**サーバ証明書**] の順にクリックします。

4 検証するサーバ証明書オブジェクトを選択します。

5 [**サーバ証明書の検証** ] をクリックします。

[**証明書ステータス**] フィールドに、証明書のステータスが表示されます。証明書が有効でない場合は、理由が示されます。

ルート認証局または自己署名証明書の取り消し

鍵または認証局が侵害されている場合、証明書が別の証明書によって置き換えられてしまった場合、証明書が CRL から削除された場合、操作が中止された場合など、さまざまな理由で証明書を取り消す必要があると判断することがあるかもしれません。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 ［**証明書管理**］タイルに移動し、［**サーバ証明書の管理**］をクリックします。

4 証明書を選択してから、［ **サーバ証明書の取り消し**］をクリックします。

［証明書の取り消し］ページが開きます。メッセージに従って証明書を取り消します。

5 ［**OK**］をクリックして、証明書の取り消しを確認します。

別のサーバへのサーバ証明書オブジェクトの移動

739 ページの 「サーバ証明書オブジェクトのバックアップ」および740 ページの 「サーバ証明書オブジェクトの復元」で説明されているバックアップと復元の手順を利用して、1 つのサーバから別のサーバにサーバ証明書オブジェクトを移動できます。

1 サーバ証明書オブジェクトが機能することを確認します。

2 サーバ証明書オブジェクトをバックアップします。

3 目的のサーバに、サーバ証明書オブジェクトを復元します。

重要 : 必ず、バックアップメディアを保護してください。

サーバ証明書オブジェクトのキーマテリアルの置換

サーバ証明書オブジェクト内の秘密鍵と証明書を置き換えることができます。それらを置き換える場合、サーバ証明書オブジェクトのバックアップ中に作成された、内部で生成された PFX ファイルのみを使用するようにします。秘密鍵、サーバ証明書、および完全な証明書チェーンが含まれている場合は、外部で生成された PFX ファイルも使用できます。ファイル内の鍵と証明書は、必ずしもオブジェクトのものと一致している必要はありません。ファイル内のデータによって、オブジェクトの鍵と証明書が上書きされます。

サーバ証明書オブジェクト内の秘密鍵と証明書の置換は、慎重に行う必要があります。鍵と証明書がオブジェクトのものと正確に一致しない場合、現在のサーバ証明書オブジェクトを削除して新規に作成するのと同じことになります。オブジェクトを削除することによって生じる結果の詳細については、739 ページの 「サーバ証明書オブジェクトのバックアップ」セクションを参照してください。

鍵および証明書がオブジェクトのものと一致する場合、Secure Authentication Services (SAS) によって使用されるいくつかの属性が再生成される以外は、キーマテリアルの置換によって生じる影響はありません。

サーバ証明書オブジェクトのキーマテリアルを置換するには、次の手順を実行します。

1 万一に備えて、秘密鍵と一緒にサーバ証明書オブジェクトをバックアップします。詳細については、739 ページの 「サーバ証明書オブジェクトのバックアップ」を参照してください。

2 OpenText Identity Console を起動します。

3 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

4 ［**証明書管理**］に移動して、［**サーバ証明書の管理**］をクリックします。

5 変更するサーバ証明書オブジェクトを選択します。

6 ［ **サーバ証明書の置き換え**］をクリックします。

［サーバ証明書の置き換え］ページが開きます。

7 ［**ファイル名**］を選択して、［**パスワード**］を入力します。

8 ［**OK**］をクリックします。

いずれのページからも操作を開始できます。これらの操作により、証明書、秘密鍵、および証明書チェーン内のその他の証明書が置き換えられます。

これで、サーバの秘密鍵と証明書が置換されました。サーバ証明書は完全に機能するようになります。必要に応じて、将来の使用に備え、再度バックアップファイルを保存します。

重要 : 必ず、バックアップメディアを保護してください。

ユーザ証明書のタスク

- 744 ページの 「ユーザ証明書の作成」
- 744 ページの 「ユーザ証明書の一括作成」
- 745 ページの「ユーザオブジェクトへの公開鍵証明書のインポート (秘密鍵は省略可)」
- 746 ページの 「ユーザ証明書のプロパティの表示」
- 746 ページの 「ユーザ証明書のエクスポート」
- 747 ページの 「ユーザ証明書と秘密鍵のエクスポート」
- 747 ページの 「ユーザ証明書の検証」
- 749 ページの 「ユーザ証明書の取り消し」
- 749 ページの 「ユーザ証明書と秘密鍵の削除」

ユーザ証明書の作成

このタスクについては、720 ページの 「ユーザ証明書の作成」で説明されています。

ユーザ証明書の一括作成

この機能により、一連の操作で、同時に複数のユーザのユーザ証明書を作成できます。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**ユーザ証明書の管理**] の順にクリックします。

この操作により、ユーザ証明書の作成を支援するウィザードが開きます。

4 ユーザ証明書を作成するすべてのユーザをブラウズして選択します。

5 各ユーザの証明書を作成するウィザードの指示に従います。ウィザードの各ページに関する特定の情報については、⃝をクリックします。

ユーザオブジェクトへの公開鍵証明書のインポート (秘密鍵は省略可)

ユーザオブジェクト (たとえば、サードパーティの認証局によって署名された証明書) に、任意の公開鍵証明書をインポートできます。この証明書は、次の 2 種類のファイルのいずれかとして表示されます。

- **DER:** 公開鍵証明書のみが含まれます。
- **PFX または PKCS #12:** 公開鍵証明書と秘密鍵が含まれます。

インポートされると、証明書はユーザオブジェクトに保存され、使用可能な証明書のリストに表示されます。

注 : PKCS #12 証明書をインポートすると、公開鍵証明書と秘密鍵のみが、ユーザオブジェクトに保存されます。他の証明書は保存されません。ユーザの証明書チェーン内の他の証明書は、CN=Trusted Roots.CN=Security コンテナに保存されるはずです (チェーン内の各証明書の新しいルート認証局オブジェクトを作成します)。

ユーザオブジェクトに公開鍵証明書をインポートするには、次の手順を実行します。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**ユーザ証明書の管理**] の順にクリックします。

4 公開鍵証明書をインポートするユーザオブジェクトをブラウズして選択します。

5 [**ユーザ証明書の作成** ＋] をクリックします。

6 ユーザ証明書のニックネームを指定します。

ニックネームは固有のもので、証明書を特定するために役立つものである必要があります。[**証明書ニックネーム**] フィールドには最大 64 文字まで入力できます。

7 [インポート] 作成方法を選択し、[**次へ**] をクリックします。

8 インポートする証明書をブラウズして選択し、[**OK**] をクリックします。

9 (状況によって実行) 秘密鍵と一緒に証明書をインポートする場合、秘密鍵のパスワードを入力し、[**次へ**] をクリックします。

10 [完了] をクリックします。

これでユーザオブジェクトに証明書が保存され、このユーザが使用できる証明書のリストにその証明書が表示されます。

ユーザ証明書のプロパティの表示

任意の OpenText eDirectory オブジェクトで表示できる OpenText eDirectory の権利とプロパティに加え、ユーザ証明書に固有のプロパティ (発行者、証明書のステータス、秘密鍵のステータス、検証期間など) を表示することもできます。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**ユーザ証明書の管理**] の順にクリックします。

4 証明書のニックネームをクリックして証明書の詳細を表示します。

5 確認し終わったら、[**閉じる**] をクリックします。

ユーザ証明書のエクスポート

別のユーザとセキュリティで保護された電子メールを交換するには、まず、そのユーザの公開鍵証明書を入手する必要があります。証明書を入手する 1 つの方法は、OpenText Identity Console を使用して証明書をエクスポートすることです。あるいは、LDAP や電子メールを使用して、相手の証明書を入手することもできます。

自分または他のユーザの公開鍵証明書をエクスポートするには、次の手順に従います。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 [**証明書管理**] タイルに移動して、[**ユーザ証明書の管理**] をクリックします。

4 証明書を選択してから、[**エクスポート**] をクリックします。

ユーザ証明書をファイルにエクスポートするためのページが開きます。証明書を所有するユーザとしてログインしている場合は、「747 ページの 「ユーザ証明書と秘密鍵のエクスポート」」を参照してください。

5 秘密鍵をエクスポートする場合は、[**秘密鍵のエクスポート**] チェックボックスをオンにします。

6 パスワードを入力し、パスワードを再入力します。

7 秘密鍵をエクスポートしない場合は、エクスポート形式を選択してから [**次へ**] をクリックします。

8 [Save the Exported Certificate to a File] をクリックし、ファイルを任意の場所に保存します。

9 [OK] をクリックします。

ユーザ証明書と秘密鍵のエクスポート

セキュリティで保護された電子メール、認証、または暗号化の証明書を使用するには、秘密鍵と証明書の両方が、暗号化対応アプリケーションで使用できるようになっていなければなりません。ユーザ証明書と秘密鍵をエクスポートし、アプリケーションで使用できるよう、そのアプリケーションがアクセスできる場所に配置する必要があります。

ユーザのオブジェクトに含まれる秘密鍵は、そのユーザに属しています。その秘密鍵をエクスポートできるのは、そのユーザとしてログインした場合のみです。その他のユーザは、ネットワーク管理者であっても、他のユーザの秘密鍵をエクスポートする権利はありません。

自分の秘密鍵と証明書をエクスポートするには、次の手順に従います。

1 OpenText Identity Console を起動します。

2 証明書を所有するユーザとして、OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[証明書管理] タイル > [ユーザ証明書] の順にクリックします。

4 証明書を選択してから、[エクスポート] をクリックします。

これにより、ユーザ証明書をファイルにエクスポートする手順を示すウィザードが開きます。

5 [秘密鍵のエクスポート] チェックボックスをオンにします。

6 パスワードを入力し、パスワードを再入力します。

7 閉じるをクリックします。

暗号化されたファイルが、指定した場所に書き込まれます。これで、暗号化対応アプリケーションにインポートできる状態になります。

重要 : バックアップとして使用できるように、エクスポートしたファイルを保持しておくことができます。その場合は、セキュアな場所に保管する必要があります。ファイルの暗号化に使用するパスワードは記憶するか、または必要なときに確実に使用でき、かつ他のユーザにはアクセスできない安全な場所に保存しておく必要があります。

ユーザ証明書の検証

証明書に問題があることが疑われる場合、または失効していると思われる場合は、OpenText Identity Console を使用して簡単に証明書を検証できます。外部認証局によって発行された証明書を含め、OpenText eDirectory ツリー内のすべての証明書を検証できます。

証明書の検証プロセスには、証明書のデータだけでなく、証明書チェーン内のデータに対する複数のチェックが含まれます。証明書チェーンは、ルート認証局証明書と、必要に応じて 1 つ以上の中間認証局の証明書からなります。

証明書チェーン内のすべての証明書が有効であれば、結果は有効になります。現在の時刻が証明書の有効期間内であるか、証明書が取り消されていないか、信頼されている認証局によって署名されているかなど、事前定義された一連の条件を証明書が満たすと、その証明書は有効であると見なされます。CRL 配布ポイント拡張機能または OCSP AIA 拡張機能を使用している証明書に限り、取り消されているかどうかがチェックされます。

証明書チェーン内の 1 つ以上の証明書が無効であると判明した場合、または有効であると断定できない場合、結果は無効になります。このような場合は、無効と見なされた証明書とその理由に関する追加情報が提供されます。理由に関する詳細については、[ヘルプ] をクリックしてください。

証明書を検証するには、次の手順を実行します。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**ユーザ証明書**] の順にクリックします。

4 検証するユーザ証明書を選択します。

5 [**検証** ] をクリックします。

[**証明書ステータス**] フィールドに、証明書のステータスが表示されます。証明書が有効でない場合は、理由が示されます。

注 : ユーザ証明書にサードパーティの認証局による署名が付けられている場合、証明書チェーンがセキュリティコンテナ内のルート認証局コンテナに入っていなければ (CN=Trusted Roots.CN=Security)、証明書を正常に検証できません。通常、証明書チェーンは 1 つのルートレベルの認証局で構成されるか、中間認証局とルートレベルの認証局で構成されます。ルート認証局コンテナの名前は「ルート認証局」であること、そして証明書チェーンに含まれる各証明書が、それぞれ独自のルート認証局オブジェクトに格納されていることが必要です。ルート認証局コンテナおよびルート認証局オブジェクトの作成方法については、721 ページの 「ルート認証局コンテナの作成」および 721 ページの 「ルート認証局オブジェクトの作成」を参照してください。

外部認証局によって署名されたユーザ証明書または中間認証局の証明書を検証する場合は、外部認証局の証明書がルート認証局オブジェクトに格納されていなければ、証明書を正常に検証できません。ルート認証局オブジェクトは、「ルート認証局」という名前のルート認証局コンテナ内にあること、そしてそのルート認証局コンテナがセキュリティコンテナに含まれていることが必要です。

ユーザ証明書の取り消し

鍵または認証局が侵害されている場合、証明書が別の証明書によって置き換えられてしまった場合、証明書が CRL から削除された場合など、さまざまな理由で証明書を取り消す必要があると判断することがあるかもしれません。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**ユーザ証明書**] の順にクリックします。

4 取り消すユーザ証明書を選択します。

5 [**取り消し** ] をクリックします。

この操作により、[証明書を取り消す] ウィザードが起動します。メッセージに従って証明書を取り消します。

6 [**OK**] をクリックします。

ユーザ証明書と秘密鍵の削除

ユーザ証明書が無効になった場合、または何らかの形での秘密鍵のセキュリティ侵害が疑われる場合は、ユーザ証明書と秘密鍵を削除する必要があります。

ユーザ証明書と秘密鍵を削除するには、その前に、ユーザ証明書を取り消さなければなりません。詳細については、749 ページの 「ユーザ証明書の取り消し」を参照してください。

ユーザ証明書と秘密鍵を削除するには、次の手順に従います。

1 OpenText Identity Console を起動します。

2 証明書を所有するユーザとして、または適切な権利を持つ管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**ユーザ証明書**] の順にクリックします。

4 証明書を削除するユーザオブジェクトをブラウズして選択します。

5 削除するユーザ証明書を選択します。

6 **削除** をクリックします。

X.509 証明書の自己プロビジョニング

このセクションでは、X.509 の自己プロビジョニング機能について説明します。

- 750 ページの 「概要」
- 751 ページの 「ユーザ自己プロビジョニング」

- 752 ページの 「サーバ自己プロビジョニング」
- 753 ページの 「証明書の自己プロビジョニングと証明書の発行タスク」

概要

X.509 認定を作成するときには、認証局 (CA) が証明書を発行する前に特定して裏付けを取っておく必要のある、多くの重要な情報があります。最も重要なタスクは、以下の 2 つです。

- 証明書のサブジェクト名の識別情報の検証 (証明書を作成する対象の個人またはオブジェクトの識別情報を検証する)。
- 証明書のサブジェクト名の妥当性検証 (サブジェクト名が、証明書を作成する対象の個人またはオブジェクトの識別情報を正しく表していることを検証する)。

これら 2 つのタスクには非常に時間がかかる可能性があり、多くの場合は独立した管理者や管理グループによって行われます。

OpenText Certificate Server ではこれまで常に、OpenText eDirectory のセキュア識別情報管理機能を利用して、これらの検証に必要な時間を節約し、手間を省いてきました。OpenText Identity Console では、管理者がユーザ証明書を一括で作成できるようになっています。つまり、多数のユーザの証明書を一度に作成できます。認証局は、証明書の識別情報が OpenText eDirectory アカウントに関連付けられていることを確認し、それによって証明書のサブジェクトの識別情報を検証します。ただし、証明書のサブジェクト名の妥当性は認証局によって検証されていませんでした。このため、OpenText Certificate Server で証明書を作成するには、証明書を作成する担当者またはソフトウェアには必ず、組織認証局に対する管理権が必要でした。

自己プロビジョニングにより、ユーザまたはサーバは、認証局に対する管理権がなくても、別の管理者または管理グループの介入なく、組織認証局のセキュリティを維持しながら、証明書を生成できます。

OpenText Certificate Server は、証明書の OpenText 識別情報が OpenText eDirectory アカウントに関連付けられていることを確認することによって、証明書のサブジェクトの識別情報を検証します。認証局は OpenText eDirectory 内の情報を確認することで、証明書のサブジェクト名の妥当性も検証します。したがって、組織認証局は OpenText eDirectory のセキュリティ識別情報管理機能を利用して、組織認証局のセキュリティを維持しながら、管理タスクを減らすことができます。

ユーザ自己プロビジョニング

これまで、ユーザ証明書を作成するには認証局に対する管理権だけでなく、ユーザオブジェクトに対する権利も必要でした。ユーザ自己プロビジョニングにより、認証局に対する管理権が不要になります。ただし、userCertificate 属性、NDSPKI:UserCertificateInfo 属性、および SAS:SecretStore 属性に対する読み込み (R) 権と書き込み (W) 権は引き続き必要です。

証明書の作成を要求するユーザが認証局に対する管理権を持っている場合、証明書の作成は、ユーザ自己プロビジョニングが有効にされているかどうかによって影響を受けることはありません。証明書の作成を要求するユーザが認証局に対する管理権を持っていない場合は、要求に含まれるサブジェクト名が、そのユーザの OpenText eDirectory DN、および sasAllowableSubjectNames 属性に含まれるすべての値と比較されます。

サブジェクト名が一致すると、認証局は、すべてのサブジェクト代替名を調べて適切であることを確認します。認証局はそのために、複数のサブジェクト代替名がないことを確認します。代替名が存在する場合、そのタイプは電子メールの名前でなければならず、ユーザオブジェクトに設定済みの電子メールの名前と一致していなければなりません。これらすべての確認に成功すると、認証局に対する管理権を認証局から要求されることなく、証明書を作成できます。

ユーザ自己プロビジョニングを使用するには、次の手順に従います。

1 ユーザ自己プロビジョニングを有効にします。
   1a OpenText Identity Console を起動します。
   1b 組織認証局に対する管理権を持つ管理者として OpenText eDirectory ツリーにログインします。
   1c ホームページで、[**証明書管理**] タイル > [**CA 管理**] の順にクリックし、[**CA 環境設定** ⚙] をクリックします。
   1d [**ユーザ自己プロビジョニングを有効にする**] を選択します。
   1e [**適用**] > [**OK**] の順にクリックします。
2 管理者として、OpenText Identity Console にログインします。
3 [**権利管理**] タイルで、[**トラスティの変更**] > [**オブジェクトの検索** 🔍] の順にクリックします。
4 継承する権利を持つオブジェクトをブラウズして選択し (たとえば、ツリーのルートやコンテナ)、[**OK**] をクリックします。
5 [**トラスティの追加**] をクリックし、オブジェクトを選択して、[**OK**] をクリックします。
6 [**Assigned Rights**] をクリックします。
7 [プロパティの追加] をクリックします。[SELECT PROPERTY] ページが表示されます。
8 [**スキーマ内のすべてのプロパティを表示する**] チェックボックスをオンにします。
9 userCertificate 属性を選択してから、[**OK**] をクリックします。
10 [**読み込み**] 権と [**書き込み**] 権を選択します。
11 [**継承**] チェックボックスをオンにします。

12 NDSPKI:UserCertificateInfo 属性および SAS:SecretStore 属性について、ステップ 5 からステップ 11 を繰り返します。

13 ［完了］>［適用］の順にクリックします。

14 ［OK］をクリックします。

サーバ自己プロビジョニング

これまで、サーバ証明書を作成するには、認証局に対する管理権だけでなく、サーバ証明書が作成されたコンテキストに対する管理権も必要でした。サーバ自己プロビジョニングにより、認証局に対する管理権が不要になります。ただし、サーバ証明書が作成されたコンテキストに対する管理権は引き続き必要です。

サーバ証明書を作成するには、認証局に対する管理権が必要です。証明書の作成は、サーバ自己プロビジョニングが有効にされているかどうかによって影響を受けることはありません。認証局に対する必要な管理権を持っていない場合は、OpenText Identity Console の［証明書管理］タイルから認証局を操作するため、［CA を操作するために読み込み権を要求する］オプションを有効にしてください。次のいずれかに当てはまる場合、認証局に対する管理権は必要ありません。

- 要求に含まれるサブジェクト名が、サーバの OpenText eDirectory DN と、DNS または eDirectory SLP 検索で判断された IP または DNS アドレスと比較される場合。サブジェクト名がこのいずれかと一致した場合、認証局に対する管理権を認証局から要求されることなく、証明書を作成できます。
- サブジェクト名の CN 以外のコンポーネントが、認証局証明書のサブジェクト名の CN 以外のコンポーネントと一致する場合。
- サブジェクト代替名に、リバース DNS 検索で認証局によって再検証される IP アドレス / DNS 名しかない場合。

デフォルトでは、認証局の NDSPKI:Private Key 属性に対する書き込み権はサーバに付与されません。OpenText Identity Console の［認証局の設定］タスクで、［CA を操作するために書き込み権を要求する］オプションが有効にされている場合、認証局の NDSPKI:Private Key 属性に対する読み込み権をサーバに付与する必要があります。

注 : サーバ自己プロビジョニングが有効にされているサーバに対して PKI ヘルスチェックが実行されると、サーバのサーバ証明書が自動的に作成されるか (存在しない場合)、置き換えられる (期限切れの場合) ことに注意してください。詳細については、769 ページの「PKI ヘルスチェック」を参照してください。

サーバ自己プロビジョニングを使用するには、次の手順に従います。

1 OpenText eDirectory 9.0 と、OpenText Identity Console アプリケーションがインストールされていることを確認します。

eDirectory 8.8 と OpenText Identity Console アプリケーションは、どちらも OES 2 に組み込まれており、OES 2 のインストール時に OpenText eDirectory を必要とするコンポーネントのいずれかを選択すると、自動的にインストールされます。

2 サーバ自己プロビジョニングを有効にします。

2a OpenText Identity Console を起動します。

2b 組織認証局に対する管理権を持つ管理者として OpenText eDirectory ツリーにログインします。

2c［**証明書管理**］タイル >［**CA 管理**］の順に移動して、［**CA 環境設定** ］をクリックします。

2d［**ユーザ自己プロビジョニングを有効にする**］を選択します。

2e［**適用**］>［OK］の順にクリックします。

証明書の自己プロビジョニングと証明書の発行タスク

証明書の発行タスクでは、PKCS#10 証明書署名要求 (CSR) を使用して証明書を作成できます。ユーザはこのタスクを使用して、どの OpenText eDirectory オブジェクトにも関連付けられていない証明書を作成できます。証明書の作成を要求するユーザが認証局に対する管理権を持っている場合、証明書の作成には影響がありません。証明書の作成を要求するユーザが認証局に対する管理権を持っていなければ、証明書の要求はユーザ自己プロビジョニングリクエストとして扱われます。ただし、ユーザがオブジェクトの userCertificate 属性、NDSPKI:UserCertificateInfo 属性、および SAS:SecretStore 属性に対する管理権を持っている必要はありません。これは、証明書は OpenText eDirectory に保管されないためです。したがって、オブジェクトに対する権利は必要ありません。

認証局に対する管理権を持たないユーザが証明書を発行するには、ユーザ自己プロビジョニングが有効にされている必要があります。それには、751 ページの 「ユーザ自己プロビジョニング」のステップ 1 から 3 に従います。

証明書の発行タスクについては、725 ページの 「公開鍵証明書の発行」を参照してください。

外部アプリケーションでの OpenText eDirectory 証明書の使用

一部のお客様は、X.509 認定と鍵が必要な、eDirectory 以外のアプリケーション (たとえば、Apache や OpenSSL) を使用しています。そのようなアプリケーションのほとんどは、追加の設定を加えなければ自己署名 (値なし) 証明書を使用するように設定されています。この設定は、アプリケーションを本物の X.509 認定と鍵で構成できるようになるまでの一時的なソリューションとして意図されています。

残念ながら、多くの管理者は、時間がかかり過ぎるという理由や難しすぎるという理由で、自己署名証明書を置き換えていません。さらに、X.509 認定は定期的に期限切れになるよう設計されているため、定期的に証明書を置き換えるのは、継続的な管理タスクになります。

以降のセクションで、この問題に対するソリューションを説明します。

- 754 ページの 「PKI ヘルスチェック機能」
- 755 ページの 「eDirectory 証明書をエクスポートするための SAS:Service オブジェクトの設定」

PKI ヘルスチェック機能

eDirectory 以外のアプリケーションに X.509 認定を提供してほしいというお客様からの要望に応え、現在、OpenText Certificate Server に組み込まれた PKI ヘルスチェックコードで X.509 認定と鍵を自動的にファイルシステムにエクスポートして、OpenText eDirectory が生成した証明書と OpenText eDirectory で管理される証明書を、eDirectory 以外のアプリケーションが利用できるようになっています。

PKI ヘルスチェックが実行されると、証明書の秘密鍵を含め、既存の証明書が自動的に上書きされます。ただし、有効な証明書と秘密鍵が削除されないようにするため、PKI ヘルスチェックは既存の証明書と鍵が OpenText eDirectory で設定されているものと同じであるかどうかを判別します。OpenText eDirectory で設定されているものと異なる場合は、PKI ヘルスチェックはそれらのファイルをバックアップしてから上書きします。したがって、外部ソース (VeriSign* など) から取得された証明書が削除されることはありません。

SAS:Service オブジェクトにサーバの環境設定が作成された後、指定のサーバに関連付けられた鍵と証明書が自動的にファイルシステムにエクスポートされます。OpenText eDirectory 内で鍵と証明書が置き換えられるか更新される場合 (たとえば、サーバ証明書オブジェクトが削除されて、同じ名前で新しいサーバ証明書オブジェクトが作成された場合)、次の PKI ヘルスチェックの実行時に新しい鍵と証明書が自動的にファイルシステムにエクスポートされます。

注 : OpenText Certificate Server に組み込まれた PKI ヘルスコードは、OpenText Certificate Server がロード / 再ロードされるたびに実行されます。OpenText Certificate Server を再ロードするには、次のいずれかの方法を使用できます。

- サーバを再起動する
- OpenText eDirectory を再起動する
- 手動で PKI サーバをアンロードしてからロードする
- OpenText eDirectory 修復 (NDSRepair) を実行する

  OpenText eDirectory 修復の実行中に OpenText Certificate Server がシャットダウンし、修復が完了すると再ロードされます。

PKI ヘルスチェックの詳細については、769 ページの 「PKI ヘルスチェック」を参照してください。

PKI ヘルスチェックで X.509 認定と鍵を自動的にファイルシステムにエクスポートするには、その前に、SAS:Service オブジェクトを設定する必要があります。これは、PKI ヘルスチェックが SAS:Service オブジェクトの設定を読み込むためです。SAS:Service オブジェクトを設定する方法については、755 ページの 「eDirectory 証明書をエクスポートするための SAS:Service オブジェクトの設定」を参照してください。

eDirectory 証明書をエクスポートするための SAS:Service オブジェクトの設定

OpenText eDirectory のサーバ証明書をファイルシステムにエクスポートするには、その前に、SAS:Service オブジェクトにサーバ用の環境設定を作成する必要があります。使用している OpenText eDirectory サーバによって、自動的に作成できる場合も、手動で作成できる場合もあります。以降のセクションで、これらのオプションについて詳しく説明します。

- 755 ページの「eDirectory 証明書の使用を有効にするために SAS:Service オブジェクトを手動で設定する」

eDirectory 証明書の使用を有効にするために SAS:Service オブジェクトを手動で設定する

OES 2 を OpenText eDirectory サーバとして使用していない場合、eDirectory 証明書をエクスポートするには SAS:Service オブジェクトを手動で設定する必要があります。この環境設定で、サーバ証明書の名前を指定します。複数のサーバ証明書をエクスポートする必要がある場合は、複数の環境設定を作成できます。同じ証明書を異なるファイルパスにエクスポートすることも、異なる証明書をそれぞれ異なるファイルパスにエクスポートすることもできます。

注 : ファイルの競合を避けるため、各環境設定で固有のファイルパスを使用する必要があります。公開鍵のパスと秘密鍵のパスは固有にして、互いに異なり、他の環境設定とも異なるようにする必要があります。

SAS:Service オブジェクトに環境設定を作成するには、次の手順に従います。

1. OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**SAS サービスオブジェクト**] の順にクリックします。
2. [**SAS サービスオブジェクト**] をクリックします。
3. 環境設定を作成する SAS:Service オブジェクトを選択します。
4. [**新規作成 (Create New)** ＋] をクリックして、同期サーバ証明書を作成します。[サーバ証明書の同期] ウィンドウが表示されます。
5. [**証明書**] フィールドで、エクスポートする証明書をブラウズ () します。
6. [**公開鍵パス**] フィールドに、アプリケーションが証明書を見つけて使用するためのパスを指定します。例 : C:/novell/nds/servercert.pem
7. [**秘密鍵パス**] フィールドに、アプリケーションが証明書の秘密鍵を見つけて使用するためのパスを指定します。例 : C:/novell/nds/serverkey.pem
8. 使用する鍵のタイプを選択します。OpenSSL を実行している場合は、[PKCS#8] を選択します。Apache を実行している場合は、[PKCS#1] を選択します。
9. **OK** をクリックします。

   環境設定が作成されます。名前、パス、鍵のパス、鍵のタイプが表示されます。
10. [**保存**] をクリックします。

別の環境設定を作成するには、ステップ 4 からステップ 9 を繰り返します。

OES サーバを OpenText eDirectory サーバとして使用している場合は、サーバが SAS:Service オブジェクトに環境設定を作成するように自動的に設定することができます。

注 : OES 2 のインストール時に OpenText eDirectory 証明書の使用が有効にされると (デフォルト)、インストールコードによって SSL CertificateDNS オブジェクトの環境設定が作成され、証明書と鍵が以下のファイルにエクスポートされます。

鍵ファイル : /etc/ssl/servercerts/serverkey.pem

証明書ファイル : /etc/ssl/servercerts/servercert.pem

ルート認証局オブジェクトのタスク

- 756 ページの 「ルート認証局コンテナの作成」
- 756 ページの 「ルート認証局オブジェクトの作成」
- 756 ページの 「ルート認証局オブジェクトのプロパティの表示」
- 757 ページの 「ルート認証局オブジェクトの検証」
- 758 ページの 「ルート認証局証明書の取り消し」

ルート認証局コンテナの作成

このタスクについては、721 ページの 「ルート認証局コンテナの作成」で説明されています。

ルート認証局オブジェクトの作成

このタスクについては、721 ページの 「ルート認証局オブジェクトの作成」で説明されています。

ルート認証局オブジェクトのプロパティの表示

任意の OpenText eDirectory オブジェクトで表示できる OpenText eDirectory の権利とプロパティに加え、ルート認証局オブジェクトに固有のプロパティを表示することもできます。これらのプロパティには、発行者、証明書のステータス、検証期間があります。

1. OpenText Identity Console を起動します。
2. 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

   このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。
3. OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**ルート認証局管理**] の順にクリックします。
4. [**ルート認証局**] タブで、詳細を表示する証明書のニックネームを選択します。
5. [**キャンセル**] をクリックしてページを閉じます。

ルート認証局オブジェクトの検証

証明書に問題があることが疑われる場合、または失効していると思われる場合は、OpenText Identity Console を使用して簡単に証明書を検証できます。外部認証局によって発行された証明書を含め、OpenText eDirectory ツリー内のすべての証明書を検証できます。

証明書の検証プロセスには、証明書のデータだけでなく、証明書チェーン内のデータに対する複数のチェックが含まれます。証明書チェーンは、ルート認証局証明書と、必要に応じて 1 つ以上の中間認証局の証明書からなります。

証明書チェーン内のすべての証明書が有効であれば、結果は有効になります。現在の時刻が証明書の有効期間内であるか、証明書が取り消されていないか、信頼されている認証局によって署名されているかなど、事前定義された一連の条件を証明書が満たすと、その証明書は有効であると見なされます。CRL 配布ポイント拡張機能または OCSP AIA 拡張機能を使用している証明書に限り、取り消されているかどうかがチェックされます。

証明書チェーン内の 1 つ以上の証明書が無効であると判明した場合、または有効であると断定できない場合、結果は無効になります。このような場合は、無効と見なされた証明書とその理由に関する追加情報が提供されます。理由に関する詳細については、[**ヘルプ**] をクリックしてください。

ルート認証局証明書を検証するには、次の手順に従います。

1. OpenText Identity Console を起動します。
2. 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

   このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。
3. OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**ルート認証局管理**] の順にクリックします。
4. [**ルート認証局**] タブで証明書を選択して、[**ルート認証局証明書の検証**] をクリックします。

   [**証明書ステータス**] フィールドに、証明書のステータスが表示されます。証明書が有効でない場合は、理由が示されます。

注 : オブジェクトに保管されている証明書が自己署名証明書ではない場合、証明書チェーンがセキュリティコンテナ内のルート認証局コンテナに入っていなければ (CN=Trusted Roots.CN=Security)、証明書を正常に検証できません。通常、証明書チェーンは 1 つのルートレベルの認証局で構成されるか、中間認証局とルートレベルの認証局で構成されます。ルート認証局コンテナの名前は「ルート認証局」であること、そして証明書チェーンに含まれる各証明書が、それぞれ独自のルート認証局オブジェクトに格納されていることが必要です。ルート認証局コンテナおよびルート認証局オブジェクトの作成方法については、721 ページの 「ルート認証局コンテナの作成」および 721 ページの 「ルート認証局オブジェクトの作成」を参照してください。

ルート認証局証明書の取り消し

鍵または認証局が侵害されている場合、証明書が別の証明書によって置き換えられてしまった場合、証明書が CRL から削除された場合など、さまざまな理由で証明書を取り消す必要があると判断することがあるかもしれません。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、［**証明書管理**］タイル > ［**ルート認証局管理**］の順にクリックします。

4 変更するルート認証局オブジェクトをブラウズしてクリックします。

5 証明書を選択してから、［**取り消し**］をクリックします。

この操作により、［証明書を取り消す］ウィザードが起動します。メッセージに従って証明書を取り消します。

6 ［**OK**］> ［**完了**］の順にクリックします。

証明書取り消しリスト (CRL) のタスク

OpenText Certificate Server には、証明書取り消しリスト (CRL) を管理するためのシステムが用意されています。これはオプションシステムですが、組織認証局によって作成された証明書を取り消せるようにするには、このシステムを実装する必要があります。

CRL は、取り消された証明書とその理由を公開するリストです。

- 759 ページの 「手動による CRL コンテナの作成」
- 759 ページの 「CRL コンテナの削除」
- 760 ページの 「CRL 設定オブジェクトの作成」
- 760 ページの 「CRL 設定オブジェクトのアクティブ化」
- 761 ページの 「CRL 設定オブジェクトのプロパティの表示と変更」
- 763 ページの 「CRL 設定オブジェクトの削除」
- 763 ページの 「CRL オブジェクトの作成」
- 764 ページの 「CRL ファイルのエクスポート」
- 765 ページの 「CRL ファイルの置き換え」
- 765 ページの 「CRL ファイルの有効期間の延長」
- 766 ページの 「CRL オブジェクトのプロパティの表示」
- 766 ページの 「CRL オブジェクトの削除」

手動による CRL コンテナの作成

NetIQ Certificate Server のインストール時に、ユーザが CRL コンテナの作成に必要な権利を持っている場合は、CRL コンテナが作成されます。そうでない場合は、インストールが完了してから、適切な権利を持つユーザが手動で CRL コンテナを作成できます。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**CA 管理**] > [**CRL**] タブの順にクリックします。

CRL コンテナがすでに存在する場合は、組織認証局のプロパティページが表示されます。

CRL コンテナが存在しない場合は、CRL コンテナとそのコンテナに格納する CRL 設定オブジェクトを作成するためのウィザードが起動します。

4 ウィザードに従って操作を完了します。

注 : CRL コンテナがセキュリティコンテナとは別のコンテナに作成されている場合、[CRL] タブのツリー認証局オブジェクト上で ndspkiCRLContainerDN 属性を手動で入力し、CRL を一覧に含める必要があります。

CRL コンテナの削除

CRL コンテナを削除することはできますが、そうすることはお勧めしません。

通常、関連する配布ポイントを含む最後の証明書の有効期限が切れてから 1 日経過するまで、CRL コンテナ、CRL 設定オブジェクト、CRL オブジェクト、CRL ファイルは削除しません。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**CA 管理**] > [**CRL**] タブの順にクリックします。

4 削除する CRL コンテナをブラウズして選択します。

5 [**削除** 🗑] > [**OK**] の順にクリックします。

CRL 設定オブジェクトの作成

CRL 設定オブジェクトは、CRL コンテナ内に作成できます。このオブジェクトは、OpenText eDirectory ツリーで使用可能な CRL オブジェクトの環境設定情報を格納するものです。通常、ツリーには CRL 設定オブジェクトが 1 つだけあります。新しい組織認証局を作成またはロールオーバする場合には、複数の CRL 設定オブジェクトが必要になりますが、新しい証明書を作成するために使用できる CRL 設定オブジェクトは 1 つだけです。

CRL 設定オブジェクトは、CRL コンテナ内に存在します。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 [**証明書管理**] タイル > [**CA 管理**] > [**CRL**] タブの順に移動し、次のいずれかを実行します。

- CRL コンテナが存在しない場合は、[+] をクリックして CRL コンテナを作成します。表示されるページ内のすべての詳細を指定します。
- CRL コンテナは存在するものの、CRL 設定オブジェクトが存在しない場合は、[CRL コンテナ] を選択して、[CRL 環境設定] タブと [CRL 情報] タブで詳細を変更します。

4 [**保存**] をクリックします。

5 [**OK**] をクリックします。

注 : ここで指定する CRL ファイルパスが、OpenText eDirectory インストールパスと対応していることを確認します。

CRL 設定オブジェクトのアクティブ化

OpenText eDirectory ツリー内で複数の CRL 設定オブジェクトを同時にアクティブにすることはできません。複数の CRL 設定オブジェクトがある場合は、どのオブジェクトをアクティブにするか選択する必要があります。デフォルトでは、最初に作成された CRL 設定オブジェクトがアクティブになります。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**CA 管理**] > [**CRL**] タブの順にクリックします。

4 CRL 設定オブジェクトを選択し、[**アクティブにする**] をクリックします。

CRL 設定オブジェクトのプロパティの表示と変更

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**CA 管理**] の順にクリックします。

4 [**CRL**] タブをクリックします。

5 表示または変更する CRL 設定オブジェクトの名前をクリックします。

6 [**OK**] または [**適用**] をクリックします。

注 : 証明書の検証中に、CRL 設定を無効にすることができます。CRL 設定を無効にするには、環境変数 NDSD_DISABLE_CRL_CONFIG を任意の値に設定する必要があります。すでに CRL が設定されている OpenText eDirectory ツリーを使用する場合は、OpenText eDirectory をアップグレードする前に手動で CRL 設定オブジェクト (objectclass: ndspkiCRLConfiguration) および CRL 配布ポイントオブジェクト (objectclass: cRLDistributionPoint) を手動で削除してください。

- 761 ページの 「LDAP マッピング」
- 762 ページの 「HTTP 配布ポイントの場所」

LDAP マッピング

証明書取り消しリストの標準 LDAP タイプでは、CRL のサイズが 64KB に制限されます。この制限を変更するには、OpenText 定義済みタイプの CRL ディレクトリエントリを作成する必要があります。LDAP 配布ポイントを検索できるようにするために、次の手順に従って、標準 LDAP タイプを OpenText LDAP タイプにマップする必要があります。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory にログインします。

3 OpenText Identity Console のホームページで、[**LDAP 環境設定**] タイルをクリックします。

4 [**タイプ**] ドロップダウンで、[**LDAP グループの表示 (View LDAP Groups)**] > [**検索**] の順に選択し、マップする必要がある LDAP グループを選択します。

5 ［属性マップ］ページで、次の変更を行います。

5a プライマリ LDAP 属性 certificateRevocationList;binary (およびセカンダリ属性 certificateRevocationList) から OpenText eDirectory 属性 certificateAuthorityList へのデフォルトマッピングを、OpenText eDirectory 属性 ndspkiCertificateRevocationList に変更します (つまり、OpenText eDirectory 属性を certificateAuthorityList から ndspkiCertificateRevocationList に変更します)。

5b プライマリ LDAP 属性 authorityRevocationList;binary (セカンダリ属性 authorityRevocationList) から OpenText eDirectory 属性 authorityRevocationList へのデフォルトマッピングを、OpenText eDirectory 属性 ndspkiAuthorityRevocationList に変更します (つまり、OpenText eDirectory 属性を authorityRevocationList から ndspkiAuthorityRevocationList に変更します)。

5c プライマリ LDAP 属性 deltaRevocationList;binary (セカンダリ属性 deltaRevocationList) から OpenText eDirectory 属性 deltaRevocationList へのデフォルトマッピングを、OpenText eDirectory 属性 ndspkiDeltaRevocationList に変更します (つまり、OpenText eDirectory 属性を deltaRevocationList から ndspkiDeltaRevocationList に変更します)。

6 ［保存］をクリックします。

7 ［タイプ］ドロップダウンで、［LDAP サーバの表示 (View LDAP Servers)］>［検索］の順に選択します。

8 LDAP 配布ポイントをホストするサーバを選択します。

9 ［情報］ドロップダウンをクリックします。

これにより、LDAP サービスが再起動して、CRL 属性の適切なマッピングを使用するようになります。

LDAP 管理の詳細については、403 ページの第 14 章「LDAP Services for OpenText eDirectory の環境設定」を参照してください。

HTTP 配布ポイントの場所

HTTP 配布ポイントを使用するように Certificate Server を設定する際は、証明書を検証するユーザがアクセスできる場所を指定することが重要です。配布ポイントが含まれる証明書の CRL をユーザが見つけられなければ、証明書は無効であると見なされます。配布ポイントを配置しなければならない場所は、配布ポイントの HTTP アドレスで指定された Web サーバが使用できるディレクトリ内です。そのディレクトリが、認証局をホストしているサーバと同じサーバ上にない場合は、CRL を手動で、またはスクリプトを使用して移動するか、マウント済みのディレクトリ上に作成する必要があります。

CRL 設定オブジェクトの削除

CRL 設定オブジェクトは削除できますが、推奨されていません。CRL 設定オブジェクトを削除すると、サーバは CRL ファイルを作成しなくなります。CRL オブジェクトで指定された場所に CRL ファイルがすでに存在する場合は、有効期限が切れるまで、証明書の検証でそのファイルが使用し続けられます。証明書が期限切れになると、CRL 配布ポイントでその既存の CRL ファイルを参照しているすべての証明書の検証が失敗します。

通常、関連する配布ポイントを含む最後の証明書の有効期限が切れてから 1 日経過するまで、CRL コンテナ、CRL 設定オブジェクト、CRL オブジェクト、CRL ファイルは削除しません。

1. OpenText Identity Console を起動します。
2. 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

   このタスクに必要な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。
3. OpenText Identity Console のホームページで［**CA 管理**］を選択します。
4. 削除する CRL 設定オブジェクトを選択します。
5. **削除** をクリックします。

CRL オブジェクトの作成

このタスクによって、OpenText eDirectory 内にサードパーティ CRL を保管するための CRL オブジェクト (cRLDistributionPoint) を作成できます。このオブジェクトは、OpenText eDirectory ツリー内の任意のコンテナに作成できます。ただし一般的な規則として、OpenText CRL オブジェクトは CRL 設定オブジェクト内にあるので、手動で作成する必要はありません。CRL 設定オブジェクトを作成すると、CRL オブジェクトが自動的に作成されます。

CRL オブジェクトには、CRL の詳細情報が記載された CRL ファイルが含まれています。OpenText CRL オブジェクトの場合、サーバが新しい証明書を発行するたびに、CRL ファイルは自動的に作成および更新されます。その他の CRL オブジェクトの場合、サードパーティの認証局から CRL ファイルをインポートする必要があります。

注 : CRL 配布ポイントという用語は、さまざまな意味で使用されます。CRL 配布ポイントとは、CRL オブジェクトの OpenText eDirectory スキーマオブジェクト名であり、一般用語では、CRL 情報が公開される場所を意味するために使用されます。

CRL オブジェクトを作成するには、次の手順に従います。

1. OpenText Identity Console を起動します。
2. 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

   このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[CA 管理] > [CRL の作成] の順に選択します。
4 オブジェクトの名前を入力し、オブジェクトを配置するコンテキストを指定します。
5 CRL をコピーしてフィールドに貼り付けるか、CRL ファイルから読み込みます。
6 [ネスト] をクリックしてオブジェクトを作成します。

CRL ファイルのエクスポート

CRL 配布ポイントオブジェクトに含まれる CRL は、ファイルにエクスポートすることができます。

OpenText CRL ファイルをエクスポートするには :

1 OpenText Identity Console を起動します。
2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。
3 OpenText Identity Console のホームページで、[CA 管理] > [CRL の作成] の順に選択します。
4 [CRL] タブをクリックします。
5 CRL 設定オブジェクトの名前をクリックします。
6 [CRL のエクスポート] をクリックします。
7 出力フォーマットを選択してから [次へ] をクリックします。
8 エクスポートした CRL をファイルに保存するには、[保存] をクリックし、ファイルの場所を指定します。
9 [OK] をクリックします。

サードパーティの CRL ファイルをエクスポートするには、次の手順に従います。

1 OpenText Identity Console のホームページで、[証明書管理] タイルをクリックします。
2 [CA 管理] > [CRL] タブの順にクリックし、エクスポートする CRL のチェックボックスをオンにします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。
3 [CRL のエクスポート] をクリックします。
4 出力フォーマットを選択してから [次へ] をクリックします。
5 エクスポートした CRL をファイルに保存するには、[保存] をクリックし、ファイルの場所を指定します。
6 [OK] > [OK] をクリックします。

CRL ファイルの置き換え

CRL ファイルを置き換えることはできますが、そうすることはお勧めしません。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[**証明書管理**] > [**CA 管理**] の順に選択します。

4 [**CRL**] タブをクリックします。

5 CRL 設定オブジェクトの名前をクリックし、[**詳細**] をクリックします。

6 [**置換** ] をクリックします。

7 [**OK**] をクリックして続行します。

8 新しい CRL ファイルをブラウズして選択します。

9 **OK** をクリックします。

CRL ファイルが CRL 設定オブジェクトにない場合は、[**インポート**] ボタンが表示されます。

CRL ファイルの有効期間の延長

管理者は、Identity Console を使用して CRL ファイルの有効期間を延長できます。有効期間を延長するには、次の手順を実行します。

1 OpenText Identity Console を起動します。

2 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。

3 OpenText Identity Console のホームページで、[証明書管理] > [**CA 管理**] の順に選択します。

4 [**CRL**] タブをクリックします。

5 CRL ファイルの名前をクリックします。

6 [**次回の CRL 発行**] にある [**有効期間を延長する時間**] を選択し、隣りのボックスで時間数を指定します。1 から 12 時間までの任意の値をこのフィールドに入力できます。

7 [**すぐに発行**] > [**保存**] の順にクリックします。

CRL オブジェクトのプロパティの表示

OpenText CRL オブジェクトのプロパティを表示するには :

1. OpenText Identity Console を起動します。
2. 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

   このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。
3. OpenText Identity Console のホームページで、[**証明書管理**] > [**CA 管理**] の順に選択します。
4. [**CRL**] タブをクリックします。
5. CRL 設定オブジェクトの名前をクリックします。

   これで、CRL オブジェクトのプロパティを確認できます。
6. プロパティを確認し終わったら、[**保存**] または [**キャンセル**] をクリックします。

サードパーティの CRL オブジェクトのプロパティを表示するには、次の手順に従います。

1. OpenText Identity Console を起動します。
2. 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

   このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。
3. OpenText Identity Console のホームページで、[**証明書管理**] > [**CA 管理**] の順に選択します。
4. [**CRL**] タブをクリックします。

   これで、CRL オブジェクトのプロパティを確認できます。
5. プロパティを確認し終わったら、[**OK**] または [**適用**] をクリックします。

CRL オブジェクトの削除

CRL オブジェクトを削除すると、サーバが次に CRL ファイルを生成するときに CRL オブジェクトが再作成されます。Identity Console を使用して作成した CRL オブジェクトを削除しそれをインポートする場合、そのオブジェクトは永続的に失われ、そのオブジェクトを参照するすべての証明書は無効とみなされます。

通常、関連する配布ポイントを含む最後の証明書の有効期限が切れてから 1 日経過するまで、CRL コンテナ、CRL 設定オブジェクト、CRL オブジェクト、CRL ファイルは削除しません。

1. OpenText Identity Console を起動します。
2. 適切な権利を持った管理者として OpenText eDirectory ツリーにログインします。

   このタスクのための適切な権利を確認するには、779 ページの 「タスクを実行するために必要なエントリ権」を参照してください。
3. OpenText Identity Console のホームページで、[証明書管理] > [**CA 管理**] の順に選択します。

4 削除する CRL オブジェクトをブラウズしてクリックします。

5 ［削除］>［OK］の順にクリックします。

OpenText eDirectory のタスク

- 767 ページの 「複数のセキュリティコンテナ、組織認証局、KAP コンテナ、および W0 オブジェクトの解決」
- 768 ページの 「セキュリティコンテナの復元または再作成」
- 768 ページの 「KAP および W0 の復元または再作成」

複数のセキュリティコンテナ、組織認証局、KAP コンテナ、および W0 オブジェクトの解決

OpenText Certificate Server は、OpenText eDirectory ツリー内の複数のサーバにインストールできます。ただし、OpenText Certificate Server が適切に機能するためには、ツリー内にセキュリティコンテナ、組織認証局、KAP コンテナ、および W0 オブジェクトがそれぞれ 1 つだけ存在していなければなりません。

OpenText Certificate Server を OpenText eDirectory ツリー内の複数のサーバにインストールする場合は、OpenText eDirectory に OpenText Certificate Server の各インストール間で複製を許可する必要があります。OpenText eDirectory に複製を許可しない場合、別のサーバへのインストールで、ツリーにすでにセキュリティコンテナ、組織認証局、KAP コンテナ、および W0 オブジェクトが存在することが認識されず、これらのオブジェクトが同じ eDirectory ツリー内の別のサーバで再作成される可能性があります。

以下の項目で、考えられるシナリオとその解決方法について説明します。

- 同じ OpenText eDirectory ツリー内に、複数のセキュリティコンテナがあり、それぞれのコンテナに組織認証局が含まれていて、1 つの KAP コンテナ内に 1 つの W0 オブジェクトがある場合は、証明書を発行しないでください。この問題を解決するには、テクニカルサポート部門にお問い合わせください。
- 同じ OpenText eDirectory ツリー内に、2 つの KAP コンテナが含まれる 1 つのセキュリティコンテナがある場合は、証明書を発行しないでください。この問題を解決するには、テクニカルサポート部門にお問い合わせください。
- 同じ OpenText eDirectory ツリー内に、2 つの組織認証局が含まれる 1 つのセキュリティコンテナがあり、1 つの W0 オブジェクトが含まれる 1 つの KAP コンテナがある場合、両方の組織認証局から発行されたすべてのサーバ証明書とユーザ証明書を削除します。次に、両方の認証局を削除してから新しい組織認証局を作成します。必要に応じて、新しいサーバ証明書とユーザ証明書を発行します。
- 同じ OpenText eDirectory ツリー内に、複数のセキュリティコンテナがあり、それぞれのコンテナに組織認証局が含まれている一方、W0 オブジェクトが含まれる KAP コンテナが 1 つしかない場合は、すべての組織認証局から発行されたすべてのサーバ証明書とユーザ証明書を削除します。さらに、KAP コンテナと W0 オブジェクトが含まれて

いないすべてのセキュリティコンテナを削除します。残りのセキュリティコンテナの名前が *セキュリティ*ではない場合は、名前を *セキュリティ*に変更します。必要に応じて、新しいサーバ証明書とユーザ証明書を発行します。

- 同じ OpenText eDirectory ツリー内に、複数のセキュリティコンテナがあり、そのうち 1 つのコンテナだけに、組織認証局と KAP コンテナおよびその中の W0 オブジェクトが含まれている場合は、KAP コンテナと W0 オブジェクトが含まれていないすべてのセキュリティコンテナを削除します。残りのセキュリティコンテナの名前が *セキュリティ*ではない場合は、名前を *セキュリティ*に変更します。

セキュリティコンテナの復元または再作成

セキュリティコンテナを削除した場合、セキュリティコンテナを復元または再作成するまでは、組織認証局を作成することができません。

セキュリティコンテナを復元するには、セキュリティコンテナが含まれる OpenText eDirectory のパーティションを復元します。

セキュリティコンテナを再作成するには、次の 2 つの方法のいずれかを使用します。

- OpenText Identity Console を使用して、[**ツリービュー**] タイル > [**追加**] の順にクリックし、[**ツリーのセキュリティコンテナ (Tree's Security Container)**] をクリックして [**名前**] をしてから、[**次へ**] をクリックします。コンテナ名は「セキュリティ」でなければなりません。
- OpenText eDirectory ツリー内の任意のサーバに、OpenText Certificate Server を再インストールします。

KAP および W0 の復元または再作成

KAP または W0 オブジェクトを削除しないでください。削除すると、それまでに作成したすべてのユーザ証明書が無効になります。これらのオブジェクトのいずれかを削除した場合、この問題を解決する方法については、TID 番号 3032354 の「How to Restore or Recreate KAP and W0 Objects」を参照してください。問題が修正されるまでは、OpenText Certificate Server、Single Sign-on、NMAS、または OpenText eDirectory をインストールしようとしないでください。

アプリケーションのタスク

このセクションでは、OpenText 証明書を使用するように暗号化対応アプリケーションを設定する方法について説明します。

このセクションに記載する情報の一部は最新のものではありませんが、参考になります。暗号化対応アプリケーションで証明書を使用する方法の最新情報については、アプリケーションのマニュアルを参照してください。

セキュリティで保護された電子メールをアプリケーションで使用できるようにするための一般的なプロセスは、次のとおりです。

1. 組織認証局の自己署名証明書 (727 ページの 「組織認証局の自己署名証明書のエクスポート」を参照)、ユーザ証明書、および証明書と一致する秘密鍵を .pfx ファイルにエクスポートします (747 ページの 「ユーザ証明書と秘密鍵のエクスポート」を参照)。
2. .pfx ファイルを電子メールクライアントにインポートします。
3. 電子メールを保護するように電子メールクライアントを設定します。

ブラウザを使用して、インターネットでサーバに対する SSL 接続を作成するためには、ユーザ証明書またはサーバ証明書に署名した認証局を信頼する必要があります。認証局を信頼していなければ、アプリケーションがエラーを表示する可能性があります。一部のアプリケーションでは、警告メッセージで、そのアプリケーションにまだ既知ではない認証局が発行したユーザ証明書またはサーバ証明書の受諾または拒否を選択できるようになっています。企業の組織認証局によって署名されたサーバ証明書またはユーザ証明書は、常にこのような警告およびエラーを生成します。これは、組織認証局が、アプリケーションで信頼する認証局としてリストされていないためです。組織認証局の自己署名証明書をアプリケーションにインストールすることで、このような警告やエラーを防ぐことができます。組織認証局をブラウザにインストールすると、組織認証局が自動的に信頼する認証局として追加されます。

アプリケーションで組織認証局を信頼する認証局として受け入れるようにするには、次の手順に従います。

1. 組織認証局の自己署名証明書をエクスポートします (727 ページの 「組織認証局の自己署名証明書のエクスポート」を参照)。

   注 : インターネットブラウザは、.der または .crt フォーマットの証明書を認識します。

2. ブラウザのマニュアルで説明されている指示に従って、ブラウザに証明書をインポートします。

PKI ヘルスチェック

OpenText Certificate Server には、Certificate Server コンポーネントのヘルスと整合性を維持するプロセスが組み込まれています。このプロセスは、PKI ヘルスチェックと呼ばれ、次のタイミングで実行されます。

- サーバが再起動するとき。
- OpenText eDirectory が起動するとき。
- DSRepair の実行が終了するとき。

PKI ヘルスチェックによって、次のタスクが実行されます。

表 25-1 PKI ヘルスチェックのタスク

タスク	機能
SAS サービスオブジェクトへのサーバのリンクの検証	このタスクでは、サーバオブジェクトから SAS:Service オブジェクトへのリンクの有無を確認します。リンクが存在する場合、オブジェクトに適切な名前が付けられていて、オブジェクトがサーバと同じコンテキスト内にあることを確認します。リンクが存在しない場合、適切に名前が付けられたオブジェクトがサーバと同じコンテキスト内にあるかどうかを確認します。そのようなオブジェクトが存在する場合、サーバからオブジェクトへのリンクを作成します。
SAS サービスオブジェクトの検証	このタスクでは、SAS:Service オブジェクトの有無を確認します。存在しない場合は新しいオブジェクトを作成し、サーバオブジェクトから新規オブジェクトへのリンクを作成します。その後、SAS:Service オブジェクトに OpenText eDirectory に対する必要な権利があるかどうかを確認します。必要な権利がなければ、SAS:Service オブジェクトに必要な権利を与えようと試みます。
KMO へのリンクの検証	このタスクでは、SAS:Service オブジェクトにリンクされているサーバ証明書オブジェクト (KMO) のリストを読み込みます。KMO のすべてに適切な名前が付けられているかどうかを確認し、適切でない名前を修正します。このタスクでは、KMO のすべてがサーバオブジェクトと同じコンテキスト内にあるかどうかも確認し、必要に応じて正しいコンテキストに移動しようと試みます。
サーバ証明書 (KMO) の確認	このタスクでは、サーバオブジェクトと同じコンテナ内にあるすべての KMO の名前を読み込んでリストに取り込みます。続いて、リストに含まれる KMO ごとに次の処理を行います。 ◆ NDSPKI:Not Before 属性と NDSPKI:Not After 属性に、証明書の有効期限日を取り込もうとします。 ◆ パブリックにホストサーバ属性に対する読み込み権があるかどうかを確認します。 ◆ KMO からサーバへのバックリンクを確認します。バックリンクが異なるサーバを対象としている場合、KMO を無視してリストから削除します。 ◆ 秘密鍵を読み取ってラップ解除を試みます。

タスク	機能
KMO へのリンクの再検証	このタスクでは、SAS:Service オブジェクトにリンクされているサーバ証明書オブジェクト (KMO) のリストを読み込みます。このリスト内の各 KMO を、サーバ証明書 (KMO) の確認で作成したリストと照合します。サーバ証明書 (KMO) の確認を使用して、リンクされている証明書に問題がないかどうかを確認し、KMO が使用できない場合は証明書のリンクを解除します。このタスクでは、このサーバで使用できるリンク解除済み KMO があるかどうかを確認し、存在する場合はその KMO をリンクします。
デフォルト証明書の作成	このタスクでは、サーバ自己プロビジョニングが組織認証局オブジェクトで有効にされているかどうかを確認します。サーバ自己プロビジョニングが有効にされていない場合、このステップはスキップされます。サーバ自己プロビジョニングが有効にされている場合、NPKICreateDefaultCertificates() API を呼び出します。この API は、次の条件に当てはまる場合は SSL CertificateDNS 証明書を作成または置換します。 ◆ 証明書が存在しない場合。 ◆ 証明書に有効期限がないか、証明書が間もなく期限切れになる場合。 ◆ 証明書のサブジェクト名が、サーバに設定されているデフォルトの IP アドレスおよび DNS アドレスと一致しない場合。 **注 :** OpenText eDirectory では、SSL CertificateIP は自動的に作成されません。SSL 証明書 DNS には、[サブジェクトの代替名] にリストされているすべての IP が含まれます。 さらに、この API は、サーバに設定されている IP アドレスおよび DNS アドレスをすべて取得し、次の条件に当てはまる場合は各アドレスの証明書を作成または置換します (IP AG *ip address*、IP DNS *dns nam* など)。 ◆ 証明書が存在しない場合。 ◆ 証明書の有効期限が切れているか、証明書が間もなく期限切れになる場合。

タスク	機能
外部サービスの証明書の同期化	このタスクでは、SAS:Service オブジェクトから環境設定を読み込みます。設定済みの各エントリに対して、指定の KMO オブジェクトから証明書と秘密鍵を取得します。指定のディレクトリが存在しない場合、そのディレクトリの作成を試みます。次に、秘密鍵をラップ解除して、暗号化されていないフォーマットに変換します。既存の秘密鍵と証明書ファイルを、指定の KMO のものと比較します。鍵と証明書が同じではない場合、既存の秘密鍵と証明書ファイルのバックアップを作成し、既存の秘密鍵と証明書で上書きします。鍵は PEM フォーマットで書き出されます。
ファイルシステムへの eDirectory 認証局証明書のエクスポート	このタスクを実行する方法は、実行しているオペレーティングシステムによって異なります。 SSCert.der および SSCert.pem ファイルには、組織認証局の RSA 証明書が含まれます。組織認証局に ECDSA 証明書がある場合、OpenText eDirectory は ECDSA 証明書を SSECCert.der および SSECCert.pem ファイルにエクスポートして、それらのファイルを SSCert.der および SSCert.pem ファイルと同じディレクトリに保管します。 ◆ **Windows:** PKI 作業ディレクトリ内の SSCert.der および SSCert.pem ファイルに、OpenText eDirectory 内の組織認証局の証明書と同じ証明書が含まれているかどうか確認します。同じでない場合は、ファイルの置換を試みます。 デフォルトの PKI 作業ディレクトリ : c:\Novell\NDS\DIBFiles\CertServ\ ◆ **Linux (OES Linux 以外) の場合 :** OpenText eDirectory データディレクトリ内の SSCert.der および SSCert.pem ファイルに、OpenText eDirectory 内の組織認証局の証明書と同じ証明書が含まれているかどうか確認します。同じでない場合は、ファイルの置換を試みます。 デフォルトの OpenText eDirectory データディレクトリ : /var/opt/novell/eDirectory/data ◆ **OES Linux:** /etc/opt/novell/certs/SSCert.der および /etc/opt/novell/certs/SSCert.pem ファイルに、OpenText eDirectory 内の組織認証局の証明書と同じ証明書が含まれているかどうか確認します。証明書が同じではない場合、ファイルを置き換えるために、組織認証局の証明書を /etc/ssl/certs ディレクトリに追加してから、c_rehash プログラムを実行します。ただし、ファイルを置き換える前に、既存のすべての証明書のバックアップを作成します。

公開鍵暗号化の基本

このセクションでは、公開鍵暗号化の基本について説明します。

- 773 ページの 「概要」
- 773 ページの 「セキュアなデータ転送」
- 773 ページの 「鍵ペア」
- 776 ページの 「信頼関係の確立」

概要

Web ページのブラウズや公開チャットフォーラムなど、ほとんどのインターネット通信のコンテンツは、必要なデバイスを持っていれば誰でもモニタすることができます。オンラインショッピングでのクレジットカード情報のやり取りといったデータ転送のコンテンツは、機密にしておかなければなりません。

公開鍵暗号化は、インターネット上でのデータ転送を機密にして保護するために広く使用されている手法です。具体的には、公開鍵暗号化とは、メッセージの送信者を認証するため、そしてメッセージのコンテンツを暗号化するために、「鍵」と呼ばれるデジタルコードを使用するシステムのことを指します。

セキュアなデータ転送

データ転送は、次の 2 つによって機密かつセキュアになります。

- **認証** : データ受信者が、データ送信者の身元、またはデータ送信者が主張する本人そのものであることを把握します。
- **暗号化** : 送信されるデータは、対象とする受信者だけが読み取れるように暗号化されます。

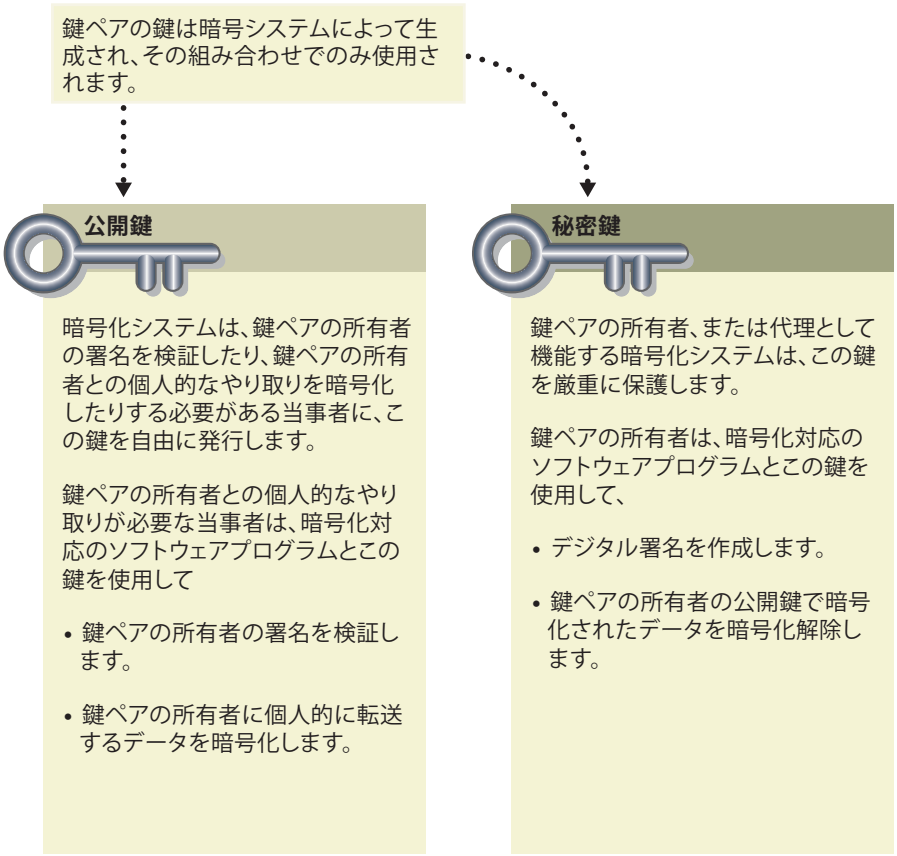
鍵ペア

認証と暗号化はどちらも、関連するデジタルコード、つまり「鍵」のペアを使用して行われます。各ペアの一方の鍵は一般に配布され、もう一方の鍵は厳重に機密保持されます。

データ送信者のそれぞれには、それが個人、ソフトウェアプログラム、また銀行や企業などの機関であるかに関わらず、公開鍵暗号化システムによって鍵ペアが発行されます。

鍵ペアを構成する各鍵の基本的な原則と役割について、以下の図に要約します。

図 25-1 鍵ペアの基本的な説明



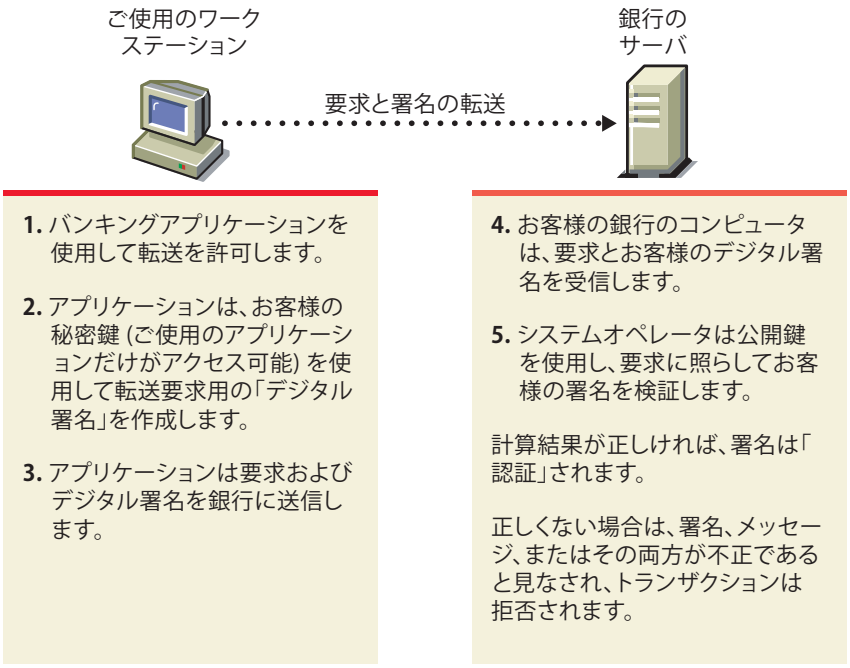
- 774 ページの 「鍵ペアと認証」
- 775 ページの 「鍵ペアと暗号化」

鍵ペアと認証

「認証」とは、データ送信者が誰であるか、またはデータ送信者が主張するデータ送信元を、データ受信者が正確に把握することを意味します。

たとえば、銀行に、自分の口座から別の口座への送金を許可するとします。その場合、銀行は、そのメッセージを送信したのが口座名義人本人であること、そしてメッセージが送信中に改ざんされていないことを証明しなければなりません。以下の図に、公開鍵暗号化を使用したオンライントランザクションのプロセスを示します。

図 25-2　公開鍵によるプロセス



デジタル署名とその検証については、777 ページの 「デジタル署名」を参照してください。

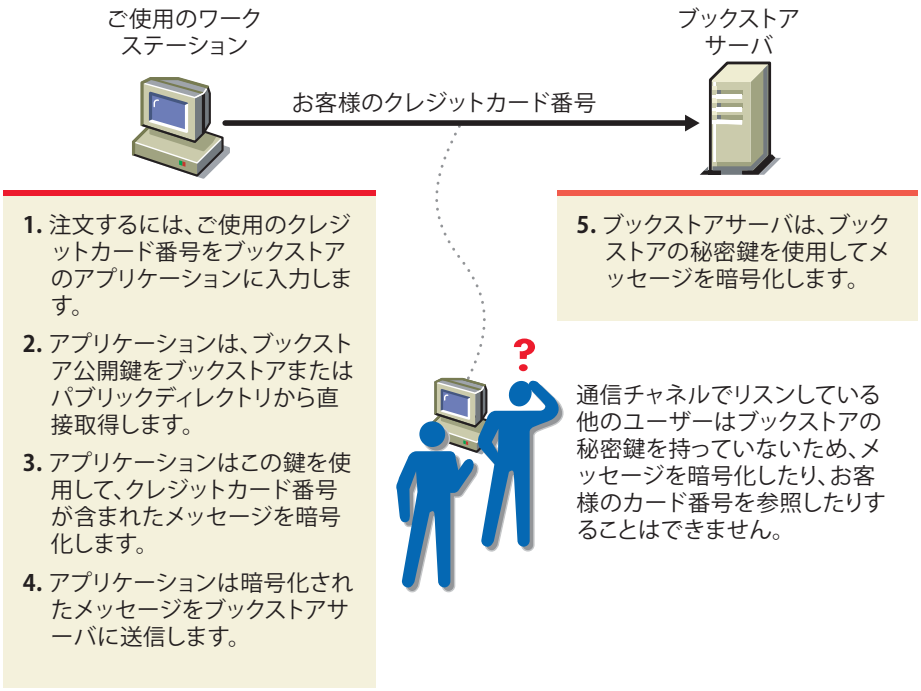
鍵ペアと暗号化

*暗号化*とは、対象とする受信者だけがデータを読み取れることを意味します。

たとえば、インターネットのベンダに本を注文して、その代金の支払にクレジットカードを使用しなければならないとします。対象とする受信者以外には、誰にもクレジットカード番号が読み取られないようにしなければなりません。

以下の図に示す暗号化プロセスが、クレジットカード番号を安全に送信するメカニズムを提供します。

図 25-3 暗号化プロセス



信頼関係の確立

送信者と受信者が互いを知っていて信頼している場合、公開鍵を交換するだけで、認証と暗号化を含めたセキュアなデータ転送を確立できます。この場合、送信者と受信者のそれぞれが、相手の公開鍵と自分の公開鍵を使用します。

しかし通常は、セキュアなデータ転送を必要とする当事者には、互いの識別情報を信頼する基盤がありません。送信者と受信者が共通して信頼するサードパーティに、それぞれの識別情報を証明してもらう必要があります。

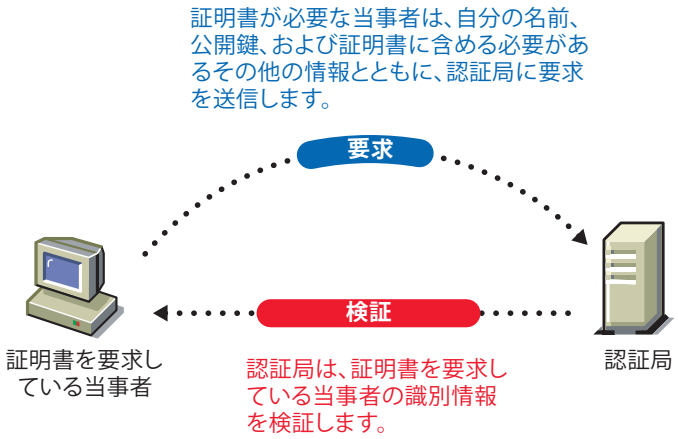
- 776 ページの 「認証局」
- 777 ページの 「デジタル署名」
- 778 ページの 「証明書チェーン」
- 779 ページの 「ルート認証局」

認証局

公開鍵暗号化環境で識別情報を証明する必要がある当事者は、認証局と呼ばれる、信頼できるサードパーティのサービスを利用します。

認証局の主な目的は、当事者の身元または当事者が主張している本人であることを検証してから、その当事者が使用するための公開鍵証明書を発行することです。公開鍵証明書は、その証明書に含まれる公開鍵が、証明書に指名されている当事者に属することを検証するためのものです。

図 25-4 証明書要求



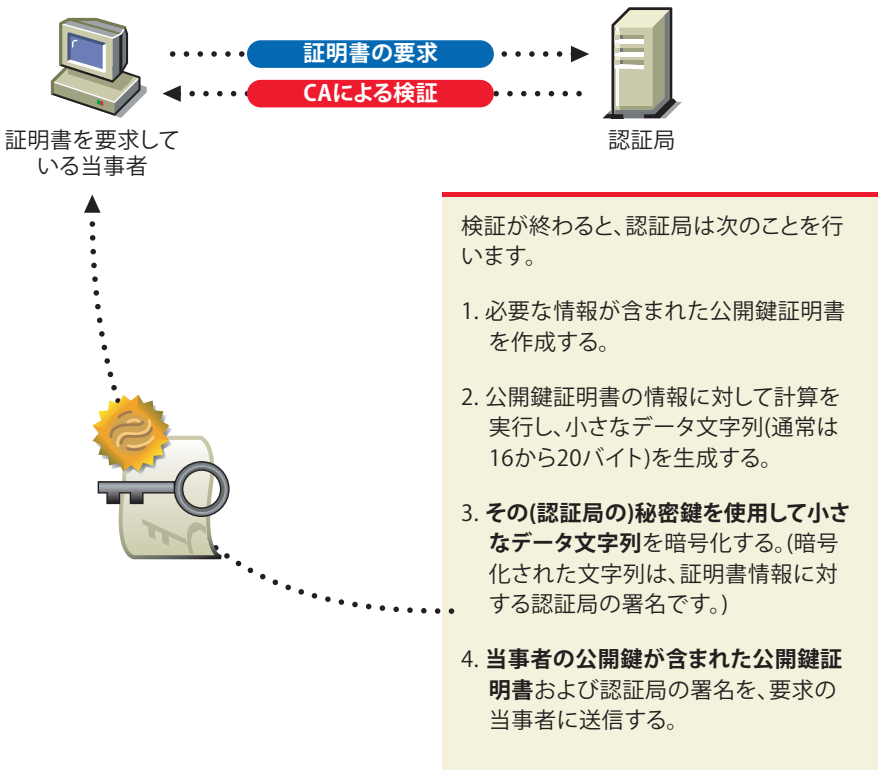
認証局は、証明書を要求している当事者の識別情報が正しいものであると認めると、電子「証明書」を発行し、その証明書にデジタル署名を付けます。

デジタル署名

紙の文書上の個人の署名がその文書の信ぴょう性を表すのと同じく、デジタル署名は電子データの信ぴょう性を表します。

デジタル署名を作成するには、署名の作成に使用されるソフトウェアで、署名を付けたデータを署名者の秘密鍵にリンクします。以下の図に、認証局が公開鍵証明書のデジタル署名を作成するために従うプロセスを示します。

図 25-5　デジタル署名



デジタル署名は、署名者とデータを固有にリンクします。署名者以外は、署名者の秘密鍵を持っていないため、署名を複製することはできません。さらに、署名者がデータに署名したことを否認することもできません。これは、「否認防止」と呼ばれます。

認証局が公開鍵証明書に署名を付けるということは、認証局がその公開している確立済みのポリシーに従って、公開鍵の所有者の識別情報を確認したことを保証するということです。

署名付きのデータ (公開鍵証明書など) を受信すると、署名作成ソフトウェアは当初使用した計算と同じ計算をそのデータに適用して、データの信ぴょう性を検証します。データが改ざんされていなければ、両方の計算で同じ結果が出ます。これにより、データも署名も送信中に改ざんされていないと確実に見なすことができます。

証明書チェーン

証明書チェーンとは、順序付けられた証明書のリストのことです。証明書は、サーバ証明書またはユーザ証明書の後に、その証明書を発行した認証局の証明書が続くように順序付けられます。

認証局の証明書には、認証局自体が署名を付ける場合も (つまり、自己署名証明書)、別の認証局が署名を付ける場合もあります。自己署名証明書を使用する認証局は、一般にルート認証局と呼ばれます。自己署名証明書を使用しない認証局は、一般に従属認証局または中間認証局と呼ばれます。

ユーザ証明書またはサーバ証明書が、自己署名証明書を使用する認証局によって署名付けされている場合、証明書チェーンは、エンドエンティティの証明書とルート認証局証明書だけで構成されます。

ユーザ証明書またはサーバ証明書が中間認証局によって署名付けされている場合は、証明書チェーンが長くなります。最初の 2 つの要素は同じくエンドエンティティの証明書と、その後に続く中間認証局の証明書です。ただし、中間認証局の証明書の後には、その上位認証局の証明書が続きます。このリストが、最後の証明書がルート認証局証明書になるまで続きます。したがって、証明書チェーンは無限に長くなる可能性があります。しかし実際には、ほとんどの証明書には 2 つか 3 つの証明書しかありません。

ルート認証局

ユーザ証明書またはサーバ証明書チェーンを構成する証明書のうち、少なくとも 1 つの証明書を信頼しなければ、デジタル署名を検証することはできません。ユーザまたはサーバの証明書自体を信頼することも、チェーンを構成する任意の証明書を信頼することもできます。通常、信頼する証明書は、ルート認証局証明書です。

証明書を使用できるアプリケーションソフトウェアの大半には、信頼する証明書のリストがすでにインストールされています。これらの証明書はルート認証局のものなので、「信頼されたルート」と呼ばれます。ルート認証局は一般に、営利目的の認証局です。必要に応じて、信頼する証明書のリストに他の認証局を追加したり、リストから認証局を削除したりできます。

タスクを実行するために必要なエントリ権

以下のリストに、管理者が OpenText eDirectory ツリー内で OpenText Certificate Server タスクを管理するために必要となる、具体的なエントリ権を記載します。これらの権利は、必要最小限のエントリ権です。

このリストは、会社の証明書認証局および証明書管理の一部またはすべてを管理させるために、管理者が別のユーザに権利を付与する場合にも役立ちます。

***表 25-2** 管理者のエントリ権*

タスク	必要なエントリ権
OpenText Certificate Server のインストール	eDirectory ツリーへの初期インストール : ◆ ツリーのルートに対するスーパバイザ 後続のインストール : ◆ W0 オブジェクトに対するスーパバイザ ◆ サーバ証明書オブジェクトを作成するために必要な権利 ユーザがサーバ証明書オブジェクトを作成するために必要な権利を持っていない場合、インストールは終了しますが、必要な権利を持つユーザが手動でサーバ証明書オブジェクトを作成し、これらのサーバ証明書を使用するアプリケーションを手動で設定する必要があります。

タスク	必要なエントリ権
組織認証局の作成	◆ セキュリティコンテナに対するスーパバイザ
組織認証局のプロパティと証明書の表示	◆ 組織認証局のオブジェクトに対するブラウズ
組織認証局の証明書のエクスポート	◆ 組織認証局のオブジェクトに対するブラウズ
公開鍵証明書の発行	◆ 組織認証局のオブジェクトの NDSPKI:Private Key 属性に対する読み込み ただし、公開鍵証明書を発行しようとしているオブジェクトが NCP サーバの場合は、次の権利が必要です。 ◆ 組織認証局のオブジェクトの NDSPKI:Private Key 属性に対する書き込み
組織認証局のバックアップと復元	◆ 組織認証局のオブジェクトに対するスーパバイザ
組織認証局の別のサーバへの移動	◆ 組織認証局のオブジェクトに対するスーパバイザ
組織認証局の証明書の検証	◆ 組織認証局のオブジェクトに対するブラウズ
組織認証局の置き換え	◆ 組織認証局のオブジェクトに対するスーパバイザ
組織認証局の削除	◆ 組織認証局のオブジェクトに対する削除
サーバ証明書オブジェクトの作成	◆ サーバのコンテナに対するスーパバイザ ◆ 組織認証局のオブジェクトの NDSPKI:Private Key 属性に対する読み込み (組織認証局を使用する場合のみ) ただし、公開鍵証明書を発行しようとしているオブジェクトが NCP サーバの場合は、次の権利が必要です。 ◆ サーバのコンテナに対するスーパバイザ ◆ 組織認証局のオブジェクトの NDSPKI:Private Key 属性に対する書き込み

タスク	必要なエントリ権
サーバ証明書オブジェクトへの公開鍵証明書のインポート	◆ サーバ証明書オブジェクトの NDSPKI:Public Key Certificate 属性に対する書き込み ◆ サーバ証明書オブジェクトの NDSPKI:Certificate Chain 属性に対する書き込み
サーバ証明書オブジェクトの削除	◆ サーバ証明書オブジェクトに対する削除
サーバ証明書オブジェクトからのルート認証局または公開鍵証明書のエクスポート	◆ サーバ証明書オブジェクトに対するブラウズ
サーバ証明書オブジェクトのプロパティと証明書の表示	◆ サーバ証明書オブジェクトに対するブラウズ
サーバ証明書オブジェクトのバックアップと復元	◆ バックアップするサーバ証明書オブジェクトを所有するサーバオブジェクトに対するスーパバイザ ◆ 復元するサーバオブジェクトのコンテナに対する作成
サーバ証明書の検証	◆ サーバ証明書オブジェクトに対するブラウズ
サーバ証明書の取り消し	◆ 認証局秘密鍵に対する読み込み、サーバ証明書オブジェクトに対する削除、またはホストサーバ (つまり、NCP™ サーバオブジェクト) に対するスーパバイザ
サーバ証明書の鍵生成要素の置換	◆ サーバ証明書オブジェクトの NDSPKI:PrivateKey 属性に対する書き込み

タスク	必要なエントリ権
ユーザ証明書の作成	◆ 組織認証局のオブジェクトの NDSPKI:Private Key 属性に対する読み込み ◆ ユーザオブジェクトの NDSPKI:userCertificateInfo 属性に対する読み込みと書き込み ◆ ユーザオブジェクトの SAS:SecretStore 属性に対する読み込みと書き込み ◆ ユーザオブジェクトの userCertificate 属性に対する読み込みと書き込み ただし、公開鍵証明書を発行しようとしているオブジェクトが NCP サーバの場合は、次の権利が必要です。 ◆ 組織認証局のオブジェクトの NDSPKI:Private Key 属性に対する書き込み ◆ ユーザオブジェクトの NDSPKI:userCertificateInfo 属性に対する読み込みと書き込み ◆ ユーザオブジェクトの SAS:SecretStore 属性に対する読み込みと書き込み ◆ ユーザオブジェクトの userCertificate 属性に対する読み込みと書き込み
ユーザオブジェクトへの公開鍵証明書のインポート	◆ ユーザオブジェクトの NDSPKI:userCertificateInfo 属性に対する読み込みと書き込み ◆ ユーザオブジェクトの NDSPKI:userCertificate 属性に対する読み込みと書き込み
ユーザ証明書のプロパティの表示	◆ ユーザオブジェクトに対するブラウズ
ユーザ証明書のエクスポート	◆ ユーザオブジェクトに対するブラウズ
ユーザの秘密鍵と証明書のエクスポート	◆ ユーザとしてログインする必要がある
ユーザ証明書と秘密鍵の削除	◆ NDSPKI:userCertificateInfo 属性に対する読み込みと書き込み ◆ userCertificate 属性に対する読み込みと書き込み
ユーザ証明書の検証	◆ ユーザオブジェクトに対するブラウズ

タスク	必要なエントリ権
ユーザ証明書の取り消し	◆ 認証局証明書の秘密鍵に対する読み込み、ユーザオブジェクトに対する削除、またはそのユーザとしてログインしている場合は userCertificate 属性に対する書き込み
ルート認証局コンテナの作成	◆ セキュリティコンテナに対する作成
ルート認証局オブジェクトの作成	◆ ルート認証局オブジェクトを配置するルート認証局コンテナに対する作成
ルート認証局オブジェクトのプロパティの表示	◆ ルート認証局オブジェクトに対するブラウズ
ルート認証局証明書の置き換え	◆ ルート認証局オブジェクトの NDSPKI:Not After 属性に対する読み込みと書き込み ◆ ルート認証局オブジェクトの NDSPKI:Not Before 属性に対する読み込みと書き込み ◆ ルート認証局オブジェクトの NDSPKI:Subject Name 属性に対する読み込みと書き込み ◆ ルート認証局オブジェクトの NDSPKI: ルート認証局の証明書属性に対する読み込みと書き込み
ルート認証局証明書の検証	◆ ルート認証局オブジェクトに対するブラウズ
ルート認証局証明書の取り消し	◆ 認証局秘密鍵に対する読み込み、またはルート認証局オブジェクトに対する削除
ルート認証局オブジェクトの削除	◆ ルート認証局オブジェクトに対する削除
CRL コンテナの作成	◆ セキュリティコンテナに対するスーパバイザ ◆ 組織認証局のオブジェクトの ndspkiCRLContainerDN 属性に対する書き込み
CRL コンテナの削除	◆ CRL コンテナに対する削除
CRL 設定オブジェクトの作成	◆ CRL コンテナに対するスーパバイザ
CRL 設定オブジェクトのアクティブ化	◆ 組織認証局のオブジェクトの ndspkiCRLConfigurationDNList 属性に対する書き込み

タスク	必要なエントリ権
CRL 設定オブジェクトのプロパティの表示と変更	変更 : ◆ CRL 設定オブジェクトに対するスーパバイザ または ◆ CRL 設定オブジェクトで変更する属性に対する書き込み 表示 : ◆ CRL 設定オブジェクトに対するブラウズ
CRL 設定オブジェクトの削除	◆ CRL 設定オブジェクトに対する削除
CRL オブジェクトの作成	◆ CRL 設定オブジェクトに対するスーパバイザ
CRL ファイルのエクスポート	◆ certificateRevocationList 属性に対する読み込み
CRL ファイルの置き換え	◆ CRL オブジェクトに対するブラウズ
CRL オブジェクトのプロパティの表示	◆ certificateRevocationList 属性に対するブラウズ
CRL オブジェクトの削除	◆ CRL 配布ポイントに対する削除
セキュリティコンテナの作成	◆ OpenText eDirectory ツリーのルートに対する作成
SAS サービスオブジェクトの作成	◆ オブジェクトのコンテナに対するスーパバイザ ◆ オブジェクトを作成するサーバの SAS:Service DN 属性に対する書き込み

26 パスワードを管理する

このセクションでは、ユニバーサルパスワード、パスワードポリシー、およびパスワードセルフサービスの概要を説明します。

- 785 ページの 「ユニバーサルパスワードについて」
- 788 ページの 「非可逆パスワードストレージを理解する」
- 790 ページの 「パスワードポリシー」
- 790 ページの 「ユニバーサルパスワードの導入」
- 796 ページの 「パスワードポリシーを使用したパスワードの管理」
- 826 ページの 「セキュリティ上の考慮事項」
- 828 ページの 「eDirectory へのハッシュベースのパスワードのインポート」

ユニバーサルパスワードについて

ユニバーサルパスワードは、NMAS (NetIQ モジュラー認証サービス) モジュールのコンポーネントである、セキュアパスワードマネージャで管理します。セキュアパスワードマネージャは、さまざまな OpenText 製品と OpenText パートナーの製品で、パスワードベースの認証方式の管理を簡素化します。管理ツールは 1 つのパスワードだけを公開し、下位互換性に対応するための舞台裏の処理をすべて公開することはしません。

セキュアパスワードマネージャと、ユニバーサルパスワードを管理または使用するその他のコンポーネントは、OpenText eDirectory の一部としてインストールされます。ただし、ユニバーサルパスワードはデフォルトで有効になっていません。認証とパスワード設定用のすべての API がユニバーサルパスワードをサポートするように移行されるので、既存のすべての管理ツールは、これらの新しいライブラリを備えたクライアント上で実行すると、自動的にユニバーサルパスワードと連動します。

注 : OpenText Identity Console はユニバーサルパスワードをサポートしています。また、ネットワーク内の既存システムのために、引き続き NDS パスワードもサポートしています。ユーザに対してユニバーサルパスワードを設定して有効にすると、OpenText Identity Console の機能によって、自動的に NDS パスワードがユニバーサルパスワードにアップグレード / マイグレートされます。

ユニバーサルパスワードの安全性

他のパスワードシステムとの相互運用を容易にするには、復号可能なユニバーサルパスワード暗号化が必要です。管理者は、そのシステムのコストと利点を評価する必要があります。OpenText eDirectory に保管されたユニバーサルパスワードを使用すると、複数の異なるパスワードを管理しようとする場合よりも、セキュリティが強化されたり、より便利

になったりする可能性があります。OpenText では、OpenText eDirectory に保管されたユニバーサルパスワードが確実に保護されるよう、複数のレベルでセキュリティを提供しています。

ユニバーサルパスワードは、次の 3 つのレベルのセキュリティで保護されます。

- パスワード自体の暗号化
- OpenText eDirectory の権利
- ファイルシステムの権利

ユニバーサルパスワードは、ユーザ固有の鍵で暗号化されます。ユニバーサルパスワードとユーザ鍵は、どちらも OpenText eDirectory のみが読み込めるシステム属性に格納されます。ユーザの鍵は、ツリー鍵を使用して暗号化されて保管されます。ツリー鍵は、各マシンで固有の NICI 鍵によって保護されます。ツリー鍵も NICI 鍵も OpenText eDirectory 内には保管されないことに注意してください。つまり、これらの鍵と、これらの鍵で保護するデータが一緒に保管されることはありません。

ツリー鍵は各マシン上のツリー内にありますが、ツリーごとにツリー鍵は異なります。したがって、あるツリー鍵で暗号化されたデータは、同じツリー内のマシン上でしか元に戻すことはできません。このように、ユニバーサルパスワードは保管されている間、3 重の暗号化によって保護されます。

それぞれの鍵は、OpenText eDirectory の権利によっても保護されます。ユニバーサルパスワードを変更する権利を持つのは、スーパバイザ権を持つ管理者、またはその鍵を所有するユーザだけです。

ファイルシステムの権利は、適切な権利を持つユーザだけに鍵へのアクセスを許可するために使用されます。

デフォルトでは、ユーザ固有の鍵とツリー鍵は 3 DES 鍵です。OpenText eDirectory 9.2 では、AES 256 ビットキーをサポートしています。AES 256 ビットのツリーキーを作成する方法については、『NICI Administration Guide (NICI 管理ガイド)』の「Creating an AES 256-Bit SDI Key」を参照してください。管理者は、diagpwd ユーティリティを使用してパスワードを再暗号化できます。詳細については、823 ページの 「ユニバーサルパスワードの診断ユーティリティ」を参照してください。

注 : このユーティリティを実行するユーザが自分のユニバーサルパスワードを取得できるように、パスワードポリシーが設定されている必要があります。

高度なセキュリティを必要とする環境にユニバーサルパスワードを導入する場合は、次の対策を講じることができます。

1. 次のディレクトリとファイルがセキュリティで保護されていることを確認します。

プラットフォーム	ディレクトリまたはファイル
Windows	◆ %SystemRoot%\SysWOW64\Novell\nici ◆ %SystemRoot%\System32\ (NICI DLL のインストール先)
Linux	◆ /var/opt/novell/nici ◆ /etc/opt/novell/nici64.cfg ◆ /opt/novell/lib64/libccs2.so、および同じディレクトリ内の NICI 共有ライブラリ

NICI および OpenText eDirectory のファイルの具体的な場所について詳しくは、ご使用のシステムのマニュアルを参照してください。

2. 他のセキュリティシステムの場合と同様に、鍵が保管されているサーバへの物理的アクセスを制限することは非常に重要です。

ユニバーサルパスワード

従来、パスワードのさまざまな制限のために、管理者は複数のパスワード (簡易パスワード、NDS パスワード、拡張パスワード) を管理する必要がありました。さらに、複数のパスワードを同期させる必要もありました。

- NDS パスワード : 古い NDS パスワードは、非可逆性のハッシュ形式で保管されます。このパスワードは NDS システム専用であり、他の任意のシステムで使える別のフォームに変換することはできません。
- 簡易パスワード : 簡易パスワードは、当初、管理者がユーザとパスワード (平文およびハッシュ化) を外部 (Active Directory や iPlanet など) の nds-cluster-config ディレクトリからインポートできるようにするために実装されました。

  簡易パスワードには、パスワードポリシー (最小長、有効期限など) が適用されないという制約があります。
- 拡張パスワード : OpenText では拡張パスワードをサポートしなくなっています。拡張パスワードは、ユニバーサルパスワードの前身です。拡張パスワードには何らかのパスワードポリシーが適用されますが、その設計には他のパスワードとの一貫性がありません。このパスワードは、片方向同期を提供し、単純パスワードや NDS パスワードに取って代わるパスワードです。

OpenText では、さまざまなパスワードシステムと認証システムを一貫性のあるネットワークに統合して管理する仕組みを簡素化する手段として、ユニバーサルパスワードを導入しました。

ユニバーサルパスワードは、次のような方法でパスワードの問題に対処します。

- OpenText eDirectory へのすべてのアクセスを 1 つのパスワードで行えます。
- パスワードに拡張文字を使用できるようにします。
- 高度なパスワードポリシーの適用を可能にします。
- OpenText eDirectory から他のシステムへのパスワード同期を可能にします。

パスワード管理のほとんどの機能には、ユニバーサルパスワードが有効になっている必要があります。

詳細については、790 ページの 「ユニバーサルパスワードの導入」を参照してください。

パスワードポリシー

ユニバーサルパスワードにより、高度なパスワードポリシーの作成が可能になります。パスワードポリシーは、エンドユーザパスワードの作成および置き換えに関する基準を指定した管理者定義ルールの集まりです。NMAS では、パスワードポリシーを OpenText eDirectory 内のユーザに割り当てて適用できます。

パスワードポリシーは、OpenText Identity Console を使用して管理します。

詳細については、796 ページの 「パスワードポリシーを使用したパスワードの管理」を参照してください。

パスワード同期

OpenText Identity Manager には、接続システム間でパスワードを同期する機能が含まれています。この機能には、次の利点があります。

- 双方向パスワード同期
- 接続システムへのパスワードポリシーの適用
- 同期失敗時の電子メール通知
- ユーザのパスワード同期ステータスのチェック機能

詳細については、「Connected System Support for Password Synchronization」を参照してください。

非可逆パスワードストレージを理解する

ユニバーサルパスワードは暗号化されてから OpenText eDirectory に保存されます。必要なときにはいつでもこれらのパスワードを OpenText eDirectory で取得できます。たとえば、認証時などです。

ユニバーサルパスワードの代替方法として、OpenText eDirectory 9.2 では、パスワードベースのキー導出関数 2 (PBKDF2) ハッシュアルゴリズム (RFC 2898) を使用してハッシュされたパスワードの保存がサポートされます。パスワードの PBKDF2 ハッシュが有効になっている場合、ユーザのパスワードを取得することはできません。詳細については、815 ページの 「ユニバーサルパスワードの設定オプション」を参照してください。

重要 : OpenText eDirectory 9.2 以降、パスワードポリシーでユニバーサルパスワードを無効にすると、パスワードの PBKDF2 ハッシュが自動的に有効になります。ユニバーサルパスワードが無効になっている既存のパスワードポリシーは、OpenText eDirectory 9.2 にアップグレードする前はユーザに適用されません。しかし、サーバを eDirectory 9.2 にアップグ

レードすると、これらのパスワードポリシーはツリー内のすべてのユーザに自動的に適用されます。これを回避するには、アップグレードする前に、これらのパスワードポリシーのすべての割り当てを削除します。

NDS パスワードから PBKDF2 パスワードへ切り替える場合、SCRAM ログインメソッドに手動で切り替える必要もあります。SCRAM ログインメソッドの詳細については、668 ページの 「パスワード認証」を参照してください。

注

- PBKDF2 ハッシュアルゴリズムを使用して作成されたパスワードは大文字と小文字を区別します。この点、大文字と小文字を区別しない NDS パスワードとは異なります。
- PBKDF2 ハッシュアルゴリズムで作成されたパスワードは、パスワードポリシーの nspmXCharHistoryLimit と nspmXCharLimit ルールをサポートしません。
- デフォルトでは、PBKDF2 は、SHA-256 を使用するように設定されています。ただし、これは nspmPBKDF2HashAlgorithm 属性を使用して、SHA-384 および SHA-512 を使用するように設定できます。この属性を設定する場合、次のいずれかの値を同じフォーマットで指定できます。
  - sha256
  - sha384
  - sha512
- デフォルトでは、PBKDF2 は繰り返しカウント 1 を使用するように設定されています。繰り返し回数は、nspmPBKDF2IterationCount 属性を使って増やすことができます。繰り返しカウントを増加させると、ldap バインドのパフォーマンスが低下します。
- SCRAM ログインメソッドでは、パスワードへの OTP の追加はサポートされません。NDS ログインメソッドをハッシュベース OTP (HOTP) と共に使用するユーザがツリー内にいる場合、そのようなユーザに対して SCRAM ログインメソッドの使用を許可しないでください。

非可逆パスワードストレージの有効化

1. OpenText Identity Console を起動します。
2. OpenText Identity Console のホームページで、[**認証管理**] > [**パスワードポリシー**] の順に選択します。
3. [**パスワードポリシーの作成** ＋] をクリックして、パスワードポリシーウィザードを起動します。
4. ポリシーの名前を入力し、[**次へ**] をクリックします。
5. ユニバーサルパスワードを選択解除し、PBKDF2 パスワードを有効にします。
6. パスワードポリシーウィザードを完了します。

パスワードポリシー

ユニバーサルパスワードにより、高度なパスワードポリシーの作成が可能になります。パスワードポリシーは、エンドユーザパスワードの作成および置き換えに関する基準を指定した管理者定義ルールの集まりです。NMAS では、パスワードポリシーを OpenText eDirectory 内のユーザに割り当てて適用できます。

パスワードポリシーは、OpenText Identity Console を使用して管理します。

詳細については、796 ページの 「パスワードポリシーを使用したパスワードの管理」を参照してください。

ユニバーサルパスワードの導入

このセクションでは、ユニバーサルパスワードを導入して管理する方法について説明します。

ユニバーサルパスワードを導入するには、セクション 2.1 からセクション 2.8 の手順に従います。

- 790 ページの 「ステップ 1: ユニバーサルパスワードの必要性を特定する」
- 791 ページの 「ステップ 2: セキュリティコンテナが使用可能であることを確認する」
- 791 ページの「ステップ 3: SDI ドメインキーサーバがユニバーサルパスワードに対応可能であることを検証する」
- 793 ページの 「ステップ 4: SDI 鍵の一貫性を確認する」
- 793 ページの 「ステップ 5: ユニバーサルパスワードを有効にする」
- 794 ページの 「後方互換性」
- 794 ページの 「パスワードの管理」
- 795 ページの 「注意を要する問題」

ステップ 1: ユニバーサルパスワードの必要性を特定する

次の質問のどちらか一方でも答えが「はい」である場合は、ユニバーサルパスワードを導入して使用する計画を立ててください。

- 各国のユーザに、OpenText の Web ベースのサービスにアクセスさせること、または OpenText Client for Windows を使用して OpenText ファイルとプリントサービスにアクセスさせることを計画していますか。
- OpenText Identity Manager とその高度なパスワードポリシーおよびパスワード同期機能を使用する予定ですか。

ステップ 2: セキュリティコンテナが使用可能であることを確認する

NMAS は、実質的にセキュリティドメインである OpenText eDirectory ツリーにグローバルポリシーを保管します。セキュリティポリシーは、ツリー内のすべてのサーバで使用できる必要があります。

NMAS は、認証ポリシーとログインメソッド設定データを、[ルート] パーティションから分岐して作成されたセキュリティコンテナに格納します。この情報は、NMAS を使用可能なすべてのサーバで読み込みアクセスできる必要があります。セキュリティコンテナの目的は、ログイン、認証、キー管理などのセキュリティプロパティに関するグローバルポリシーを保持することです。

OpenText eDirectory 9.0 以降には、セキュリティコンテナキャッシング機能が備わっています。この機能により、セキュリティコンテナのデータがローカルサーバのキャッシュに保存されるため、NMAS はログインが試行されるたびにセキュリティコンテナにアクセスする必要はありません。671 ページの 「セキュリティオブジェクトのキャッシュ」 を参照してください。

NMAS および eDirectory 8.8.*x* 以降では、セキュリティコンテナを独立したパーティションとして作成し、作成したコンテナを広い範囲で複製することをお勧めします。このパーティションは、ツリー内の信頼性の高い複数のサーバでのみ、読み書き可能なパーティションとして複製することをお勧めします。

警告 : セキュリティコンテナはグローバルポリシーを格納しており、サーバでは OpenText eDirectory ツリーに指定したセキュリティポリシー全般が変更される可能性があるため、書き込み可能なレプリカを配置するサーバを選択する場合には注意が必要です。ユーザが NMAS を使用してログインするためには、ユーザオブジェクトとセキュリティコンテナのレプリカを NMAS サーバ上に配置する必要があります。

詳細については、TID 3393169 を参照してください。

ステップ 3: SDI ドメインキーサーバがユニバーサルパスワードに対応可能であることを検証する

SDI ドメインキーサーバが最小構成要件を満たしており、ツリー内の他のサーバによる配布と使用に一貫性のある鍵を使用していることを検証する必要があります。以下の手順は非常に重要です。概説されているとおりに従わないと、ユニバーサルパスワードを有効にしたときに、重大なパスワードの問題が発生する可能性があります。

1. Windows サーバのコマンドプロンプトで、sdidiag.exe を実行します。

   sdidiag.exe は、OpenText eDirectory には付属していません。sdidiag.exe をインストールしてから実行します。このファイルは、TID 2974092 に関連付けられたセキュリティパッチの一部として使用可能になります。
2. 管理者として、サーバ (フルコンテキスト)、ツリー名、ユーザ名、およびパスワードを入力してログインします。

3 すべてのサーバが 3DES ツリー鍵として 168 ビットの鍵を使用していること、AES 256 ビットのツリー鍵として 256 ビットの鍵を使用していることを確認します。

この要件を満たしていることを確認するには、TID 3364214 の指示に従ってください。

4 コマンド CHECK -v >> *インストールフォルダ*\sdinotes.txt を入力します。

画面に、CHECK コマンドの結果が表示されます。

5 問題がない場合は、793 ページの 「ステップ 4: SDI 鍵の一貫性を確認する」に進みます。

または

インストールフォルダ\sdinotes.txt ファイルに記載された説明に従って、構成と鍵の問題を解決してからステップ 6 に進みます。

6 SDI ドメインキーサーバ上で NICI *3.0* が実行されていることを検証します。

古いバージョンの場合、OpenText eDirectory を 9.0 以降にアップグレードします。これにより、NICI が 3.0 以降にアップグレードされます。

7 (オプション)SDIDIAG CHECK コマンドを再実行します。詳細については、ステップ 4 を参照してください。

SDIDIAG の使用方法の詳細については、TID 3364214 を参照してください。

SDI ドメインキーサーバの追加または削除

SDI ドメインキーサーバであるサーバを削除するには、次の手順に従います。

1 sdidiag.exe は、OpenText eDirectory には付属していません。sdidiag.exe は、ソフトウェアライセンスおよびダウンロード (https://sld.microfocus.com) ポータルからダウンロードできます。ダウンロードした後、sdidiag.exe を実行します。

2 セキュリティコンテナと W0.KAP.Security オブジェクトに対する管理権を持つ管理者として、サーバ (フルコンテキスト)、ツリー名、ユーザ名、およびパスワードを入力してログインします。

3 コマンド RS -s *servername* を入力します。

たとえば、Novell_Inc ツリー内の組織 Novell に含まれるコンテナ PRV に server1 が存在する場合は、.server1.PRV.Novell.Novell_Inc をサーバ名として入力します。

サーバを SDI ドメインキーサーバとして追加するには、次の手順に従います。

1 Windows サーバで、コマンドプロンプトボックスを開き、sdidiag.exe を実行します。

2 管理者として、サーバ (フルコンテキスト)、ツリー名、ユーザ名、およびパスワードを入力してログインします。

3 コマンド AS -s *servername* を入力します。

たとえば、Novell_Inc ツリー内の組織 Novell に含まれるコンテナ PRV に server1 が存在する場合は、.server1.PRV.Novell.Novell_Inc をサーバ名として入力します。

ステップ 4: SDI 鍵の一貫性を確認する

ツリー全体で、暗号鍵のすべてのインスタンスに一貫性があることを検証します。各サーバに、ツリー内の他のサーバとセキュアに通信するために必要な暗号鍵があることを確認するには、次の手順に従います。

1 Windows サーバのコマンドプロンプトで、sdidiag.exe を実行します。

2 コマンド CHECK -v >> sys:system\sdinotes.txt -n *container DN* を入力します。

たとえば、Acme_Inc ツリー内の組織 Acme に含まれるコンテナ USR にユーザ Bob が存在する場合は、.USR.Acme.Acme_Inc. をコンテナ識別名 (DN) として入力します。

このコマンドは、さまざまなサーバとキードメインの間で鍵の一貫性に関するすべての問題を報告します。

画面に、CHECK コマンドの結果が表示されます。

3 問題が報告されていない場合は、ユニバーサルパスワードを有効にできる状態になっています。793 ページの 「ステップ 5: ユニバーサルパスワードを有効にする」に進みます。

または

問題が報告された場合は、sdinotes.txt ファイルの指示に従います。

ほとんどの場合、コマンド RESYNC -T を実行するよう求められます。NMAS がユニバーサルパスワードの認証中に -1418 または -1460 エラーを報告した場合は、このコマンドを繰り返し実行できます。

SDIDIAG のオプションと操作方法の詳細については、以下を参照してください。

- TID 3364214
- TID 7005397

ステップ 5: ユニバーサルパスワードを有効にする

1 OpenText Identity Console にログインします。

2 OpenText Identity Console のホームページで、[**認証管理**] > [**パスワードポリシー**] の順にクリックします。

3 [**パスワードポリシーの作成** ＋] をクリックして、パスワードポリシーウィザードを起動します。

4 ポリシーの名前を入力し、[**次へ**] をクリックします。

5 [**設定**] タブ > [**全般**] > [**ユニバーサルパスワードを有効にする**] の順にクリックします。

6 パスワードポリシーウィザードを完了します。

重要 : コンテナ パーティションのルートになっているコンテナにポリシーを割り当てると、ポリシー割り当てはサブコンテナのユーザを含めたパーティション内のすべてのユーザに継承されます。コンテナがパーティションルートであるかどうかを知るには、そのコンテナを参照して、隣にパーティションアイコンが表示されているかどうかを確認します。

パーティションのルートではないコンテナにポリシーを割り当てると、そのポリシー割り当ては、その特定のコンテナ内のユーザだけに継承されます。サブコンテナ内のユーザには継承されません。パーティションのルートではないコンテナの下位にあるユーザすべてにポリシーを適用する場合は、それぞれのサブコンテナに個別のポリシーを割り当てる必要があります。

後方互換性

ユニバーサルパスワードは、既存のサービスに対する後方互換性を確保するように設計されています。デフォルトでは、このサービスで変更されたパスワードは、ユーザオブジェクトの簡易パスワードおよび NDS パスワードと同期することができます。［認証管理］の［パスワードポリシー］を使用して、同期するパスワードを選択できます。

ただし、パスワードで国際文字が使用されている場合は例外です。古いクライアントでは国際文字がさまざまに変換されるため、実際の値が一致しなくなります。システム全体の国際文字を使用したパスワードのすべてが正常に機能するよう、すべての OpenText Client をアップグレードすることをお勧めします。

OpenText およびサードパーティのバックアップおよび復元アプリケーションには、NetWare SMS (Storage Management Services) インフラストラクチャが使用されます。これらの OpenText およびサードパーティ製品が混合環境で動作することになっている場合、製品で使用するシステムパスワードに拡張文字を含めることはできません。

注 : ユニバーサルパスワード対応のアプリケーションおよびサービスと、拡張文字対応のアプリケーションおよびサービスを調べるには、TID 3065822 を参照してください。多くのアプリケーションやサービスでは、ユニバーサルパスワードを使用しない場合は拡張文字を使用できます。

パスワードの管理

ユニバーサルパスワードを管理するのには、次の手段を使用できます。

- **OpenText Identity Console (推奨):** OpenText Identity Console を使用してパスワードを管理する場合、後方互換性を確保するために、ユニバーサルパスワードは簡易パスワードおよび NDS パスワードの値と同期されるように自動的に設定されます。OpenText Identity Console の［**認証管理**］>［**パスワードポリシー**］で、システムにインストールされて設定済みの個々のパスワードと認証方式をきめ細かく管理できます。

  OpenText Identity Console の［パスワード管理］タイルでは、パスワードポリシーを使用して、NDS パスワード、簡易パスワード、および配布パスワードとのユニバーサルパスワードの同期方法を指定できます。さらに、管理者がユーザのユニバーサルパスワードを設定するために使用できる OpenText Identity Console タスクが用意されています。

注意を要する問題

- ユーザの NDS パスワードを無効にすると、NDS パスワードは、ユーザにとって不明の任意の値に設定されます。以下に、ログインメソッドがこの変更を処理する方法をリストして説明します。
  - NDS パスワードが無効にされても、簡易パスワードメソッドは無効にされません。ユニバーサルパスワードが有効にされて使用可能になっている場合、簡易パスワードメソッドはユニバーサルパスワードを使用します。それ以外の場合は、簡易パスワードを使用します。ユニバーサルパスワードが有効にされていても、設定されていない場合、簡易パスワードメソッドは簡易パスワードを使用してユニバーサルパスワードを設定します。
  - NDS パスワードが無効にされても、拡張パスワードメソッドは無効にされません。拡張パスワードメソッドは、ログインにはユニバーサルパスワードを使用しません。
  - NDS パスワードが無効にされても、NDS パスワードメソッド (ユニバーサルパスワード) は無効にされません。ユニバーサルパスワードが有効にされて使用可能になっている場合、NDS パスワードメソッドはユニバーサルパスワードを使用します。それ以外の場合は、NDS パスワードを使用します。ユニバーサルパスワードが有効にされていても、設定されていない場合、NDS パスワードメソッドは NDS パスワードを使用してユニバーサルパスワードを設定します。
- 新しいユーザを作成する場合やヘルプデスクへの問い合わせに応答する場合などに、管理者がユーザのパスワードを変更する場合、パスワードポリシーでパスワードを期限切れにする設定が有効になっていると、セキュリティ上の理由からパスワードは自動的に失効します。パスワードを期限切れにする設定とは、パスワードポリシーの［**高度なパスワードルール**］にある［**Number of days before password expires (0-365)**］です。この特定の機能については、日数は重要ではありませんが、この設定を有効にする必要があります。

  注 : この振る舞いを無効にするには、パスワードポリシーで［**Do not expire the user's password when the administrator sets the password**］オプションを選択します。

- パスワードポリシーを作成してユニバーサルパスワードを有効にし、さらに高度なパスワードルールを有効にすると、NDS パスワークの既存のパスワード設定ではなく、有効にされた高度なパスワードルールが適用されます。従来のパスワード設定は無視されます。パスワードポリシーを作成しても、以前の設定が自動的にマージまたはコピーされることはありません。

  たとえば、NDS パスワードで使用する猶予ログイン回数を設定している場合、ユニバーサルパスワードを有効にするときに、パスワードポリシーの高度なパスワードルールで猶予ログイン回数の設定を再作成する必要があります。

  NMAS では、ユーザオブジェクトの NDS パスワード設定を、対応するパスワードポリシー設定に置き換えます。たとえば、猶予ログイン回数がユーザオブジェクトでは 4 に設定されていて、パスワードポリシーでは 5 に設定されている場合、ユーザがログインした時点、またはパスワードが変更された時点で、そのユーザオブジェクトの猶予ログイン回数は 5 に変更されます。

パスワードポリシーを使用したパスワードの管理

パスワードポリシーを使用してユーザが自分のパスワードを作成する方法に対するルールを設定することで、セキュリティを強化できます。ユーザ自身でパスワード忘れや、パスワードリセットに対処できるセルフサービスオプションを付加することにより、ヘルプデスクの運用コストの削減にもつながります。

このセクションで説明する内容は以下のとおりです。

- 796 ページの 「パスワードポリシー機能の概要」
- 797 ページの 「パスワードポリシーの計画」
- 801 ページの 「パスワードポリシーを使用するために必要な事前タスク」
- 803 ページの 「パスワードポリシーの作成」
- 820 ページの 「ユーザへのパスワードポリシーの割り当て」
- 822 ページの 「ユーザに適用されているポリシーの識別」
- 822 ページの 「ユーザのパスワードの設定」
- 823 ページの 「ユニバーサルパスワードの診断ユーティリティ」
- 825 ページの 「パスワードポリシーのトラブルシューティング」

パスワードポリシー機能の概要

Password Policy (パスワードポリシー) は、エンドユーザパスワードの作成および交換に関する基準を指定した管理者定義ルールの集まりです。NMAS では、パスワードポリシーを OpenText eDirectory 内のユーザに割り当てて適用できます。

パスワードポリシーにパスワード忘れセルフサービス機能を組み込んで、パスワードを忘れた場合のヘルプデスクへの問い合わせ件数を削減することもできます。セルフサービス機能としては、パスワードリセットセルフサービスもあります。パスワードリセットセルフサービスでは、ユーザが管理者によってパスワードポリシーに指定されたルールを確認しながら、自分のパスワードを変更することができます。ユーザはこれらの機能に、Identity Manager ユーザアプリケーションまたは OpenText Identity Console からアクセスします。

パスワードポリシーで高度なパスワードルール、パスワード同期、およびパスワード忘れセルフサービスのさまざまな機能を使用するには、ユーザのユニバーサルパスワードを有効にする必要があります。ユニバーサルパスワードの導入方法については、790 ページの「ユニバーサルパスワードの導入」を参照してください。

パスワードポリシーを作成するには、パスワードポリシーウィザードを使用します。OpenText Identity Console で、［**認証管理**］>［**パスワードポリシー**］>［**パスワードポリシーの作成** ＋］の順にクリックします。パスワードポリシーの作成方法の詳細については、803 ページの 「パスワードポリシーの作成」を参照してください。

パスワードポリシーの計画

- 797 ページの 「ツリー内でパスワードポリシーを割り当てる方法の計画」
- 797 ページの 「パスワードポリシーのルールの計画」
- 799 ページの 「ユーザのログインおよびパスワードメソッド変更の計画」

ツリー内でパスワードポリシーを割り当てる方法の計画

管理を簡素化するために、デフォルトポリシーをツリー全体に割り当て、その他に使用するポリシーをツリーのできるだけ高い階層に割り当てることをお勧めします。

NMAS は、ユーザに適用されているパスワードポリシーを判別します。詳細については、820 ページの 「ユーザへのパスワードポリシーの割り当て」を参照してください。

パスワードポリシーのルールの計画

パスワードポリシーで高度なパスワードルールを使用することで、パスワードにビジネスポリシーを適用できます。

Novell Client (4.9.1)、Identity Manager ユーザアプリケーション、および OpenText Identity Console には、パスワードポリシーに含まれるパスワードルールが表示されることに留意してください。ユーザが LDAP サーバを介して、または接続システムで自分のパスワードを変更する場合に備え、ユーザがパスワードルールをすぐに利用できるようにして、ルールに準拠したパスワードが作成されるようにする必要があります。

Identity Manager のパスワード同期機能を使用する場合は、パスワードポリシーが割り当てられているユーザが、接続システムのパスワード同期に参加させるユーザと一致していることを確認する必要があります。パスワードポリシーはツリー中心で割り当てられます。これに対し、パスワード同期は、各サーバのドライバごとに設定されます。パスワード同期で期待される結果を出すためには、パスワード同期用のドライバを実行するサーバ上の読み書き可能レプリカまたはマスタレプリカに含まれるユーザが、ユニバーサルパスワードが有効にされたパスワードポリシーを割り当てたコンテナと一致するようにします。パーティションルートコンテナにパスワードポリシーを割り当てることによって、そのコンテナとサブコンテナ内のすべてのユーザに確実にパスワードポリシーが割り当てられます。

高度なパスワードルール

高度なパスワードルールでは、ユニバーサルパスワードに次の条件を定義できます。

- パスワードの有効期間 : パスワードポリシーは、過去に OpenText eDirectory で提供していたポリシー機能と同じ機能を提供するため、パスワードの変更頻度や再使用可能かどうかを指定できます。
- パスワードに含まれる文字 : 文字、数字、小文字または大文字および特殊文字を組み合わせることができます。安全性の低いと思われるパスワード (企業名など) は排除できます。パスワードに含める、以前のパスワードで使用されていない「新しい」文字の数を要件としたり、指定のパスワードで許容するパスワードポリシー違反の数を設定したりすることもできます。

パスワードポリシーで高度なパスワードルールを使用するには、ユニバーサルパスワードを有効にする必要があります。ポリシーにユニバーサルパスワードが有効にされていない場合は、代わりに NDS® パスワードを対象に設定されている制限が適用されます。

注 : パスワードポリシーを作成してユニバーサルパスワードを有効にすると、NDS パスワードに対する既存のパスワード設定の代わりに、高度なパスワードルールが適用されます。従来のパスワード設定は無視されます。パスワードポリシーを作成しても、以前の設定が自動的にマージまたはコピーされることはありません。

たとえば、NDS パスワードで使用する猶予ログイン回数を設定している場合、ユニバーサルパスワードを有効にするときに、パスワードポリシーの高度なパスワードルールで猶予ログイン回数の設定を再作成する必要があります。

後でパスワードポリシーでユニバーサルパスワードを無効にすると、既存のパスワード設定は無視されなくなります。既存の設定が、NDS パスワードに適用されることになります。

NMAS 3.1 以降では、ユーザオブジェクトの NDS パスワード設定を、対応するパスワードポリシー設定に置き換えます。たとえば、猶予ログイン回数がユーザオブジェクトでは 4 に設定されていて、パスワードポリシーでは 5 に設定されている場合、ユーザがログインした時点、またはパスワードが変更された時点で、そのユーザオブジェクトの猶予ログイン回数は 5 に変更されます。

ポリシーの適用

ツリー内のユーザにパスワードポリシーを割り当てると、そのポリシーの高度なパスワードルールに準拠していない限り、パスワード変更は適用されません。Novell Client 4.9 SP2 以降では、ルールも表示されます。アクセスにどちらのメソッドを使用するとしても、ルールに準拠していないパスワードは拒否されます。NMAS は、これらのルールを適用するアプリケーションです。

ポリシーに既存のパスワードのコンプライアンスをチェックするように指定することで、ユーザはルールに準拠していない既存のパスワードを変更しなければなくなります。コンプライアンスのチェックオプションが有効にされている場合、パスワードポリシーのルールを満たしていないパスワードには、失効したパスワードとしてマークが付けられます。

ユーザがポータルを介して認証する際に、有効にされているパスワード忘れセルフサービス機能の設定を求めるプロンプトを出すように指定することもできます。これは、認証後サービスと呼ばれます。たとえば、パスワードを忘れた場合に電子メールで送信するパスワードのヒントをユーザに設定させたい場合、認証後サービスを使用して、ユーザのログイン時にパスワードのヒントを設定するよう求めるプロンプトを出すことができます。

認証後サービスの設定は、［パスワードを忘れた場合］プロパティページの最後のオプションです。

ユーザのログインおよびパスワードメソッド変更の計画

ユーザがログインしたりパスワードを変更したりするには、いくつかの方法があります。ユニバーサルパスワードをサポートするためのアップグレードの詳細については、790 ページの 「ユニバーサルパスワードの導入」を参照してください。

このセクションでは、それぞれのケースでユニバーサルパスワードをサポートするための追加要件について説明します。

- 799 ページの 「Novell Client」
- 800 ページの 「Identity Manager ユーザアプリケーションと OpenText Identity Console」
- 801 ページの 「その他のプロトコル」
- 801 ページの 「接続システム」

Novell Client

Novell Client を使用している場合は、バージョン 4.9 SP2 以降にアップグレードします。

ユーザは OpenText Identity Console や、環境に応じてその他の企業のポータルからログインできるので、Novell Client の使用は必須ではありません。また、Active Directory でのパスワード同期に、Novell Client は必要なくなっています。

以下の表に、ユニバーサルパスワードに関する Novell Client バージョン間の違いについての説明と、レガシー Novell Client を操作する際の推奨事項を記載します。

表 26-1 Novell Client でのユニバーサルパスワード

Novell Client バージョン (Novell Client Version)	ログイン	パスワードの変更
4.9 より前	NMAS を介さないため、ユニバーサルパスワードはサポートされません。NDS パスワードを使用して直接ログインします。	NMAS を介する代わりに、NDS パスワードを直接変更します。 ユニバーサルパスワードを使用している場合、NDS パスワードとユニバーサルパスワードが同期された状態に維持されない可能性があります。この問題を防ぐには、次の 3 つのオプションがあります。 ◆ すべてのクライアントをバージョン 4.9 以降にアップグレードします。 ◆ コンテナの属性値を使用して、レガシークライアントがパスワードを変更できないようブロックします。このソリューションの場合、レガシークライアントは引き続きログインできますが、パスワードを変更することはできません。パスワード変更は、必ず新しいバージョンの Novell Client または OpenText Identity Console を使用して行う必要があります。 ◆ [Remove the NDS Password when Setting Universal Password] のパスワードポリシー設定を使用します。この場合、NDS パスワードを使用したログインとパスワード変更がどちらも防止されるので、抜本的な対策になります。
4.9	ユニバーサルパスワードをサポートしています。	ユニバーサルパスワードにパスワードポリシーのルールを適用します。 ユーザがルールに準拠していないパスワードを作成しようとすると、パスワード変更は拒否されます。ただし、ルールのリストは、ユーザには表示されません。
4.9 SP2 以降	ユニバーサルパスワードをサポートしています。	ユニバーサルパスワードにパスワードポリシーのルールを適用します。 さらに、ルールに準拠したパスワードを作成できるよう、ユーザにルールを表示します。

Identity Manager ユーザアプリケーションと OpenText Identity Console

Identity Manager ユーザアプリケーションと OpenText Identity Console が提供するパスワードセルフサービスでは、ユーザがパスワードをリセットしたり、パスワード忘れセルフサービス (パスワードポリシーで指定されている場合) を設定したりできます。

- パスワードポリシーでは、[Synchronize NDS password when setting Universal Password] のデフォルト設定を受け入れることをお勧めします。

その他のプロトコル

ユニバーサルパスワードがサポートされるよう、OpenText eDirectory、LDAP サーバ、NMAS、および OpenText Identity Console がアップグレードされていることを確認します。

ユニバーサルパスワードで AFP、CIFS、およびその他のプロトコルを使用する方法の詳細については、790 ページの 「ユニバーサルパスワードの導入」を参照してください。

接続システム

Identity Manager のパスワード同期を使用する場合は、ユーザパスワード変更が正常に行われるよう、次の要件を満たしていることを確認してください。

- システムのすべての Identity Manager ドライバが、Identity Manager フォーマットにアップグレードされていること。
- Identity Manager ドライバの環境設定に、新しいパスワード同期ポリシーが含まれていること。
- パスワード同期の設定で、ユニバーサルパスワードの使用が設定されていること。双方向パスワード同期が必要な場合は、配布パスワードも指定されていること。
- 必要に応じて、パスワードをキャプチャするパスワードフィルタが接続システムに導入されていること。

詳細については、『OpenText Identity Manager 4.10 Password Management Guide (OpenText Identity Manager 4.5 パスワード管理ガイド) (https://www.netiq.com/documentation/identity-manager-4.10/)』の「*Connected System Support for Password Synchronization (接続システムでのパスワード同期のサポート)* (https://www.netiq.com/documentation/idenity-manager-4.10/)」を参照してください。

パスワードポリシーを使用するために必要な事前タスク

パスワードポリシーのすべての機能を利用するには、環境を準備するために行っておかなければならない手順があります。

OpenText Identity Console アプリケーションは、ソフトウェアライセンスおよびダウンロードポータルからダウンロードできます。

1. ユニバーサルパスワードをサポートするように環境をアップグレードします。

   詳細については、790 ページの 「ユニバーサルパスワードの導入」を参照してください。
2. ユニバーサルパスワードをサポートするようにクライアント環境をアップグレードします。

   799 ページの 「ユーザのログインおよびパスワードメソッド変更の計画」および 790 ページの 「ユニバーサルパスワードの導入」を参照してください。

重要 : OpenText Identity Console アプリケーションを実行すると、OpenText Identity Console が RAC モードで実行されます。この場合、管理者が自分自身に特定の役割を割り当てていない限り、管理者にはタスクが 1 つも表示されません。管理者には必ず、すべての OpenText Identity Console タスクにアクセスできる役割を割り当ててください。

重要 : OpenText eDirectory をアップグレードせずに最新バージョンの OpenText Identity Console にアップグレードした後、パスワードポリシーを変更または作成すると、OpenText Identity Console にエラーが表示されます。

3 OpenText eDirectory 内に、単純認証に対して TLS を要求するための LDAP グループサーバオブジェクトを設定します。

これは、OpenText Identity Console を設定する場合のデフォルト設定です。単純認証に対して TLS を要求することは、パスワードセルフサービス機能で推奨されており、OpenText Identity Console で、[**認証管理**] タイル > [**パスワードポリシー**] を使用するために必要です。

単純認証に対して TLS を要求する場合、LDAP SSL ポートに必要な追加設定はありません。

重要 : 単純認証に対して TLS を要求しないと、ユーザがクリアテキストパスワードを使用して OpenText Identity Console にログインすることを許可することになります。

このオプションを使用することもできますが、別のステップが必要になります。

デフォルトで、パスワードセルフサービス機能が LDAP SSL ポートとして想定するのは、PortalServlet.properties ファイル内の System.DirectoryAddress 設定で指定されているポートです。LDAP SSL ポートがそれとは異なる場合は、PortalServlet.properties ファイルに次の鍵ペアを追加して、正しいポートを指示する必要があります。

`LDAPSSLPort=`*`your_port_number`*

たとえば、Tomcat を実行しているとしたら、この鍵ペアを tomcat\webapps\nps\WEB_INF ディレクトリ内にある PortalServlet.properties ファイルに追加します。

4 パスワード忘れ機能のための電子メール通知を有効にするには、SMTP サーバを設定し、電子メールテンプレートをカスタマイズします。

これで、パスワードポリシーのすべての機能を利用できる状態になりました。803 ページの 「パスワードポリシーの作成」の説明に従って、ポリシーを作成します。

パスワードポリシーの作成

OpenText Identity Console で、［**認証管理**］タイル > ［**パスワードポリシー**］を使用して、新しいパスワードポリシーを作成します。

ウィザードの各ステップについては、オンラインヘルプを参照してください。

1. OpenText Identity Console のホームページにログインして、［**認証管理**］> ［**パスワードポリシー**］の順に選択します。
2. ［**パスワードポリシーの作成** ＋］をクリックして、新しいパスワードポリシーを作成します。
3. ウィザードで表示される指示に従い、ポリシーの高度なパスワードルール、ユニバーサルパスワードの設定オプション、およびパスワード忘れの選択項目を作成します。
4. 必要に応じて、個々のユーザ、組織、または会社全体にパスワードポリシーを割り当てます。
5. 新しいポリシーの設定を確認し、［**作成**］をクリックして、［**OK**］をクリックします。

高度なパスワードルール

図 26-1 に、高度なパスワードルールの最初のセクションを示します。

図 26-1 *高度なパスワードルール*



パスワードの構文

パスワードポリシーで使用するパスワードの構文には、次の 3 つのオプションのいずれかを指定できます。

- **Microsoft の複雑さのポリシーを使用する**
- **Use Microsoft Server 2008 Password Policy**
- **Novell の構文を使用する**

警告 : OpenText Identity Console では、サーバにインストールされている NMAS のバージョンに関係なく、Microsoft Server 2008 のパスワードポリシーのタイプを使用してポリシーを作成することができます。ただし、このオプションを使用するには、NMAS 3.3.4 以降がインストールされている必要があります。これより前のバージョンの NMAS がインストールされている場合は、新しいパスワードポリシーは正常に機能しません。

- **Microsoft の複雑さのポリシーを使用する**

  この設定では、Microsoft* の複雑さのポリシーの要件を適用することができます。OpenText eDirectory と Microsoft Active Directory の間でパスワードを同期させる必要がある場合は、このオプションを使用します。

  ポリシーにこのオプションが選択されている場合、このポリシーが割り当てられているすべてのユーザは、ユニバーサルパスワードに実装された Microsoft の複雑さのポリシーの条件を満たすパスワードを作成しなければなりません。以下の条件があります。

  - 最小パスワード長は 6 文字です。
  - 最大パスワード長は 128 文字です。
  - パスワードには、大文字、小文字、数字、および特殊文字の 4 つのタイプのうち、3 つのタイプの文字が少なくとも 1 つ含まれる必要があります。
    - 大文字 - Basic Latin および Latin-1 文字セットに含まれる文字の大文字すべて。
    - 小文字 - Basic Latin および Latin-1 文字セットに含まれる文字の小文字すべて。
    - 数字 - 0、1、2、3、4、5、6、7、8、9。
    - 特殊文字 - その他すべての文字。
  - ユーザ属性 CN、Given Name、Surname、Full Name、および displayName の値をパスワードに含めることはできません。
  - OpenText eDirectory アカウントのユーザ属性 CN の値そのものをパスワードに含めることはできません。属性の長さが 3 文字未満の場合、NMAS はこのチェックを行いません。

- **Use Microsoft Server 2008 Password Policy**

  この設定では、Microsoft* Windows Server 2008 パスワードポリシーの複雑性の要件を適用することができます。OpenText eDirectory と Microsoft Active Directory の間でパスワードを同期させる必要がある場合は、このオプションを使用します。

  ポリシーにこのオプションが選択されている場合、このポリシーが割り当てられているすべてのユーザは、ユニバーサルパスワードに実装された Microsoft Windows Server 2008 の複雑さのポリシーの条件を満たすパスワードを作成しなければなりません。このオプションを選択すると、［高度なパスワードルール］ページのいくつかのオプションが、この複雑さのポリシーの条件を満たすように設定されます。以下の条件があります。

  - 最小パスワード長は、デフォルトでは 7 文字です。［**Minimum number of characters in password (1-512)**］オプションを使用して、環境の最小パスワード長を設定できます。最小文字数の設定方法の詳細については、810 ページの 「パスワード長」を参照してください。
  - 最大パスワード長は 512 文字です。

- パスワードには、大文字、小文字、数字、非英数字、およびその他の文字の 5 つのタイプのうち、3 つのタイプの文字が少なくとも 1 つ含まれる必要があります。
  - 大文字 - 発音区別符付きを含むヨーロッパ言語の文字、ギリシャ文字、キリル文字の大文字すべて。
  - 小文字 - 発音区別符付きを含むヨーロッパ言語の文字、ギリシャ文字、キリル文字の小文字すべて。
  - 数字 - 0、1、2、3、4、5、6、7、8、9。
  - 非英数字 - 次の特殊文字すべて : ( ) ` ~ ! @ # $ % ^ & * - + = | \ { } [ ] : ; " ' < > , . ? / _
  - その他の文字 - 英字として分類されている一方、大文字でも小文字でもない Unicode 文字すべて。これには、アジア言語の Unicode 文字が含まれます。
- 除外するパスワードのリストに含まれる単語をパスワードに含めることはできません。除外するパスワードの長さが 3 文字未満の場合、NMAS はこのチェックを行いません。除外パスワードの詳細については、809 ページの 「パスワード除外」を参照してください。
- アカウントのCN属性またはFull Name属性の値が3文字以上の1つの単語である場合、CN 属性全体または Full Nama 属性の全体または一部をパスワードに含めることはできません。属性の値の一部は 3 文字以上の連続する文字として定義され、その両端はコンマ、ピリオド、ダッシュ、ハイフン、下線、スペース、シャープ記号、またはタブで区切られます。

  注 : Microsoft 2008 パスワードポリシーを使用している場合、CN 属性と displayName 属性は、AD における samAccountName ルールと displayName ルールと同様と見なされます。

- パスワードで許容される複雑さのポリシー違反の最大数は、デフォルトでは 2 です。[**Maximum number of complexity policy violations in password (0-5)**] オプションを使用して、パスワードで許容される複雑さのポリシー違反の最大数を設定できます。違反の最大許容数の設定方法の詳細については、811 ページの 「Password Complexity Violations」を参照してください。

- **Novell の構文を使用する**

  この設定では、パスワードポリシーに Novell の構文を使用できます。このオプションはデフォルトで選択されています。Novell の構文を使用したポリシーの標準設定は、次のとおりです。
  - 最小パスワード長は、デフォルトでは 4 文字です。[**パスワードの最小文字数 (1 ～ 512)**] オプションを使用して、環境の最小パスワード長を設定できます。最小文字数の設定方法の詳細については、810 ページの 「パスワード長」を参照してください。
  - 最大パスワード長は、デフォルトでは 12 文字です。[**Minimum number of characters in password (1-512)**] オプションを使用して、環境の最大パスワード長を設定できます。最大文字数の設定方法の詳細については、810 ページの 「パスワード長」を参照してください。

パスワード構文の優先順位

OpenText Identity Console のパスワードポリシー外で、ディレクトリ管理または LDAP を使用してパスワードポリシーの属性を変更すると、パスワードポリシーの 1 つ以上のタイプの間に競合が発生する可能性があります。たとえば、LDAP を使用して、同じパスワードポリシーに対して Microsoft の複雑さのポリシーと Microsoft Windows 2008 パスワードポリシーを有効にするなどです。

競合が発生した場合、OpenText eDirectory は次の優先順位に従います。

- Microsoft Server 2008 パスワードポリシー
- Microsoft の複雑さのポリシー
- Novell 構文

パスワード管理インタフェース外でのパスワードポリシー変更の詳細については、814 ページの 「パスワードポリシーインタフェースの外部でパスワードポリシーを変更する」を参照してください。

パスワードの変更

- **Do not expire the user's password when the administrator sets the password**

  このオプションでは、ユーザが自分のパスワードにアクセスして変更することが必要になります。この機能により、デフォルトを上書きできます。パスワードの有効期限が設定されている場合、OpenText eDirectory ではデフォルトの振る舞いとして、ユーザのパスワードは、管理者がパスワードを設定する際に失効します。

- **固有パスワードを要求する**

  このオプションが選択されている場合、ユーザはパスワードを履歴リストに含まれているパスワードに変更できなくなります。ユーザがパスワードを変更して履歴リストに含まれているパスワードを再利用しようとすると、そのパスワードはパスワードポリシーによって拒否され、ユーザは別のパスワードを指定するよう求められます。

  次の 2 つの値のいずれかを使用することで、固有パスワードを強制する方法を指定できます。

  - [**Remove password from history list after a specified number of days (0-365)**] と [**History list size (1-255)**]。

    固有パスワードを要求する場合は、照合用の履歴リストに以前に使用したパスワードを保管する日数を指定できます。

    たとえば、制限を 30 日間に指定した場合、ユーザが以前に「mountains99」というパスワードを使用したとすると、そのパスワードが履歴リストに 30 日間維持されます。この期間中に、ユーザが自分のパスワードを変更して「mountains99」を再利用しようとすると、そのパスワードはパスワードポリシーによって拒否され、ユーザは別のパスワードを指定するよう求められます。30 日間が過ぎると、古いパスワードは照合用に保管されなくなり、そのパスワードを再利用しても、パスワードポリシーによって拒否されなくなります。

    固有パスワードを要求する場合、照合用の履歴リストに保管するパスワードの数を指定することもできます。たとえば、値を 3 に指定すると、ユーザが以前に使用した 3 つのパスワードが保管されます。履歴リストから削除するまでの日数が

経過する前に、ユーザが自分のパスワードを変更して履歴リストに含まれているパスワードを再利用しようとすると、そのパスワードはパスワードポリシーによって拒否され、ユーザは別のパスワードを指定するよう求められます。

注

- ［Use Microsoft Server 2008 Password Policy］オプションを選択すると、デフォルトで［固有パスワードを要求する］オプションも選択されます。
- ［固有パスワードを要求する］が選択されていて、［Remove password from history list after a specified number of days (0-365)］を選択し、日数を指定しなかった場合、［Password Lifetime］セクションの［Number of days before password expires (0-365)］フィールドに設定した値の 8 倍の期間中、パスワードが履歴リストに維持されます。どちらのフィールドにも値が設定されていない場合、パスワードは 365 日間、履歴リストに維持されます。
- パスワード履歴リストのサイズと日数を指定し、パスワード履歴リストのパスワード数が指定したサイズに達している場合、パスワードの期限が切れるまで、ユーザは自分のパスワードを変更できません。管理者は、パスワード履歴リストのサイズに達していても、ユーザのパスワードを変更または設定できます。
- パスワード履歴リストの 1 つ以上のパスワードが期限切れになって、リストがいっぱいでなくなると、ユーザは再び自分のパスワードを変更できるようになります。この制限が設けられている理由は、ユーザが何度も自分のパスワードを変更したためにパスワードがパスワード履歴リストに追加されなくなり、ユーザが以前に使用したパスワードを再利用できるようになるという事態を防ぐためです。
- パスワード履歴リストのサイズが指定されていない場合、パスワードの履歴がいっぱいになることはありません。
- 指定されたパスワードをパスワード履歴リストの以前に使用されたパスワードを比較する際の OpenText eDirectory の振る舞いは Active Directory とは異なります。パスワード履歴リストのサイズが「N」に設定されている場合、Active Directory は、指定されたパスワードを、以前に使用された N 個のパスワードと比較します。一方、OpenText eDirectory は指定されたパスワードを以前に使用された「n+1」個のパスワードと比較します。

- ［Remove password from history list when the list is full］と［History list size (1-255)］。

  固有のパスワードが必要な場合は、比較のために履歴リストに保存されるパスワードの数を指定できます。このオプションは先入れ先出しで機能し、最も古いパスワードが履歴リストから削除されます。たとえば、履歴リストがいっぱいの場合、履歴リストに現在存在しない新しいパスワードをユーザが作成すると、履歴リスト内の最も古いパスワードが削除されます。

注

- ［Use Microsoft Server 2008 Password Policy］オプションを選択すると、［Remove password from history list when the list is full］オプションもまたデフォルトで選択されます。Microsoft Server 2008 構文が有効である場合、履歴リストサイズ範囲は 0 個～ 24 個のパスワードです。

- このオプションを選択した場合、[Number of days before password can be changed]および[Number of days before password expires]の両方のオプションを選択し、それぞれに最小日数を指定してください。
- パスワード履歴リストのサイズに 0 を指定すると、NMAS は、ユーザが作成した新しいパスワードをそのユーザの現在のパスワードのみと比較します。

- [Number of characters different from current password and passwords from history (0-6)]、および指定した文字数。。

  このオプションを選択した場合、ユーザは、以前のパスワードで未使用の「新しい」文字を少なくともこの指定数含んでいるパスワードを指定する必要があります。このオプションはデフォルトで選択されています。

  次の値を使用して、未使用文字の固有性の程度を指定することができます。

  - [Number of passwords in history to be considered for character exclusion (0-10)]、および指定した文字数

    新しいパスワードに特定数の未使用文字を含めるように要求する場合、使用済み文字の検査時に考慮すべき過去のパスワードの数を指定できます。

    たとえば、少なくとも新しい 3 文字を含めるように指定し、以前の 5 つのパスワードを文字除外で考慮するよう指定した場合、ユーザが新しいパスワード「mountains99」を作成したとき、以前の 5 つのパスワードで未使用の文字が 3 文字以上そのパスワードに含まれる必要があります。ユーザの以前のパスワードが「maintains99」である場合、新しいパスワードと 2 文字しか異なっていないため、このパスワードはパスワードポリシーによって拒否され、ユーザは別のパスワードを指定するよう促されます。

注

- [Number of characters different from current password and passwords from history (0-6)]および[Number of passwords in history to be considered for character exclusion (0-10)]の両方のオプションがデフォルトで選択されます。ただし、両方のオプションの値はデフォルトで 0 に設定されています。
- [Number of characters different from current password and passwords from history (0-6)]オプションの値が 0 に設定される場合、このオプションは無効になります。
- [文字除外の対象とみなされる履歴内のパスワードの数 (0-10)]オプションの値が 0 に設定される場合、OpenText eDirectory で「新しい」文字が検査されるときに現在のパスワードだけが考慮されます。
- これらのオプションを使用するには、パスワードポリシーでユニバーサルパスワードを有効にする必要があります。

パスワード有効期間

- Number of days before password can be changed (0-365)

  このオプションは、指定した時間が経過するまでユーザがユニバーサルパスワードを変更できないように制限します。たとえば、この値が 30 に設定されている場合、ユーザは同じパスワードを 30 日間保持した後、それを変更できるようになります。

- Number of days before password expires (0-365)

このオプションは、指定した時間が経過した後、ユーザのパスワードを期限切れにします。たとえば、この値が 90 に設定されている場合、ユーザのパスワードが設定された時点から 90 日後に期限切れになります。猶予ログインが有効な場合、ユーザは指定された回数にわたって期限切れパスワードを使ってログインできます。また、［猶予ログイン制限］オプションを選択していない場合、無制限の猶予ログインが許可されます。

注

- ［**Use Microsoft Server 2008 Password Policy**］オプションを選択した場合、［**Number of days before password can be changed**］および［**Number of days before password expires**］オプションもまたデフォルトで選択されます。Microsoft Server 2008 構文が有効になっている場合、両方のオプションの範囲は 0 ～ 999 日です。
- 新しいユーザを作成する場合やヘルプデスクへの問い合わせに応答する場合などに、管理者がユーザのパスワードを変更する場合、パスワードポリシーでパスワードを期限切れにする設定が有効になっていると、パスワードは自動的に期限切れになります。この特定の機能については、日数は重要ではありませんが、この設定を有効にする必要があります。［**Do not expire the user's password when the administrator sets the password**］オプションを選択した場合、このセキュリティ拡張機能がオーバーライドされます。

- **Limit the number of grace logins allowed (0-254)**

  パスワードの期限が切れた場合、この値は、期限切れパスワードを使ってユーザが OpenText eDirectory にログインできる回数を示します。猶予ログインが有効でない場合、パスワードの有効期限が切れた後にユーザはログインできなくなり、管理者の支援でパスワードをリセットする必要が生じます。この値が 1 以上である場合、ユーザはパスワードの変更を強制される前に、指定の回数にわたってログインする機会を与えられます。ただし、すべての猶予ログインが使用されるまでにユーザがパスワードを変更しない場合、ユーザはロックアウトされて、OpenText eDirectory にログインできなくなります。また、［**Limit the number of grace logins allowed**］オプションを選択していない場合、無制限の猶予ログインが許可されます。

パスワード除外

- **Exclude the following passwords**

これにより、除外するパスワードを手動で指定できます。このオプションを使用すると、(パターンや OpenText eDirectory 属性ではなく) 特定の単語や単独の文字を除外することができます。また、*、+、%、空白文字などの特定の特殊文字を含むパスワードを除外することもできます。たとえば、パスワード除外リストに「*」文字を追加した場合、ユーザが「Pa55w0rd*!」というパスワードを指定しようとすると、指定したパスワードが無効であるというエラーを受け取ります。この機能は、環境内のアプリケーションで問題の原因となる特殊文字を含むパスワードを指定しないようユーザを制限する必要がある場合に、役立ちます。

NMAS 3.1.3 以降、除外リスト内の文字列をパスワードに含めることはできず、比較では大文字小文字が区別されません。たとえば、除外リストに「test」がある場合、Test、TEST、ltest, test1、latest をパスワードにすることはできません。

パスワード除外は、セキュリティ上のリスクになると思われる少数の単語に関してのみ役立つことに注意してください。備わっている除外リスト機能は、辞書などの長い用語リストとして使われるよう意図されていません。除外する用語を長いリストにすると、サーバのパフォーマンスに影響を与える可能性があります。パスワードの「辞書攻撃」から防御するには、長い除外リストを使用する代わりに、[高度なパスワードルール] を使用して、パスワードに数値を含めるようにすることをお勧めします。

- **Exclude passwords that match attribute values**

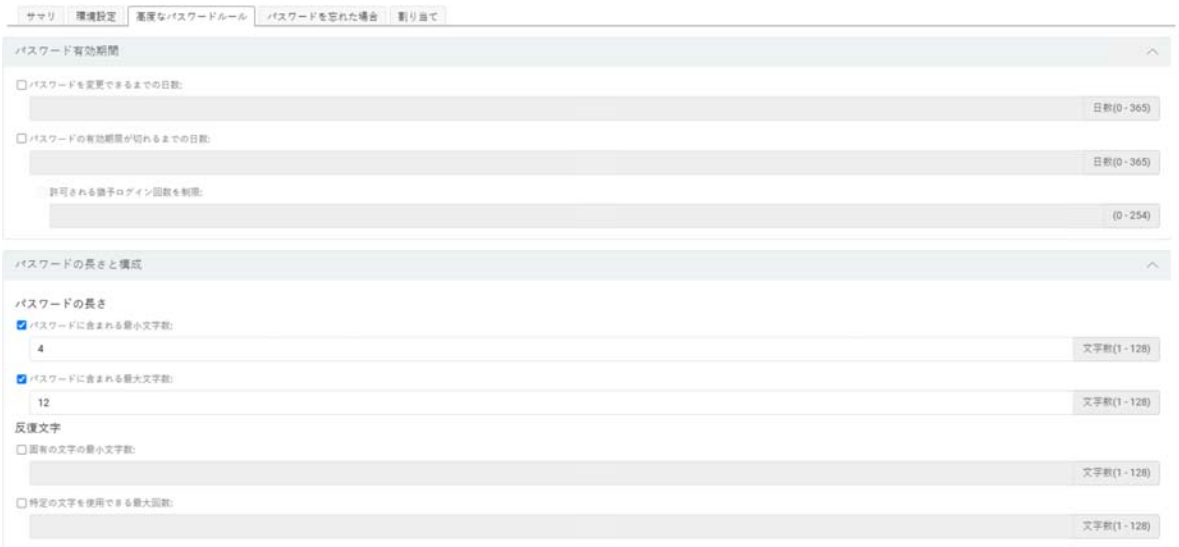
  これにより、パスワードとして使用されないよう除外するユーザオブジェクト属性を選択できます。たとえば、下の名 (Given Name) 属性をリストに追加し、この属性に Frank という値が含まれる場合、frank、frank1、1frank のどれもパスワードとして使用できません。

  除外するパスワードの長さが 3 文字未満の場合、NMAS はこのチェックを行いません。

  リストで属性値を追加したり削除したりするには、プラスボタンとマイナスボタンを使用します。

注 : [Use Microsoft complexity policy] オプションを選択した場合、デフォルトで [Exclude passwords that match attribute values] オプションもまた選択されます。Microsoft 複雑性ポリシー構文が有効になっている場合、照合する属性値のリストに次の値が入ります : 共通名、表示名、フルネーム、名 (ファーストネーム)、姓 (ラストネーム)。

***図 26-2** 高度なパスワードルール (続き)*



パスワード長

- **Minimum number of characters in password (1-512)**
- **Maximum number of characters in password (1-512)**

注

- NMAS を使用して作成されるパスワードの最大長は 512 文字です。

- ［Use Microsoft complexity policy］オプションを選択した場合、［Minimum number of characters in password］、［Maximum number of characters in password］のどちらのオプションも使用不可です。
- ［Use Microsoft Server 2008 Password Policy］オプションを選択した場合、［Minimum number of characters in password］オプションのみが使用可能です。デフォルトではこのオプションが選択されています。
- ［Use Novell syntax］オプションを選択した場合、［Minimum number of characters in password］および［Maximum number of characters in password］の両方のオプションもデフォルトで選択されます。

Password Complexity Violations

- Maximum number of complexity policy violations in password (0-5)

  このオプションを使用すると、管理者は、環境内のパスワードに対して許容する複雑性ポリシー違反の数を設定できます。デフォルトで、Microsoft Server 2008 パスワードポリシーでは、5 種類の文字 (大文字、小文字、数字、英数字以外の文字、その他の文字) のうち 3 種類から少なくとも 1 つの文字を含む必要があります。したがって、デフォルトで許容される違反の数は 2 です。Microsoft Server 2008 パスワードポリシーのポリシー要件の詳細については、803 ページの 「パスワードの構文」を参照してください。

  パスワードポリシーの制限を厳しくしたり緩めたりするには、許容されるデフォルトの違反数を変更できます。たとえばデフォルト設定を 1 に変更した場合、すべてのパスワードは、上記の 5 種類の文字タイプうち 4 種類からそれぞれ少なくとも 1 文字を含む必要があります。設定値が 4 の場合、パスワードは、5 種類の文字タイプのうち 1 種類のみの文字を含む必要があります。

注 : ［Use Microsoft Server 2008 Password Policy］オプションを選択した場合、［Maximum number of complexity policy violations in password (0-5)］オプションのみが使用可能になります。デフォルトではこのオプションが選択されています。

反復する文字

- Minimum number of unique characters (1-512)
- Maximum number of times a specific character can be used (1-512)
- Maximum number of times a specific character can be repeated sequentially (1-512)

注 : ［Use Microsoft complexity policy］または［Use Microsoft Server 2008 Password Policy］のいずれかのオプションを選択した場合、［Minimum number of unique characters］、［Maximum number of times a specific character can be used］、および［Maximum number of times a specific character can be repeated sequentially (1-512)］オプションは使用不可です。

大文字と小文字の区別あり

OpenText eDirectory で［**パスワードで大文字と小文字を区別する**］オプションを使用すると、OpenText eDirectory 9.2 にアップグレードされたすべてのクライアントでパスワードの大文字と小文字が区別されます。

注

- ［**Allow the password to be case sensitive**］オプションは、［**Use Novell syntax**］オプションを選択した場合にのみ使用可能になります。デフォルトではこのオプションが選択されています。
- ユニバーサルパスワードを無効にすることを選択した場合は、大文字と小文字を区別するオプションがデフォルトで選択されて無効になります。

［**Allow the password to be case sensitive**］オプションは、［**Use Novell syntax**］オプションを選択した場合にのみ使用可能になります。デフォルトではこのオプションが選択されています。

［**Allow the password to be case sensitive**］を選択した場合、次の 4 つのオプションがあります。

- **Allow the password to be case sensitive**
  - **Minimum number of upper case characters required in the password (1-512)**
  - **Maximum number of upper case characters allowed in the password (1-512)**
  - **Minimum number of lower case characters required in the password (1-512)**
  - **Maximum number of lower case characters allowed in the password (1-512)**

［**Allow the password to be case sensitive**］を選択しない場合、パスワードでは大文字小文字が区別されず、次の 2 つのオプションがあります。

- **Minimum number of alphabetic characters allowed in password (1-512)**
- **Maximum number of alphabetic characters allowed in password (1-512)**

重要 : ［**Allow passwords to be case sensitive**］オプションを選択しない場合でも、パスワードは大文字小文字を区別して保存され、複数のシステム間で同期される際に大文字小文字が区別されます。［**Allow passwords to be case sensitive**］オプションを選択しない場合、パスワード文字の大文字小文字は無視されます。

図 26-3 高度なパスワードルール (最後)



数字

- Allow numeric characters in password
  - Disallow numeric as first character
  - Disallow numeric as last character
  - Minimum number of numerals in password (1-512)
  - Maximum number of numerals in password (1-512)

注 : [Allow numeric characters in password] オプションは、[Use Novell syntax] オプションを選択した場合にのみ使用可能です。デフォルトではこのオプションが選択されています。

英数字以外の文字

英数字以外の文字とは、数字 (0 ～ 9) でも英字 (アルファベット文字) でもない文字です。アルファベット文字は、a ～ z、A ～ Z、および Latin-1 コードページ 850 のアルファベット文字と定義されます。

- Allow non-alphanumeric characters in the password
  - Disallow non-alphanumeric character as first character
  - Disallow non-alphanumeric character as last character
  - Minimum number of non-alphanumeric characters (1-512)
  - Maximum number of non-alphanumeric characters (1-512)

- **Allow non-US ASCII characters**

  このオプションを使用すると、基本ラテン文字セット以外の文字 (拡張文字とも呼ばれる) をパスワードに含めることができます。

注 : [**Allow non-alphanumeric characters in the password**] オプションは、[**Use Novell syntax**] オプションを選択した場合にのみ使用可能です。デフォルトではこのオプションが選択されています。

アルファベット以外の文字

アルファベット以外の文字とは、アルファベット文字 (英字) ではない文字です。アルファベット文字は、a ～ z、A ～ Z、および Latin-1 コードページ 850 のアルファベット文字と定義されます。

- **Allow non-alphabetic characters in the password**
  - **Minimum number of non-alphabetic characters (1-512)**
  - **Maximum number of non-alphabetic characters (1-512)**

注

- [**Allow non-alphabetic characters in the password**] オプションは、[**Use Novell syntax**] オプションを選択した場合にのみ使用可能です。
- [**アルファベット以外の文字をパスワードで使用することを許可**] オプションを使用する場合、ポリシーでパスワードを過度に厳しく制限しないようにしてください。たとえば、複数の非アルファベット文字や数字を要求すると同時に、使用できる非アルファベット文字数を*制限する*ようなポリシーです。

パスワードポリシーインタフェースの外部でパスワードポリシーを変更する

OpenText Identity Console の［認証管理］タイルを使ってパスワードポリシーを作成、変更、割り当てることに加えて、パスワードポリシーインタフェースの外部でポリシーを変更することもできます。次の方法があります。

- ディレクトリ管理インタフェースを使用してポリシーオブジェクトを直接変更する。
- ldapmodify コマンドラインツールを使用してポリシーオブジェクトを直接変更する。

ただし、パスワードポリシーインタフェースの外部でパスワードポリシーを操作することは推奨されません。そのような操作ですべての属性を正しく設定しない場合、環境で問題が発生する可能性があるためです。たとえば 1 つのポリシーに関して複数のポリシータイプを設定した場合、優先順位が「最も高い」ポリシータイプだけが有効になり、OpenText eDirectory は、適用される「より低い」ポリシータイプのポリシールールをすべて無視します。パスワードポリシータイプの優先度の詳細については、806 ページの 「パスワード構文の優先順位」を参照してください。

さらに、パスワードポリシーインタフェースを使用せずに Microsoft Server 2008 パスワードポリシータイプから Microsoft 複雑性ポリシータイプに変更した場合、OpenText Identity Console はポリシーオブジェクト内の既存の Microsoft Server 2008 パスワードポリシー属性 (nspmAD2K8Syntax) を削除しません。代わりに、OpenText Identity Console はこの属性の値を False に設定します。この状況では、OpenText eDirectory は両方のポリシータイプで設定されたすべてのポリシーとルールを無視します。

さらに、LDAP を使用してポリシーの特定のルールを変更するときに、別の問題が発生することがあります。ポリシーを変更した結果として 2 つのルールが競合するようになった場合、OpenText eDirectory は、ポリシー内の選択されていない (または False に設定されている) 競合ルールではなく、選択されている (または True に設定されている) ルールを適用します。

たとえば、あるポリシーを作成した後、数字を禁止してアルファベット以外の文字を許可するようそのポリシーを変更したとします。nspmNonAlphaCharactersAllowed 属性の値が True に設定されているため、nspmNumericCharactersAllowed が False に設定されていても、数字を含むアルファベット以外のすべての文字が許可されます。

パスワードのランダム生成

特定のパスワードを指定する代わりに、ユーザはランダムに生成されるパスワードを要求することもできます。ランダムに生成されるパスワードは、そのユーザに割り当てられている複雑さの要件、および他のパスワードポリシー制限に自動的に準拠します。

ランダムに生成される Microsoft Server 2008 パスワード

Microsoft Server 2008 パスワードポリシー用のランダム生成パスワードは、他のパスワードポリシータイプを使ってランダム生成されるパスワードと比べて、次の点で異なります。

- Microsoft Server 2008 パスワードポリシータイプを使用するパスワードポリシーを割り当てられているユーザがランダム生成パスワードを要求する場合、NMAS は、ポリシーで許容されるパスワード複雑性違反数に基づいてパスワードを生成します。
- 許容されるパスワード複雑性違反数が最大値 (5) に設定されている場合、ランダム生成されるすべてのパスワードは、大文字または小文字のアルファベット文字だけで構成されます。
- パスワード複雑性の要件が極端に厳しく設定されている場合、ランダム生成されたパスワードであっても、パスワードポリシーに対して無効である可能性があります。
- Microsoft Server 2008 パスワードポリシーの下でランダム生成されるパスワードの最大長は 16 文字です (ただしポリシーで設定された最小長が 16 文字を超える場合を除く)。最小長が 16 を超える場合、生成されるパスワードの長さは、ポリシーで設定された最小長です。たとえば、Microsoft Server 2008 ポリシーを使用してパスワードの最小長が 20 文字と設定されている場合、ランダム生成されるパスワードの長さは常に 20 文字になります。

ユニバーサルパスワードの設定オプション

次の図は、ユニバーサルパスワードの設定オプションの例を示しています。

図 26-4 環境設定オプション



- **Enable Universal Password**

  このポリシーでユニバーサルパスワードを有効にします。ユニバーサルパスワードを有効にするか無効にするかを選択できます。

- **Enable the Advanced Password Rules**

  このポリシーの［高度なパスワードルール］ページにある［高度なパスワードルール］を有効にします。これらの高度なパスワードルールでは、パスワード有効期間とパスワードに含む文字を制御できるため、環境のセキュリティ向上に役立ちます。

- **パスワード同期**

  - **Remove the NDS password when setting Universal Password**

    このオプションを選択すると、ユニバーサルパスワードが設定されるときに NDS パスワードが無効になります。また、NDS パスワードが設定されるときに、OpenText eDirectory 以外には認識できないランダムな値に NDS パスワードハッシュが設定されます。ランダム値にハッシュ化できるパスワードが存在することも、存在しないこともあります。

  - **Synchronize NDS password when setting Universal Password**

    このオプションを選択した場合、ユニバーサルパスワードが設定されるときに、同じパスワードを使って同時に NDS パスワードも設定されます。

  - **Synchronize Simple Password when setting Universal Password**

    注 : このオプションを設定しても、ICE を使用してユーザパスワードをインポートする機能には影響がありません。

このオプションを選択した場合、ユニバーサルパスワードが設定されるときに、同じパスワードを使って同時に単純パスワードも設定されます。

- **Synchronize Distribution Password when setting Universal Password**

  Identity Manager メタディレクトリエンジンが OpenText eDirectory 内のユーザのユニバーサルパスワードを取得または設定できるかどうかを決定します。

  このオプションを選択した場合、ユニバーサルパスワードが設定されるときに、同じパスワードを使って同時に配布パスワードも設定されます。

  接続システムに対するパスワード同期を実行するために、配布パスワードを Identity Manager とともに使用できます。また、このオプションにより、メタディレクトリエンジンは OpenText eDirectory 内のユーザのユニバーサルパスワードを取得することもできます。

- **Universal Password Retrieval**

注 : ユニバーサルパスワードを無効にすることを選択した場合は、次のオプションはデフォルトで無効になります。

- **Allow user to retrieve password**

  パスワードを忘れた場合のセルフサービス機能がユーザに代わってパスワードを取得し、それを電子メールでユーザに送信できるかどうかを決定します。このオプションを選択しない場合、パスワードポリシーの［パスワードを忘れた場合］ページの対応する機能がぼかし表示になります。

  このオプションにより、ユーザは NMAS LDAP 拡張機能を使用して自分のパスワードを取得することができます。

- **Allow admin to retrieve passwords**

  この機能を使用するサードパーティ製品またはサービスを使ってユーザのパスワードを取得することができます。

  このオプションはお勧めできません。稼働するためにこの機能を必要とする特定のオブジェクト (SAMBA または freeRADIUS サービスオブジェクトなど) にパスワード読み取り権限を割り当てるには、［**Allow the following to retrieve passwords**］オプションを代わりに使用してください。

  ［**Allow admin to retrieve passwords**］が選択されている場合、ターゲットオブジェクトの ACL 属性に対する書き込み特権を持つユーザはターゲットオブジェクトのパスワードを取り出すことができます。

- **Allow the following to retrieve passwords**

  パスワードを取得できるオブジェクトを挿入することができます。

  注 : 十分な特権を持たないメンバーが、特定のユーザに関して［**パスワード状態のチェック**］タスクを使用すると -672 エラーを受け取ります。

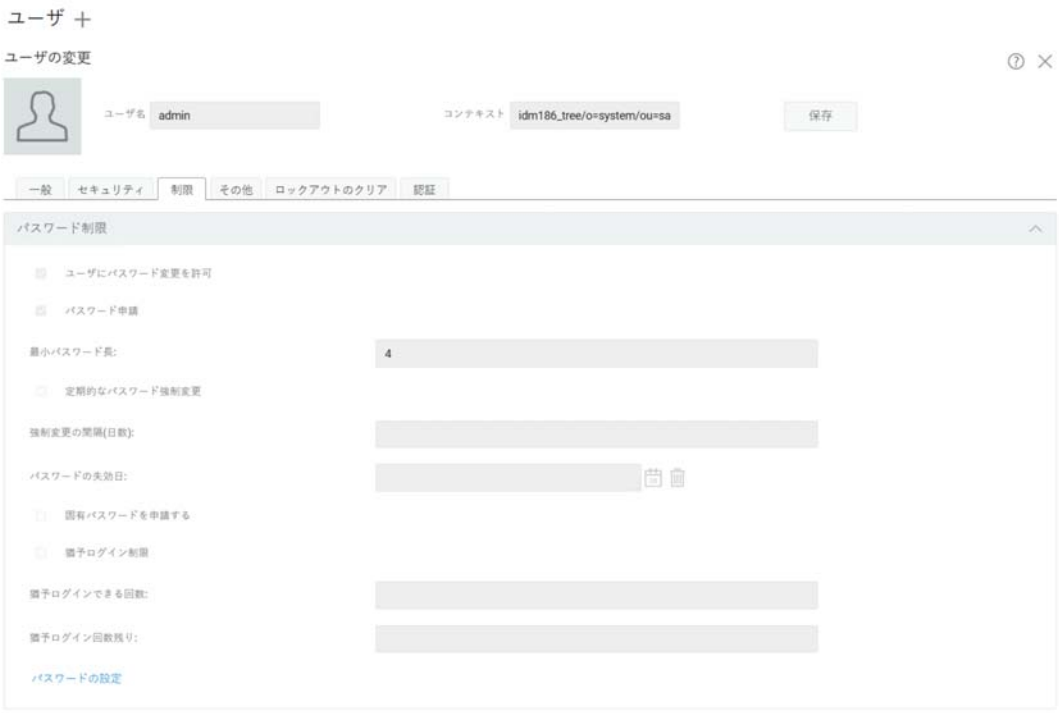
- **認証**

  - **Verify whether existing passwords comply with the password policy (verification occurs on login)**

このオプションを選択した場合、ユーザが OpenText Identity Console からログインすると、ユーザの既存のパスワードが検査されて、ユーザのパスワードポリシーにある高度なパスワードルールに準拠しているかどうかが確認されます。既存のパスワードが準拠していない場合、ユーザはそれを変更するよう要求されます。ユニバーサルパスワードを無効にすると、このオプションもデフォルトで無効になります。

管理者は、図で示すように、[ユーザ] > [制限] > [パスワード制限] でユーザの設定を変更できます。

図 26-5



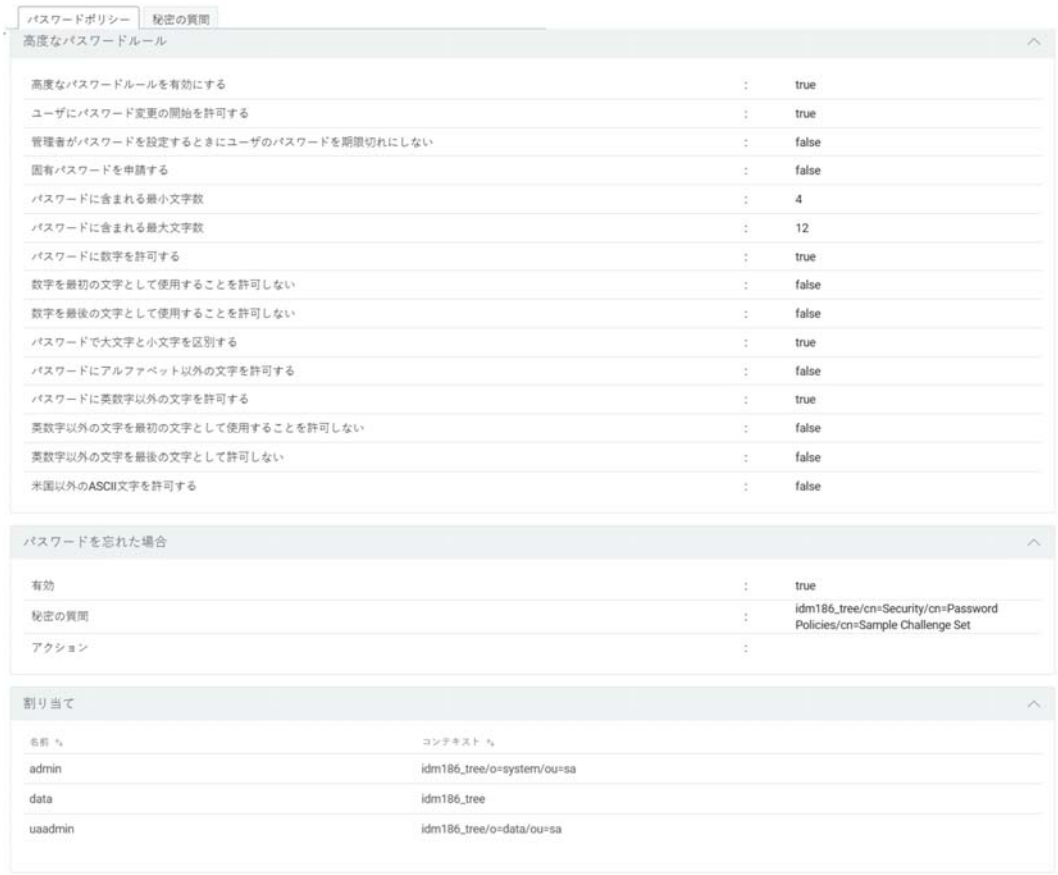
OpenText eDirectory 9.2.7 以降でもこれらの設定を変更できます。ただし、ユーザがログインするとすぐに、これらの設定はパスワードポリシーによって上書きされます。これは、パスワードの有効期限だけでなく、passwordExpirationInterval ([定期的なパスワード強制変更])、[ユーザにパスワード変更を許可]、[固有パスワードを申請する] にも影響します。その結果、ポリシーを割り当てることでユーザレベルで行われたすべての設定が、ポリシーによって上書きされることになります。

[ユニバーサルパスワード] を無効にすると、このオプションもデフォルトで無効になります。ただし、[パスワードポリシー] ウィンドウで [ユニバーサルポリシー] を手動で有効にできます。

図26-6 ポリシーの概要



図 26-7 ポリシーの概要 (続き)



注 : 管理者以外のユーザが自分のパスワードを変更できるのは、管理者が［**ユーザ**］>［**制限**］>［**パスワード制限**］ウィンドウで［**ユーザにパスワード変更を許可**］および［**パスワード申請**］チェックボックスを有効にしている場合だけです。ただし、(パスワードポリシーで割り当てられた) ユーザオブジェクトに対して行った設定は、ポリシールールによって上書きされます。

ユーザへのパスワードポリシーの割り当て

OpenText eDirectory 内のユーザにパスワードポリシーを割り当てる方法として、(ログインポリシーオブジェクトを使用して) ツリー全体にポリシーを割り当てたり、特定のパーティションやコンテナ、または特定のユーザに割り当てたりすることができます。管理を単純化するために、ツリー内のできるだけ高いレベルでパスワードポリシーを設定することをお勧めします。

重要 : OpenText eDirectory ツリー全体、またはサブコンテナに非常に多くの (何万もの) ユーザを含むツリーのコンテナにパスワードポリシーを割り当てた場合、OpenText Identity Console がハングすることがあります。

この場合、パスワードポリシー割り当てごとのユーザ数を制御するために、下位レベルのコンテナに個別にパスワードポリシーを割り当てることを考慮できます。

ポリシーは、1 つ以上のオブジェクトに割り当てるまでは有効になりません。パスワードポリシーは次のオブジェクトに割り当てることができます。

- ログインポリシーオブジェクト

  ツリー内のすべてのユーザに対し、デフォルトのパスワードポリシーを作成することをお勧めします。これを実行するには、ポリシーを作成して、ログインポリシーオブジェクトに割り当てます。ログインポリシーオブジェクトは、ツリーのルート直下のセキュリティコンテナ内にあります。

- パーティションのルートであるコンテナ

  コンテナ パーティションのルートになっているコンテナにポリシーを割り当てると、ポリシー割り当てはサブコンテナのユーザを含めたパーティション内のすべてのユーザに継承されます。コンテナがパーティションルートであるかどうかを知るには、そのコンテナを参照して、隣にパーティションアイコンが表示されているかどうかを確認します。

- パーティションのルートではないコンテナ

  パーティションのルートではないコンテナにポリシーを割り当てると、そのポリシー割り当ては、その特定のコンテナ内のユーザだけに継承されます。サブコンテナ内のユーザには継承されません。パーティションのルートではないコンテナの下位にあるユーザすべてにポリシーを適用する場合は、それぞれのサブコンテナに個別のポリシーを割り当てる必要があります。

- 特定のユーザ

1 人のユーザに対して一度に 1 つのポリシーのみを有効にできます。NMAS は、次の順序でポリシーを検索し、最初に見つかったものを適用することで、ユーザに対して有効にするポリシーを決定します。

1. **ユーザごとの割り当て** : パスワードポリシーがユーザに割り当てられている場合、そのポリシーが適用されます。
2. **コンテナ** : パスワードポリシーがユーザに割り当てられていない場合、NMAS はそのユーザが存在するコンテナに割り当てられたポリシーを適用します。
3. **パーティションルートのコンテナ** : パスワードポリシーが、ユーザやそのユーザの上位コンテナにも割り当てられていない場合、パーティションルートのコンテナに割り当てられたポリシーが適用されます。
4. **ログインポリシーオブジェクト** : ポリシーが、ユーザやその他のコンテナにも割り当てられていない場合、ログインポリシーオブジェクトに割り当てられたポリシーが適用されます。これはツリー内のすべてのユーザに対するデフォルトポリシーです。

次の図に示すプロパティページの例では、どのオブジェクトパスワードポリシーを割り当てるかを指定します。

図 26-8 パスワードポリシーの割り当て



ユーザに適用されているポリシーの識別

1 人のユーザに対して一度に 1 つのポリシーのみが有効になります。特定のユーザまたはコンテナに対してどのポリシーが有効になっているか検出するには、次の手順を実行します。

1 OpenText Identity Console のホームページで、[**認証管理**] タイル > [**パスワードポリシー**] の順にクリックします。
2 パスワードポリシーをクリックし、[**割り当て**] タブに移動します。

ツリー内に複数のポリシーが存在する場合、820 ページの 「ユーザへのパスワードポリシーの割り当て」で説明されている方法で、NMAS はユーザに割り当てるポリシーを決定します。

ユーザのパスワードの設定

管理者またはヘルプデスク担当者は、OpenText Identity Console のタスクを使用してユーザのユニバーサルパスワードを設定できます。このタスクでは、ユーザに適用されているパスワードポリシーのパスワードルールが表示されます。

1 OpenText Identity Console のホームページで、[**認証管理**] > [**パスワードポリシー**] の順に選択します。
2 [**パスワードポリシーの作成** ＋] をクリックし、[**ユニバーサルパスワードを有効にする**] をオンにしてパスワードを作成します。
3 OpenText Identity Console のホームページで、[**ユーザ管理**] タイルをクリックします。
4 該当するユーザをブラウズして選択します。

ユーザにパスワードポリシーがすでに割り当てられ、ユニバーサルパスワードが有効になっている場合は、パスワードを変更できます。

［高度なパスワードルール］がポリシーで有効になっている場合、従うべきルールのリストが［**パスワードの設定**］に表示されます。

5 表示されるすべてのパスワードルールに準拠していることを確認しながら、ユーザのパスワードを作成します。

6 ［**保存**］をクリックします。

ユーザのユニバーサルパスワードが変更されます。

パスワード同期が環境で設定されている場合、ユーザの新しいパスワードは、それを受け入れるよう設定されている接続先システムに配布されます。

注 : 新しいユーザを作成する場合やヘルプデスクへの問い合わせに応答する場合などに、管理者がユーザのパスワードを変更する場合、パスワードポリシーでパスワードを期限切れにする設定が有効になっていると、パスワードは自動的に期限切れになります。［**Number of days before password expires**］という設定が高度なパスワードルールに含まれています。この特定の機能については、日数は重要ではありませんが、この設定を有効にする必要があります。

［**Do not expire the user's password when the administrator sets the password**］オプションは、この機能をオーバーライドします。

ユニバーサルパスワードの診断ユーティリティ

OpenText eDirectory には、状態をチェックして、ユニバーサルパスワードを再暗号化するためのユーティリティが用意されています。ユニバーサルパスワードの診断ユーティリティ (diagpwd) は、管理者がユーザのユニバーサルパスワード (UP)、簡易パスワード、NDS パスワード、および配布パスワード (DP) の状態を確認できるようにするツールです。これらのパスワードの同期状態もレポートします。diagpwd ユーティリティの構文の例を以下に示します。

```
diagpwd LDAP_SERVER_ADDR TLS_PORT CA_CERT_FILE SEARCH_BASE SEARCH_SCOPE
BIND_DN [BIND_PWD] -t
```

オプション	説明
LDAP_SERVER_ADDR	ターゲット LDAP サーバのアドレスを指定します。
TLS_PORT	ターゲット LDAP サーバの LDAP セキュアポート (TLS) を指定します。
CA_CERT_FILE	ターゲット LDAP サーバのルート認証局証明書を含む、PEM エンコードされたファイルのパスを指定します。
SEARCH_BASE	searchbase を検索の開始ポイントとして使用します。

オプション	説明
SEARCH_SCOPE	検索のスコープを指定します。スコープとして、ベースオブジェクトを示す「base」、1 レベルを示す「one」、またはサブツリー検索を示す「sub」を指定します。
BIND_DN	管理者の LDAP DN です。たとえば、cn=admin,o=company と指定します。
BIND_PWD	管理者の LDAP パスワード。 **注 :** このパラメータはオプションです。コマンドラインで指定されていない場合は、ユーザに対してプロンプトが表示されます。
-t	これにより、UP、DP、簡易パスワード、およびパスワード履歴が 256 ビット AES キーを使用して再暗号化されます。AES 256 ビットのツリーキーを作成した後に、このオプションを使用します。 **注 :** このオプションを使用している場合は、パスワードポリシーを使用して、このユーティリティを実行しているユーザが自分の UP を取得できるようにしてください。

注 : diagpwd ユーティリティは、OpenText eDirectory 9.1 SP2 以上でサポートされています。

diagpwd ユーティリティのインストール

diagpwd ユーティリティをインストールするには、OpenText eDirectory のセットアップフォルダで次のコマンドを実行します。

```
rpm -i /eDirectory-setup-location/novell-nmas-ldap-ext-client-9.2.0-0.x86_64.rpm
```

例

サーバ 192.168.1.1 上のユーザ cn=user1,ou=users,o=company のパスワードの状態を確認するには、次のコマンドを実行します。

```
diagpwd 192.168.1.1 636 /home/user1/cert.pem cn=user1,ou=users,o=company base cn=admin,o=company
```

ou=users,o=company サブツリー内のすべてのユーザのパスワードの状態を確認するには、次のコマンドを実行します。

```
diagpwd 192.168.1.1 636 /home/user1/cert.pem ou=users,o=company sub cn=admin,o=company
```

ou=users,o=company サブツリー内のすべてのユーザのパスワードを AES 256 ビットキーで再暗号化するには、次のコマンドを実行します。

```
diagpwd 192.168.1.1 636 /home/user1/cert.pem ou=users,o=company sub
cn=admin,o=company -t
```

パスワードポリシーのトラブルシューティング

- 825 ページの 「ユーザにパスワードポリシーが割り当てられていないことを示すエラー」
- 825 ページの 「秘密の質問の答えを使用する」
- 825 ページの 「新しいコンテナ内のユーザへのアクセス権限の付与」
- 826 ページの 「NMAS LDAP 転送エラー」

ユーザにパスワードポリシーが割り当てられていないことを示すエラー

ユニバーサルパスワードの設定タスクで、ユーザにパスワードポリシーが割り当てられていないというエラーが表示された場合、実際にはユーザにパスワードポリシーが割り当てられているのであれば、SSL の問題である可能性があります。SSL の問題を診断して解決するには、次のタスクを実行します。

- SSL の環境設定に問題があることを確かめるには、ポリシー割り当て表示タスクを使用して、そのユーザのポリシーを確認します。ポリシー割り当て表示タスクに NMAS 転送エラーが表示される場合、これは SSL が正しく設定されていないことを示している可能性があります。
- OpenText Identity Console を実行する Web サーバとプライマリ OpenText eDirectory ツリーの間で SSL が正しく設定されていることを確認してください。Web サーバと OpenText eDirectory の間で証明書が設定されていることを確認します。
- 単純認証用に TLS を必要としない場合、802 ページのステップ 3 の注記で説明されているように、正しい LDAP SSL ポートを確実に指定する必要があります。

秘密の質問の答えを使用する

OpenText Identity Console でサポートされるブラウザを使用していることを確認してください。

新しいコンテナ内のユーザへのアクセス権限の付与

OpenText Identity Console または OpenText のポータル製品のいずれか (ユーザアプリケーションなど) をセットアップするときには、ポータルユーザコンテナを指定します。通常は、ツリーのすべてのユーザがポータル機能にアクセスできるように、ツリー内の高いレベルでコンテナを指定します。すべてのユーザがそのコンテナの下にある場合は、すべてのユーザが「パスワードを忘れた場合」および「パスワードのリセット」セルフサービス機能にアクセスできます。

NMAS LDAP 転送エラー

マルチサーバ環境で Identity Manager をインストールしている場合、OpenText Identity Console で一部のパスワード管理機能を使用すると、「NMAS LDAP 転送エラー」で始まるエラーが表示されることがあります。

このエラーの一般的な原因の 1 つは、Identity Manager で必要な NMAS 拡張機能を持っていない LDAP サーバを PortalServlet.properties ファイルが参照していることです。PortalServlet.properties ファイルを開き、LDAP サーバのアドレスが Identity Manager のインストール場所と同じサーバであることを確認してください。

この他の考えられる原因 :

- LDAP サーバが稼働していない。
- OpenText Identity Console サーバと LDAP サーバの間で LDAP 用に SSL が設定されていない。
- リモート Identity Manager サーバを管理するために OpenText Identity Console で他のツリーにログインするとき、リモートサーバの IP アドレスの代わりにサーバ名を使用するとエラーが発生することがあります。
- 認証先となるツリーのルート認証局証明書を、信頼された証明書として Web サーバにインポートする必要があります。keytool.exe を使用して、証明書を Web サーバにエクスポートできます。

セキュリティ上の考慮事項

他のパスワードシステムとの相互運用を容易にするには、復号可能なユニバーサルパスワード暗号化が必要です。管理者は、システムのコストと利点を評価する必要があります。OpenText eDirectory に保管されたユニバーサルパスワードを使用すると、複数のパスワードを管理しようとする場合よりも、セキュリティが強化されたり、より便利になったりする可能性があります。

OpenText eDirectory 内のユニバーサルパスワードは、パスワード自体のトリプル DES 暗号化、OpenText eDirectory 権限、およびファイルシステム権限の 3 つのセキュリティレベルで保護されます。

- NICI 3.0 より前では、ユニバーサルパスワードはトリプル DES、ユーザ固有の鍵で暗号化されました。ユニバーサルパスワードとユーザ鍵は、どちらも OpenText eDirectory のみが読み込めるシステム属性に格納されました。ユーザ鍵 (3DES) はツリー鍵で暗号化されて格納され、ツリー鍵は各マシンで固有の NICI 鍵によって保護されていました。ツリー鍵も NICI 鍵も OpenText eDirectory 内には保管されなかったことに注意してください。これらは、保護対象のデータと共に保管されませんでした。ツリー鍵はツリー内の各マシンに存在していましたが、ツリーごとにツリー鍵が異なっていたため、ツリー鍵で暗号化されたデータは同じツリー内のマシンでのみ復号可能でした。こうして、ユニバーサルパスワードが保管されるときには 3 つのレベルの暗号化で保護されていました。

NICI 3.0 は AES 256 ビットストレージ鍵をサポートしています。したがって、他の鍵を安全にラップするためにストレージ鍵を使用するアプリケーションは、新しいアルゴリズムを処理できる必要があります。しかし、古い 3-DES 鍵を使って現在ラップされているデータは、変更なしで引き続きアクセス可能になります。

NICI 3.0 は AES 256 ビットツリー鍵をサポートします。しかし、OpenText eDirectory はデフォルトで AES 256 ビットツリー鍵を作成しません。バージョン 9.0 以前の環境でこの鍵を作成すると、ツリー鍵に依存するサービスで問題が発生する可能性があります。鍵を作成する前に、すべての OpenText eDirectory サーバを 9.2 に更新することをお勧めします。詳細については、「Creating an AES 256-Bit Tree Key (AES 256 ビットツリーキーを作成する)」を参照してください。

- それぞれの鍵は、OpenText eDirectory の権利によっても保護されます。ユニバーサルパスワードを変更する権利を持つのは、スーパバイザ権を持つ管理者、またはその鍵を所有するユーザだけです。

注 : NMAS/nds-cluster-config 拡張機能を使用して、ユーザが自分のパスワードを読み取ったり管理者がユニバーサルパスワードを読み取ったりできるように、パスワードポリシーを設定することができます。デフォルトではこれが有効になっていません。

- ファイルシステム権限は、適切な権限を持つユーザのみが鍵にアクセスできるようにします。

  高度なセキュリティを必要とする環境でユニバーサルパスワードを導入する場合は、次のような追加の対策を実行できます。

  - 次のディレクトリとファイルがセキュリティで保護されていることを確認します。

Windows	%SystemRoot%\SysWOW64\Novell\nici
	%SystemRoot%\System32\ (NICI DLL のインストール先)
Linux	/var/opt/novell/nici
	/etc/opt/novell/nici64.cfg
	/opt/novell/lib64/libccs2.so、および同じディレクトリ内の NICI 共有ライブラリ

    NICI および OpenText eDirectory のファイルの具体的な場所について詳しくは、ご使用のシステムのマニュアルを参照してください。

  - 他のセキュリティシステムの場合と同様に、鍵が保管されているサーバへの物理的アクセスを制限することは非常に重要です。

パスワード管理に関連するセキュリティ上の考慮事項については、697 ページの 「セキュリティ上の考慮事項」を参照してください。

eDirectory へのハッシュベースのパスワードのインポート

パスワードは、DIGEST-MD5、crypt、SHA、SSHA ハッシュの LDIF によって eDirectory にインポートできます。MD5 ハッシュベースのパスワードを OpenText eDirectory にインポートするには、次の手順を実行します。

1 次のコマンドを使用して、MD5 ハッシュを base64 形式で作成します。

```
echo -n <password> | openssl md5 -binary | base64
```

注 : eDirectory がサポートするハッシュベースのパスワードは base64 形式のパスワードのみです。

2 次の例に示されているように、MD5 ハッシュの作成時に返されるテキストを LDIF ファイルに追加します。

```
dn: cn=sp1,o=novell
control: 2.16.840.1.113719.1.27.101.5
changetype: modify
replace: userPassword
userPassword: {md5}CSbJUP4kfDtGXrE+JY7kaNI5oGU=
```

注 : LDIF ファイルで変更される、ユーザに適用されているパスワードポリシーがないことを確認します。

3 単純パスワードメソッドと DIGEST MD5 NMAS メソッドの両方をインストールし、単純パスワードをデフォルトのメソッドにします。

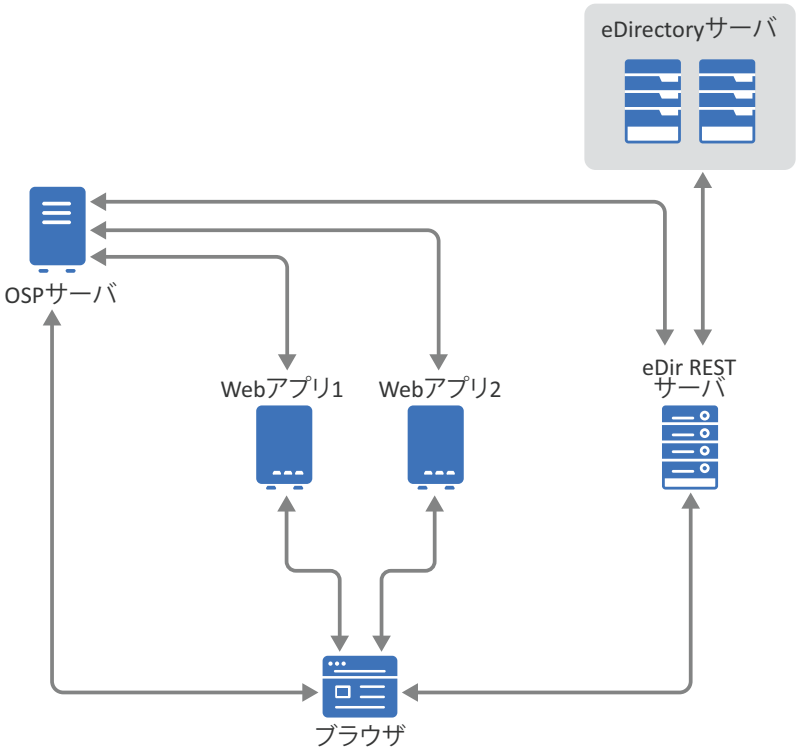
4 次のコマンドを使用して、-l オプション (LDAP ターゲットハンドラを表す) を指定して ice を使用します。

```
ice -S LDIF -f ./change_pass.ldiff -D LDAP -s 164.99.163.236 -p 636 -d
cn=admin,o=novell -w n -l -L /var/opt/novell/eDirectory/data/SSCert.der
```

27 REST サービス

OpenText eDirectory には、REST Web サービスを使用して OpenText eDirectory のさまざまな機能を利用するための REST API がいくつか組み込まれています。API を使用すると、任意の要求を呼び出して OpenText eDirectory サーバからそれぞれの応答を得ることができます。REST Web サービスは、Docker コンテナとしてデプロイできます。詳細については次の図を参照してください。これらの図は、OpenText eDirectory での REST Web サービスのアーキテクチャを示しています。

図 27-1 *OpenText eDirectory での REST Web サービスのアーキテクチャ*



注 : OpenText eDirectory の REST サービスは、OAUTH2 プロトコルを使用して、One SSO Provider (OSP) による認証を提供します。

このドキュメントでは、次のタスクを実行する方法について説明します。

- 830 ページの 「セキュリティの推奨事項」
- 830 ページの 「OpenText eDirectory 向けの REST サービスのインストールを計画する」
- 834 ページの 「OpenText eDirectory 用の REST サービスの設定」
- 836 ページの 「データ永続性の管理」
- 837 ページの 「REST サービスによる監査」

- 838 ページの 「REST コンテナを使用して LDAP パスワードを変更する」
- 838 ページの 「REST コンテナを使用してサーバ証明書を変更する」
- 839 ページの 「OpenText eDirectory 用の REST サービスのアップグレード」
- 841 ページの 「Rest API マニュアル」

セキュリティの推奨事項

- Docker コンテナには、デフォルトではリソースの制約がありません。そのため、すべてのコンテナには、ホストのカーネルによって提供されるすべての CPU リソースおよびメモリリソースへのアクセスが提供されます。コンテナで使用できるリソースの量に制限を設定することにより、実行中の 1 つのコンテナによってより多くのリソースが消費され、実行中の他のコンテナがリソース不足になることがないようにする必要があります。
  - Docker run コマンドの --memory フラグを使用して、Docker コンテナで、コンテナによって使用されるメモリに対してハード制限が適用されるようにする必要があります。
  - Docker run コマンドの --cpuset-cpus フラグを使用して、Docker コンテナで、実行中のコンテナによって使用される CPU の容量に制限が適用されるようにする必要があります。
- --pids-limit には 300 を設定して、任意の時点でコンテナ内で生成されるカーネルスレッドの数を制限する必要があります。これは、DoS 攻撃を防ぐためです。
- Docker run コマンドの --restart フラグを使用して、障害発生時のコンテナ再起動ポリシーを 5 に設定する必要があります。
- REST コンテナは、必ず、コンテナの起動後にヘルスステータスが［良好］と表示されてから使用する必要があります。コンテナのヘルスステータスを確認するには、次のコマンドを実行します。

  ```
  docker ps <container_name/ID>
  ```

- REST コンテナは、常に root 以外のユーザ (nds) として起動されます。追加のセキュリティ対策として、デーモンでのユーザ名前空間の再マッピングを有効にして、コンテナ内からの特権昇格攻撃を防止します。ユーザ名前空間の再マッピングの詳細については、「ユーザ名前空間でコンテナを分離する」を参照してください。

OpenText eDirectory 向けの REST サービスのインストールを計画する

このセクションでは、REST サービスをインストールする前のセットアップの準備について説明します。REST をインストールおよび設定するには、次のタスクを実行する必要があります。

- pkcs12 サーバ証明書を取得します。外部認証局または OpenText Identity Console によって生成されたサーバ証明書を使用できます。詳細については、719 ページの 「サーバ証明書オブジェクトを作成する」を参照してください。

- 認証局証明書ファイルを .pem 形式で取得してください。たとえば、OpenText eDirectory 認証局証明書 (SSCert.pem) を使用できます。
- (省略可能) REST サービスをインストールする前に、OSP をインストールして設定します。詳細については、「OSP コンテナの展開」を参照してください。
- 次の環境設定パラメータを使用して、環境設定ファイルを作成します。たとえば、edirapi.conf ファイルを作成します。環境設定ファイルの値は、要件に応じて変更できます。

  注 : 2 つのサンプル環境設定ファイルが tarball イメージにバンドルされます。ビジネス要件に応じてこれらのファイルに変更を加えることもできます。

  OSP ありで REST を構成する場合は、次に示すように環境設定ファイルを作成します。

```
listen = ":9000"
ldapserver = "192.168.1.1:636"
ldapuser = "cn=admin,ou=sa,o=system"
ldappassword = "novell"
pfxpassword = "novell"
ospmode = "true"
osp-token-endpoint = "https://10.10.10.10:8543/osp/a/idm/auth/oauth2/
getattributes"
osp-authorize-url = "https://10.10.10.10:8543/osp/a/idm/auth/oauth2/
grant"
osp-logout-url = "https://10.10.10.10:8543/osp/a/idm/auth/app/logout"
osp-redirect-url = "https://10.10.10.10:9000/eDirAPI/v1/edirtree/
authcoderedirect"
osp-client-id = "identityconsole"
ospclientpass = "novell"
ospcert = "/etc/opt/novell/eDirAPI/cert/SSCert.pem"
bcert = "/etc/opt/novell/eDirAPI/cert/"
loglevel = "error"
check-origin = "true"
origin = "https://10.10.10.10:9000,https://192.168.1.1:8543"
```

  OSP なしで REST を構成する場合は、OSP パラメータを指定せずに、次に示すように環境設定ファイルを作成します。

```
listen = ":9000"
pfxpassword = "novell"
ospmode = "false"
bcert = "/etc/opt/novell/eDirAPI/cert/"
edir-hosts = "<ip_address-1>:636,<ip_address-2>:636"
```

***表 27-1** 環境設定ファイル内の環境設定パラメータの説明*

環境設定パラメータ	説明
listen	コンテナ内の REST サーバの待ち受けポートとして 9000 を指定します。
ldapserver	OpenText eDirectory ホストサーバの IP を指定します

環境設定パラメータ	説明
ldapuser	eDirectory ツリーの管理者権限を持つユーザのユーザ名を指定します . **重要 :** ユーザ名は小文字にする必要があります。ユーザ名が大文字の場合、eDirAPI Docker コンテナの初期化に失敗する場合があります。
ldappassword	LDAP サーバのパスワードを指定します
pfxpassword	.pfx 証明書ファイルのパスワードを指定します
ospmode	OSP を OpenText Identity Console と統合するには true を指定します。false に設定すると、OpenText Identity Console で LDAP ログインが使用されます
osp-token-endpoint	この URL は、OSP サーバから特定の属性をフェッチして、認証トークンの有効性を検証するために使用します
osp-authorize-url	この URL は、ユーザが認証トークンを取得するために資格情報を提供する際に使用します
osp-logout-url	この URL を使用して、ユーザと OSP サーバ間のセッションを終了します
osp-redirect-url	OSP サーバは、認証トークンを付与した後、ユーザをこの URL にリダイレクトします
osp-client-id	REST を OSP に登録したときに提供された OSP クライアント ID を指定します
ospclientpass	REST を OSP に登録したときに提供された OSP クライアントパスワードを指定します
ospcert	OSP サーバの認証局証明書の場所を指定します
bcert	OpenText Identity Console の認証局証明書の場所を指定します
loglevel	ログファイルに含めるログレベルを指定します。このパラメータは、「fatal」、「error」、「warn」、または「info」に設定できます。
check-origin	これを true に設定すると、REST サーバで要求のオリジン値が比較されます。使用可能なオプションは、true または false のいずれかです。DNS 設定を使用する場合、*check-origin* パラメータ値が false に設定されている場合でも、*origin* パラメータは必須です。

環境設定パラメータ	説明
origin	eDirAPI は、要求のオリジン値をこのフィールドで指定された値と比較します。 **注 :** eDirAPI 1.4 以降では、このパラメータは check-origin パラメータとは独立しています。また、DNS 設定を使用する場合は必須です。
enableaudit	REST サービスの監査を有効にするには、このオプションを true に設定します。使用可能なオプションは、true または false のいずれかです。
enableservicestartaudit	このオプションを true に設定すると、REST サービスの開始イベントが通知されます。使用可能なオプションは、true または false のいずれかです。
enableservicestopaudit	このオプションを true に設定すると、REST サービスの停止イベントが通知されます。使用可能なオプションは、true または false のいずれかです。
enablelogsessioncreationaudit	このオプションを true に設定すると、REST サービスのセッション作成イベントが通知されます。使用可能なオプションは、true または false のいずれかです。
enablelogsessionterminationaudit	このオプションを true に設定すると、REST サービスのセッション終了イベントが通知されます。使用可能なオプションは、true または false のいずれかです。
auditlogmaxsize	それぞれの REST サービスの監査ログファイルサイズの最大限度を指定します。デフォルトでは、ファイルサイズは 50MB です。
edirapilogmaxsize	それぞれの REST サーバのログファイルサイズの最大限度を指定します。
scope	REST サーバが OAuth 用語で言うところのリソースサーバとして使用されている場合は、そのスコープを指定します。デフォルトでは、edirapi *<tree_name>* に設定されています。
maxclients	edirapi にアクセスできる同時クライアントの最大数。この制限を超えるクライアントはキュー内で待機する必要があります。
edir-hosts	edir-host は、OpenText Identity Console を介して接続する eDirectory の IP アドレスを格納するパラメータです。

重要

- OSP 関連の環境設定パラメータは、OSP と REST サービスを統合する計画の場合にのみ使用してください。
- REST サービスの監査を有効にするには、環境設定ファイルに監査関連パラメータを設定する必要があります。
- OSP HTTPS URL は、2048 ビットキーが含まれている証明書を使用して検証する必要があります。この検証は、4096 または 8192 ビットキーが含まれている証明書では失敗します。

OpenText eDirectory 用の REST サービスの設定

OpenText eDirectory 向けの REST API を設定するには、次の手順を実行します。

1 ソフトウェアライセンスおよびダウンロードポータルから eDirAPI_<*バージョン*>_Container.tar.gz をダウンロードします。たとえば、eDirAPI_140_Container.tar.gz のようになります。

2 次のコマンドを使用して、そのイメージを Docker のローカルレジストリにロードする必要があります。

```
tar -xvf eDirAPI_140_Container.tar.gz
```

```
docker load --input eDirAPI_140/eDirAPI_140.tar.gz
```

3 次のコマンドを使用して、Docker コンテナを作成します。

```
docker create --name edirapi-container --volume <volume-name>:/config/
--network=<network-type> --env ACCEPT_EULA=Y edirapi:<version>
```

次に例を示します。

```
docker create --name edirapi-container --volume edirapi-volume:/config/
--network=host --env ACCEPT_EULA=Y edirapi:1.4.0.0000
```

注

- ACCEPT_EULA 環境変数を Y に設定することによって、EULA に同意することができます。対話型モードの Docker create コマンドで -it オプションを使用することにより、コンテナの起動中に画面のプロンプトから EULA に同意することもできます。
- 上のコマンドの --volume パラメータによって、環境設定データとログデータを保存するボリュームが作成されます。本書では、edirapi-volume というサンプルボリュームを作成しました。

4 次のコマンドを使用して、ローカルファイルシステムからコンテナの /etc/opt/novell/eDirAPI/cert/keys.pfx に、サーバ証明書ファイル (.pfx) をコピーします。

```
docker cp <absolute path of server certificate file> edirapi-
container:/etc/opt/novell/eDirAPI/cert/keys.pfx
```

次に例を示します。

```
docker cp /home/user/keys.pfx edirapi-container:/etc/opt/novell/
eDirAPI/cert/keys.pfx
```

5 次のコマンドを使用して、ローカルファイルシステムからコンテナの /etc/opt/novell/eDirAPI/cert/SSCert.pem に、認証局証明書ファイル (.pem) をコピーします。

```
docker cp <absolute path of CA certificate file> edirapi-container:/
etc/opt/novell/eDirAPI/cert/SSCert.pem
```

次に例を示します。

```
docker cp /home/user/SSCert.pem edirapi-container:/etc/opt/novell/
eDirAPI/cert/SSCert.pem
```

複数の OpenText eDirectory ツリーに接続する場合は、すべての接続ツリーに対して個々の認証局証明書を取得する必要があります。たとえば、3 つの eDirectory ツリーに接続する場合、3 つすべての認証局証明書を Docker コンテナにコピーする必要があります。

```
docker cp /home/user/SSCert.pem edirapi-container:/etc/opt/novell/
eDirAPI/cert/SSCert.pem
docker cp /home/user/SSCert1.pem edirapi-container:/etc/opt/novell/
eDirAPI/cert/SSCert1.pem
docker cp /home/user/SSCert2.pem edirapi-container:/etc/opt/novell/
eDirAPI/cert/SSCert2.pem
```

6 次のコマンドを使用して、ローカルファイルシステムからコンテナの /etc/opt/novell/eDirAPI/conf/edirapi.conf に、環境設定ファイル (edirapi.conf) をコピーします。

```
docker cp <absolute path of CA certificate file> edirapi-container:/
etc/opt/novell/eDirAPI/conf/edirapi.conf
```

例 :

```
docker cp /home/user/edirapi.conf edirapi-container:/etc/opt/novell/
eDirAPI/conf/edirapi.conf
```

注 : 匿名バインドを無効にした eDirAPI を展開するには、edirapi.conf ファイルに "ldapserver"、"ldapuser"、および "ldappassword" パラメータが必要です。

サンプルの環境設定ファイルを以下に示します。

```
listen = ":9000"
pfxpassword = "novell"
bcert = "/etc/opt/novell/eDirAPI/cert/"
ospmode=false
edir-hosts = "<ip_address>:636"
```

注 : OpenText Identity Console から OpenText eDirectory にアクセスするには、「edir-hosts="x.x.x.x:ldaps_port"」を edirapi.conf ファイルに追加する必要があります。

例 :

edir-hosts="10.10.10.10:636"

7 次のコマンドで、Docker コンテナを起動します。

```
docker start <edirapi-container-name>
```

次に例を示します。

```
docker start edirapi-container
```

REST コンテナ内のデフォルトのログファイルの場所は、/config/eDirAPI/var/log/ です。この場所には、次のログファイルがあります。

- container-startup.log
- edirapi.log

注

- REST サーバの認証局証明書は、BEGIN CERTIFICATE で始まり、END CERTIFICATE で終わる必要があります。それ以外の値を指定した場合は、REST サーバはエラーメッセージを表示します。
- REST コンテナ内で最大 42000 の接続をサポートするには、次の 3 つのコマンドを実行して、ポートの範囲を拡大する必要があります。

```
ulimit -n 999999
cat /proc/sys/net/ipv4/ip_local_port_range
echo 1024 65535 > /proc/sys/net/ipv4/ip_local_port_range
```

- コンテナ起動ログファイルでは、次のメッセージを無視できます。

```
Setting IDCONSOLEMODE from Environment to false
```

データ永続性の管理

REST コンテナとともに、データ永続性のためのボリュームも作成されます。ボリュームを使用して古いコンテナの環境設定パラメータを使用するには、次の手順を実行します。

1. (省略可能) 次のコマンドを使用して、現在の Docker コンテナを停止します。

   ```
   docker stop edirapi-container
   ```

2. 次のコマンドを使用して、2 番目のコンテナを作成します。

   ```
   docker create --name edirapi-container-2 --network=host --volume
   edirapi-volume-2:/config/ edirapi:1.4.0<version>
   ```

3. 次のコマンドを使用して、2 番目のコンテナを開始します。

   ```
   docker start edirapi-container-2
   ```

4. (省略可能) これで、次のコマンドを使用して最初のコンテナを削除できるようになりました。

   ```
   docker rm edirapi-container
   ```

REST サービスによる監査

OpenText eDirectory REST サーバは、Sentinel サーバに CEF 監査イベントログを送信することができます。REST サーバは、ArcSight Smart Connector を使用して監査データを送信します。SmartConnector の設定と使用方法の詳細については、『ArcSight SmartConnector ユーザガイド』を参照してください。

REST サービスの監査を有効にするには、環境設定ファイル内の監査関連パラメータを設定する必要があります。詳細については、830 ページの 「OpenText eDirectory 向けの REST サービスのインストールを計画する」を参照してください。

REST イベントを理解する

デフォルトでは、すべての REST イベントが有効になっています。ユーザの組織で不要な場合は、特定のイベントを無効にすることができます。OpenText eDirectory REST サーバでは、次のイベントの監査が可能です。

イベント	説明
ENABLESERVICESTARTAUDIT	REST サービスが開始された場合にイベントが生成されます
ENABLESERVICESTOPAUDIT	REST サービスが停止した場合にイベントが生成されます
ENABLELOGSESSIONCREATIONAUDIT	REST セッションが作成された場合にイベントが生成されます
ENABLELOGSESSIONTERMINATIONAUDIT	REST セッションが終了した場合にイベントが生成されます

注 : ログは /var/opt/novell/eDirAPI/log/edirapi_auditlog.log にあります。

例

次のセッションの作成イベントの例を参照してください。

```
Oct 10 15:37:17 eDirAPI
CEF:0|NetIQ|eDirAPI|1.0|000B0510|SESSION_CREATE|3|dvc=10.71.128.233
dvchost=SLES12SP3-SHREYAS-128233 rt=Oct 10 2019 15:37:17 dtz=IST
src=164.99.136.60 spt=59132 suser=cn\=admin,o\=novell
duser=cn\=admin,o\=novell cn1Label=CorrelationID cn1=rtpL9xt-
tzBR92fEGt9rrczA_1M2vHrGM4Q_8AjEmSU= cs1Label=Client Address
cs1=164.99.136.60 cs2Label=Tree Name cs2=SHREYAS_TREE2 sproc=eDirAPI
sourceServiceName=edirapi reason=201 outcome=Success
```

REST コンテナを使用して LDAP パスワードを変更する

REST コンテナを使用して LDAP パスワードを変更するには、次の手順を実行します。

1 次のコマンドを使用して、コンテナにログインします。

```
docker exec -it <container_name> bash
```

2 次のコマンドを使用して、パスワードストアに新しいパスワードを保存します。

```
LD_LIBRARY_PATH=/opt/novell/lib64/:/opt/novell/eDirectory/lib64/:/opt/
netiq/common/openssl/lib64/ /opt/novell/eDirAPI/sbin/passwdstore -a
<Admin DN>
```

上記のコマンドを使用すると、パスワードを求めるプロンプトが表示されます。新しいパスワードを入力します。

次に例を示します。

```
LD_LIBRARY_PATH=/opt/novell/lib64/:/opt/novell/eDirectory/lib64/:/opt/
netiq/common/openssl/lib64/ /opt/novell/eDirAPI/sbin/passwdstore -a
cn=admin,o=novell
```

3 次のコマンドを使用して、コンテナコンソールを終了します。

```
exit
```

4 コンテナを再起動します

```
docker restart <container name>
```

REST コンテナを使用してサーバ証明書を変更する

REST コンテナを使用してサーバ証明書を変更するには、次の手順を実行します。

1 次のコマンドを実行して、コンテナの任意の場所に新しいサーバ証明書 (例 : new-keys.pfx) をコピーします。

```
docker cp /path/to/new-keys.pfx <container_id/name>:/tmp/new-keys.pfx
```

2 次のコマンドを使用して、コンテナにログインします。

```
docker exec -it <container_name> bash
```

3 NLPCERT を実行して、キーを保存します。

```
LD_LIBRARY_PATH=/opt/novell/lib64/:/opt/novell/eDirectory/lib64/:/opt/
netiq/common/openssl/lib64/ /opt/novell/eDirAPI/sbin/nlpcert -i /tmp/
new-keys.pfx -o /etc/opt/novell/eDirAPI/conf/ssl/private/cert.pem
```

上記のコマンドを使用すると、サーバ証明書のパスワード入力を求めるプロンプトも表示されます。パスワードを入力します。

4 次のコマンドを使用して、コンテナコンソールを終了します。

```
exit
```

5 コンテナを再起動します

```
docker restart <container name>
```

OpenText eDirectory 用の REST サービスのアップグレード

REST イメージの新しいバージョンが使用可能になった場合、管理者は、最新バージョンの REST サービスをコンテナに展開するためのアップグレード手順を実行できます。アップグレードを実行する前に、すべての必要なアプリケーション関連データを Docker ボリュームに永続的に保存するようにしてください。Docker コンテナを使用して REST サービスをアップグレードするには、次の手順を実行します。

1 ソフトウェアライセンスおよびダウンロードポータルから最新バージョンの Docker イメージをダウンロードしてロードし、最新バージョンの OpenText eDirectory をインストールする手順を実行します。

2 最新の Docker イメージがロードされた後、次のコマンドを使用して現在の Docker コンテナを停止します。

```
docker stop <edirapi-container-name>
```

例 :

```
docker stop edirapi-container-1
```

3 次のコマンドを実行して、既存の OpenText eDirectory コンテナを削除します。

```
docker rm < edirapi -container-name>
```

例 :

```
docker rm edirapi -container-1
```

4 (省略可能) 次のコマンドを実行して、廃止された OpenText eDirectory Docker イメージを削除します。

```
docker rmi <edirapi-image ID>
```

例 :

```
docker rmi edirapi: 1.4.1.0000
```

5 次のコマンドを使用して、新しい REST Docker イメージを使用して新しいコンテナを作成します。

```
docker create --name < edirapi -container-name> --env ACCEPT_EULA=Y --
network=<network-type> --volume <volume-name>:/config/
edirapi:<version>
```

例 :

```
docker create --name edirapi -container-2 --env ACCEPT_EULA=Y --
network=host --volume edirapi -volume:/config/ edirapi: 1.4.1.0000
```

6 次のコマンドを使用して、ローカルファイルシステムから新しく作成されたコンテナに、環境設定ファイル ((edirapi.conf)) を /etc/opt/novell/eDirAPI/conf/edirapi.conf としてコピーします。

```
docker cp <absolute path of configuration file> edirapi-
containername:/etc/opt/novell/eDirAPI/conf/edirapi.conf
```

例 :

```
docker cp /home/user/edirapi.conf edirapi-container-2:/etc/opt/novell/
eDirAPI/conf/edirapi.conf
```

サンプルの環境設定ファイルを以下に示します。

```
listen = ":9000"

pfxpassword = "novell"

bcert = "/etc/opt/novell/eDirAPI/cert/"

ospmode=false

edir-hosts = "<ip_address-1>:636,<ip_address-2>:636"
```

注 :

- OpenText eDirectory 1.7.2 以上にアップグレードする場合、コンテナにコピーする前に、OpenText eDirectory サーバ IP を edirapi.conf ファイルに追加する必要があります。
- OpenText eDirectory を複数の OpenText eDirectory ツリーに接続する場合は、それらの IP アドレスまたはドメイン名をコンマで区切って入力します。

---

7 次のコマンドを使用して、2 番目のコンテナを開始します。

```
docker start <edirapi-container-name>
```

例 :

```
docker start edirapi -container-2
```

8 実行中のコンテナのステータスを確認するには、次のコマンドを実行します。

```
docker ps -a
```

注 : REST Services 1.4 は、REST 1.3 に続く次のリリースです。これら 2 つのリリースの間に、REST サービスの暫定リリースはありませんでした。1.4 の後、OpenText eDirectory では REST 1.4.0.0100 をリリースし、OpenSSL 1.0.2zd のサポートを追加しました。この OpenSSL バージョンには、潜在的なセキュリティ脆弱性の修正が含まれています。この問題の詳細については、このページを参照してください。

Rest API マニュアル

REST API マニュアルには、次のリンクのいずれかを使用してアクセスできます。

- リソースサーバモード (https://www.netiq.com/documentation/edirectory-93/resources/ResourceModeOpenapi/) の API マニュアル。このモードの認証は、OSP トークンを使用して行われます。
- Simple ログインモード (https://www.netiq.com/documentation/edirectory-93/resources/SimpleLoginOpenapi/) の API マニュアル。このモードの認証は、LDAP ユーザ名とパスワードを使用して行われます。

A NMAS の注意事項

この付録では、次のトピックについて説明します。

- 843 ページの 「独立したパーティションとしてのセキュリティコンテナの設定」
- 843 ページの 「複数のセキュリティコンテナを持つツリーのマージ」

独立したパーティションとしてのセキュリティコンテナの設定

NMAS は、OpenText eDirectory ツリー全体に適用されるポリシーの保存に依存しています。OpenText eDirectory ツリーは、実際にはセキュリティドメインとして機能します。セキュリティポリシーは、ツリー内のすべてのサーバで使用できる必要があります。

NMAS には、OpenText eDirectory ツリーの［ルート］から作成された、セキュリティコンテナ内の認証ポリシーとログイン方式の設定データが格納されます。この情報は、NMAS を使用可能なすべてのサーバで読み込みアクセスできる必要があります。セキュリティコンテナの目的は、ログイン、認証、キー管理などのセキュリティプロパティに関するグローバルポリシーを保持することです。

NMAS により、独立したパーティションとしてセキュリティコンテナを作成し、作成したコンテナを広い範囲で複製することを推奨します。このパーティションは、ツリー内の信頼性の高い複数のサーバでのみ、読み書き可能なパーティションとして複製することをお勧めします。

注 : セキュリティコンテナはグローバルポリシーを格納しており、サーバでは OpenText eDirectory ツリーに指定したセキュリティポリシー全般が変更される可能性があるため、書き込み可能なレプリカを配置するサーバを選択する場合には注意が必要です。NMAS を使用してユーザがログインするには、ユーザオブジェクトのレプリカが NMAS サーバ上に存在する必要があります。

複数のセキュリティコンテナを持つツリーのマージ

一方のツリーまたは両方のツリーにセキュリティコンテナがインストールされている OpenText eDirectory ツリーをマージする場合、特に注意が必要です。この手順は時間を要する複雑な作業になる可能性があるため、本当に実行する必要があるかどうか確認してください。

複数のセキュリティコンテナを持つツリーをマージするには次を実行します。

1. OpenText Identity Console で、マージするツリーを指定します。
2. ソースツリーにするツリーと、ターゲットツリーにするツリーを指定します。

ソースツリーおよびターゲットツリーについて、次のセキュリティ上の配慮事項に注意してください。

- ソースツリーの組織の認証局によって署名された証明書をすべて削除すること。
- ソースツリーの組織の認証局を削除すること。
- ソースツリー上の OpenText SecretStore に保存されたすべてのユーザシークレットを削除すること。
- ソースツリー内のすべての NMAS ログインメソッドを削除し、ターゲットツリーに再インストールすること。
- ツリーをマージする際に、ソースツリー内に存在したすべての NMAS ユーザを再登録すること。
- ソースツリー内に存在したすべてのユーザおよびサーバに、ツリーをマージする際に新しい証明書を作成すること。
- ソースツリー内に存在したすべてのユーザのシークレットを SecretStore に再インストールすること。

ソースツリーおよびターゲットツリーの両方が Security というコンテナをツリーのルート直下に保持していない場合、またはツリーのうち一方だけが Security コンテナを保持している場合、これ以上の処置は不要です。その他の場合には、このセクションの残りの手順を続けます。

注 : EBA が有効なサーバが含まれる 2 つの OpenText eDirectory ツリーをマージしないでください。

ツリーのマージ前に実行する製品固有の操作

このセクションでは、次のことを説明します。

- 844 ページの 「OpenText Certificate Server」
- 846 ページの 「OpenText Single Sign-on」
- 846 ページの 「NMAS」
- 847 ページの 「OpenText セキュリティドメインインフラストラクチャ」
- 847 ページの 「その他のセキュリティ固有の操作」

OpenText Certificate Server

製品の使用状況によっては、指定されたオブジェクトと項目が存在しない可能性があります。次の手順で指定されたオブジェクトや項目がソースツリーに存在しない場合には、手順を省略できます。

1 ソースツリー内のルート認証局証明書は、ターゲットツリーにインストールする必要があります。

ルート認証局証明書は、ルート認証局コンテナに含まれる、ルート認証局オブジェクトに格納されています。ルート認証局コンテナはツリー内の任意の場所に作成できます。ただし、セキュリティコンテナ内のルート認証局コンテナに存在するルート認証局証明書は、ソースツリーからターゲットツリーに手動で移動する必要があります。

2 ターゲットツリーにルート認証局証明書をインストールします。

2a ソースツリーのセキュリティコンテナでルート認証局を選択します。

2b ソースツリーで使用されている正確な名前 (ステップ 2a) で、ルート認証局コンテナをターゲットツリーのセキュリティコンテナに作成します。

2c ソースツリーで、選択したルート認証局コンテナのルート認証局オブジェクトを開き、証明書をエクスポートします。

重要 : 選択した場所とファイル名は、次の手順で使用するため覚えておいてください。

2d ターゲットツリーで、ステップ 2b で作成したコンテナのルート認証局オブジェクトを作成します。ソースツリーと同じ名前を指定し、証明書を求められたら、ステップ 2c で作成したファイルを指定します。

2e ソースツリーのルート認証局オブジェクトを削除します。

2f 選択したルート認証局コンテナ内のすべてのルート認証局オブジェクトがターゲットツリーにインストールされるまで、ステップ 2c からステップ 2e を繰り返します。

2g ソースツリーのルート認証局コンテナを削除します。

2h すべてのルート認証局コンテナがソースツリー内で削除されるまで、ステップ 2a からステップ 2f を繰り返します。

3 ソースツリーの組織の認証局を削除します。

組織の認証局オブジェクトは、セキュリティコンテナ内に存在します。

重要 : ソースツリーの組織の認証局によって署名された証明書は、この手順以降は使用できません。これには、ソースツリーの組織の認証局によって署名された、サーバ証明書とユーザ証明書も含まれます。

---

4 ソースツリーの組織の認証局によって署名された証明書を持つ、ソースツリー内の暗号化キーオブジェクト (KMO) をすべて削除します。

他の組織の認証局によって署名された証明書を持つ、ソースツリー内の暗号化キーオブジェクトは引き続き有効で、削除する必要はありません。

暗号化オブジェクトに署名している CA の識別情報がわからない場合は、暗号化オブジェクトのプロパティページにある［証明書］タブの［ルート認証局証明書］セクションを参照します。

5 ソースツリーの組織の認証局によって署名された、ソースツリー内のユーザ証明書をすべて削除します。

ソースツリー内のユーザがすでに証明書とプライベートキーをエクスポートしている場合、エクスポートされた各証明書とキーは引き続き使用できます。を実行した後では、OpenText eDirectory 内に残っているプライベートキーと証明書は使用できません。ステップ 3

証明書を持つユーザごとに、ユーザオブジェクトのプロパティを開きます。［セキュリティ］タブの［証明書］セクションの下に、そのユーザのすべての証明書を示すテーブルが表示されます。発行者として組織の認証局が設定されている証明書はすべて削除してください。

OpenText Single Sign-on

OpenText Single Sign-on がソースツリー内のサーバのいずれかにインストールされている場合、ソースツリーのユーザ用 OpenText Single Sign-on シークレットをすべて削除する必要があります。

ソースツリーで OpenText Single Sign-on を使用するすべてのユーザについて、ユーザオブジェクトのプロパティを開きます。［セキュリティ］タブの［SecretStore］セクションの下に、そのユーザのシークレットすべてが表示されます。表示されたシークレットをすべて削除します。

注 : 製品の使用状況によっては、指定されたオブジェクトと項目が存在しない可能性があります。指定されたオブジェクトや項目が存在しない場合には、この手順を省略できます。

NMAS

製品の使用状況によっては、指定されたオブジェクトと項目が存在しない可能性があります。指定されたオブジェクトや項目が存在しない場合には、手順を省略できます。

1 ターゲットツリーに、ソースツリーに存在してターゲットツリーには存在しない、NMAS ログインメソッドをインストールします。

必要なすべてのクライアントとサーバのログインコンポーネントをターゲットツリーに適切にインストールするには、OpenText のオリジナルソースやベンダー提供のソースを使用して、すべて新しいログインメソッドをインストールすることを推奨します。

メソッドは既存のサーバファイルから再インストールできますが、通常、OpenText のパッケージやベンダー提供のパッケージを使用してクリーンインストールを行う方が簡単で確実です。

2 ソースツリーで以前に確立されたログインシーケンスをターゲットツリーで使用できるようにするには、対象のログインシーケンスを移行します。

2a OpenText Identity Console で、ソースツリーのセキュリティコンテナを選択します。

2b ［ログインポリシー］オブジェクトを右クリックして、［プロパティ］をクリックします。

2c ［Defined Login Sequences］ドロップダウンリストに表示されたログインシーケンスごとに、使用するログインメソッド (右側のペインに表示) をメモします。

2d ターゲットツリー内のセキュリティコンテナを選択し、ステップ 2c でメモしたログインメソッドと同じものを使用して、ログインシーケンスを複製します。

2e 操作が終了したら［OK］をクリックします。

3 ソースツリーの NMAS ログインセキュリティ属性を削除します。

3a ソースツリーのセキュリティコンテナで、ログインポリシーオブジェクトを削除します。

3b ソースツリーの許可されたログインメソッドコンテナで、ログインポリシーオブジェクトを削除します。

3c ソースツリーの許可されたログインメソッドコンテナを削除します。

3d ソースツリーの許可されたポストログインメソッドコンテナで、ログインポリシーオブジェクトを削除します。

3e ソースツリーの許可されたポストログインメソッドコンテナを削除します。

注 : 許可されたログインメソッドを削除するには、ldapdelete を使用します。

OpenText セキュリティドメインインフラストラクチャ

製品の使用状況によっては、指定されたオブジェクトと項目が存在しない可能性があります。指定されたオブジェクトや項目が存在しない場合には、手順を省略できます。

1 W0 オブジェクトとソースツリーの KAP コンテナを削除します。

KAP コンテナは、セキュリティコンテナ内に存在します。W0 オブジェクトは、KAP コンテナ内に存在します。

2 ソースツリー内のすべてのサーバで、Linux では /var/opt/novell/nici/uid/nicisdi.key ファイル、Windows では %SystemRoot%\SysWOW64\Novell\NICI\nicisdi.key を削除することにより、セキュリティドメインインフラストラクチャ (SDI) キーを削除します。

重要 : ソースツリー内のすべてのサーバで、このファイルを削除したことを確認します。

その他のセキュリティ固有の操作

ソースツリー内にセキュリティコンテナが残っている場合、ツリーをマージする前に、セキュリティコンテナを削除します。

ツリーのマージを実行する

DSMerge ユーティリティを使用して、OpenText eDirectory ツリーをマージします。詳細については、849 ページの付録 B「OpenText eDirectory 用の Linux コマンドとそれらの使用法」を参照してください。

ツリーのマージ後に実行する製品固有の操作

このセクションでは、次のことを説明します。

- 847 ページの 「OpenText Certificate Server」
- 848 ページの 「OpenText Single Sign-on」
- 848 ページの 「NMAS」

OpenText Certificate Server

OpenText Certificate Server を使用している場合、ツリーのマージ後に、ソースツリー内に前に存在していたサーバとユーザに対して必要に応じて証明書を再発行します。

OpenText Single Sign-on

OpenText Single Sign-on を使用している場合、ツリーのマージ後に、ソースツリー内に前に存在していたユーザの SecretStore シークレットを必要に応じて再作成する必要があります。

NMAS

NMAS を使用している場合、ツリーのマージ後に、ソースツリー内に存在していた NMAS ユーザを必要に応じて再登録する必要があります。

詳細については、667 ページの第 24 章「OpenText eDirectory の認証フレームワークについて」を参照してください。

B OpenText eDirectory用のLinuxコマンドとそれらの使用法

この章では、Linux での OpenText eDirectory 用の各ユーティリティとそれらの使用法を一覧にまとめています。

- 849 ページの 「一般ユーティリティ」
- 854 ページの 「LDAP 固有のコマンド」

一般ユーティリティ

このセクションでは、Linux での OpenText eDirectory ユーティリティとその使用法を一覧で示します。

注 : インストール後に、ユーティリティの ndsconfig、ndscheck、および ndslogin をサーバのインストール場所 (この場所は、デフォルトでは /opt/novell/eDirectory/bin です) から実行してください。インストールパッケージから ndsconfig を実行しないでください。

OpenText eDirectory ユーティリティの使用法の詳細については、各ユーティリティのマニュアルページおよび 925 ページの 「Linux でのユーティリティのトラブルシューティング」を参照してください。

コマンド	説明	使用法
nds-install	OpenText eDirectory をインストールするユーティリティです。	`nds-install [-h] [--help] [-i] [-j] [-u]`

コマンド	説明	使用法
ndsconfig	OpenText eDirectory の設定	ndsconfig <new> [-t <treename>] [-n <server context>] [-a <admin FDN>] [-w <password>] [-B ip_address1|interface1@port1,ip_address2|interface2@port2....] [-b port to bind] [-i] [-S <server name>] [-D <instance path>] [-d <path for dib>] [-m <module>] [-e] [-R|-r] [-c] [-L <ldap port>] [-l <SSL port>] [-P <LDAP URLs>] [-o http port] [-O https port] [--config-file <absolute path for configuration file>] [--configure-eba-now <yes/no>] ndsconfig <def> [-t <treename>] [-n <server context>] [-a <admin FDN>] [-w <password>] [-B ip_address1|interface1@port1,ip_address2|interface2@port2....] [-b port to bind] [-i] [-S <server name>] [-D <instance path>] [-d <path for dib>] [-m <module>] [-e] [-R|-r] [-c] [-L <ldap port>] [-l <SSL port>] [-P <LDAP URLs>] [-o http port] [-O https port] [--config-file <absolute path for configuration file>] [--configure-eba-now <yes/no>] ndsconfig add [-t <treename>] [-n <server context>] [-a <admin FDN>] [-w <password>] [-B ip_address1|interface1@port1,ip_address2|interface2@port2....] [-b port to bind] [-E] [-e] [-R|-r] [-c] [-L <ldap port>] [-l <SSL port>] [-P <LDAP URLs>] [-o http port] -O [https port] [-S <server name>] [-D <instance path >] [-d <path for dib>] [-p <IP address[:port]>] [-m <module>] [--config-file <absolute path for configuration file>] [--configure-eba-now <yes/no>] ndsconfig rm [-a <*admin FDN*>] [-w <*admin password*>] [-W <*obfuscated_password_file*>] [-c] [--config-file <*configuration file*>] ndsconfig upgrade [-a <admin FDN>] [-w <password>] [-c] [-j] [--config-file <absolute path for configuration file>] [--configure-eba-now <yes/no>] ndsconfig {set <*valuelist*> | get [<*paramlist*>] | get help [<*paramlist*>]}

コマンド	説明	使用法
ndscheck	ツリーの状態をチェックするユーティリティです。	ndscheck [--help | -?] Display command usage ndscheck [--version | -v] Display version information ndscheck [-h <*hostname port*>] [-a <*admin FDN*>] [-F <*log file*>] [-D] [-q] [-w <*admin password*>] [-W] [--config-file <*file name*>] ndscheck [-a <*admin FDN*>] [-W] [--config-file <*file name*>] 次に例を示します。 ndscheck -a admin.novell -W --config-file /etc/opt/novell/eDirectory/conf-1/nds.conf
ndsmanage	OpenText eDirectory のインスタンスを一覧表示するユーティリティです。	ndsmanage [-a] ndsmanage [<*username*>]

コマンド	説明	使用法
ndsbackup	OpenText eDirectory オブジェクトのアーカイブを作成し、OpenText eDirectory オブジェクトを追加または抽出します。	ndsbackup c [f <*ndsbackupfile*>] [e] [v] [w] [X<*exclude-file*>] [R] [Replica-server-name] [-a <*admin-user*>] [-I <*include-file*>] [-E <*password*>] [--config-file <*configuration_file_path*>]... [eDirectoryobject] ndsbackup r [f <*ndsbackupfile*>] [e] [v] [w] [X<*exclude-file*>] [R] [Replica-server-name] [-a <admin-user>] [-I <*include-file*>] [-E <*password*>] [--config-file <*configuration_file_path*>]... [eDirectoryobject] ndsbackup t [f <*ndsbackupfile*>] [e] [v] [w] [X<*exclude-file*>] [R] [Replica-server-name] [-a <admin-user>] [-I <*include-file*>] [-E <*password*>] [--config-file <configuration_file_path>]... [eDirectoryobject] ndsbackup x [f <*ndsbackupfile*>] [e] [v] [w] [X<*exclude-file*>] [R] [Replica-server-name] [-a <admin-user>] [-I <*include-file*>] [-E <*password*>] [--config-file <*configuration_file_path*>]... [eDirectoryobject] ndsbackup s [e] [v] [w] [X<*exclude-file*>] [R] [Replica-server-name] [-a <*admin-user*>] [-I <*include-file*>] [-E <*password*>] [--config-file <*configuration_file_path*>]... [eDirectoryobject] ndsbackup --version ndsbackup [option] [file] [-a <*admin FDN*>] [-p passstore] [--config-file <*file name*>] 次に例を示します。 ndsbackup cvf /tmp/test.bak -a admin.novell -p passstore --config-file /etc/opt/novell/eDirectory/conf-1/nds.conf
ndslogin	OpenText eDirectory の認証を検証するための診断ユーティリティです。	ndslogin [-t treename] [-p password] [-s] [-n] [-c] [[-i] | [-I]] [[-h hostname[:port]] | [--config-file <configuration file>]] <userFDN>

コマンド	説明	使用法
ndsd	NDS デーモン	/opt/novell/eDirectory/sbin/ndsd [--config-file *configfile*]
ndsimonitor	HTTP を使用して、OpenText eDirectory ツリーにあるサーバを監視および診断します。	/opt/novell/eDirectory/bin/ndsimonitor [-l [-d <*path of ndsimonitor conf files*>] | u] [-h <*local_interface:port*>] [--config-file <*configuration_file_path*>]
ndsmerge	2 つの OpenText eDirectory ツリーをマージするユーティリティです。	ndsmerge [-m target-tree target-admin source-admin [target-container]] [-c] [-t] [-r target-tree source-admin] [-h <*local_interface:port*>] [--config-file <*configuration_file_path*>]
ndsrepair	レコード、スキーマ、バインダリオブジェクト、および外部参照など、OpenText eDirectory データベースの問題を修復および修正するためのユーティリティです。	ndsrepair {-U |-E |-C |-P [Ad] |-S [Ad]|-N |-T |-J <*entry_id*>} [-A <*yes/no*>] [-O <*yes/no*>][-F <*filename*>] [-h <*local_interface:port*>] [--config-file <*configuration_file_path*>]
		ndsrepair -R [-l <*yes/no*>][-u <*yes/no*>][-m <*yes/no*>][-i <*yes/no*>][-f <*yes/no*>][-d <*yes/no*>][-t <yes/no>][-o <yes/no>][-r <yes/no>][-v <yes/no>][-c <yes/no>][-A <yes/no>][-O <yes/no>][-F <*filename*>] [-h <*local_interface*>] [--config-file <*configuration_file_path*>]
	使用するために解放できる、データベース内の空き領域に関する情報を表示するように、ndsrepair に指示できます。	ndsrepair –I [--config-file <*configuration_file_path*>]
ndssch	OpenText eDirectory スキーマ拡張ユーティリティです。	ndssch [-h <*hostname*>[:<port>]][-t <*treename*>][-F <*logfile*>] <*admin-FDN*> <*schemafile*> ...
		ndssch [-h <*hostname*>[:<port>]][-t <*treename*>] [-d] <*admin-FDN*> <*schemafile*> [schema description] ...
ndssnmp	OpenText eDirectory の SNMP サービスモジュールです。	/opt/novell/eDirectory/bin/ndssnmp
ndssnmpconfig	SNMP トラップ環境設定ユーティリティです	ndssnmpconfig [-h <*hostname[:port]*>] [-p <*password*>] [-a <*userFDN*>] [-c <*command*>]
ndssnmpsa	OpenText eDirectory SNMP サブエージェントデーモンです。	/opt/novell/eDirectory/bin/ndssnmpsa

コマンド	説明	使用法
ndsstat	サーバ情報を表示するユーティリティです	ndsstat { -r -s -p <*partitionname*>} [-n] [[-h <*hostname* | *IP address*>:<*port*>] | [--config-file <*configuration file*>]]
ndstrace	サーバのデバッグメッセージを表示するユーティリティです	ndstrace [-l|-u|-c "command1;....."|--version] [-h <*local_interface:port*>] [--config-file <*configuration_file_path*>]
nds-uninstall	OpenText eDirectory をアンインストールするユーティリティです。	nds-uninstall [-s][-h]
nldap	NDS デーモンの LDAP サービス	/opt/novell/eDirectory/sbin/nldap
nmasinst	NMAS 環境設定ユーティリティ	nmasinst -i <*admin-FDN*> <*treename*> [-h <*hostname*>[:port]] nmasinst -addmethod <*admin-FDN*> <*treename*> <*config.txt file*> [-h <*hostname*>[:port]]
npki	Novell 公開鍵インフラストラクチャサービス	/opt/novell/eDirectory/sbin/npki

LDAP 固有のコマンド

コマンド	説明	使用法
ldapconfig	LDAP サーバおよび LDAP グループオブジェクトを設定するユーティリティです	ldapconfig get [...] | set <*attribute-value-list*> [-t <*treename*> | -p <*hostname*>[:port] | --config-file <*configuration file*>] [-w <*password*>] [-a <*user FDN*>] [-f] ldapconfig [-t <*treename*> | -p <*hostname*>[:port]] [-w <*password*> | --config-file <*configuration file*>] [-a <*user FDN*>] [-V] [-R] [-H] [-f] -v <*attribute*>,<*attribute2*>... ldapconfig [-t <*treename*> | -p hostname[:port] | --config-file <*configuration file*>] [-w <*password*>] [-a <*admin FDN*>] [-V] [-R] [-H] [-f] -s <*attribute*>=<*value*>,...

コマンド	説明	使用法
ldapadd ldapmodify	LDAP サーバからエントリを追加または変更します	ldapmodify [-a] [-c] [-C] [-M] [-P] [-r] [-n] [-v] [-F] [-l <limit>] [-M[M]] [-d <debuglevel>] [-e <key filename>] [-D <binddn>] [[-W]|[-w <passwd>]] [-H ldap://[ldap host ip]:[ldap-port]] [-P <version>] [-Z[Z]] [-f <file>] ldapadd [-c] [-C] [-l] [-M] [-P] [-r] [-n] [-v] [-F] [-l <*limit*>] [-M[M]] [-d <*debuglevel*>] [-e <*key filename*>] [-D <*binddn*>] [[-W]| [-w <*passwd*>]] [-H ldap://[ldap host ip]:[ldap-port] [-P <*version*>] [-Z[Z]] [-f <*file*>]
ldapdelete	LDAP サーバからエントリを削除します	ldapdelete [-n] [-v] [-c] [-r] [-l] [-C] [-M] [-d <*debuglevel*>] [-e <*key filename*>] [-f <*file*>] [-D <*binddn*>] [[-W]| [-w <*passwd*>]] [-H ldap://[ldap host ip]:[ldap-port]} [-Z[Z]] [dn]...
ldapmodrdn	LDAP のエントリの相対識別名 (RDN) 変更ツールです	ldapmodrdn [-r] [-n] [-v] [-c] [-C] [-l] [-M] [-s <*newsuperior*>] [-d <*debuglevel*>] [-e <*key filename*>] [-D <*binddn*>] [[-W]|[-w <*passwd*>]] [-h <*ldaphost*>] [-p <*ldapport*>] [-Z[Z]] [-f <*file*>] [dn <*newrdn*>]
ldapsearch	LDAP の検索ツールです	ldapsearch [-n] [-u] [-v] [-t] [-A] [-T] [-C] [-V] [-M] [-P] [-L] [-d <*debuglevel*>] [-e <*key filename*>] [-f <*file*>] [-D <*binddn*>] [[-W]| [-w <*bindpasswd*>]] [-h <*ldaphost*>] [-p <*ldapport*>] [-b <*searchbase*>] [-s <*scope*>] [-a <*deref*>] [-l <*time limit*>] [-z <*size limit*>] [-Z[Z]] filter [attrs....]

コマンド	説明	使用法
ndsindex	OpenText eDirectory データベースインデックスの作成、リスト表示、一時停止、再開、または削除に使用するユーティリティです。	ndsindex list [-h <*hostname*>] [-p <*port*>] [-D <*bind DN*>*] [-W|[-w <*password*>]] [-l <*limit*>] [-s <*eDirectory Server DN*>] [-Z[Z]] [<*indexName1*>, <*indexName2*>.....] ndsindex add [-h <*hostname*>] [-p <*port*>] [-D <*bind DN*>] -W|[-w <*password*>] [-l <*limit*>] [-s <*eDirectory Server DN*>] [-Z[Z]] <*indexDefinintion1*> [<*indexDefinintion2*>.....] ndsindex delete [-h <*hostname*>] [-p <*port*>] [-D <*bind DN*>] [-W|[-w <*password*>]] [-l <*limit*>] [-s <*eDirectory Server DN*>] [-Z[Z]] <*indexName1*> [<*indexName2*>.....] ndsindex resume [-h <*hostname*>] [-p <*port*>] -D <*bind DN*> [-W|[-w <*password*>]] [-l <*limit*>] [-s <*eDirectory Server DN*>] [-Z[Z]] <*indexName1*> [<*indexName2*>.....] ndsindex suspend [-h <*hostname*>] [-p <*port*>] [-D <*bind DN*>] [-W|[-w <*password*>]] [-l <limit>] [-s <*eDirectory Server DN*>] [-Z[Z]] <*indexName1*> [<*indexName2*>.....]
ice	このユーティリティは、エントリをファイルからLDAP ディレクトリにインポートし、ファイルのディレクトリ内のエントリを変更し、エントリをファイルにエクスポートし、ファイルの属性とクラス定義を追加します。	ice -S LDAP -s server1.acme.com -p 636 -L cert-server1.pem -d cn=admin,c=us -w password -F objectClass=* -c sub -D LDIF -f server1.ldif -e des -E secret ice -S LDIF -f server1.ldif -e des -E secret -D LDAP -s server2.acme.com -p 636 -L cert-server2.pem -d cn=admin,c=us -w password

ユーザ名およびパスワードでの特殊文字

ユーザ名およびパスワードで特殊文字を使用すると、OpenText eDirectory のインストール時またはスキーマ拡張時に値が渡される場合に問題が生じる可能性があります。ユーザ名またはパスワードに $ や # などの特殊文字が含まれる場合、その特殊文字の前にバックスラッシュ (\) を付けてエスケープしてください。

たとえば、cn=admin$name.o=container という管理者ユーザ名は、cn=admin\$name.o=container で渡す必要があります。

コマンドラインにパラメータ値を入力する場合、その文字をエスケープしたり、その値の前後を一重引用符で囲んだりできます。

次に例を示します。

```
cn=admin\$name.o=container
```

または

```
'cn=admin$name.o=container'
```

C OpenSLP for OpenText eDirectoryの設定

この付録では、ネットワーク管理者向けに、Novell Client なしで OpenSLP for OpenText eDirectory をインストールための適切な環境設定について説明します。

- 859 ページの 「Service Location Protocol」
- 859 ページの 「SLP の基本」
- 862 ページの 「環境設定パラメータ」

Service Location Protocol

OpenSLP は、IETF Service Location Protocol Version 2.0 規格のオープンソースによる実装です。この規格は、IETF Request-For-Comments (RFC) 2608 (http://www.ietf.org/rfc/rfc2608.txt?number=2608) で文書化されました。

SLP v2 プロトコルの実装に加え、OpenSLP ソースコードが提供するインタフェースは、SLP 機能にプログラム的にアクセスするためのもう 1 つの IETF 規格を実装したもので、RFC 2614 (http://www.ietf.org/rfc/rfc2614.txt?number=2614) で文書化されています。

SLP の動作の詳細を理解するためには、この 2 つのドキュメントを参照し、熟読してください。読みやすい文書ではありませんが、インターネットでの SLP の正しい設定を行うためには重要なドキュメントです。

OpenSLP プロジェクトの詳細については、OpenSLP (http://www.OpenSLP.org) の Web サイトと SourceForge (http://sourceforge.net/projects/openslp) の Web サイトを参照してください。OpenSLP の Web サイトには、環境設定に関する貴重なヒントを含んださまざまな文書があります。ただし、このガイドの作成時点では、これらのドキュメントの多くは未完成です。

SLP の基本

Service Location Protocol では、次の 3 種類のコンポーネントが定義されています。

- ユーザエージェント (UA)
- サービスエージェント (SA)
- ディレクトリエージェント (DA)

ユーザエージェントは、クライアントがサービスを問い合わせたり、サービスがそれ自体を通知するためのプログラムインタフェースを提供します。ユーザエージェントはディレクトリエージェントに接続し、指定したスコープ内の指定したサービスクラスに登録されたサービスを問い合わせます。

サービスエージェントは、SLP で登録されたローカルサービスを持続的に格納し、維持する場所を提供します。サービスエージェントは主として、登録済みのローカルサービスをメモリ内データベースとして維持します。この場合、サービスはローカル SA がない限り SLP で登録できません。クライアントがサービスを検出するのは UA ライブラリ内のみですが、登録するには SA が必要です。これは主に、ディレクトリエージェントを受信して登録を維持するためには、登録済みサービスの存在を SA が定期的に表明する必要があるためです。

ディレクトリエージェントは、通知されたサービスに対して長期間持続的にキャッシュを提供し、ユーザエージェントがサービスを検索するためのアクセスポイントとなります。キャッシュ機能を提供する DA は、SA が新しいサービスを通知するのを受信し、これらの通知をキャッシュします。DA のキャッシュは短時間で完了します。ディレクトリエージェントは、期限切れのアルゴリズムを使用してエントリキャッシュを有効期限切れにします。ディレクトリエージェントが起動すると、持続的な格納領域 (通常はハードドライブ) からキャッシュを読み込み、アルゴリズムに従ってエントリを有効期限切れにします。新しい DA が起動したり、キャッシュが削除されると、DA はこの条件を検出して受信中のすべての SA に特別な通知を送信します。SA は、DA が直ちにキャッシュを作成できるようにローカルデータベースをダンプします。

ディレクトリエージェントが存在しない場合、UA は SA が応答できる一般的なマルチキャスト方式のクエリを使用し、DA がキャッシュを作成するのとほぼ同じ方法で、要求されたサービスのリストを作成します。このクエリによって返されるサービスのリストは、DA が提供するリストと比較すると不完全かつ局所的です。特に、多くのネットワーク管理者が使用するマルチキャスト方式でのフィルタ処理では、ブロードキャストおよびマルチキャストの対象がローカルサブネットのみに制限されるためです。

つまり、指定されたスコープに対してユーザエージェントが検索するものは、すべてディレクトリエージェントに依存します。

OpenText Service Location Providers

OpenText バージョンの SLP では、強力なサービスアドバタイジング環境を提供するため、SLP 標準が一部変更されていますが、そのためにスケーラビリティが一部制限されます。

たとえば、サービスアドバータイズのフレームワークの拡張性を改善するために、サブネット上でのブロードキャストまたはマルチキャストのパケット数が制限されます。SLP の仕様では、これを管理するために、ディレクトリエージェントのクエリに関してサービスエージェントおよびユーザエージェントに制限を加えています。必要なスコープに対応するための最初に検出されたディレクトリエージェントは、サービスエージェント (つまり結果的にローカルユーザエージェント) がそのスコープ上の将来の要求すべてに使用するエージェントとなります。

OpenText SLP を実装すると、クエリ情報の検索について既知のディレクトリエージェントがすべてスキャンされます。スキャンの所要時間は 300 ミリ秒とかなり長く、したがって、約 3 ～ 5 秒以内で 10 台のサーバしかスキャンできません。SLP がネットワーク上で正しく設定されている場合にはこのような検索の必要はありません。OpenSLP では、ネットワークが実際に SLP トラフィック用に設定されていると見なされます。OpenSLP の応答タイムアウト値は OpenText の SLP サービスプロバイダの応答タイムアウト値よりも大きい値です。ディレクトリエージェント数は、エージェントの情報が正確で完全であるかどうかに関係なく、最初に応答するディレクトリエージェントに制限されます。

ユーザエージェント

ユーザエージェントの物理形式は、アプリケーションにリンクされたスタティックライブラリまたはダイナミックライブラリです。ユーザエージェントにより、アプリケーションは SLP サービスに対して問い合わせることができます。

ユーザエージェントは、アルゴリズムに従って、クエリの送信先になるディレクトリエージェントのアドレスを取得します。指定したスコープの DA アドレスを取得すると、ユーザエージェントはそのスコープから応答がなくなるまで同じアドレスを使用し続けます。応答がなくなると、ユーザエージェントはそのスコープに対する別の DA アドレスを取得します。ユーザエージェントは、指定されたスコープのディレクトリエージェントのアドレスを次の方法で検索します。

1. 現在の要求のソケットハンドルが、指定したスコープの DA に接続されているかどうかを確認する。複数の要求の場合は、すでにキャッシュ化された接続がある可能性がある。
2. 指定したスコープと一致している DA の、既知のローカル DA キャッシュをチェックする。
3. 指定したスコープでローカル SA に対して DA を確認し、その後キャッシュに新しいアドレスを追加する。
4. 指定したスコープに一致する DA のネットワーク設定済みのアドレスを DHCP に問い合わせ、その後キャッシュに新しいアドレスを追加する。
5. 既知のポートで DA のディスカバリ要求をマルチキャストし、その後キャッシュに新しいアドレスを追加する。

スコープを指定しない場合、指定スコープは「デフォルト」になります。つまり、SLP 設定ファイルで静的に定義されたスコープがなく、クエリでスコープを指定していない場合は、使用されるスコープは「デフォルト」という単語になります。また、OpenText eDirectory では、登録の際にスコープが指定されないことに注意してください。ただし、OpenText eDirectory で使用されるスコープが常に「デフォルト」というわけではありません。スコープが静的に設定されている場合、そのスコープがすべてのローカル UA 要求および SA 登録に対して、指定したスコープがない場合のデフォルトのスコープになります。

サービスエージェント

サービスエージェントの物理形式は、ホストマシン上での個別のプロセスです。Windows の場合は、slpd.exe がローカルマシン上のサービスとして実行されます。ユーザエージェントは、既知のポート上のループバックアドレスにメッセージを送信することによって、ローカルサービスエージェントを問い合わせます。

サービスエージェントは、潜在 DA アドレスに DA 検出要求を直接送信することにより、ディレクトリエージェントおよびそれがサポートするスコープリストを検出してキャッシュします。DA 検出要求は、次の方法で送信されます。

1. 静的に設定された DA アドレスをすべてチェックする (その後 SA の既知の DA キャッシュに新しい DA アドレスを追加します)。
2. DHCP から DA とスコープのリストを要求する (その後 SA の既知の DA キャッシュに新しいリストを追加します)。

3. 既知のポートで DA の検出要求をマルチキャストする (その後 SA の既知の DA キャッシュに新しいポートを追加します)。
4. DA によって定期的にブロードキャストされた DA のアドバータイズパケットを受信する (その後 SA の既知の DA キャッシュに新しいアドバータイズパケットを追加します)。

ユーザエージェントは常に最初にローカルサービスエージェントに対して問い合わせます。ローカルサービスエージェントの応答によってユーザエージェントが次の検出段階を続行するかどうかが決定されるため、このことは重要な点です (DHCP のこのケースについては、「861 ページの 「ユーザエージェント」」の手順 3 および 4 を参照してください)。

環境設定パラメータ

SLP 環境設定パラメータは slp.conf ファイルに保存されます。このファイルは、UNIX および Linux プラットフォーム上では /etc にあり、Windows プラットフォーム上では %systemroot%/slp.conf です。これらのパラメータは、ネットワークに関する操作を調整するために変更できます。たとえば、次の各パラメータでは DA 検出が制御されます。

```
net.slp.useScopes = <comma-delimited scope list>
net.slp.DAAddresses = <comma-delimited address list>
net.slp.passiveDADetection = <"true" or "false">
net.slp.activeDADetection = <"true" or "false">
net.slp.DAActiveDiscoveryInterval = <0, 1, or a number of seconds>
```

useScopes オプションは、SA の通知先のスコープ、および、サービスまたはクライアントアプリケーションで作成された登録またはクエリに指定したスコープが存在しない場合に、クエリが作成されるスコープを示します。OpenText eDirectory は常にデフォルトのスコープに通知し、問い合わせを行うため、このリストが OpenText eDirectory のすべての登録およびクエリに対するデフォルトのスコープリストになります。

DAAddresses オプションはコンマで区切られた IP アドレスのリストで、アドレスは 10 進数とドットで表記されます。このアドレスが他のすべてに対して優先されます。設定された DA のこのリストが登録またはクエリのスコープをサポートしない場合、検出を無効にしていない限りは、SA および UA はマルチキャスト方式で DA を検出します。

passiveDADetection オプションのデフォルトは「TRUE」です。ディレクトリエージェントは、設定に応じて定期的にそれ自体の存在をサブネットの既知のポート上にブロードキャストします。これらのパケットは DAAdvert パケットと名付けられます。このオプションに「FALSE」を設定した場合、ブロードキャスト方式のすべての DAAdvert パケットは SA に無視されます。

activeDADetection オプションのデフォルトも「TRUE」です。この設定により、SA はすべての DA に対して、指示された DAAdvert パケットで応答するように、定期的にブロードキャスト方式で要求できます。指示されたパケットはブロードキャストではありませんが、この要求に対する応答では SA に直接送信されます。このオプションに「FALSE」を設定した場合、SA は定期的な DA の検出要求をブロードキャストしません。

DAActiveDiscoveryInterval オプションは try-state パラメータです。デフォルト値は 1 です。これは、初期化の際に、SA が DA の検出要求を 1 回送る設定であることを意味する特別な値です。このオプションに 0 を設定すると、activeDADetection オプションに「FALSE」を設定した場合と結果は同じです。その他の値は、検出をブロードキャストする間隔を秒数で表します。

このオプションを正しく使用すると、サービスアドバータイズに使用するネットワーク帯域幅を適切に設定できます。ただし、デフォルト設定は平均的なネットワークで拡張性を最適化するように設計されています。

注 : デフォルトでは、SLP に対して IPV4 プロトコルが有効になっており、IPV6 は無効になっています。IPV6 を有効にするには、slp.conf ファイル内の次の行のコメントを外します。

net.slp.useIPv6 = true

OpenSLP 2.0 が付属しているのは Windows だけであるため、この設定は Windows でのみ有効です。

---

slptool ユーティリティ

このユーティリティは、OpenSLP で提供されているコマンドラインユーティリティです。slptool は、サービスの登録または登録解除と、スコープ、サービスタイプ、属性、および使用可能なサービスの問い合わせに使用できます。

次に例を示します。

- サービスを登録するには、次の構文を使用します。

  構文 : slptool register url *[attrs]*

  ```
  slptool register service:myserv.x://myhost.com
  "(attr1=val1),(attr2=val2)"
  ```

- サービスを登録解除するには、次の構文を使用します。

  構文 : slptool deregister *url*

  ```
  slptool deregister service:myserv.x://myhost.com
  ```

- 使用可能なサービスを検索するには、次の構文を使用します。

  構文 : slptool findsrvs *service-type [filter]*

  ```
  slptool findsrvs service:myserv.x
  ```

  ```
  slptool findsrvs service:myserv.x "(attr1=val1)"
  ```

- 設定済みのスコープを検索するには、次の構文を使用します。

  構文 : slptool findscopes

D OpenText eDirectory での DNS の使用方法

クライアントがサーバに OpenText eDirectory ツリーに存在しない完全修飾名 (たとえば、admin.novell.novell_inc) の解決を要求する場合、または OpenText Identity Console for Linux や OpenText eDirectory のインストールアプリケーションなどのスタンドアロンアプリケーションを使用してツリー内の名前を解決する場合で、通信するサーバがまだ存在しない場合は、OpenText eDirectory ではサービスディスカバリプロトコルを使用して名前を解決します。サービスディスカバリプロトコルはネットワークアプリケーションのクラスで、分散コンポーネントを使用してネットワーク内の必要なサービスを検索して使用できます。

OpenText eDirectory は、従来 SAP および SLP を使用してネットワークサービスを検索し、通知してきました。DNS はディスカバリプロトコルとして OpenText eDirectory 8.7.1 で追加されました。この追加機能では、OpenText eDirectory で認識していないツリー名 (通信しているサーバにツリーのコピーが保持されていないか、スタンドアロンアプリケーションを使用しているため) を問い合わせる場合、検出を試みるマシンが、スタンドアロンアプリケーションを実行しているマシンであるか、OpenText Identity Console などの JClient アプリケーションを実行しているマシンであるか、またはサーバであるかに関係なく、OpenText eDirectory のディスカバリプロトコルを次の順序で使用します。

1. DNS (Domain Name System)
2. SLP (Service Location Protocol)
3. SAP (Service Advertising Protocol)

OpenText eDirectory では、DNS プロトコルを使用する場合、名前 (たとえば、「prod_server4.provo.novell.novell-_inc」などのサーバ名) が渡されるとそれを使用して、その名前全体の解決をそのままで試みます。次に、OpenText eDirectory では検出マシンの DNS 検索リストにそれぞれの名前を追加してから、マシンの DNS サーバにその名前のアドレスが存在するかどうかをその DNS サーバに問い合わせます。たとえば、検出マシンの DNS 検索リストに「dev.novell.com」および「test.novell.com」が含まれている場合、OpenText eDirectory では prod_server4.provo.novell.novell_inc.dev.novell.com および prod_server4.provo.novell.novell_inc.test.novell.com を検索します。

その後、OpenText eDirectory は渡された名前からコンポーネントを切り離します。たとえば、「prod_server4.provo.novell.novell_inc」を解決する場合、OpenText eDirectory は provo.novell.novell_inc、novell.novell_inc、novell_inc の順に試みます。OpenText eDirectory は、各々の異なる検索コンテキストに対して検索を行い、最終的にツリーのルートの単一コンポーネントに検索を試みます。クライアントは接続が成功するまで各アドレスを試行します。アドレスの試行には、DNS サーバから戻されたレコードの順序を使用します。検出を試みるコンピュータが OpenText eDirectory 8.7.1 以降を実行している限り、レプリカリング内のサーバがどのコードリビジョンを実行しても問題ありません。

OpenText eDirectory ツリー名は、クライアントが名前解決のために使用する DNS ドメインの下の A、AAAA、またはサービス (SRV) リソースレコードを使用する DNS に加えることをお勧めします。A または AAAA レコードを使用する場合、OpenText eDirectory サーバはデフォルトの 524 ポートで実行する必要があります。サーバが他のポートを使用する場合は、SRV レコードを使用します。

次のリソースレコードのサンプルでは、ツリー名は「novell_inc」で DNS 検索コンテキストは「provo.novell.com」です。

レコード	例
マシンは、	novell_inc.provo.novell.com.IN A 192.168.1.2
AAAA	novell_inc.provo.novell.com.IN AAAA 4321:0:1:2:3:4:567:89ab
SRV	_ldap._tcp.novell_inc.provo.novell.com.SRV 0 0 389 server1.novell_inc.provo.novell.com SRV 10 0 389 server2.novell_inc.provo.novell.com

冗長性を維持するため、または複数のホスト (レプリカリング内のサーバ) を A レコードに指定するためには、A レコードを 2 つ以上作成します。OpenText eDirectory は、それらすべてを検索します。A、AAAA、および SRV レコードの詳細については、「DNS resource records」を参照してください。

DNS サーバのレコードエントリを、対応するパーティションルートを保持している何かに指示させる必要はありません。検出するマシンがツリーを認識しているサーバと通信でき次第、ツリー全体を参照して名前を解決できます。たとえば、DNS に「novell_inc」を設定している場合、「novell_inc」ルートを保持するサーバのいずれにも設定する必要はありません。必要なのは「novell_inc」ツリー内の任意のサーバを参照することだけです。これは、ツリー内でそのサーバに接続できれば、そのサーバがツリー周辺でユーザを参照するためです。

E OpenText eDirectory での GSSAPI の設定

OpenText eDirectory の SASL-GSSAPI メカニズムを使用すれば、LDAP を通じて Kerberos チケットによって OpenText eDirectory から認証を受けることができます。OpenText eDirectory のユーザパスワードを入力する必要はありません。Kerberos チケットは、Kerberos サーバから認証を受けることによって取得する必要があります。

この機能は主に、Kerberos インフラストラクチャがすでに配置された環境がある LDAP アプリケーションユーザにとって便利です。このため、このようなユーザは、個別の LDAP ユーザパスワードを入力することなく、LDAP サーバへの認証を行うことができます。

SASL-GSSAPI の現在の実装は、RFC 2222 (http://www.ietf.org/rfc/rfc2222.txt?number=2222) に準拠しており、認証メカニズムとしては Kerberos v5 のみをサポートしています。

次のセクションでは、GSSAPI の設定方法と、OpenText eDirectory で Kerberos を使用して実行できるさまざまなタスクについて説明し、その他の有用な情報も提供します。

- 867 ページの 「概念」
- 868 ページの 「OpenText eDirectory での GSSAPI の動作」
- 869 ページの 「GSSAPI を設定するための前提条件」
- 872 ページの「SASL-GSSAPI メソッドを使用して設定された OpenText eDirectory ツリーをマージする」
- 872 ページの 「SASL-GSSAPI メソッドの管理」
- 874 ページの 「ログインシーケンスの作成」
- 874 ページの 「LDAP での SASL-GSSAPI の使用方法」
- 874 ページの 「エラーメッセージ」
- 874 ページの 「よく使用される用語」

概念

- 868 ページの 「Kerberos について」
- 868 ページの 「SASL について」
- 868 ページの 「GSSAPI について」

Kerberos について

Kerberos は、ネットワーク上でエンティティを認証する手段を提供する標準プロトコルです。このプロトコルは、信頼されるサードパーティのモデルに基づいています。このモデルでは、共有されるシークレットが必要で、対称型のキー暗号化が使用されます。

詳細については、RFC 1510 (http://www.ietf.org/rfc/rfc1510.txt?number=1510) を参照してください。

SASL について

SASL(Simple Authentication and Security Layer) は、認証の抽象化を行う層をアプリケーションに提供します。これは、認証モジュールをプラグインで接続できるフレームワークです。

詳細については、RFC 2222 (http://www.ietf.org/rfc/rfc2222.txt?number=2222) を参照してください。

GSSAPI について

GSSAPI(Generic Security Services Application Program Interface) は、API の標準セットを通して認証とその他のセキュリティサービスを提供します。さまざまな認証メカニズムがサポートされていますが、最も一般的なのは Kerberos v5 です。

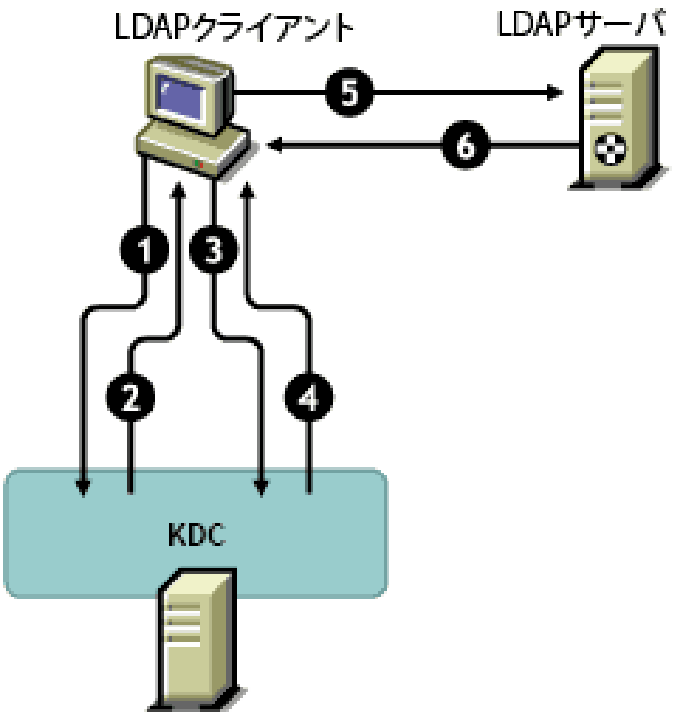
GSS API の形式に関する詳細については、RFC 1964 (http://www.ietf.org/rfc/rfc1964.txt?number=1964) を参照してください。

この SASL-GSSAPI 実装は、RFC 2222 (http://www.ietf.org/rfc/rfc2222.txt?number=2222) セクション 7.2 に規定されているものです。

OpenText eDirectory での GSSAPI の動作

次の図は、GSSAPI が LDAP サーバとともにどのように動作するかを示しています。

図 E-1 GSSAPI の動作



この図の数字は、それぞれ次のことを示しています。

1. OpenText eDirectory ユーザは、チケット認可チケット (TGT) と呼ばれる初期チケットの要求を、LDAP クライアントを通して Kerberos KDC (Key Distribution Center) サーバに送信します。

   Kerberos KDC としては、MIT または Microsoft* のいずれかのものを使用できます。
2. KDC は、TGT を送って LDAP クライアントに応答します。
3. LDAP クライアントは TGT を KDC に返信し、LDAP サービスチケットを要求します。
4. KDC は、LDAP サービスチケットを送って LDAP クライアントに応答します。
5. LDAP クライアントは LDAP サーバに対して ldap_sasl_bind を実行し、LDAP サービスチケットを送信します。
6. LDAP サーバは GSSAPI メカニズムを利用して LDAP サービスチケットを確認し、その結果に基づいて、ldap_sasl_bind が成功したか失敗したかを LDAP クライアントに返信します。

GSSAPI を設定するための前提条件

GSSAPI を設定するには、まず次の作業を行う必要があります。

- **SASL-GSSAPI メソッド :** SASL-GSSAPI メソッドをインストールします。Modular Authentication Services (「NMAS の機能」) の「ログインメソッドのインストール」セクションを参照してください。

注 : OpenText eDirectory SASL-GSSAPI メソッドは、Domain Services for Windows がインストールされた Open Enterprise Server (OES) バージョン 2 または 11 のインストール環境には機能しません。

SASL-GSSAPI がコンピュータにインストールされているかどうかを確認するには、次のように入力します。

```
ldapsearch -x -h osg-dt-srv9 -b " " -s base | grep -i sasl
```

SASL-GSSAPI がインストールされている場合、コマンドの出力は次のようになります。

```
supportedSASLMechanisms: NMAS_LOGIN
supportedSASLMechanisms: GSSAPI
```

- **キー配布センター (KDC: Key Distribution Center):** Kerberos KDC(MIT または Active Directory) をネットワーク上にインストールします。

  Microsoft KDC (Active Directory) を使用する場合は、Kerberos ツールをインストールしておく必要があります。これらのツールは Windows インストールの一部であり、Windows インストール CD 上の \support\tools\setup.exe(Windows XP) または \support\tools\suptools.msi(Windows 2003) からインストールできます。

- **時刻同期 :** このメソッドを機能させるために、NMAS クライアントマシン、NMAS サーバマシン、および KDC マシンの時刻を同期します。ネットワーク時刻の同期の詳細については、105 ページの 「ネットワーク時刻の同期」を参照してください。

- **Kerberos LDAP 拡張** Kerberos LDAP 拡張を追加します。詳細については、871 ページの「Kerberos の LDAP 拡張の追加」を参照してください。

重要

- Open Enterprise Server (OES) で、Domain Services for Windows または DNS サービスが設定されているサーバ上に Kerberos LDAP 拡張を追加しないでください。
- Kerberos の管理で収集される Kerberos 情報では、大文字と小文字が区別されるため、大文字と小文字を正確に指定する必要があります。

ネットワークの特性に関する前提

SASL-GSSAPI メカニズムは次の前提に基いて動作します。

- ネットワーク上のすべてのコンピュータの時刻がある程度正確に同期されている。言い換えると、システム時刻が 5 分以上異なる 2 台のコンピュータがネットワーク上に存在しない。
- MAN 環境や WAN 環境では時刻同期に関するこのような要件を満たすことが難しいため、SASL-GSSAPI メカニズムは主に LAN での使用を想定している。ただし、このメカニズムは LAN のみに限定されていない。
- Kerberos サーバと Kerberos 管理者を検証なしで無条件に信頼する。
- DoS(Denial-of-Service) 攻撃に対抗する手段にはならない。詳細については、RFC 1510 (http://www.ietf.org/rfc/rfc1510.txt?number=1510) を参照してください。

Kerberos の LDAP 拡張の追加

Kerberos の LDAP 拡張では、Kerberos キーの管理機能が提供されています。

Kerberos の LDAP 拡張を使用するには、C 言語用の LDAP ライブラリをインストールする必要があります。詳細については、「LDAP Libraries for C (https://www.microfocus.com/documentation/edirectory-developer-documentation/ldap-libraries-for-c/)」を参照してください。

Kerberos の LDAP 拡張を追加または削除するには、krbLdapConfig ユーティリティを使用します。スタンドアロンの OpenText eDirectory パッケージがディレクトリに抽出された場合、このファイルのパスは extracted_folder/nmas/NmasMethods/Novell/GSSAPI/Kerberos_ldap_extensions/Linux/krbLdapConfig です。

例 : /misc/eDir88/Linux/nmas/NmasMethods/Novell/GSSAPI/Kerberos_ldap_extensions/Linux/krbLdapConfig

Kerberos LDAP 拡張を追加するには、次の構文を使用します。

```
krbldapconfig {-i | -u} -D bind_DN [-w bind_DN_password] [-H ldap://[ldap host ip]:[ldap-port] [-e trusted_root_cert]
```

krbldapconfig ユーティリティのパラメータについて、次の表で説明します。

パラメータ	説明
-i	Kerberos LDAP 拡張を OpenText eDirectory に追加します。
-u	Kerberos LDAP 拡張を OpenText eDirectory から削除します。
-D *bind_fdn*	管理者または管理者と同等の権利を持つユーザの FDN を指定します。 この指定は、cn=admin,o=org の形式で指定する必要があります。
-w *bind_fdn_password*	バインド FDN(bind_fdn) のパスワードを指定します。
-h *ldap_server*	Kerberos LDAP 拡張をインストールする必要がある LDAP サーバのホスト名または IP アドレスを指定します。
-p *port*	LDAP サーバが動作しているポートを指定します。
-e *trusted_root_file*	SSL バインド用のルート認証局証明書のファイル名を指定します。 SSL ポートを使用する場合は、-e オプションを指定してください。 詳細については、872 ページの 「ルート認証局証明書のエクスポート」を参照してください。

注 : -h オプションを指定しなかった場合は、krbldapconfig の起動元のローカルホストの名前がデフォルトとして使用されます。

LDAP サーバポートとルート認証局証明書を指定しなかった場合は、ポート 389 がデフォルトで使用されます。

LDAP サーバポートを指定せずにルート認証局証明書を指定した場合は、ポート 636 がデフォルトで使用されます。

この拡張を追加するには次のように入力します。

```
krbldapconfig -i -D cn=admin,o=org -w password -h ldapserver -p 389
```

削除するには次のように入力します。

```
krbldapconfig -u -D cn=admin,o=org -w password -h ldapserver -p 389
```

重要 : インストールによる変更を有効にするには、LDAP サーバを手動でリフレッシュする必要があります。詳細については、422 ページの 「LDAP サーバをリフレッシュする」を参照してください。

ルート認証局証明書のエクスポート

1 OpenText Identity Console のホームページで、[**証明書管理**] タイル > [**ルート認証局管理**] の順にクリックして、[オブジェクトの変更] ページを開きます。

2 [**ルート認証局**] タブをクリックし、[**ルート認証局証明書 (Trusted Root Certificate)**] を選択して、証明書の詳細を表示します。

3 **エクスポート** をクリックします。

4 秘密鍵をエクスポートするかどうかを指定します。秘密鍵をエクスポートする場合、秘密鍵を保護するためのパスワードを指定する必要があります。

5 [**OK**] をクリックします。

6 [**Save the exported certificate**] をクリックします。

7 [**Save File (ファイルの保存)**] をクリックします。

8 **閉じる** をクリックします。

SASL-GSSAPI メソッドを使用して設定された OpenText eDirectory ツリーをマージする

ツリーのどちらか一方または両方が SASL-GSSAPI メソッドを使用して設定された 2 つのツリーをマージする場合は、ソースツリーにあるすべての Kerberos オブジェクトをターゲットツリー内で手動で作成する必要があります。

SASL-GSSAPI メソッドの管理

次の Kerberos 操作を実行できます。

- 873 ページの 「サービスプリンシパルの管理」

サービスプリンシパルの管理

このセクションでは次のことについて説明します。

- 873 ページの 「LDAP サーバ用のサービスプリンシパルの作成」
- 873 ページの 「OpenText eDirectory 用のサービスプリンシパルキーの抽出」

LDAP サーバ用のサービスプリンシパルの作成

KDC に付属する Kerberos 管理ツールを使用して、暗号化タイプと salt タイプをそれぞれ AES256-CTS と Normal に設定して OpenText eDirectory サービスプリンシパルを作成します。

プリンシパルの名前は、ldap/*MYHOST.MYDNSDOMAIN@REALMNAME* の形式にする必要があります。

MIT KDC を使用している場合は、次のようなコマンドを実行します。

kadmin:addprinc -randkey -e aes256-cts:normal ldap/server.novell.com@MITREALM

重要 : 作成したサービスプリンシパルのホスト名は、小文字でなければなりません。ホスト名が大文字の場合、認証は失敗します。たとえば、ホスト名が「myHost.com」の場合、LDAP サービスプリンシパルのホスト名の構文は、ldap/myhost.com@<realmname> のようになります。

ベストプラクティス

- すべてのキーのタイプを AES256 にすることをお勧めします。
- LDAP サービスプリンシパルキーを定期的に変更します。LDAP サービスプリンシパルキーを変更した場合は、必ず OpenText eDirectory 内のプリンシパルオブジェクトを更新してください。

OpenText eDirectory 用のサービスプリンシパルキーの抽出

KDC に付属する Kerberos 管理ツールを使用して、873 ページの 「LDAP サーバ用のサービスプリンシパルの作成」で作成した LDAP サービスプリンシパルのキーを抽出し、ローカルファイルシステムに保存します。この作業を行うには、Kerberos 管理者の協力が必要です。

MIT KDC を使用している場合は、次のようなコマンドを実行します。

kadmin: ktadd -k /*directory_path*/*keytabfilename* -e aes256-cts:normal ldap/server.novell.com@MITREALM

Microsoft KDC を使用している場合は、たとえば Active Directory で ldapMYHOST というユーザを作成してから、次のコマンドを実行します。

ktpass -princ ldap/MYHOST.MYDNSDOMAIN@MYREALM -mapuser ldapMYHOST -pass *mypassword* -out *MYHOST.keytab*

このコマンドを実行すると、プリンシパル (ldap/MYHOST.MYDNSDOMAIN@MYREALM) がユーザアカウント (ldapMYHOST) にマップされ、ホストプリンシパルのパスワードが mypassword に設定され、MYHOST.keytab ファイルに鍵が抽出されます。

ログインシーケンスの作成

ログインシーケンスの作成については、674 ページの 「ログインとポストログインのメソッドとシーケンスの管理」を参照してください。

LDAP での SASL-GSSAPI の使用方法

SASL-GSSAPI をインストールすると、SASL-GSSAPI が他の SASL 方式と共に rootDSE の supportedSASLMechanisms 属性に追加されます。

LDAP サーバは、SASL に問い合わせて環境設定時にインストールしたメカニズムを検索し、インストールされたメカニズムを自動でサポートします。また、supportedSASLMechanisms 属性を使って rootDSE で現在サポートされている SASL メカニズムをレポートします。

そのため、GSSAPI をインストールすると、GSSAPI がデフォルトのメカニズムになります。

ただし、明示的に SASL GSSAPI メカニズムを使用して LDAP の操作を行う場合は、コマンドラインで GSSAPI を指定できます。

たとえば、OpenLDAP で GSSAPI メカニズムを使用して検索を行うには、次のように入力します。

```
ldapsearch -H ldap://10.10.10.10:389  -b "" -s base "objectclass=*""
```

エラーメッセージ

SASL-GSSAPI のエラーメッセージは次の場所に記録されます。

- Linux の場合 : ndsd.log

詳細については、938 ページの 「OpenText eDirectory のエラーログを管理する」を参照してください。

よく使用される用語

次の表に、Kerberos と GSSAPI でよく使用される用語の定義を示します。

表 E-1 *Kerberos/GSSAPI の用語*

用語	定義
KDC (Key Distribution Center)	ユーザを認証してチケットを発行する Kerberos サーバ。
プリンシパル	KDC に登録されているエンティティ (ユーザまたはサービスインスタンス)。
レルム	複数の KDC によって管理されるドメインまたはプリンシパルのグループ。
サービスチケット (ST)	特定のサービスプリンシパルの共有キーを使って暗号化されたクライアント情報、サービス情報、およびセッションキーを格納しているレコード。
チケット認可チケット (TGT)	チケットのタイプの 1 つで、クライアントはそれを使用すると追加の Kerberos チケットを入手できる。

F セキュリティ上の考慮事項

この付録では、次のトピックについて説明します。

- 877 ページの 「LDAP バインド」
- 878 ページの 「Nessus のスキャン結果」

LDAP バインド

LDAP バインドは、安全な接続を使用して実行してください。常に SSL/TLS 接続を使用し、次の考慮事項に留意することをお勧めします。

- 回線上で転送されたキーが見破られることがあります。したがって、企業ネットワークを傍受またはパケットスニッフィングから物理的に保護する必要があります。
- 権限を持つ担当者のみが物理的にアクセスできる安全な場所にサーバを配置する必要があります。
- 企業のファイアウォールの外でユーザがこの製品を使用する場合は、VPN を使用する必要があります。
- サーバが企業ネットワークの外部からアクセスできる場合は、サーバへの直接アクセスを防ぐためにファイアウォールを設定する必要があります。
- 監視ログを定期的に確認する必要があります。
- 異なる管理任務を、別々の個人に割り当てる必要があります。管理の委任では、ディレクトリオブジェクトをきめ細かく制御できます。
- Kerberos の管理には特定の LDAP サーバを正しいサーバとして指定することをお勧めします。

重要: ユーザはサーバの IP アドレスではなく、DNS 名を使用して LDAP サーバにアクセスする必要があります。これは、IP アドレスから DNS 名への変換が保護されていないためです。

Nessus のスキャン結果

Nessus のポートスキャンによって、次の脆弱性が報告されました。

• 正しく設定されていない LDAP サーバによって、ユーザがその LDAP サーバに接続して情報を問い合わせることができる

説明 : OpenText eDirectory LDAP サーバでは、Null バインドがデフォルトで有効になっていますが、そのサーバ上で無効にすることができます。サーバのセキュリティを向上させるには、LDAP サーバのポート 389 に対して NULL バインドを無効にします。詳細については、407 ページの 「LDAP オブジェクトを環境設定する」を参照してください。

解決策 : Null バインドをサーバで無効にしてください。

• 正しく設定されていない LDAP サーバによって、ディレクトリベースが Null で設定される

説明 : ディレクトリ構造の事前の知識がなくても情報を取得することができます。匿名ユーザは、Null バインドを利用して「LdapMiner」などのツールによって LDAP サーバに問い合わせることができます。

解決策 : この状態が発生しないようにする方法はありませんが、このようなセキュリティ上の脅威は、Null バインドを無効にすることによって最小限に抑えることができます。

• リモートサービスで弱い SSL 暗号スイートの使用がサポートされている

説明 : リモートホストで弱い暗号化を提供しているか、または暗号化をまったく提供していない SSL 暗号の使用がサポートされています。

解決策 : 影響を受けるアプリケーションで弱い暗号を使用しないように再設定してください (可能な場合)。

• リモートのディレクトリサーバで情報が漏洩されている

説明 : このホストは OpenText eDirectory サーバで、PUBLIC オブジェクトに対するブラウズ権を保持しています。

解決策 : OpenText eDirectory を使用するアプリケーションが PUBLIC 権利の保持に依存しない場合、PUBLIC に与えられる権利を認証済みユーザ (ROOT) のみに割り当ててください。これが外部システムの場合は、インターネットからのポート 524 へのアクセスをブロックすることをお勧めします。ただし、パブリックブラウズ権を削除しても、ツリーおよびサーバ名にはアクセスできます。

•SSL 証明書が不明な認証局で署名されている

説明 : リモートホストの X.509 認定が、よく知られているパブリック認証局によって署名されていません。リモートホストが実稼働中の公開ホストの場合、これにより SSL の使用が無効にされます。これは、誰でもがその通信の途中で接続を確立してリモートホストに対して攻撃できるためです。

解決策：この状態は、そのサーバの証明書に署名した認証局の証明書が、クライアントアプリケーションの信頼された証明書ストアにない場合に発生します。そのサーバに対してよく知られている認証局から証明書を購入し、その証明書を展開します。また、そのサーバの証明書がツリーの組織の認証局、あるいは外部またはサードパーティの認証局から発行されていた場合は、アプリケーションの信頼された証明書ストアに、その認証局の証明書をインポートまたは追加します。

詳細については、712 ページの 「使用する認証局のタイプの決定」を参照してください。

G OpenText eDirectory イベントと CEF イベントのマッピング

このセクションでは、次の情報について説明します。

- 881 ページの 「OpenText eDirectory イベントと CEF イベントのマッピング」
- 886 ページの 「CEF イベント」

OpenText eDirectory イベントと CEF イベントのマッピング

表 G-1 は、対応する CEF イベントにマッピングされた OpenText eDirectory の内部イベントを一覧で示しています。各 OpenText eDirectory イベントとそれらの説明については、OpenText eDirectory サービスページを参照してください。CEF イベントについては、886 ページの 「CEF イベント」を参照してください。

表 G-1 *OpenText eDirectory イベントと対応付けられた CEF イベント*

イベントカテゴリ	CEF イベント	OpenText eDirectory イベント
セキュリティ	CONNECTION このイベントの例については、887 ページの 「接続」を参照してください。	DSE_CONNECTION
セキュリティ	LOGIN このイベントの例については、887 ページの 「ログイン」を参照してください。	DSE_LOGIN_EX DSE_NMAS_LOG_SRVR_BEGIN_LOGIN DSE_NMAS_LOG_FINISH_LOGIN_STATUS DSE_NMAS_LOG_SASL_MECHANISM_RESULT DSE_EBA_REQ_BA_MATERIAL
セキュリティ	LOGOUT このイベントの例については、887 ページの 「ログアウト」を参照してください。	DSE_LOGOUT

イベントカテゴリ	CEF イベント	OpenText eDirectory イベント
セキュリティ	ADD_MEMBER このイベントの例については、888 ページの「メンバーの追加」を参照してください。	DSE_ADD_VALUE
セキュリティ	DELETE_MEMBER このイベントの例については、888 ページの「メンバーの削除」を参照してください。	DSE_DELETE_VALUE
セキュリティ	INTRUDER_DETECTED このイベントの例については、888 ページの「不正侵入者が検出されました」を参照してください。	DSE_ADD_VALUE
セキュリティ	ACCOUNT_UNLOCK このイベントの例については、889 ページの「アカウントのロック解除」を参照してください。	DSE_DELETE_VALUE
セキュリティ	LOGIN_DISABLED このイベントの例については、889 ページの「ログインが無効化されました」を参照してください。	DSE_ADD_VALUE
セキュリティ	LOGIN_ENABLED このイベントの例については、889 ページの「ログインが有効化されました」を参照してください。	DSE_DELETE_VALUE
セキュリティ	ACL_CHANGED このイベントの例については、890 ページの「ACL が変更されました」を参照してください。	DSE_ADD_VALUE DSE_DELETE_VALUE
セキュリティ	CHANGE_SECURITY_EQUALS このイベントの例については、890 ページの「同等セキュリティの変更」を参照してください。	DSE_ADD_VALUE DSE_DELETE_VALUE
セキュリティ	VERIFY_PASSWORD このイベントの例については、891 ページの「パスワードの確認」を参照してください。	DSE_VERIFY_PASS

イベントカテゴリ	CEF イベント	OpenText eDirectory イベント
セキュリティ	AUDIT_CONFIG このイベントの例については、891 ページの 「監査の環境設定」を参照してください。	DSE_ADD_VALUE DSE_DELETE_VALUE
セキュリティ	CHANGE_PASSWORD このイベントの例については、891 ページの 「パスワードの変更」を参照してください。	DSE_CHGPASS DSE_NMAS_LOG_SET_PWD DSE_NMAS_LOG_SET_DIST_PWD DSDSE_NMAS_LOG_DELETE_PWD DSE_NMAS_LOG_DELETE_DIST_PWD DSE_NMAS_LOG_CHANGE_PWD DSE_NMAS_LOG_DELETE_ALL_LOGIN_SECRET DSE_NMAS_LOG_SET_LOGIN_SECRET DSE_NMAS_LOG_DELETE_LOGIN_SECRET
セキュリティ	CHANGE_LOGIN_CONFIG このイベントの例については、891 ページの 「ログイン環境設定の変更」を参照してください。	DSE_NMAS_LOG_SET_LOGIN_CONFIG DSE_NMAS_LOG_DELETE_LOGIN_CONFIG DSE_NMAS_LOG_DELETE_ALL_LOGIN_CONFIG
セキュリティ	QUERY_CREDENTIALS このイベントの例については、892 ページの 「資格情報の照会」を参照してください。	DSE_NMAS_LOG_GET_PWD_HISTORY DSE_NMAS_LOG_GET_PWD DSE_NMAS_LOG_GET_LOGIN_CONFIG DSE_NMAS_LOG_CHECK_PWD_SYNTAX_POLICY DSE_NMAS_LOG_GET_ALL_LOGIN_SECRET DSE_NMAS_LOG_GET_PWD_STATUS DSE_NMAS_LOG_GET_ALL_LOGIN_CONFIG DSE_NMAS_LOG_GET_DIST_PWD DSE_NMAS_LOG_GET_PWD_HISTORY
セキュリティ	IMPERSONATE このイベントの例については、892 ページの 「なりすまし」を参照してください。	DSE_IMPERSONATE
セキュリティ	AUTHENTICATE このイベントの例については、892 ページの 「認証」を参照してください。	DSE_AUTHENTICATE

イベントカテゴリ	CEF イベント	OpenText eDirectory イベント
オブジェクト	CREATE_OBJECT このイベントの例については、893 ページの「オブジェクトの作成」を参照してください。	DSE_CREATE_ENTRY DSE_ADD_ENTRY
オブジェクト	DELETE_OBJECT このイベントの例については、894 ページの「オブジェクトの削除」を参照してください。	DSE_REMOVE_ENTRY
オブジェクト	RENAME_OBJECT このイベントの例については、894 ページの「オブジェクトのリネーム」を参照してください。	DSE_RENAME_ENTRY
オブジェクト	MOVE_OBJECT このイベントの例については、894 ページの「オブジェクトの移動」を参照してください。	DSE_MOVE_SOURCE_ENTRY DSE_MOVE_DEST_ENTRY
オブジェクト	DSA_READ このイベントの例については、894 ページの「DSA の読み取り」を参照してください。	DSE_DSA_READ
オブジェクト	SEARCH このイベントの例については、895 ページの「検索」を参照してください。	DSE_SEARCH
属性	READ_ATTRIBUTE このイベントの例については、895 ページの「属性の読み込み」を参照してください。	DSE_READ_ATTR
属性	DELETE_ATTRIBUTE このイベントの例については、896 ページの「属性の削除」を参照してください。	DSE_DELETE_ATTRIBUTE
属性	ADD_VALUE このイベントの例については、896 ページの「値の追加」を参照してください。	DSE_ADD_VALUE
属性	DELETE_VALUE	DSE_DELETE_VALUE

イベントカテゴリ	CEF イベント	OpenText eDirectory イベント
属性	COMPARE_ATTRIBUTE_VALUE このイベントの例については、897 ページの 「属性値の比較」を参照してください。	DSE_COMPARE_ATTR_VALUE
LDAP	LDAP_BIND	DSE_LDAP_BIND
	LDAP_UNBIND	DSE_LDAP_UNBIND
	LDAP_CONNECTION	DSE_LDAP_CONNECTION
	LDAP_SEARCH	DSE_LDAP_SEARCH
	LDAP_ADD	DSE_LDAP_ADD
	LDAP_COMPARE	DSE_LDAP_COMPARE
	LDAP_MODIFY	DSE_LDAP_MODIFY
	LDAP_DELETE	DSE_LDAP_DELETE
	LDAP_MODIFY_DN	DSE_LDAP_MODDN
	LDAP_ABANDON	DSE_LDAP_ABANDON
	LDAP_EXTENDED_OPERATION	DSE_LDAP_EXTOP
	LDAP_SYSTEM_EXTENDED_OPERATION	DSE_LDAP_SYSEXTOP
	LDAP_PASSWORD_MODIFY	DSE_LDAP_PASSWDMODIFY
	MODIFY_LDAP_SERVER_CONFIGURATION	DSE_LDAP_MODLDAPSERVER
	UNKNOWN_LDAP_OPERATION	DSE_LDAP_UNKNOWNOP
	LDAP_BIND_RESPONSE	DSE_LDAP_BINDRESPONSE
	LDAP_SEARCH_RESPONSE	DSE_LDAP_SEARCHRESPONSE
	LDAP_SEARCH_ENTRY_RESPONSE	DSE_LDAP_SEARCHENTRYRESPONSE
	LDAP_ADD_RESPONSE	DSE_LDAP_ADDRESPONSE
	LDAP_COMPARE_RESPONSE	DSE_LDAP_COMPARERESPONSE
	LDAP_MODIFY_RESPONSE	DSE_LDAP_MODIFYRESPONSE
	LDAP_DELETE_RESPONSE	DSE_LDAP_DELETERESPONSE
	LDAP_MODIFY_DN_RESPONSE	DSE_LDAP_MODDNRESPONSE
	LDAP_EXTENDED_OPERATION_RESPONSE	DSE_LDAP_EXTOP_RESPONSE

イベントカテゴリ	CEF イベント	OpenText eDirectory イベント
EBA	MODIFY_SERVICE_CONFIG このイベントの例については、897 ページの 「サービスの環境設定の変更」を参照してください。	DSE_EBA_ISSUE_NCPCA_CERT DSE_EBA_REVOKE_NCPCA_CERT DSE_EBA_MOVE_EBA_CA DSE_EBA_ISSUE_CRL DSE_EBA_REQ_SERVER_BA_MATERIAL

CEF イベント

CEF イベントは、次のカテゴリに分類されます。

- 886 ページの 「セキュリティイベント」
- 893 ページの 「オブジェクトイベント」
- 895 ページの 「属性イベント」
- 897 ページの 「EBA イベント」

セキュリティイベント

このイベントセットは、OpenText eDirectory のセキュリティ操作の監査に適用できます。セキュリティ操作には、アクセスの付与や取り消し、ログイン、パスワード変更、照会があります。このイベントセットは、OpenText eDirectory システムに対する不正侵入者の検出にも役立ちます。

セキュリティイベントの例 :

このセクションには、次のセキュリティイベントの例が含まれています。

- 887 ページの 「接続」
- 887 ページの 「ログイン」
- 887 ページの 「ログアウト」
- 888 ページの 「メンバーの追加」
- 888 ページの 「メンバーの削除」
- 888 ページの 「不正侵入者が検出されました」
- 889 ページの 「アカウントのロック解除」
- 889 ページの 「ログインが無効化されました」
- 889 ページの 「ログインが有効化されました」
- 890 ページの 「ACL が変更されました」
- 890 ページの 「同等セキュリティの変更」

- 891 ページの 「パスワードの確認」
- 891 ページの 「監査の環境設定」
- 891 ページの 「パスワードの変更」
- 891 ページの 「ログイン環境設定の変更」
- 892 ページの 「資格情報の照会」
- 892 ページの 「なりすまし」
- 892 ページの 「認証」

注 : 次の各セクションで紹介する例は、参考のためにのみ示しています。

接続

[接続] をクリックして、次の例に示すように、システムコンポーネント間で通信チャネルが作成されるときにイベントを生成します。

```
Oct 31 17:00:22 NetIQ
CEF:0|NetIQ|eDirectory|9.1|CEF0B035E|CONNECTION|0|dvc=164.99.179.194
dvchost=SLES12SP2-194 rt=Oct 31 2017 17:00:22 dtz=IST
sourceServiceName=CN\=SLES12SP2-194,OU\=server,OU\=co,O\=in
sproc=eDirectory#DS src=164.99.179.164 spt=23017 duser=CN\=SLES12SP2-
194,OU\=server,OU\=co,O\=in cn1Label=Connection ID cn1=246358976
cn2Label=Created(1)/Terminated(0) cn2=1 cs1Label=Client Address
cs1=164.99.179.164:23017 cs2Label=Module cs2=LDAP Server cs3Label=Tree
Name cs3=TEST-CEF-AGN cs4Label=Correlation ID cs4=eDirectory#4294967295#
flexString2Label=SubEvent flexString2=DSE_CONNECTION cat=Security reason=0
outcome=Success
```

ログイン

[ログイン] をクリックして、新しいセッションが作成されるときにイベントを生成します。たとえば、OpenText eDirectory システムにログインする場合。

```
Oct 31 17:00:22 NetIQ
CEF:0|NetIQ|eDirectory|9.1|CEF0B035C|LOGIN|1|dvc=164.99.179.194
dvchost=SLES12SP2-194 rt=Oct 31 2017 17:00:22 dtz=IST
sourceServiceName=CN\=SLES12SP2-194,OU\=server,OU\=co,O\=in
sproc=eDirectory#NMAS src=164.99.179.164 spt=59737
suser=CN\=admin,OU\=novell,OU\=co,O\=in
duser=CN\=admin,OU\=novell,OU\=co,O\=in cs1Label=Client Address
cs1=164.99.179.164:59737 cs2Label=Class Name cs2=User cs3Label=Tree Name
cs3=TEST-CEF-AGN cs4Label=Correlation ID cs4=nmas#262183# cs6Label=Server
Name cs6=CN\=SLES12SP2-194,OU\=server,OU\=co,O\=in flexString1Label=Login
Method flexString1=0 flexString2Label=SubEvent
flexString2=DSE_NMAS_LOG_FINISH_LOGIN_STATUS flexNumber2Label=Grouping
flexNumber2=386 cat=Security reason=0 outcome=Success
```

ログアウト

[ログアウト] をクリックして、既存のセッションが停止するときにイベントを生成します。たとえば、OpenText eDirectory システムからログアウトする場合。

```
Jan 09 18:34:15 eDirectory
CEF:0|NetIQ|eDirectory|9.1|CEF0B0303|LOGOUT|1|dvc=164.99.179.194
dvchost=SLES12SP2-194 rt=Nov 03 2017 13:10:32 dtz=IST
sourceServiceName=CN\=SLES12SP2-194,OU\=server,OU\=co,O\=in
sproc=eDirectory#DS src=164.99.44.5 spt=53738 suser=[Public]
duser=CN\=SLES12SP2-194,OU\=server,OU\=co,O\=in cs1Label=Client Address
cs1=164.99.44.5 cs2Label=Class Name cs2=User cs3Label=Tree Name cs3=TEST-
CEF-NOV3 cs4Label=Correlation ID cs4=eDirectory#17# cs6Label=Object DN
cs6=CN\=admin,OU\=novell,OU\=co,O\=in flexString2Label=SubEvent
flexString2=DSE_LOGOUT flexNumber2Label=Grouping flexNumber2=127
cat=Security reason=0 outcome=Success
```

メンバーの追加

[**メンバーの追加**] をクリックして、次の例に示すように、新しいユーザがグループに追加されるときにイベントを生成します。

```
Jan 09 18:34:15 eDirectory
CEF:0|eDirectory|eDirectory|9.1|CEF0B0336|ADD_MEMBER|1|dvc=164.99.179.156
dvchost=SLES12-SP3-156.labs.blr.novell.com rt=Jan 09 2018 18:34:15 dtz=IST
sourceServiceName=CN\=SLES12-SP3-156,OU\=lnx-
server,OU\=server,OU\=co,O\=in sproc=eDirectory#DS src=164.99.179.158
spt=54936 suser=CN\=admin,OU\=novell,OU\=co,O\=in duser=CN\=grp1,OU\=lnx-
users,OU\=novell,OU\=co,O\=in cs2Label=Class Name cs2=Group cs3Label=Tree
Name cs3=NEW-TREE-9th cs4Label=Correlation ID cs4=eDirectory#14#bc560efc-
53d4-4ad9-85b4-fc0e56bcd453 cs6Label=Member DN cs6=CN\=lynx-user,OU\=lnx-
users,OU\=novell,OU\=co,O\=in flexString2Label=SubEvent
flexString2=DSE_ADD_VALUE flexNumber2Label=Grouping flexNumber2=3676
cat=Security reason=0 outcome=Success
```

メンバーの削除

[**メンバーの削除**] をクリックして、次の例に示すように、ユーザがグループから削除されるときにイベントを生成します。

```
Jan 09 18:35:06 eDirectory
CEF:0|eDirectory|eDirectory|9.1|CEF0B0337|DELETE_MEMBER|1|dvc=164.99.179.1
56 dvchost=SLES12-SP3-156.labs.blr.novell.com rt=Jan 09 2018 18:35:06
dtz=IST sourceServiceName=CN\=SLES12-SP3-156,OU\=lnx-
server,OU\=server,OU\=co,O\=in sproc=eDirectory#DS src=164.99.179.158
spt=54936 suser=CN\=admin,OU\=novell,OU\=co,O\=in duser=CN\=grp1,OU\=lnx-
users,OU\=novell,OU\=co,O\=in cs2Label=Class Name cs2=Group cs3Label=Tree
Name cs3=NEW-TREE-9th cs4Label=Correlation ID cs4=eDirectory#14#9136617f-
4412-48da-bf33-7f6136911244 cs6Label=Member DN cs6=CN\=lynx-user,OU\=lnx-
users,OU\=novell,OU\=co,O\=in flexString2Label=SubEvent
flexString2=DSE_DELETE_VALUE flexNumber2Label=Grouping flexNumber2=3687
cat=Security reason=0 outcome=Success
```

不正侵入者が検出されました

[**不正侵入者が検出されました**] をクリックして、次の例に示すように、不正侵入者が検出されるときにイベントを生成します。

```
Jan 09 18:35:06 eDirectory
CEF:0|NetIQ|eDirectory|9.1|CEF0B0357|INTRUDER_DETECTED|5|dvc=164.99.179.19
4 dvchost=SLES12SP2-194 rt=Oct 17 2017 19:50:20 dtz=IST
sourceServiceName=CN\=SLES12SP2-194,OU\=server,OU\=co,O\=in
sproc=eDirectory#DS src=164.99.179.194 spt=0 suser=CN\=SLES12SP2-
194,OU\=server,OU\=co,O\=in
duser=CN\=raghu,OU\=lens,OU\=QA,OU\=HD,OU\=DSLR,OU\=SLR,OU\=digital,OU\=ca
mera,O\=sony,L\=tokyo,dc\=co,C\=jp cs1Label=Intruder Address cs1=TCP:
164.99.179.164:33584 cs2Label=Reset Time cs2=10/17/17 19:52:20
cs3Label=Tree Name cs3=TEST-CEF222 cs4Label=Correlation ID
cs4=eDirectory#0#349e5670-0b80-4c99-b7f0-70569e34800b cs6Label=Class
cs6=User flexString2Label=SubEvent flexString2=DSE_ADD_VALUE
flexNumber2Label=Grouping flexNumber2=102 cat=Security reason=0
outcome=Success
```

アカウントのロック解除

[**アカウントのロック解除**] をクリックして、次の例に示すように、ロックされたアカウントがロック解除されるときにイベントを生成します。

```
Jan 09 19:10:32 eDirectory
CEF:0|eDirectory|eDirectory|9.1|CEF0B035F|ACCOUNT_UNLOCK|2|dvc=164.99.179.
156 dvchost=SLES12-SP3-156.labs.blr.novell.com rt=Jan 09 2018 19:10:32
dtz=IST sourceServiceName=CN\=SLES12-SP3-156,OU\=lnx-
server,OU\=server,OU\=co,O\=in sproc=eDirectory#DS src=164.99.179.156
spt=0 suser=CN\=SLES12-SP3-156,OU\=lnx-server,OU\=server,OU\=co,O\=in
duser=CN\=rr,OU\=lnx-users,OU\=novell,OU\=co,O\=in cs2Label=Class Name
cs2=User cs3Label=Tree Name cs3=NEW-TREE-9th cs4Label=Correlation ID
cs4=eDirectory#0#ad3a0226-764e-488c-b90a-26023aad4e76
flexString2Label=SubEvent flexString2=DSE_DELETE_VALUE
flexNumber2Label=Grouping flexNumber2=122 cat=Security reason=0
outcome=Success
```

ログインが無効化されました

[**ログインが無効化されました**] をクリックして、次の例に示すように、ユーザアカウントが無効にされるときにイベントを生成します。

```
Jan 09 18:18:48 eDirectory
CEF:0|eDirectory|eDirectory|9.1|CEF0B0356|LOGIN_DISABLED|2|dvc=164.99.179.
156 dvchost=SLES12-SP3-156.labs.blr.novell.com rt=Jan 09 2018 18:18:48
dtz=IST sourceServiceName=CN\=SLES12-SP3-156,OU\=lnx-
server,OU\=server,OU\=co,O\=in sproc=eDirectory#DS src=164.99.179.158
spt=54936 suser=CN\=admin,OU\=novell,OU\=co,O\=in duser=CN\=lynx-
user1,OU\=lnx-users,OU\=novell,OU\=co,O\=in cs2Label=Class Name cs2=User
cs3Label=Tree Name cs3=NEW-TREE-9th cs4Label=Correlation ID
cs4=eDirectory#14#f04b6deb-df9b-4f4b-a8e8-eb6d4bf09bdf
flexString2Label=SubEvent flexString2=DSE_ADD_VALUE
flexNumber2Label=Grouping flexNumber2=100 cat=Security reason=0
outcome=Success
```

ログインが有効化されました

[**ログインが有効化されました**] をクリックして、次の例に示すように、ユーザアカウントが有効にされるときにイベントを生成します。

```
Jan 09 18:18:56 eDirectory
CEF:0|eDirectory|eDirectory|9.1|CEF0B0355|LOGIN_ENABLED|2|dvc=164.99.179.1
56 dvchost=SLES12-SP3-156.labs.blr.novell.com rt=Jan 09 2018 18:18:56
dtz=IST sourceServiceName=CN\=SLES12-SP3-156,OU\=lnx-
server,OU\=server,OU\=co,O\=in sproc=eDirectory#DS src=164.99.179.158
spt=54936 suser=CN\=admin,OU\=novell,OU\=co,O\=in duser=CN\=lynx-
user1,OU\=lnx-users,OU\=novell,OU\=co,O\=in cs2Label=Class Name cs2=User
cs3Label=Tree Name cs3=NEW-TREE-9th cs4Label=Correlation ID
cs4=eDirectory#14#f99f0883-251e-424e-a724-83089ff91e25
flexString2Label=SubEvent flexString2=DSE_DELETE_VALUE
flexNumber2Label=Grouping flexNumber2=107 cat=Security reason=0
outcome=Success
```

ACL が変更されました

[ACL が変更されました] をクリックして、次の例に示すように、オブジェクトの ACL が変更されたときにイベントを生成します。

```
Jan 09 18:04:56 eDirectory
CEF:0|eDirectory|eDirectory|9.1|CEF0B0354|ACL_CHANGED|3|dvc=164.99.179.156
dvchost=SLES12-SP3-156.labs.blr.novell.com rt=Jan 09 2018 18:04:56 dtz=IST
sourceServiceName=CN\=SLES12-SP3-156,OU\=lnx-
server,OU\=server,OU\=co,O\=in sproc=eDirectory#DS src=164.99.179.158
spt=52120 suser=CN\=admin,OU\=novell,OU\=co,O\=in duser=CN\=lynx-
user,OU\=lnx-users,OU\=novell,OU\=co,O\=in cn1Label=ACL Added cn1=1
cs1Label=Value cs1=Entry ID: .CN\=lynx-user.OU\=lnx-
users.OU\=novell.OU\=co.O\=in.T\=NEW-TREE-9th., Attribute ID: [All
Attributes Rights], Privileges: Attribute Read cs2Label=Class Name cs2=User
cs3Label=Tree Name cs3=NEW-TREE-9th cs4Label=Correlation ID
cs4=eDirectory#18#c4f344f7-db17-4366-8a19-f744f3c417db cs6Label=Trustee
cs6=CN\=lynx-user,OU\=lnx-users,OU\=novell,OU\=co,O\=in
flexString2Label=SubEvent flexString2=DSE_ADD_VALUE
flexNumber2Label=Grouping flexNumber2=83 cat=Security reason=0
outcome=Success
```

同等セキュリティの変更

[同等セキュリティの変更] をクリックして、次の例に示すように、オブジェクトで同等セキュリティが変更されるときにイベントを生成します。

```
Jan 09 18:29:38 eDirectory
CEF:0|eDirectory|eDirectory|9.1|CEF0B0341|CHANGE_SECURITY_EQUALS|3|dvc=164
.99.179.156 dvchost=SLES12-SP3-156.labs.blr.novell.com rt=Jan 09 2018
18:29:38 dtz=IST sourceServiceName=CN\=SLES12-SP3-156,OU\=lnx-
server,OU\=server,OU\=co,O\=in sproc=eDirectory#DS src=164.99.179.156
spt=0 suser=CN\=SLES12-SP3-156,OU\=lnx-server,OU\=server,OU\=co,O\=in
duser=CN\=raghu,OU\=lnx-users,OU\=novell,OU\=co,O\=in cn1Label=Add/Remove
cn1=1 cs2Label=Class Name cs2=User cs3Label=Tree Name cs3=NEW-TREE-9th
cs4Label=Correlation ID cs4=eDirectory#0#6d1355d0-0401-4858-8475-
d055136d0104 cs6Label=Equivalent DN cs6=CN\=grp,OU\=novell,OU\=co,O\=in
flexString2Label=SubEvent flexString2=DSE_ADD_VALUE
flexNumber2Label=Grouping flexNumber2=3639 cat=Security reason=0
outcome=Success
```

パスワードの確認

[**パスワードの確認**] をクリックして、アカウントパスワードが確認されるときにイベントを生成します。

監査の環境設定

[**監査の環境設定**] をクリックして、次の例に示すように、監査サービスを制御するパラメータが変更されるときにイベントを生成します。

```
Jan 09 18:27:12 eDirectory
CEF:0|eDirectory|eDirectory|9.1|CEF0B0006|AUDIT_CONFIG|2|dvc=164.99.179.15
6 dvchost=SLES12-SP3-156.labs.blr.novell.com rt=Jan 09 2018 18:27:12
dtz=IST sourceServiceName=CN\=SLES12-SP3-156,OU\=lnx-
server,OU\=server,OU\=co,O\=in sproc=eDirectory#DS src=164.99.179.160
spt=54980 suser=CN\=srv-160,OU\=server,OU\=co,O\=in duser=CN\=SLES12-SP3-
156,OU\=lnx-server,OU\=server,OU\=co,O\=in cs1Label=Attribute Value
cs1=cefEvents\=ACL_CHANGED $$QUERY_CREDENTIALS cs2Label=Class Name cs2=NCP
Server cs3Label=Tree Name cs3=NEW-TREE-9th cs4Label=Correlation ID
cs4=eDirectory#16#8dcd3ede-baf8-4e71-9f1e-de3ecd8df8ba cs6Label=Attribute
Name cs6=cefConfiguration flexString2Label=SubEvent
flexString2=DSE_ADD_VALUE flexNumber2Label=Grouping flexNumber2=3631
cat=Security reason=0 outcome=Success
```

パスワードの変更

[**パスワードの変更**] をクリックして、次の例に示すように、アカウントパスワードが変更されるときにイベントを生成します。

```
Oct 31 17:06:11 NetIQ
CEF:0|NetIQ|eDirectory|9.1|CEF0290064|CHANGE_PASSWORD|1|dvc=164.99.179.194
dvchost=SLES12SP2-194 rt=Oct 31 2017 17:06:11 dtz=IST
sourceServiceName=CN\=SLES12SP2-194,OU\=server,OU\=co,O\=in
sproc=eDirectory#NMAS src=164.99.179.194 spt=0
suser=CN\=admin,OU\=novell,OU\=co,O\=in duser=raghu,novell,co,in
cs2Label=Class Name cs2=User cs3Label=Tree Name cs3=TEST-CEF-AGN
cs4Label=Correlation ID cs4=nmas#0# cs6Label=Server Name
cs6=CN\=SLES12SP2-194,OU\=server,OU\=co,O\=in flexString2Label=SubEvent
flexString2=DSE_NMAS_LOG_SET_LOGIN_SECRET flexNumber2Label=Grouping
flexNumber2=405 cat=Security reason=0 outcome=Success
```

ログイン環境設定の変更

[**ログイン環境設定の変更**] をクリックして、次の例に示すように、アカウントのログイン環境設定が変更されるときにイベントを生成します。

```
Nov 02 10:21:00 NetIQ
CEF:0|NetIQ|eDirectory|9.1|CEF0290061|CHANGE_LOGIN_CONFIG|1|dvc=164.99.179
.194 dvchost=SLES12SP2-194 rt=Nov 02 2017 10:21:00 dtz=IST
sourceServiceName=CN\=SLES12SP2-194,OU\=server,OU\=co,O\=in
sproc=eDirectory#NMAS src=164.99.179.194 spt=0
suser=CN\=admin,OU\=novell,OU\=co,O\=in duser=raghu,novell,co,in
cs2Label=Class Name cs2=User cs3Label=Tree Name cs3=TEST-CEF-AGN
cs4Label=Correlation ID cs4=nmas#0# cs6Label=Server Name
cs6=CN\=SLES12SP2-194,OU\=server,OU\=co,O\=in flexString2Label=SubEvent
flexString2=DSE_NMAS_LOG_SET_LOGIN_CONFIG flexNumber2Label=Grouping
flexNumber2=2034 cat=Security reason=0 outcome=Success
```

資格情報の照会

[**資格情報の照会**] をクリックして、次の例に示すように、特定のアカウントの資格情報が要求されるときにイベントを生成します。

```
Nov 02 10:21:00 NetIQ
CEF:0|NetIQ|eDirectory|9.1|CEF0290062|QUERY_CREDENTIALS|1|dvc=164.99.179.1
94 dvchost=SLES12SP2-194 rt=Nov 02 2017 10:21:00 dtz=IST
sourceServiceName=CN\=SLES12SP2-194,OU\=server,OU\=co,O\=in
sproc=eDirectory#NMAS src=164.99.179.194 spt=0
suser=CN\=admin,OU\=novell,OU\=co,O\=in duser=raghu,novell,co,in
cs2Label=Class Name cs2=User cs3Label=Tree Name cs3=TEST-CEF-AGN
cs4Label=Correlation ID cs4=nmas#0# cs6Label=Server Name
cs6=CN\=SLES12SP2-194,OU\=server,OU\=co,O\=in flexString2Label=SubEvent
flexString2=DSE_NMAS_LOG_GET_LOGIN_CONFIG flexNumber2Label=Grouping
flexNumber2=2035 cat=Security reason=0 outcome=Success
```

なりすまし

[**なりすまし**] をクリックして、次の例に示すように、アカウントのなりすましが発生するたびにイベントを生成します。

```
Nov 02 10:29:38 NetIQ
CEF:0|NetIQ|eDirectory|9.1|CEF0B0231|IMPERSONATE|1|dvc=164.99.179.194
dvchost=SLES12SP2-194 rt=Nov 02 2017 10:29:38 dtz=IST
sourceServiceName=CN\=SLES12SP2-194,OU\=server,OU\=co,O\=in
sproc=eDirectory#DS src=164.99.179.194 spt=56451
suser=CN\=admin,OU\=novell,OU\=co,O\=in
duser=CN\=raghu,OU\=novell,OU\=co,O\=in cs3Label=Tree Name cs3=TEST-CEF-
AGN cs4Label=Correlation ID cs4=eDirectory#10# cs6=CN\=SLES12SP2-
194,OU\=server,OU\=co,O\=in flexString2Label=SubEvent
flexString2=DSE_IMPERSONATE flexNumber2Label=Grouping flexNumber2=2048
cat=Security reason=0 outcome=Success
```

認証

[**認証**] をクリックして、次の例に示すように、ユーザがセッションを認証するときにイベントを生成します。

```
Nov 02 10:32:39 NetIQ
CEF:0|NetIQ|eDirectory|9.1|CEF0B035D|AUTHENTICATE|1|dvc=164.99.179.194
dvchost=SLES12SP2-194 rt=Nov 02 2017 10:32:39 dtz=IST
sourceServiceName=CN\=SLES12SP2-194,OU\=server,OU\=co,O\=in
sproc=eDirectory#DS src=164.99.179.194 spt=0
suser=CN\=impuser,OU\=novell,OU\=co,O\=in
duser=CN\=impuser,OU\=novell,OU\=co,O\=in cs2Label=Class Name cs2=User
cs3Label=Tree Name cs3=TEST-CEF-AGN cs4Label=Correlation ID
cs4=eDirectory#12# cs6Label=Server Name cs6=CN\=SLES12SP2-
194,OU\=server,OU\=co,O\=in flexString2Label=SubEvent
flexString2=DSE_AUTHENTICATE flexNumber2Label=Grouping flexNumber2=2058
cat=Security reason=0 outcome=Success
```

オブジェクトイベント

このイベントセットは、OpenText eDirectory のオブジェクト関連操作の監査に適用できます。オブジェクト操作には、オブジェクトの作成、削除、リネーム、移動、または照会があります。

オブジェクトイベントの例 :

このセクションには、次のオブジェクトイベントの例が含まれています。

- 893 ページの 「オブジェクトの作成」
- 894 ページの 「オブジェクトの削除」
- 894 ページの 「オブジェクトのリネーム」
- 894 ページの 「オブジェクトの移動」
- 894 ページの 「DSA の読み取り」
- 895 ページの 「検索」

注 : 次の各セクションで紹介する例は、参考のためにのみ示しています。

オブジェクトの作成

[**オブジェクトの作成**] をクリックして、次の例に示すように、OpenText eDirectory ツリーに新しいオブジェクトが作成されるときにイベントを生成します。

```
Oct 23 23:57:19 NetIQ
CEF:0|NetIQ|eDirectory|9.1|CEF0B0001|CREATE_OBJECT|0|dvc=164.99.179.60
dvchost=WIN-37D8M9SKD2U rt=Oct 23 2017 23:57:19 dtz=Pacific Daylight Time
sourceServiceName=CN\=WIN-37D8M9SKD2U-NDS,O\=novell sproc=eDirectory#DS
src=164.99.179.58 spt=52362 suser=CN\=Admin,O\=novell
duser=CN\=user001,O\=novell cs2Label=Class Name cs2=User cs3Label=Tree
Name cs3=TREE910W cs4Label=Correlation ID cs4=eDirectory#17#dc0fee11-5cd9-
47d4-b981-cdb8ecd47e07 flexString2Label=SubEvent
flexString2=DSE_CREATE_ENTRY flexNumber2Label=Grouping flexNumber2=677768
cat=Objects reason=0 outcome=Success
```

オブジェクトの削除

[**オブジェクトの削除**] をクリックして、次の例に示すように、OpenText eDirectory ツリーからオブジェクトが削除されるときにイベントを生成します。

```
Oct 24 00:02:35 NetIQ
CEF:0|NetIQ|eDirectory|9.1|CEF0B0309|DELETE_OBJECT|0|dvc=164.99.179.60
dvchost=WIN-37D8M9SKD2U rt=Oct 24 2017 00:02:35 dtz=Pacific Daylight Time
sourceServiceName=CN\=WIN-37D8M9SKD2U-NDS,O\=novell sproc=eDirectory#DS
src=164.99.179.58 spt=52362 suser=CN\=Admin,O\=novell
duser=CN\=user001,O\=novell cs2Label=Class Name cs2=User cs3Label=Tree
Name cs3=TREE910W cs4Label=Correlation ID cs4=eDirectory#17#2b97f69d-2984-
4f96-a83c-0b6c828bc462 flexString2Label=SubEvent
flexString2=DSE_REMOVE_ENTRY flexNumber2Label=Grouping flexNumber2=677993
cat=Objects reason=0 outcome=Success
```

オブジェクトのリネーム

[**オブジェクトのリネーム**] をクリックして、次の例に示すように、オブジェクトが名前変更されるときにイベントを生成します。

```
Oct 24 02:06:23 NetIQ
CEF:0|NetIQ|eDirectory|9.1|CEF0B0003|RENAME_OBJECT|0|dvc=164.99.179.60
dvchost=WIN-37D8M9SKD2U rt=Oct 24 2017 02:06:23 dtz=Pacific Daylight Time
sourceServiceName=CN\=WIN-37D8M9SKD2U-NDS,O\=novell sproc=eDirectory#DS
src=164.99.179.58 spt=55434 suser=CN\=Admin,O\=novell
duser=CN\=u1,O\=novell cs2Label=Class Name cs2=User cs3Label=Tree Name
cs3=TREE910W cs4Label=Correlation ID cs4=eDirectory#17#28250918-af9c-4098-
b56a-5757e456102a cs6Label=New Object DN cs6=CN\=u1changed,O\=novell
flexString2Label=SubEvent flexString2=DSE_RENAME_ENTRY
flexNumber2Label=Grouping flexNumber2=683314 cat=Objects reason=0
outcome=Success
```

オブジェクトの移動

[**オブジェクトの移動**] をクリックして、次の例に示すように、オブジェクトが移動されるときにイベントを生成します。

```
Oct 24 02:18:57 NetIQ
CEF:0|NetIQ|eDirectory|9.1|CEF0B0004|MOVE_OBJECT|0|dvc=164.99.179.60
dvchost=WIN-37D8M9SKD2U rt=Oct 24 2017 02:18:57 dtz=Pacific Daylight Time
sourceServiceName=CN\=WIN-37D8M9SKD2U-NDS,O\=novell sproc=eDirectory#DS
src=164.99.179.58 spt=55434 suser=CN\=Admin,O\=novell
duser=CN\=u1changed,O\=novell cs2Label=Class Name cs2=User cs3Label=Tree
Name cs3=TREE910W cs4Label=Correlation ID cs4=eDirectory#17#28789395-394f-
49d5-bb4e-b95410b0f9b5 cs6Label=New DN cs6=CN\=u1changed,OU\=org,O\=novell
flexString2Label=SubEvent flexString2=DSE_MOVE_SOURCE_ENTRY
flexNumber2Label=Grouping flexNumber2=683861 cat=Objects reason=0
outcome=Success
```

DSA の読み取り

[**DSA の読み取り**] をクリックして、次の例に示すように、オブジェクトが読み取られるときにイベントを生成します。

```
Oct 24 02:36:27 NetIQ
CEF:0|NetIQ|eDirectory|9.1|CEF0B0230|DSA_READ|0|dvc=164.99.179.60
dvchost=WIN-37D8M9SKD2U rt=Oct 24 2017 02:36:27 dtz=Pacific Daylight Time
sourceServiceName=CN\=WIN-37D8M9SKD2U-NDS,O\=novell sproc=eDirectory#DS
src=164.99.179.60 spt=20928 suser=CN\=WIN-37D8M9SKD2U-NDS,O\=novell
duser=CN\=WIN-37D8M9SKD2U-NDS,O\=novell cs2Label=Class Name cs2=NCP Server
cs3Label=Tree Name cs3=TREE910W cs4Label=Correlation ID cs4=eDirectory#1#
flexString2Label=SubEvent flexString2=DSE_DSA_READ cat=Objects reason=0
outcome=Success
```

検索

[**検索**] をクリックして、次の例に示すように、検索操作が要求されるときにイベントを生成します。

```
Oct 24 02:36:29 NetIQ
CEF:0|NetIQ|eDirectory|9.1|CEF0B033C|SEARCH|0|dvc=164.99.179.60
dvchost=WIN-37D8M9SKD2U rt=Oct 24 2017 02:36:29 dtz=Pacific Daylight Time
sourceServiceName=CN\=WIN-37D8M9SKD2U-NDS,O\=novell sproc=eDirectory#DS
src=164.99.179.60 spt=21184 suser=CN\=WIN-37D8M9SKD2U-NDS,O\=novell
duser=CN\=Security cn1Label=Scope cn1=2 cn2Label=Nodes To Search cn2=100
cs2Label=Class Name cs2=SAS:Security cs3Label=Tree Name cs3=TREE910W
cs4Label=Correlation ID cs4=eDirectory#2# flexString2Label=SubEvent
flexString2=DSE_SEARCH flexNumber2Label=Grouping flexNumber2=684639
cat=Objects reason=0 outcome=Success
```

属性イベント

このイベントセットは、OpenText eDirectory の属性関連操作の監査に適用できます。属性操作には、属性の作成、削除、リネーム、移動、または検索があります。

属性イベントの例 :

このセクションには、次の属性イベントの例が含まれています。

- 895 ページの 「属性の読み込み」
- 896 ページの 「属性の削除」
- 896 ページの 「値の追加」
- 896 ページの 「値の削除」
- 897 ページの 「属性値の比較」

注 : 次の各セクションで紹介する例は、参考のためにのみ示しています。

属性の読み込み

[**属性の読み込み**] をクリックして、次の例に示すように、オブジェクトの属性が読み込まれるときにイベントを生成します。

```
Oct 26 11:38:35 NetIQ
CEF:0|NetIQ|eDirectory|9.1|CEF0B0323|READ_ATTRIBUTE|0|dvc=164.99.179.60
dvchost=WIN-37D8M9SKD2U rt=Oct 25 2017 23:08:35 dtz=India Standard Time
sourceServiceName=CN\=WIN-37D8M9SKD2U-NDS,O\=novell sproc=eDirectory#DS
src=164.99.179.60 spt=18369 suser=CN\=WIN-37D8M9SKD2U-NDS,O\=novell
duser=CN\=WIN-37D8M9SKD2U-NDS,O\=novell cs2Label=Class Name cs2=NCP Server
cs3Label=Tree Name cs3=TREE910W cs4Label=Correlation ID cs4=eDirectory#1#
cs6Label=Attribute Name cs6=cefConfiguration flexString2Label=SubEvent
flexString2=DSE_READ_ATTR cat=Attributes reason=0 outcome=Success
```

属性の削除

[**属性の削除**] をクリックして、次の例に示すように、オブジェクトから属性が削除されるときにイベントを生成します。

```
Oct 24 22:54:36 NetIQ
CEF:0|NetIQ|eDirectory|9.1|CEF0B0009|DELETE_ATTRIBUTE|0|dvc=164.99.179.60
dvchost=WIN-37D8M9SKD2U rt=Oct 24 2017 22:54:36 dtz=Pacific Daylight Time
sourceServiceName=CN\=WIN-37D8M9SKD2U-NDS,O\=novell sproc=eDirectory#DS
src=164.99.179.60 spt=21184 suser=CN\=WIN-37D8M9SKD2U-NDS,O\=novell
duser=CN\=WIN-37D8M9SKD2U-NDS,O\=novell cs2Label=Class Name cs2=NCP Server
cs3Label=Tree Name cs3=TREE910W cs4Label=Correlation ID
cs4=eDirectory#2#a9ea8944-6a78-4a69-9c11-727635aa79e8 cs6Label=Attribute
Name cs6=Network Address flexString2Label=SubEvent
flexString2=DSE_DELETE_ATTRIBUTE flexNumber2Label=Grouping
flexNumber2=736694 cat=Attributes reason=0 outcome=Success
```

値の追加

[**値の追加**] をクリックして、次の例に示すように、属性に値が追加されるときにイベントを生成します。

```
Oct 24 02:38:12 NetIQ
CEF:0|NetIQ|eDirectory|9.1|CEF0B0006|ADD_VALUE|0|dvc=164.99.179.60
dvchost=WIN-37D8M9SKD2U rt=Oct 24 2017 02:38:12 dtz=Pacific Daylight Time
sourceServiceName=CN\=WIN-37D8M9SKD2U-NDS,O\=novell sproc=eDirectory#DS
src=164.99.179.60 spt=0 suser=CN\=WIN-37D8M9SKD2U-NDS,O\=novell
duser=.[Pseudo Server] cs1Label=Attribute Value cs1=720575940530274304
cs3Label=Tree Name cs3=TREE910W cs4Label=Correlation ID
cs4=eDirectory#0#f9787bd7-0541-47ca-9391-5a4bada90f02 cs6Label=Attribute
Name cs6=treeReferral flexString2Label=SubEvent flexString2=DSE_ADD_VALUE
flexNumber2Label=Grouping flexNumber2=684713 cat=Attributes reason=0
outcome=Success
```

値の削除

[**値の削除**] をクリックして、次の例に示すように、属性から値が削除されるときにイベントを生成します。

```
Oct 24 02:38:12 NetIQ
CEF:0|NetIQ|eDirectory|9.1|CEF0B0007|DELETE_VALUE|0|dvc=164.99.179.60
dvchost=WIN-37D8M9SKD2U rt=Oct 24 2017 02:38:12 dtz=Pacific Daylight Time
sourceServiceName=CN\=WIN-37D8M9SKD2U-NDS,O\=novell sproc=eDirectory#DS
src=164.99.179.60 spt=0 suser=CN\=WIN-37D8M9SKD2U-NDS,O\=novell
duser=.[Pseudo Server] cs1Label=Attribute Value cs1=720575940530274304
cs3Label=Tree Name cs3=TREE910W cs4Label=Correlation ID
cs4=eDirectory#0#1c411e7f-9657-474e-8e8e-80fc92921f96 cs6Label=Attribute
Name cs6=localReferral flexString2Label=SubEvent
flexString2=DSE_DELETE_VALUE flexNumber2Label=Grouping flexNumber2=684714
cat=Attributes reason=0 outcome=Success
```

属性値の比較

[**属性値の比較**] をクリックして、次の例に示すように、属性値が比較されるときにイベントを生成します。

EBA イベント

このイベントセットは、OpenText eDirectory の EBA 関連操作の監査に適用できます。EBA 関連の操作としては、EBA の有効化や無効化、サーバの追加や削除、サーバ間の EBACA の移動、さらには ndslogin さえもあり得ます。

EBA イベントの例 :

このセクションには、次の EBA イベントの例が含まれています。

- 897 ページの 「サービスの環境設定の変更」

注 : 次の各セクションで紹介する例は、参考のためにのみ示しています。

サービスの環境設定の変更

次の例に示されているように、OpenText eDirectory ツリーに EBA 関連の変更がある場合、[**サービスの環境設定の変更**] をクリックしてイベントを生成します。

```
Sep 17 16:34:56 eDirectory
CEF:0|eDirectory|eDirectory|9.2|CEF0B0503|MODIFY_SERVICE_CONFIG|1|dvc=164.
99.179.216 dvchost=SLESSP1-216 rt=Sep 17 2019 16:34:56 dtz=IST
sourceServiceName=CN\=SLESSP1-216,O\=novell sproc=eDirectory#DS
src=164.99.179.216 spt=0 cs1Label=CertAuthOldSvrAddress cs1=164.99.179.216
cs2Label=CertAuthNewSvrAddress cs2=164.99.179.216 cs3Label=Tree Name
cs3=TREE216 cs4Label=Correlation ID cs4=eDirectory#4294967295#
cs6Label=Server Name cs6=CN\=SLESSP1-216,O\=novell
flexString2Label=SubEvent flexString2=DSE_EBA_MOVE_EBA_CA
flexNumber2Label=Grouping flexNumber2=205 cat=Security reason=0
outcome=Success
```

H トラブルシューティング

- 899 ページの 「SNMP のトラブルシューティング」
- 904 ページの 「OpenText iMonitor のトラブルシューティング」
- 905 ページの 「破損通知のトラブルシューティング」
- 909 ページの 「OpenText eDirectory への移行」
- 916 ページの 「スキーマのトラブルシューティング」
- 917 ページの 「DSRepair のトラブルシューティング」
- 917 ページの 「レプリケーションのトラブルシューティング」
- 918 ページの 「クローン DIB に関する問題のトラブルシューティング」
- 918 ページの 「OpenText 公開鍵インフラストラクチャサービスのトラブルシューティング」
- 925 ページの 「Linux でのユーティリティのトラブルシューティング」
- 926 ページの 「NMAS のトラブルシューティング」
- 929 ページの 「ディレクトリサービスがロードされない場合の HTTPSTK へのアクセス」
- 931 ページの 「データ暗号化のトラブルシューティング」
- 934 ページの 「OpenText eDirectory 管理ツールボックス」
- 936 ページの 「SASL-GSSAPI に関する問題のトラブルシューティング」
- 938 ページの 「OpenText eDirectory のエラーログを管理する」
- 941 ページの 「その他」
- 950 ページの 「IPV6 に関する問題のトラブルシューティング」
- 950 ページの 「EBA のトラブルシューティング」

SNMP のトラブルシューティング

必要なトラップが生成されない

トラップは、対応するバーブの要求がサーバで受信された場合にのみ送信されます。それ以外の場合にトラップは送信されません。たとえば、ndsRemoveEntry (トラップ番号 108) の要求が送信された場合にのみ、ndsDeleteAttribute が送信されます。ただし、アプリケーションでは常に ACL が読み込まれ、そのユーザに削除操作を実行する十分な権利があるかどうかが確認されます。この場合、ndsDeleteAttribute トラップは生成されません。ただし、OpenText iMonitor を使用して特定のサーバのバーブ統計情報を表示できます。

すべてのイベントの発生時にトラップを取得するには、時間間隔を 0 に設定します。

トラップを有効にすると、失敗した場合にのみ送信できます。トラップを有効にすると、すべての状態でトラップを取得できます。

マスタエージェントが再起動される際に、ndssnmpsa を再起動する必要があります

ndssnmpsa を再起動するには、ndssnmpsa を停止した後に再度開始します。

ndssnmpsa を停止するには、次のように入力します。

Linux: /etc/init.d/ndssnmpsa stop

ndssnmpsa を開始するには、次のように入力します。

Linux: /etc/init.d/ndssnmpsa start

SNMP グループオブジェクト

SNMP グループオブジェクトのインストールが失敗した場合、サーバコンソールで次のコマンドを実行すると、この問題を修正できます。

ndsconfig *add -m snmp*

Windows サーバでの SNMP オブジェクト作成エラー

サポートされている Windows プラットフォームのサーバに OpenText eDirectory をインストールしている間に、SNMP グループオブジェクトの作成エラーが発生した場合は、SNMP グループオブジェクトを手動で作成する必要があります。SNMP オブジェクトを手動で作成する手順については、539 ページの第 18 章「OpenText eDirectory の SNMP サポート」を参照してください。

OpenText eDirectory SNMP 初期化コンポーネント。エラーコード : -255 または初期化エラー。エラーコード : -255

原因としては、OpenText eDirectory SNMP 環境設定ファイル内で hostname:port または IP_address:port を SERVER コマンドのパラメータとして指定しなかったことが考えられます。

OpenText eDirectory SNMP 環境設定ファイルは ndssnmp.cfg です。これは次のディレクトリにあります。

- Linux: /etc/opt/novell/eDirectory/conf/ndssnmp/
- Windows: *install_directory*\SNMP\

LDAP SNMP 統計が報告されない

匿名バインドが無効になっていると、LDAP SNMP 統計が報告されません。

この問題を解決するには、次の操作を行います。

1. 匿名バインドを許可します。
2. サブエージェントを起動します。
3. 匿名バインドを無効にします。

サブエージェントにアクセスする際のセグメント化失敗エラー

誤った OpenText eDirectory パスワードでユーザがサブエージェント (ndssnmpsa) を起動しようとすると、セグメント化失敗のエラーが発生します。

このエラーを回避するには、サブエージェントの起動時に、正しい OpenText eDirectory パスワードを使用します。

OpenText eDirectory 8.7.3 から OpenText eDirectory 9.0 にアップグレードした後に発生する問題

OpenText eDirectory 8.7.3 から OpenText eDirectory 9.0 にアップグレードした後に、次のエラーが発生する場合があります。

```
%%% Attempting to restart the NetIQ eDirectory SNMP subagent (ndssnmpsa)...
Starting NDS SNMP Subagent ...
Initialization failure. Error code : -255
Please Wait...
Done

%%% Unable to start ndssnmpsa... Please try starting it manually...
```

OpenText eDirectory 9.0 では、OpenText eDirectory が localhost をリスンしないため、このエラーが発生します。以前のバージョンの ndssnmp.cfg ファイルには、デフォルトで SERVER localhost が設定されていました。

このエラーを解決するには、ndssnmp.cfg ファイルを手動で編集し、監視対象となる OpenText eDirectory サーバのホスト名を指定する必要があります。

たとえば、ndssnmp.cfg ファイルに次のように入力します。

```
SERVER test-server
```

test-server は、デフォルトの NCP ポート (524) で実行されている OpenText eDirectory のホスト名です。OpenText eDirectory が別のポート (例 : 1524) で実行されている場合、エントリは次のようになります。

```
SERVER test-server:1524
```

NDS サブエージェントの起動時のエラー

サブエージェントが失敗して次のメッセージが表示される場合があります。

```
Unable to load library: libnetsnmp.so
```

この問題を解決するには、net-snmp ライブラリ (libnetsnmp.so) のメジャーバージョン番号を使用して、環境変数 SNMP_MAJOR_VERSION をエクスポートします。例 : 次のコマンドを使用できます。

```
export SNMP_MAJOR_VERSION=10
```

ndssnmpsa の再起動

Linux 上でマスタエージェントが再起動される際に、ndssnmpsa を再起動する必要があります。

ndssnmpsa を再起動するには、ndssnmpsa を停止した後に再度開始します。

ndssnmpsa を停止するには、次のコマンドを入力します。

/etc/init.d/ndssnmpsa stop

ndssnmpsa を開始するには、次のように入力します。

/etc/init.d/ndssnmpsa start

edir.mib のコンパイル

Windows の OpenText eDirectory MIB ファイル (*<eDirectoryInstallRootDir>*\snmp\edir.mib) では、コンパイル時に HP OpenView でいくつかのエラーおよび警告が出されます。これらのエラーは無視することができます。

SNMP 設定ファイルの変更

LDAP がクリアテキストモードで実行されるように設定されていない場合は、OpenText eDirectory SNMP サブエージェントを起動する前に、SNMP 環境設定ファイル (SSLKEY C:\Novell\nds\trust.der など) でルート認証局証明書ファイルの名前を指定する必要があります。

ndssnmp.cfg は、Windows 上の C:\novell\nds\snmp にあります。

新しいツリーをインストールした後の SNMP の使用

OpenText eDirectory 9.0 を初めてインストールする (新しいツリーを作成する) 際に、サーバにインストールされている Windows SNMP サービスに依存するサービスが 1 つ以上ある場合、OpenText eDirectory は SNMP サービスをシャットダウンできません。このような場合は、OpenText eDirectory をインストールした後に SNMP を使用することができません。

次の手順に従って、SNMP サービスを再起動してください。

1. **［スタート］**>**［設定］**>**［コントロールパネル］**>**［管理ツール］**>**［サービス］**の順にクリックします。
2. **［名前］**の一覧で**［SNMP サービス］**を右クリックし、**［停止］**をクリックします。
3. **［Yes to All］**をクリックします。
4. **［名前］**の一覧で**［SNMP サービス］**を右クリックし、**［開始］**をクリックします。

OpenText eDirectory のアンインストール時に SNMP をアンインストールする方法

Windows SNMP サービスがサーバにインストールされ、SNMP サービスに依存するサービスが 1 つ以上ある場合は、OpenText eDirectory をアンインストールしても C:\novell\nds フォルダ内の SNMP ファイルがすべて削除されるわけではありません。ただし、SNMP レジストリエントリの削除や、OpenText SNMP エージェントが DS および SNMP サービスによって行う設定解除プロセスなど、その他のアンインストールプロセスは正常に完了します。

アンインストールを完了するには、次の手順を実行します。

1 ［スタート］>［設定］>［コントロールパネル］>［管理ツール］>［サービス］の順にクリックします。
2 ［名前］の一覧で［SNMP サービス］を右クリックし、［停止］をクリックします。
3 ［Yes to All］をクリックします。
4 ［名前］の一覧で［SNMP サービス］を右クリックし、［開始］をクリックします。
5 C:\novell\nds フォルダに残っている SNMP ファイルを手動で削除します。

OpenText eDirectory をインストールすると Windows 2012 上の SNMP が停止する

OpenText eDirectory をインストールした後、SNMP が機能を停止し、次のエラーメッセージが表示されます。

SNMP subagent error -672

解決策 :

1 OpenText eDirectory をインストールした後で、SNMP サービスをインストールし、設定します。
2 OpenText eDirectory サーバで dssnmpsupport.exe を実行します。

注 : dssnmpsupport.exe は、MpsSvc service が OpenText eDirectory サーバで実行されている場合にのみ、適用してください。

EBA が有効になっていると SNMP サブエージェントのロードに失敗する

EBA が有効になっているサーバでは、SNMP エージェントのロードに失敗します。これは、SNMP エージェントをロードするコマンド (/etc/init.d/ndssnmpsa start) を入力し、資格情報を指定した後に発生します。返されるエラーメッセージは、ndssnmp.cfg の SERVER パラメータに定義されている値によって異なります。

この問題を解決するには :

- n4u.server.fips_tls を 0 に設定して、OpenText eDirectory サーバで FIPS モードを無効にします。
- OpenText eDirectory を再起動します。

SERVER パラメータは、サーバのホスト名または IP アドレスを指す必要があります。SERVER パラメータは、/etc/opt/novell/eDirectory/conf/ndssnmp ディレクトリ内の ndssnmp.cfg ファイルにあります。デフォルトでは、localhost に設定されていますが、OpenText eDirectory サーバの IP アドレスまたはホスト名に変更する必要があります。

OpenText iMonitor のトラブルシューティング

iMonitor を使用しての 2 バイト文字を含むオブジェクトの参照

OpenText iMonitor を使用して eDirectory ツリー内のオブジェクトをブラウズする際、名前に 2 バイト文字が含まれているオブジェクトについては、オブジェクトプロパティへのハイパーリンクが正しく設定されないことがあります。

単一のサーバツリーでのエージェントヘルスチェック

OpenText iMonitor のエージェントヘルスチェック機能を単一のサーバツリーで実行すると、［破損しやすいデータ］ステータスが原因で、［結果］カラムに［警告］アイコンが表示されます。これは、ツリーが正常でないということでも、エージェントヘルスチェックが設計どおりに機能していないということでもありません。破損しやすいデータとは、現在のところ少なくとも 1 つのレプリカにも同期されていないデータです。単一のサーバツリーは、その性質上、別の場所にデータのレプリカを作成していないため、このようなデータには常に重大な障害が発生する危険性があります。ハードディスクに障害が発生した場合、データを失うことになります。

単一のサーバツリーの破損しやすいデータまたは読み込み可能なレプリカ数に関するヘルスチェック警告を表示させたくない場合は、ndsimonhealth.ini ファイルを編集することにより、これらのヘルスチェックを無効にすることができます。ヘルスチェックを無効にするには、次のエントリを変更します。

perishable_data-active: OFF

および

ring_readable-Min_Marginal: 1 または ring_readable-active: OFF

この設定により、読み込み可能なレプリカ数および破損しやすいデータに関する警告が無効になります。

OpenText iMonitor レポートで 1 時間ごとのレコードが保存されない

OpenText iMonitor のカスタムレポート機能は、カスタムレポートを作成するときに、ユーザが指定した URL を保存対象のレポート (保存される HTML ファイル) に挿入するように設計されています。このため、保存された実行済みのカスタムレポートを開くと、カスタムレポートが実行された時点で URL によって取得されたデータではなく、アクティブな (現在の) データが表示されます。この問題は、OpenText iMonitor の今後のリリースで解決される予定です。

作成および変更のタイムスタンプ

Linux プラットフォームではファイルの作成時刻が保持されないため、OpenText iMonitor では作成時刻と変更時刻が両方とも同じように表示されます。

OpenText iMonitor で整列されていないレポート画面レイアウトの実行

ナビゲーションフレームとアシスタントフレームは Linux で 2 回表示されます。

この問題に対処するには、ページを更新します。

OpenText iMonitor にエラー -672 が表示される

- **Linux:** dsdump ツールが OpenText iMonitor と同時に実行されていると、OpenText iMonitor にエラー -672 が表示されます。この問題を解決するには、dsdump ツールを終了してから、iMonitor を開始します。
- **Windows:** dsbrowse ツールまたは dsedit ツールが OpenText iMonitor と同時に実行されていると、OpenText iMonitor にエラー -672 が表示されます。この問題を解決するには、dsbrowse ツールおよび dsedit ツールを終了してから、OpenText iMonitor を開始します。

OpenText iMonitor にエラー -702 が表示される

ユーザオブジェクトにグループエントリの ACL 権利を割り当てると、eDirectory サーバをアップグレードした後のエントリの検証中に、OpenText iMonitor にエラーメッセージが表示されます。

ndsimonhealth.conf ファイルの ValidACLFlags の値を手動で更新し、eDirectory サーバを再起動する必要があります。

タイムスタンプが 16 進数形式で表示される

Time 構文属性を 1970 年 1 月 1 日より前の値に設定すると、OpenText iMonitor はこの属性のタイムスタンプを標準の日時形式ではなく 16 進数形式で表示します。OpenText iMonitor では、1970 年 1 月 1 日以降の値をとるすべての属性を日時形式で表示します。

Internet Explorer 11 での OpenText iMonitor トレース設定の問題

OpenText iMonitor のトレース設定は、Internet Explorer 10 では使用できません。

この問題を回避するには、Internet Explorer 10 を互換モードで起動し、OpenText iMonitor のアドレスを［信頼済みサイト］のリストに追加してから、ブラウザを再起動します。

破損通知のトラブルシューティング

破損通知は、削除、移動、名前変更、復元などの操作中に、OpenText eDirectory が参照整合性を確保するためにオブジェクトに付加する操作属性です。たとえば、グループ A にユーザ B というメンバーが含まれているとき、ユーザ B が削除されると、ディレクトリは

自動的にグループ A からユーザ B への参照を削除します。OpenText eDirectory 9.0 では、削除、移動、および名前変更の操作によって生成される破損通知が、デフォルトで最適化されます。

注 : 破損通知を含むオブジェクトは、エージェントのアウトバウンド同期の実行時、およびインバウンド同期サイクルの最後に実行されるようにスケジュールされている破損通知処理の実行時に調査の対象となります。

破損通知には大きく分類して 3 つの種類があります。

- プライマリ破損通知には、停止 (0001)、復元 (0000)、移動 (0002)、新規 RDN(0005)、およびツリーの新規 RDN(0008) の各種類がありあます。
- セカンダリ破損通知は、一般的にプライマリ破損通知に関連付けられており、プライマリ破損通知で指定された操作の通知が必要なエージェントおよびパーティションを表します。セカンダリ破損通知には、バックリンク (0006)、使用中 (000C)、およびツリーの移動 (000a) の各種類があります。
- トラッキング破損通知には移動禁止 (0003)、古い RDN(0004)、およびツリーの古い RDN(0007) の各種類があります。

トラッキング破損通知以外の破損通知は、次の同期ステータスのセットを使用して移動する必要があります。

- 初期化ステータスまたは発行済み (0)
- 通知済み (1)
- パージ準備完了 (2)
- パージ可能 (4)

ステータスは破損通知属性のフラグフィールドで記録されます。破損通知が次のステータスに進む前に、現在のステータスは必ず実オブジェクトのすべてのレプリカに同期されます。リング内のすべてのレプリカが破損通知ステータスを与えられているかどうかを判断するために、遷移ベクトルからベクトルが計算されます。OpenText eDirectory 8.6 以降では、保存されていない破損通知ベクトルが使用されます。以前のバージョンの OpenText eDirectory では、パージベクトルが使用されます。破損通知の変更タイムスタンプ (MTS) が計測ベクトルよりも古い場合、担当サーバは該当する破損通知を次のステータスに進めることができます。

「バックリンク」のセカンダリ破損通知の場合、該当する破損通知を含むオブジェクトのマスタレプリカを持つエージェントがステータスを進めます。「使用中」のセカンダリ破損通知の場合、レプリカが存在している間は該当する破損通知を作成したエージェントがステータスを進めます。レプリカが存在しない場合、パーティションのマスタを保持しているエージェントが「使用中」の破損通知のステータスを進めます。「ツリーの移動」の破損通知の場合、ルートパーティションのマスタがステータスを進めます。

プライマリ破損通知は、すべてのセカンダリ破損通知が最後のステータスまで進められた後でのみ、ステータスを進めることができます。プライマリ破損通知が最後のステータスまで進んだ後で、そのステータスがリング内のすべてのサーバに同期されると、残っているのは属性を持たないオブジェクトであるオブジェクトハスクのみとなり、これらはシス

テムのパージプロセスによってパージされます。トラッキング破損通知は、プライマリ破損通知の削除の準備が完了した後か、Inhibit_move の場合はプライマリ破損通知がマスタレプリカの OBF_NOTIFIED ステータスに移動された後で削除されます。

破損通知の処理を担当するレプリカは、指定したパーティションがインバウンド同期サイクルを終了した後で、パーティションごとにスケジュールされているバックグラウンド処理 (破損通知処理) を実行します。パーティションにその他のレプリカがない場合、アウトバウンドレプリケーション処理がハートビート間隔でスケジュールされたままになります。その後、アウトバウンドレプリケーション処理によって破損通知処理が開始されます。破損通知処理は手動ではスケジュールできず、また、その必要もありません。同期化が実行されると、遷移ベクトルが更新され、パージベクトルおよび Obit ベクトルを進めます。これらのベクトルが進められると、破損通知のステータスを進めることができます。これと同時に、インバウンド同期に自動スケジュールが実行されると、破損通知処理サイクルが完了します。すなわち、破損通知処理の起動要因はオブジェクト同期です。

削除されたオブジェクトの場合、「停止」のプライマリ破損通知に関連するすべての破損通知が最後のステータス (パージ可能) まで進められ、そのステータスがすべてのレプリカに同期された後で、新しい処理がデータベースに残っているエントリハスクの削除を担当します。これらのハスクを削除するために、パージ処理が自動的に実行されます。パージ処理は、手動でスケジュールでき、259 ページの 「エージェントアクティビティの表示」で説明しているように、その自動スケジュール間隔を変更することもできます。

孤立した破損通知の解決

破損通知オブジェクトを表示した状態で、リング周辺の破損通知を比較しながらレプリカリングを調べます。

- 破損通知のコピーがないレプリカがあり、すべての属性値がパージ可能ではない場合、このオブジェクトはレプリカリング周辺で矛盾しており、破損通知が孤立していると考えられます。
- オブジェクトがすべてのレプリカに矛盾なく存在している場合、同期エラー、または破損通知処理にエラーが発生しているために次のステータスに進まない可能性があります。

この問題を回避するには次の手順に従ってください。

- **推奨される方法 :** レプリカリング内のサーバのいずれかに OpenText eDirectory 8.6 以降のバージョンを使用する場合、iMonitor のオブジェクトをブラウズし、［単一のエントリの送信 (Send Single Entry)］を選択します。これにより、その他のすべてのレプリカに信頼されていない送信が実行されます。
- **避けるべき方法 :** 孤立した破損通知のコピーを持つレプリカリングにあるすべてのサーバが OpenText eDirectory 8.6 よりも前のバージョンの場合、-a オプションを指定して DSBrowse をロードし、オブジェクトをブラウズして、エントリにタイムスタンプを設定します。これにより、このサーバに存在するオブジェクトを信頼されたコピーとして指定します。Novell では、実際にはオブジェクトを信頼されたオブジェクトとして指定することはお勧めしません。

外部参照の孤立した破損通知の解決

オブジェクトの破損通知がこのサーバに保存されていない (該当するオブジェクトが外部参照の) 場合、次を実行します。

- 一致する破損通知が実オブジェクトに含まれているかどうかをチェックします。一致する破損通知が含まれていない場合、この破損通知は孤立しています。
- 一致する破損通知が実オブジェクトに含まれている場合、実オブジェクトの破損通知の問題を解決してから、ExtRef パーティションにある破損通知の問題を処理してください。

この問題を回避するには次の手順に従ってください。

- **推奨されない方法 :** タイムスタンプオプションが選択されている DSRepair を実行します。
- **推奨されない方法 :** 実レプリカをサーバに移動し、使用可能な状態になってから破損通知が処理されるのを待ちます。破損通知が処理されたら、レプリカは削除してもかまいません。

破損通知に関する同期の問題の解決

破損通知が正確に同期するようにするには、次のことを実行してください。

- iMonitor のエージェント同期のページを使用して同期エラーをチェックおよび解決してください。
- 破損通知は、レプリカリングのコピーを保持するすべてのエージェントがステータスの変更を確認した後でのみステータスを変更できます。すべてのレプリカがデータを認識したことを確認するには、次のようないくつかの方法があります。

  破損通知を含むエントリを参照しながら、エントリ同期リンクをクリックします。すべてのレプリカに同期されていない属性がすべて表示されます。

  破損通知属性値の中で、最も古いタイムスタンプを検索します。検索されたタイムスタンプの時刻と現在時刻の差が、パーティション同期ページの最大リングデルタフィールドの間隔よりも大きい必要があります。

  遷移ベクトルを調査します。

破損通知に関するエラーの検索

エラーを検索するには、エージェントプロセスのステータス : 破損通知を検査してください。

- エージェントプロセスステータス : 破損通知で起こりうる一般的な問題には、次のものがあります。

  -625、-622、-634、および -635 の通信エラー。詳細については、サーバ情報レポートを参照してください。

-601 および -603 は、適切に削除されていないサーバ、またはサーバオブジェクトに不明なベースクラスが含まれるサーバを示します。

- このページに表示されるエラーは致命的なものではありません。そのパーティションで破損通知処理が次回実行されるときに、この操作が再試行されます。このページに表示された問題を解決して、再試行まで待機してください。

以前の操作

過去には、停止した破損通知を解決するためにいくつかの異なる手段をとりました。これらの手段の一部は、費用のかかるパーティション操作、または将来的に問題の原因となる可能性のあるドキュメント化されていない機能の使用に関連しています。

1 つ目に使用されていたのは、マスタを保持しているレプリカを切り替える方法です。マスタはさまざまなステータスを通じてバックリンクの破損通知の移動を担当しているエージェントなので、この手段が有効な場合もあります。レプリカに矛盾がありマスタが削除されたオブジェクトを保持していない場合、該当するマスタを、破損通知と一緒に削除されたエントリを保持していたエージェントに切り替えると、新規エージェントに破損通知のステータスを次に進めてパージすることのできるライセンスが与えられます。［Send Single Entry］を選択すると、より確実で危険性の低い方法で、レプリカの矛盾が原因で停止している破損通知の問題を解決します。

2 つ目に使用されていたのは、DSRepair を実行してすべての破損通知を削除するスイッチを使用する方法です。(DSRepair を起動して、停止している破損通知を解決するサードパーティ製のアプリケーションがあります。) この方法はお勧めできません。これらのスイッチを使用すると、このエージェントにあるすべての破損通知が削除されます。したがって停止していない破損通知も削除される可能性があり、レプリカの矛盾性がさらに深まり、より多くの停止している破損通知が作成される場合があります。これは分散された操作ではないので、停止している破損通知のあるすべてのサーバで DSRepair を実行する必要があります。この操作は、これらのサーバの 1 つで処理未了のまま削除されてしまうパーティションの破損通知を取得する可能性を高めます。処理未了のまま破損通知を削除すると、孤立した破損通知を新たに生み出す可能性があり、その結果として数年後にレプリカタイプの変更、新規レプリカの追加、またはその他のパーティション操作を実行したときに問題が生じる場合があります。

3 つ目に使用されていたのは、カスタムモード操作で DSBrowse を使用してエントリにタイムスタンプを設定するか、または RSRepair で -0T スイッチを使用してオブジェクトを信頼されたオブジェクトに指定する方法です。この操作によりエントリは信頼されたエントリに指定され、他のすべてのレプリカにこのエントリを同期します。その他のサーバで変更されたデータを失う可能性があるので、この操作の実行には細心の注意を払う必要があります。破損通知のクリーンアップの方法としては、なるべく使用しないことをお勧めします。

OpenText eDirectory への移行

Sun One スキーマの OpenText eDirectory への移行

Sun One スキーマを OpenText eDirectory に移行するには、次の手順に従ってください。

手順 1: スキーマキャッシュの更新操作を実行する

次のコマンドを使用して、スキーマの比較中に検出されたエラーをエラーファイルに書き込むことができます。

```
ice -e LDIF error file name -C -a -SLDAP -s Sun ONE server -p Sun ONE port
-DLDAP -s eDirectory server -p eDirectory port
```

次に例を示します。

```
ice -e err.ldf -C -a -SLDAP -s sun_srv1 -p sun_port1 -DLDAP -s edir_srv2 -
p edir_port2
```

スキーマの比較中に検出されたエラーはすべてエラーファイル (例では err.ldf) に書き込まれます。Root DSE を読み込むためにサーバから認証が要求されない限り、この操作を実行するためにログインする必要はありません。Microsoft アクティブディレクトリでは、Root DSE を読み込むための認証が要求されます。

手順 2: エラーを解決するためにエラー LDIF ファイルを訂正する

- Sun ONE では、OpenText eDirectory では定義されていないいくつかのスキーマ定義が公式に定義されています。これらの定義には、objectClasses、attributeTypes、ldapSyntaxes および subschemSubentry などの属性が含まれます。これらの定義は内部に存在し、スキーマにとって非常に重要です。したがって、これらの定義を変更することはできません。これらの定義を変更しようとすると、次のエラーが発生します。

  LDAP エラー : 53 (DSA が動作しません)

  これらの定義の参照が含まれるすべてのレコードに対して、次のエラーが表示されます。

  LDAP エラー : 16 : (該当する属性はありません)

  したがって、これらのオブジェクトへの参照が含まれるレコードまたはこれらの定義の変更を試行するレコードは、LDIF エラーファイル (例では err.ldf) にコメントとして記入されている必要があります。
- 一部の Sun ONE の objectClasses 定義にはネーミング属性がありません。これらの objectClasses を追加すると、OpenText eDirectory で次のエラーが発生することがあります。

  LDAP エラー : 80 (NDS エラー : ネーミングがあいまいです (-651)

  このエラーは、Sun ONE では eDirectory とは異なるネーミングルールの決定のメソッドが使用されていることが原因で発生します。

  これを解決するには、次の 3 つのいずれかのオプションを使用できます。

  オプション 1:

  エラーの原因となっている各 objectClasses を確認し、それぞれに有効なネーミング属性を追加します。

  次に例を示します。

  ネーミング属性［cn］をオブジェクトクラス netscapeMachineData に追加するには、X-NDS_NAMING フラグが包含されるように err.ldf ファイル内のエントリ (次の例で *強調表示*されている部分) を次のように変更します。

```
dn: cn=schemachangetype: modifyadd: objectClassesobjectClasses: (
2.16.840.1.113730.3.2.32 NAME 'netscapeMachineData'
    DESC 'iPlanet defined objectclass' SUP top STRUCTURAL MAY c'n '
X-NDS_NAMING 'cn' )-
```

オプション 2:

エラーの原因となっている各 objectClasses を確認し、該当するものをすべて AUXILIARY または ABSTRACT に変更します。

次に例を示します。

netscapeMachineData のオブジェクトクラス定義を “STRUCTURAL” から “AUXILIARY” に変更するには、次のように err.ldf ファイルのエントリ (次の例で *強調表示*されている部分) を変更します。

```
dn: cn=schemachangetype: modifyadd: objectClassesobjectClasses: (
2.16.840.1.113730.3.2.32 NAME 'netscapeMachineData'
   DESC 'iPlanet defined objectclass' SUP top AUXILIARY )-
```

netscapeMachineData のオブジェクトクラス定義を “STRUCTURAL” から “ABSTRACT” に変更するには、次のように err.ldf ファイルのエントリ (次の例で *強調表示*されている部分) を変更します。

```
dn: cn=schemachangetype: modifyadd: objectClassesobjectClasses: (
2.16.840.1.113730.3.2.32 NAME 'netscapeMachineData'
   DESC 'iPlanet defined objectclass' SUP top ABSTRACT )-
```

オプション 3:

OpenText eDirectory の Top の定義に cn を追加すると、すべての objectClasses の潜在的なネーミング属性になります。

cn を Top に追加するには 2 つの方法があります。

- **方法 1:**

  次のようなファイルを作成し、topsch.ldf と命名します。

  ```
  version : 1

  dn:cn=schema

  changetype :modify

  delete : objectclasses

  objectclasses : ( 2.5.6.0 NAME 'top' STRUCTURAL )

  -

  add:objectclasses

  objectclasses : (2.5.6.0 NAME 'top' STRUCTURAL MAY cn)
  ```

  次のインポート / エクスポート変換コマンドラインを使用します。

  ```
  ice -SLDIF -f LDIF_file_name -DLDAP -s eDirectory_server -p
  eDirectory_port -d eDirectory_Admin_DN -w eDirectory_password
  ```

  次に例を示します。

```
ice -SLDIF -f topsch.ldf -DLDAP -s edir_srv2 -p edir_port2 -d
cn=admin,o=org -w pwd1
```

- **方法 2:**

  1. OpenText Identity Console のホームページで、[**スキーマ管理**] タイル > [**属性**] の順にクリックします。
  2. [**属性の作成** +] をクリックします。
  3. [**属性名**] を指定し、[**構文**] を指定してから、[**属性フラグの追加**] を指定します。
  4. [**作成**] をクリックします。

- 一部の objectClass 定義には、必須属性リストとして userPassword が含まれる場合があります。これらの objectClasses を OpenText eDirectory に追加すると、次のエラーが発生します。

  LDAP エラー : 16 (該当する属性はありません)

  このエラーを解決するには、objectClass 定義を変更して、ndsLoginProperties から新しい objectClass を継承し、必須属性リストから userPassword 属性を削除します。

  次に例を示します。

  必須属性リストに userPassword が含まれる objectClass の場合 :

  ```
  version : 1
  dn: cn=schemaz
  changetype: modify
  add: objectClasses
  objectClasses: ( 0.9.2342.19200300.100.4.19 NAME 'simpleSecurityObject'
  DESC '
   Standard LDAP objectClass' SUP top STRUCTURAL MUST userPassword )
  ```

  次のように変更する必要があります (最終行の変更に注意してください):

  ```
  version : 1
  dn: cn=schema
  changetype: modify
  add: objectClasses
  objectClasses: ( 0.9.2342.19200300.100.4.19 NAME 'simpleSecurityObject'
  DESC '
   Standard LDAP objectClass' SUP (ndsLoginProperties $ top) STRUCTURAL)
  ```

手順 3: LDIF ファイルをインポートする

次の OpenText インポート / エクスポート変換コマンドを使用して変更されたスキーマ比較 LDIF ファイル (例では err.ldf) をインポートします。

```
ice -e error_file -SLDIF -f modified_LDIF_file -DLDAP -s eDirectory_server
-p eDirectory_port -d eDirectory_Admin_DN -w eDirectory_password
```

例 :

```
ice -e errors.ldf -SLDIF -f err.ldf -DLDAP -s edir_srv2 -p edir_port2 -d
cn=admin,o=org -w pwd1
```

ICE を使用した、Active Directory スキーマの OpenText eDirectory への移行

ICE を使用して、スキーマを Active Directory から OpenText eDirectory に移行すると、あいまいなネーミングエラー (-651) というエラーメッセージが表示され、スキーマの Computer オブジェクトクラスへの移行が失敗します。

これを解決するには、次の手順を実行します。

手順 1: スキーマキャッシュの更新操作を実行する

ICE を使用して、スキーマを Active Directory から OpenText eDirectory へ移行し、ICE のエラーログオプション (-e) が指定されていることを次のように確認します。

```
ice -e error_file -S ldap -s Active_Directory_server -p
Active_Directory_port -d Active_Directory_full_admin_context -w
Active_Directory_password -D ldap -s eDirectory_server -p eDirectory_port -
d eDirectory_full_admin_context -w eDirectory_password
```

例 :

```
ice -e err.ldf -S ldap -s activesrv1 -p activeport1 -d cn=admin,o=company -
w activepwd -D ldap -s edirsrv2 -p edirport2 -d cn=admin,o=company -w
edirpwd
```

手順 2: エラーを解決するためにエラー LDIF ファイルを訂正する

失敗したエントリは次のように err.ldf ファイルに記述されています。

```
dn: cn=schema

changetype: modify

delete: objectclasses

objectclasses: ( 2.16.840.1.113719.1.1.6.1.4 NAME 'computer' )

-

add: objectclasses

objectclasses: ( 2.16.840.1.113719.1.1.6.1.4 NAME 'computer' SUP (device $
 user ) STRUCTURAL MAY (operator $ server $ status $ cn $ networkAddress $
 local PolicyFlags $ defaultLocalPolicyObject $ machineRole $ location $
 netbootInitialization $ netbootGUID $ netbootMachineFilePath $ siteGUID $
 operatingSystem $ operatingSystemVersion $ operatingSystemServicePack $
 operatingSystemHotfix $ volumeCount $ physicalLocationObject $ dNSHostName
 $ policyReplicationFlags $ managedBy $ rIDSetReferences $ catalogs $
 netbootSIFFile $ netboot MirrorDataFile ) X-NDS_NOT_CONTAINER '1' X
 -NDS_NONREMOVABLE '1' X-NDS_NAME 'Computer' )

-
```

次の例で強調表示されている部分のように、エラーファイル (例では err.ldf) でこのエントリを編集して、Computer オブジェクトクラスの定義にあるスーペリアオブジェクトクラスのリストから user オブジェクトクラスを削除します。

```
dn: cn=schema
```

```
changetype: modify

delete: objectclasses

objectclasses: ( 2.16.840.1.113719.1.1.6.1.4 NAME 'computer' )

-

add: objectclasses

objectclasses: ( 2.16.840.1.113719.1.1.6.1.4 NAME 'computer' SUP device
 STRUCTURAL MAY (operator $ server $ status $ cn $ networkAddress $ local
 PolicyFlags $ defaultLocalPolicyObject $ machineRole $ location $
 netbootInitialization $ netbootGUID $ netbootMachineFilePath $ siteGUID $
 operatingSystem $ operatingSystemVersion $ operatingSystemServicePack $
 operatingSystemHotfix $ volumeCount $ physicalLocationObject $ dNSHostName
 $ policyReplicationFlags $ managedBy $ rIDSetReferences $ catalogs $
 netbootSIFFile $ netbootMirrorDataFile ) X-NDS_NOT_CONTAINER '1' X
 -NDS_NONREMOVABLE '1' X-NDS_NAME 'Computer' )
```

手順 3: LDIF ファイルをインポートする

次の ICE コマンドを使用して、変更されたエントリをインポートします。

```
ice -S ldif -f LDIF_file -D ldap -s Novell_eDirectory_server -p port_number
-d full_admin_context -w password
```

次に例を示します。

```
ice -S ldif -f err.ldf -D ldap -s edirsrv1 -p edirport1 -d
cn=admin,o=company -w pwd1
```

OpenLDAP から OpenText eDirectory への移行

OpenLDAP サーバから移行したデータは、MD5 パスワードを含んでいることがあります。この場合、適切な NMAS メソッドがインストールされていないと、アプリケーションが中断することがあります。NMAS メソッドの SimplePassword を、次に示すコマンドを使用して OpenText eDirectory 用にインストールする必要があります。

nmasinst -addmethod *admin_context treename configfile* -h *Hostname*:*port*-w *password*

例 : nmasinst -addmethod admin.novell eDir-Tree /Linux/eDirectory/nmas/NmasMethods/Novell/SimplePassword/config.txt -h eDir_srv:524 -w secret

OpenLDAP スキーマの OpenText eDirectory への移行

OpenLDAP スキーマを OpenText eDirectory に移行するには、次の手順に従ってください。

手順 1: スキーマキャッシュの更新操作を実行する

次のコマンドを使用して、スキーマの比較中に検出されたエラーをエラーファイルに書き込むことができます。

```
ice -e error_file -C -a -S ldap -s OpenLDAP_server -p Open_LDAP_port - D
ldap -s eDirectory_server -p eDirectory_port -d
eDirectory_full_admin_context -w eDirectory_password
```

次に例を示します。

```
ice -e err.ldf -C -a -SLDAP -s open_srv1 -p open_port1 -DLDAP -s edir_srv2
-p edir_port2 -d cn=admin,o=novell -w secret
```

スキーマの比較中に検出されたエラーはすべてエラーファイル (例では err.ldf) に書き込まれます。

手順 2: エラーを解決するためにエラー LDIF ファイルを訂正する

Open LDAP では、いくつかのスキーマ定義が公式に定義されています。これらの定義には、objectClasses、attributeTypes、ldapSyntaxes、および subschemSubentry などの属性が含まれます。これらの定義は内部に存在し、スキーマにとって非常に重要です。したがって、これらの定義を変更することはできません。これらの定義を変更しようとすると、次のエラーが発生します。

```
LDAP error : 53 (DSA is unwilling to perform)
```

これらの定義の参照が含まれるすべてのレコードに対して、次のエラーが表示されます。

```
LDAP error : 16 ( No such attribute )
```

したがって、これらのオブジェクトへの参照が含まれるレコードまたはこれらの定義の変更を試行するレコードは、LDIF エラーファイル (例では err.ldf) にコメントとして記入されている必要があります。

Open LDAP データの OpenText eDirectory への移行

次のコマンドを実行してデータを移行します。

```
ice -e error_data.ldif -SLDAP -s OpenLDAP_server -p OpenLDAP_port  -d
admin_context  -w password -t -b dc=blr,dc=novell,dc=com -F objectclass=* -
DLDAP -d  admin_context -w password  -l  -F
```

次に例を示します。

```
ice -e err_data.ldif -SLDAP -s open_srv1 -p open_port1  -d
cn=administrator,dc=blr,dc=novell,dc=com -w secret1 -t -b
dc=blr,dc=novell,dc=com -F objectclass=* -DLDAP -d cn=admin,o=novell -w
secret2 -l -F
```

オブジェクトによっては、前方参照やオブジェクトへの内部依存のために失敗するものもありますが、アプリケーションが中断することはほとんどありません。

移行後に PAM を OpenText eDirectory で動作可能にする

OpenLDAP から OpenText eDirectory に移行したら、PAM が OpenText eDirectory で動作するようにするため、いくつか変更する必要があります。

/etc/ldap.conf ファイルの変更

```
# The distinguished name to bind to the server with.
# Optional: default is to bind anonymously.
  binddn cn=admin,o=acme
  ...
# The credentials to bind with.
# Optional: default is no credential.
  bindpw secret
  ...
# The search scope.
  scope sub
  ...
# Filter to AND with uid=%s
  pam_filter objectclass=inetorgperson
  ...
# Remove old password first, then update in
# cleartext. Necessary for use with Novell
# Directory Services (NDS)
  pam_password nds
  ...
  ssl off
  ...
```

ディレクトリ内のデータの変更

この変更は、OpenLDAP のユーザオブジェクトでパスワードのハッシュアルゴリズムとして CRYPT を使用するシナリオにのみ行います。

OpenText Identity Console を使用して、規定値のある次の属性を、すべてのユーザオブジェクトを持つコンテナに追加します。

属性 : sasDefaultLoginSequence

値 : 単純パスワード

スキーマのトラブルシューティング

このセクションには、スキーマのトラブルシューティングについての情報が含まれています。

スキーマのトラブルシューティング

オブジェクトから補助クラスを分離しても、値が即時に削除されることはなく、存在しないというマークが付けられます。オブジェクトの実際の検証中に DRL プロセスがそれらの値をクリーンアップするまで、補助クラスはエントリに関連付けられています。

DRL はリソースを消費するバックグラウンドプロセスであるため、このクリーンアップの間は他の操作が遅くなります。クリーンアッププロセスに要する時間は、システム内に実際にあるオブジェクトおよび外部参照の数によって決まります。この処理は CPU とメモリに負荷がかかるため、頻繁に実行しないでください。デフォルトでは、Backlinker バックグラウンドプロセスは、ndsd が開始した 50 分後に実行され、その後 13 時間ごとに実行されます。

エントリから補助クラスをクリアするには 0 ～ 13 時間かかることがあり、それに加えてシステム内でそのエントリを処理するにも時間がかかります。

この問題を回避するには、DSTrace または iMonitor から Backlinker をトリガーして、補助クラスのエントリを削除します。

注 : オブジェクトが削除されると、値は即時にパージされます。この削除は他のバックグラウンドプロセスによって処理されるからです。

LDAP 構文マッピングのトラブルシューティング

DS 構文の中には、LDAP 構文に固有にマッピングされないものもあります。これは、OpenText eDirectory 9.1 以前で発生します。

この問題は、OpenText eDirectory 9.1 SP1 で修正されています。以前のマッピングに戻る場合は、NDSD_NLDAP_PRE911_SCHEMA 環境変数を任意の値に設定します。

DSRepair のトラブルシューティング

Linux 上の NFS マウントされた DIB での DSRepair の実行

Linux システム上の NFS マウントされた DIB で ndsrepair (DSRepair) 操作を実行しようとすると、-732 エラーまたは -6009 エラーが表示される可能性があります。

-R オプションを指定して DSRepair を実行したときにハングアップする

インデックス化された属性で暗号化属性を有効化した後に、-R オプションを指定して ndsrepair (DSRepair) を実行すると、ハングアップします。

レプリケーションのトラブルシューティング

OpenText eDirectory では、の強力なディレクトリサービス、およびレプリケーションによる耐障害性が提供されています。レプリケーションにより、OpenText eDirectory データベースまたはその一部のコピーを、複数のサーバで同時に保持できます。

OpenText Identity Console を使用した暗号化レプリケーションの設定

レプリカリング内のいずれかのサーバが停止している場合、OpenText Identity Console を使用して暗号化レプリケーションを設定することはできません。

暗号化複製を使用したツリーのマージに失敗する

暗号化複製が有効になっている場合、ツリーのマージに失敗します。マージを行う前に、各ツリーでセキュリティ保護された複製を無効にします。ツリーに EBA が有効なサーバが含まれる場合、ツリーのマージが成功することがあります。ただし、マージ後、ツリーが不安定になり、複製が適切ではないため認証が失敗する原因となります。

OpenText eDirectory のレプリカの問題からの回復

OpenText eDirectory パーティションのレプリカは、常に複数作成しておいてください。レプリカを複数作成しておくことで、あるレプリカがハードディスクの故障で破損したり失われたりした場合でも、OpenText Identity Console を使用してそのレプリカを削除し、破損していないレプリカから作成した新しいレプリカに置き換えることができます。

レプリケーションの削除の詳細については、『*OpenText eDirectory 9.0 管理ガイド*』の レプリカの管理を参照してください。

クローン DIB に関する問題のトラブルシューティング

クローン DIB の -601 および -603 エラーによる失敗

暗号化属性および暗号化複製がツリーレベルで有効になっている場合、クローン DIB は次のエラーにより失敗します。

- SAS の設定時に、ターゲットサーバ上でクローン DIB が -601 エラーにより失敗します。
- クローン DIB の後で、新たに作成されたクローンオブジェクトが -603 エラーにより失敗します。

これらの問題を回避するには、暗号化属性と暗号化複製を無効にします。

オフラインのバルクロード処理の直後にクローン DIB に失敗する

オフラインのバルクロード処理の直後にサーバのクローンを作成する場合、バルクロード処理がインデックスの無効化オプション付きで行われていると、クローン作成に失敗することがあります。

ただしこの問題は、バルクロード処理の完了後数時間以内に dibclone を実行した場合は発生しません。

暗号化複製機能を有効にしたクローン作成における問題

ソースサーバで暗号化複製機能を有効にしてクローンを作成する際に、クローン作成されたサーバを一時的に除外するように ER ポリシーを変更します。この設定は、クローン作成されたサーバの設定が完了した後で変更できます。

OpenText 公開鍵インフラストラクチャサービスのトラブルシューティング

PKI 操作が機能しない

OpenText Identity Console で PKI 操作が機能しない場合、OpenText PKI サービスが Linux で実行されていないことが考えられます。「npki - 1」と入力して PKI サービスを開始してください。

証明書を作成できない場合は、NICI モジュールが正しくインストールされているか確認する必要があります。「サーバ上の NICI モジュールを初期化する」を参照してください。NICI が初期化されているかどうかを確認するには、「サーバ上に NICI がインストールおよび初期化されているかどうかを確認する」を参照してください。

重要なレプリカというエラーコードが表示され、既存の OpenText eDirectory オブジェクトの別のサーバへの移動が失敗した後に、マルチサーバツリー内でツリーキーサーバとして機能している OpenText eDirectory サーバの削除

この操作を完了するには、［セキュリティコンテナ］>［KAP］以下にある W0 オブジェクトで、キーサーバ DN 属性をこのサーバからツリーキーをダウンロードしたツリー内にある別のサーバに変更します。

1. OpenText Identity Console のホームページで、［**オブジェクトの管理 (Obejct Management)**］タイルをクリックします。
2. ［タイプ］ドロップダウンリストで、［**セキュリティドメインキーリスト (Security Domain Key List)**］を選択し、［**検索**］をクリックします。
3. W0 オブジェクト (通常は W0.KAP.Security) を選択します。
4. ［**NDSPKI:SD Key Server DN**］を選択して、［**項目の変更**］をクリックします。
5. ［**セキュリティドメインキーサーバの DN**］フィールドで別のサーバの名前とコンテキストを指定してから、［**OK**］をクリックします。
6. ［**適用**］をクリックし、［**OK**］をクリックします。

認証局を保持している OpenText eDirectory サーバのアンインストール中に、サーバに作成された KMO がツリー内の別のサーバに移動されて無効になる

この場合、ツリーの CA および KMO をもう一度作成する必要があります。詳細については、『*OpenText™ eDirectory CE 24.4 (v9.3) eDirectory 管理ガイド*』の「「組織の認証局オブジェクトを作成する」」および「「サーバ証明書オブジェクトを作成する」」を参照してください。

ツリーの認証局を作成した OpenText eDirectory サーバは、アンインストールしないことをお勧めします。

PKIDiag の使用

PKIDiag は、Certificate Server オブジェクトを診断および修復するために設計されたユーティリティです。PKIDiag を使用すると、次のような処理を行うことができます。

- サーバが移動された場合に、サーバ関連オブジェクトが正しい名前になっていて包含スキームに一致するように、それらのオブジェクトを名前変更または移動する。
- 必要なオブジェクトが存在しない場合に、それらのオブジェクトを作成する。
- オブジェクト間で必要な権限を付与する。
- オブジェクトがリンクされていない場合にリンクする。

- SSL CertificateIP 証明書および SSL CertificateDNS 証明書が存在しない場合に、それらを作成する。
- SSL CertificateIP 証明書および SSL CertificateDNS 証明書が期限切れになっている場合、期限切れが近づいている場合、または名前が正しくない場合に、それらを修復する。

PKIDiag 機能は、OpenText Identity Console のサーバ自動ヘルスチェックおよび［デフォルト証明書の作成］タスクという他の 2 つのプロセスで使用されます。

サーバ自動ヘルスチェックは、サーバが再起動された場合、または DSREPAIR が実行された場合に必ず実行されます。デフォルト証明書の作成は、Certificate Server のインストール時に作成されたデフォルト証明書を置き換える場合に使用する処理です。詳細については、734 ページの 「デフォルトのサーバ証明書オブジェクトを作成する」を参照してください。

PKIDiag とその使用方法の詳細については、TID #3640106 を参照してください。

サーバが同期するのを待機中

ユーザ証明書が作成された後、クライアントが新しい証明書を組み込むビューを更新できないことがあります。「サーバが同期するのを待機中」というメッセージが示されたダイアログボックスが表示されます。この時点で、ユーザ証明書の作成は完了していますが、その作成に関わったサーバの同期はまだ完了していません。このダイアログボックスは、ユーザの証明書の作成に影響を与えずに閉じることができます。

証明書のニックネーム再利用のエラー

ユーザ証明書の作成中にエラーが発生する場合は、その証明書に別のニックネームの使用を試みてください。指定したニックネームは、再利用できない可能性があります。

ユーザの秘密鍵のエクスポートで -1426 エラー

ユーザオブジェクトが存在するパーティションのレプリカを保持するサーバはすべて、同じレベルの暗号化 (米国向け / 全世界向け NICI、または制限付き NICI をインポート) が必要です。各サーバで同じレベルの暗号化を保持していない場合、キーサイズが大きすぎると、ユーザの秘密鍵をエクスポートする際に、-1426 エラーが表示される場合があります。

-1426 エラーの発生後にユーザの秘密鍵をエクスポートするには、パーティションのレプリカを保持するサーバの暗号化をアップグレードするか、エクスポート可能な暗号化を保持するサーバからレプリカを削除する必要があります。

サーバが有効期限切れの SSL CertificateIP 証明書を使用している

OpenText eDirectory 9.0 では、SSL CertificateIP 証明書はサポートされません。以前のバージョンから OpenText eDirectory 9.0 にアップグレードする場合、SSL CertificateIP 証明書はサーバに関連付けられたままになります。環境内の証明書の有効期限が切れた場合、SSL CertificateIP 証明書は自動的には更新されません。

OpenText eDirectory 9.0 にアップグレードした後ならいつでも、SSL CertificateIP 証明書の代わりに、SSL CertificateDNS 証明書の使用を開始できます。

外部認証局

VeriSign などの一部のサードパーティの認証局では、サーバ証明書の署名に中間認証局を使用しています。これらの証明書をサーバ証明書オブジェクトにインポートするには、サーバ証明書とともに中間認証局とルート認証局証明書が、単一の PKCS #7 形式のファイル (.P7B) 内に含まれている必要があります。ご使用の認証局でこのファイルを提供できない場合は、Internet Explorer 5.5 以降がインストールされたクライアントマシン上で次の手順を実行して、このファイルを作成できます。

1. サーバ証明書を Internet Explorer にインポートします。このインポートをするには、ファイルをダブルクリックするか、[**ファイル**] > [**開く**] を選択してから、そのファイル名を選択します。
2. Internet Explorer で、この外部認証局の証明書が信頼された証明機関としてリストされていない場合、中間認証局とルートレベル認証局を同じ方法でインポートします。
3. Internet Explorer で、[**ツール**] > [**インターネットオプション**] の順に選択します。[**コンテンツ**] タブを選択してから、[**証明書**] ボタンを選択します。
4. [**個人**] タブで、対象のサーバ証明書を探します。そのサーバ証明書を選択して、[**エクスポート**] をクリックします。
5. ウィザードで [エクスポート ファイルの形式] ページが表示されるまでデフォルトを受け入れてから、このページで [Cryptographic Message Syntax Standard - PKCS #7 証明書 (.p7b)] 形式を選択します。
6. ウィザードを続行します。

   これで、PKCS #7 ファイルをサーバ証明書オブジェクトにインポートできるようになります。

サーバの移動

サーバオブジェクトを移動する場合、そのサーバの LDAP オブジェクト、SAS サービスオブジェクト、およびサーバ証明書オブジェクト (暗号化キーオブジェクト) も移動する必要があります。しかし、その次にそのサーバを再起動する際に、サーバ自動ヘルスチェック機能によって各オブジェクトは移動されます。

DNS サポート

サーバに DNS が設定されている場合、サーバ証明書のデフォルトのサブジェクト名は、次のようになります。

.CN=< *サーバの DNS 名* >.O=< *ツリー名* >

サーバに DNS が設定されていない場合、デフォルトのサブジェクト名はサーバの完全識別名です。デフォルトのサブジェクト名を変更するには、証明書の作成処理中に [カスタム] を選択します。

OpenText eDirectory からサーバを削除する

OpenText eDirectory™ からサーバを削除し、その後同じ名前の同じコンテキストにそのサーバを再インストールする場合に、削除されたサーバを表す SAS サービスオブジェクトが存在するときは、そのオブジェクトも削除した場合にのみ、そのサーバの再インストールが正常に実行されます。

このプロセスは、次のように実行する必要があります。

1. デフォルトの証明書をバックアップする必要があるかどうかを判断します。バックアップする場合、それらをバックアップします。
2. デフォルトの証明書を削除します。
3. SAS オブジェクトを削除します。

たとえば、MYSERVER という名前のサーバの場合、SAS Service - MYSERVER という名前の SAS オブジェクトがサーバと同じコンテナに存在する可能性があります。サーバをツリーから削除した後で、サーバをツリーに再インストールする前に、この SAS オブジェクトを (Identity Console を使用して) 手動で削除する必要があります。

サーバが組織の認証局または SD キーサーバの場合、さらにいくつかの手順を実行する必要があります。これらの手順は、「TID #3623407」に記載されています。

このサーバ用に作成されたデフォルトのサーバ証明書も削除する必要があります。これは、サーバが再度挿入されたときにそれらのサーバ証明書が再作成されるようにするためです。

これらの証明書は、「SSL Certificate IP - MYSERVER」および「SSL Certificate DNS - MYSERVER」です。これらの証明書を削除する際は、注意する必要があります。データがこれらの証明書のいずれかを使用して暗号化されていた場合は、証明書を削除する前にデータを取得する必要があります。

認証局のサブジェクト名の制限事項

サブジェクト名に @ 文字が含まれたサーバ証明書が原因で、SSL 接続が失敗する場合があります。この問題の解決方法については、テクニカルサポートに問い合わせてください。

証明書の検証の速度

証明書の検証プロセスには、証明書のデータだけでなく、証明書チェーン内のデータに対する複数のチェックが含まれます。証明書チェーンは、ルート認証局証明書と、必要に応じて 1 つ以上の中間認証局の証明書からなります。

証明書とその関連付けられた証明書チェーン内の情報を検証することは、時間のかかる処理ではありません。ただし、次の場合、検証にかかる時間が長くなることがあります。

- 証明書が外部認証局によって署名されていた場合、1 つ以上の証明書に CRL 配布ポイント拡張が含まれています。

  この証明書を検証するには、チェーン内の該当する証明書ごとの CRL を取得する必要があります。この後、CRL を検査して、証明書が取り消されたかどうかを調べる必要があります。

CRL が多い場合、または CRL 配布ポイントを運用しているサーバがビジー状態の場合、証明書の検証にしばらく時間がかかる場合があります。次のことを 1 つ以上を行って、必要な時間を短縮できます。

- 証明書の取り消しステータスを確認するために使用する接続の速度をアップグレードします。
- 認証局プロバイダに問い合わせます。

- 1 つ以上の証明書に OCSP AIA 拡張がある場合。OCSP レスポンダがビジー状態のときは、検証にかなりの時間がかかることがあります。
- ユーザ証明書を検証している場合。

  サーバ証明書の場合、証明書チェーン全体がサーバ証明書とともに暗号化キーオブジェクトに保存されます。このため、サーバ証明書が検証済みの場合は、クライアントは 1 つのオブジェクトを読み込むだけで、必要な証明書すべてを入手できます。ただし、ユーザ証明書は異なる方法で保存されます。ユーザ証明書自体だけがユーザオブジェクトに保存されます。このため、クライアントはユーザ証明書を検証するために、セキュリティコンテナ内に格納されている他のオブジェクトから証明書チェーンを取得する必要があります。

  組織の認証局によって署名されたユーザ証明書を検証するには、クライアントは認証局の証明書を取得するために組織の認証局のオブジェクトを読み込む必要があります。外部認証局によって署名されたユーザ証明書を検証するには、クライアントはユーザ証明書に一致する証明書チェーンを作成するために、セキュリティコンテナ内のルート認証局コンテナを読み込む必要があります。後者の場合、ユーザ証明書の検証が成功するように、管理者が外部認証局の証明書をルート認証局コンテナにインポート済みである必要があります。

  信頼されなくなった有効期限切れの証明書をルート認証局コンテナから削除することで、ユーザ証明書の検証に要する時間を短縮することができます。

組織の認証局削除後の証明書検証

組織の認証局を削除する (バックアップと復元手順の実行中以外) 場合、自己署名証明書をエクスポートして、ルート認証局コンテナ内に新しいルート認証局を作成する必要があります。新しいルート認証局を作成しなかった場合、これらの証明書を検証する際に次の動作が発生します。

- 削除した認証局によって署名されたユーザ証明書が無効になる。これは、ユーザ証明書に署名した認証局の証明書が組織の認証局オブジェクトまたはルート認証局コンテナ内で見つからないためです。これらのユーザ証明書を有効なままにしたい場合は、以前に削除した認証局の自己署名証明書をルート認証局コンテナに追加する必要があります。
- 削除した認証局によって署名されたサーバ証明書は、引き続き有効である。これは、削除した認証局の証明書がサーバ証明書とともに暗号化キーオブジェクトに格納されるためです。

  鍵の暗号漏洩、またはなんらかのセキュリティ違反 (抜け穴) のために、組織の認証局を削除する場合、削除した認証局によって署名されたすべてのユーザ証明書およびサーバ証明書をただちに取り消す必要があります。それらの証明書を取り消すことができない場合は、それらを削除して、それぞれの所定の場所に新しい証明書を作成します。また、この組織の認証局の証明書をブラウザにインポートした可能性のあるすべてのユーザに対して、その証明書を削除するように指示する必要もあります。

セキュリティコンテナの名前変更

セキュリティコンテナの名前を変更することはできません。

ツリー認証局を再作成した後、OpenText eDirectory が証明書を検証できない

OpenText eDirectory を最新バージョンにアップグレードした場合、またはカスタムロケーションにインストールした場合、OpenText eDirectory は認証局を再作成した後で証明書を検証することができません。

この問題を回避するため、OpenText eDirectory のインストールパスが C:\NetIQ\eDirectory (Windows の場合) および /var/opt/novell/eDirectory (Linux の場合) 以外である場合は、ツリー認証局を再作成するとき、または CRL オブジェクトの作成中に、OpenText eDirectory のインストールパスから見た正しい CRL ファイルパスを指定する必要があります。エラーを避けるには、[認証局の設定] ウィザードから認証局を再作成する際に、OpenText Identity Console > [**証明書管理**] タイル > [**CA 管理**] で [カスタム] オプションを選択し、正しい CRL ファイルパスを指定する必要があります。たとえば OpenText eDirectory は、デフォルトで CRL ファイルをパス C:\NetIQ\eDirectory\htdoc\crl\ にインストールします。eDirectory をカスタムロケーションにインストールすることを選択した場合 (C:*CustomLocation*\eDirectory\)、ツリー認証局を再作成する際に CRL ファイルパスを忘れずに更新してください。C:*CustomLocation*\eDirectory\htdoc\crl\ のように指定します

注 : OpenText eDirectory が以前のいずれかのバージョンでデフォルトの場所 (C:\Novell\NDS\) にインストールされている場合でも、最新バージョンにアップグレードした後、ツリー認証局を再作成する際に前述の回避策を実行する必要があります。

ユーザが DES ツリーキーを使用してツリー内のパスワードを変更しようとすると、OpenText eDirectory でエラーメッセージが表示される

OpenText eDirectory サーバを 9.2 以降にアップグレードした後で、OpenText eDirectory ツリーに DES ツリーキーがある場合、パスワードを変更できず、エラー「-16060」が表示される場合があります。これは、パスワードが DES ツリーキー (56 ビット) で保護され、nds.conf ファイルで FIPS モードが有効になっている (n4u.server.fips_tls=1) 場合に発生します。

この問題を回避するには、次の手順を実行してください。

1. n4u.server.fips_tls を 0 に設定して、OpenText eDirectory サーバで FIPS モードを無効にします。
2. SDIdiag 2.7.0 を使用して、SD -G コマンドで新しい 3DES ツリーキーを生成します。

   注 : SDIdiag 2.7.0 をインストールするには、eDirectory 8.8.8 P11 を新しいマシンにインストールしてから、SDIdiag 2.7.0 を同じマシンにインストールする必要があります。

   ---

3. Diagpwd ユーティリティを使用してパスワードを再暗号化します。詳細については、823 ページの 「ユニバーサルパスワードの診断ユーティリティ」を参照してください。

4 n4u.server.fips_tls を 1 に設定して、OpenText eDirectory サーバで FIPS モードを有効にします。

5 これで、ユーザのパスワードを正常に変更できるはずです。詳細については、TID 7024695 を参照してください。

OpenText eDirectory 認証局が証明書取り消しリスト (CRL) の作成に失敗する

OpenText eDirectory 内の新しい認証局 (CA) が、証明書取り消しリスト (CRL) の作成に失敗します。OpenText Identity Console から CRL にアクセスしようとするときに、エラー -603 が表示され、ndspkiCertificateRevocationList 属性も見つかりません。

現時点で解決策はありません。

Linux でのユーティリティのトラブルシューティング

OpenText インポート / エクスポート変換ユーティリティ

LDAP サーバがリフレッシュまたはアンロードされると、OpenText インポート / エクスポート変換操作の実行中に「LBURP 操作がタイムアウトしました (LBURP operation is timed out)」というメッセージが画面上に表示されます。LBURP 操作がタイムアウトした場合、サーバは後で復元されます。

ndsmerge ユーティリティ

マージ操作後は、PKI サーバはアクティブになっていません。したがって、npki -l コマンドを使用して再起動する必要があります。

異なるバージョンの製品では、マージ操作が成功しないことがあります。サーバで古いバージョンの NDS または OpenText eDirectory が実行されている場合は、最新バージョンの OpenText eDirectory にアップデートしてからマージ操作を続行してください。

ツリーに付随する同じ名前のコンテナがソースツリーおよびターゲットツリーの両方にある場合、2 つのツリーのマージは成功しません。どちらかのコンテナ名を変更してから、マージ操作を続行してください。

結合操作の実行中に、「-611 不正な包含ルールです」というエラーメッセージが表示される場合があります。ndsrepair を実行してスキーマを変更します。次に、ndsrepair -S を実行し、[**オプションスキーマ拡張機能**] を選択します。

フルバックアップを実行する前にインクリメンタルバックアップを実行しようとすると、バックアップ操作が失敗します。

DSTrace ユーティリティ

DSTrace 画面をオンにすると、参照リンクに対するプライマリオブジェクトが不正であることを示すエラーメッセージが表示される場合があります。OpenText eDirectory が正常に機能している場合は、このメッセージを無視してください。

ndsbackup ユーティリティ

OpenText eDirectory のバックアップの実行中に、「NDS エラー : NDS サーバへの接続に失敗しました」というエラーメッセージが表示される場合があります。これは、OpenText eDirectory がデフォルトのポート 524 以外のポートをリスンしていることが原因で発生する場合があります。コマンドラインで、OpenText eDirectory が設定されているポート番号を入力します。たとえば、OpenText eDirectory がポート番号 1524 に設定されている場合は次のように入力します。

ndsbackup sR 164.99.148.82:1524

データのバックアップ中に、OpenText eDirectory から「NDS エラー : パスワードが必要です (NDS Error: Requires a Password)」というエラーが表示される場合があります。これは、サーバに暗号化をマークする属性があるのに、-E オプションを有効にしてバックアップデータを暗号化または復号化していないためです。

フルバックアップを実行する前にインクリメンタルバックアップを実行しようとすると、バックアップ操作が失敗します。

エラー - 786 DSRepair の実行中

DSRepair を使用する場合、マシン内の DSRepair を実行する特定のパーティションに DIB の 3 倍の空き容量が必要になります。

OpenText eDirectory ユーティリティからユーザが NDS パスワードを使用して認証するように要求される

ユニバーサルパスワードを使用している場合、すべての OpenText eDirectory コマンドラインツールを認証するために、ユニバーサルパスワードを NDS パスワードと同期する必要があります。

NMAS のトラブルシューティング

NMAS のエラーコード

NMAS の全エラーコードの一覧は、NMAS NDK にあります。

ログインメソッドおよびログインシーケンスの問題

- 各製品で NMAS ログインメソッドを正しく使用するには、OpenText eDirectory パーティション内の少なくとも 1 つの NMAS サーバが、NMAS を使用する予定のユーザオブジェクトの読み込み / 書き込みレプリカを保持する必要があります。

- ログインメソッドまたはポストログインメソッドによっては、それらがアクティブにされた場合に最初の［パスワード］フィールドを使用しないものもあります。パスワードを入力するためのプロンプトが表示されたら、［パスワード］フィールドを無視してプロンプトを閉じることができます。
- Novell Client が［パスワード］フィールドを表示するように設定されている (デフォルト設定) 場合、簡易パスワードや NDS パスワードなどの 2 つのパスワードメソッドは、AND シーケンスで使用することができません。

管理に関する問題

- 段階的認証のユーザに対しては、明示的な権限を与える必要があります。継承された権限は機能しません。たとえば、管理者のスーパバイザ権は［ルート］コンテナで定義されます。管理者の権限は、ボリュームオブジェクトでは定義されません。管理者がボリュームのセキュリティラベルを「Logged In」から他のセキュリティラベルに変更した場合、管理者は適切な権限を取得できません。管理者は、ボリューム、ディレクトリ、またはボリュームのファイルに対して明示的な権限を割り当てる必要があります。
- ユニバーサルパスワードが有効になっている場合に、簡易パスワードを設定しようとすると、-1697 エラーメッセージが返されます。
- DSBackup (ndsbackup)、DSRepair (ndsrepair)、DSMerge (ndsmerge) などの OpenText eDirectory ユーティリティは、NDS パスワードのみで動作しますが、NMAS の簡易パスワードでは動作しません。OpenText eDirectory 9.0 は、ユニバーサルパスワードを使用します。

  ユニバーサルパスワードの詳細については、『*Password Management 3.3.2 Administration Guide (NetIQ Password Management 3.3.2 管理ガイド)* (https://www.netiq.com/documentation/password_management33/pwm_administration/data/bookinfo.html)』を参照してください。
- ラベルを作成または名前変更する際に、［**OK**］をクリックするか、またはタブを切り替えると、［**Save Changes made for Labels?**］プロンプトに［**いいえ**］と応答した場合でも、常にそのラベルは作成または名前変更されます。変更をキャンセルする場合は、［**キャンセル**］ボタンをクリックする必要があります。ラベルが作成されると、そのラベルを削除することはできません。ただし、使用しないための名前 (Unused_x) などに名前を変更することができます。
- NMAS に対して XDAS 監査を使用すると、次のイベントの DN 形式は LDAP 表記では生成されません。
  - 00290035 SASL メカニズムの結果
  - 00290061 Set Login の設定
  - 00290062 Get Login の設定
  - 00290064 Set Login のシークレット

注 : lsc ファイルで示されているように、ID (00290035、00290061 など) は NMAS イベント ID を指定します。NMAS イベント ID は、XDAS 形式の subEvent フィールドの一部です。

Linux でどのメソッドを使用してもログインできない

NMAS をインストールおよび設定した後で、OpenText eDirectory サーバを再起動します。

メソッドの以前のインスタンスをアンインストールしてからそのメソッドを再インストールした後で、OpenText eDirectory サーバを再起動します。

ICE ユーティリティを使用して追加したユーザが、Linux で簡易パスワードを使用してログインできない

OpenText インポート / エクスポート変換ユーティリティで簡易パスワードを使用するユーザを追加する場合は、-l オプションを使用します。

Windows マシンで SLP_NETWORK_ERROR(-23) が発生する

サービスロケーションプロトコル (SLP) クエリは、DHCP アドレスを持つ仮想マシン、または SLP がブロードキャストされない物理マシンもしくは仮想マシンで -23 SLP_NETWORK_ERROR を返します。

次のいずれかの方法で、ネットワークにディレクトリエージェントを設定することにより、この SLP エラーを回避することができます。

1. C:\Windows\System32\Novell\eDir\OpenSLP\slp.conf ファイルを c:\Windows\ ディレクトリにコピーします。
2. テキストエディタで slp.conf ファイルを開いて、次の行を変更します。

   ```
   ;net.slp.DAAddresses = myDay1,myDa2,myDa3
   ```

   変更後 :

   ```
   net.slp.DAAddresses = <Give your DA Address>
   ```

3. 変更内容を保存し、ファイルを閉じます。

または

1. C:\Windows\System32\Novell\eDir\OpenSLP\slp.conf ファイルを c:\Windows\ ディレクトリにコピーします。
2. テキストエディタで slp.conf ファイルを開いて、次の行を変更します。

   ```
   ;net.slp.isDA = true
   ```

   変更後 :

   ```
   net.slp.isDA = true
   ```

3. 変更内容を保存し、ファイルを閉じます。

Windows で OpenText eDirectory のインストール時に誤ったインストール先パスが［インストール先パス］フィールドに表示される

OpenText eDirectory のインストール時に、デフォルトのインストール場所を受け入れるのではなく、［**ブラウズ**］アイコンをクリックして別の場所を選択してから、フォルダを選択せずに［ブラウズ］ダイアログを閉じた場合、誤ったインストール先パスが［**インストー**

ル先パス (Installation Path)］フィールドに表示されます。この問題は、OpenText eDirectory を Windows Server 2012 Standard Edition (64 ビット) および Windows Server 2012 R2 (64 ビット) にインストール中にのみ発生します。

この問題を回避するには、パスを所要の場所に手動で変更してください。

SLP が Windows で正しく設定されていない場合、サーバの追加が失敗する

SPLD がすでにインストールされていて実行中であると、サーバをツリーに追加するときに (現在のツリーをブラウズする必要があります)、OpenText eDirectory のインストールが失敗します。Windows は［*launch.exe died*］というメッセージを表示します。

OpenText eDirectory を正常にインストールするには、システムを再起動せずに次の手順を実行します。

1. サービスロケーションプロトコルサービスを停止します。
2. C:\Windows\slp.conf ファイルを削除します。
3. C:\Windows\System32\Novell\eDir\OpenSLP フォルダを削除します。
4. Registry HKLM\SYSTEM\CurrentControlSet\Services\slpd から SLPD サーバのレジストリキーを削除します。
5. 管理者の役割で再度セットアップを実行します。

ディレクトリサービスがロードされない場合の HTTPSTK へのアクセス

DS がロードされない場合に HTTPSTK(HTTP プロトコルスタック) にアクセスできる管理者ユーザを事前に設定できます。事前に設定された管理者ユーザ (sadmin) には、OpenText eDirectory 管理者ユーザオブジェクトと同等の権利があります。サーバが、eDirectory が適切に機能していない状態の場合、このユーザとしてサーバにログインし、OpenText eDirectory を使用せずに実行できる必要なすべての診断およびデバッグ作業を実行します。

Windows で sadmin パスワードを設定する

DHost リモートマネージャページ (/dhost URL またはルートページからアクセス可能) を使用して、sadmin パスワードを設定します。sadmin パスワードを設定または変更するには、OpenText eDirectory サーバで dhost.exe を実行している必要があります。

1. Web ブラウザを開きます。
2. アドレス (URL) フィールドに、次の形式で入力します。

   http://*server.name*:*port*/dhost

   たとえば、次のように入力します。

   `http://MyServer:80/dhost`

DHost iConsole へのアクセスに、サーバの IP アドレスを使用することもできます。次に例を示します。

```
http://137.65.135.150:80/dhost
```

3 ユーザ名、コンテキスト、パスワードを指定します。

4 [HTTP サーバ] をクリックしてから、sadmin パスワードを指定します。

5 指定したパスワードを確認入力して、[送信] をクリックします。

Linux で sadmin パスワードを設定する

Linux で sadmin パスワードを設定するには、[DHost Remote Management] ページまたは ndsconfig ユーティリティのいずれかを使用することができます。

DHost リモート管理ページの使用

/dhost URL またはルートページから [DHost Remote Manager] ページにアクセスして、sadmin パスワードを設定します。sadmin パスワードを設定または変更するには、OpenText eDirectory サーバを実行している必要があります。

1 Web ブラウザを開きます。

2 アドレス (URL) フィールドに、次の形式で入力します。

http://*server.name*:*port*/dhost

たとえば、次のように入力します。

```
http://MyServer:80/dhost
```

DHost iConsole へのアクセスに、サーバの IP アドレスを使用することもできます。次に例を示します。

```
http://137.65.135.150:80/dhost
```

3 ユーザ名、コンテキスト、パスワードを指定します。

4 [HTTP サーバ] をクリックしてから、sadmin パスワードを指定します。

5 指定したパスワードを確認入力して、[送信] をクリックします。

ndsconfig の使用

ndsconfig ユーティリティを使用して、sadmin パスワードを設定します。sadmin パスワードを設定または変更するには、OpenText eDirectory サーバで ndsd を実行している必要があります。

サーバコンソールで次のコマンドを入力します。

ndsconfig set http.server.sadmin-pwd=*password*

ここで *password* は、新しい sadmin パスワードです。

ndsconfig ユーティリティの使用に関する詳細については、『*OpenText™ eDirectory CE 24.4 (v9.3) インストールガイド*』の「「ndsconfig ユーティリティのパラメータ」」を参照してください。

データ暗号化のトラブルシューティング

OpenText eDirectory 9.0 では、特定の機密データをディスクに保存したり、クライアントがそのデータにアクセスしたりしている間に、それらのデータを暗号化できます。この章では、OpenText eDirectory 9.0 で暗号化属性やレプリケーション機能を使用しているときに発生する場合があるエラーの情報を提供します。

OpenText eDirectory の他のエラーメッセージの詳細については、OpenText エラーコード Web サイト (http://www.novell.com/documentation/nwec/) を参照してください。

-6090 0xFFFFE836 ERR_ER_DISABLED

OpenText eDirectory レプリカ同期処理が、ターゲットサーバに対し暗号化されたレプリケーションを開始しようとしました。しかし、ターゲット OpenText eDirectory サーバでは暗号化されたレプリカ同期処理が無効になっています

考えられる原因

暗号化されたレプリケーションがターゲット OpenText eDirectory サーバで無効になっています。

[アクション]

ターゲット OpenText eDirectory サーバで暗号化されたレプリケーションを有効にします。

-6089 0xFFFFE837 ERR_REQUIRE_SECURE_ACCESS

アプリケーション (クライアントアクセス) が、クリアテキストチャネルから暗号化属性にアクセスしようとしました。

Source

OpenText eDirectory または NDS。

考えられる原因

暗号化属性が、セキュリティ保護されたチャネルからのみアクセスできるように設定されています。アプリケーションが、クリアテキストチャネルから暗号化属性にアクセスしようとしています。

アクション

アプリケーションは、LDAP セキュアチャネルまたは HTTP セキュアチャネルなどのような、セキュリティ保護されているチャネルから暗号化属性にアクセスする必要があります。

考えられる原因

このエラーをレプリカの作成中に受け取った場合は、レプリカリングの 1 つ以上のサーバが暗号化用にマークされたいくつかの属性を持ち、セキュアチャネルからのみアクセスできるように設定されています。

アクション

セキュリティ保護されていないチャネルから暗号化属性にアクセスできるように、暗号化属性ポリシーの設定を変更します。詳細については、325 ページの第 11 章「OpenText eDirectory のデータを暗号化する」を参照してください。

考えられる原因

暗号化されたレプリケーションがパーティションレベルまたはパーティションのレプリカ間で設定されているときにこのエラーが表示された場合、レプリカリングで eDirectory 8.8.x 以前のサーバが使用されています。

[アクション]

レプリカリング内のすべてのサーバを eDirectory 8.8.x に対応するバージョンにアップグレードします。

-666 FFFFFD66 INCOMPATIBLE NDS VERSION

本文がここに入ります。

考えられる原因

暗号化されたレプリケーションがパーティションレベルで有効になっていて、このパーティションのレプリカを OpenText eDirectory サーバに追加しようとしている場合、このサーバの OpenText eDirectory のバージョンはソースサーバのバージョンと互換性がありません。

[アクション]

サーバを OpenText eDirectory と互換性のあるバージョンにアップグレードします。

考えられる原因

親パーティションが eDirectory 8.8.x 以前のサーバ (混合バージョンのリング) を使用しており、子パーティションで ER が有効になっている場合は、パーティションのマージおよび / または結合操作は不許可になり、ERR_INCOMPATIBLE_DS_VERSION エラーが返されます。

これは、子パーティションにパーティションレベルで ER が有効な機密データが格納されており、親パーティションに eDirectory 8.8.x 以前のサーバが存在するためです。eDirectory 8.8.x サーバ間でのみ ER が有効になっていると、マージの際、eDirectory 8.8.x 以前のサーバに複製しているときに機密データが危険にさらされます。

[アクション]

1. サーバを OpenText eDirectory と互換性のあるバージョンにアップグレードします。

   または

2. ペアレントまたはチャイルドパーティションで ER を無効にします。

注 : ER を無効にしている間、レプリケーションはクリアテキストフォームで行われます。

重複暗号化アルゴリズムの問題

LDIF を使用して暗号化用の属性を追加する場合は、重複したアルゴリズムを一つの属性に関連付けないでください。

例えば、「タイトル」を AES および DES 暗号化アルゴリズムで暗号化された属性としてマークすると、どちらのアルゴリズムが最終的に考慮されるのか不明瞭になります。LIMBER が実行されるときに毎回、タイトル属性トグルが AES と DES の間に表示されます。そのため、設定変更があったように見えてしまいます。

そのようなことを避けるため、同じ属性に重複してアルゴリズムを割り当てるのを避けることをお勧めします。

ストリーム属性の暗号化

ストリーム属性が平文データとして存在する可能性があります。これは、OpenText eDirectory 9.0 ではストリーム属性が暗号化されないことが原因です。

OpenText Identity Console を使用した暗号化レプリケーションの設定

レプリカリング内のいずれかのサーバが停止している場合、OpenText Identity Console を使用して暗号化レプリケーションを設定することはできません。

OpenText Identity Console を使用した暗号化属性の表示または変更

オブジェクトの属性が暗号化されている場合、OpenText Identity Console では、オブジェクトを表示することも変更することもできません。

この問題を回避するために、セキュアチャネルを通じて暗号化属性の表示や変更を行うことができます。これには、次の 2 つの方法があります。

- LDAP: LDAP 要求は、セキュアチャネルを通じて送信する必要があります。そのため、サーバのルート認証局証明書を使用する必要があります。
- ICE: LDIF スクリプトを使用してオブジェクトを変更することができます。この場合、ICE はセキュアチャネルを使用する必要があります。
- OpenText Identity Console 1.7 以降を使用します。

注 : 暗号化属性の表示または変更を行う場合は、OpenText Identity Console 1.7 以降を使用することをお勧めします。

または、暗号化属性を表示または変更できるように、EA ポリシーの requireSecure 属性を無効にすることにより、［セキュアチャンネルが必要です］オプションを無効にできます。この操作により、クリアテキストチャネルから、すべてのクライアントがオブジェクトと暗号化属性にアクセスできるようになります。これで、OpenText Identity Console がオブジェクトにアクセスできるようになります。

暗号化複製が有効になっているツリーのマージに失敗する

暗号化複製が有効になっている場合、ツリーのマージに失敗します。マージを行う前に、各ツリーでセキュリティ保護された複製を無効にします。

Limber で -603 エラーが表示される

暗号化属性ポリシーパーティションのサブリファレンスレプリカのみがサーバにある場合、Limber では、-603 エラーが表示されます。

この問題を回避するには、次のいずれかの手順を実行します。

- NCP サーバオブジェクトへの読み込みアクセスを許可します。これは、OpenText Identity Console から、ツリールートにトラスティを追加し、NCP サーバオブジェクトへの読み込みアクセス権を与えることによって実行できます。属性に attrEncryptionDefinition および attrEncryptionRequiresSecure を指定します。
- LDAP または ndssch を通じて､次の属性へのパブリック読み込みアクセスを許可します。
  - attrEncryptionDefinition
  - attrEncryptionRequiresSecure

OpenText eDirectory 管理ツールボックス

OpenText eDirectory 管理ツールボックス (eMBox) を使用すると、サーバ上でもリモートでも OpenText eDirectory のバックエンドユーティリティすべてにアクセスできます。

eMBox を OpenText Identity Console とあわせて使用すると、DSRepair、DSMerge、バックアップと復元、サービスマネージャなどの OpenText eDirectory ユーティリティに Web ベースでアクセスできます。

重要 : eMBox タスクを実行するには、OpenText Identity Console を使用して、管理対象のツリーに［役割とアクセス制御］を設定する必要があります。

すべての機能は、ローカルサーバまたはリモートのいずれからでもコマンドラインクライアントを通じて使用できます。eMBox クライアントを使用して、1 つのサーバまたはワークステーションから複数のサーバに対するタスクを実行できます。バックアップ、DSRepair、DSMerge、スキーマの操作、および OpenText eDirectory サービスマネージャなどのすべての OpenText eDirectory 管理ツール (eMTools) を実行するには、OpenText eDirectory サーバに eMBox がロードされ、実行されている必要があります。

eMTool サービスを停止できない

コマンド serviceStop -n*{service}* を実行しているときに、*{service}* がサービス (libsasl.so、libncpengine.so、libhttpstk.so、または libdsloader.so) の 1 つである場合は、次のエラーが起こります。

```
Service {service} could not be stopped, Error : -660
```

これはエラーではありません。この進行 (具体的には libsasl.so、libncpengine.so、libhttpstk.so、および libdsloader.so) は、他のモジュールがこれらに依存しているため、止めることはできません。

復元を実行すると -6020 エラーになる

デフォルトの場所にロールフォワードログがある場合、DSBK または eMBox クライアントを使用して復元操作を実行すると -6020 エラーになります。このエラーを回避するには、restore コマンドに -s スイッチを指定する必要があります。

移動したオブジェクトの削除

2 台以上のサーバが含まれるツリーでは、移動したオブジェクトの削除に失敗する場合があります (エラー : 637)。

ダイナミックグループの移動に関する問題

dynamicgroup という Object Class 属性を持つダイナミックグループオブジェクトを他のコンテナに移動すると、ダイナミックグループが機能しなくなります。移動後、ダイナミックメンバーにクエリおよび検索を実行しても機能しません。

からネットワークアドレスを修復する際の問題

eMBox からネットワークアドレスを修復しているとき、eMBox が修復用の最新フィックスで更新されていないと、次のエラーが発生します。

エラー : このサーバのネットアドレスが見つかりませんでした。エラー : 11004

エラー : 接続できませんでした。エラー : 11004

フランス語のマニュアルページの参照

Red Hat Linux でフランス語のマニュアルページを参照するには、次のようにエクスポートします。

```
export MANPATH=/opt/novell/man/frutf8:/opt/novell/eDirectory/man/frutf8
```

移動したオブジェクトの削除

2 台以上のサーバが含まれるツリーでは、移動したオブジェクトの削除に失敗する場合があります (エラー : 637)。

OpenText eDirectory のクライアント制限により OpenText eDirectory でログアウトイベントが生成されない

OpenText eDirectory では、OpenText Identity Console からログアウトしたときにログアウトイベントが生成されません。これは、OpenText eDirectory のクライアント側に存在する技術上の制限によるものです。

アプリケーションの監査では、NWDS API を使用してログアウトイベントを受信できます。LDAP を使用するアプリケーションでは、バインド解除イベントでログアウトを監視できます。

DSTrace 実行中に TERM によって生じる問題

TIME および TAGS のタグが有効であるように表示されますが (下線表示)、デフォルトでは有効ではありません。TERM を Linux ターミナルから VT100 または xterm に設定すると、これらのタグが有効であるように表示されます (下線表示)。この問題は、dtterm などの他のターミナルでは発生しません。

eMBox で 2 バイト文字が処理されない

eMBox では、eMBox クライアントや OpenText Identity Console を使用してロールフォワードディレクトリを設定するときに 2 バイト文字が処理されません。処理するには、DSBK を使用します。

SASL-GSSAPI に関する問題のトラブルシューティング

このセクションでは、SASL-GSSAPI 認証メカニズムによって記録されたエラーメッセージを説明します。

複数のユーザオブジェクトによる問題

同一の Kerberos プリンシパルが複数の OpenText eDirectory ユーザオブジェクトに関連付けられている場合、SASL GSSAPI との LDAP バインドは失敗します。

認証 ID

RFC2222 では、ユーザおよびクライアントによって送信される認証 ID のサポートについて規定されています。これは、SASL GSSAPI メソッドではサポートされていません。

ログファイル

Linux インストールでは、エラーメッセージは ndsd.log ファイルに記録されます。

エラーメッセージ

エラーメッセージ	原因
SASL-GSSAPI: Reading Object user_FDN FAILED OpenText eDirectory error code	このエラーは、OpenText eDirectory 内で生成されます。ケルベロスのプリンシパル名がユーザオブジェクト (userdn) にアタッチされていません。
SASL-GSSAPI: Reading Object Realm_FDN FAILED OpenText eDirectory error code	このエラーは、OpenText eDirectory 内で生成されます。レルムオブジェクトは存在しません。
SASL-GSSAPI: Not enough memory	特定の操作を行うためのメモリが不足しています。
SASL-GSSAPI: Invalid Input	クライアントからの入力が不良、または無効です。

エラーメッセージ	原因
SASL-GSSAPI: NMAS error NMAS error code	このエラーは、NMAS で生成される初期エラーです。
SASL-GSS: Invalid LDAP service principal name *LDAP_service_principal_name*	LDAP サービスのプリンシパル名が無効になっています。
SASL-GSS: OpenText eDirectory からの LDAP サービスプリンシパルキーの読み込みに失敗しました	**原因 :** LDAP サービスのプリンシパルオブジェクトが作成されていません。 **原因 :** レルムオブジェクトのマスタキーが変更されています。 **原因 :** LDAP サービスのプリンシパルオブジェクトが、属するレルムのサブツリーで見つかりませんでした。
SASL-GSS: GSS コンテキストの作成に失敗しました	**原因 :** クライアント、KDC、および OpenText eDirectory サーバで、時間が同期していません。 **原因 :** LDAP サービスプリンシパルのキーが Kerberos データベースで変更されましたが、OpenText eDirectory で更新されていません。 **原因 :** 暗号化タイプがサポートされていません。
SASL GSSAPI: Invalid user FDN = *user_FDN*	クライアントに提供されたユーザ FDN が有効ではありません。
SASL GSSAPI: No user DN is associated with principal *client_principal_name*	サブツリーの下のユーザオブジェクトが、ケルベロスプリンシパル名にアタッチされていません。
SASL GSSAPI: 複数のユーザ DN がプリンシパル *client_principal_name* に関連付けられています	サブツリー下の複数のユーザオブジェクトが、同じプリンシパルに関連付けられています。
ldap_simple_bind_s: Invalid credentials major = 1, minor =0	**原因 :** 原因として、KDC サーバの LDAP サービスプリンシパルと OpenText eDirectory サーバの LDAP サービスプリンシパルの間で、バージョンが一致していないことが考えられます。これは、keytab ファイルに LDAP サービスプリンシパルキーを取り出すたびに、キーのバージョンナンバーが増加するためです。 **アクション :** 次の手順を実行します。 1. バージョン番号が同期するように、OpenText eDirectory サーバでキーを更新します。 2. クライアントでチケットを破棄します。 3. プリンシパル用に TGT を再度取得します。 4. LDAP sasl バインド操作を実行します。

OpenText eDirectory のエラーログを管理する

エラーログは、OpenText eDirectory のインストール中に自動的に開始されます。

メッセージの重大度レベル

すべてのメッセージには重大度レベルが添付されており、そのメッセージがどれだけ重要であるかを判断する助けになります。

エラーメッセージ	説明
Fatal: 致命的エラーのメッセージは、データや機能の損失のような重大な問題を示します。	**例 :** ◆ OpenText eDirectory サーバが、モジュールのロード中に、NCPEngine や DSLoader などのシステムモジュールのロードに失敗した場合は、致命的エラーが報告され、ログに記録されます。 ◆ OpenText eDirectory サーバがセキュアポート 636 でのバインドに失敗すると、致命的エラーが報告され、ログに記録されます。
警告 : 重大とは限らないメッセージですが、将来的に問題を引き起こす原因になる可能性があります。	**例 :** ◆ ツリー内のいずれか2台のサーバ間で接続エラーが発生し、結果的にサーバが不正アドレスのキャッシュに追加された。サーバは、不正アドレスのキャッシュをリセットすると、この状態から回復できます。 ◆ LDAP クライアントアプリケーションがバインドを実行し、バインドを解除しないで接続を閉じた場合、LDAP サーバは適切な警告メッセージを記録する必要があります。 ◆ OpenText eDirectory サーバがファイル記述子をすべて消費してしきい値に達した場合、結果としてサーバは受信要求を処理して応答することができず、アプリケーションのエラーが発生します。

エラーメッセージ	説明
エラー : 無効と見なされる操作が原因で示されるメッセージです。問題の発生を警告するものではありません。	**例** : • クライアントアプリケーションがオブジェクトを追加しようとしたときに、そのオブジェクトの属性定義がスキーマに定義されていない場合、OpenText eDirectory サーバは ERR_NO_SUCH_ATTRIBUTE エラーを報告します。 • 無効なパスワードを使用してユーザがログインしようとすると、OpenText eDirectory サーバは ERR_FAILED_AUTHENTICATION エラーを報告します。
情報 : 操作が正常に完了したことや、OpenText eDirectory サーバ内のイベントについて説明するメッセージです。	**例** : • モジュールが正常にロードまたはアンロードされたときに、操作に関する情報を示すメッセージを記録しておきたい場合があります。 • データベースキャッシュの設定が変更された場合、設定が正常に保存されたことことを示す情報メッセージをログに記録する必要があります。
デバッグ : 開発者がプログラムをデバッグする際に役立つ情報が含まれるメッセージです。	**例** : ダイナミックグループの検索時に、エントリ ID、パーティション ID、およびメンバーの DN とともに、すべてのダイナミックグループメンバーを表示します。この情報は、すべてのメンバーが OpenText eDirectory レベルで返されることを確認する際に役立ちます。

エラーログを設定する

Linux での重大度レベルの設定 : サーバ側メッセージに対してエラーログ設定を行う場合は、/etc/opt/novell/eDirectory/conf/nds.conf 環境設定ファイルで、n4u.server.log-levels パラメータと n4u.server.log-file パラメータを使用できます。

使用できる重大度レベルは、LogFatal、LogWarn、LogErr、LogInfo、および LogDbg です (重大度が高い順)。重大度レベルの詳細については、938 ページの 「メッセージの重大度レベル」を参照してください。

デフォルトでは重大度レベルは「LogFatal」に設定されます。このため、重大度レベルが致命的エラーであるメッセージのみがログに記録されます。

重大度レベルを設定するには、nds.conf ファイル内で、n4u.server.log-levels パラメータを次のように設定します。

n4u.server.log-levels=*severity_level*

次に例を示します。

- 重大度レベルを LogInfo 以上に設定するには、次のように入力します。

  ```
  n4u.server.log-levels=LogInfo
  ```

  この設定を使用すると、重大度レベルが LogInfo 以上 (つまり、LogFatal、LogWarn、および LogErr) のメッセージが、ログファイルに記録されます。
- 重大度レベルを LogWarn 以上に設定するには、次のように入力します。

  ```
  n4u.server.log-levels=LogWarn
  ```

  この設定を使用すると、重大度レベルが LogWarn 以上 (LogFatal) のメッセージが、ログファイルに記録されます。
- 重大度レベルを LogDbg 以上に設定するには、次のように入力します。

  ```
  n4u.server.log-levels=LogDbg
  ```

  この設定を使用すると、重大度レベルが LogDbg のメッセージが、ログファイルに記録されます。

注 : ログレベル (n4u.server.log-levels) を LogDbg に設定する際に、環境変数 NDSD_EVENT_DISK_CACHE を true に設定する必要があります。

Linux でのログファイル名の指定 : メッセージの記録先にするログファイルの場所を指定するには、nds.conf ファイル内で n4u.server.log-file パラメータを使用します。デフォルトでは、ndsd.log ファイルにメッセージが書き込まれます。

たとえば、メッセージを /tmp/edir.log に記録するには、次のように入力します。

```
n4u.server.log-file=/tmp/edir.log
```

システムのログにメッセージを記録するには、次のように n4u.server.log-file パラメータを使用します。

```
n4u.server.log-file=syslog
```

Windows での重大度レベルの設定

使用できる重大度レベルは、LogFatal、LogWarn、LogErr、LogInfo、および LogDbg です (重大度が高い順)。重大度のレベルの詳細については、「938 ページの 「メッセージの重大度レベル」」を参照してください。

重大度レベルを設定するには、次の操作を行います。

1. ［スタート］>［設定］>［コントロールパネル］>［OpenText eDirectory サービス］の順にクリックします。
2. ［サービス］タブで、［dhlog.dlm］を選択します。
3. ［開始パラメータ］ボックスにログのレベルを入力します。

   たとえば、ログのレベルを LogErr 以上に設定するには、次のように入力します。

   ```
   LogLevels=LogErr
   ```

4 ［設定］をクリックします。

5 ［ACS 環境設定］タブで、［DhostLogger］のプラス記号をクリックします。

設定した値で LogLevel パラメータが更新されます。

注 : Windows では、ドライバトレースレベルの重大度は動作しません。

Windows でのログファイル名とパスの指定

1 ［スタート］>［設定］>［コントロールパネル］>［OpenText eDirectory サービス］の順にクリックします。

2 ［サービス］タブで、［dhlog.dlm］を選択します。

3 ［開始パラメータ］に、ログファイルのパスを次のように入力します。

```
LogFile=file_path
```

たとえば、ログファイルのパスを /tmp/Err.log に設定するには、［開始パラメータ］に次のように入力します。

```
LogFile=/tmp/Err.log
```

4 ［設定］をクリックします。

5 ［ACS 環境設定］タブで、［DhostLogger］のプラス記号をクリックします。

設定した値で LogFile パラメータが更新されます。

ログファイルサイズの指定 Windows 上

1 ［スタート］>［設定］>［コントロールパネル］>［OpenText eDirectory サービス］の順にクリックします。

2 ［サービス］タブで、［dhlog.dlm］を選択します。

3 ［開始パラメータ］に、ログファイルのパスを次のように入力します。

```
LogSize=size
```

デフォルトのファイルサイズは 1MB です。

4 ［設定］をクリックします。

5 ［ACS 環境設定］タブで、［DhostLogger］のプラス記号をクリックします。

設定した値で LogSize パラメータが更新されます。

その他

コンテナのバックアップ

ndsbackup を使用しながらオブジェクトを多数 (100 万個程度) 持つコンテナをバックアップするには、コンテナ内のオブジェクトのリストを取得し、個々のバックアップを開始するために、かなりの時間がかかる可能性があります。

OpenText eDirectory への繰り返しログイン

OpenText eDirectory に繰り返しログインすると、使用可能なメモリがすべて消費される可能性があります。iMonitor を使用してログイン更新属性を無効にすると、この問題を解決できます。

イベントシステム統計を有効にする

OpenText eDirectory でイベントが発生して消費されるたびに、そのイベントの時刻関連の統計が保持されます。この情報は、イベントコンシューマの問題を解決するのに役立ちます。この統計はディレクトリの通常の機能に必要ではないため、パフォーマンス上の理由で無効にされています。iMonitor 詳細設定パラメータを使用することで、イベント統計を実行時に有効にすることができます。

イベント統計を表示するには、ENABLE_EVENT_STATISTICS パラメータを設定して、サーバを再起動します。このパラメータは永続的な設定パラメータです。

Linux でのメモリ破損問題のトラッキング

Linux プラットフォームで、OpenText eDirectory はデフォルトのメモリアロケータとして Google malloc (libtcmalloc) を使用します。

メモリ破損の問題をトラッキングするには、ndsd 起動スクリプトに MALLOC_CHECK_ 環境変数を設定します。起動スクリプトはこの変数が設定されているか確認します。設定されている場合は、デフォルトシステムの malloc が使用され、設定されていない場合は、libtcmalloc がロードされます。

ndsd での MALLOC_CHECK_ の設定

- MALLOC_CHECK_ が 0 に設定されている場合、検出されたヒープ破損は無視されて通知されません。
- MALLOC_CHECK_ が 2 に設定されている場合、直ちに中止が呼び出されます。

  このおかげで、メモリ破損の本当の原因を早い段階で特定することができます。さもなければ、原因を後で追求することは難しくなります。

異常ログアウト後に TCP 接続が終了しない

ワークステーションのクラッシュまたは停電のために突然に停止したクライアントホストを、OES Linux サーバが検出できないことがあります。しかし、接続はアクティブのままで、デフォルトのタイムアウト時間 (12 ～ 15 分間ほど) が経過してから切断されます。同時接続数を 1 に設定している場合は、接続を手動で終了するか、またはタイムアウトまでの時間を予測して待ってから再びログインすることをお勧めします。この状況は、ウォッチドッグプロセスが接続を正常に閉じることができなかったときに発生します。そのため、同時接続数が 1 に設定されていて、接続がウォッチドッグによって正常に閉じられていないと、ユーザはログインすることができません。Linux カーネルには、keepalive プローブのサーバ側動作を変更するための 3 つのパラメータが提供されています。TCP レベルで対処方法を実行するには、これらのパラメータを使用します。

これらのパラメータは /proc/sys/net/ipv4/ ディレクトリにあります。

- tcp_keepalive_time: 接続が使用されていない場合に、接続を生かしておくために TCP keepalive パケットを送信する頻度を指定します。この値は keepalive が有効である場合にのみ使用します。

  tcp_keepalive_time には、秒数を整数で指定します。デフォルト値は 7200 秒すなわち 2 時間です。この値はたいていのホストに適しており、多くのネットワークリソースを必要としません。この値を低く設定すると、不要なトラフィックのためにネットワークリソースを使用することになります。
- tcp_keepalive_probes: 接続が切断されたと判断するまでに TCP keepalive プローブを送信する頻度を指定します。

  tcp_keepalive_probes には整数値を指定します。推奨値は 50 未満で、tcp_keepalive_time 値と tcp_keepalive_interval 値によって決まります。デフォルトは、9 プローブ送信後に接続が切断されているとアプリケーションに通知するよう設定されています。
- tcp_keepalive_intvl: 各 keepalive プローブの応答の持続期間を指定します。この値は、接続の keepalive が停止するまでの時間を計算するために重要です。

  tcp_keepalive_intvl には整数値を指定します。デフォルトは 75 秒です。1 プローブが 75 秒だとすると、9 プローブは約 11 分となります。tcp_keepalive_probes 変数および tcp_keepalive_intvl 変数のデフォルト値を使用して、keepalive によって接続が時間切れになるまでのデフォルト時間を評価できます。

余分なネットワークトラフィックが大量に発生せずかつ問題は解決されるように、これら 3 つのパラメータを変更します。一例として、次のように変更できます (検出時間は 3 分)。

- tcp_keepalive_time set -120
- tcp_keepalive_probes - 3
- tcp_keepalive_intvl - 20

注 : パラメータ設定値に注意し、すでに有効である接続の設定をしないようにします。

設定はファイルが変更された直後に有効になります。どのサービスも再起動する必要はありません。ただし、設定は現行のセッションにのみ有効です。サーバを再起動すると、設定はデフォルト設定に戻ります。

設定を (再起動後も) 永続的なものにするには、以下の手順を行います。

次のエントリを /etc/sysctl.conf に追加します。

- net.ipv4.tcp_keepalive_time=120
- net.ipv4.tcp_keepalive_probes=3
- net.ipv4.tcp_keepalive_intvl=20

すべてのクライアントおよびサーバは LAN 経由で接続されている場合にのみ、上記の設定を推奨します。

ユーザオブジェクトに対して ldapsearch を実行中に、システムエラー (-632) の NDS エラーが発生する

簡易パスワードを使用してユーザオブジェクトをインポートし、ユーザオブジェクトがインポートされたコンテナのユニバーサルパスワードを有効にします。DS サーバを停止して NDSD_TRY_NMASLOGIN_FIRST=true を環境に設定してから、DS サーバを起動します。簡易パスワードを使用してインポートされたユーザオブジェクトに対して ldapsearch を実行すると、次のようなエラーが発生します。

```
ldap_bind: Unknown error,  additional info: NDS error: system failure (-
632)
```

この問題を解決するには、ユーザオブジェクトに対して ldapsearch を実行する前に、デフォルトのログインシーケンスを、ユーザオブジェクトがインポートされたコンテナの簡易パスワードとして設定してください。

LDAP が NMAS にユーザのログインを要求する際、NMAS はデフォルトのログインシーケンスを使用します。これらのユーザに対してデフォルトのログインシーケンスを指定しない場合、NMAS は NDS シーケンスを使用します。ユーザをインポートした際にこれらのユーザに NDS パスワードが割り当てられていない場合は、NDS シーケンスは動作しません。ユニバーサルパスワードを有効にすると、ユーザが簡易パスワードを使用してログインする際に、簡易パスワードは NDS パスワードおよびユニバーサルパスワードと同期されます。

SecretStore の無効化 Linux で

OpenText eDirectory 管理者は、次の処理を使用して Linux で SecretStore を無効にできます。

1. nds-modules ディレクトリに移動して、次の SecretStore モジュールの名前を変更して移動します。

   libsss.so

   libssncp.so

   libssldp.so

2. サーバを再起動します。

Windows での SecretStore の無効化

OpenText eDirectory 管理者は、次の処理を使用して Windows で SecretStore を無効にできます。

1. novell\nds ディレクトリに移動して、次の SecretStore モジュールを名前変更するか移動します。

   lsss.dll

   sss.dlm

   ssncp.dlm

   ssldp.dlm

2. サーバを再起動します。

dsbk 環境設定ファイルの場所

dsbk.conf ファイルは、OpenText eDirectory の特定のインスタンスに関連した場所ではなく、/etc 内に格納されています。

DIB ディレクトリがデフォルト以外のパスにある場合、ldif2dib でエラーログファイルを開けない

dib ディレクトリがデフォルト以外の場所に移動されている場合、ldif2dib ではデフォルトのログファイル (ldif2dib.log) を開くことができません。

この問題を回避するには、-b スイッチを使用してログファイルの場所を明示的に指定します。

システムクラッシュの後で ndsd が起動しない

システムクラッシュや電源障害の後に、OpenText eDirectory サービス (ndsd) が起動しないことがあります。OpenText eDirectory を再起動するには、次の手順を実行します。

1. /var/opt/novell/eDirectory/data/ndsd.pid ファイルを削除します。
2. /etc/init.d/ndsd start コマンドを入力します。

Linux コンピュータで、すべてのタグが有効になっている場合に、DSTrace を実行してはいけない

すべてのタグが有効になっている場合、次の場所では DSTRace を実行しないでください。

- **ジャーナルモードでロードされたシステム :** ndsd メモリを構築する可能性があります。
- **インラインモードのサーバ :** ndsd がクラッシュします。

LDAP が匿名検索要求に関する RFC に準拠していない

匿名バインドが無効になっているときに認証されていない検索をクライアントが行うと、LDAP サーバは検索結果の代わりに、不適切な認証であることを示すバインド結果 operationsError を返します。

OpenText eDirectory 9.0 のカスタムインスタンスでのポートのトラブルシューティング

OpenText eDirectory 9.0 で、インスタンスのデフォルトサーバがダウンしたときに、カスタムロケーションに新しいインスタンスを設定する場合、インスタンスのデフォルトポートが取得されます。デフォルトインスタンスのポートはカスタムロケーションのインスタンスに割り当てられるため、デフォルトインスタンスは開始されません。

ホストを再起動する前に「「eDirectory 8.8 のカスタムインスタンスでのポートのトラブルシューティング」」の手順に従ってください。

ホストの再起動

ホストを再起動すると、デフォルトのインスタンスバイナリを使用して作成されたデフォルトインスタンスのみが開始されます。

パスを設定し、ndsmanage を使用して、その他のインスタンスを開始させることができます。

所定の NCP ポート上のループバックアドレスで ndsd が監視していない

複数の OpenText eDirectory インスタンスが存在する場合、2 番目以降のインスタンスは、ループバックアドレス上の NCP ポートではなくデフォルトの 524 ポートをリスンしようとします。

この問題を回避するには、2 番目のインスタンスの「n4u.server.tcp-port」パラメータを監視対象のポートに設定します。n4u.server.tcp-port パラメータは nds.conf ファイルに記載されています。

重要 : OpenText eDirectory 9.0 にアップグレードする前に、OpenText eDirectory のすべてのインスタンスを起動する必要があります。

LDAP トランザクション OID

LDAP トランザクションのサポートでは、supportedGroupingTypes と transactionGroupingType OID の値は同じ (2.16.840.1.113719.1.27.103.7) になります。

LDAP トレースの -5871 エラーおよび -5875 エラー

LDAP トレースの -5871 エラーと -5875 エラーは、通常、LDAP クライアントがアンバインドを実行せずに閉じようとすると起きます。そのため、これらのエラーを心配する必要はなく、無視できます。これらのエラーの詳細については、エラーコード Web サイト (http://www.novell.com/documentation/nwec/) を参照してください。

ツリーの名前が変更されると NDSCons が -625 エラーを出す

プライマリサーバのツリーの名前を変更してセカンダリサーバの DHost をシャットダウンすると、NDSCons ユーティリティはセカンダリサーバに転送失敗エラーメッセージ -625 を表示しますが、DHost はプライマリサーバとセカンダリサーバの両方で実行し続けます。エラーが起きるのは、プライマリサーバでツリーの名前が変更されたときに、NDSCons がセカンダリサーバで実行中であったためです。NDSCons を閉じてから再起動すると、NDSCons は正常に動作します。

注 : ツリー名の変更は、EBA 有効のサーバがツリーに含まれている場合にはサポートされない操作です。

複数の NIC をリスンすると OpenText eDirectory ldapsearch のパフォーマンスが低下する

この問題を回避するには次の手順に従ってください。

環境設定ファイルで、ldapsearch のパフォーマンスを低下させる NIC を無効にします。

または

DSTrace で set NDSTRACE =!ARC1 コマンドを使用して、詳細参照コスト (ARC) を有効にします。

Linux プラットフォームで同時接続ユーザ数を制限できない

Linux プラットフォームでは同時接続ユーザ数を制限できません。従来の動作 (厳密なポートベースのチェック) を実行するには、nds.conf ファイルで次のパラメータを設定します。

```
n4u.server.mask-port-number=0
```

SLP が原因で ndsd がシャットダウンに失敗する

ネットワーク上に SLP ディレクトリエージェント (DA) を設定していない場合、SLP を使用するサービスの検索に時間がかかることがあります。OpenText eDirectory のシャットダウン中に、ndsd は SLP を使用する操作を実行しようとしますが、init スクリプトによって通常許可されているよりも長い時間がそれにかかる場合、強制的にシャットダウンされます。

この問題を解決するには :

1. config ディレクトリに hosts.nds という名前の空のファイルを作成します。サーバの config ディレクトリは、ndsconfig get n4u.server.confdir コマンドを実行して取得できます。
2. /opt/novell/eDirectory/sbin/pre_ndsd_start で export NDS_USESLP=0 を指定することによって、環境変数 NDS_USESLP を 0 に設定します。
3. OpenText eDirectory を再起動します。

Windows での NLDAP の再起動

NLDAP を停止した後に、サーバを再起動して NLDAP をロードする必要があります。

LDAP 経由の SecretStore

SecretStore 機能は、LDAP 経由では動作しません。この問題を解決するには、アンロード ($ nldap -u) を実行してから、ロード (nldap -l) を実行します。

SecretStore のロック解除後に、パスフレーズを変更できない

ユーザの資格情報と誤ったパスフレーズでログインして、忘れたパスワードを取得しようとすると、SecretStore はロックされます。SecretStore のロックは、管理者権限で解除できます。また、OpenText SecureLogin クライアントを使用すると、パスフレーズなしでログインできます。パスフレーズを変更しようとすると、ログインは失敗し、エラーが返されます。

HTTP サーバで SSL CertificateIP の有効期限が切れた後もその証明書を使用する

OpenText eDirectory の古いバージョンから OpenText eDirectory 9.0 にアップグレードした場合、HTTP サーバでは、SSL CertificateIP の有効期限が切れた後も、引き続きその証明書を使用します。これは、eDirectory 8.8 SP8 が、SSL CertificateIP を保守せず、SSL CertificateIP の有効期限が切れたり、SSL CertificateIP が削除されたりした場合でも、その証明書を再発行しないためです。

このため、SSL CertificateIP の有効期限が切れたり、SSL CertificateIP が削除されたりした場合は、OpenText Identity Console を使用するか、SSL CertificateIP の代わりに SSL CertificateDNS を使用して、手動で作成する必要があります。

OpenText eDirectory に 2 つの異なる ldapsearch バイナリが含まれている

LDAP ツールのセット (ldapadd、ldapconfig、ldapdelete、ldapmodify、ldapmodrdn、および ldapsearch) が 2 つ、OpenText eDirectory がインストールされている SLES システム上に (openldap2-client rpm とともに) 存在します。一方のセットは、SLES オペレーティングシステムによってインストールされて /usr/bin にあり、他方のセットは、OpenText eDirectory によってインストールされて /opt/novell/eDirectory/bin にあります。

両方の LDAP ツールのセットの基本機能は同じですが、それぞれのセットで、基本機能のほかに独自の機能を追加しています。PATH 環境変数のパスの設定によって、使用するツールのセットは異なります。このため使用できる機能も異なります。

ldapsearch が何も結果を返さない

バインドユーザが検索フィルタの一部である属性すべての読み込み権限を持っていないと、ldapsearch は結果を何も返しません。

この問題を回避するには、バインドユーザが、検索フィルタの一部である属性すべての読み込み権限があることを確認します。

OpenText eDirectory 9.1 で仮想一覧の表示にエラーメッセージが表示される

OpenText eDirectory 9.1 で、仮想一覧の表示 (VLV) が実行されている OpenText eDirectory サーバにパーティションレプリカすべてが存在しているわけではない場合、VLV がエラーメッセージを表示します。

すべてのパーティションレプリカが、VLM が実行されている OpenText eDirectory サーバ内に存在するようにします。

datadir を新しい場所に移動した後、OpenText eDirectory が起動しない

SLES 12 以降で OpenText eDirectory を設定した後に datadir を新しい場所に移動する場合は、次の手順を実行してください。

- 場所 /usr/lib/systemd/system/ にあるサービスファイル内で、ndsd.pid ファイルを新しい場所に更新します。

  たとえば、nds.conf ファイルがもともと /etc/opt/novell/eDirectory に置かれている場合は、次のようにサンプルのサービスファイルが作成されます。

/usr/lib/systemd/system/ndsdtmpl-etc-opt-novell-eDirectory-conf-ds.conf@.service。

- systemctl daemon-reload コマンドを使用して、デーモンを再ロードします。
- OpenText eDirectory サーバを再起動します。

Restricted の実行ポリシーにより、OpenText eDirectory のインストールが失敗する

PowerShell に対する Windows の実行ポリシーが Restricted に設定されている場合、OpenText eDirectory のインストールは失敗します。

この問題を回避するには、PowerShell に対する Windows 実行ポリシーを Remotesigned に設定します。

LDAP 操作の実行中に、OpenText eDirectory に -659 エラーコードが表示される

複数のサーバ間で同一のユーザに対して LDAP バインド操作を同時に実行すると、1 つのサーバで操作が成功する可能性はありますが、同期が次のレプリカサーバに対して直ちに発生します。ただし、異なるサーバでの操作のタイムスタンプが一致しないため、-659 エラーコードが表示されて、ログインが失敗する場合があります。

この問題を解決するには、NDSD_CC_SKULK_DELAY 環境変数の値を 5 以上に設定します。詳細については、131 ページの 「同期メソッド」を参照してください。それでも同じエラーが発生する場合は、NDSD_CC_SKULK_DELAY を 5 以上に設定するのに加えて、新たに導入された環境変数 NDSD_RETRY_MODIFY を true に設定します。

注

- 上記の環境変数を設定すると、サーバは 100 ミリ秒から最大 2 秒の遅延で同じ操作を再試行するため、OpenText eDirectory のパフォーマンスに影響する可能性があります。
- NDSD_RETRY_MODIFY 環境変数を設定する前に、OpenText eDirectory サーバを停止する必要があります。環境変数を設定した後で、サーバを再起動します。
- NDSD_CC_SKULK_DELAY 環境変数の値を 5 未満に設定すると、NDSD_RETRY_MODIFY 環境変数は無効になります。

最大値がある属性に対して報告されるアラートを無効化する

OpenText eDirectory 9.2.5 以降では、最大値がある属性 (DirXML-EntitlementResult および oidpInstanceData) がデフォルトのしきい値を超えると、OpenText eDirectory は DSE_HIGH_VALUED_ATTR イベントを生成します。この情報は、次の場所にある hvAttr-alert.log に記録されます。

- **Linux:** /var/opt/novell/eDirectory /log
- **Windows:** C:\NetIQ\eDirectory

DirXML-EntitlementResult 属性の基本しきい値は、5000 に設定されています。それ以降は、500 ごとにイベントが生成されます。oidpInstanceData 属性の場合、基本しきい値は 16KB です。これらの値はデフォルトで設定されています。

Linux プラットフォームでアラートを無効にする場合は、/etc/opt/novell/eDirectory/conf ディレクトリにある env ファイルに NDSD_DISABLE_HIGHVALUED_ATTRIBUTES_ALERT 環境変数を追加して、その値を true に設定します。Windows プラットフォームでは、［コントロールパネル］>［システム］>［システムの詳細設定］>［環境変数］>［システム環境変数］>［新規］の順に移動して、新しい変数 NDSD_DISABLE_HIGHVALUED_ATTRIBUTES_ALERT を追加し、その値を true に指定して、システムを再起動します。

破損通知プロセスの処理エラー -785

この問題を回避するには、各リモート NCP サーバオブジェクトの CachedAttrsOnExtRefs から nsimAssignments 属性を削除し、中央サーバにはその属性を残したままにします。

各サーバで ndsrepair -R -l yes を実行し、その後にバックリンクプロセスの実行を強制 (https://www.netiq.com/documentation/edirectory-92/edir_install/data/a7fcuas.html) する操作をトリガします。

注 : この属性は、ndsconfig upgrade が実行されるか、その後の OpenText eDirectory の更新時に再度追加されます。

IPV6 に関する問題のトラブルシューティング

Linux および Windows の未指定の IPv6 アドレスのリスナ

未指定の IPv6 アドレスのリスナは、Linux の IPv4 および IPv6 の両方の接続を受け入れます。この動作のため、Linux では IPv4 と IPv6 の両方の未指定のリスナを同時に同じポートで開始できません。そのため、未指定の IPv6 アドレス用にすでに設定されているリスナがあると、未指定の IPv4 アドレスのリスナは開始できません。Linux は、LDAP リスナ用に未指定のアドレスを使用します。

Windows の場合、未指定の IPv6 リスナは IPv6 接続のみを受け入れます。したがって、IPv6 接続に加えて IPv4 接続を受け入れるには、別個に IPv4 リスナを設定する必要があります。

デフォルトでは、IPv4 リスナと IPv6 リスナの両方は ldapInterfaces 用に設定されます。ldapInterfaces は、プラットフォームに応じて必要なリスナを開始します。

EBA のトラブルシューティング

ユーザ DN に OID のない属性が含まれると、EBA ベースのログインが失敗する

ユーザ DN に OID のない属性が含まれると、EBA ベースのログインが失敗します。

この問題を回避するには、スキーマのユーザ DN 属性に OID を追加する必要があります。