

Oracle Advanced Security

管理者ガイド

リリース 2 (9.2)

2002 年 7 月

部品番号 : J06283-01

ORACLE®

Oracle Advanced Security 管理者ガイド, リリース 2 (9.2)

部品番号 : J06283-01

原本名 : Oracle Advanced Security Administrator's Guide, Release 2 (9.2)

原本部品番号 : A96573-01

原本著者 : Laurel Hale

原本協力者 : Gary Gilchrist, Min-Hank Ho, Michael Hwa, Sudha Iyer, Adam Lindsey Jacobs, Lakshmi Kethana, Andrew Koyfman, Van Le, Nina Lewis, Janaki Narasinghanallur, Andy Philips, Ramana Turlapati, Valarie Moore

Copyright © 1996, 2002 Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

プログラム（ソフトウェアおよびドキュメントを含む）の使用、複製または開示は、オラクル社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権に関する法律により保護されています。

当プログラムのリバース・エンジニアリング等は禁止されています。

このドキュメントの情報は、予告なしに変更されることがあります。オラクル社は本ドキュメントの無謬性を保証しません。

* オラクル社とは、**Oracle Corporation**（米国オラクル）または**日本オラクル株式会社**（日本オラクル）を指します。

危険な用途への使用について

オラクル社製品は、原子力、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションを用途として開発されておりません。オラクル社製品を上述のようなアプリケーションに使用することについての安全確保は、顧客各位の責任と費用により行ってください。万一かかる用途での使用によりクレームや損害が発生いたしましても、日本オラクル株式会社と開発元である **Oracle Corporation**（米国オラクル）およびその関連会社は一切責任を負いかねます。当プログラムを米国国防総省の米国政府機関に提供する際には、『**Restricted Rights**』と共に提供してください。この場合次の **Notice** が適用されます。

Restricted Rights Notice

Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このドキュメントに記載されているその他の会社名および製品名は、あくまでその製品および会社を識別する目的にのみ使用されており、それぞれの所有者の商標または登録商標です。

目次

はじめに	xix
------------	-----

第 I 部 概要

1 Oracle Advanced Security の概要

Oracle Advanced Security について	1-2
イントラネットまたはインターネット環境におけるセキュリティ	1-2
セキュリティの脅威	1-2
Oracle Advanced Security の機能	1-4
データ・プライバシー	1-4
データの整合性	1-6
認証	1-7
シングル・サインオン	1-12
認可	1-13
Oracle Advanced Security のアーキテクチャ	1-14
ネットワーク・プロトコル境界でのデータ転送の保護	1-15
システム要件	1-16
Oracle Advanced Security の制限	1-17

第 II 部 暗号化、整合性および JDBC

2 データの暗号化および整合性の構成

Oracle Advanced Security 暗号化	2-2
概要	2-2
Advanced Encryption Standard	2-2
DES アルゴリズムのサポート	2-3
トリプル DES のサポート	2-3
高速暗号化のための RSA RC4 アルゴリズム	2-3
Oracle Advanced Security データの整合性	2-4
サポートしているデータの整合性アルゴリズム	2-4
Diffie-Hellman ベースの鍵管理	2-5
認証鍵フォールドイン	2-5
データの暗号化および整合性の構成	2-6
暗号化および整合性をアクティブにする	2-6
暗号化および整合性の指定	2-7
暗号化シードの設定	2-9
Oracle Net Manager を使用した暗号化および整合性パラメータの構成	2-9

3 JDBC Thin のサポート

Java 実装について	3-2
Java Database Connectivity のサポート	3-2
JDBC Thin の保護	3-3
実装の概要	3-4
不明瞭化	3-4
構成パラメータ	3-5
クライアント暗号化レベル : ORACLE.NET.ENCRYPTION_CLIENT	3-5
クライアントの暗号化選択リスト : ORACLE.NET.ENCRYPTION_TYPES_CLIENT	3-6
クライアント整合性レベル : ORACLE.NET.CRYPTO_CHECKSUM_CLIENT	3-6
クライアントの整合性選択リスト : ORACLE.NET.CRYPTO_CHEKSUM_TYPES_CLIENT	3-7

第 III 部 認証方式の構成

4 RADIUS 認証の構成

RADIUS の概要	4-2
RADIUS 認証モード	4-4
同期認証モード	4-4
要求 / 応答（非同期）認証モード	4-6
RADIUS 認証、認可およびアカウントを使用可能にする	4-9
タスク 1: Oracle データベース・サーバーと Oracle クライアントに RADIUS をインストールする	4-10
タスク 2: RADIUS 認証の構成	4-10
タスク 3: ユーザーの作成とアクセス権の付与	4-18
タスク 4: 外部 RADIUS 認可の構成（オプション）	4-18
タスク 5: RADIUS アカウントの作成	4-20
タスク 6: RADIUS クライアント名の RADIUS サーバー・データベースへの追加	4-21
タスク 7: RADIUS とともに使用する認証サーバーの構成	4-21
タスク 8: 認証サーバーとともに使用する RADIUS の構成	4-21
タスク 9: マッピング・ロールの構成	4-22
RADIUS を使用したデータベースへのログイン	4-23
RSA ACE/Server 構成チェックリスト	4-23

5 CyberSafe 認証の構成

CyberSafe 認証の構成	5-2
タスク 1: CyberSafe Server のインストール	5-2
タスク 2: CyberSafe TrustBroker Client のインストール	5-2
タスク 3: CyberSafe Application Security Toolkit のインストール	5-2
タスク 4: Oracle データベース・サーバーに対するサービス・プリンシパルの構成	5-3
タスク 5: CyberSafe からサービス表を抽出する	5-4
タスク 6: Oracle データベース・サーバーのインストール	5-5
タスク 7: Oracle Advanced Security を CyberSafe とともにインストールする	5-5
タスク 8: Oracle Net と Oracle9i のインストール	5-5
タスク 9: CyberSafe 認証の構成	5-5
タスク 10: 認証サーバーでの CyberSafe ユーザーの作成	5-8
タスク 11: Oracle データベース・サーバーに外部認証 Oracle ユーザーを作成する	5-9

タスク 12: CyberSafe/Oracle ユーザーの初期チケットの取得	5-9
タスク 13: CyberSafe によって認証された Oracle データベース・サーバーに接続	5-10
トラブルシューティング	5-10
kinit を使用してチケット認可チケットを取得できない場合	5-10
初期チケットはあるが接続できない場合	5-10
サービス・チケットはあるが接続できない場合	5-11
何も問題はないが別の問合せが失敗する場合	5-11

6 Kerberos 認証の構成

Kerberos 認証を使用可能にする	6-2
タスク 1: Kerberos のインストール	6-2
タスク 2: Oracle データベース・サーバーに対するサービス・プリンシパルの構成	6-2
タスク 3: Kerberos からのサービス表の抽出	6-3
タスク 4: Oracle データベース・サーバーと Oracle クライアントのインストール	6-4
タスク 5: Oracle Net Services と Oracle Advanced Security のインストール	6-4
タスク 6: Oracle Net Services と Oracle9i のインストール	6-5
タスク 7: Kerberos 認証の構成	6-5
タスク 8: Kerberos ユーザーの作成	6-10
タスク 9: 外部認証された Oracle ユーザーの作成	6-10
タスク 10: Kerberos/Oracle ユーザーの初期チケットの取得	6-11
Kerberos 認証アダプタで使用するユーティリティ	6-11
okinit ユーティリティを使用して初期チケットを取得	6-12
oklist ユーティリティを使用して資格証明を表示	6-13
okdstry ユーティリティを使用してキャッシュ・ファイルから資格証明を削除	6-13
Kerberos によって認証された Oracle データベース・サーバーに接続	6-14
Windows 2000 ドメイン・コントローラ KDC との相互運用の構成	6-14
タスク 1: Windows 2000 ドメイン・コントローラ KDC と相互運用するための Oracle Kerberos クライアントの構成	6-15
タスク 2: Oracle クライアントと相互運用するための Windows 2000 ドメイン・コントローラ KDC の構成	6-16
タスク 3: Windows 2000 ドメイン・コントローラ KDC と相互運用するための Oracle データベースの構成	6-18
タスク 4: Kerberos/Oracle ユーザーの初期チケットの取得	6-18
トラブルシューティング	6-19

7 Secure Sockets Layer 認証の構成

Oracle 環境における SSL	7-2
SSL で可能なこと	7-2
Oracle 環境における SSL の構成要素	7-3
Oracle 環境における SSL の機能 : SSL ハンドシェイク	7-5
非 Oracle クライアントと Oracle データベース・サーバー間の SSL	7-6
SSL と他の認証方式の併用	7-7
アーキテクチャ : Oracle Advanced Security と SSL	7-7
SSL と他の認証方式の併用	7-9
SSL とファイアウォール	7-10
SSL 使用時の問題	7-11
SSL を使用可能にする	7-12
タスク 1: Oracle Advanced Security および関連製品のインストール	7-12
タスク 2: クライアントでの SSL の構成	7-12
タスク 3: サーバーでの SSL の構成	7-22
タスク 4: データベースへのログオン	7-29
nCipher セキュア・アクセラレータの使用	7-30
nCipher セキュア・アクセラレータを使用するために必要な Oracle のコンポーネント	7-30
nCipher セキュア・アクセラレータを使用するための Oracle Advanced Security の構成	7-31
nCipher セキュア・アクセラレータの使用に関するトラブルシューティング	7-31

8 Entrust 対応の SSL 認証の構成

概要	8-2
Oracle Advanced Security	8-2
Entrust/PKI	8-2
Entrust 対応の Oracle Advanced Security	8-3
システムの構成要素	8-4
Entrust/PKI 6.0 for Oracle	8-5
Entrust/Server Login Toolkit 6.0	8-6
Entrust/IPSEC Negotiator Toolkit 6.0	8-6
Entrust 認証手続き	8-7
Entrust 認証を使用可能にする	8-8
Entrust プロファイルの作成	8-8
Oracle Advanced Security および関連製品のインストール	8-9
クライアントおよびサーバーにおける SSL の構成	8-10

クライアントにおける Entrust の構成	8-10
サーバーにおける Entrust の構成	8-11
データベース・ユーザーの作成	8-13
データベースへのログイン	8-14
問題点と制限事項	8-14
Oracle Advanced Security における Entrust のトラブルシューティング	8-15
プラットフォームに関係なく Entrust 実行時に戻るエラー・メッセージ	8-15
Windows プラットフォームで Entrust 実行時に戻るエラー・メッセージ	8-17
Entrust を実行するための一般的なチェックリスト（すべてのプラットフォームに共通）	8-19

9 複数の認証方式の構成

ユーザー名とパスワードによる接続	9-2
Oracle Advanced Security 認証を使用禁止にする方法	9-2
複数の認証方式の構成	9-4
外部認証を使用する場合の Oracle9i の構成	9-5
sqlnet.ora での SQLNET.AUTHENTICATION_SERVICES パラメータの設定	9-5
REMOTE_OS_AUTHENT が TRUE に設定されていないことを確認	9-6
OS_AUTHENT_PREFIX を NULL 値に設定	9-6

第 IV 部 Oracle DCE Integration

10 Oracle DCE Integration の概要

Oracle DCE Integration の要件	10-2
システム要件	10-2
下位互換性	10-2
分散コンピューティング環境	10-2
Oracle DCE Integration の構成要素	10-3
DCE 通信 / セキュリティ	10-3
DCE Cell ディレクトリ・サービス Native Naming	10-4
DCE の柔軟な配置方法	10-5
リリース制限	10-5

11 Oracle DCE Integration を使用する DCE の構成

Oracle DCE Integration を使用する DCE の構成	11-2
タスク 1: 新規のプリンシパルとアカウントの作成	11-2
タスク 2: サーバーのキーをキータブ・ファイルにインストール	11-3
タスク 3: Oracle DCE Integration で使用する DCE CDS を構成	11-3

12 Oracle DCE Integration を使用する Oracle9i の構成

DCE アドレス・パラメータ	12-2
Oracle9i と Oracle Net Services の構成	12-3
タスク 1: サーバーの構成	12-3
タスク 2: 外部的に認証されるアカウントの作成と命名	12-4
タスク 3: DCE Integration の外部ロールの設定	12-6
タスク 4: SYSDBA および SYSOPER で Oracle データベースに接続するための DCE の構成	12-9
タスク 5: クライアントの構成	12-11
タスク 6: DCE CDS ネーミングを使用するクライアントの構成	12-13

13 DCE 環境の Oracle データベースへの接続

リスナーの起動	13-2
DCE 環境の Oracle データベース・サーバーに接続	13-3
方法 1	13-3
方法 2	13-4

14 DCE 環境と非 DCE 環境の相互運用性

非 DCE 環境のクライアントから DCE 環境の Oracle サーバーに接続	14-2
サンプル・パラメータ・ファイル	14-2
listener.ora ファイル	14-2
tnsnames.ora ファイル	14-4
CDS にアクセスできないときに、tnsnames.ora を使用して名前を検索	14-5
SQL*Net 2.2 以前のリリース	14-5
SQL*Net リリース 2.3 と Oracle Net Services	14-5

第 V 部 Oracle9i エンタープライズ・ユーザー・セキュリティ

15 エンタープライズ・ユーザー・セキュリティの管理

第 I 部: 概要 / 概念	15-2
エンタープライズ・ユーザー・セキュリティの概要	15-2
エンタープライズ・ユーザー・セキュリティ	15-3
エンタープライズ・ユーザーと認証方式	15-4
エンタープライズ・ユーザーとパスワード認証	15-6
エンタープライズ・ユーザー・セキュリティのディレクトリ・エントリ	15-7
ユーザーのデータベース・ログイン情報のセキュリティ	15-12
SSL 使用時のエンタープライズ・ユーザー・セキュリティのプロセス	15-16
パスワード使用時のエンタープライズ・ユーザー・セキュリティのプロセス	15-17
共有スキーマ	15-18
概要	15-18
共有スキーマの構成	15-19
共有スキーマの作成	15-20
共有スキーマ	15-20
エンタープライズ・ユーザーのスキーマへのマッピング	15-21
カレント・ユーザー・データベース・リンク	15-23
エンタープライズ・ユーザー・セキュリティのツール	15-24
Oracle Enterprise Security Manager	15-24
Oracle Enterprise Login Assistant	15-25
Oracle Wallet Manager	15-25
配置に関する考慮事項	15-26
セキュリティ資格証明の集中管理に伴うセキュリティ	15-26
エンタープライズ・ドメイン内のデータベースのメンバーシップ	15-27
第 II 部: SSL 認証とパスワード認証の初期構成	15-27
前提条件	15-28
タスク 1: SSL 用のデータベースの構成	15-32
タスク 2: Wallet の作成とリスナーの起動	15-36
タスク 3: データベースのインストールの検証	15-39
タスク 4: グローバル・スキーマとグローバル・ロールの作成	15-40
第 III 部: SSL 認証の最終構成	15-41
タスク 5: データベース・クライアントの構成	15-42
タスク 6: エンタープライズ・ドメインの構成	15-43

タスク 7: エンタープライズ・ユーザーの構成	15-44
タスク 8: エンタープライズ・ユーザーとしてのログイン	15-47
第 IV 部: パスワード認証の最終構成	15-49
タスク 9: エンタープライズ・ドメインの構成	15-50
タスク 10: Oracle コンテキストの構成	15-51
タスク 11: エンタープライズ・ユーザーの構成	15-54
タスク 12: パスワード認証を受けたエンタープライズ・ユーザーでの接続	15-57
第 V 部: エンタープライズ・ユーザー・セキュリティのトラブルシューティング	15-57
データベースへの接続時の ORA-# エラー	15-57
ユーザー・スキーマ・エラーのチェックリスト	15-60
DOMAIN-READ-ERROR のチェックリスト	15-62
暗号化された秘密鍵の復号化に関する障害 (Windows のみ)	15-63
トレース機能の有効化	15-63

16 ローカルまたは外部ユーザーからエンタープライズ・ユーザーへの移行

ローカルまたは外部ユーザーからエンタープライズ・ユーザーへの移行による利点	16-2
ユーザー移行ユーティリティの概要	16-3
一括ユーザー移行処理の概要	16-4
ORCL_GLOBAL_USR_MIGRATION_DATA 表	16-5
移行によるユーザーの元のデータベース・スキーマへの影響	16-8
移行処理	16-9
移行処理を実行するための前提条件	16-10
必須のデータベース権限	16-10
必須のディレクトリ権限	16-10
ユーザー移行ユーティリティの実行に必要な設定	16-10
ユーザー移行ユーティリティのコマンドライン構文	16-11
ユーザー移行ユーティリティ・ヘルプの表示	16-13
ユーザー移行ユーティリティ・パラメータのリスト	16-13
ユーザー移行ユーティリティの使用例	16-21
ユーザーの移行時にユーザー所有スキーマを保持	16-21
ユーザーの移行および共有スキーマへのマッピング	16-22
PARFILE、USERSFILE および LOGFILE パラメータを使用したユーザーの移行	16-26

ユーザー移行ユーティリティでのトラブルシューティング	16-27
ユーザー移行ユーティリティの一般的なエラー・メッセージ	16-27
ユーザー移行ユーティリティの一般的なログ・メッセージ	16-34
ユーザー移行ユーティリティのエラー・メッセージとログ・メッセージの要約	16-37

17 Oracle Wallet Manager の使用方法

概要	17-2
PKCS #12 サポート	17-5
サード・パーティー製 Wallet のインポート	17-5
Oracle Wallet のエクスポート	17-6
複数証明書サポート	17-7
LDAP ディレクトリのサポート	17-9
Wallet の管理	17-10
Oracle Wallet Manager の起動	17-10
Wallet の新規作成	17-11
既存の Wallet のオープン	17-12
Wallet のクローズ	17-12
LDAP ディレクトリへの Wallet のアップロード	17-13
LDAP ディレクトリからの Wallet のダウンロード	17-14
変更内容の保存	17-15
開いている Wallet を新しい位置に保存	17-15
システム・デフォルトへの保存	17-15
Wallet の削除	17-16
パスワードの変更	17-16
自動ログインの使用方法	17-17
証明書の管理	17-17
ユーザー証明書の管理	17-18
信頼できる証明書の管理	17-22

18 Oracle Enterprise Login Assistant の使用方法

Oracle Enterprise Login Assistant について	18-2
Oracle Enterprise Login Assistant の起動	18-2
証明書認証エンタープライズ・ユーザーに対する資格証明の管理	18-3
ローカル・システム上の既存 Wallet のオープン	18-3
LDAP ディレクトリへの接続と新規 Wallet のダウンロード	18-6

パスワードの変更	18-7
LDAP ディレクトリへの Wallet のアップロード	18-10
ログアウトと SSL 接続を使用禁止にする方法	18-10
パスワード認証エンタープライズ・ユーザーに対する資格証明の管理	18-11
パスワードの変更	18-11

19 Oracle Enterprise Security Manager の使用方法

概要	19-2
Oracle Enterprise Security Manager のインストールと構成	19-2
タスク 1: Oracle Internet Directory の構成	19-2
タスク 2: Oracle Enterprise Manager のインストール	19-3
タスク 3: Oracle Enterprise Security Manager の起動	19-4
タスク 4: ディレクトリへのログイン	19-5
エンタープライズ・ユーザーの管理	19-6
エンタープライズ・ユーザーの新規作成	19-6
ディレクトリ・ベースの定義	19-8
新規エンタープライズ・ユーザーのパスワードの定義	19-10
初期エンタープライズ・ロール割当ての定義	19-11
Wallet の作成	19-12
ディレクトリ内のユーザーのブラウズ	19-13
データベース・アクセスを使用可能にする方法	19-16
Oracle コンテキストの管理	19-17
Oracle コンテキストのバージョン	19-17
Oracle コンテキストのプロパティの定義	19-18
データベースのディレクトリへの登録	19-20
ユーザー検索ベースの定義	19-21
Oracle コンテキスト管理者の定義	19-22
パスワード・アクセシブル・ドメインの管理	19-25
データベース・セキュリティの管理	19-27
データベース管理者の管理	19-27
データベース・スキーマ・マッピングの管理	19-28
エンタープライズ・ドメインの管理	19-31
エンタープライズ・ドメイン内のデータベースのメンバーシップの定義	19-32
エンタープライズ・ドメインで使用するデータベース・セキュリティ・オプションの管理	19-35
エンタープライズ・ドメイン管理者の管理	19-35

エンタープライズ・ドメイン・データベース・スキーマ・マッピングの管理	19-36
エンタープライズ・ロールの管理	19-38
エンタープライズ・ロールへのデータベース・グローバル・ロール・メンバーシップの 割当て	19-40
エンタープライズ・ロール権限受領者の管理	19-42

第 VI 部 付録

A データの暗号化と整合性のパラメータ

サンプル sqlnet.ora ファイル	A-2
データの暗号化と整合性のパラメータ	A-4
暗号化と整合性のレベル設定	A-5
暗号化と整合性の選択リスト	A-7
ランダム鍵ジェネレータのシード	A-10

B 認証パラメータ

CyberSafe 認証を使用したクライアントとサーバーのパラメータ	B-2
Kerberos 認証を使用するクライアントとサーバーのパラメータ	B-2
RADIUS 認証を使用するクライアントとサーバーのパラメータ	B-3
sqlnet.ora ファイル・パラメータ	B-3
最小限の RADIUS パラメータ	B-8
初期化ファイル (init.ora) パラメータ	B-8
SSL を使用するクライアントとサーバーのパラメータ	B-9
SSL 認証パラメータ	B-9
Cipher Suite パラメータ	B-10
SSL バージョン・パラメータ	B-11
SSL クライアント認証パラメータ	B-12
Wallet の場所	B-14

C RADIUS による認証デバイスの統合

RADIUS 要求 / 応答ユーザー・インタフェース	C-2
RADIUS 要求 / 応答ユーザー・インタフェースのカスタマイズ	C-2

D Oracle Advanced Security FIPS 140-1 の設定

構成パラメータ	D-2
サーバーの暗号化レベルの設定	D-2
クライアントの暗号化レベルの設定	D-2
サーバーの暗号化選択リスト	D-3
クライアントの暗号化選択リスト	D-3
暗号シード値	D-3
FIPS パラメータ	D-3
インストール後のチェック	D-4
ステータス情報	D-4
物理的なセキュリティ	D-4

E Microsoft Active Directory でのエンタープライズ・ユーザー・セキュリティの使用

Active Directory をサポートする Oracle9i ディレクトリ・サーバーの機能	E-2
ディレクトリ・ネーミング	E-2
エンタープライズ・ユーザー・セキュリティ	E-2
Active Directory との統合	E-3
Active Directory の概要	E-4
ディレクトリ・サーバーの自動検出	E-4
Microsoft のツールとの統合	E-6
Oracle Net ディレクトリ・ネーミングのユーザー・インタフェースの機能拡張	E-6
ディレクトリ・オブジェクト型記述の拡張機能	E-7
Windows ログイン資格証明との統合	E-8
Active Directory の Oracle ディレクトリ・オブジェクト	E-8
Active Directory で Oracle9i を使用するための要件	E-10
Oracle スキーマの作成	E-12
Oracle コンテキストの作成	E-13
ディレクトリ・ネーミング・ソフトウェアの要件	E-13
エンタープライズ・ユーザー・セキュリティソフトウェアの要件	E-14
Active Directory を使用するための Oracle9i の構成	E-15
接続性のテスト	E-17
クライアント・コンピュータからの接続性テスト	E-17
Microsoft のツールによる接続性テスト	E-18

Oracle ディレクトリ・オブジェクトのアクセス制御リストの管理	E-21
セキュリティ・グループ	E-21
セキュリティ・グループへのアクセス	E-22
エンタープライズ・ドメインの作成	E-26

F Java SSL の Oracle 実装

前提条件	F-2
Oracle Java SSL 機能	F-3
Oracle Java SSL でサポートされた SSL Cipher Suite	F-3
Oracle Wallet Manager での証明書と鍵管理	F-4
セキュリティを意識したアプリケーションのサポート	F-4
Oracle Java SSL の例	F-5
例 : SSLServerExample プログラム	F-6
例 : SSLClientExample プログラム	F-10
例 : SSLProxyClientExample プログラム	F-14
Oracle Java SSL のトラブルシューティング	F-16
Oracle Java SSL の API	F-17
パブリック・クラス : OracleSSLCredential	F-17
パブリック・インタフェース : OracleSSLProtocolVersion	F-19
パブリック・クラス : OracleSSLServerSocketFactoryImpl	F-20
パブリック・クラス : OracleSSLSession	F-21
パブリック・クラス : OracleSSLSocketFactoryImpl	F-22
パブリック・インタフェース : OracleX509TrustManagerInterface	F-23

G 略称と頭字語

用語集

索引



1-1	ネットワーク認証サービスでのユーザーの認証方式	1-8
1-2	Oracle ネットワーク環境での Oracle Advanced Security	1-14
1-3	Oracle Net と認証アダプタ	1-15
2-1	Oracle Advanced Security の「Encryption」ウィンドウ	2-10
2-2	Oracle Advanced Security の「Integrity」ウィンドウ	2-12
4-1	Oracle 環境での RADIUS	4-2
4-2	同期認証シーケンス	4-4
4-3	非同期認証シーケンス	4-7
4-4	Oracle Advanced Security の「Authentication」ウィンドウ	4-11
4-5	Oracle Advanced Security の「Other Params」ウィンドウ	4-13
5-1	Oracle Advanced Security の「Authentication」ウィンドウ (Cybersafe)	5-6
5-2	Oracle Advanced Security の「Other Params」ウィンドウ (Cybersafe)	5-7
6-1	Oracle Advanced Security の「Authentication」ウィンドウ (Kerberos)	6-6
6-2	Oracle Advanced Security の「Other Params」ウィンドウ (Kerberos)	6-7
7-1	インターネットから Oracle サーバーへの接続	7-6
7-2	SSL と Oracle Advanced Security の関係	7-8
7-3	SSL と他の認証方式との関係	7-9
7-4	Oracle Advanced Security の「SSL」ウィンドウ (クライアント)	7-15
7-5	「SSL Cipher Suites」ウィンドウ	7-19
7-6	Oracle Advanced Security の「SSL」ウィンドウ (クライアント)	7-20
7-7	Oracle Advanced Security の「SSL」ウィンドウ (サーバー)	7-26
7-8	Oracle Advanced Security の「SSL」ウィンドウ (サーバー)	7-28
8-1	Entrust 認証手続き	8-7
9-1	Oracle Advanced Security の「Authentication」ウィンドウ	9-3
15-1	Oracle コンテキスト内の関連エントリ	15-9
15-2	エンタープライズ・ユーザー・セキュリティの要素 (SSL 認証)	15-14
15-3	エンタープライズ・ユーザー・セキュリティの要素 (パスワード認証)	15-15
15-4	エンタープライズ・ユーザー・セキュリティの動作の仕組み	15-16
15-5	「OracleService」ダイアログ・ボックス	15-38
18-1	「Enterprise Login Assistant Login」ウィンドウ (Wallet が見つかった場合)	18-4
18-2	「Enterprise Login Assistant Logged-In」ウィンドウ	18-5
18-3	「Enterprise Login Assistant Login」ウィンドウ (Wallet が見つからない場合)	18-5
18-4	「Enterprise Login Assistant Directory Login」ウィンドウ	18-6
18-5	「Enterprise Login Assistant Change Password」ウィンドウ	18-8
19-1	「Directory Server Login」ウィンドウ	19-4
19-2	Oracle Enterprise Security Manager: 「Create User」ウィンドウ (「User Naming」タブ) ...	19-7
19-3	Oracle Enterprise Security Manager: 「Browse Directory」ウィンドウ	19-8
19-4	Oracle Enterprise Security Manager: 「Create User」ウィンドウ (「Password」タブ)	19-10
19-5	Oracle Enterprise Security Manager: 「Add Enterprise Roles」ウィンドウ	19-11
19-6	Oracle Enterprise Security Manager: 「Create User」ウィンドウ (「Wallet」タブ)	19-12
19-7	Oracle Enterprise Security Manager: メイン・ウィンドウ (「All Users」タブ)	19-14
19-8	Oracle Enterprise Security Manager: 「Edit User」ウィンドウ	19-15
19-9	Oracle Enterprise Security Manager: 「General」タブ	19-18

19-10	Oracle Enterprise Security Manager: ディレクトリのブラウズ (ユーザー検索ベース)	19-22
19-11	Oracle Enterprise Security Manager の「Administrators」タブ	19-23
19-12	Oracle Enterprise Security Manager: 「Add Users」ウィンドウ	19-24
19-13	Oracle Enterprise Security Manager: 「Database Schema Mappings」タブ	19-29
19-14	Oracle Enterprise Security Manager: 「Add Database Schema Mappings」ウィンドウ	19-30
19-15	Oracle Enterprise Security Manager: 「Create Enterprise Domain」ウィンドウ	19-31
19-16	Oracle Enterprise Security Manager: 「Databases」タブ (データベースのメンバーシップ)	19-33
19-17	Oracle Enterprise Security Manager: 「Add Databases」ウィンドウ	19-34
19-18	Oracle Enterprise Security Manager: 「Database Schema Mappings」タブ	19-37
19-19	Oracle Enterprise Security Manager: 「Create Enterprise Role」ウィンドウ	19-38
19-20	Oracle Enterprise Security Manager: 「Database Global Roles」タブ	19-40
19-21	Oracle Enterprise Security Manager: 「Database Authentication Required」ウィンドウ	19-41
19-22	Oracle Enterprise Security Manager: 「Enterprise Users」タブ	19-43
E-1	Active Directory のディレクトリ・オブジェクト型記述	E-7
E-2	Active Directory ユーザーとコンピュータでの Oracle ディレクトリ・オブジェクト	E-9

表

1-1	スマートカードの利点	1-11
1-2	トークン・カードの利点	1-11
1-3	認証方式とシステム要件	1-16
2-1	暗号化とデータの整合性の指定	2-9
2-2	有効な暗号化アルゴリズム	2-11
2-3	有効な整合性アルゴリズム	2-13
3-1	ORACLE.NET.ENCRIPTION_CLIENT パラメータの属性	3-5
3-2	ORACLE.NET.ENCRIPTION_TYPES_CLIENT パラメータの属性	3-6
3-3	ORACLE.NET.CRYPTO_CHECKSUM_CLIENT パラメータの属性	3-6
3-4	ORACLE.NET.CRYPTO_CHECKSUM_TYPES_CLIENT パラメータの属性	3-7
4-1	RADIUS 認証構成要素	4-3
4-2	RADIUS 構成パラメータ	4-22
5-1	CyberSafe TrustBroker のサービス・プリンシパル名のフィールド値	5-3
6-1	okinit ユーティリティのオプション	6-12
6-2	oklist ユーティリティのオプション	6-13
7-1	Oracle Advanced Security Cipher Suites	7-18
12-1	DCE アドレス・パラメータと定義	12-2
12-2	外部ロール構文の構成要素の設定	12-6
15-1	エンタープライズ・ユーザー認証: 選択条件	15-4
15-2	Oracle コンテキスト内の管理グループ	15-11
15-3	Oracle Enterprise Security Manager または Database Configuration Assistant による データベースのディレクトリへの登録における相違点	15-31
15-4	エンタープライズ・ドメインの設定	15-43
15-5	エンタープライズ・ドメインの設定	15-50
15-6	データベースへの接続時の ORA-# エラー	15-58
16-1	ORCL_GLOBAL_USR_MIGRATION_DATA 表のスキーマ	16-5
16-2	フェーズ 1 とフェーズ 2 の間に変更できるインタフェース表の列の値	16-7
16-3	共有スキーマ・マッピングの選択と CASCADE オプションの指定による影響	16-8
16-4	ユーザー移行ユーティリティのエラー・メッセージ	16-37
16-5	ユーザー移行ユーティリティのログ・メッセージ	16-38
17-1	KeyUsage の値	17-7
17-2	Oracle Wallet Manager による Oracle Wallet へのユーザー証明書のインポート	17-8
17-3	Oracle Wallet Manager による Oracle Wallet への信頼できる証明書のインポート	17-8
17-4	証明書要求: フィールドと説明	17-18
17-5	使用可能なキー・サイズ	17-19
17-6	PKI Wallet エンコーディング規格	17-25
19-1	Oracle Enterprise Security Manager の認証方法	19-5
19-2	「Create User」ウィンドウのフィールド	19-7
19-3	ディレクトリ検索基準	19-14
19-4	Oracle コンテキストのプロパティ	19-18
19-5	Oracle Enterprise Security Manager および Database Configuration Assistant による データベースのディレクトリへの登録における相違点	19-20
19-6	Oracle コンテキスト管理者	19-22

19-7	Oracle Enterprise Security Manager: Oracle コンテキストのオブジェクト	19-27
19-8	Oracle Enterprise Security Manager のデータベース・セキュリティ・オプション	19-35
A-1	アルゴリズムのタイプの選択	A-4
A-2	暗号化と整合性のレベル設定	A-5
A-3	暗号化と整合性の選択リスト	A-7
B-1	CyberSafe 構成パラメータ	B-2
B-2	Kerberos 認証パラメータ	B-2
B-3	SQLNET.AUTHENTICATION_SERVICES パラメータの属性	B-3
B-4	SQLNET.RADIUS_AUTHENTICATION パラメータの属性	B-3
B-5	SQLNET.RADIUS_AUTHENTICATION_PORT パラメータの属性	B-4
B-6	SQLNET.RADIUS_AUTHENTICATION_TIMEOUT パラメータの属性	B-4
B-7	SQLNET.RADIUS_AUTHENTICATION_RETRIES パラメータの属性	B-4
B-8	SQLNET.RADIUS_SEND_ACCOUNTING パラメータの属性	B-5
B-9	SQLNET.RADIUS_SECRET パラメータの属性	B-5
B-10	SQLNET.RADIUS_ALTERNATE パラメータの属性	B-5
B-11	SQLNET.RADIUS_ALTERNATE_PORT パラメータの属性	B-6
B-12	SQLNET.RADIUS_ALTERNATE_TIMEOUT パラメータの属性	B-6
B-13	SQLNET.RADIUS_ALTERNATE_RETRIES パラメータの属性	B-6
B-14	SQLNET.RADIUS_CHALLENGE_RESPONSE パラメータの属性	B-7
B-15	SQLNET.RADIUS_CHALLENGE_KEYWORD パラメータの属性	B-7
B-16	SQLNET.RADIUS_AUTHENTICATION_INTERFACE パラメータの属性	B-7
B-17	SQLNET.RADIUS_CLASSPATH パラメータの属性	B-8
B-18	Wallet の場所パラメータ	B-14
C-1	サーバーの暗号化レベルの設定	C-2
D-1	v\$session_connect_info からのサンプル出力	D-4
E-1	Oracle ディレクトリ・オブジェクト	E-9
G-1	略称と頭字語	G-2

はじめに

このマニュアルは、Oracle Advanced Security の管理者ガイドです。

Oracle Advanced Security には、エンタープライズ・ネットワークを保護し、インターネットに安全に拡張するための、包括的な一連のセキュリティ機能が組み込まれています。

Oracle Advanced Security は、複数のネットワークの暗号化および認証のソリューションを 1 つのソースに統合し、シングル・サインオン・サービスおよびセキュリティ・プロトコルを提供します。

この管理者ガイドでは、Oracle Advanced Security の実装、構成および管理の方法について説明します。

この章の項目は、次のとおりです。

- [対象読者](#)
- [このマニュアルの構成](#)
- [関連文書](#)
- [表記規則](#)

対象読者

このマニュアルは、Oracle Advanced Security の実装、構成および管理を行う次のユーザーとシステム担当者を対象としています。

- 実装の専門家
- システム管理者
- セキュリティ管理者

このマニュアルの構成

このマニュアルは、次の部と章で構成されています。

第 I 部：概要

第 1 章「Oracle Advanced Security の概要」

この章では、このリリースで提供される Oracle Advanced Security の機能の概要について説明します。

第 II 部：暗号化、整合性および JDBC

第 2 章「データの暗号化および整合性の構成」

この章では、既存の Oracle Net Services リリース 2 (9.2) ネットワーク内でのデータの暗号化および整合性の構成方法について説明します。

第 3 章「JDBC Thin のサポート」

この章では、Oracle Advanced Security の Java 実装の概要について説明します。この Java 実装によって、Java Database Connectivity (JDBC) Thin クライアントが Oracle9i データベースに安全に接続できます。

第 III 部：認証方式の構成

第 4 章「RADIUS 認証の構成」

この章では、RADIUS (Remote Authentication Dial-In User Service) を使用する Oracle の構成方法について説明します。RADIUS が Oracle 環境でどのように動作するかについて概説するとともに、RADIUS 認証およびアカウントを使用可能にする方法について説明します。また、サード・パーティ認証デバイス・ベンダーがその認証デバイスを統合するためにカスタマイズできる要求 / 応答ユーザー・インタフェースについても説明します。

第 5 章「CyberSafe 認証の構成」

この章では、CyberSafe を使用する Oracle の構成方法を説明し、Oracle ユーザーを認証する CyberSafe の構成手順について簡単に説明します。

第 6 章「Kerberos 認証の構成」

この章では、MIT Kerberos を使用する Oracle の構成方法を説明し、Oracle ユーザーを認証する Kerberos の構成手順について簡単に説明します。

第 7 章「Secure Sockets Layer 認証の構成」

この章では、Oracle Advanced Security の SSL 機能について説明するとともに、SSL の構成方法について説明します。

第 8 章「Entrust 対応の SSL 認証の構成」

この章では、Secure Socket Layer (SSL) 認証のための、Entrust 対応 Oracle Advanced Security の構成および使用方法について説明します。

第 9 章「複数の認証方式の構成」

この章では、Oracle Advanced Security で使用できる認証方式について説明するとともに、従来のユーザー名およびパスワードの認証の使用方法について説明します。また、Oracle クライアントが特定の認証方式を使用し、Oracle サーバーが任意の方式を受け入れられるようにネットワークを構成する方法も説明します。

第 IV 部 : Oracle DCE Integration

第 10 章「Oracle DCE Integration の概要」

この章では、Open Software Foundation (OSF) DCE と Oracle DCE Integration について簡単に説明します。

第 11 章「Oracle DCE Integration を使用する DCE の構成」

この章では、Oracle DCE Integration を使用する DCE の構成手順について説明します。また、DCE CDS ネーミング・アダプタの構成方法も説明します。

第 12 章「Oracle DCE Integration を使用する Oracle9i の構成」

この章では、クライアントとサーバーが DCE 環境の Oracle サーバーにアクセスできるように、構成ファイルに追加する必要がある DCE パラメータについて説明します。また、外部ロールにマップする DCE グループをセットアップするなど、Oracle サーバー上で必要な構成作業についても説明します。DCE CDS ネーミング・アダプタを使用するクライアントの構成方法も説明します。

第 13 章「DCE 環境の Oracle データベースへの接続」

この章では、DCE 環境の Oracle データベースに接続する方法について説明します。

第 14 章「DCE 環境と非 DCE 環境の相互運用性」

この章では、非 DCE 環境のクライアントが、TCP/IP などの別のプロトコルを使用して Oracle データベースにアクセスする方法について説明します。

第 V 部：Oracle9i エンタープライズ・ユーザー・セキュリティ

第 15 章「エンタープライズ・ユーザー・セキュリティの管理」

この章では、Oracle ディレクトリとセキュリティの統合について説明します。この章では、そのコンポーネントについて説明するとともに、コンポーネント間の介入の概要について説明します。

第 16 章「ローカルまたは外部ユーザーからエンタープライズ・ユーザーへの移行」

この章では、ユーザー移行ユーティリティについて説明します。このユーティリティを使用すると、データベース・ユーザーを LDAP ディレクトリに一括して移行し、エンタープライズ・ユーザーとして格納および管理できます。ユーティリティの構文、前提条件および使用例を記載します。

第 17 章「Oracle Wallet Manager の使用方法」

この章では、Oracle Wallet Manager の構成方法と使用方法について説明します。

第 18 章「Oracle Enterprise Login Assistant の使用方法」

この章では、Oracle Enterprise Login Assistant の構成方法と使用方法について説明します。

第 19 章「Oracle Enterprise Security Manager の使用方法」

この章では、企業のデータベース管理者が Oracle Enterprise Security Manager を使用して、Oracle9i データベースのエンタープライズ・ドメインでデータベースのセキュリティを管理する方法について説明します。

第 VI 部：付録

付録 A「データの暗号化と整合性のパラメータ」

この付録では、Oracle Advanced Security データ暗号化および整合性構成パラメータについて説明します。

付録 B「認証パラメータ」

この付録では、Oracle Advanced Security 認証構成ファイル・パラメータについて説明します。

付録 C 「RADIUS による認証デバイスの統合」

この付録では、サード・パーティ認証デバイス・ベンダーがそのデバイスを統合し、RADIUS 要求 / 応答認証で使用されている Graphical User Interface (GUI) をカスタマイズする方法について説明します。

付録 D 「Oracle Advanced Security FIPS 140-1 の設定」

この付録では、FIPS 140-1 Level 2 で求められる構成に準拠するために必要となる Sqlnet.ora 構成パラメータについて説明します。

付録 E 「Microsoft Active Directory でのエンタープライズ・ユーザー・セキュリティの使用」

この付録では、エンタープライズ・ユーザー・セキュリティのために、LDAP 準拠のディレクトリ・サービスとして Microsoft Active Directory を使用する方法について説明します。

付録 F 「Java SSL の Oracle 実装」

この付録では、Java SSL の Oracle での実装の構成要素とその使用方法の概要について説明します。

付録 G 「略称と頭字語」

この付録では、このマニュアルで使用されている略称と頭字語を定義します。

関連文書

詳細は、次のマニュアルを参照してください。

- Security Dynamics 社の『ACE/Server Administration Manual』
- Security Dynamics 社の『ACE/Server Client for UNIX』
- Security Dynamics 社の『ACE/Server Installation Manual』
- 『Oracle9i Net Services 管理者ガイド』
- 『Oracle9i Heterogeneous Connectivity Administrator's Guide』
- 『Oracle9i JDBC 開発者ガイドおよびリファレンス』
- 『Oracle Internet Directory 管理者ガイド』
- 『Oracle9i データベース管理者ガイド』

このマニュアルに記載されている例の多くは、Oracle のインストール時にデフォルトでインストールされるシード・データベースのサンプル・スキーマを使用しています。これらのスキーマがどのように作成されているか、およびその使用方法については、『Oracle9i サンプル・スキーマ』を参照してください。

リリース・ノート、インストレーション・マニュアル、ホワイト・ペーパーまたはその他の関連文書は、OTN-J (Oracle Technology Network Japan) に接続すれば、無償でダウンロードできます。OTN-J を使用するには、オンラインでの登録が必要です。次の URL で登録できます。

<http://otn.oracle.co.jp/membership/>

OTN-J のユーザー名とパスワードを取得済みであれば、次の OTN-J Web サイトの文書セクションに直接接続できます。

<http://otn.oracle.co.jp/document/>

サード・パーティ・ベンダーからの情報は、次の資料を参照してください。

- 『RADIUS Administrator's Guide』
- 『CyberSafe TrustBroker Release Notes』
- 『CyberSafe TrustBroker Administrator's Guide』
- 『CyberSafe TrustBroker Navigator Administrator's Guide』
- 『CyberSafe TrustBroker UNIX User's Guide』
- 『CyberSafe TrustBroker Windows and Windows NT User's Guide』
- 『CyberSafe TrustBroker Client』
- 『CyberSafe TrustBroker Server』

- CyberSafe Trust Broker のマニュアル
- Kerberos バージョン 5 ソース配布から Kerberos を構築してインストールする方法を説明した資料
- 『Entrust/PKI for Oracle』
- 『Administering Entrust/PKI on UNIX』
- 『Transarc DCE User's Guide and Reference』
- 『Transarc DCE Application Development Guide』
- 『Transarc DCE Application Development Reference』
- 『Transarc DCE Administration Guide』
- 『Transarc DCE Administration Reference』
- 『Transarc DCE Porting and Testing Guide』
- 『Application Environment Specification/Distributed Computing』
- 『Transarc DCE Technical Supplement』

表記規則

このマニュアル・セットの本文とコード例に使用されている表記規則について説明します。

- [本文の表記規則](#)
- [コード例の表記規則](#)
- [Windows オペレーティング・システムの表記規則](#)

本文の表記規則

本文中には、特別な用語が一目でわかるように様々な表記規則が使用されています。次の表は、本文の表記規則と使用例を示します。

規則	意味	例
太字	太字は、本文中に定義されている用語または用語集に含まれている用語、あるいはその両方を示します。	この句を指定する場合は、 索引構成表 を作成します。
固定幅フォントの大文字	固定幅フォントの大文字は、システムにより指定される要素を示します。この要素には、パラメータ、権限、データ型、Recovery Manager キーワード、SQL キーワード、SQL*Plus またはユーティリティ・コマンド、パッケージとメソッドの他、システム指定の列名、データベース・オブジェクトと構造体、ユーザー名、およびロールがあります。	この句は、NUMBER 列に対してのみ指定できます。 BACKUP コマンドを使用すると、データベースのバックアップを作成できます。 USER_TABLES データ・ディクショナリ・ビューの TABLE_NAME 列を問い合わせます。 DBMS_STATS.GENERATE_STATS プロシージャを使用します。
固定幅フォントの小文字	固定幅フォントの小文字は、実行可能ファイル、ファイル名、ディレクトリ名およびサンプルのユーザー指定要素を示します。この要素には、コンピュータ名とデータベース名、ネット・サービス名、接続識別子の他、ユーザー指定のデータベース・オブジェクトと構造体、列名、パッケージとクラス、ユーザー名とロール、プログラム・ユニット、およびパラメータ値があります。 注意： 一部のプログラム要素には、大文字と小文字の両方が使用されます。この場合は、記載されているとおりに入力してください。	sqlplus と入力して SQL*Plus をオープンします。 パスワードは orapwd ファイルに指定されています。 データ・ファイルと制御ファイルのバックアップを /disk1/oracle/dbs ディレクトリに作成します。 department_id、department_name および location_id の各列は、hr.departments 表にあります。 初期化パラメータ QUERY_REWRITE_ENABLED を true に設定します。 oe ユーザーで接続します。 これらのメソッドは JRepUtil クラスに実装されます。

規則	意味	例
固定幅フォントの 小文字の イタリック	固定幅フォントの小文字のイタリックは、プレースホルダまたは変数を示します。	<i>parallel_clause</i> を指定できます。 <i>Uold_release</i> .SQL を実行します。 <i>old_release</i> は、アップグレード前にインストールしたリリースです。

コード例の表記規則

コード例は、SQL、PL/SQL、SQL*Plus またはその他のコマンドラインを示します。次のように、固定幅フォントで、通常の本文とは区別して記載しています。

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

次の表に、コード例の記載上の表記規則と使用例を示します。

規則	意味	例
[]	大カッコで囲まれている項目は、1 つ以上のオプション項目を示します。大カッコ自体は入力しないでください。	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	中カッコで囲まれている項目は、そのうちの 1 つのみが必要であることを示します。中カッコ自体は入力しないでください。	{ENABLE DISABLE}
	縦線は、大カッコまたは中カッコ内の複数の選択肢を区切るために使用します。オプションのうち 1 つを入力します。縦線自体は入力しないでください。	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	<p>水平の省略記号は、次のどちらかを示します。</p> <ul style="list-style-type: none"> ■ 例に直接関係のないコード部分が省略されていること。 ■ コードの一部が繰返し可能であること。 	CREATE TABLE ... AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;

規則	意味	例
.	垂直の省略記号は、例に直接関係のない数行のコードが省略されていることを示します。	<pre>SQL> SELECT NAME FROM V\$DATAFILE; NAME ----- /fsl/dbs/tbs_01.dbf /fsl/dbs/tbs_02.dbf . . . /fsl/dbs/tbs_09.dbf 9 rows selected.</pre>
その他の表記	大カッコ、中カッコ、縦線および省略記号以外の記号は、示されているとおりに入力してください。	<pre>acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;</pre>
イタリック	イタリックの文字は、特定の値を指定する必要があるプレースホルダまたは変数を示します。	<pre>CONNECT SYSTEM/system_password DB_NAME = database_name</pre>
大文字	大文字は、システムにより指定される要素を示します。これらの用語は、ユーザー定義用語と区別するために大文字で記載されています。大カッコで囲まれている場合を除き、記載されているとおりの順序とスペルで入力してください。ただし、この種の用語は大 / 小文字区別がないため、小文字でも入力できます。	<pre>SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;</pre>
小文字	小文字は、ユーザー指定のプログラム要素を示します。たとえば、表名、列名またはファイル名を示します。 注意： 一部のプログラム要素には、大文字と小文字の両方が使用されます。この場合は、記載されているとおりに入力してください。	<pre>SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;</pre>

Windows オペレーティング・システムの表記規則

次の表は、Windows オペレーティング・システムの表記規則とその使用例を示しています。

規則	意味	例
「スタート」→ を選択	プログラムの起動方法を示します。	Database Configuration Assistant を起動するには、「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Configuration and Migration Tools」→「Database Configuration Assistant」を選択します。
ファイル名と ディレクトリ名	ファイル名とディレクトリ名は、大 / 小文字区別がありません。左山カッコ (<)、右山カッコ (>)、コロン (:)、二重引用符 (")、スラッシュ (/)、パイプ () およびハイフン (-) の特殊文字は使用できません。特殊文字の円記号 (¥) は、引用符内にある場合でも要素セパレータとして扱われます。ファイル名が ¥¥ で開始する場合、Windows では汎用命名規則を使用しているとみなされます。	c:¥winnt"¥"system32 は C:¥WINNT¥SYSTEM32 と同じです。
C:¥>	現在のハード・ディスク・ドライブの Windows コマンド・プロンプトを表します。プロンプトは、現在作業中のサブディレクトリを示します。このマニュアルでは、コマンド・プロンプトと呼びます。	C:¥oracle¥oradata>
特殊文字	Windows コマンド・プロンプトでは、二重引用符 (") のエスケープ文字として、特殊文字の円記号 (¥) が必要な場合があります。丸カッコおよび一重引用符 (') にエスケープ文字は不要です。エスケープ文字および特殊文字の詳細は、Windows オペレーティング・システムのマニュアルを参照してください。	C:¥>exp scott/tiger TABLES=emp QUERY=¥"WHERE job='SALESMAN' and sal<1600¥" C:¥>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)
HOME_NAME	Oracle ホーム名を表します。ホーム名は 16 文字以内の英数字で指定できます。ホーム名で使える特殊文字はアンダースコア (_) のみです。	C:¥> net start OracleHOME_ NAMETNSListener

規則	意味	例
<code>ORACLE_HOME</code> と <code>ORACLE_BASE</code>	<p>Oracle8 リリース 8.0 以前では、Oracle コンポーネントをインストールすると、すべてのサブディレクトリはトップレベルの <code>ORACLE_HOME</code> ディレクトリの下に、次のいずれかの名前（デフォルトの場合）で配置されていました。</p> <ul style="list-style-type: none">■ Windows NT の場合は、<code>C:\%orant</code>■ Windows 98 の場合は、<code>C:\%orawin98</code> <p>このリリースでは、Optimal Flexible Architecture (OFA) ガイドラインに準拠しています。トップレベルの <code>ORACLE_HOME</code> ディレクトリの下にすべてのサブディレクトリが配置されるわけではありません。<code>ORACLE_BASE</code> と呼ばれるトップレベル・ディレクトリがあります。<code>ORACLE_BASE</code> のデフォルトは、<code>C:\%oracle</code> です。他の Oracle ソフトウェアがインストールされていないコンピュータに Oracle の最新リリースをインストールする場合、最初の Oracle ホーム・ディレクトリのデフォルト設定は、<code>C:\%oracle%\orann</code>（nn は最新のリリース番号）となります。Oracle ホーム・ディレクトリは、<code>ORACLE_BASE</code> のすぐ下にあります。</p> <p>このマニュアルでのディレクトリ・パスの例はすべて、OFA 表記規則に準拠しています。</p>	<p><code>%ORACLE_HOME%\rdbms\admin</code> ディレクトリにアクセスします。</p>

第 I 部

概要

第 I 部では、Oracle Advanced Security とその機能を紹介します。次の章で構成されています。

- [第 1 章「Oracle Advanced Security の概要」](#)

Oracle Advanced Security の概要

この章では、Oracle Advanced Security とその機能を紹介します。これらの機能は、Oracle9i、Oracle Designer、Oracle Developer など、Oracle Net Services をインタフェースとするデータベースおよび関連製品で使用できます。

次の項目について説明します。

- [Oracle Advanced Security について](#)
- [Oracle Advanced Security の機能](#)
- [Oracle Advanced Security のアーキテクチャ](#)
- [ネットワーク・プロトコル境界でのデータ転送の保護](#)
- [システム要件](#)
- [Oracle Advanced Security の制限](#)

Oracle Advanced Security について

Oracle Advanced Security は、エンタープライズ・ネットワークを保護し、企業内ネットワークをインターネットに安全に拡張するための、包括的な一連のセキュリティ機能を提供します。Oracle Advanced Security は、ネットワークの暗号化および認証のソリューションを 1 つのソースに統合し、シングル・サインオン・サービスおよびセキュリティ・プロトコルを提供します。業界標準を組み込むことにより、Oracle ネットワークにおいて、他に類のないセキュリティが実現されます。

この項では、次の項目について説明します。

- [イントラネットまたはインターネット環境におけるセキュリティ](#)
- [セキュリティの脅威](#)

イントラネットまたはインターネット環境におけるセキュリティ

Oracle データベースは、インターネット上で最大規模の最も人気の高い Web サイトの原動力となっています。世界中の組織において、Oracle9i と Oracle Net Services をベースとする分散データベースおよびクライアント / サーバー・アプリケーションが記録的な勢いで配備されています。分散コンピューティングの普及にともなって、企業はますます多くの情報をコンピュータに格納しています。従業員レコード、財務レコード、顧客注文情報、製品テスト情報およびその他の機密データが、ファイル棚からファイル構造へと移されています。Web 上にある機密データの量が増加したことにより、データが危険にさらされる可能性が高まりました。

セキュリティの脅威

分散環境で使用するデータ量が増加したため、ユーザーは次に示すような様々なセキュリティの脅威に直面しています。

- [傍受とデータの盗難](#)
- [データの改ざん](#)
- [ユーザー ID の偽造](#)
- [パスワード関連の脅威](#)

傍受とデータの盗難

インターネットや Wide Area Network (WAN) 環境では、公共通信事業者と私設ネットワークが、安全性に欠ける陸上回線、障害を受けやすいマイクロ波と衛星リンク、または多数のサーバーをネットワークの一部で使用しているため、貴重なデータが利害関係者にのぞかれる可能性があります。ビルまたは大学・高校などの構内の Local Area Network (LAN) 環境では、物理配線にアクセスできる内部の者が他人のデータを盗み見たり、ネットワークの **Sniffer** (通信を傍受するツール) をインストールしたりできます。

データの改ざん

分散環境になったことで、サイト間で転送されるデータが悪意のある第三者に改ざんされ、整合性が損なわれる可能性も考えられます。

ユーザー ID の偽造

分散環境では、識別情報を偽造して機密情報にアクセスされる危険性が高まっています。クライアント B からサーバー A に接続しているユーザーが、本人であるという保証はあるでしょうか。

さらに、分散環境では、犯罪者が接続を乗っ取る可能性があります。クライアント B とサーバー A が本当にクライアント B およびサーバー A であるという保証はあるのでしょうか。サーバー A の人事システムからサーバー B の給与支払いシステムに転送中のトランザクションが傍受されて、サーバー B として偽装する端末に送られる可能性があります。

パスワード関連の脅威

大規模なシステムでは、多くの場合、ユーザーが使用する様々なアプリケーションやサービスに対して、ユーザーは複数のパスワードを記憶する必要があります。たとえば、開発者はワークステーションで開発中のアプリケーションにアクセスし、電子メール送信のために PC を利用して、テスト、バグの報告、構成管理などの目的で何台かのコンピュータまたはイントラネット・サイトを使用します。

通常、ユーザーは次のいずれかの方法によって複数パスワードの管理を行っています。

- ユーザーは、辞書の中にある名前、架空の人物または言葉などの容易に推測できるパスワードを選択する可能性があります。これらのパスワードはすべて、**辞書攻撃**に弱いという弱点があります。
- また、ユーザーはすべてのマシン上や Web サイト上で同じパスワードを使用できるように、パスワードを標準化する可能性があります。この方法では、パスワードが見破られた場合の危険が増大します。また、1 つのパスワードがわかれば容易に推測できるような類似したパスワードを使用することが考えられます。
- 複雑なパスワードを指定したユーザーは、それを簡単に見つけられる場所に書き留めてしまったり、単に忘れてしまう可能性があります、その管理やサポートにコストがかかります。

いずれの場合も、パスワードの機密性が損なわれ、サービスの可用性が低下します。さらに、複数のユーザー・アカウントとパスワードを管理する作業は複雑で、時間も費用もかかります。

Oracle Advanced Security の機能

Oracle Advanced Security は、様々な方法でデータのプライバシー、整合性、認証、シングル・サインオンおよびアクセス認可を提供します。

たとえば、ユーザーはデータ・プライバシーを守るために、Oracle Net 固有の暗号化または Secure Sockets Layer (SSL) を構成できます。また、Oracle Advanced Security では、Kerberos、スマートカードおよびデジタル証明書などの厳密な認証の方式を選択できます。

Oracle Advanced Security のこれらの機能については次の項で説明します。

- [データ・プライバシー](#)
- [データの整合性](#)
- [認証](#)
- [シングル・サインオン](#)
- [認可](#)

データ・プライバシー

Oracle Advanced Security は、次の暗号化メソッドを使用してデータ送信時のプライバシーを保護します。

- [RC4 暗号化](#)
- [DES 暗号化](#)
- [トリプル DES 暗号化](#)
- [Advanced Encryption Standard](#)

ネットワーク暗号化メソッドは、ユーザーの構成オプションとして選択できるため、データ転送の種類に応じて様々なレベルのセキュリティとパフォーマンスを提供できます。

Oracle Advanced Security の以前のバージョンの暗号化メソッドには、米国内向け、アップグレード用および輸出用の 3 つのエディションがあり、それぞれ異なる鍵の長さを使用していました。リリース 2 (9.2) では、これまで米国内版でのみ使用可能であった暗号化アルゴリズムと鍵の長さを完全に補完したものが組み込まれています。以前のバージョンの製品を使用しているユーザーは、個々の製品リリースに対応する米国内向けバージョンを入手できます。

注意： 米国政府の暗号化製品に対する輸出ガイドラインの規制が緩和されました。これによって、オラクル社は、最も強力な暗号化機能を採用した Oracle Advanced Security を、ほとんどすべての顧客に出荷できるようになりました。

RC4 暗号化

RC4 暗号化モジュールには、RSA Security 社の RC4 暗号化アルゴリズムが採用されています。各セッションに固有のランダムに生成される秘密鍵を使用することによって、すべてのネットワーク通信（すべてのデータ値、SQL 文、ストアド・プロシージャのコールと結果を含む）を完全に保護しています。クライアント、サーバーまたはその両者が暗号化モジュールの使用を要求して、データを保護することができます。Oracle の最適化された処理系によって、パフォーマンスへの影響を最小限に抑えて高度なセキュリティが実現されています。RC4 アルゴリズムでは、40 ビット、56 ビット、128 ビットおよび 256 ビットの暗号化鍵を利用できます。

DES 暗号化

米国データ暗号化規格（DES）アルゴリズムでは、対称鍵暗号化を使用して、ネットワーク通信を保護しています。Oracle Advanced Security は、標準の最適化された 56 ビット鍵の暗号化アルゴリズムを実装しています。また、下位互換性のために、40 ビット・バージョンの DES40 も提供しています。

トリプル DES 暗号化

Oracle Advanced Security は、トリプル DES 暗号化（3DES）もサポートします。この暗号化では、メッセージ・データを DES アルゴリズムに 3 回通すことによって暗号化を行います。3DES は高度なメッセージのセキュリティを実現しますが、パフォーマンスへの影響を伴います。影響の度合いは暗号化を実行するプロセッサの速度にもよりますが、通常、3DES では、データ・ブロックを暗号化するために、標準の DES アルゴリズムに比べて 3 倍の時間がかかります。

3DES には 2 つの鍵を使用する場合と 3 つの鍵を使用場合があります、有効な鍵の長さは、それぞれ 112 ビットと 168 ビットです。両バージョンとも外部**暗号ブロック連鎖**（CBC）モードで動作します。

Advanced Encryption Standard

米国商務省国立標準技術研究所（National Institute of Standards and Technology: NIST）により、米国連邦情報処理標準（Federal Information Processing Standard: FIPS）刊行物 197 として承認された Advanced Encryption Standard（AES）は、DES にかわるものとして開発された新しい暗号化アルゴリズム規格です。AES は、128、192 および 256 ビットの長さの暗号鍵を使用して、128 ビットのデータ・ブロックを処理できる対称型ブロック暗号です。暗号鍵の長さによって、それぞれ AES-128、AES-192 および AES-256 と呼ばれます。3 つのバージョンとも外部 CBC モードで動作します。

米国連邦情報処理標準

Oracle Advanced Security リリース 8.1.6 は、米国連邦情報処理標準（FIPS）140-1 Level2 セキュリティ・レベルの検証を受けています。これによって、Oracle Advanced Security が米国連邦政府の標準に準拠していることが外部から立証されたことになります。FIPS の構成設定については、[付録 D「Oracle Advanced Security FIPS 140-1 の設定」](#)で説明しています。

関連項目：

- [第 2 章「データの暗号化および整合性の構成」](#)
- [付録 A「データの暗号化と整合性のパラメータ」](#)

データの整合性

転送中のデータ・パケットの**整合性**を保証するために、Oracle Advanced Security では暗号論的に安全なメッセージ・ダイジェストを、MD5 または SHA ハッシュ・アルゴリズムを使用して生成し、ネットワーク上で送信される各メッセージにそのメッセージ・ダイジェストを組み込むことができます。

データ整合性アルゴリズムは、ほとんどオーバーヘッドを必要とせず、次のような攻撃からデータを保護します。

- データ変更
- パケット削除
- 再生攻撃

注意： SHA は MD5 よりも若干遅いですが、メッセージ・ダイジェスト値が大きくなることで、激しい衝突や反転攻撃に対してより強力に保護できます。

関連項目： MD5 および SHA の詳細は、[第 2 章「データの暗号化および整合性の構成」](#)を参照してください。

認証

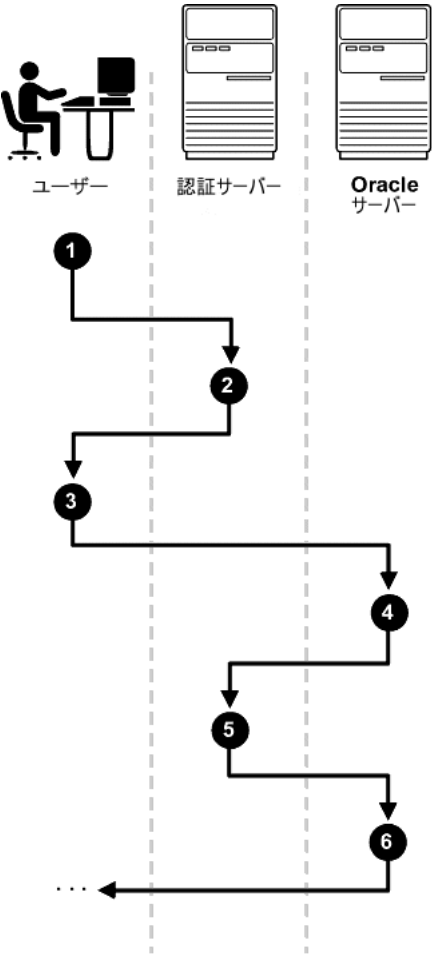
分散環境では、ユーザー ID を認証することが重要です。これを認証しないと、ネットワーク・セキュリティの信頼性がなくなります。パスワードは現在使用されているなかで最も一般的な[認証](#)方式であり、Oracle Advanced Security では各種サード・パーティの認証サービス、および SSL やデジタル証明書を使用して、ユーザー認証を強化しています（[図 1-1](#) を参照してください）。

Oracle Advanced Security 認証方式の多くで集中認証サービスを使用しています。集中認証サービスにより、分散環境でのユーザー、クライアント、サーバーの識別情報における確実性が高まります。ネットワークのすべてのメンバー（クライアント対サーバー、サーバー対サーバー、ユーザー対クライアント、ユーザー対サーバー）の認証を集中化することにより、ネットワーク上で個別性を偽っているノードの脅威に効果的に対応できます。

集中ネットワーク認証サービスの動作

[図 1-1](#) に、集中ネットワーク認証サービスの一般的な動作を示します。

図 1-1 ネットワーク認証サービスでのユーザーの認証方式



1. ユーザー（クライアント）が認証サービスを要求し、トークンやパスワードなどの識別情報を提供します。
2. 認証サーバーはユーザーの識別情報を検証し、クライアントにチケットまたは資格証明を戻します。これには、有効期限が指定されている場合があります。
3. クライアントは Oracle サーバーに対して、データベース接続などのサービス要求とともにこの資格証明を渡します。
4. サーバーは、資格証明の有効性を検証するために、資格証明を認証サーバーに戻します。
5. 認証サーバーは資格証明を承認すると、Oracle サーバーにその旨を通知します。これによって、ユーザーは認証されます。
6. 認証サーバーが資格証明を承認しない場合、認証は失敗し、サービス要求は拒否されます。

サポートされている認証方式

Oracle Advanced Security では次の認証方式をサポートしています。

- [Secure Sockets Layer](#)（デジタル証明書付き）
- [Entrust/PKI](#)
- [Remote Authentication Dial-In User Service](#)
- [Kerberos](#) と [CyberSafe](#)
- [スマートカード](#)
- [トークン・カード](#)

Secure Sockets Layer

Secure Sockets Layer（SSL）はネットワーク接続を保護するための業界標準のプロトコルです。SSL では、[認証](#)、データの[暗号化](#)、データの[整合性](#)を実現し、[公開鍵インフラストラクチャ](#)の一端を担っています。

Oracle Advanced Security の SSL を使用することで、任意のクライアントと任意のサーバーとの間で安全な通信を確立できます。SSL は、サーバーのみを認証、クライアントのみを認証、またはクライアントとサーバーの両方を認証するように構成できます。

SSL では、デジタル証明書（X.509 v3）および[公開鍵と秘密鍵のペア](#)を使用して、ユーザーとシステムの認証を行います。

SSL 機能は SSL のみでも使用することができますが、Oracle Advanced Security でサポートされている他の認証方式とともに使用することもできます。

Entrust/PKI

Oracle Advanced Security では、Entrust Technologies 社が提供している Entrust/PKI ソフトウェアで使用する公開鍵インフラストラクチャをサポートしています。Entrust 対応の Oracle Advanced Security を使用すると、Entrust ユーザーは Entrust のシングル・サインオンを Oracle アプリケーションに取り込み、Oracle ユーザーは Entrust ベースのシングル・サインオンを Oracle アプリケーションに取り込むことができます。

Remote Authentication Dial-In User Service

Remote Authentication Dial-In User Service (RADIUS) は、リモート認証およびアクセスを可能にするための最もよく知られているクライアント・サーバー・セキュリティ・プロトコルのことです。Oracle Advanced Security ではクライアント・サーバー・ネットワークにこの標準を採用して、RADIUS プロトコルをサポートする認証方式であればどのような認証方式でも使用できるようにしています。RADIUS は、トークン・カード、スマートカード、生体的認証などの様々な認証メカニズムで 사용할 ことができます。

Kerberos と CyberSafe

Oracle Advanced Security がサポートする Kerberos と CyberSafe によって、Oracle ユーザーはシングル・サインオンと集中認証サービスを利用できます。Kerberos は、共有シークレットを利用する信頼性の高いサード・パーティ認証システムです。Kerberos は、サード・パーティが安全であるという前提に基づいて、シングル・サインオン機能、パスワード集中格納、データベース・リンク認証、拡張 PC セキュリティを提供します。これらの機能は、Kerberos 認証サーバーまたは Cybersafe Active Trust (Kerberos をベースとした商用の認証サーバー) を介して提供されます。

注意： Kerberos 用の Oracle 認証は、データベース・リンク認証（プロキシ認証ともいう）を提供しています。CyberSafe はプロキシ認証をサポートしていません。

スマートカード

RADIUS 準拠のスマートカードは、クレジット・カードに似たハードウェア・デバイスです。スマートカードにはメモリーとプロセッサが内蔵されており、クライアントのワークステーションにあるスマートカード・リーダーで読み取ります。

表 1-1 にスマートカードの利点を示します。

表 1-1 スマートカードの利点

利点	説明
パスワードのセキュリティの強化	スマートカードは、2つの要素による認証に基づいています。スマートカードはロックすることができ、そのロックを解除できるのは、カードの所有者であり、正確な個人識別番号（PIN）を知っている人のみです。
パフォーマンスの向上	精巧なスマートカードにはハードウェア・ベースの暗号化チップが組み込まれており、ソフトウェア・ベースの暗号化に比べてスループットがよくなります。また、スマートカードにはユーザー名を格納することもできます。
ワークステーションからのアクセス可能性	ユーザーはハードウェア・デバイスにスマートカードを挿入することによりログインします。このデバイスは、スマートカードを読み取り、カードで必要とする認証情報（PIN など）をユーザーに入力するように求めます。ユーザーが正しい認証情報を入力した後で他の認証情報が必要になった場合、それはスマートカードによって生成され入力されます。
使い勝手の良さ	ユーザーが覚えておく必要があるのは、PIN のみです。複数のパスワードを覚えておく必要はありません。

トークン・カード

トークン・カード（SecurID または RADIUS に準拠）は、いくつかの異なるメカニズムにより、使いやすくすることができます。一部のトークン・カードは、認証サービスと同期化されているワンタイム・パスワードを動的に表示します。サーバーは認証サービスと連絡を取り合うことによって、トークン・カードが提供するパスワードをいつでも検証できます。トークン・カードの中にはキーパッドを備えるものがあり、要求 / 応答に基づいて操作します。この場合は、サーバーが要求（番号）を提供し、ユーザーがその番号をトークン・カードに入力します。トークン・カードは応答（最初の番号から暗号的に導出される別の番号）を提供し、それをユーザーが入力してサーバーに渡します。

表 1-2 にトークン・カードの利点を示します。

表 1-2 トークン・カードの利点

利点	説明
パスワードのセキュリティの強化	ユーザーとして偽装するためには、トークン・カードとともにそれを作動させるための個人識別番号（PIN）の両方を所持している必要があります。これを、2つの要素による認証といいます。
使い勝手の良さ	ユーザーが覚えておく必要があるのは、PIN のみです。複数のパスワードを覚えておく必要はありません。

表 1-2 トークン・カードの利点（続き）

利点	説明
責任の明確化	トークン・カードでは、厳密な認証メカニズムを提供しているため、ユーザーは自らが行ったアクションについて責任があります。
ワークステーションからのアクセス性	ユーザーは、PIN を使用すればどのワークステーションからでもログインできます。これにより、ハードウェア・デバイスを追加することなく、2つの要素による厳密な認証を実行できます。

RADIUS アダプタを介して、SecurID トークンを使用できます。

シングル・サインオン

集中認証サービスでは、1つに統合されたユーザーのサインオン（**シングル・サインオン**）を使用できます。この機能によって、ユーザーは1つのパスワードで複数のアカウントとアプリケーションにアクセスできるため、複数のパスワードを使用する必要がなくなり、システム管理者はユーザーのアカウントとパスワードを容易に管理できます。

Oracle Advanced Security のシングル・サインオンでは、最初の接続時にユーザーに対する厳密な認証が行われ、以降の他のデータベースやサービスへの接続時は、この認証が透過的に行われます。シングル・サインオンを使用すると、ユーザーは1つのパスワードで複数のアカウントとアプリケーションにアクセスできます。Oracle Advanced Security では、Kerberos と CyberSafe も含めて、多くのシングル・サインオンの形式をサポートしています。

また、Oracle Advanced Security では、**LDAP v3** 準拠のディレクトリ・サービスを組み込むことにより、Oracle ユーザー用の SSL ベースのシングル・サインオンもサポートしています。統合したディレクトリ・サービスと Oracle の PKI 実装を組み合わせ、SSL ベースのシングル・サインオンを Oracle9i データベースで使用できます。シングル・サインオンによって一度ユーザーの認証が行われると、以降の接続ではユーザーのデジタル証明書が利用されます。

これによって、ユーザーの使い勝手が向上し、セキュリティ管理者の集中管理が可能となります。

認可

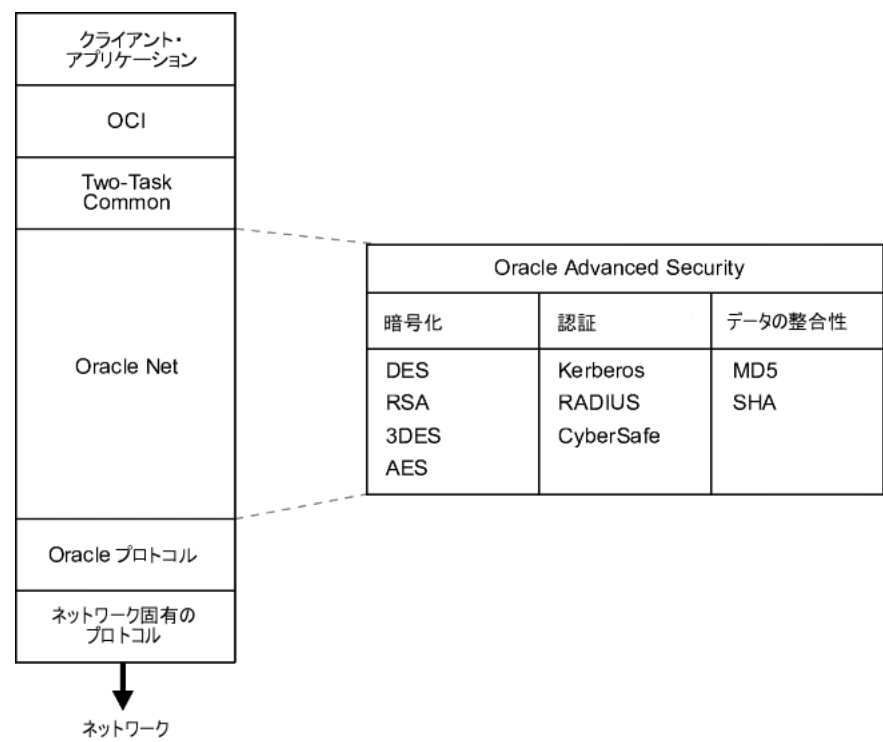
ユーザー認証（Oracle9i のロールと権限を許可する機能）は、Oracle Advanced Security でサポートされる認証方式を使用して大幅に強化されています。たとえば、Solaris など、特定のオペレーティング・システムでは、Oracle Advanced Security は DCE による認証をサポートしています。

また、Oracle Advanced Security のエンタープライズ・ユーザー・セキュリティによっても認証がサポートされています（第 15 章「エンタープライズ・ユーザー・セキュリティの管理」を参照してください）。Oracle Advanced Security は、LDAP バージョン 3 準拠のディレクトリと統合して、ユーザーと認証を集中管理できます。Oracle Advanced Security ライセンスにより、認可の格納と取出しのみでなく、ユーザーの管理にも Oracle Internet Directory を配置する権利が与えられています。Oracle Internet Directory を他の目的で使用する場合は、別にライセンス契約を結ぶ必要があります。

Oracle Advanced Security のアーキテクチャ

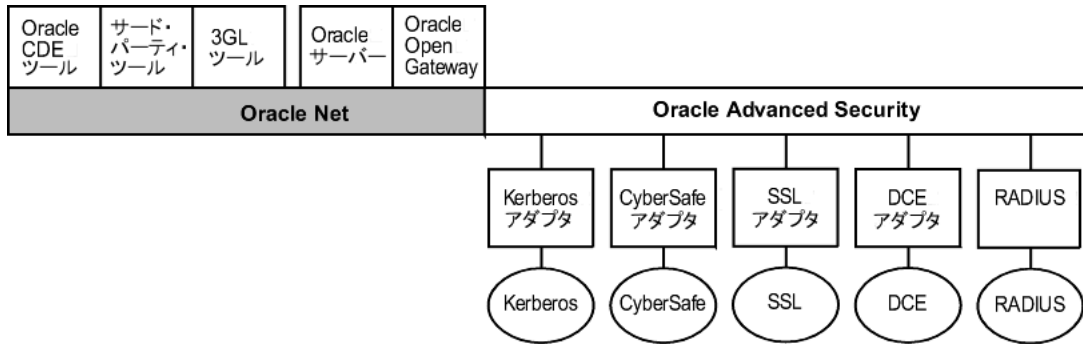
Oracle Advanced Security は、Oracle のサーバー・インストレーションまたはクライアント・インストレーションを補完するアドオン製品です。図 1-2 は、Oracle ネットワーク環境での Oracle Advanced Security のアーキテクチャを示しています。

図 1-2 Oracle ネットワーク環境での Oracle Advanced Security



Oracle Advanced Security では、既存の Oracle プロトコル・アダプタと同様のアダプタによって認証をサポートしています。図 1-3 に示すように、認証アダプタを Oracle Net インタフェースの下に統合することにより、既存のアプリケーションは新しい認証システムを透過的に利用できます。既存のアプリケーションを変更する必要はありません。

図 1-3 Oracle Net と認証アダプタ



関連項目： Oracle ネットワーク環境でのスタック通信の詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。

ネットワーク・プロトコル境界でのデータ転送の保護

Oracle Advanced Security は Oracle Connection Manager によって完全にサポートされているので、複数のネットワーク・プロトコル間で安全にデータを転送できます。たとえば、NetWare (SPX/IPX) のような LAN プロトコルを使用するクライアントは、LU6.2、TCP/IP、DECNet のような異なるネットワーク・プロトコルを使用する大型サーバーと、安全にデータを共有できます。ネットワーク・インフラストラクチャの弱点を補いながら、最大のパフォーマンスを実現するために、Connection Manager は復号化 / 再暗号化のコストと危険を回避して暗号化されたデータをプロトコル間で渡します。

システム要件

Oracle Advanced Security は、Oracle Net サーバーまたは Oracle Net クライアントにバンドルされているアドオン製品です。Oracle Advanced Security は、クライアントとサーバーの両方に対して購入し、インストールする必要があります。

Oracle Advanced Security リリース 2 (9.2) では、Oracle Net リリース 2 (9.2) が必須で、Oracle9i Enterprise Edition をサポートしています。表 1-3 は、その他のシステム要件の一覧です。

注意： Oracle Advanced Security は、Oracle9i Standard Edition では使用できません。

表 1-3 認証方式とシステム要件

認証方式	システム要件
Cybersafe Active Trust	<ul style="list-style-type: none">■ バージョン 1.1 以上の CyberSafe GSS ランタイム・ライブラリを、Oracle クライアントを実行するマシンと Oracle サーバーを実行するマシンの両方にインストールする必要があります。■ リリース 1.2 以上の Cybersafe Active Trust を、認証サーバーを実行する物理的に安全なマシンにインストールする必要があります。■ リリース 1.2 以上の Cybersafe Active Trust Client を、Oracle クライアントを実行するマシンにインストールする必要があります。
Kerberos	<ul style="list-style-type: none">■ MIT Kerberos バージョン 5、リリース 1.1。■ Kerberos 認証サーバーを、物理的に安全なマシンにインストールする必要があります。
RADIUS	<ul style="list-style-type: none">■ Internet Engineering Task Force (IETF) RFC #2138、Remote Authentication Dial In User Service (RADIUS) および RFC #2139 RADIUS Accounting の標準に準拠する RADIUS サーバー。■ 要求 / 応答認証を使用可能にするには、JavaSoft の Java Development Kit リリース 1.1 で指定されている Java Native Interface をサポートするオペレーティング・システムで RADIUS を実行する必要があります。
SSL	<ul style="list-style-type: none">■ Oracle Wallet Manager バージョン 2.1 と互換性のある Wallet。旧リリースの Oracle Wallet Manager で作成された Wallet は上位互換ではありません。

表 1-3 認証方式とシステム要件（続き）

認証方式	システム要件
Entrust/PKI	<ul style="list-style-type: none">■ Entrust/IPSEC Negotiator Toolkit Release 5.0.2■ Entrust/PKI 5.0.2

Oracle Advanced Security の制限

Oracle Applications では、Oracle Advanced Security の暗号化とデータの整合性をサポートしています。しかし、Oracle Advanced Security ではデータを安全に送信するために Oracle Net Services が必須となるため、Oracle の統合会計アプリケーション、人事管理アプリケーションおよび生産管理アプリケーションの一部を Microsoft Windows で実行する場合は、Oracle Advanced Security の外部認証機能を使用できません。これらの製品で Oracle Display Manager（ODM）を使用する部分では、Oracle Advanced Security を利用できません。これは、ODM で Oracle Net Services が使用されていないためです。

第 II 部

暗号化、整合性および JDBC

第 II 部では、既存の Oracle ネットワークに対してデータの暗号化および整合性を構成する方法と、Oracle Advanced Security の Java 実装について説明します。次の章で構成されています。

- [第 2 章「データの暗号化および整合性の構成」](#)
- [第 3 章「JDBC Thin のサポート」](#)

関連項目： Oracle オペレーティング・システム固有のマニュアル

データの暗号化および整合性の構成

この章では、Oracle Advanced Security における Oracle Net Services 固有のデータの暗号化と整合性の構成方法について説明します。次の項目について説明します。

- Oracle Advanced Security 暗号化
- Oracle Advanced Security データの整合性
- Diffie-Hellman ベースの鍵管理
- データの暗号化および整合性の構成

Oracle Advanced Security 暗号化

この項では、現行リリースの Oracle Advanced Security で使用できるデータ暗号化アルゴリズムについて説明します。

- [概要](#)
- [Advanced Encryption Standard](#)
- [DES アルゴリズムのサポート](#)
- [トリプル DES のサポート](#)
- [高速暗号化のための RSA RC4 アルゴリズム](#)

注意： Oracle Advanced Security の 8.1.7 より前のリリースでは、米国内版、アップグレード版および輸出版の 3 つのエディションがあり、それぞれ異なる鍵の長さを使用していました。このリリースには、これまで米国内版でのみ使用可能であった暗号化アルゴリズムと鍵の長さを完全に補完したものが組み込まれています。以前のバージョンの製品を使用しているユーザーは、個々の製品リリースに対応する米国内向けバージョンを入手できます。

概要

安全性の高い暗号システムの目的は、鍵を使用して **平文** データを解読不可能な **暗号文** に変換することにあります。暗号化は、正しい鍵を知らない限り、対応する平文に再変換できない（計算不可能な）方法で行われます。対称型暗号システムでは、同一データの暗号化と復号化に同一の鍵が使用されます。Oracle Advanced Security では、Oracle Net Services 通信の機密保護のために、DES、3DES および RC4 の対称型暗号システムを利用できます。

Advanced Encryption Standard

このリリースでは、米国連邦情報処理標準（Federal Information Processing Standard: FIPS）の新しい暗号化アルゴリズムである Advanced Encryption Standard（AES）がサポートされます。AES は、すべての米国政府組織およびビジネスでネットワーク上の機密データを保護するために使用できます。この暗号化アルゴリズムでは、3 種類の標準的な鍵の長さ（128 ビット、192 ビットおよび 256 ビット）が定義されています。すべてのバージョンとも外部 **暗号ブロック連鎖**（CBC）モードで動作します。

DES アルゴリズムのサポート

Oracle Advanced Security では、データ暗号化規格 (DES) アルゴリズムを使用できます。DES は長年に渡って米国政府の暗号化規格として利用され、金融業界では利用が義務付けられることもあります。長年にわたる規格であるため、DES は世界中で採用され、様々なアプリケーションで使用されています。

トリプル DES のサポート

Oracle Advanced Security は、トリプル DES 暗号化 (3DES) をサポートします。この暗号化では、メッセージ・データを DES アルゴリズムに 3 回渡すことによって暗号化を行います。3DES は高度なメッセージのセキュリティを実現しますが、パフォーマンスへの影響を伴います。影響の度合いは暗号化を実行するプロセッサの速度にもよりますが、通常、3DES では、データ・ブロックを暗号化するために、標準の DES アルゴリズムに比べて 3 倍の時間がかかります。

3DES には 2 つの鍵を使用する場合と 3 つの鍵を使用する場合があり、有効な鍵の長さは、それぞれ 112 ビットと 168 ビットです。両バージョンとも外部**暗号ブロック連鎖** (CBC) モードで動作します。

DES40 アルゴリズム

DES40 アルゴリズムは、Oracle Advanced Security、Oracle Advanced Networking Option および Secure Network Services のすべてのリリースで利用可能な DES の一種で、40 ビットの鍵を提供するためにその秘密鍵が事前に処理されています。DES40 は、米国の輸出法が現在よりも厳しかった頃に、米国とカナダ以外の地域の顧客に DES ベースの暗号化を提供するために設計されました。現在、Oracle Advanced Security リリース 2 (9.2) では、DES40、DES および 3DES のすべてを輸出用に使用できるようになりました。海外の顧客については、下位互換性を提供するために DES40 もサポートしています。

高速暗号化のための RSA RC4 アルゴリズム

RSA Data Security 社が開発した RC4 アルゴリズムは、データを高速で暗号化するアルゴリズムの国際標準になりました。RC4 は DES の数倍の速度で動作する可変長の鍵の長さのストリーム暗号なので、大量のデータ転送でもパフォーマンスへの影響を最小限に抑えて暗号化できます。

Oracle Advanced Security リリース 2 (9.2) では、40 ビット、56 ビット、128 ビットおよび 256 ビットの鍵の長さを持つ RC4 を実装できます。これによって、下位互換性と強力な暗号化を、パフォーマンスを犠牲にすることなく実現できます。

関連項目：

- 2-9 ページ「クライアントとサーバーでの暗号化の構成」
- 2-11 ページの表 2-2「有効な暗号化アルゴリズム」

Oracle Advanced Security データの整合性

ネットワーク・データの暗号化によってデータのプライバシーが守られるため、無許可の第三者はネットワーク上で転送中の平文データを見ることはできません。また、Oracle Advanced Security では、次の 2 つの形態の攻撃からデータを保護しています。

- データ変更攻撃

この形態の攻撃は、無許可の第三者が転送中のデータを傍受し、変更して再転送することで発生します。たとえば、銀行への 100 ドルの預け入れが傍受され、金額が 10,000 ドルに変更されて再送されます。

- 再生攻撃

この形態の攻撃は、有効なデータの全体が反復的に再送されることで発生します。たとえば、銀行からの 100 ドルの払い戻しが傍受され、その払い戻しが 100 回再送されて最終的に 10,000 ドルの払い戻し金額になります。

サポートしているデータの整合性アルゴリズム

Oracle Advanced Security では、鍵で順序付けられた Message Digest 5 (MD5) アルゴリズムまたは Secure Hash Algorithm (SHA-1) を使用して、これらの攻撃からデータを保護できます。これらのハッシュ・アルゴリズムは、データがなんらかの方法で変更された場合に変わるチェックサムを作成します。この保護機能は暗号化プロセスとは独立して動作するため、暗号化の有無に関係なくデータの整合性を保つことができます。

関連項目：

- 2-12 ページ「クライアントとサーバーでの整合性の構成」
- 2-13 ページの表 2-3「有効な整合性アルゴリズム」

Diffie-Hellman ベースの鍵管理

暗号データの機密性は、通信関係者間で共有される秘密鍵の存在に依存しています。鍵は接続の両端の関係者間で排他的に共有する機密情報です。鍵がなければ、暗号メッセージを復号化したり、暗号チェックサム付きのメッセージをわからないように変更することは非常に困難（計算不可能）です。このような秘密鍵を提供および維持することを、鍵管理といいます。

マルチユーザー環境では、安全な鍵の配布が難しくなります。Oracle Advanced Security は、よく知られた **Diffie-Hellman 鍵折衝アルゴリズム** を使用して、暗号化およびデータの整合性の両面において安全な鍵の配布を実現しています。

暗号化を使用して暗号データを保護するときは、鍵を頻繁に変更して、鍵の安全性が損なわれた場合の影響を最小限に抑える必要があります。そのため、Oracle Advanced Security の鍵管理機能では、セッションごとにセッション鍵が変更されます。

認証鍵フォールドイン

認証鍵フォールドインを使用する目的は、Diffie-Hellman 鍵折衝に対する第三者の攻撃（介在者による攻撃）を阻止することです。この暗号化では、クライアントとサーバーのみが認識している共有シークレットを、Diffie-Hellman によって折衝される最初のセッション鍵と組み合わせることによって、セッション鍵の安全性を大幅に強化しています。

クライアントとサーバーは、Diffie-Hellman によって生成されるセッション鍵を使用して通信を開始します。クライアントは、サーバーに対して認証を行う場合、そのクライアントとサーバーのみが認識している共有シークレットを設定します。次に、Oracle Advanced Security は、その共有シークレットと Diffie-Hellman セッション鍵を組み合わせ、より強力なセッション鍵を生成します。このセッション鍵により、介在者による攻撃を阻止できます。

注意： 認証鍵フォールドイン機能は、Oracle Advanced Security に組み込まれているため、システム管理者またはネットワーク管理者による構成作業は必要ありません。

データの暗号化および整合性の構成

この項では、Oracle Advanced Security における Oracle Net Services 固有の暗号化と整合性の構成方法について説明します。ここでは、Oracle Net Services がインストール済みであると想定しています。

ネットワーク管理者またはセキュリティ管理者は、暗号化と整合性の構成パラメータを設定します。データの暗号化と整合性を使用するクライアントとサーバーのシステム上のプロファイル (sqlnet.ora) には、この項で説明するパラメータの一部またはすべてが含まれている必要があります。

- [暗号化および整合性をアクティブにする](#)
- [暗号化および整合性の指定](#)
- [暗号化シードの設定](#)
- [Oracle Net Manager を使用した暗号化および整合性パラメータの構成](#)

関連項目： 暗号化、整合性および認証のために SSL 機能を構成する方法は、[第 7 章「Secure Sockets Layer 認証の構成」](#)を参照してください。

暗号化および整合性をアクティブにする

すべてのネットワーク接続で、クライアントとサーバーの両方が複数の暗号化アルゴリズムと複数の整合性アルゴリズムをサポートできます。接続が確立されるときに、sqlnet.ora ファイルで指定されているアルゴリズムの中から、サーバーがどのアルゴリズムを使用するかを選択します。

サーバーは、クライアントとサーバーで使用可能なアルゴリズムの中から一致するアルゴリズムを探し、サーバー自体のリストとクライアントのリストの両方に含まれるアルゴリズムの中で最初アルゴリズムを選択します。接続の片側がアルゴリズムのリストを指定していない場合は、インストールされているすべてのアルゴリズムが使用可能になります。サーバーとクライアントの両側でインストールされていないアルゴリズムを指定すると、エラー・メッセージ ORA-12650 が表示され、接続が失敗します。

暗号化パラメータと整合性パラメータを定義するには、ネットワーク上のクライアントとサーバーの sqlnet.ora ファイルを変更します。

Oracle Advanced Security で使用可能な暗号化アルゴリズム ([表 2-2](#)) のいずれかまたはすべて、および使用可能な整合性アルゴリズム ([表 2-3](#)) の 1 つまたは両方を構成できます。各接続セッションに使用できるのは、1 つの暗号化アルゴリズムと 1 つの整合性アルゴリズムのみです。

注意： Oracle Advanced Security は、クライアントとサーバーで使用可能な最初の暗号化アルゴリズムおよび整合性アルゴリズムを選択します。優先する折衝の順にアルゴリズムと鍵の長さを選択することをお勧めします（つまり、多くの場合、最も強力な鍵の長さを最初に選択します）。

関連項目： [付録 A「データの暗号化と整合性のパラメータ」](#)

暗号化および整合性の指定

暗号化または整合性をオンにするかどうかを指定する場合、Oracle Advanced Security の暗号化と整合性の構成パラメータに対して 4 つの値のうちのいずれかを指定することができます。次に、その 4 つの値をセキュリティ・レベルの低い順にリストします。値 REJECTED はクライアントとサーバーの間の通信について最小レベルのセキュリティを提供し、値 REQUIRED は最高レベルのネットワーク・セキュリティを提供します。

- REJECTED
- ACCEPTED
- REQUESTED
- REQUIRED

各パラメータのデフォルト値は ACCEPTED です。

REJECTED

接続先から要求されてもセキュリティ・サービスを使用可能にしない場合は、この値を選択します。

この使用例では、接続元でセキュリティ・サービスを使用禁止に指定します。接続先が REQUIRED に設定されている場合は、エラー・メッセージ ORA-12650 が表示されて接続が終了します。接続先が REQUESTED、ACCEPTED または REJECTED に設定されている場合、エラーは発生せずに、セキュリティ・サービスがオフのまま接続が継続されます。

ACCEPTED

接続先から要求または依頼されたときにセキュリティ・サービスを使用可能にする場合は、この値を選択します。

この使用例では、接続元からはセキュリティ・サービスを要求しませんが、接続先が REQUIRED または REQUESTED に設定されている場合は、そのセキュリティ・サービスが使用可能になります。接続先が REQUIRED または REQUESTED に設定されていて、一致する暗号化または整合性アルゴリズムが見つかったら、エラーは発生せずに、セキュリティ・サービスがオンのまま接続が継続されます。接続先のパラメータが REQUIRED に設定され

ていて、一致するアルゴリズムが見つからない場合、エラー・メッセージ ORA-12650 が表示されて接続が終了します。

接続先が REQUESTED に設定されていて、一致するアルゴリズムが見つからない場合、または接続先が ACCEPTED または REJECTED に設定されている場合は、エラーは発生しないで、セキュリティ・サービスがオフのまま接続が継続されます。

REQUESTED

接続先でセキュリティ・サービスの使用を許可されているときにそのセキュリティ・サービスを使用可能にする場合は、この値を選択します。

この使用例では、接続元がセキュリティ・サービスは必要だが必須ではないことを指定します。接続先が ACCEPTED、REQUESTED または REQUIRED を指定すると、セキュリティ・サービスが使用可能になります。接続先が使用可能にしたアルゴリズムと一致するアルゴリズムが見つからない場合は、セキュリティ・サービスが使用可能になりません。接続先が REQUIRED を指定して、一致するアルゴリズムが見つからない場合は、接続が失敗します。

REQUIRED

セキュリティ・サービスを使用可能にする場合、または保護されていない接続を禁止する場合は、この値を選択します。

この使用例では、接続元がセキュリティ・サービスを使用可能にする必要があることを指定します。接続先が REJECTED を指定した場合、または接続先と互換性のあるアルゴリズムが見つからない場合は、接続が失敗します。

表 2-1 では、クライアントとサーバーの構成パラメータの組合せに基づいてセキュリティ・サービスが使用可能となる場合と使用可能とならない場合を示しています。サーバーまたはクライアントが REQUIRED を指定した場合は、共通のアルゴリズムが存在しないと接続が失敗します。サーバーもクライアントも REQUIRED を指定しないで、セキュリティ・サービスが使用可能になっている場合は、共通のサービス・アルゴリズムが存在しないとサービスが使用禁止になります。

表 2-1 暗号化とデータの整合性の指定

	クライアント				
		REJECTED	ACCEPTED	REQUESTED	REQUIRED
サーバー	REJECTED	OFF	OFF	OFF	接続が失敗
	ACCEPTED	OFF	OFF ¹	ON	ON
	REQUESTED	OFF	ON	ON	ON
	REQUIRED	接続が失敗	ON	ON	ON

¹ この値は OFF にデフォルト設定されています。ユーザーが Oracle Net Manager を使用するか、`sqlnet.ora` ファイルを変更することによってこのパラメータを変更しないかぎり、暗号化とデータの整合性は使用可能になりません。

暗号化シードの設定

クライアントとサーバーで乱数を生成するには、3 つのシードを使用します。シードの 1 つはユーザー定義の暗号化シード (`sqlnet.crypto_seed=`) で、10 ～ 70 文字を指定でき、この指定はいつでも変更できます。Diffie-Hellman 鍵変換では乱数を使用して、各接続セッションに固有のセッション鍵を生成します。

Oracle Net Manager を使用した暗号化および整合性パラメータの構成

Oracle Net Manager を使用して、暗号化および整合性パラメータを設定または変更できます。この項では、次の項目について説明します。

- クライアントとサーバーでの暗号化の構成
- クライアントとサーバーでの整合性の構成

関連項目：

- 有効な暗号化アルゴリズムについては、付録 A「データの暗号化と整合性のパラメータ」を参照してください。
- 構成情報の詳細は、Oracle Net Manager のオンライン・ヘルプを参照してください。

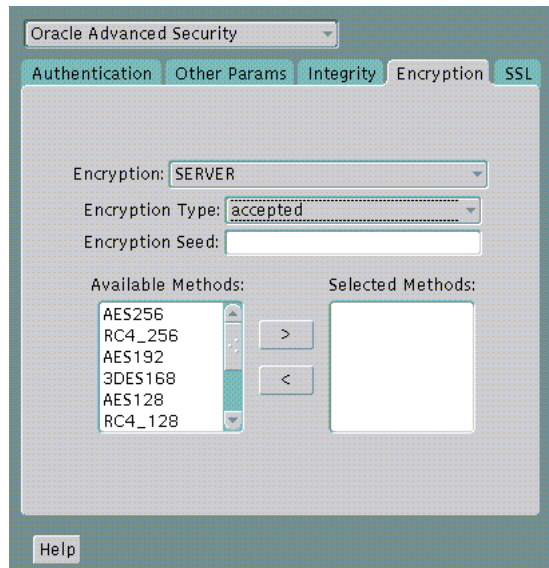
クライアントとサーバーでの暗号化の構成

クライアントとサーバーで暗号化を構成する手順は次のとおりです。

1. Oracle Net Manager を起動します。
 - UNIX の場合は、`$ORACLE_HOME/bin` から `netmgr` を実行します。

- Windows NT の場合は、「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Network Administration」→「Oracle Net Manager」の順に選択します。
2. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。
3. 右側のウィンドウ・ペインのリストから、「Oracle Advanced Security」を選択します。「Oracle Advanced Security」タブ・ウィンドウが表示されます (図 2-1)。

図 2-1 Oracle Advanced Security の「Encryption」ウィンドウ



4. 「Encryption」タブを選択します。
5. 構成するシステムに応じて、「Encryption」プルダウン・リストから「CLIENT」または「SERVER」を選択します。
6. 「Encryption Type」リストから、次のいずれかを選択します。
 - REQUESTED
 - REQUIRED
 - ACCEPTED
 - REJECTED
7. 「Encryption Seed」フィールドで、10 ～ 70 文字のランダムな文字を入力します。クライアントの暗号化シードはサーバーの暗号化シードとは別のものにします。

8. 「Available Methods」リストで暗号化アルゴリズムを選択します。次に、右矢印「>」をクリックして「Selected Methods」リストに移動します。追加するメソッドすべてに対して同じ作業を繰り返します。
9. 「File」→「Save Network Configuration」を選択します。sqlnet.ora ファイルが更新されます。
10. この手順を繰り返して、もう一方のシステムの暗号化を構成します。2つのシステムのsqlnet.ora ファイルに、次のエントリが含まれている必要があります。

- サーバー

```
SQLNET.ENCRYPTION_SERVER = [accepted | rejected | requested | required]
SQLNET.ENCRYPTION_TYPES_SERVER = (valid_encryption_algorithm [,valid_encryption_algorithm])
SQLNET.CRYPTO_SEED = "10-70 random characters"
```

- クライアント

```
SQLNET.ENCRYPTION_CLIENT = [accepted | rejected | requested | required]
SQLNET.ENCRYPTION_TYPES_CLIENT = (valid_encryption_algorithm [,valid_encryption_algorithm])
SQLNET.CRYPTO_SEED = "10-70 random characters"
```

有効な暗号化アルゴリズムと対応する有効値を表 2-2 にまとめます。

表 2-2 有効な暗号化アルゴリズム

アルゴリズム名	有効値
RC4 256 ビット鍵	RC4_256
RC4 128 ビット鍵	RC4_128
RC4 56 ビット鍵	RC4_56
RC4 40 ビット鍵	RC4_40
AES 256 ビット鍵	AES256
AES 192 ビット鍵	AES192
AES 128 ビット鍵	AES128
3つの鍵を使用する 3DES	3DES168
2つの鍵を使用する 3DES	3DES112

表 2-2 有効な暗号化アルゴリズム（続き）

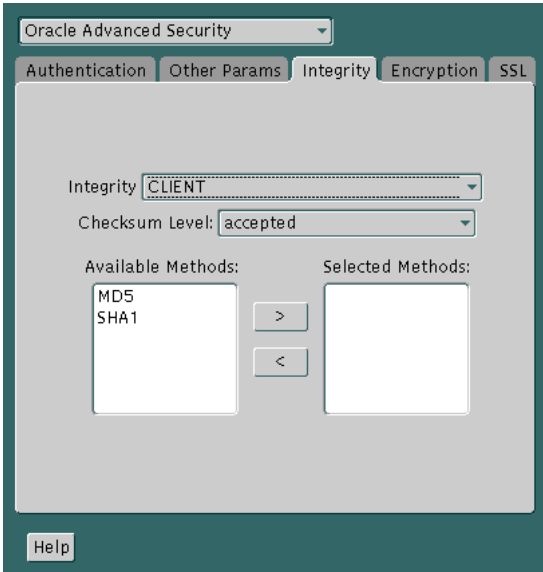
アルゴリズム名	有効値
DES 56 ビット鍵	DES
DES 40 ビット鍵	DES40

クライアントとサーバーでの整合性の構成

クライアントとサーバーでデータの整合性を構成する手順は次のとおりです。

1. Oracle Net Manager を起動します。
 - UNIX の場合は、\$ORACLE_HOME/bin から netmgr を実行します。
 - Windows NT の場合は、「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Network Administration」→「Oracle Net Manager」の順に選択します。
2. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。
3. 右側のウィンドウ・ペインのリストから、「Oracle Advanced Security」を選択します。「Oracle Advanced Security」タブ・ウィンドウが表示されます（図 2-2）。

図 2-2 Oracle Advanced Security の「Integrity」ウィンドウ



4. 「Integrity」タブを選択します。

- 5. 構成するシステムによって、「Integrity」リストの「CLIENT」または「SERVER」を選択します。
- 6. 「Checksum Level」リストから、次のいずれかのチェックサム・レベル値を選択します。
 - REQUESTED
 - REQUIRED
 - ACCEPTED
 - REJECTED
- 7. 「Available Methods」リストで整合性アルゴリズムを選択します。次に、右矢印「>」をクリックして「Selected Methods」リストに移動します。追加するメソッドすべてに対して同じ作業を繰り返します。
- 8. 「File」→「Save Network Configuration」を選択します。sqlnet.ora ファイルが更新されます。
- 9. この手順を繰り返して、もう一方のシステムの整合性を構成します。2つのシステムのsqlnet.ora ファイルに、次のエントリが含まれている必要があります。
 - サーバー

```
SQLNET.CRYPTO_CHECKSUM_SERVER = [accepted | rejected | requested | required]
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER = (valid_crypto_checksum_algorithm
[,valid_crypto_checksum_algorithm])
```
 - クライアント

```
SQLNET.CRYPTO_CHECKSUM_CLIENT = [accepted | rejected | requested | required]
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT = (valid_crypto_checksum_algorithm
[,valid_crypto_checksum_algorithm])
```

有効な整合性アルゴリズムと対応する有効値を表 2-3 に示します。

表 2-3 有効な整合性アルゴリズム

アルゴリズム名	有効値
MD5	MD5
SHA-1	SHA1

JDBC Thin のサポート

この章では、Oracle Advanced Security の Java 実装について説明します。この Java 実装によって、Java Database Connectivity (JDBC) Thin クライアントが Oracle9i データベースに安全に接続できます。次の項目について説明します。

- [Java 実装について](#)
- [構成パラメータ](#)

関連項目： JDBC の詳細（例を含む）は、『Oracle9i JDBC 開発者ガイド およびリファレンス』を参照してください。

Java 実装について

Oracle Advanced Security の Java 実装は、Oracle Advanced Security が組み込まれた Oracle9i データベースと通信する JDBC Thin クライアントに対して、ネットワークの暗号化と整合性の保護機能を提供します。

この項では、次の項目について説明します。

- [Java Database Connectivity のサポート](#)
- [JDBC Thin の保護](#)
- [実装の概要](#)
- [不明瞭化](#)

Java Database Connectivity のサポート

業界標準の Java インタフェースである [Java Database Connectivity \(JDBC\)](#) は、Java プログラムからリレーショナル・データベースに接続するための Java 標準です。Sun Microsystems では JDBC 標準を定義していますが、オラクル社では JDBC 標準をその独自の JDBC ドライバに実装または拡張しています。

Oracle JDBC ドライバは、JDBC アプリケーションを作成し、Oracle データベースと通信を行うために使用されます。Oracle には、2 つのタイプの JDBC ドライバが実装されています。C 言語ベースの Oracle Net クライアントの最上部に構築されている JDBC Thick ドライバと、ダウンロード可能なアプレットをサポートする JDBC Thin (Pure Java) ドライバです。JDBC に対する Oracle の拡張には、次の機能が含まれています。

- データ・アクセスおよび操作
- LOB アクセスおよび操作
- Oracle オブジェクト型マッピング
- オブジェクト参照アクセスおよび操作
- 配列アクセスおよび操作
- アプリケーション・パフォーマンスの機能強化

JDBC Thin の保護

JDBC Thin ドライバはインターネットで使用するダウンロード可能なアプレット用に設計されているため、Oracle では Thin クライアントを使用する Oracle Advanced Security の暗号化および整合性アルゴリズムを Java で 100% 実現するよう設計しています。Oracle Advanced Security では、JDBC Thin ドライバに対して次のような機能が提供されます。

- データ暗号化
- データの整合性チェック
- JDBC Thin クライアントから Oracle RDBMS への接続を保護する
- 安全な通信チャネルに対してデータを送信するアプレットを開発者が作成できるようにする
- Java Server Pages (JSP) を持つ中間層のサーバーから Oracle RDBMS への接続を保護する
- Oracle9i データベースから Oracle Advanced Security がインストールされている旧リリースの Oracle データベースへの接続の保護

Oracle JDBC Thin ドライバでは、認証のために Oracle O3LOGON プロトコルを実装しています。Oracle JDBC Thin ドライバでは、Oracle Advanced Security の SSL 実装や、RADIUS、Kerberos、SecurID などのサード・パーティ認証機能はサポートしていません。ただし、Oracle JDBC OCI (Thick) ドライバ・サポートでは、Thick クライアント・サポートと同様に、すべての Oracle Advanced Security 機能を実装しています。

Oracle Advanced Security は暗号化を継続し、C で書かれたアルゴリズムを使用して Oracle Net クライアントと Oracle サーバー間の Oracle Net Services の通信の整合性をチェックします。Oracle Advanced Security の Java 実装によって、次の暗号化アルゴリズムの Java バージョンが提供されます。

- RC4_256
- RC4_128
- RC4_56
- RC4_40
- DES56
- DES40

注意： Oracle Advanced Security では、DES が暗号ブロック連鎖 (CBC) モードで実行されます。

また、暗号的に安全なメッセージ・ダイジェストである Message Digest 5 (MD5) を使用して、JDBC Thin の整合性がチェックされます。

実装の概要

サーバー側では、アルゴリズムの折衝および鍵の生成は、Oracle Advanced Security 固有の暗号化とまったく同様に機能します。これにより、クライアントとサーバーの低位および上位互換性が維持されています。

クライアント側では、アルゴリズムの折衝および鍵の生成は、C ベースの Oracle Advanced Security 暗号化とまったく同じ方法で行われます。クライアントとサーバーは、従来の Oracle Net クライアントと同様の方法で、暗号化アルゴリズムを折衝し、乱数を生成し、Diffie-Hellman を使用してセッション鍵を交換し、Oracle Password Protocol (O3LOGON 鍵フォールドイン) を使用します。JDBC Thin には、Oracle Net クライアントが pure Java で完全に実装されています。

不明瞭化

Java の暗号化コードは、このリリースでは不明瞭化されています。つまり、暗号化および復号化機能を含んだ Java クラスおよびメソッドは、不明瞭化ソフトウェアを使用して保護されています。

Java バイト・コードの**不明瞭化**は、Java プログラムの形式で作成された知的所有物を保護するために企業がよく使用するプロセスです。このプロセスではコード内の Java シンボルが混ぜ合せられます。これにより、元のプログラムの構造は変えずに、その内容を隠すためにクラス、メソッドおよび変数の名前のみを変更して、プログラムが正しく実行されるようにしています。不明瞭化を行っていない Java コードをデコンパイルして読み取ることは可能ですが、不明瞭化された Java コードは、米国政府の輸出規制を満たすためにデコンパイルが困難になっています。

構成パラメータ

いくつかの構成パラメータを含むプロパティ・クラス・オブジェクトは、Oracle Advanced Security インタフェースに渡されます。この章では、次の項目に関する構成パラメータをリストします。

- クライアント暗号化レベル: ORACLE.NET.ENCRYPTION_CLIENT
- クライアントの暗号化選択リスト: ORACLE.NET.ENCRYPTION_TYPES_CLIENT
- クライアント整合性レベル: ORACLE.NET.CRYPTO_CHECKSUM_CLIENT
- クライアントの整合性選択リスト: ORACLE.NET.CRYPTO_CHEKSUM_TYPES_CLIENT

クライアント暗号化レベル: ORACLE.NET.ENCRYPTION_CLIENT

このパラメータは、クライアントがサーバーと折衝するセキュリティのレベルを定義します。表 3-1 は、このパラメータの属性を示しています。

表 3-1 ORACLE.NET.ENCRYPTION_CLIENT パラメータの属性

属性	説明
パラメータ・タイプ	String 型
パラメータ・クラス	静的
指定できる値	REJECTED、ACCEPTED、REQUESTED、REQUIRED
デフォルト値	ACCEPTED
構文	<code>up.put("oracle.net.encryption_client", level)</code>
例	<code>up.put("oracle.net.encryption_client", "REQUIRED")</code> 。この場合、up は <code>Properties up=new Properties()</code> として定義されています。

クライアントの暗号化選択リスト : ORACLE.NET.ENCRIPTION_TYPES_CLIENT

このパラメータは、使用する暗号化アルゴリズムを定義します。表 3-2 は、このパラメータの属性を示しています。

表 3-2 ORACLE.NET.ENCRIPTION_TYPES_CLIENT パラメータの属性

属性	説明
パラメータ・タイプ	String 型
パラメータ・クラス	静的
指定できる値	RC4_256、RC4_128、RC4_56、RC4_40、DES56C、DESC40C
構文	<code>up.put("oracle.net.encryption_types_client",alg)</code>
例	<code>up.put("oracle.net.encryption_types_client", "DESC40C")</code> 。この場合、 <code>up</code> は <code>Properties</code> <code>up=new Properties()</code> として定義されています。

注意： このコンテキストでは、「C」は CBC（暗号ブロック連鎖）モードのことを指しています。

クライアント整合性レベル : ORACLE.NET.CRYPTO_CHECKSUM_CLIENT

このパラメータは、データ整合性のためにクライアントがサーバーと折衝するセキュリティのレベルを定義します。表 3-3 は、このパラメータの属性を示しています。

表 3-3 ORACLE.NET.CRYPTO_CHECKSUM_CLIENT パラメータの属性

属性	説明
パラメータ・タイプ	String 型
パラメータ・クラス	静的
指定できる値	REJECTED、ACCEPTED、REQUESTED、REQUIRED
デフォルト値	ACCEPTED
構文	<code>up.put("oracle.net.crypto_checksum_client",level)</code>
例	<code>up.put("oracle.net.crypto_checksum_client", "REQUIRED")</code> 。この場合、 <code>up</code> は <code>Properties</code> <code>up=new Properties()</code> として定義されています。

クライアントの整合性選択リスト : ORACLE.NET.CRYPTO_CHEKSUM_TYPES_CLIENT

このパラメータは、使用するデータ整合性アルゴリズムを定義します。表 3-4 は、このパラメータの属性を示しています。

表 3-4 ORACLE.NET.CRYPTO_CHEKSUM_TYPES_CLIENT パラメータの属性

属性	説明
パラメータ・タイプ	String 型
パラメータ・クラス	静的
指定できる値	MD5
構文	<code>up.put("oracle.net. crypto_checksum_types_client",alg)</code>
例	<code>up.put("oracle.net. crypto_checksum_types_client","MD5")</code> 。この場合、up は <code>Properties up=new Properties()</code> として定義されていま す。

第 III 部

認証方式の構成

第 III 部では、既存の Oracle ネットワークで認証方式を構成する方法について説明します。次の章で構成されています。各章で、Oracle Advanced Security でサポートされている特定の認証方式について説明します。

- [第 4 章「RADIUS 認証の構成」](#)
- [第 5 章「CyberSafe 認証の構成」](#)
- [第 6 章「Kerberos 認証の構成」](#)
- [第 7 章「Secure Sockets Layer 認証の構成」](#)
- [第 8 章「Entrust 対応の SSL 認証の構成」](#)
- [第 9 章「複数の認証方式の構成」](#)

注意： Oracle Advanced Security リリース 2 (9.2) は、認証方式の動的なロードをサポートしています。これにより、使用する認証方式をインストール時にすべて指定する必要はなくなりました。Oracle Advanced Security の初期インストールが完了した後であれば、いつでも使用可能な任意の認証方式を実装できます。

RADIUS 認証の構成

この章では、Oracle9i データベースの Oracle Advanced Security で Remote Authentication Dial-In User Service (RADIUS) を使用できるように構成する方法について説明します。次の項目について説明します。

- [RADIUS の概要](#)
- [RADIUS 認証モード](#)
- [RADIUS 認証、認可およびアカウントを使用可能にする](#)
- [RADIUS を使用したデータベースへのログイン](#)
- [RSA ACE/Server 構成チェックリスト](#)

注意： RSA Security 社の認証製品である SecurID は、Oracle Advanced Security で直接サポートされていませんが、RADIUS 準拠製品として認定されています。したがって、RADIUS のもとで SecurID を実行できます。

詳細は、RSA Security 社の SecurID のマニュアルを参照してください。

RADIUS の概要

RADIUS は広く使用されているクライアント / サーバー・セキュリティ・プロトコルであり、リモート認証とリモート・アクセスを実現します。Oracle Advanced Security では、この業界標準をクライアント / サーバー・ネットワーク環境で使用しています。

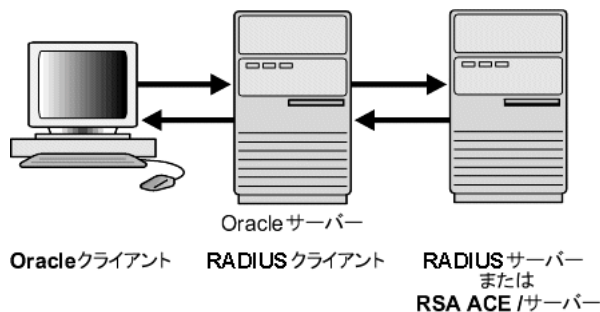
RADIUS のプロトコルをインストールして構成すると、トークン・カードやスマートカードなど、RADIUS 標準をサポートする認証方式をネットワークで使用できます。さらに、RADIUS を使用すると、Oracle クライアントまたは Oracle データベース・サーバーを変更することなく、認証方式を変更できます。

ユーザーの立場で見ると、認証手続き全体が透過的です。ユーザーが Oracle データベース・サーバーへのアクセスを要求すると、RADIUS クライアントとして動作する Oracle データベース・サーバーが RADIUS サーバーに通知します。RADIUS サーバーは次のように動作します。

- ユーザーのセキュリティ情報を検索します。
- 適切な認証サーバー（複数可）と Oracle データベース・サーバーの間で認証と認可情報を渡します。
- Oracle データベース・サーバーへのユーザー・アクセスを許可します。
- セッション情報のログを記録します。ユーザーがいつ、どのくらいの頻度で、どのくらいの時間、Oracle データベース・サーバーに接続していたかなどを記録します。

Oracle/RADIUS 環境を図 4-1 に示します。

図 4-1 Oracle 環境での RADIUS



Oracle データベース・サーバーは RADIUS クライアントとして動作し、Oracle クライアントと RADIUS サーバーの間で情報の受渡しを行います。同様に、RADIUS サーバーは Oracle データベース・サーバーと適切な認証サーバー（複数可）の間で情報の受渡しを行います。認証の構成要素を表 4-1 にリストします。

表 4-1 RADIUS 認証構成要素

構成要素	格納される情報
Oracle クライアント	RADIUS を使用した通信のための構成設定
Oracle データベース・サーバー / RADIUS クライアント	Oracle クライアントと RADIUS サーバーの間で情報の受渡しを行うための構成設定 秘密鍵ファイル
RADIUS サーバー	全ユーザーの認証と認可情報 各クライアントの名前または IP アドレス 各クライアントの共有シークレット すでに認証されているユーザーが再接続せずに別のログイン・オプションを選択できる無制限の数のメニュー・ファイル
認証 1 つ以上のサーバー	パスワードや PIN など、使用中の認証方式に対応するユーザー認証情報 注意： RADIUS サーバーは認証サーバーにもなります。

RADIUS サーバーのベンダーが認証サーバーのベンダーでもある場合が多く、この場合は認証を RADIUS サーバー上で処理できます。たとえば、RSA ACE/Server は、RADIUS サーバーでもあり認証サーバーでもあります。このため、この製品は、ユーザーのパスワードを認証します。

関連項目： sqlnet.ora ファイルの詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。

RADIUS 認証モード

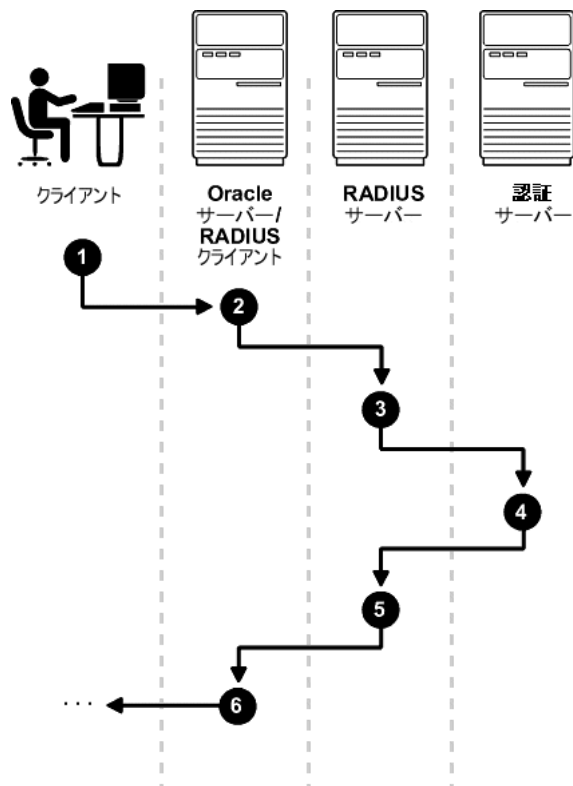
ユーザー認証には次の 2 通りあります。

- 同期認証モード
- 要求 / 応答（非同期）認証モード

同期認証モード

同期モードの RADIUS では、パスワードや SecurID トークン・カードなど、各種の認証方式を使用することができます。図 4-2 は、同期認証が発生する順序を示しています。

図 4-2 同期認証シーケンス



1. ユーザーは接続文字列、パスコード、またはその他の値を入力してログインします。クライアント・マシンは、このデータを Oracle サーバーに渡します。
2. RADIUS クライアントとして動作している Oracle データベース・サーバーは、Oracle クライアントからのデータを RADIUS サーバーに渡します。
3. RADIUS サーバーはデータを検証するため、スマートカードや SecurID ACE などの適切な認証サーバーにデータを渡します。
4. 認証サーバーは、アクセス受入れメッセージまたはアクセス拒否メッセージを RADIUS サーバーに返します。
5. RADIUS サーバーは、この応答を Oracle データベース・サーバー /RADIUS クライアントに渡します。
6. Oracle データベース・サーバー /RADIUS クライアントは、Oracle クライアントに応答を渡します。

例 : SecurID トークン・カードによる同期認証

SecurID 認証では、各ユーザーがトークン・カードを持ち、カードに表示される動的番号は 60 秒ごとに変わります。Oracle データベース・サーバー /RADIUS クライアントにアクセスするために、ユーザーは個人識別番号 (PIN) とユーザーの SecurID カード上に表示されている動的番号を含む有効なパスコードを入力します。Oracle データベース・サーバーは、この認証情報を Oracle クライアントから RADIUS サーバー（この場合は、確認のための認証サーバー）に渡します。認証サーバー（RSA ACE/Server）によってユーザーが確認されると、受入れパケットが Oracle データベース・サーバーに送られ、次に、Oracle クライアントに渡されます。これでユーザーが認証され、適切な表やアプリケーションへのアクセスが可能となります。

関連項目 :

- [第 1 章「Oracle Advanced Security の概要」](#)
- [1-11 ページ「トークン・カード」](#)
- RSA Security 社提供のマニュアルも参照してください。

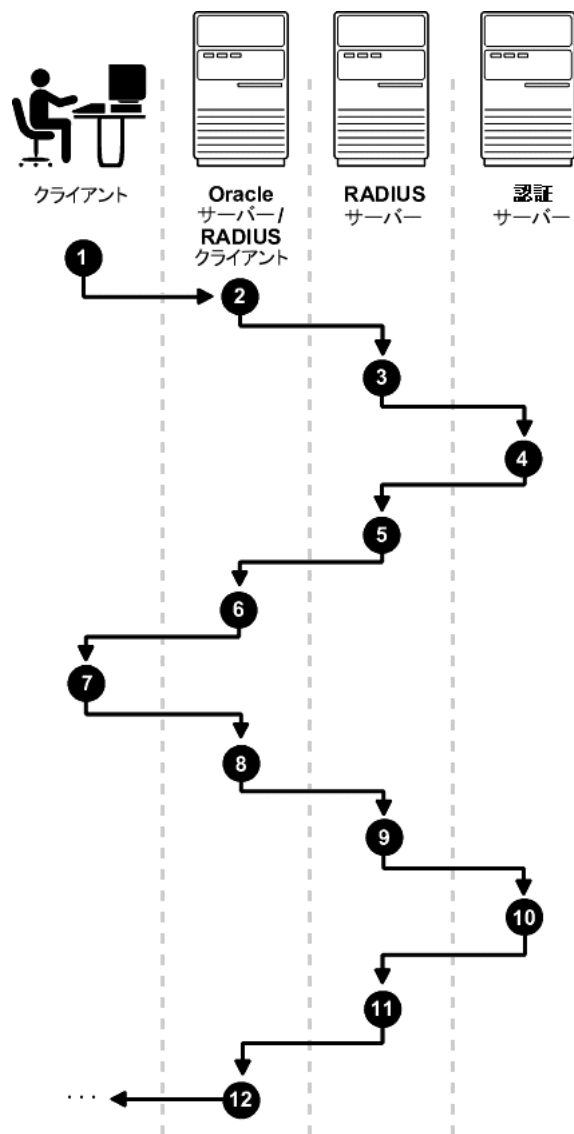
要求 / 応答（非同期）認証モード

システムで非同期モードを使用している場合は、ユーザーは SQL*Plus CONNECT 文字列でユーザー名とパスワードを入力する必要がありません。そのかわりに、プロセスの後半で、グラフィカル・ユーザー・インタフェースを使用して入力します。

図 4-3 に、要求 / 応答認証（非同期認証）のシーケンスを示します。

注意： RADIUS サーバーが認証サーバーの場合、図 4-3 の手順 3、4 および 5 と、手順 9、10 および 11 がそれぞれに結合されます。

図 4-3 非同期認証シーケンス



1. ユーザーは、Oracle データベース・サーバーへの接続を要求します。クライアント・マシンは、データを Oracle データベース・サーバーに渡します。
2. RADIUS クライアントとして動作している Oracle データベース・サーバーは、Oracle クライアントからのデータを RADIUS サーバーに渡します。
3. RADIUS サーバーは、スマートカード、SecurID ACE、トークン・カードなどの適切な認証サーバーにデータを渡します。
4. 認証サーバーはランダム番号などの要求を RADIUS サーバーに送ります。
5. RADIUS サーバーは、この要求を Oracle データベース・サーバー /RADIUS クライアントに送ります。
6. Oracle データベース・サーバー /RADIUS クライアントは、この要求を Oracle クライアントに送ります。GUI によってユーザーに要求が表示されます。
7. ユーザーは要求に対する応答を入力します。応答を生成するために、たとえば、ユーザーは受け取った要求をトークン・カードに入力できます。これにより、GUI に入力する動的なパスワードをトークン・カードから取得できます。Oracle クライアントは、Oracle データベース・サーバー /RADIUS クライアントにユーザーの応答を渡します。
8. Oracle データベース・サーバー /RADIUS クライアントが、ユーザーの応答を RADIUS サーバーに送ります。
9. RADIUS サーバーは、ユーザーの応答を検証するため、適切な認証サーバーに渡します。
10. 認証サーバーは、アクセス受入れメッセージまたはアクセス拒否メッセージを RADIUS サーバーに返します。
11. RADIUS サーバーは、応答を Oracle データベース・サーバー /RADIUS クライアントに渡します。
12. Oracle データベース・サーバー /RADIUS クライアントは、Oracle クライアントに応答を渡します。

例：スマートカードによる非同期認証

スマートカード認証では、ユーザーはスマートカード（クレジット・カードのような形で、情報を格納するための IC が埋め込まれたプラスチック・カード）を読み取り用のハードウェア・デバイスに挿入することによってログインします。Oracle クライアントは、このスマートカードに格納されているログイン情報を Oracle データベース・サーバー /RADIUS クライアントと RADIUS サーバーを経由して認証サーバーに送ります。認証サーバーは、RADIUS サーバーと Oracle データベース・サーバーを経由して Oracle クライアントに要求を戻し、認証情報を入力するようユーザーに求めます。この認証情報には、PIN やスマートカードに格納されている他の認証情報を使用することができます。

Oracle クライアントは、Oracle データベース・サーバーと RADIUS サーバーを経由して、ユーザーの応答を認証サーバーに送ります。ユーザーが入力した番号が有効である場合、認証サーバーは RADIUS サーバーと Oracle データベース・サーバーを経由して、受入れパ

ケットを Oracle クライアントに戻します。これでユーザーが認証され、適切な表やアプリケーションへのアクセスが認可されます。ユーザーが入力した情報が正しくない場合は、認証サーバーはユーザーのアクセスを拒否するメッセージを戻します。

例 : ActivCard トークンによる非同期認証

ActivCard トークンの 1 つに、キーパッドを装備した、動的パスワードを表示できる携帯式のデバイスがあります。Oracle データベース・サーバーへのアクセスを要求するため、ユーザーがパスワードを入力すると、その情報は Oracle データベース・サーバー /RADIUS クライアントと RADIUS サーバーを経由して、適切な認証サーバーに渡されます。認証サーバーは、RADIUS サーバーと Oracle データベース・サーバーを経由して、クライアントに要求を戻します。ユーザーが要求をトークンに入力すると、ユーザーが応答として返送する番号がトークンに表示されます。

Oracle クライアントは、Oracle データベース・サーバーと RADIUS サーバーを経由して、ユーザーの応答を認証サーバーに送ります。ユーザーが入力した番号が有効である場合、認証サーバーは RADIUS サーバーと Oracle データベース・サーバーを経由して、受入れパケットを Oracle クライアントに戻します。これでユーザーが認証され、適切な表やアプリケーションへのアクセスが認可されます。ユーザーが入力した応答が正しくない場合は、認証サーバーはユーザーのアクセスを拒否するメッセージを戻します。

RADIUS 認証、認可およびアカウントを使用可能にする

RADIUS 認証およびアカウントを使用可能にするには、次の作業を実行します。

- [タスク 1: Oracle データベース・サーバーと Oracle クライアントに RADIUS をインストールする](#)
- [タスク 2: RADIUS 認証の構成](#)
- [タスク 3: ユーザーの作成とアクセス権の付与](#)
- [タスク 4: 外部 RADIUS 認可の構成 \(オプション\)](#)
- [タスク 5: RADIUS アカウントの作成](#)
- [タスク 6: RADIUS クライアント名の RADIUS サーバー・データベースへの追加](#)
- [タスク 7: RADIUS とともに使用する認証サーバーの構成](#)
- [タスク 8: 認証サーバーとともに使用する RADIUS の構成](#)
- [タスク 9: マッピング・ロールの構成](#)

タスク 1: Oracle データベース・サーバーと Oracle クライアントに RADIUS をインストールする

RADIUS は、Oracle9i の通常のインストール時に、Oracle Advanced Security とともにインストールされます。

関連項目： Oracle Advanced Security と RADIUS アダプタのインストールの詳細は、オペレーティング・システム固有の Oracle9i インストレーション・ガイドを参照してください。

タスク 2: RADIUS 認証の構成

この作業には、次の手順があります。

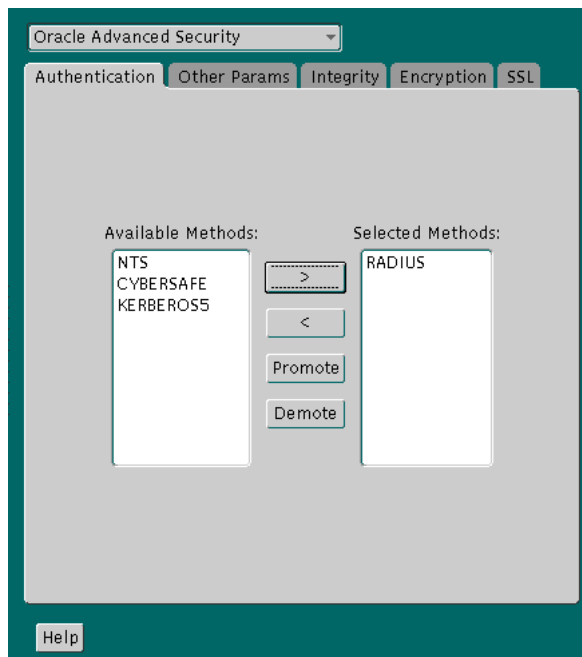
- [手順 1: Oracle クライアントでの RADIUS の構成](#)
- [手順 2: Oracle データベース・サーバーでの RADIUS 構成](#)
- [手順 3: その他の RADIUS 機能の構成](#)

特に指示されていないかぎり、これらの構成作業を行うには、Oracle Net Manager を使用するか、テキスト・エディタを使用して sqlnet.ora ファイルを変更します。

手順 1: Oracle クライアントでの RADIUS の構成

1. Oracle Net Manager を起動します。
 - UNIX の場合は、\$ORACLE_HOME/bin から netmgr を実行します。
 - Windows NT の場合は、「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Network Administration」→「Oracle Net Manager」の順に選択します。
2. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。
3. 右側のウィンドウ・ペインのリストから、「Oracle Advanced Security」を選択します。「Oracle Advanced Security」タブ・ウィンドウが表示されます (図 4-4)。

図 4-4 Oracle Advanced Security の「Authentication」ウィンドウ



4. 「Authentication」タブを選択します。
5. 「Available Methods」リストから「RADIUS」を選択します。
6. 右矢印「>」をクリックして、「RADIUS」を「Selected Methods」リストに移動します。他に使用するメソッドがあれば同じ方法で移動します。
7. 「Selected Methods」リストでメソッドを選択し、そのリストのその位置に対して「Promote」または「Demote」をクリックして、選択したメソッドを使用する順に並べます。たとえば、RADIUS が最初に使用するサービスとなるようにをリストの一番上に置きます。
8. 「File」→「Save Network Configuration」を選択します。

sqlnet.ora ファイルが更新され、次のエントリが追加されます。

```
SQLNET.AUTHENTICATION_SERVICES=(RADIUS)
```

手順 2: Oracle データベース・サーバーでの RADIUS 構成

- Oracle データベース・サーバーに RADIUS 秘密鍵ファイルを作成
- サーバー (sqlnet.ora ファイル) での RADIUS パラメータの構成
- Oracle データベース・サーバー初期化パラメータの設定

Oracle データベース・サーバーに RADIUS 秘密鍵ファイルを作成

1. RADIUS サーバーから RADIUS 秘密鍵を取得します。共有秘密鍵は、RADIUS サーバーの管理者が RADIUS クライアントごとに作成します。鍵の長さは、必ず 17 文字以上にする必要があります。
2. Oracle データベース・サーバーに、ディレクトリ \$ORACLE_HOME/network/security (UNIX の場合) または %ORACLE_HOME%\network\security (Windows NT の場合) を作成します。
3. RADIUS サーバーからコピーした共有シークレットを格納するために、radius.key ファイルを作成します。このファイルを手順 2 で作成したディレクトリに入れます。
4. 共有秘密鍵をコピーし、この鍵のみを、Oracle データベース・サーバーに作成した radius.key ファイルに貼り付けます。
5. セキュリティを確保するために、radius.key のファイル・パーミッションを読取り専用に変更し、Oracle の所有者のみがアクセスできるようにします (Oracle は、このファイルの機密保護をファイル・システムの機能に依存します)。

関連項目： 秘密鍵の取得方法は、RADIUS サーバーの管理マニュアルを参照してください。

サーバー (sqlnet.ora ファイル) での RADIUS パラメータの構成

1. Oracle Net Manager を起動します。
 - UNIX の場合は、\$ORACLE_HOME/bin から netmgr を実行します。
 - Windows NT の場合は、「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Network Administration」→「Oracle Net Manager」の順に選択します。
2. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。
3. 右側のウィンドウ・ペインのリストから、「Oracle Advanced Security」を選択します。「Oracle Advanced Security」タブ・ウィンドウが表示されます (図 4-4)。
4. 「Authentication」タブを選択します。
5. 「Available Methods」リストから「RADIUS」を選択します。
6. 右矢印「>」をクリックして、「RADIUS」を「Selected Methods」リストに移動します。

7. 選択した方式を使用優先順位の高い順に並べます。「Selected Methods」リストで方式を選択し、「Promote」または「Demote」をクリックして並べ替えます。たとえば、最初に使用するサービスを RADIUS にするには、リストの先頭に移動します。
8. 「Other Params」タブを選択します。「Other Params」ウィンドウが表示されます (図 4-5)。

図 4-5 Oracle Advanced Security の「Other Params」ウィンドウ

Oracle Advanced Security

Authentication Other Params Integrity Encryption SSL

Authentication Service: RADIUS

Host Name: localhost

Port Number: 1645

Timeout (seconds): 15

Number of Retries: 3

Secret File: /vobs/oracle/network/s

Send Accounting: OFF

Challenge Response: OFF

Default Keyword: challenge

Interface Class Name: DefaultRadiusInterface

Help

9. 「Authentication Service」リストから「RADIUS」を選択します。
10. 「Host Name」フィールドで、デフォルトのプライマリ RADIUS サーバーとして「localhost」をそのまま使用するか、または別のホスト名を入力します。
11. 「Secret File」フィールドのデフォルト値が有効であるか確認してください。
12. 「File」→「Save Network Configuration」を選択します。

sqlnet.ora ファイルが更新され、次のエントリが追加されます。

```
SQLNET.AUTHENTICATION_SERVICES=RADIUS
```

```
SQLNET.RADIUS_AUTHENTICATION=RADIUS_server_{hostname|IP_address}
```

Oracle データベース・サーバー初期化パラメータの設定

次の位置に格納されている初期化パラメータ・ファイルを構成します。

- UNIX の場合は、`$ORACLE_BASE/admin/db_name/pfile`
- Windows NT の場合は、`%ORACLE_BASE%\admin\%db_name%\pfile`

次の値を初期化パラメータ・ファイルに設定します。

```
REMOTE_OS_AUTHENT=FALSE
OS_AUTHENT_PREFIX=""
```

注意： `REMOTE_OS_AUTHENT` を `TRUE` に設定すると、非保護プロトコル (TCP など) を使用するユーザーがオペレーティング・システム許可ログイン (以前の OPS\$ ログイン) を実行できるので、セキュリティが侵害されるおそれがあります。

関連項目： Oracle9i データベース・サーバーにおける初期化パラメータの設定方法は、『Oracle9i データベース・リファレンス』および『Oracle9i データベース管理者ガイド』を参照してください。

手順 3: その他の RADIUS 機能の構成

- [デフォルト設定の変更](#)
- [要求 / 応答の構成](#)
- [代替 RADIUS サーバーのパラメータの設定](#)

デフォルト設定の変更

1. Oracle Net Manager を起動します。
 - UNIX の場合は、`$ORACLE_HOME/bin` から `netmgr` を実行します。
 - Windows NT の場合は、「スタート」→「プログラム」→「Oracle-*HOME_NAME*」→「Network Administration」→「Oracle Net Manager」の順に選択します。
2. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。
3. 右側のウィンドウ・ペインのリストから、「Oracle Advanced Security」を選択します。「Oracle Advanced Security」タブ・ウィンドウが表示されます (図 4-5)。
4. 「Other Params」タブを選択します。
5. 「Authentication Service」リストから「RADIUS」を選択します。
6. 次のいずれかのフィールドに対するデフォルト設定を変更します。

フィールド	説明
Port Number	プライマリ RADIUS サーバーのリスニング・ポートを指定します。デフォルト値は 1645 です。
Timeout (seconds)	プライマリ RADIUS サーバーに対する Oracle データベース・サーバーの応答待ち時間を指定します。デフォルトは 15 秒です。
Number of Retries	<p>プライマリ RADIUS サーバーに対する Oracle データベース・サーバーのメッセージ再送回数を指定します。デフォルトの再試行の回数は 3 回です。</p> <p>RADIUS アカウントの構成方法は、4-20 ページの「タスク 5: RADIUS アカウントの作成」を参照してください。</p>
Secret File	<p>Oracle データベース・サーバー上の秘密鍵の場所を指定します。このフィールドは、秘密鍵自体ではなく、秘密鍵ファイルの場所を指定します。</p> <p>秘密鍵の指定方法は、4-12 ページの「Oracle データベース・サーバーに RADIUS 秘密鍵ファイルを作成」を参照してください。</p>

7. 「File」→「Save Network Configuration」を選択します。

sqlnet.ora ファイルが更新され、次のエントリが追加されます。

```
SQLNET.RADIUS_AUTHENTICATION_PORT=(PORT)
SQLNET.RADIUS_AUTHENTICATION_TIMEOUT=
(NUMBER OF SECONDS TO WAIT FOR response)
SQLNET.RADIUS_AUTHENTICATION_RETRIES=
(NUMBER OF TIMES TO RE-SEND TO RADIUS server)
SQLNET.RADIUS_SECRET=(path/radius.key)
```

要求 / 応答の構成

要求 / 応答（非同期）モードでは、グラフィカル・インタフェースを表示してパスワードを最初に要求し、ユーザーがトークン・カードから取得する動的パスワードなど、他の追加情報を要求します。RADIUS アダプタにより、最適なプラットフォーム非依存の状態を得るためにこのインタフェースは Java ベースとなっています。

注意： 認証デバイスのサード・パーティ・ベンダーは、そのデバイスに適したグラフィカル・ユーザー・インタフェースにカスタマイズする必要があります。たとえば、スマートカード・ベンダーは Oracle クライアントがスマートカードから動的パスワードなどのデータを読み取れるように、Java インタフェースをカスタマイズします。スマートカードは、要求を受け取ると、PIN などの情報を入力するようにユーザーに求めて、応答します。

関連項目： 要求 / 応答ユーザー・インタフェースのカスタマイズ方法は、[付録 C「RADIUS による認証デバイスの統合」](#)を参照してください。

要求 / 応答を構成する手順は、次のとおりです。

1. JDK 1.1.7 または JRE 1.1.7 を使用する場合は、環境変数 `JAVA_HOME` を使用して、Oracle クライアントを実行するシステムの JRE または JDK のある場所を設定します。
 - UNIX の場合は、プロンプトで次のコマンドを入力します。

```
% setenv JAVA_HOME /usr/local/packages/jre1.1.7B
```
 - Windows NT の場合は、「スタート」→「設定」→「コントロール パネル」→「システム」→「環境」の順に選択し、環境変数 `JAVA_HOME` を次のように設定します。

```
c:¥java¥jre1.1.7B
```

注意： この手順は、JDK/JRE の他のバージョンでは必要ありません。

2. Oracle Net Manager を起動します。
 - UNIX の場合は、`$ORACLE_HOME/bin` から `netmgr` を実行します。
 - Windows NT の場合は、「スタート」→「プログラム」→「Oracle - *HOME_NAME*」→「Network Administration」→「Oracle Net Manager」の順に選択します。
3. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。
4. 右側のウィンドウ・ペインのリストから、「Oracle Advanced Security」を選択します。「Oracle Advanced Security」タブ・ウィンドウが表示されます ([図 4-5](#))。
5. 「Other Params」タブを選択します。
6. 「Authentication Service」リストから「RADIUS」を選択します。
7. 「Challenge Response」フィールドで、「ON」と入力して、要求 / 応答を使用可能にします。
8. 「Default Keyword」フィールドで、要求のデフォルト値をそのまま使用するか、RADIUS サーバーから要求を依頼するためのキーワードを入力します。

注意： キーワード機能は Oracle が提供するもので、すべての RADIUS サーバーでサポートされているとは限りません。この機能は、RADIUS サーバーでサポートされている場合にのみ使用することができます。

キーワードを設定すると、ユーザーはパスワードを使用せずに識別情報を検証されるようになります。ユーザーがパスワードを入力しなかった場合は、ここで設定したキーワードが RADIUS サーバーに渡され、RADIUS サーバーから運転免許証の番号や誕生日などの要求が戻されます。ユーザーがパスワードを入力した場合は、RADIUS サーバーの構成によって、要求が戻される場合と戻されない場合があります。

9. 「Interface Class Name」フィールドで、デフォルト値「DefaultRadiusInterface」をそのまま使用するか、要求 / 応答変換を処理するために作成したクラスの名前を入力します。デフォルトの RADIUS インタフェース以外を使用する場合は、sqlnet.ora ファイルを編集して、SQLNET.RADIUS_CLASSPATH=(location) を追加する必要があります。ここで、location は jar ファイルの完全なパス名です。このパラメータは、デフォルトでは次のように設定されます。

```
$ORACLE_HOME/network/jlib/netradius.jar:
$ORACLE_HOME/JRE/lib/vt.jar
```

10. 「File」→「Save Network Configuration」を選択します。

sqlnet.ora ファイルが更新され、次のエントリが追加されます。

```
SQLNET.RADIUS_CHALLENGE_RESPONSE=([ON | OFF])
SQLNET.RADIUS_CHALLENGE_KEYWORD=(KEYWORD)
SQLNET.RADIUS_AUTHENTICATION_INTERFACE=(name of interface including the package
name delimited by "/" for ".")
```

代替 RADIUS サーバーのパラメータの設定

代替 RADIUS サーバーを使用する場合は、テキスト・エディタを使用して sqlnet.ora ファイルに次のパラメータを設定します。

```
SQLNET.RADIUS_ALTERNATE=(hostname or ip address of alternate radius server)
SQLNET.RADIUS_ALTERNATE_PORT=(1812)
SQLNET.RADIUS_ALTERNATE_TIMEOUT=(number of seconds to wait for response)
SQLNET.RADIUS_ALTERNATE_RETRIES=(number of times to re-send to radius server)
```

タスク 3: ユーザーの作成とアクセス権の付与

ユーザーにアクセス権を付与する手順は、次のとおりです。

1. SQL*Plus を起動し、次のコマンドを実行して Oracle データベース・サーバーの外部で認証されるユーザーを作成し、アクセス権を付与します。

```
SQL> CONNECT system/manager@database_name;  
SQL> CREATE USER username IDENTIFIED EXTERNALLY;  
SQL> GRANT CREATE SESSION TO USER username;  
SQL> EXIT
```

Windows NT を使用する場合は、Oracle Enterprise Manager の Security Manager ツールを使用することもできます。

関連項目：

- 『Oracle9i データベース管理者ガイド』
- 『Oracle9i Heterogeneous Connectivity Administrator's Guide』

2. RADIUS サーバーの users ファイルにも同じユーザーを入力します。

関連項目： RADIUS サーバーの管理マニュアル

タスク 4: 外部 RADIUS 認可の構成（オプション）

Oracle データベースに接続する RADIUS ユーザーに対して外部 RADIUS 認可が必要な場合は、次の手順を実行して Oracle サーバー、Oracle クライアントおよび RADIUS サーバーを構成する必要があります。

Oracle サーバー（RADIUS クライアント）の構成手順：

1. OS_ROLE パラメータを init.ora ファイルに追加し、このパラメータを次のように TRUE に設定します。

```
OS_ROLE=TRUE
```

データベースを再起動します。init.ora ファイルに対する変更がシステムに反映されます。

2. 4-15 ページの「要求 / 応答の構成」にリストされている手順が済んでいない場合は、サーバーの RADIUS 要求 / 応答モードを ON に設定します。
3. 外部的に識別されるユーザーおよびロールを追加します。

Oracle クライアント（ユーザーがログインする場所）の構成手順：

4-15 ページの「[要求 / 応答の構成](#)」にリストされている手順が済んでいない場合は、クライアントの RADIUS 要求 / 応答モードを ON に設定します。

RADIUS サーバーの構成手順：

1. 次の属性を RADIUS サーバー属性構成ファイルに追加します。

属性名	コード	型
VENDOR_SPECIFIC	26	int 型
ORACLE_ROLE	1	String 型

2. SMI ネットワーク管理プライベート・エンタープライズ・コードが含まれている RADIUS サーバー属性構成ファイルで、111 番を Oracle のベンダー ID として割り当てます。

たとえば、RADIUS サーバー属性構成ファイルに次のように入力します。

```
VALUE VENDOR_SPECIFIC ORACLE 111
```

3. 次の構文を使用して、ORACLE_ROLE 属性を外部 RADIUS 認可を使用するユーザーのユーザー・プロファイルに追加します。

```
ORA_databaseSID_rolename[_[A] | [D]]
```

ここで、

- ORA は、このロールを Oracle 用に使用することを指定します。
- databaseSID は、データベース・サーバーの init.ora ファイルに構成されている Oracle システム識別子です。
- rolename は、データ・ディクショナリに定義されているロール名（たとえば、SYSDBA）です。
- A は、ユーザーにはこのロールに対する管理者権限があることを示すオプションの文字です。
- D は、このロールをデフォルトで使用可能にすることを示すオプションの文字です。

Oracle ロールにマップする RADIUS グループが、ORACLE_ROLE の構文に準拠していることを確認します。

例：

```
USERNAME      USERPASSWD="user_password",
               SERVICE_TYPE=login_user,
               VENDOR_SPECIFIC=ORACLE,
               ORACLE_ROLE=ORA_ora920_sysdba
```

関連項目： サーバーの構成方法は、RADIUS サーバーの管理マニュアルを参照してください。

タスク 5: RADIUS アカウントの作成

RADIUS アカウントでは、Oracle データベース・サーバーへのアクセス情報を記録し、RADIUS アカウント・サーバーのファイルに格納します。この機能は、RADIUS サーバーと認証サーバーの両方でサポートされている場合にのみ使用することができます。

Oracle データベース・サーバーでの RADIUS アカウントの設定

RADIUS アカウントを使用可能にしたり、使用禁止にするには、次のようにします。

1. Oracle Net Manager を起動します。
 - UNIX の場合は、\$ORACLE_HOME/bin から netmgr を実行します。
 - Windows NT の場合は、「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Network Administration」→「Oracle Net Manager」の順に選択します。
2. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。
3. 右側のウィンドウ・ペインのリストから、「Oracle Advanced Security」を選択します。「Oracle Advanced Security」タブ・ウィンドウが表示されます (図 4-5)。
4. 「Other Params」タブを選択します。
5. 「Authentication Service」リストから「RADIUS」を選択します。
6. アカウントを使用可能にするには、「Send Accounting」フィールドに「ON」を入力し、使用禁止にするには「OFF」を入力します。
7. 「File」→「Save Network Configuration」を選択します。

sqlnet.ora ファイルが更新され、次のエントリが追加されます。

```
SQLNET.RADIUS_SEND_ACCOUNTING= ON
```

RADIUS アカウント・サーバーの構成

RADIUS アカウントは、RADIUS 認証サーバーと同じホストまたは別のホストにあるアカウント・サーバーから成ります。

関連項目： RADIUS アカウントの構成方法は、RADIUS サーバーの管理マニュアルを参照してください。

タスク 6: RADIUS クライアント名の RADIUS サーバー・データベースへの追加

Internet Engineering Task Force (IETF) RFC #2138 Remote Authentication Dial In User Service (RADIUS) および RFC #2139 RADIUS Accounting の規格に準拠する RADIUS サーバーを使用できます。RADIUS サーバーにはいろいろな種類があるため、固有の相互運用要件がないか、RADIUS サーバーのマニュアルで確認しておいてください。

次の手順により、RADIUS クライアント名を Livingston RADIUS サーバーに追加します。

1. /etc/raddb/clients にあるクライアント・ファイルを開きます。次のテキストと表が表示されます。

```
@ (#) clients 1.1 2/21/96 Copyright 1991 Livingston Enterprises Inc
This file contains a list of clients which are allowed to make authentication
requests and their encryption key. The first field is a valid hostname. The
second field (separated by blanks or tabs) is the encryption key.
Client Name                               Key
```

2. CLIENT NAME 列に、Oracle データベース・サーバーが実行されているホストのホスト名または IP アドレスを入力します。KEY 列に共有シークレットを入力します。

CLIENT NAME 列に入力する値は、クライアントの名前または IP アドレスにかかわらず、RADIUS サーバーによって異なります。

3. clients ファイルを保存して閉じます。

関連項目： RADIUS サーバーの管理マニュアル

タスク 7: RADIUS とともに使用する認証サーバーの構成

認証サーバーの構成方法は、認証サーバーのマニュアルを参照してください。

関連項目： 使用可能なリソースのリストは、xxiv ページの「[関連文書](#)」を参照してください。

タスク 8: 認証サーバーとともに使用する RADIUS の構成

RADIUS サーバーのマニュアルを参照してください。

タスク 9: マッピング・ロールの構成

RADIUS サーバーでベンダー・タイプ属性をサポートしている場合は、ロールを RADIUS サーバーに格納して管理することができます。RADIUS を使用した CONNECT 要求があると、Oracle データベース・サーバーがロールをダウンロードします。

この機能を使用するには、Oracle データベース・サーバーと RADIUS サーバーの両方でロールを構成します。

次の手順で、Oracle データベース・サーバーにロールを構成します。

1. テキスト・エディタを使用して、Oracle データベース・サーバーの初期化パラメータ・ファイルに OS_ROLES パラメータを設定します。
2. Oracle データベース・サーバーを停止して再起動します。
3. IDENTIFIED EXTERNALLY を指定して、RADIUS サーバーが Oracle データベース・サーバー上で管理対象とする各ロールを作成します。

RADIUS サーバーでロールを構成するには、次の構文を使用します。各項目の意味は、[表 4-1](#) を参照してください。

ORA_DatabaseName.DatabaseDomainName_RoleName

例：

ORA_USERDB.US.ORACLE.COM_MANAGER

表 4-2 RADIUS 構成パラメータ

パラメータ	説明
DatabaseName	ロールを作成する Oracle データベース・サーバーの名前。 DB_NAME 初期化パラメータと同じ値です。
DatabaseDomainName	Oracle データベース・サーバーが属するドメインの名前。 DB_DOMAIN 初期化パラメータと同じ値です。
RoleName	Oracle データベース・サーバーに作成したロールの名前。

4. RADIUS 要求 / 応答モードを構成します。
要求 / 応答モードを構成する手順は、次の項を参照してください。
 - 4-6 ページ「要求 / 応答（非同期）認証モード」
 - 4-15 ページ「要求 / 応答の構成」

RADIUS を使用したデータベースへのログイン

同期認証モードを使用している場合は、SQL*Plus を起動して、プロンプトで次のコマンドを入力します。

```
CONNECT username/password@database_alias
```

このコマンドでログインできるのは、要求 / 応答が ON になっていない場合のみです。

要求 / 応答（非同期）モードを使用している場合は、SQL*Plus を起動して、プロンプトで次のコマンドを入力します。

```
CONNECT /@database_alias
```

このコマンドでログインできるのは、要求 / 応答が ON の場合のみです。

注意： 要求 / 応答モードは、ログインのすべてのケースについて構成できます。

RSA ACE/Server 構成チェックリスト

RSA ACE/Server を RADIUS サーバーとして使用している場合は、最初の接続を行う前に、次の項目をチェックしてください。

- 共有シークレットを送信するように RSA ACE/Server のホスト・エージェントが設定されていること。バージョン 5.0 の場合、この設定を行うには「SENT Node secret」チェックボックスをオフのままにします。RSA ACE/Server がエージェントへの共有シークレットの送信に失敗した場合は、ノード検証障害メッセージが RSA ACE/Server ログに記録されます。
- RSA の SecurID トークンを使用している場合は、トークンと RSA ACE/Server との同期が行われていること。

関連項目： トラブルシューティングの詳細は、RSA ACE/Server のマニュアルを参照してください。

CyberSafe 認証の構成

この章では、Oracle9i または Oracle9i データベースの Oracle Advanced Security を構成し、Kerberos ベースの認証サーバーである CyberSafe TrustBroker を使用して Oracle ユーザーを認証する方法について説明します。次の項目について説明します。

- [CyberSafe 認証の構成](#)
- [トラブルシューティング](#)

CyberSafe 認証の構成

CyberSafe 認証を構成する手順は次のとおりです。

- タスク 1: CyberSafe Server のインストール
- タスク 2: CyberSafe TrustBroker Client のインストール
- タスク 3: CyberSafe Application Security Toolkit のインストール
- タスク 4: Oracle データベース・サーバーに対するサービス・プリンシパルの構成
- タスク 5: CyberSafe からサービス表を抽出する
- タスク 6: Oracle データベース・サーバーのインストール
- タスク 7: Oracle Advanced Security を CyberSafe とともにインストールする
- タスク 8: Oracle Net と Oracle9i のインストール
- タスク 9: CyberSafe 認証の構成
- タスク 10: 認証サーバーでの CyberSafe ユーザーの作成
- タスク 11: Oracle データベース・サーバーに外部認証 Oracle ユーザーを作成する
- タスク 12: CyberSafe/Oracle ユーザーの初期チケットの取得
- タスク 13: CyberSafe によって認証された Oracle データベース・サーバーに接続

タスク 1: CyberSafe Server のインストール

認証サーバーとして動作するシステムに対してこの作業を行います。

関連項目： xxiv ページの「[関連文書](#)」に示した CyberSafe のマニュアルを参照してください。

タスク 2: CyberSafe TrustBroker Client のインストール

Oracle データベース・サーバーと Oracle クライアントを実行するシステムに対してこの作業を行います。

関連項目： xxiv ページの「[関連文書](#)」に示した CyberSafe のマニュアルを参照してください。

タスク 3: CyberSafe Application Security Toolkit のインストール

クライアントとサーバーの両方のシステムでこの作業を行います。

関連項目： xxiv ページの「[関連文書](#)」に示した CyberSafe のマニュアルを参照してください。

タスク 4: Oracle データベース・サーバーに対するサービス・プリンシパルの構成

Oracle データベース・サーバーでクライアントの識別情報を検証するには、CyberSafe TrustBroker Master Server が稼働するシステム上で、Oracle データベース・サーバーのサービス・プリンシパルを構成します。必要な場合は、レルムも構成します。

プリンシパルの名前の書式は次のとおりです。

`kservice/kinstance@REALM`

表 5-1 に、サービス・プリンシパル名のフィールド値を示します。

表 5-1 CyberSafe TrustBroker のサービス・プリンシパル名のフィールド値

フィールド	説明
<code>kservice</code>	Oracle サービスを表す大 / 小文字の区別がある文字列。これは、データベース・サービス名と同じでなくてもかまいません。
<code>kinstance</code>	通常は、Oracle が稼働しているシステムの完全修飾名です。
<code>REALM</code>	サーバーのドメイン名。REALM は常に大文字である必要があり、一般的に、DNS ドメイン名が指定されます。 <code>xst</code> を使用するとき REALM の値を入力しないと、 <code>kdb5_edit</code> は現行ホストのレルムを使用し、それをコマンド出力に表示します。

注意： この項で説明するユーティリティ名は、実行可能プログラムです。ただし、CyberSafe のユーザー名 CYBERUSER とレルム SOMECO.COM は単なる例です。

たとえば、Oracle サービスが `oracle` で、Oracle が稼働しているシステムの完全修飾名が `dbserver.someco.com` で、レルムが `SOMECO.COM` の場合、プリンシパル名は次のようになります。

`oracle/dbserver.someco.com@SOMECO.COM`

次のように、`kdb5_edit` をルートとして実行して、サービス・プリンシパルを作成します。

```
# cd /krb5/admin
# ./kdb5_edit
```

oracle/dbserver.someco.com@SOMECO.COM というプリンシパルを、CyberSafe が認識するサーバー・プリンシパルのリストに追加するには、kdb5_edit で次のように入力します。

```
kdb5_edit: ark oracle/dbserver.someco.com@SOMECO.COM
```

タスク 5: CyberSafe からサービス表を抽出する

CyberSafe からサービス表を抽出して、それを Oracle データベース・サーバーと CyberSafe TrustBroker クライアントの両方のシステムにコピーします。

たとえば、dbserver.someco.com のサービス表を抽出するには、次の手順を実行します。

1. kdb5_edit に次のように入力します。

```
kdb5_edit: xst dbserver.someco.com oracle
'oracle/dbserver.someco.com@SOMECO.COM' added to keytab
'WRFFILE:dbserver.someco.com-new-srvtab'
kdb5_edit: exit
# /krb5/bin/klist -k -t dbserver.someco.com-new-srvtab
```

xst を使用するときには REALM（この例では、SOMECO.COM）を入力しないと、kdb5_edit は現行ホストのレルムを使用し、それを前述の入力例のようにコマンド出力に表示します。

2. サービス表を抽出した後、その表に古いエントリと新しいエントリがあることを確認します。新しいエントリがサービス表にない場合、または新しいエントリを追加する必要がある場合は、kdb5_edit を使用してエントリを追加します。
3. CyberSafe サービス表を CyberSafe TrustBroker クライアント・システムに移動します。サービス表が CyberSafe クライアントと同一のシステム上にある場合は、それを次の例で示すように移動します。

```
# mv dbserver.someco.com-new-srvtab /krb5/v5srvtab
```

CyberSafe TrustBroker クライアントと異なるシステムにサービス表がある場合は、FTP などのプログラムを使用してファイルを転送します。FTP を使用する場合は、ファイルをバイナリ・モードで転送します。

4. Oracle データベース・サーバー・プログラム（実行ファイル）の所有者がサービス表（前述の例では、/krb5/v5srvtab）を読み取ることができるようにします。ファイル所有者を Oracle ユーザーに設定するか、Oracle が属しているグループに対してファイルの読取り可能にします。セキュリティが侵害されるおそれがあるため、ファイルをすべてのユーザーに対して読取り可能にしないでください。

タスク 6: Oracle データベース・サーバーのインストール

CyberSafe TrustBroker クライアントを実行しているシステムと同じシステムに Oracle データベース・サーバーをインストールします。

関連項目： オペレーティング・システム固有の Oracle9i インストール・ガイド

タスク 7: Oracle Advanced Security を CyberSafe とともにインストールする

Oracle9i の Custom インストール時に、CyberSafe を Oracle Advanced Security とともにインストールします。Oracle Universal Installer によってインストール・プロセス全体がガイドされます。

関連項目： オペレーティング・システム固有の Oracle9i インストール・ガイド

タスク 8: Oracle Net と Oracle9i のインストール

サーバーとクライアントの両方のシステムで、Oracle Net と Oracle9i を構成します。

関連項目： オペレーティング・システム固有の Oracle9i インストール・ガイド

タスク 9: CyberSafe 認証の構成

次の作業により、Oracle データベース・サーバーおよびクライアントの sqlnet.ora ファイルにパラメータを設定して、CyberSafe を構成します。

- クライアントと Oracle データベース・サーバーでの CyberSafe の構成
- 初期化パラメータ・ファイルへの REMOTE_OS_AUTHENT の設定 (init.ora)

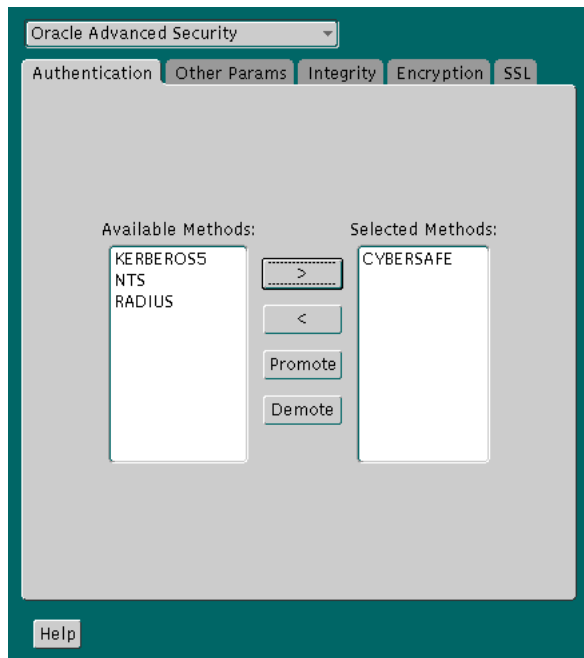
クライアントと Oracle データベース・サーバーでの CyberSafe の構成

クライアントとデータベース・サーバーの両方で CyberSafe 認証サービス・パラメータを構成する手順は、次のとおりです。

1. Oracle Net Manager を起動します。
 - UNIX の場合は、\$ORACLE_HOME/bin から netmgr を実行します。
 - Windows NT の場合は、「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Network Administration」→「Oracle Net Manager」の順に選択します。
2. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。

3. 右側のウィンドウ・ペインのリストから、「Oracle Advanced Security」を選択します。
「Oracle Advanced Security」タブ・ウィンドウが表示されます (図 5-1)。

図 5-1 Oracle Advanced Security の「Authentication」ウィンドウ (Cybersafe)



4. 「Authentication」タブを選択します。
5. 「Available Methods」リストで、「CYBERSAFE」を選択します。
6. 右矢印「>」をクリックして、「CYBERSAFE」を「Selected Methods」リストに移動します。
7. 選択した方式を使用優先順位の高い順に並べます。これを行うには、「Selected Methods」リストで方式を選択して、「Promote」または「Demote」をクリックします。たとえば、最初に使用するサービスを CYBERSAFE にするには、リストの先頭に移動します。
8. 「Other Params」タブを選択します (図 5-2)。

図 5-2 Oracle Advanced Security の「Other Params」ウィンドウ (Cybersafe)



9. 「Authentication Service」リストから「CYBERSAFE」を選択します。
10. 次の例のように「GSSAPI Service」フィールドに GSSAPI サービスの名前を入力します。

```
oracle/dbserver.someco.com@SOMECO.COM
```

「タスク 4: Oracle データベース・サーバーに対するサービス・プリンシパルの構成」で説明している書式でプリンシパル名を追加します。

11. 「File」→「Save Network Configuration」を選択します。

sqlnet.ora ファイルが更新され、次のエントリが追加されます。

```
SQLNET.AUTHENTICATION_SERVICES= (CYBERSAFE)
SQLNET.AUTHENTICATION_GSSAPI_SERVICE=KSERVICE/KINSTANCE@REALM
```

初期化パラメータ・ファイルへの REMOTE_OS_AUTHENT の設定

次のパラメータを初期化パラメータ・ファイル（init.ora）に追加します。

```
REMOTE_OS_AUTHENT=FALSE
```

注意： REMOTE_OS_AUTHENT を TRUE に設定すると、非保護プロトコル（TCP など）を使用するユーザーがオペレーティング・システム許可ログイン（以前の OPS\$ ログイン）を実行できるので、セキュリティが侵害されるおそれがあります。

CyberSafe ユーザー名には長い名前を使用できますが、Oracle ユーザー名は 30 文字に制限されているので、OS_AUTHENT_PREFIX の値として NULL を使用することをお勧めします。

```
OS_AUTHENT_PREFIX=""
```

構成ファイルを変更して変更内容を有効な状態にしてから、Oracle データベース・サーバーを再起動します。

関連項目： Oracle データベース・サーバーを再起動する方法は、オペレーティング・システム固有の Oracle マニュアルおよび『Oracle9i データベース管理者ガイド』を参照してください。

タスク 10: 認証サーバーでの CyberSafe ユーザーの作成

CyberSafe で Oracle ユーザーを認証するには、管理ツールがインストールされている CyberSafe 認証サーバーでユーザーを作成する必要があります。次の手順では、レルムがすでに存在することを前提としています。

注意： この項で説明するユーティリティ名は、実行可能プログラムです。ただし、CyberSafe のユーザー名 CYBERUSER とレルム SOMECO.COM は単なる例です。

/krb5/admin/kdb5_edit を認証サーバー上でルートとして実行し、CYBERUSER などの新しい CyberSafe ユーザーを作成します。

次のように入力します。

```
# kdb5_edit
kdb5_edit:
ank cyberuser
Enter password:
<password> (password does not display)
```

```
Re-enter password for verification:
<password> (password does not display)
kdb5_edit: quit
```

関連項目： レルムの作成方法は、xxiv ページの「[関連文書](#)」に記載されている Cybersafe マニュアルを参照してください。

タスク 11: Oracle データベース・サーバーに外部認証 Oracle ユーザーを作成する

SQL*Plus を実行して Oracle ユーザーを作成し、Oracle データベース・サーバーで次のコマンドを入力します（Oracle ユーザーは、大文字で入力し、二重引用符で囲む必要があります）。

この例では、OS_AUTHENT_PREFIX が NULL ("") に設定されます。

```
SQL> CONNECT / AS SYSDBA;
SQL> CREATE USER "CYBERUSER@SOMECO.COM" IDENTIFIED EXTERNALLY;
SQL> GRANT CREATE SESSION TO "CYBERUSER@SOMECO.COM";
```

関連項目： 『Oracle9i データベース管理者ガイド』

タスク 12: CyberSafe/Oracle ユーザーの初期チケットの取得

ユーザーがデータベースに接続するには、クライアント上で kinit を実行して[初期チケット](#)を取得する必要があります。

1. 次のように入力します。

```
% kinit cyberuser
```
2. パスワードを入力します（パスワードは表示されません）。
3. 現在所有しているチケットをリストするには、クライアントで klist を実行します。システム・コマンド・プロンプトで次のように入力します。

```
% klist
```

次の情報が表示されます。

作成日	有効期限	サービス
11-Aug-99 16:29:51	12-Aug-99 00:29:21	krbtgt/SOMECO.COM@SOMECO.COM
11-Aug-99 16:29:51	12-Aug-99 00:29:21	oracle/dbserver.someco.com@SOMECO.COM

タスク 13: CyberSafe によって認証された Oracle データベース・サーバーに接続

kinit を実行して初期チケットを取得した後は、ユーザー名またはパスワードを使用せずに Oracle データベース・サーバーに接続できます。次のようなコマンドを入力します。

```
% sqlplus /@net_service_name
```

net_service_name は Oracle Net のネット・サービス名です。

たとえば、次のように指定します。

```
% sqlplus /@npddoc_db
```

関連項目： 第 1 章「[Oracle Advanced Security の概要](#)」および『Oracle9i Heterogeneous Connectivity Administrator's Guide』

トラブルシューティング

この項では、構成に関する一般的な問題とその解決方法について説明します。

kinit を使用してチケット認可チケットを取得できない場合

- krb.conf を調べて、デフォルトのレルムが適切であるかどうかを確認します。
- レルムに指定されているホスト上で TrustBroker Master Server が稼働しているかどうかを確認します。
- Master Server にユーザー・プリンシパルのエントリがあること、およびパスワードが一致しているかどうかを確認します。
- krb.conf ファイルと krb.realms ファイルが Oracle で読取り可能になっているかどうかを確認します。

初期チケットはあるが接続できない場合

- 接続を試みた後で、サービス・チケットをチェックします。
- データベース・サーバー側の sqlnet.ora ファイルに、CyberSafe Master Server が認識するサービスに対応するサービス名があることをチェックします。
- 関係するすべてのシステムで、クロックのずれが数分以内であることをチェックします。

サービス・チケットはあるが接続できない場合

- クライアントおよびデータベース・サーバー上でクロックをチェックします。
- v5srvtab ファイルが適切な場所に存在し、Oracle で読取り可能になっているかどうかをチェックします。
- データベース・サーバー側のプロファイル (sqlnet.ora) で指定されているサービスに対して、v5srvtab ファイルが生成されていることをチェックします。

何も問題はないが別の問合せが失敗する場合

- 初期チケットが転送可能であるかどうかをチェックします。kinit -f を実行して、初期チケットを取得している必要があります。
- 資格証明の有効期限をチェックします。
- 資格証明の有効期限が切れている場合は、接続をクローズし kinit を実行して新しい初期チケットを取得します。

Kerberos 認証の構成

この章では、Oracle9i の Oracle Advanced Security で Kerberos 認証を使用できるように構成する方法、および Kerberos を構成して Oracle データベース・ユーザーを認証する方法について説明します。次の項目について説明します。

- [Kerberos 認証を使用可能にする](#)
- [Kerberos 認証アダプタで使用するユーティリティ](#)
- [Windows 2000 ドメイン・コントローラ KDC との相互運用の構成](#)
- [トラブルシューティング](#)

Kerberos 認証を使用可能にする

Kerberos 認証を使用可能にする手順は次のとおりです。

- [タスク 1: Kerberos のインストール](#)
- [タスク 2: Oracle データベース・サーバーに対するサービス・プリンシパルの構成](#)
- [タスク 3: Kerberos からのサービス表の抽出](#)
- [タスク 4: Oracle データベース・サーバーと Oracle クライアントのインストール](#)
- [タスク 5: Oracle Net Services と Oracle Advanced Security のインストール](#)
- [タスク 6: Oracle Net Services と Oracle9i のインストール](#)
- [タスク 7: Kerberos 認証の構成](#)
- [タスク 8: Kerberos ユーザーの作成](#)
- [タスク 9: 外部認証された Oracle ユーザーの作成](#)
- [タスク 10: Kerberos/Oracle ユーザーの初期チケットの取得](#)

タスク 1: Kerberos のインストール

認証サーバーとして動作するシステムに Kerberos をインストールします。

関連項目： Kerberos のインストール方法は、Kerberos バージョン 5 ソース配布から Kerberos を構築してインストールする方法を説明した資料を参照してください。

タスク 2: Oracle データベース・サーバーに対するサービス・プリンシパルの構成

Kerberos を使用して自己を認証するクライアントの識別情報を Oracle データベース・サーバーで検証できるようにするには、Oracle9i の[サービス・プリンシパル](#)を作成する必要があります。

プリンシパルの名前を次の書式で指定する必要があります。

kservice/kinstance@REALM

サービス・プリンシパルの各フィールドには、次の値を指定します。

kservice	Oracle サービスを表す大 / 小文字の区別がある文字列。これは、データベース・サービス名と同じでもかまいません。
kinstance	通常は、Oracle9i が稼働しているシステムの完全修飾名です。

REALM

データベース・サーバーのドメイン名。REALM は常に大文字である必要があります、一般的に、DNS ドメイン名が指定されます。

注意： この項で説明するユーティリティ名は、実行可能プログラムです。ただし、Kerberos ユーザー名 `krbuser` とレルム `SOMECO.COM` は単なる例です。

たとえば、`kservice` が `oracle`、`Oracle9i` が稼働しているシステムの完全修飾名が `dbserver.someco.com` で、レルムが `SOMECO.COM` の場合、プリンシパル名は次のようになります。

```
oracle/dbserver.someco.com@SOMECO.COM
```

通常は、DNS ドメイン名をレルムの名前として使用します。**サービス・プリンシパル**を作成するには、`kadmin.local` を実行します。UNIX の場合は、次の構文を使用し、`root` ユーザーでこのコマンドを入力します。

```
# cd /kerberos-install-directory/sbin
# ./kadmin.local
```

`oracle/dbserver.someco.com@SOMECO.COM` という**プリンシパル**を、Kerberos が認識するサーバー・プリンシパルのリストに追加するには、次のように入力します。

```
kadmin.local:addprinc -randkey oracle/dbserver.someco.com@SOMECO.COM
```

タスク 3: Kerberos からのサービス表の抽出

Kerberos から**サービス表**を抽出し、そのサービス表を Oracle データベース・サーバー/Kerberos クライアント・システムにコピーします。

たとえば、次の手順を使用して `dbserver.someco.com` のサービス表を抽出します。

1. 次のコマンドを入力して、サービス表を抽出します。

```
kadmin.local:ktadd -k /tmp/keytab oracle/dbserver.someco.com

Entry for principal oracle/dbserver.someco.com with kvno 2,
encryption DES-CBC-CRC added to the keytab
WRFILE: 'WRFILE:/tmp/keytab

kadmin.local:exit

oklist -k -t /tmp/keytab
```

2. サービス表を抽出した後、サービス表に古いエントリと新しいエントリがあることを確認します。新しいエントリがサービス表内にない場合、または追加する必要がある場合は、`kadmin.local` を使用してエントリを追加します。

ktadd を使用するときレلمを入力しないと、`kadmin.local` が現行ホストのレلمを使用して、それを手順 1 のようにコマンド出力に表示します。

3. Kerberos サービス表が Kerberos クライアント・システムと同じシステム上にある場合は、それを移動するのみでかまいません。サービス表が Kerberos クライアントと異なるシステム上にある場合は、FTP などのプログラムを使用してファイルを転送する必要があります。FTP を使用する場合は、ファイルをバイナリ・モードで転送します。

次に、UNIX プラットフォームでのサービス表の移動方法の例を示します。

```
# mv /tmp/keytab /etc/v5srvtab
```

サービス・ファイルのデフォルト名は、`/etc/v5srvtab` です。

4. Oracle データベース・サーバー・プログラム（実行ファイル）の所有者がサービス表（前述の例では `/etc/v5srvtab`）を読み取ることができるか確認します。このためには、ファイル所有者を Oracle ユーザーに設定するか、ファイルを Oracle が属しているグループに対して読取り可能にします。

注意： セキュリティが侵害されるおそれがあるため、ファイルをすべてのユーザーに対して読取り可能にしないでください。

タスク 4: Oracle データベース・サーバーと Oracle クライアントのインストール

Oracle データベース・サーバーとクライアント・ソフトウェアをインストールします。

関連項目： オペレーティング・システム固有の Oracle9i インストール・ガイド

タスク 5: Oracle Net Services と Oracle Advanced Security のインストール

Oracle データベース・サーバーと Oracle クライアント・システムに、Oracle Net Services と Oracle Advanced Security をインストールします。

関連項目： オペレーティング・システム固有の Oracle9i インストール・ガイド

タスク 6: Oracle Net Services と Oracle9i のインストール

Oracle データベース・サーバーとクライアントで Oracle Net Services を構成します。

関連項目：

- オペレーティング・システム固有の Oracle9i インストレーション・ガイド
- 『Oracle9i Net Services 管理者ガイド』

タスク 7: Kerberos 認証の構成

次の作業を実行して、Oracle データベース・サーバーおよびクライアントの `sqlnet.ora` ファイルに必要なパラメータを設定します。

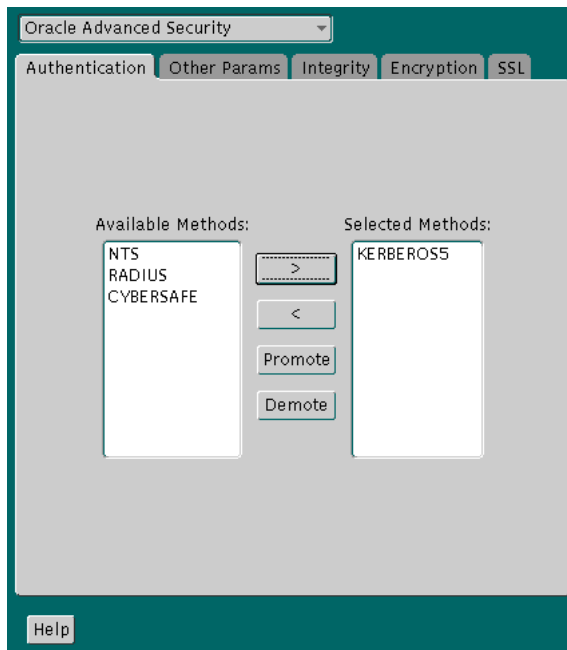
- [手順 1: クライアントとデータベース・サーバーでの Kerberos の構成](#)
- [手順 2: 初期化パラメータの設定](#)
- [手順 3: sqlnet.ora パラメータの設定 \(オプション\)](#)

手順 1: クライアントとデータベース・サーバーでの Kerberos の構成

次の手順を実行して、クライアントおよびデータベース・サーバーに Kerberos 認証サービス・パラメータを構成します。

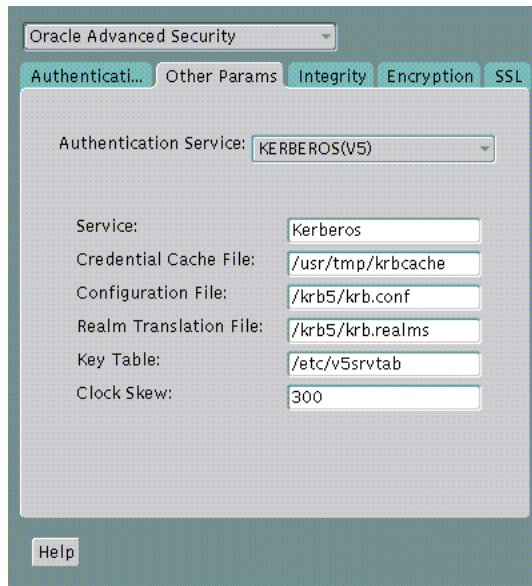
1. Oracle Net Manager を起動します。
 - UNIX の場合は、`$ORACLE_HOME/bin` から `netmgr` を実行します。
 - Windows NT の場合は、「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Network Administration」→「Oracle Net Manager」の順に選択します。
2. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。
3. 右側のウィンドウ・ペインのリストから、「Oracle Advanced Security」を選択します。「Oracle Advanced Security」ウィンドウが表示されます (図 6-1)。

図 6-1 Oracle Advanced Security の「Authentication」ウィンドウ (Kerberos)



4. 「Authentication」タブを選択します。
5. 「Available Methods」リストから「KERBEROS5」を選択します。
6. 右矢印 (>) をクリックして、「KERBEROS5」を「Selected Methods」リストに移動します。
7. 選択したメソッドを使用する順に並べます。「Selected Methods」リストで方式を選択し、「Promote」または「Demote」をクリックして並べ替えます。たとえば、最初に使用するサービスを KERBEROS5 にするには、リストの先頭に移動します。
8. 「Other Params」タブを選択します (図 6-2)。

図 6-2 Oracle Advanced Security の「Other Params」ウィンドウ (Kerberos)



9. 「Authentication Service」リストから「KERBEROS (V5)」を選択します。
10. 「Service」フィールドに「Kerberos」と入力します。このフィールドは、Kerberos の [サービス・チケット](#)を取得するために Oracle9i が使用するサービスの名前を定義します。このフィールドに値を指定すると、他のフィールドに入力できるようになります。
11. オプションとして、次のフィールドの値を入力します。
 - 「Credential Cache File」
 - 「Configuration File」
 - 「Realm Translation File」
 - 「Key Table」
 - 「Clock Skew」

関連項目： 構成するフィールドとパラメータの詳細は、Oracle Net Manager のオンライン・ヘルプおよび 6-8 ページの「[手順 3: sqlnet.ora パラメータの設定 \(オプション\)](#)」を参照してください。

12. 「File」→「Save Network Configuration」を選択します。

sqlnet.ora ファイルが更新され、次のエントリが追加されます。

```
SQLNET.AUTHENTICATION_SERVICES=(KERBEROS5)
```

```
SQLNET.AUTHENTICATION_KERBEROS5_SERVICE=kservice
```

手順 2: 初期化パラメータの設定

初期化パラメータ・ファイルにパラメータを設定する手順は次のとおりです。

1. 次のパラメータを初期化パラメータ・ファイルに追加します。

```
REMOTE_OS_AUTHENT=FALSE
```

注意: REMOTE_OS_AUTHENT を TRUE に設定すると、非保護プロトコル (TCP など) を使用するユーザーがオペレーティング・システム許可ログイン (以前の OPS\$ ログイン) を実行できるので、セキュリティが侵害されるおそれがあります。

2. Kerberos ユーザー名には長い名前を使用できますが、Oracle ユーザー名は 30 字に制限されているため、次に示すように OS_AUTHENT_PREFIX の値を NULL 値に設定することをお勧めします。

```
OS_AUTHENT_PREFIX=""
```

このパラメータを NULL に設定すると、デフォルト値の OPS\$ が上書きされます。

手順 3: sqlnet.ora パラメータの設定 (オプション)

必須パラメータの他に、sqlnet.ora ファイルの次のオプション・パラメータをクライアントおよび Oracle データベース・サーバーで設定できます。

パラメータ: SQLNET.KERBEROS5_CC_NAME=pathname_to_credentials_cache_file

説明: Kerberos 資格証明キャッシュ (CC) ファイルの完全パス名を指定します。デフォルト値はオペレーティング・システムによって異なります。UNIX では、パス名が /tmp/krb5cc_userid となります。

このパラメータは環境変数 KRB5CCNAME を使用して設定することもできますが、sqlnet.ora ファイルで設定する値が、環境変数 KRB5CCNAME で設定する値より優先されます。

例: SQLNET.KERBEROS5_CC_NAME=/usr/tmp/krb5cc

パラメータ: SQLNET.KERBEROS5_CLOCKSKEW=number_of_seconds_accepted_as_network_delay

- 説明:** このパラメータは、Kerberos 資格証明の有効期限が切れるまでの時間を秒数で指定します。資格証明がクライアントまたはデータベース・サーバーによって実際に受け取られるときに、このパラメータが使用されます。また、再生攻撃を受けないように資格証明を格納する必要があるかどうかを Oracle データベース・サーバーが判断するときも、このパラメータが使用されます。デフォルトは 300 秒です。
- 例:** `SQLNET.KERBEROS5_CLOCKSKEW=1200`
- パラメータ:** `SQLNET.KERBEROS5_CONF=pathname_to_Kerberos_configuration_file`
- 説明:** このパラメータは、Kerberos 構成ファイルの完全パス名を指定します。構成ファイルにはデフォルトの KDC (Key Distribution Center) のレルムが含まれていて、レルムを KDC ホストにマップします。デフォルトはオペレーティング・システムによって異なります。UNIX では、`/krb5/krb.conf` がデフォルトです。
- 例:** `SQLNET.KERBEROS5_CONF=/krb/krb.conf`
- パラメータ:** `SQLNET.KERBEROS5_CONF_MIT=[TRUE|FALSE]`
- 説明:** このパラメータは、新しい MIT Kerberos 構成書式を使用するかどうかを指定します。値を TRUE に設定すると、新しい構成書式ルールに基づいてそのファイルが解析されます。値を FALSE に設定すると、デフォルト (非 MIT) 構成が使用されます。デフォルトは FALSE です。
- 例:** `SQLNET.KERBEROS5_CONF_MIT=False`
- パラメータ:** `SQLNET.KERBEROS5_KEYTAB=pathname_to_Kerberos_principal/key_table`
- 説明:** このパラメータは、Kerberos プリンシパル / 秘密鍵マッピング・ファイルの完全パス名を指定します。Oracle データベース・サーバーが鍵を抽出し、クライアントから送信される認証情報を復号化するとき、このパラメータが使用されます。デフォルトはオペレーティング・システムによって異なります。UNIX では、デフォルトは `/etc/v5srvtab` です。
- 例:** `SQLNET.KERBEROS5_KEYTAB=/etc/v5srvtab`
- パラメータ:** `SQLNET.KERBEROS5_REALMS=pathname_to_Kerberos_realm_translation_file`

- 説明：** このパラメータは、Kerberos レルム変換ファイルの完全パス名を指定します。変換ファイルを使用して、ホスト名またはドメイン名をレルムにマップします。デフォルトはオペレーティング・システムによって異なります。UNIX では、デフォルトは `/etc/krb.realms` です。
- 例：** `SQLNET.KERBEROS5_REALMS=/krb5/krb.realms`

タスク 8: Kerberos ユーザーの作成

Kerberos で認証できる Oracle ユーザーを作成するには、管理ツールがインストールされている Kerberos 認証サーバー上で次の作業を実行します。レルムはすでに存在している必要があります。

注意： この項で説明するユーティリティ名は、実行可能プログラムです。ただし、Kerberos ユーザー名 `krbuser` とレルム `SOMECO.COM` は単なる例であり、実際の名前はシステムによって異なります。

`/krb5/admin/kadmin.local` を root ユーザーで実行して、`krbuser` などの新しい Kerberos ユーザーを作成します。

次に示す例は、UNIX の場合です。

```
# ./kadmin.local
kadmin.local: addprinc krbuser
Enter password for principal: "krbuser@SOMECO.COM": (password does not display)
Re-enter password for principal: "krbuser@SOMECO.COM": (password does not display)
kadmin.local: exit
```

タスク 9: 外部認証された Oracle ユーザーの作成

SQL*Plus を Oracle データベース・サーバー上で実行して、Kerberos ユーザーに対応する Oracle ユーザーを作成します。次の例では、`OS_AUTHENT_PREFIX` は `NULL` ("") に設定されます。Oracle ユーザー名は、次の例のように大文字で入力し、二重引用符で囲む必要があります。

```
SQL> CONNECT / AS SYSDBA;
SQL> CREATE USER "KRBUSER@SOMECO.COM" IDENTIFIED EXTERNALLY;
SQL> GRANT CREATE SESSION TO "KRBUSER@SOMECO.COM";
```

タスク 10: Kerberos/Oracle ユーザーの初期チケットの取得

データベースに接続する前に、Key Distribution Center (KDC) に初期チケットを要求する必要があります。要求するには、クライアントに対して次を実行します。

```
% okinit username
```

データベースに接続するときに、データベース・リンクの後に次のような参照が続く場合は、転送可能フラグ (-f) オプションを使用する必要があります。

```
sqlplus /@oracle
```

okinit -f を実行すると、データベース・リンクで使用できる資格証明が使用可能になります。Oracle クライアントで次のコマンドを実行します。

```
% okinit -f  
Password for krbuser@SOMECO.COM:password
```

Kerberos 認証アダプタで使用するユーティリティ

Oracle Kerberos 認証アダプタとともに3つのユーティリティが提供されています。これらのユーティリティは、Oracle Kerberos 認証サポートがインストールされた Oracle クライアントでの使用を意図したものです。特定のタスクの中で、これらのユーティリティを使用します。

- **okinit** ユーティリティを使用して初期チケットを取得
- **oklist** ユーティリティを使用して資格証明を表示
- **okdstry** ユーティリティを使用してキャッシュ・ファイルから資格証明を削除

注意： Solaris は、Kerberos バージョン 4 とともに出荷されます。Kerberos バージョン 4 ユーティリティが誤って使用されないように、Kerberos バージョン 5 ユーティリティをパスに指定してください。

okinit ユーティリティを使用して初期チケットを取得

okinit ユーティリティを使用して Kerberos チケットを取得しキャッシュに書き込みます。通常は、このユーティリティを使用してチケット認可チケットを取得し、ユーザーが入力したパスワードを使用して Key Distribution Center (KDC) から送られる資格証明を復号化します。チケット認可チケットはユーザーの資格証明キャッシュに格納されます。

okinit で使用できるオプションを表 6-1 にリストしています。

表 6-1 okinit ユーティリティのオプション

オプション	説明
-f	転送可能なチケット認可チケットを要求します。データベース・リンクをたどる場合は、このオプションが必要です。
-l	チケット認可チケットを含むすべてのチケットの存続期間を指定します。デフォルトで、チケット認可チケットの有効期間は 8 時間ですが、それより長い時間または短い時間も指定できます。KDC はこのオプションを無視したり、各サイトで指定できる時間を制限することができます。次の例に示すように、数字と、「w」(週)、「d」(日)、「h」(時間)、「m」(分)、「s」(秒) の修飾文字で構成される文字列で、存続期間を指定します。 okinit -l 2w1d6h20m30s この例では、チケット認可チケットの存続期間は 2 週間と 1 日と 6 時間 20 分 30 秒です。
-c	代替資格証明キャッシュを指定します。UNIX では、デフォルトが /tmp/krb5cc_uid となります。sqlnet.ora ファイルで SQLNET.KERBEROS5_CC_NAME パラメータを使用して、代替資格証明キャッシュを指定することもできます。
-?	コマンドライン・オプションのリストを表示します。

oklist ユーティリティを使用して資格証明を表示

oklist ユーティリティを実行すると、所有しているチケットのリストを表示できます。使用可能な oklist オプションを表 6-2 にリストしています。

表 6-2 oklist ユーティリティのオプション

オプション	説明
-f	資格証明のフラグを表示します。該当するフラグは「I」（資格証明がチケット認可チケット）、「F」（資格証明が転送可能なチケット）および「f」（資格証明が転送済チケット）です。
-c	代替資格証明キャッシュを指定します。UNIX では、デフォルトが /tmp/krb5cc_uid となります。sqlnet.ora ファイルで SQLNET.KERBEROS5_CC_NAME パラメータを使用して、代替資格証明キャッシュを指定することもできます。
-k	UNIX 上のサービス表のエントリ（デフォルト /etc/v5srvtab）をリストします。sqlnet.ora ファイルで SQLNET.KERBEROS5_KEYTAB パラメータを使用して、代替サービス表を指定することもできます。

このフラグ表示オプション（-f）によって、次のような追加情報が表示されます。

```
% oklist -f
27-Jul-1999 21:57:51    28-Jul-1999 05:58:14
krbtgt/SOMECO.COM@SOMECO.COM
Flags: FI
```

okdstry ユーティリティを使用してキャッシュ・ファイルから資格証明を削除

okdstry ユーティリティを使用して、次のように資格証明キャッシュ・ファイルから資格証明を削除します。

```
$ okdstry -f
```

-f コマンド・オプションによって、代替資格証明キャッシュを指定できます。UNIX では、デフォルトが /tmp/krb5cc_uid となります。sqlnet.ora ファイルで SQLNET.KRB5_CC_NAME パラメータを使用して、代替資格証明キャッシュを指定することもできます。

Kerberos によって認証された Oracle データベース・サーバーに接続

これで、ユーザー名またはパスワードを使用せずに Oracle データベース・サーバーに接続できます。次のようなコマンドを入力します。

```
$ sqlplus /@net_service_name
```

`net_service_name` は Oracle Net Services のネット・サービス名です。たとえば、次のように指定します。

```
$ sqlplus /@oracle_dbname
```

関連項目： 外部認証の詳細は、[第 1 章「Oracle Advanced Security の概要」](#) および『[Oracle9i Heterogeneous Connectivity Administrator's Guide](#)』を参照してください。

Windows 2000 ドメイン・コントローラ KDC との相互運用の構成

MIT Kerberos に準拠している Oracle Advanced Security は、Windows 2000 ドメイン・コントローラ上で、Kerberos Key Distribution Center (KDC) が発行するチケットを相互運用することによって、Oracle データベースでの Kerberos 認証を実現します。Windows 2000 ドメイン・コントローラ KDC を使用して Kerberos 認証を構成するには、次のタスクを実行します。

- [タスク 1: Windows 2000 ドメイン・コントローラ KDC と相互運用するための Oracle Kerberos クライアントの構成](#)
- [タスク 2: Oracle クライアントと相互運用するための Windows 2000 ドメイン・コントローラ KDC の構成](#)
- [タスク 3: Windows 2000 ドメイン・コントローラ KDC と相互運用するための Oracle データベースの構成](#)
- [タスク 4: Kerberos/Oracle ユーザーの初期チケットの取得](#)

タスク 1: Windows 2000 ドメイン・コントローラ KDC と相互運用するための Oracle Kerberos クライアントの構成

Oracle Kerberos クライアントで、次の手順を実行する必要があります。

手順 1: Windows 2000 ドメイン・コントローラ KDC を使用するためのクライアント Kerberos 構成ファイルの作成

Windows 2000 ドメイン・コントローラを Kerberos KDC として参照するように、次の Kerberos クライアント構成ファイルを作成します。次の例は、Windows 2000 ドメイン・コントローラがノード名 sales3854.us.acme.com 上で実行されていることを想定しています。

■ krb.conf ファイル

例：

```
SALES3854.US.ACME.COM
SALES3854.US.ACME.COM sales3854.us.acme.com admin server
```

■ krb5.conf ファイル

例：

```
[libdefaults]
default_realm=SALES.US.ACME.COM
[realms]
SALES.US.ACME.COM= {
    kdc=sales3854.us.acme.com:88
}
[domain_realm]
.us.acme.com=SALES3854.US.ACME.COM
```

■ krb5.realms ファイル

例：

```
us.acme.com SALES.US.ACME.COM
```

手順 2: sqlnet.ora ファイルでの Oracle 構成パラメータの指定

Windows 2000 ドメイン・コントローラ KDC との相互運用を行うために Oracle クライアントを構成するには、6-5 ページの「[手順 1: クライアントとデータベース・サーバーでの Kerberos の構成](#)」にリストされている sqlnet.ora ファイルの同じパラメータを使用します。

クライアントの `sqlnet.ora` ファイルに、次のパラメータを設定します。

```
SQLNET.KERBEROS5_CONF=pathname_to_Kerberos_configuration_file
SQLNET.KERBEROS5_CONF_MIT=TRUE
SQLNET.AUTHENTICATION_KERBEROS5_SERVICE=Kerberos_service_name
SQLNET.AUTHENTICATION_SERVICES=(BEQ,KERBEROS5)
```

注意： Windows 2000 オペレーティング・システムは、MIT Kerberos バージョン 5 に基づくセキュリティ・サービスとの間でのみ、相互運用を実施するように設計されているため、`SQLNET.KERBEROS5_CONF_MIT` パラメータが `TRUE` に設定されていることを確認してください。

手順 3: リスニング・ポート番号の指定

Windows 2000 ドメイン・コントローラ KDC は、UDP/TCP ポート 88 でリスニングします。`kerberos5` のシステム・ファイル・エントリが、次のように UDP/TCP ポート 88 に設定されていることを確認してください。

- (UNIX の場合)

`/etc/services` ファイルの `kerberos5` エントリが 88 に設定されていることを確認してください。

タスク 2: Oracle クライアントと相互運用するための Windows 2000 ドメイン・コントローラ KDC の構成

Windows 2000 ドメイン・コントローラで、次の手順を実行する必要があります。

関連項目： Active Directory でユーザーを作成する方法は、Microsoft 社のマニュアルを参照してください。

手順 1: ユーザーの作成

Microsoft Active Directory で Oracle クライアントの新規ユーザーを作成します。

手順 2: Oracle データベース・プリンシパルの作成

1. Microsoft Active Directory で Oracle データベースの新規ユーザーを作成します。

たとえば、Oracle データベースがホスト `sales3854.us.acme.com` 上で実行されている場合は、Active Directory でユーザー名 `sales3854.us.acme.com` とパスワード `oracle` を使用してユーザーを作成します。

注意： Active Directory では、ユーザーを `host/hostname.dns.com` (`oracle/sales3854.us.acme.com` など) で作成しないでください。Microsoft の KDC は、MIT KDC のようにマルチパートの名前をサポートしていません。Microsoft の KDC とは異なり、MIT KDC は、すべてのプリンシパルをユーザー名として扱うため、マルチパートの名前をサービス・プリンシパルに使用できます。

2. Ktpass コマンドライン・ユーティリティで次の構文を使用して、キータブ・ファイルを抽出します。

```
Ktpass -princ service/hostname@NT-DNS-REALM-NAME -mapuser account -pass password  
-out keytab.file
```

次に、前述の手順で作成したデータベース・ユーザーを使用した Ktpass の使用例を示します。

```
C:> Ktpass -princ oracle/sales3854.us.acme.com@SALES.US.COM -mapuser sales3854  
-pass oracle -out C:\temp\%v5srvtab
```

このユーティリティは Windows 2000 サポート・ツールの一部で、Windows 2000 配布メディアの `\support\reskit\netmgmt\security` フォルダにあります。

3. 抽出したキータブ・ファイルを Oracle データベースがインストールされているホスト・コンピュータにコピーします。

たとえば、前述の手順で作成されたキータブを `/krb5/v5srvtab` にコピーします。

関連項目： Windows 2000 と Kerberos 5 との相互運用の詳細は、次の URL を参照してください。

<http://www.microsoft.com/WINDOWS2000/techinfo/planning/security/kerbsteps.asp>

タスク 3: Windows 2000 ドメイン・コントローラ KDC と相互運用するための Oracle データベースの構成

Oracle データベースがインストールされているホスト・コンピュータで、次の手順を実行する必要があります。

手順 1: sqlnet.ora ファイルでの構成パラメータの設定

データベース・サーバーの sqlnet.ora ファイルで、次のパラメータの値を指定します。

```
SQLNET.KERBEROS5_CONF=pathname_to_Kerberos_configuration_file
SQLNET.KERBEROS5_KEYTAB=pathname_to_Kerberos_principal/key_table
SQLNET.KERBEROS5_CONF_MIT=TRUE
SQLNET.AUTHENTICATION_KERBEROS5_SERVICE=Kerberos_service_name
SQLNET.AUTHENTICATION_SERVICES=(BEQ, KERBEROS5)
```

注意： Windows 2000 オペレーティング・システムは、MIT Kerberos バージョン 5 に基づくセキュリティ・サービスとの間でのみ、相互運用を実施するように設計されているため、SQLNET.KERBEROS5_CONF_MIT パラメータが TRUE に設定されていることを確認してください。

手順 2: 外部認証された Oracle ユーザーの作成

6-10 ページの「[タスク 9: 外部認証された Oracle ユーザーの作成](#)」に従って、外部認証された Oracle ユーザーを作成します。ユーザー名は、たとえば ORAKRB@SALES.US.ACME.COM など、すべて大文字で作成されます。

関連項目： Oracle Net Manager を使用して sqlnet.ora ファイルのパラメータを設定する方法は、6-5 ページの「[タスク 7: Kerberos 認証の構成](#)」を参照してください。

タスク 4: Kerberos/Oracle ユーザーの初期チケットの取得

クライアントがデータベースに接続するには、[初期チケット](#)を要求する必要があります。初期チケットを要求する手順は、6-11 ページの「[タスク 10: Kerberos/Oracle ユーザーの初期チケットの取得](#)」に従ってください。

トラブルシューティング

この項では、構成に関する一般的な問題とその解決メソッドについて説明します。

- OKINIT を使用してチケット認可チケットを取得できない場合
 - krb.conf ファイルを調べて、デフォルトのレルムが適切であるかどうかを確認します。
 - レルムに対して指定されているホスト上で、KDC が稼働しているかどうかを確認します。
 - KDC にユーザー・プリンシパルのエントリがあること、およびパスワードが一致しているかどうかを確認します。
 - krb.conf ファイルと krb.realms ファイルが Oracle で読み取り可能になっているかどうかを確認します。
- 初期チケットはあるが接続できない場合
 - 接続を試みた後で、サービス・チケットをチェックします。
 - データベース・サーバー側の sqlnet.ora ファイルに、Kerberos が認識するサービスに対応するサービス名があることをチェックします。
 - 関係するすべてのシステムで、クロックのずれが数分以内に設定されていることをチェックします（または、sqlnet.ora ファイルの SQLNET.KERBEROS5_CLOCKSKEW パラメータを変更します）。
- サービス・チケットはあるが接続できない場合
 - クライアントおよびデータベース・サーバー上でクロックをチェックします。
 - v5srvtab ファイルが適切な場所に存在し、Oracle で読み取り可能になっているかどうかをチェックします（sqlnet.ora パラメータの設定を忘れないでください）。
 - データベース・サーバー側の sqlnet.ora ファイルで指定されているサービスに対して v5srvtab ファイルが生成されていることをチェックします。
- 何も問題はないが別の問合せが失敗する場合
 - 初期チケットが転送可能であるかどうかをチェックします（okinit ユーティリティを実行して、初期チケットを取得する必要があります）。
 - 資格証明の有効期限をチェックします。資格証明の有効期限が切れている場合は、接続をクローズし、okinit を実行して新しい初期チケットを取得します。

Secure Sockets Layer 認証の構成

この章では、Oracle Advanced Security で Secure Sockets Layer (SSL) プロトコルを使用する方法について説明しています。次の項目について説明します。

- Oracle 環境における SSL
- 非 Oracle クライアントと Oracle データベース・サーバー間の SSL
- SSL と他の認証方式の併用
- SSL とファイアウォール
- SSL 使用時の問題
- SSL を使用可能にする
- nCipher セキュア・アクセラレータの使用

Oracle 環境における SSL

Secure Sockets Layer (SSL) は、ネットワーク接続を保護するために Netscape 社が開発した業界標準プロトコルです。SSL は RSA 公開鍵暗号を使用して、**公開鍵インフラストラクチャ** (PKI) で、認証、暗号化およびデータの整合性を実現します。

この項では、次の内容について説明します。

- **SSL で可能なこと**
- **Oracle 環境における SSL の構成要素**
- **Oracle 環境における SSL の機能 : SSL ハンドシェイク**

SSL で可能なこと

Oracle Advanced Security は、SSL プロトコルの固有の暗号化やデータ整合性の機能に加え、SSL を介したデジタル証明の使用による認証をサポートしています。

Oracle Advanced Security の SSL 機能を使用して、クライアントとサーバーとの通信を保護することによって、次のことが可能になります。

- SSL を使用してクライアント / サーバー間の接続を暗号化
- 1 つ以上の Oracle データベース・サーバーに対するクライアントまたはサーバーの認証
- クライアントに対する Oracle データベース・サーバーの認証

SSL 機能は単独でも使用できますが、Oracle Advanced Security でサポートされている他の認証方式とともに使用することもできます。たとえば、SSL の暗号化と Kerberos の認証を組み合わせで使用できます。SSL は次のいずれかの認証モードをサポートします。

- サーバーのみがクライアントに対して認証を行います。
- クライアントとサーバーの両方が互いに対して認証を行います。
- クライアントもサーバーも互いに対して認証を行わずに、自分自身に対して SSL 暗号化機能を使用します。

関連項目：

- SSL の詳細は、IETF (Internet Engineering Task Force) 発行の『The SSL Protocol, Version 3.0』を参照してください。
- 認証方式の詳細は、**第 1 章「Oracle Advanced Security の概要」**を参照してください。

Oracle 環境における SSL の構成要素

Oracle 環境における SSL の構成要素には次のものがあります。

- [認証局](#)
- [証明書](#)
- [Wallet](#)

認証局

認証局 (Certificate Authority: CA) は、サード・パーティや他のエンティティ (ユーザー、データベース、管理者、クライアント、サーバーなど) の識別情報を証明する信頼できる第三者機関のことです。認証局では、関係者の識別情報を確認し、証明書を付与して、認証局の秘密鍵で署名します。

証明書を発行する際に必要になる個人情報は、CA ごとに異なる場合があります。ユーザーの運転免許証の提示が要求される場合や、証明書要求フォームの公証が必要な場合、または証明書を要求している人物の指紋が必要となる場合などがあります。

認証局では、認証局の公開鍵を含んだ自分自身の証明書を公開します。各ネットワーク・エンティティには、信頼できる CA の証明書のリストがあります。当該のエンティティは、他のエンティティと通信を開始する前に、相手側エンティティの証明書の署名が既知の信頼できる CA のものであることをこのリストを使用して確認します。

ネットワーク・エンティティでは、同じ CA または別の CA から証明書を取得できます。Oracle Advanced Security では、新規 Wallet のインストール時に、デフォルトで VeriSign、RSA、Entrust および GTE CyberTrust の信頼できる証明書が自動的にインストールされます。

関連項目： 7-4 ページ [「Wallet」](#)

証明書

証明書は、証明対象者の公開鍵が信頼できる認証局によって署名されたときに作成されます。証明書によって、対象者の識別情報が正しいことが保証され、公開鍵がその対象者に実際に属していることが保証されます。

証明書には証明対象者の名前、公開鍵、有効期限、シリアル番号および[証明連鎖](#)に関する情報が含まれます。証明書に関連する権限についての情報が含まれる場合もあります。

ネットワーク・エンティティが証明書を受け取ったとき、エンティティでは証明書が[信頼できる証明書](#)であること、つまり、[信頼できる認証局](#)が発行し、署名した証明書であることを確認します。証明書は、有効期限が切れるまで、または終了するまで有効です。

Wallet

Wallet は、秘密鍵、証明書および SSL に必要な信頼できる証明書も含めて、認証および署名された資格証明の格納に使用されるコンテナです。Oracle 環境で SSL を介して通信する各エンティティには、X.509 バージョン 3 の証明書、秘密鍵および信頼できる証明書の一覧を含む Wallet が必要です。

セキュリティ管理者は、Oracle Wallet Manager を使用してサーバーのセキュリティ資格証明を管理します。Wallet の所有者は、クライアントのセキュリティ資格証明を管理するのに Oracle Wallet Manager を使用します。特に、次の操作を行う場合は、Oracle Wallet Manager を使用します。

- 公開鍵と秘密鍵のペアを生成し、認証局に提出する識別情報の証明書要求を作成する場合
- Wallet にエンティティの証明書を格納する場合
- エンティティの信頼できる証明書を構成する場合

注意： Oracle Advanced Security リリース 2 (9.2) をインストールすると、Oracle Wallet Manager リリース 3.0 および Oracle Enterprise Login Assistant リリース 9.2 もインストールされます。

関連項目：

- [第 17 章「Oracle Wallet Manager の使用方法」](#)
- [17-11 ページ「Wallet の新規作成」](#)
- [17-22 ページ「信頼できる証明書の管理」](#)

Oracle 環境における SSL の機能 : SSL ハンドシェイク

SSL を介してネットワーク接続を開始する場合、クライアントとサーバーでは SSL ハンドシェイクが実行されます。SSL ハンドシェイクには、次の手順が含まれます。

- クライアントとサーバーで使用する **Cipher Suite** を確立します。この Cipher Suite には、データ転送に使用する暗号化アルゴリズムが含まれます。
- サーバーはサーバーの証明書をクライアントに送信し、クライアントは、サーバーの証明書が信頼できる CA によって署名されていることを確認します。この手順によってサーバーの識別情報が検証されます。
- クライアントの認証が必要な場合は、同様に、クライアントがクライアントの証明書をサーバーに送信し、サーバーはクライアントの証明書が信頼できる CA によって署名されていることを確認します。
- クライアントとサーバーは公開鍵暗号を使用して鍵情報を交換します。この情報に基づいて、それぞれ**セッション鍵**を生成します。これ以降、クライアントとサーバー間のすべての通信は、1 組のセッション鍵と折衝済みの **Cipher Suite** を使用して暗号化および復号化されます。

Oracle 環境における認証手続きの手順は次のとおりです。

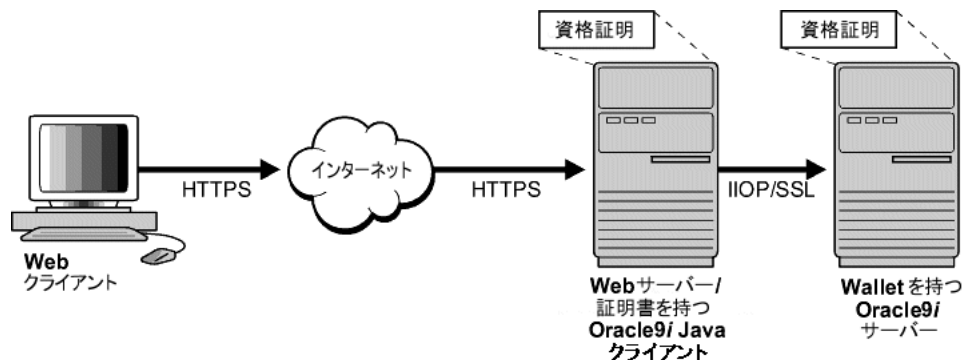
1. クライアントで、SSL を使用してサーバーへの Oracle Net 接続を開始します。
2. SSL によってクライアントとサーバー間のハンドシェイクが実行されます。
3. ハンドシェイクに成功した場合は、ユーザーがデータベースにアクセスするのに適切な**認可**を持っていることをサーバーが確認します。

非 Oracle クライアントと Oracle データベース・サーバー間の SSL

Oracle Advanced Security の SSL 機能を使用して、非 Oracle クライアントと Oracle データベース・サーバー間を安全に接続できます。たとえば、SSL を使用して、Oracle ネットワークの外部のブラウザ・クライアントが Oracle ネットワーク内の認可データに安全にアクセスできます。

図 7-1 は、SSL を使用して、インターネット上の Oracle エンティティと非 Oracle エンティティ間の接続を保護する方法を示しています。この例では、Web サーバーが Oracle9i Java クライアントとして実行されています。Web サーバーは **HTTPS** (SSL で保護された HTTP) によってメッセージを受信し、**IIOP/SSL** (SSL で保護された IIOP) によって Oracle データベース・サーバーに **CORBA** 要求を送信します。この例では、Web サーバーは、Web クライアントの証明書ではなく、サーバー自身の証明書を Oracle サーバーに渡しています。

図 7-1 インターネットから Oracle サーバーへの接続



SSL と他の認証方式の併用

Oracle Advanced Security は、サポートされている他の認証方式（Kerberos、RADIUS、CyberSafe など）と SSL を併用するように構成できます。これについては、次の項で説明します。

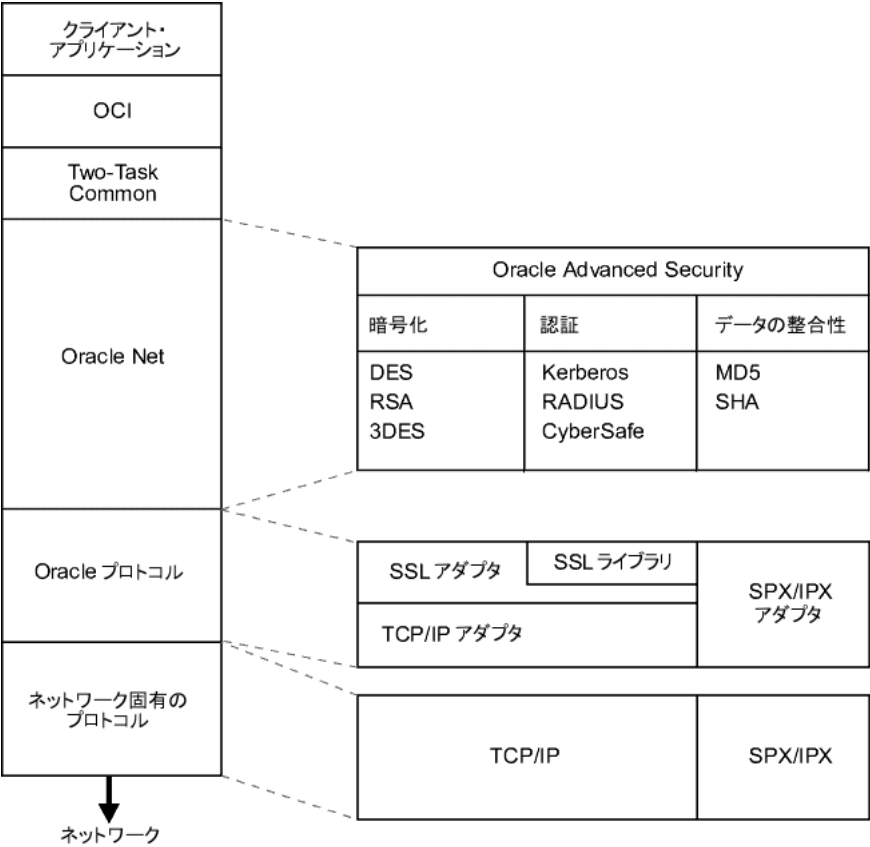
- [アーキテクチャ : Oracle Advanced Security と SSL](#)
- [SSL と他の認証方式の併用](#)

関連項目： 複数の認証方法が指定されている `sqlnet.ora` ファイルの例も含め、サポートされる他の認証方法を使用して SSL を構成する方法は、[付録 A「データの暗号化と整合性のパラメータ」](#) を参照してください。

アーキテクチャ : Oracle Advanced Security と SSL

[図 7-2](#) に、Oracle Advanced Security の実装アーキテクチャが示されています。Oracle Advanced Security は、SSL の上部にある [セッション・レイヤー](#) で動作します。SSL は [トランスポート・レイヤー](#) で TCP/IP を使用します。この機能の分離によって、サポートされている他のプロトコルと SSL を併用できます。

図 7-2 SSL と Oracle Advanced Security の関係

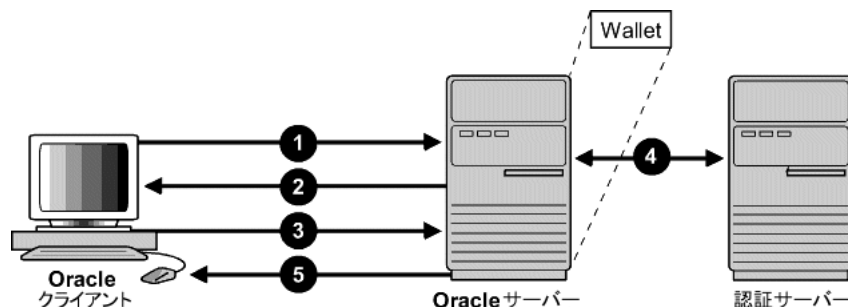


関連項目： Oracle ネットワーク環境でのスタック通信の詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。

SSL と他の認証方式の併用

図 7-3 は、SSL が Oracle Advanced Security でサポートされている他の認証方式と併用される構成を示しています。この例では、SSL を使用して初期ハンドシェイク（サーバー認証）を確立し、別の認証方式を使用してクライアントを認証します。

図 7-3 SSL と他の認証方式との関係



1. クライアントが Oracle データベース・サーバーに接続を要求します。
2. SSL がハンドシェイクを実行します。このハンドシェイク中、サーバーはクライアントに対して認証し、クライアントとサーバーは使用する Cipher Suite を確立します。
3. SSL ハンドシェイクが正常に完了すると、ユーザーがデータベースへのアクセスを要求します。
4. Oracle データベース・サーバーは、Kerberos、CyberSafe または RADIUS などの非 SSL 認証方式を使用して、ユーザーを認証サーバーで認証します。
5. 認証サーバーの確認を待って、Oracle データベース・サーバーがユーザーにアクセス権と認可を付与します。
6. ユーザーが SSL を使用して Oracle データベースに安全に接続します。

関連項目： 7-5 ページ「[Oracle 環境における SSL の機能：SSL ハンドシェイク](#)」

SSL とファイアウォール

Oracle Advanced Security は、次の 2 つのタイプのファイアウォールをサポートしています。

- アプリケーション・プロキシ・ベースのファイアウォール。Network Associates 社の Gauntlet や Axent 社の Raptor など。
- ステートフル・パケット・インスペクション型のファイアウォール。Check Point 社の Firewall-1 や Cisco 社の PIX Firewall など。

SSL を使用可能にすると、ステートフル・インスペクション型のファイアウォールはアプリケーション・プロキシ型のファイアウォールと同じように動作します。これは、ステートフル・インスペクション型のファイアウォールが暗号化パケットを復号化しないためです。

ファイアウォールは、暗号化された通信を検査しません。ファイアウォールは、イントラネット・サーバーの SSL ポート宛てのデータを検出すると、アクセス・ルールに基づいてターゲットの IP アドレスをチェックします。アクセス・ルールでは、特定の SSL ポートへの接続を許可されている SSL パケットが規定されており、その他のパケットはすべて拒否されます。

ファイアウォール・アプリケーションは、Oracle Net ファイアウォール・プロキシ・キットを使用して、データベースのネットワーク通信に対して特定のサポートを提供できます。ファイアウォールにこのプロキシ・キットが実装されていると、次の処理が発生します。

- Net Proxy (Oracle Net ファイアウォール・プロキシ・キットのコンポーネント) が、その通信をどこにルート指定すればよいか常に把握しています。
- データベース・リスナーは、SSL ハンドシェイクに参加するために、[証明書](#)へのアクセスを要求します。リスナーは SSL パケットを検査し、ターゲット・データベースを識別して、ターゲット・データベースがクライアントのリスニングに使用しているポートを返します。このポートは、SSL ポートとして指定されている必要があります。
- クライアントは、以降のすべての通信で、このサーバー指定のポートを使用して通信します。
- ファイアウォール上で開いているポートの数は、要求されたデータベース接続の数に応じて増えます。このアプローチでは、ファイアウォール上の SSL ポートがデータベースによって選択されたポートと一致する必要があるため、データベース・サーバーがランダムに選択された SSL ポートを使用することを禁止できます。この条件は、Oracle Connection Manager (Oracle Advanced Security Enterprise Edition に付属しているアプリケーション) を導入することによって回避できます。

Oracle Connection Manager を使用すると、複数の Net Manager プロトコル上のクライアント接続をルート指定できます。クライアント接続要求によって、各クライアントと Oracle Connection Manager の間に SSL 接続が確立されます。次に、Oracle Connection Manager とターゲット・データベースの間に TCP/IP 接続が確立されます。これにより、複数のクライアントがファイアウォールを通過する単一の SSL ポートを使用して、ファイアウォールの内側にある複数のデータベースに接続できます。

注意： Oracle Connection Manager を使用すれば、ファイアウォールを通過する SSL ポートが複数個開くことを回避できますが、次の点を考慮する必要があります。

- Oracle Connection Manager とデータベース間の内部接続は、SSL 接続ではありません。そのため、Oracle Advanced Security 固有の暗号化を使用して内部接続を暗号化してください。
 - 内部接続で SSL が使用されていないため、クライアントで証明書ベースの認証を使用できません。
-

SSL 使用時の問題

SSL を使用するには次の問題を考慮してください。

- SSL を使用すると、LDAP ベースのディレクトリ・サービスから認可を取り出すことができます。ディレクトリにおけるエンタープライズ・ユーザーとその権限を管理するためにクライアント側の SSL 認証が必要になります。
- SSL は認証と暗号化の両方をサポートするため、クライアントのデータベース・サーバーへの接続は、固有の暗号化を使用する標準の **Oracle Net TCP/IP** トランスポートと比べて多少遅くなります。
- 各 SSL 認証モードには、構成設定が必要となります。

注意：

- 米国政府の条例によって、二重暗号化は禁止されています。したがって、SSL 暗号化と別の暗号化方式を併用するように Oracle Advanced Security を構成すると、接続に失敗します（SSL 認証と非 SSL 認証を併用するように構成することもできません）。
 - SSL 暗号化を構成する場合は、SSL 以外の暗号化を使用禁止にする必要があります。暗号化を使用禁止にする方法は、9-2 ページの「[Oracle Advanced Security 認証を使用禁止にする方法](#)」を参照してください。
-

関連項目：

- ハードウェア・アクセラレータを使用して SSL のパフォーマンスを改善する方法は、7-30 ページの「[nCipher セキュア・アクセラレータの使用](#)」を参照してください。
- 7-12 ページ「[SSL を使用可能にする](#)」

SSL を使用可能にする

SSL を使用可能にする手順は次のとおりです。

- [タスク 1: Oracle Advanced Security および関連製品のインストール](#)
- [タスク 2: クライアントでの SSL の構成](#)
- [タスク 3: サーバーでの SSL の構成](#)
- [タスク 4: データベースへのログオン](#)

タスク 1: Oracle Advanced Security および関連製品のインストール

クライアントとサーバーの両方で Oracle Advanced Security をインストールします。Oracle Advanced Security をインストールするときに、Oracle Universal Installer によって、SSL ライブラリ、Oracle Wallet Manager および Oracle Enterprise Login Assistant がシステムに自動的にインストールされます。

関連項目： [Oracle9i プラットフォーム固有のインストレーション・ガイド](#)

タスク 2: クライアントでの SSL の構成

クライアントに SSL を構成する手順は次のとおりです。

- [手順 1: クライアントでの Wallet 作成の確認](#)
- [手順 2: サーバー DN の組込みと SSL 付き TCP/IP の使用を指定するサービス名の構成](#)
- [手順 3: 必要なクライアント構成の指定 \(Wallet の位置\)](#)
- [手順 4: クライアントに SSL Cipher Suite を設定 \(オプション\)](#)
- [手順 5: 必要な SSL バージョンの設定 \(オプション\)](#)
- [手順 6: SSL を認証サービスとして設定 \(オプション\)](#)

関連項目： 動的パラメータ名については、[付録 B「認証パラメータ」](#)を参照してください。

手順 1: クライアントでの Wallet 作成の確認

次の手順に進む前に、Wallet が作成されていることを確認してください。

関連項目：

- Wallet の詳細は、第 17 章「Oracle Wallet Manager の使用方法」を参照してください。
- 既存の Wallet をオープンする方法は、17-12 ページの「既存の Wallet のオープン」を参照してください。
- 新規 Wallet の作成方法は、17-11 ページの「Wallet の新規作成」を参照してください。

手順 2: サーバー DN の組み込みと SSL 付き TCP/IP の使用を指定するサービス名の構成

この手順は次の部分から構成されます。

- [手順 2a: サーバーの DN を指定するためのサービス名の構成](#)
- [手順 2b: SSL 付き TCP/IP を使用するためのサービス名の構成](#)

手順 2a: サーバーの DN を指定するためのサービス名の構成

Oracle Advanced Security リリース 2 (9.2) では、サーバー証明書の**識別名** (DN) とサーバーのグローバル・データベース名が照合されます。これにより、識別情報を偽っている可能性のあるサーバーに接続してしまう危険を回避できます。サーバーが有効な X.509 バージョン 3 証明書を持っていても、データベースの適切な証明書にはなりません。

DN をサーバー証明書と照合できるようにするには、tnsnames.ora ファイルを手動で編集して、サーバーの DN を指定する必要があります。サーバーの DN は、SSL_SERVER_CERT_DN パラメータを定義して指定します。クライアントは、この情報を使用して、各サーバーに対して想定している DN のリストを取得し、サーバーの DN とそのサービス名との照合を行います。[例 7-1](#) は、tnsnames.ora ファイルでの財務データベースのエントリを示しています。

例 7-1 サーバー証明書の DN と SSL 付き TCP/IP を指定するサンプル tnsnames.ora ファイル

```
finance=
(DESCRIPTION=
  (ADDRESS_LIST=
    (ADDRESS= (PROTOCOL = tcps) (HOST = finance_server) (PORT = 1575)))
  (CONNECT_DATA=
    (SERVICE_NAME= Finance.us.acme.com))
  (SECURITY=
    (SSL_SERVER_CERT_DN="cn=finance,cn=OracleContext,c=us,o=acme"))
```

tnsnames.ora ファイルは、クライアントまたは LDAP ディレクトリ上に配置できます。

また、信頼できる CA で発行された証明書の DN に、サービス名と一致する共通名 (CN) が確実に含まれていることを管理者が保証するという方法もあります。

注意： Oracle Wallet Manager を使用して、Oracle Wallet 内にある **信頼できる証明書**のうち、使用しない**認証局**に関するものを削除することをお勧めします。

関連項目：

- サーバー照合パラメータの詳細は、B-13 ページの「[SSL X.509 サーバー照合パラメータ](#)」を参照してください。
- Oracle Wallet Manager の使用方法是、[第 17 章「Oracle Wallet Manager の使用方法](#)」を参照してください。

手順 2b: SSL 付き TCP/IP を使用するためのサービス名の構成

[例 7-1](#) は、tnsnames.ora ファイルで接続プロトコルに SSL 付き TCP/IP を指定するエントリを示しています。SSL 付き TCP/IP を指定するには、tnsnames.ora ファイルの ADDRESS パラメータで、PROTOCOL に tcps を入力する必要があります。また、listener.ora ファイルの ADDRESS パラメータにも同じ情報を入力する必要があります。[例 7-2](#) は、プロトコルに SSL 付き TCP/IP を指定するエントリを示しています。

例 7-2 プロトコルに SSL 付き TCP/IP を指定するサンプル listener.ora ファイル

```
LISTENER=
  (DESCRIPTION_LIST=
    (DESCRIPTION=
      (ADDRESS= (PROTOCOL = tcps) (HOST = finance_server) (PORT = 1575))))
```

別の方法として、Oracle Net Manager を使用して SSL 付き TCP/IP を構成することもできます。

関連項目：

- 『Oracle9i Net Services 管理者ガイド』
- 『Oracle9i Net Services リファレンス・ガイド』

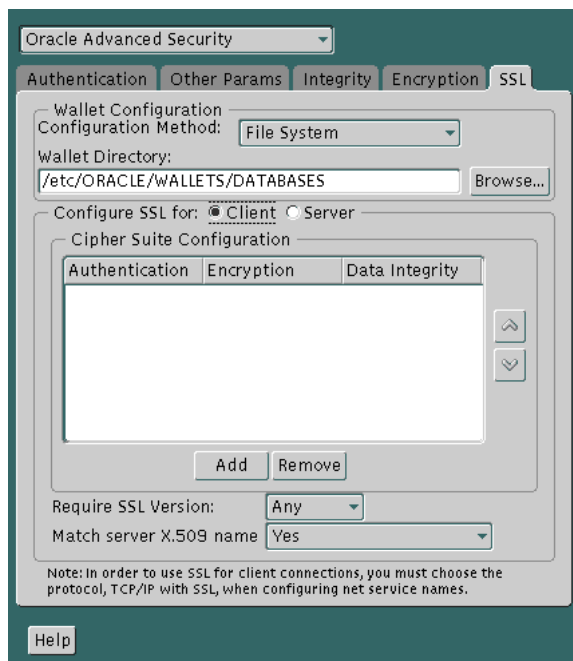
Oracle Net Manager を使用して SSL 付き TCP/IP を構成する方法は、これらのマニュアルを参照してください。

手順 3: 必要なクライアント構成の指定 (Wallet の位置)

クライアントに必要な構成パラメータを指定する手順は次のとおりです。

1. Oracle Net Manager を起動します。
 - UNIX の場合は、\$ORACLE_HOME/bin から netmgr を実行します。
 - Windows NT の場合は、「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Network Administration」→「Oracle Net Manager」の順に選択します。
2. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。
3. 右側のウィンドウ・ペインのリストから、「Oracle Advanced Security」を選択します。「Oracle Advanced Security SSL」ウィンドウが表示されます (図 7-4)。

図 7-4 Oracle Advanced Security の「SSL」ウィンドウ (クライアント)



4. 「SSL」タブを選択します。
5. 「Configure SSL for」で「Client」を選択します。

6. 「Wallet Directory」フィールドに、Oracle Wallet が格納されているディレクトリを入力するか、「Browse」ボタンをクリックし、ファイル・システムを検索してそのディレクトリを探します。

重要：

- Oracle Wallet Manager を使用して、Wallet を作成してください。
15-36 ページの「[手順 1: データベース用の Wallet の作成](#)」を参照してください。
- Oracle Net Manager を使用して、sqlnet.ora ファイルに Wallet の位置を設定してください。

Wallet の作成時および sqlnet.ora ファイルへの位置の設定時に、必ず同じ Wallet の位置を入力してください。

7. 「Match server X.509 name」ドロップダウン・リストから、次のいずれかのオプションを選択します。
 - **Yes:** サーバーの**識別名**とサービス名が一致する必要があります。SSL によって、証明書がサーバーのものであることが保証され、DN とサービス名が一致した場合のみ、接続が成功します。

注意： このチェックは、RSA 暗号が選択されている場合（これがデフォルト設定です）のみ実行できます。

- **No (デフォルト) :** SSL は DN とサービス名が一致しているかチェックしますが、その結果は使用されません。照合結果にかかわらず、接続は成功します。ただし、一致していない場合はエラーが記録されます。
- **Let Client Decide:** デフォルトを有効にします。

注意： 「No」を選択すると、次の警告が表示されます。

Security Alert

Not enforcing the server X.509 name match allows a server to potentially fake its identity. Oracle Corporation recommends selecting YES for this option so that connections are refused when there is a mismatch.

8. 「File」→「Save Network Configuration」を選択します。

クライアントで `sqlnet.ora` ファイルが更新され、次のエントリが追加されます。

```
SSL_CLIENT_AUTHENTICATION =TRUE
wallet_location =
(SOURCE=
(METHOD=File)
(METHOD_DATA=
(DIRECTORY=wallet_location)))

SSL_SERVER_DN_MATCH=(ON/OFF)
```

手順 4: クライアントに SSL Cipher Suite を設定（オプション）

Cipher Suite は、ネットワークのエンティティ間でメッセージを交換するのに使用する認証、暗号化およびデータ整合性アルゴリズムを 1 組にしたものです。SSL ハンドシェイク時に、2 つのエンティティ間で折衝し、メッセージを送受信するときに使用する Cipher Suite を確認します。

Oracle Advanced Security をインストールすると、いくつかの SSL Cipher Suite がデフォルトで設定されます。SSL_CIPHER_SUITES パラメータを設定して、デフォルトを上書きできます。たとえば、Oracle Net Manager を使用して Cipher Suite に SSL_RSA_WITH_RC4_128_SHA を追加すると、デフォルトで設定された他のすべての Cipher Suite は無視されます。

Cipher Suite は優先順位を設定できます。クライアントとサーバーで使用する Cipher Suite について折衝するとき、設定した優先順位に従って行われます。Cipher Suite の優先順位を設定するときは、次の点を考慮します。

- セキュリティのレベル。たとえば、トリプル DES 暗号化は DES よりも強力です。
- パフォーマンスへの影響。たとえば、トリプル DES 暗号化は DES よりも時間がかかります。
- 管理要件

クライアント用に選択された Cipher Suite は、サーバーで必要とされる Cipher Suite と互換性がある必要があります。たとえば、Oracle Call Interface (OCI) ユーザーの場合、サーバーではクライアント自身が認証を行う必要があります。この場合、認証の交換ができない Diffie-Hellman 匿名認証を使用した Cipher Suite は使用できません。反対に、Enterprise JavaBeans (EJB) ユーザーの場合は、サーバーでクライアント自身が認証を行う必要はありません。この場合は、Diffie-Hellman 匿名認証を使用できます。

通常、Cipher Suite の優先順位は強力なものから順に設定します。

表 7-1 は、現行のリリースの Oracle Advanced Security でサポートされている SSL Cipher Suite のリストです。これらの Cipher Suite は、Oracle Advanced Security をインストールし

たときにデフォルトで設定されます。この表には、各 Cipher Suite で使用される認証、暗号化およびデータ整合性の種類も併せて記載しています。

表 7-1 Oracle Advanced Security Cipher Suites

Cipher Suite	認証	暗号化	データの整合性
SSL_RSA_WITH_3DES_EDE_CBC_SHA	RSA	3DES EDE CBC	SHA
SSL_RSA_WITH_RC4_128_SHA	RSA	RC4 128	SHA
SSL_RSA_WITH_RC4_128_MD5	RSA	RC4 128	MD5
SSL_RSA_WITH_DES_CBC_SHA	RSA	DES CBC	SHA
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA	DH 匿名	3DES EDE CBC	SHA
SSL_DH_anon_WITH_RC4_128_MD5	DH 匿名	RC4 128	MD5
SSL_DH_anon_WITH_DES_CBC_SHA	DH 匿名	DES CBC	SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5	RSA	RC4 40	MD5
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	RSA	DES40 CBC	SHA
SSL_DH_anon_EXPORT_WITH_RC4_40_MD5	DH 匿名	RC4 40	MD5
SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA	DH 匿名	DES40 CBC	SHA

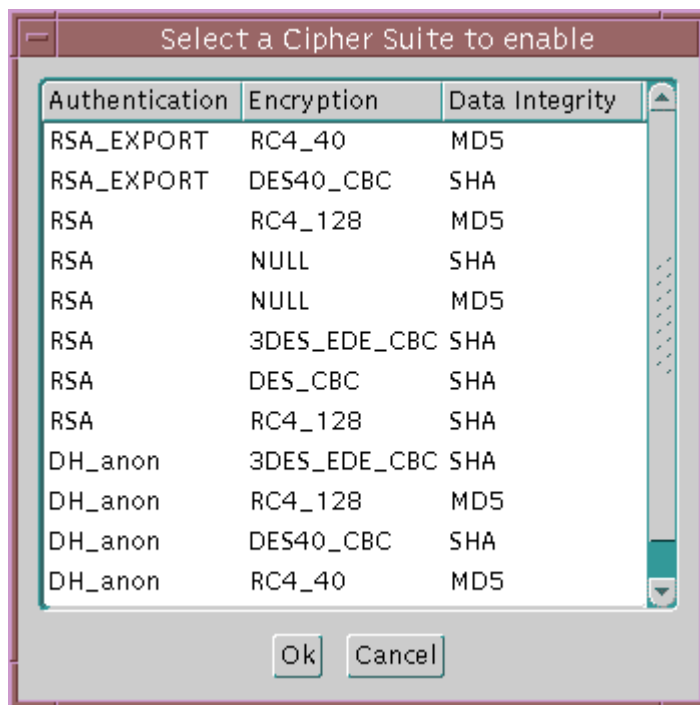
注意： sqlnet.ora ファイルで SSL_CLIENT_AUTHENTICATION パラメータが TRUE に設定されている場合は、Diffie-Hellman 匿名認証を使用するすべての Cipher Suite を使用禁止にします。使用禁止にしないと接続できません。

クライアントに Cipher Suite を指定する手順は次のとおりです。

1. Oracle Net Manager を起動します。
 - UNIX の場合は、\$ORACLE_HOME/bin から netmgr を実行します。
 - Windows NT の場合は、「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Network Administration」→「Oracle Net Manager」の順に選択します。
2. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。
3. 右側のウィンドウ・ペインのリストから、「Oracle Advanced Security」を選択します。「Oracle Advanced Security SSL」ウィンドウが表示されます (図 7-4)。
4. 「SSL」タブを選択します。

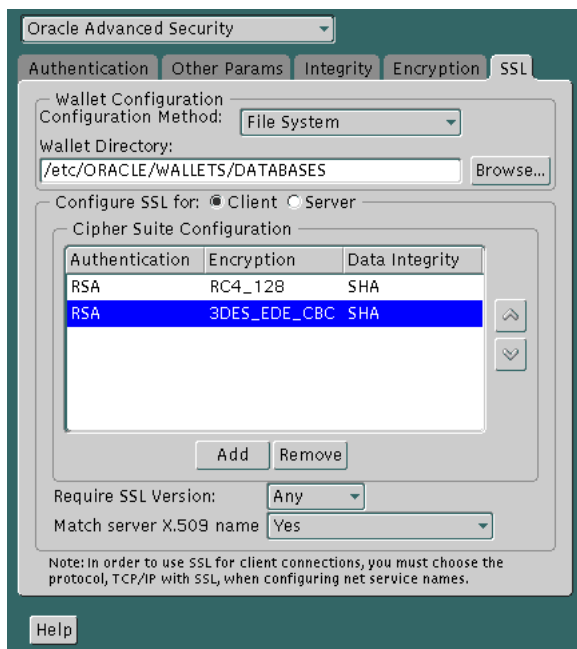
5. 「Configure SSL for」で「Client」を選択します。
6. 「Add」ボタンをクリックします。ダイアログ・ボックスに、使用可能な Cipher Suite が表示されます (図 7-5)。

図 7-5 「SSL Cipher Suites」ウィンドウ



7. Cipher Suite を選択して「OK」をクリックします。「Cipher Suite Configuration」リストが更新されます (図 7-6)。

図 7-6 Oracle Advanced Security の「SSL」ウィンドウ（クライアント）



8. 上矢印と下矢印を使用して、Cipher Suite の優先順位を設定します。
9. 「File」→「Save Network Configuration」を選択します。

sqlnet.ora ファイルが更新され、次のエントリが追加されます。

```
SSL_CIPHER_SUITES= (SSL_cipher_suite1 [,SSL_cipher_suite2])
```

手順 5: 必要な SSL バージョンの設定 (オプション)

sqlnet.ora ファイルに SSL_VERSION パラメータを設定できます。このパラメータは、クライアントの通信先のシステムで実行される必要がある SSL のバージョンを定義します。これらのシステムでは、SSL 3.0、または将来の有効なバージョンの SSL を使用する必要があります。sqlnet.ora 内でのこのパラメータのデフォルト設定は undetermined です。これは、「Oracle Advanced Security」ウィンドウの「SSL」タブのリストで「Any」を選択することによって設定されます。

クライアントに SSL のバージョンを設定する手順は次のとおりです。

1. Oracle Net Manager を起動します。
 - UNIX の場合は、\$ORACLE_HOME/bin から netmgr を実行します。
 - Windows NT の場合は、「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Network Administration」→「Oracle Net Manager」の順に選択します。
2. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。
3. 右側のウィンドウ・ペインのリストから、「Oracle Advanced Security」を選択します。「Oracle Advanced Security SSL」ウィンドウが表示されます (図 7-4)。
4. 「SSL」タブを選択します。
5. 「Configure SSL for」で「Client」を選択します。
6. 「Require SSL Version」スクロール・ボックスのデフォルトは「Any」です。このデフォルトをそのまま使用するか、構成する SSL のバージョンを選択します。
7. 「File」→「Save Network Configuration」を選択します。

sqlnet.ora ファイルが更新され、次のエントリが追加されます。

```
SSL_VERSION=UNDETERMINED
```

手順 6: SSL を認証サービスとして設定 (オプション)

sqlnet.ora ファイル内の SQLNET.AUTHENTICATION_SERVICES パラメータで、SSL 認証サービスを設定します。

SSL 認証を Oracle Advanced Security でサポートされる他の認証方式と併用する場合は、このパラメータを設定します。たとえば、サーバーがクライアントに対してサーバー自体を認証するときは SSL を使用し、クライアントがサーバーに対してクライアント自体を認証するときに Kerberos を使用する場合は、このパラメータを使用します。

SQLNET.AUTHENTICATION_SERVICES パラメータの設定:

テキスト・エディタを使用して、sqlnet.ora ファイルのこのパラメータに SSL 付き TCP/IP (TCPS) を追加します。たとえば、SSL 認証と RADIUS 認証を併用する場合は、このパラメータを次のように設定します。

```
SQLNET.AUTHENTICATION_SERVICES = (TCPS, radius)
```

SSL 認証と別の認証方式を併用しない場合は、このパラメータを設定しないでください。

タスク 3: サーバーでの SSL の構成

インストール時に、Oracle データベース・サーバーと Oracle クライアントで、Oracle Wallet の位置を除くすべての SSL パラメータにデフォルトが設定されます。サーバーで SSL を構成するには、次の手順を実行します。

- [手順 1: Wallet 作成の確認](#)
- [手順 2: 必要なサーバー構成の指定 \(Wallet の位置\)](#)
- [手順 3: サーバーに SSL Cipher Suite を設定 \(オプション\)](#)
- [手順 4: 必要な SSL バージョンの設定 \(オプション\)](#)
- [手順 5: SSL クライアント認証の設定 \(オプション\)](#)
- [手順 6: SSL を認証サービスとして設定 \(オプション\)](#)
- [手順 7: TCP/IP with SSL を使用するリスニング・エンドポイントの作成](#)

関連項目: 動的パラメータ名については、[付録 B「認証パラメータ」](#)を参照してください。

手順 1: Wallet 作成の確認

次の手順に進む前に、Wallet が作成されていることを確認してください。

関連項目:

- Wallet の詳細は、[第 17 章「Oracle Wallet Manager の使用方法」](#)を参照してください。
- 既存の Wallet をオープンする方法は、17-12 ページの「[既存の Wallet のオープン](#)」を参照してください。
- 新規 Wallet の作成方法は、17-11 ページの「[Wallet の新規作成](#)」を参照してください。

手順 2: 必要なサーバー構成の指定 (Wallet の位置)

サーバーに必要な構成パラメータを指定する手順は次のとおりです。

1. Oracle Net Manager を起動します。
 - UNIX の場合は、\$ORACLE_HOME/bin から netmgr を実行します。
 - Windows NT の場合は、「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Network Administration」→「Oracle Net Manager」の順に選択します。
2. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。
3. 右側のウィンドウ・ペインのリストから、「Oracle Advanced Security」を選択します。「Oracle Advanced Security SSL」ウィンドウが表示されます (図 7-4)。
4. 「SSL」タブを選択します。
5. 「Configure SSL for」で「Server」を選択します。
6. 「Wallet Directory」フィールドに、Oracle Wallet が格納されているディレクトリを入力するか、「Browse」をクリックし、ファイル・システムを検索してそのディレクトリを探します。

重要:

- Oracle Wallet Manager を使用して、Wallet ファイルを作成してください。15-36 ページの「[手順 1: データベース用の Wallet の作成](#)」を参照してください。
- Oracle Net Manager を使用して、sqlnet.ora ファイルに Wallet の位置を指定してください。

Wallet の作成時および sqlnet.ora ファイルへの位置の設定時に、必ず同じ Wallet の位置を入力してください。

7. 「File」→「Save Network Configuration」を選択します。

次のエントリにより sqlnet.ora ファイルと listener.ora ファイルが更新されます。

```
wallet_location =
(SOURCE=
(METHOD=File)
(METHOD_DATA=
(DIRECTORY=wallet_location)))
```

注意： リスナーは、`listener.ora` で定義されている Wallet を使用します（リスナーは任意のデータベース Wallet を使用できます）。Net Manager を使用してサーバーで SSL を構成した場合、`listener.ora` に登録された Wallet の位置と `sqlnet.ora` に登録された Wallet の位置は同じです。Oracle クライアントにとって、リスナー Wallet の位置は関係ありません。これは、クライアントがリスナーに対して実行するのが、SSL ハンドシェイクのみのためです。

リスナーが独自の Wallet を所有するためにリスナー Wallet の位置を変更するには、`listener.ora` を編集して新しい位置を入力します。

手順 3: サーバーに SSL Cipher Suite を設定（オプション）

Cipher Suite は、ネットワークのエンティティ間でメッセージを交換するのに使用する認証、暗号化およびデータ整合性アルゴリズムを 1 組にしたものです。SSL ハンドシェイク時に、2 つのエンティティ間で折衝し、メッセージを送受信するときに使用する Cipher Suite を確認します。

Oracle Advanced Security をインストールすると、いくつかの SSL Cipher Suite がデフォルトで設定されます。SSL_CIPHER_SUITES パラメータを設定して、デフォルトを上書きできます。たとえば、Oracle Net Manager を使用して Cipher Suite に SSL_RSA_WITH_RC4_128_SHA を追加すると、デフォルトで設定された他のすべての Cipher Suite は無視されます。

Cipher Suite は優先順位を設定できます。クライアントとサーバーで使用する Cipher Suite について折衝するとき、設定した優先順位に従って行われます。Cipher Suite の優先順位を設定するときは、次の点を考慮します。

- セキュリティのレベル。たとえば、トリプル DES 暗号化は DES よりも強力です。
- パフォーマンスへの影響。たとえば、トリプル DES 暗号化は DES よりも時間がかかります。
- 管理要件

サーバーに選択された Cipher Suite は、クライアントで必要とされる Cipher Suite と互換性がある必要があります。

通常、Cipher Suite の優先順位は強力なものから順に設定します。

注意： Oracle Advanced Security リリース 2 (9.2) では、Diffie-Hellman 匿名認証を使用した Cipher Suite をサーバーで設定した場合は、クライアントにも同じ Cipher Suite を設定する必要があります。同じにしないと接続できません。

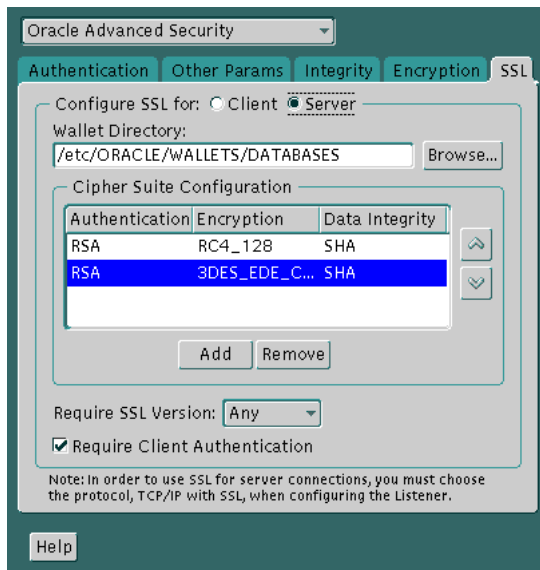
Diffie-Hellman 匿名認証を使用した Cipher Suite を使用する場合は、SSL_CLIENT_AUTHENTICATION パラメータを FALSE に設定する必要があります。7-28 ページの「[手順 5: SSL クライアント認証の設定 \(オプション\)](#)」を参照してください。

表 7-1 は、現行のリリースの Oracle Advanced Security でサポートされている SSL Cipher Suite のリストです。これらの Cipher Suite は、Oracle Advanced Security をインストールしたときにデフォルトで設定されます。この表には、各 Cipher Suite で使用される認証、暗号化およびデータ整合性の種類も併せて記載しています。

サーバーに Cipher Suite を指定する手順は次のとおりです。

1. Oracle Net Manager を起動します。
 - UNIX の場合は、\$ORACLE_HOME/bin から netmgr を実行します。
 - Windows NT の場合は、「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Network Administration」→「Oracle Net Manager」の順に選択します。
2. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。
3. 右側のウィンドウ・ペインのリストから、「Oracle Advanced Security」を選択します。「Oracle Advanced Security SSL」ウィンドウが表示されます (図 7-4)。
4. 「SSL」タブを選択します。
5. 「Configure SSL for」で「Server」を選択します。
6. 「Add」ボタンをクリックします。ダイアログ・ボックスに、使用可能な Cipher Suite が表示されます (図 7-5)。
7. Cipher Suite を選択して「OK」をクリックします。「Cipher Suite Configuration」リストが更新されます (図 7-7)。

図 7-7 Oracle Advanced Security の「SSL」ウィンドウ（サーバー）



8. 上矢印と下矢印を使用して、Cipher Suite の優先順位を設定します。

9. 「File」→「Save Network Configuration」を選択します。

sqlnet.ora ファイルが更新され、次のエントリが追加されます。

```
SSL_CIPHER_SUITES= (SSL_cipher_suite1 [,SSL_cipher_suite2])
```


手順 4: 必要な SSL バージョンの設定 (オプション)

sqlnet.ora ファイルに SSL_VERSION パラメータを設定できます。このパラメータは、クライアントの通信先のシステムで実行される必要がある SSL のバージョンを定義します。これらのシステムでは、SSL 3.0、または将来の有効なバージョンの SSL を使用する必要があります。sqlnet.ora 内でのこのパラメータのデフォルト設定は undetermined です。これは、「Oracle Advanced Security」ウィンドウの「SSL」タブのリストで「Any」を選択することによって設定されます。

サーバーに SSL のバージョンを設定する手順は次のとおりです。

1. Oracle Net Manager を起動します。
 - UNIX の場合は、\$ORACLE_HOME/bin から netmgr を実行します。
 - Windows NT の場合は、「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Network Administration」→「Oracle Net Manager」の順に選択します。
2. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。
3. 右側のウィンドウ・ペインのリストから、「Oracle Advanced Security」を選択します。「Oracle Advanced Security SSL」ウィンドウが表示されます (図 7-4)。
4. 「SSL」タブを選択します。
5. 「Configure SSL for」で「Server」を選択します。
6. 「Require SSL Version」スクロール・ボックスのデフォルトは「Any」です。このデフォルトをそのまま使用するか、構成する SSL のバージョンを選択します。
7. 「File」→「Save Network Configuration」を選択します。

sqlnet.ora ファイルが更新され、次のエントリが追加されます。

```
SSL_VERSION=UNDETERMINED
```

注意： SSL 2.0 はサーバー側ではサポートされていません。

手順 5: SSL クライアント認証の設定（オプション）

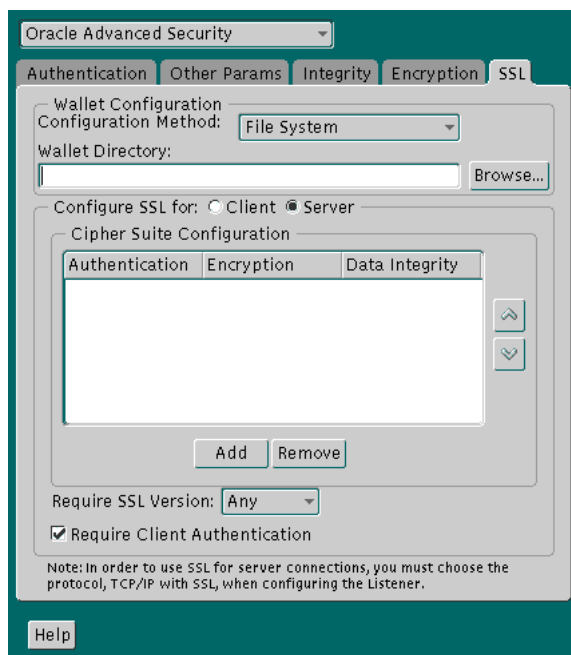
クライアント認証で SSL を使用するかどうかは `sqlnet.ora` ファイル内の `SSL_CLIENT_AUTHENTICATION` パラメータで制御します。デフォルト値は `TRUE` です。

Diffie-Hellman 匿名認証（`DH_anon`）を含む Cipher Suite を使用する場合は、このパラメータを `FALSE` に設定する必要があります。また、サーバーに対してクライアントが認証するときに、Kerberos や CyberSafe など、Oracle Advanced Security でサポートされている非 SSL 認証方式を使用する場合にも、このパラメータを `FALSE` に設定する必要があります。

このパラメータを `FALSE` に設定する手順は次のとおりです。

1. Oracle Net Manager を起動します。
 - UNIX の場合は、`$ORACLE_HOME/bin` から `netmgr` を実行します。
 - Windows NT の場合は、「スタート」→「プログラム」→「Oracle - *HOME_NAME*」→「Network Administration」→「Oracle Net Manager」の順に選択します。
2. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。
3. 右側のウィンドウ・ペインのリストから、「Oracle Advanced Security」を選択します。「Oracle Advanced Security SSL」ウィンドウが表示されます（図 7-8）。

図 7-8 Oracle Advanced Security の「SSL」ウィンドウ（サーバー）



4. 「SSL」タブを選択します。
5. 「Configure SSL for」で「Server」を選択します。
6. 「Require Client Authentication」の選択を解除します。
7. 「File」→「Save Network Configuration」を選択します。
sqlnet.ora ファイルが更新され、次のエントリが追加されます。
SSL_CLIENT_AUTHENTICATION=FALSE

手順 6: SSL を認証サービスとして設定（オプション）

sqlnet.ora ファイル内の SQLNET.AUTHENTICATION_SERVICES パラメータで、SSL 認証サービスを設定します。

SSL 認証を Oracle Advanced Security でサポートされる他の認証方式と併用する場合は、このパラメータを設定します。たとえば、サーバーがクライアントに対してサーバー自体を認証するときは SSL を使用し、クライアントがサーバーに対してクライアント自体を認証するときに Kerberos を使用する場合は、このパラメータを使用します。

SQLNET.AUTHENTICATION_SERVICES パラメータの設定：

テキスト・エディタを使用して、sqlnet.ora ファイルのこのパラメータに SSL 付き TCP/IP (TCPS) を追加します。たとえば、SSL 認証と RADIUS 認証を併用する場合は、このパラメータを次のように設定します。

```
SQLNET.AUTHENTICATION_SERVICES = (TCPS, radius)
```

SSL 認証と別の認証方式を併用しない場合は、このパラメータを設定しないでください。

手順 7: TCP/IP with SSL を使用するリスニング・エンドポイントの作成

listener.ora ファイル内に SSL 付き TCP/IP のリスニング・エンドポイントを指定して、リスナーを構成します。一般的な Oracle Net クライアントにはポート番号 2484 を、Oracle9i JServer へのクライアント接続にはポート番号 2482 を使用することをお勧めします。

関連項目：『Oracle9i Net Services 管理者ガイド』

タスク 4: データベースへのログオン

SSL 認証を使用している場合は SQL*Plus を起動して、次のように入力します。

```
CONNECT/@dnet_service_name
```

SSL 認証を使用していない場合は SQL*Plus を起動して、次のように入力します。

```
CONNECT username/password@net_service_name
```

nCipher セキュア・アクセラレータの使用

SSL ハンドシェイク操作は、システムに対して多くの処理を要求するため、サーバーおよびトランザクションのパフォーマンスが低下する可能性があります。SSL ハードウェア・アクセラレータは、他のトランザクションに応答するために CPU を解放することによって、SSL 処理をサーバーからオフロードします。Oracle Advanced Security では、nCipher BSAFE Hardware API (BHAPI) を使用して、SSL ハードウェア・アクセラレーションをサポートします。このインタフェースを使用して、Oracle は nCipher セキュア・アクセラレータと統合します。

注意： nCipher のカードおよびソフトウェアの認定バージョンを取得するには、nCipher の代理店に連絡する必要があります。

nCipher セキュア・アクセラレータを使用するために必要な Oracle のコンポーネント

nCipher セキュア・アクセラレータを使用するには、次のコンポーネントが必要です。

- nCipher セキュア・アクセラレータ
- プラットフォーム対応の、サポートされる nCipher BHAPI ライブラリ
 - (Unix の場合) libnfbhapi.so ライブラリ
 - (Windows NT の場合) nfbhapi.dll ライブラリ

注意： 保護アクセラレータをインストールし、必要なライブラリを取得するには、nCipher の代理店に連絡する必要があります。

nCipher セキュア・アクセラレータを Oracle Advanced Security で使用するには、次のタスクを実行する必要があります。

nCipher セキュア・アクセラレータを使用するための Oracle Advanced Security の構成

セキュア・アクセラレータを使用するには、nCipher BHAPI ライブラリが格納されているディレクトリへのパスを次の場所に配置する必要があります。

- (UNIX の場合) ユーザーの LD_LIBRARY_PATH
- (Windows の場合) ユーザーの PATH

これによって、実行時にライブラリがロードされるようになります。通常、nCipher カードは次の位置にインストールされます。

- (UNIX の場合) /opt/nfast
- (Windows の場合) C:\nfast

nCipher BHAPI ライブラリは、セキュア・アクセラレータがインストールされている次のディレクトリ位置にあります。

/toolkits/nfbhapi/

nCipher セキュア・アクセラレータの使用に関するトラブルシューティング

SQL*Net のトレースをオンにすると、nCipher アクセラレータが使用されているかどうかを検出できます。nCipher ソフトウェアが使用されている場合は、SQL*Net のトレース・ファイルで次のエントリを確認できます (entry から exit までの間に記録されたエラー・メッセージはありません)。

```
nzos_initbhapi: entry
nzos_initbhapi: exit
```

nCipher セキュア・アクセラレータの使用に関連する SQL*Net トレース・ファイルのエラー・メッセージ

SQL*Net トレース・ファイルの entry から exit までのエントリの間にエラー・メッセージが記録されている場合は、エラーを解決するために、次のエラー・メッセージのリストをチェックしてください。

nzos_initbhapi: Failed to load libnfbhapi.so. BHAPI will not be used

原因: nCipher BHAPI ライブラリの位置を特定できません。

処置: nCipher BHAPI ライブラリが含まれるディレクトリがユーザーのシステム・パスにあることを確認してください。

関連項目: nCipher BHAPI ライブラリへのパスをユーザーのシステム・パスに組み込む方法は、7-31 ページの「[nCipher セキュア・アクセラレータを使用するための Oracle Advanced Security の構成](#)」を参照してください。

nzos_initbhapi: Error in B_CreateSessionChooser. Returned <err #>

原因： nCipher セキュア・アクセラレータが実行されていない可能性があります。

処置： nCipher カードがインストールされているディレクトリから /bin/enquiry ユーティリティを実行し、セキュア・アクセラレータが起動されていることを確認してください。

注意： nCipher ログ・ファイルは、セキュア・アクセラレータがインストールされている次のディレクトリ位置にあります。

/log/logfile

関連項目： トラブルシューティングの詳細は、nCipher のマニュアルを参照してください。

Entrust 対応の SSL 認証の構成

この章では、Secure Socket Layer (SSL) 認証のための、Entrust 対応 Oracle Advanced Security の構成および使用方法について説明します。次の項目について説明します。

- [概要](#)
- [システムの構成要素](#)
- [Entrust 認証手続き](#)
- [Entrust 認証を使用可能にする](#)
- [問題点と制限事項](#)
- [Oracle Advanced Security における Entrust のトラブルシューティング](#)

概要

公開鍵インフラストラクチャには、デジタル証明書にバインドされる公開鍵、秘密鍵およびその他の特定のセキュリティ資格証明などの様々な要素が含まれています。これらの資格証明は、**Secure Sockets Layer (SSL)** 接続上での安全な認証、セキュリティで保護された通信チャネルの確立、およびデジタル署名を含むデジタル証明書の生成と処理に使用できます。完全な PKI には次の機能があります。

- 証明書取消しステータス・チェック
- ユーザーの鍵および証明書を容易に管理
- PKI の複雑さをユーザーに意識させずに容易に配置

この項では、次のものによって提供される PKI 実装について説明します。

- **Oracle Advanced Security**
- **Entrust/PKI**
- **Entrust 対応の Oracle Advanced Security**

Oracle Advanced Security

Oracle Advanced Security には、ユーザーの**公開鍵と秘密鍵のペア**や**トラスト・ポイント**（ユーザーが信頼するルート証明書のリスト）を作成し、安全に保管する Oracle Wallet Manager などの PKI の要素が組み込まれています。Oracle Wallet Manager に格納されたユーザーの PKI 資格証明は、SSL を介した安全な認証済みセッションを作成するために使用されます。ただし、Oracle Advanced Security は、完全な PKI の重要な要素である**証明書**の作成や証明書取消しステータス・チェックの機能は提供していません。

たとえば、Oracle Wallet Manager は PKCS#10 証明書署名要求を生成できますが、ユーザーは署名済みの証明書を**認証局**から取得し、結果の証明書を Oracle Wallet にロードする必要があります。Oracle Wallet は、Oracle アプリケーションに対する認証のみをサポートします。

Entrust/PKI

Entrust/PKI は、Entrust Technologies 社が提供する PKI 製品で、証明書生成、証明書取消し、および鍵と証明書の管理機能があります。

Entrust 対応の Oracle Advanced Security

Oracle Advanced Security と Entrust/PKI を統合すると、Entrust と Oracle の両方のユーザーが、Entrust の拡張 PKI 機能を利用してその Oracle 環境のセキュリティを強化できます。

Entrust 対応の Oracle Advanced Security は次の機能を提供します。

- 拡張 X.509 ベースの認証とシングル・サインオン
- Entrust/PKI 鍵管理との統合
- Entrust/PKI 証明書取消しとの統合

注意：

- Oracle Advanced Security は、リリース 8.1.7 で、Entrust Technologies 社による Entrust-Ready の認定を受けています。
 - <http://www.entrust.com> を参照してください。
-
-

拡張 X.509 ベースの認証とシングル・サインオン

Entrust 対応の Oracle Advanced Security では、X.509 ベースの認証とシングル・サインオンのために Entrust 資格証明が使用できます。Oracle Advanced Security では、Oracle Wallet を使用してユーザーの PKI 資格証明を保持するかわりに、Entrust/Authority によって作成され、Entrust プロファイル（.epf ファイル）に格納されている PKI 資格証明にアクセスできます。企業内で Entrust ソフトウェアを配置しているユーザーは、Entrust を使用して、Oracle9i に対する認証およびシングル・サインオンを行うことができます。

Entrust/PKI 鍵管理との統合

Entrust 対応の Oracle Advanced Security は、Entrust/PKI が提供する広範な鍵管理およびロールオーバー機能を使用します。これらの機能は、PKI 配置の複雑さをユーザーに意識させません。たとえば、ユーザーは証明書の有効期限が切れると自動的に通知を受け、管理者が構成できるプリファレンスに従って証明書が再発行されます。

Entrust/PKI 証明書取消しとの統合

Entrust は、証明書取消しステータスをシステムでチェックし、証明書の取消しを可能にする認証局コンポーネントを提供しています。

Oracle に対する認証に Entrust 資格証明を使用するユーザーは、証明書の取消しステータスがチェックされ、証明書が取り消されている場合は接続を拒否されます。

システムの構成要素

この項では、Entrust 対応の Oracle Advanced Security を使用するために必要なシステムの構成要素について説明します。

- [Entrust/PKI 6.0 for Oracle](#)
- [Entrust/Server Login Toolkit 6.0](#)
- [Entrust/IPSEC Negotiator Toolkit 6.0](#)

注意： 以降の項では、[クライアント](#)という用語は Oracle データベースに接続するクライアントを指し、[サーバー](#)という用語は Oracle データベースが常駐しているホストを指します。

Entrust/PKI 6.0 for Oracle は、Entrust の次の Web サイトからダウンロードできます。

<http://www.entrust.com>

Entrust/Server Login Toolkit および Entrust/IPSEC Negotiator Toolkit は、登録メンバーが Entrust Developer Network からダウンロードできます。メンバーシップに登録し、各製品をダウンロードするには、次の Web アドレスにアクセスします。

<http://www.entrust.com/developer/memberships/registration.htm>

注意： Oracle Advanced Security は、Entrust/PKI バージョン 5.0.2、5.1 および 6.0 をサポートします。

Entrust/PKI 6.0 for Oracle

Entrust/PKI 6.0 for Oracle には、Entrust ユーザーとインフラストラクチャに関する情報を格納するためのデータベースと、ユーザー名、公開証明書および証明書取消しリストなどの情報を格納するための Lightweight Directory Access Protocol (LDAP) 準拠のディレクトリが必要です。

Entrust/PKI 6.0 for Oracle は、次のソフトウェア・コンポーネントで構成されています。

- [Entrust/Authority](#)
- [Entrust/RA](#)
- [Entrust/Entelligence](#)

Entrust/Authority

Entrust/Authority は Entrust/PKI の中心要素です。Entrust/Authority は、認証局、証明書およびユーザー管理（例：ユーザーの作成やユーザーの資格証明を含んだユーザー・プロフィールの作成）の中心的な機能を実行します。

注意： Entrust 対応の Oracle Advanced Security は、Oracle9i で動作するバージョンの Entrust/Authority とのみ併用できます。

関連項目： 認証局の詳細は、[第 7 章「Secure Sockets Layer 認証の構成」](#)を参照してください。

Entrust/Authority は、自動ログイン（サーバー・ログインとも呼ばれます）をサポートしています。このログイン方法では、[データベース管理者](#)（DBA）は、Entrust プロファイルのパスワードをサーバーで繰り返し入力する必要はありません。自動ログインを使用すると、DBA はパスワードを 1 回入力するのみで、サーバーの Entrust プロファイルをオープンして、複数の接続要求に対して自身を認証できます。

Entrust/RA

Entrust/RA は、Entrust/Authority に対する管理者用の安全なインタフェースです。

Entrust/Entelligence

Entrust/Entelligence は、SSL 接続に対する Oracle9i サーバー・プロセスのアクセスを使用可能にすることで、クライアントとサーバーの両方でユーザーの鍵管理およびシングル・サインオン機能をサポートします。

注意： Entrust/Entelligence が Windows プラットフォームで実行されている場合は、サーバー・コンピュータに Entrust/Entelligence をインストールしないでください。

Entrust/Server Login Toolkit 6.0

Entrust/Server Login Toolkit 6.0 は、UNIX プラットフォームで動作するサーバーでシングル・サインオン機能を実現するために必要です。

Entrust/Server Login Toolkit は、SSL 接続に対する Oracle9i サーバー・プロセスのアクセスを可能にすることでシングル・サインオン機能を提供します。この機能がないと、データベース管理者または他の権限ユーザーは、すべての接続ごとに、サーバーで Entrust プロファイル用のパスワードを入力する必要があります。

Entrust/Server Login Toolkit は、Entrust 社の Web サイトからダウンロードできます。アドレスは次のとおりです。

http://www.entrust.com/developer/software/files/desc_serverlogin.cfm

Entrust/IPSEC Negotiator Toolkit 6.0

Entrust/IPSEC Negotiator Toolkit 6.0 は、Oracle Advanced Security の SSL スタックと Entrust/PKI を統合して、SSL 認証で Entrust プロファイルを使用するために、クライアントとサーバーの両方で必要です。

IPSEC Negotiator Toolkit は、Entrust 社の Web サイトからダウンロードできます。アドレスは次のとおりです。

<http://www.entrust.com/developer/software/index.htm>

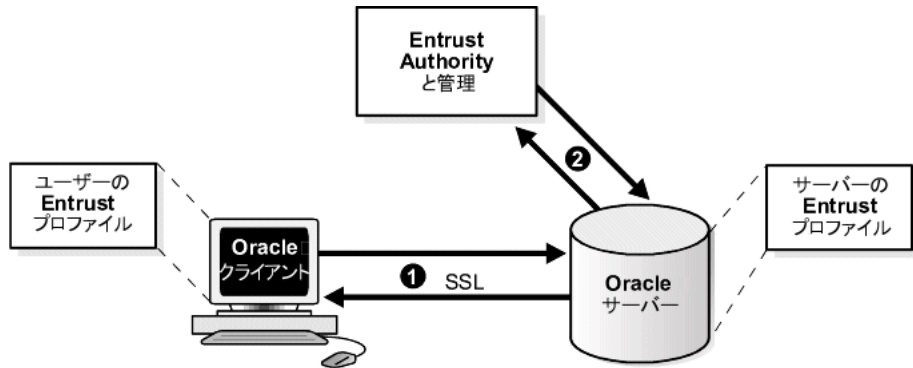
Entrust 認証手続き

図 8-1 は、次の Entrust 認証手続きを示しています。

1. Oracle クライアントの Entrust ユーザーは、SSL と Entrust 資格証明を使用してサーバーとの安全な接続を確立します。
2. サーバーの Oracle SSL アダプタは、Entrust Authority と通信して、Entrust ユーザーの証明書取消しステータスをチェックします。

注意： 図 8-1 では、クライアント・プロファイルとサーバー・プロファイルは作成済みとみなされ、作成手順は含まれていません。

図 8-1 Entrust 認証手続き



関連項目： 7-5 ページ「[Oracle 環境における SSL の機能 : SSL ハンドシェイク](#)」

Entrust 認証を使用可能にする

この項では、Entrust 対応の Oracle Advanced Security SSL 認証を使用可能にする次の作業について説明します。

- [Entrust プロファイルの作成](#)
- [Oracle Advanced Security および関連製品のインストール](#)
- [クライアントおよびサーバーにおける SSL の構成](#)
- [クライアントにおける Entrust の構成](#)
- [サーバーにおける Entrust の構成](#)
- [データベース・ユーザーの作成](#)
- [データベースへのログイン](#)

Entrust プロファイルの作成

この項では、Entrust プロファイルの作成方法について説明します。Entrust プロファイルは、管理者またはユーザーが作成できます。UNIX プラットフォームでは、管理者がすべてのクライアントの Entrust プロファイルを作成します。Windows プラットフォームでは、ユーザーが各自の Entrust プロファイルを作成できます。

管理者による Entrust プロファイルの作成

管理者は次の手順で Entrust プロファイルを作成します。

1. Entrust 管理者は、Entrust/RA ツールを使用して Entrust ユーザーを追加します。

関連項目： Entrust ユーザーの作成方法は、Entrust の管理マニュアルを参照してください。

2. 管理者はユーザー名とパスワードを入力します。
3. Entrust Authority によって、プロファイル .epf ファイルが作成されます。
4. 管理者は、プロファイルに関連したすべてのファイルをユーザーに安全に送信します。事前に設定したパスワードは、ユーザーが変更できます。

ユーザーによる Entrust プロファイルの作成

Entrust ユーザーは次の手順で自分自身の Entrust プロファイルを作成します。

1. Entrust 管理者は、Entrust/RA ツールを使用して Entrust ユーザーを追加します。その際、「New User」ダイアログ・ボックスで、「Create Profile」オプションの選択を解除する必要があります。

関連項目： Entrust プロファイルの作成方法は、Entrust の管理マニュアルを参照してください。

2. ユーザーは管理者から保護電子メール通知を受信します。この通知には、参照番号、認可コードおよび有効期限が含まれています。
3. ユーザーは、次のように Entrust/Entelligence の「Create Entrust Profiles」画面にナビゲートします。

「スタート」→「プログラム」→「Entrust」→「Entrust Profiles」→「Create Entrust Profiles」

4. ユーザーは、電子メール通知で提供された参照番号、認可コードおよび有効期限を入力します。プロファイル .epf ファイルと Entrust 初期化ファイルが作成されます。

Oracle Advanced Security および関連製品のインストール

Oracle Advanced Security リリース 2 (9.2) では、Typical モードを選択すると、Entrust サポートがインストールされます。単一の Oracle で、Oracle Wallet と Entrust プロファイルの両方を使用できます。

関連項目： オペレーティング・システム固有の Oracle9i インストレーション・ガイド

注意：

- UNIX サーバーに Entrust をインストールした場合は、以前のリリースと異なるパラメータが使用されます。
 - 8-11 ページの「UNIX サーバーにおける Entrust の構成」を参照してください。
-

クライアントおよびサーバーにおける SSL の構成

クライアントおよびサーバーで SSL を構成します。

関連項目： クライアントおよびサーバーで SSL を構成する方法は、[第 7 章「Secure Sockets Layer 認証の構成」](#)を参照してください。Oracle Wallet 位置に関する項はスキップしてください。

クライアントにおける Entrust の構成

サーバーで Entrust を構成する手順は、プラットフォームの種類によって異なります。

- [UNIX クライアントにおける Entrust の構成](#)
- [Windows クライアントでの Entrust の構成](#)

UNIX クライアントにおける Entrust の構成

クライアントが Windows 以外のプラットフォームで動作している場合は、次の手順を実行します。

1. 環境変数 JAVA_HOME を JDK または JRE の位置に設定します。

例：

```
>setenv JAVA_HOME $ORACLE_HOME/JRE
```

2. sqlnet.ora ファイルに WALLET_LOCATION を設定します。

例：

```
WALLET_LOCATION=
(SOURCE=
(METHOD=entr)
(METHOD_DATA =
(PROFILE=profile_location)
(INIFILE=initialization_file_location)
)
)
```

Windows クライアントでの Entrust の構成

クライアントが Windows プラットフォームで動作している場合は、クライアントに Entrust/Entelligence コンポーネントがインストールされていることを確認してから、次の手順を実行して Entrust 資格証明を設定します。

1. `sqlnet.ora` ファイルに `WALLET_LOCATION` パラメータを設定します。

例：

```
WALLET_LOCATION=
  (SOURCE=
    (METHOD=entr)
    (METHOD_DATA=
      (INIFILE=initialization_file_location)
    )
  )
```

`initialization_file_location` は、`.ini` ファイルへのパスです。

2. システム・トレイで「Entrust」アイコンを選択して、「Entrust_Login」ダイアログ・ボックスをオープンします。
3. プロファイル名とパスワードを入力して Entrust にログオンします。

サーバーにおける Entrust の構成

サーバーで Entrust を構成する手順は、プラットフォームの種類によって異なります。

- [UNIX サーバーにおける Entrust の構成](#)
- [Windows サーバーでの Entrust の構成](#)

UNIX サーバーにおける Entrust の構成

サーバーが UNIX プラットフォームの場合は、Entrust/Server Login Toolkit コンポーネントがインストールされていることを確認し、次の手順を実行します。

関連項目： Entrust/Server Login Toolkit のダウンロード方法は、8-4 ページの「[システムの構成要素](#)」を参照してください。

1. Oracle データベース・インスタンスを停止します。
2. 次のように、`sqlnet.ora` ファイルと `listener.ora` ファイルに `WALLET_LOCATION` を設定し、サーバーのプロファイルと Entrust 初期化ファイルへのパスを指定します。

```
WALLET_LOCATION =
  (SOURCE =
    (METHOD = ENTR)
    (METHOD_DATA =
      (PROFILE = profile_location)
      (INIFILE = initialization_file_location)
    )
  )
```

3. 次のパスが含まれるように、環境変数 `CLASSPATH` を設定します。

```
$ORACLE_HOME/JRE/lib/rt.jar
$ORACLE_HOME/JRE/lib/i18n.jar
$ORACLE_HOME/jlib/ewt*.jar
$ORACLE_HOME/jlib/help*.jar
$ORACLE_HOME/jlib/share*.jar
$ORACLE_HOME/jlib/swingall*.jar
$ORACLE_HOME/network/jlib/netentrust.jar
```

4. 次の手順で `etbinder` コマンドを入力し、自動ログイン資格証明 `.ual` ファイルを作成します。

- a. `etbinder` コマンドへのパスが含まれるように、環境変数 `PATH` を設定します。このコマンドは、`Server Login Toolkit` がインストールされている `/bin` ディレクトリにあります。
- b. Entrust ライブラリへのパスが含まれるように `LD_LIBRARY_PATH` を設定します。
- c. Entrust 初期化ファイルへの完全パスが含まれるように、環境変数 `SSL_ENTRUST_INI` を設定します。
- d. 次のコマンドを入力します。

```
etbinder
```

- e. プロファイル・ファイルの位置を入力するプロンプトが表示された場合は、ファイル名を含む完全パス名を入力します。次にプロンプトが表示されたときは、パスワードを入力します。

資格証明ファイル (`filename.ual`) が作成されたことを示すメッセージが表示されます。

注意： リスナーに TCPS リスニング・エンドポイントがあることを確認してから、リスナーを開始してください。

5. Oracle データベース・インスタンスを開始します。

Windows サーバーでの Entrust の構成

サーバーが Windows プラットフォームで動作している場合は、次の手順を実行します。

関連項目： Entrust/Entelligence のダウンロード方法は、「[システムの構成要素](#)」を参照してください。

1. Oracle データベース・インスタンスを停止します。

2. 次のように、`sqlnet.ora` ファイルと `listener.ora` ファイルに `WALLET_LOCATION` を設定し、サーバーのプロファイルと Entrust 初期化ファイルへのパスを指定します。

```
WALLET_LOCATION =
  (SOURCE =
    (METHOD = ENTR)
    (METHOD_DATA =
      (PROFILE = profile_location)
      (INIFILE = initialization_file_location)
    )
  )
```

3. Entrust の `binder` コマンドを実行して、自動ログイン資格証明を作成します。自動ログイン資格証明は、拡張子が `.ual` のファイルです。`.ual` ファイルの所有者が、Oracle サービスの所有者と同じであることを確認します。

次の順に選択して、`binder` コマンドを実行します。

「スタート」→「プログラム」→「Entrust Toolkit」→「Server Login」→「Entrust Binder」

プロファイルへのパス、パスワード、および Entrust 初期化ファイルへのパスを入力します。メッセージが表示され、資格証明ファイルが正常に作成されたことが通知されます。

4. Oracle データベース・インスタンスを開始します。

注意： Windows 環境の場合は、次のことをお勧めします。

- サーバー・コンピュータには Entrust/Entelligence をインストールしないでください。
 - 自動ログイン資格証明（`.ual` ファイル）の生成、および Entrust/Entelligence がインストールされている Windows クライアントからのデータベースへのアクセスには、Server Login Toolkit を使用してください。
-

データベース・ユーザーの作成

各 Entrust ユーザーの**識別名**に基づいて、データベースにグローバル・ユーザーを作成します。

例：

```
SQL> create user jdoe identified globally as
'cn=jdoe,o=oracle,c=us';
```

"cn=jdoe, o=oracle, c=us" は、ユーザーの Entrust 識別名です。

データベースへのログイン

1. 次のように、SQL*Plus を使用して Oracle インスタンスに接続します。

```
sqlplus /@tns_service_name
```


`tns_service_name` は Oracle インスタンスのサービス名です。
「Entrust_Login」ダイアログ・ボックスが表示されます。
2. プロファイルへのパスとパスワードを入力します。
3. `WALLET_LOCATION` パラメータに値を指定していない場合は、Entrust 初期化ファイルへのパスを入力するよう求められます。

注意： `WALLET_LOCATION` パラメータを含むファイルに初期化ファイルを指定することをお勧めします。

問題点と制限事項

Entrust と動作するようにアプリケーションを特別に修正する必要があります。製品が Entrust-ready として設計されている場合、その製品は Entrust のツールキットを使用して Entrust と統合されています。

たとえば、Oracle は、その SSL ライブラリが Oracle Wallet ではなく Entrust プロファイルにアクセスするように修正しています。したがって、Entrust プロファイルは標準の SSL ライブラリからはアクセスできません。

さらに、次の制限事項が適用されます。

- Oracle をベースとしたアプリケーションで、デジタル署名に対して Entrust コンポーネントは使用できません。
- Entrust 対応の Oracle Advanced Security との統合がサポートされているのは、Oracle9i で動作する Entrust/PKI 5.0.2、5.1 および 6.0 のみです。
- Entrust 対応の Oracle Advanced Security で以前のリリースの Entrust/Authority を使用することはサポートされていません。
- Entrust PKI および非 Entrust PKI 間の相互運用性はサポートされていません。
- リリース 8.1.7 用の Oracle Internet Directory リリース 2.1.1 およびそれ以降のリリースは、Entrust によってすでに認定されています。

Oracle Advanced Security における Entrust のトラブルシューティング

この項では、Entrust から Oracle Advanced Security ユーザーに返されるエラーの診断方法について説明します。

注意： Entrust は、Oracle Advanced Security ユーザーに次の一般的なエラー・メッセージを返します。

ORA-28890 委任ログインに失敗しました。

この項では、このエラー・メッセージの基になるエラーの詳細を取得する方法と、問題の診断方法について説明します。

プラットフォームに関係なく Entrust 実行時に戻るエラー・メッセージ

Entrust を実行しているプラットフォームの種類に関係なく、次のエラー・メッセージが表示される可能性があります。

ORA-28890 委任ログインに失敗しました。

原因： Entrust 対応の Oracle クライアントで SQL*Plus を使用してログインしようとすると、この一般エラー・メッセージが表示され、ログインできません。このエラーは、次の原因などの様々な問題によって発生する可能性があります。

- Entrust/Authority がオンラインでない
- 指定した Entrust プロファイル・パスワードが無効
- 指定した Entrust プロファイルへのパスが無効
- 指定した Entrust 初期化ファイルが無効
- Entrust Server Login プログラムがサーバー上で実行されなかった

処置： Entrust のエラーの詳細を取得するには、SQL*Plus のトレースをオンにして、トレースに Entrust のエラー・コードが出力されるようにします。sqlnet.ora ファイルに次のパラメータを設定すると、トレースを使用可能にできます。

クライアント

- TRACE_LEVEL_CLIENT=16
- TRACE_DIRECTORY_CLIENT=<valid_client_directory_name>
- TRACE_FILE_CLIENT=client
- TRACE_UNIQUE_CLIENT=ON

サーバー

- `TRACE_LEVEL_SERVER=16`
- `TRACE_DIRECTORY_SERVER=<valid_server_directory_name>`
- `TRACE_FILE_SERVER=server`
- `TRACE_UNIQUE_SERVER=ON`

作成されたトレース・ファイル内で文字列 `IKMP` を検索します。エラー・メッセージは、この文字列の付近にあり、発生した問題の詳細を確認できます。この詳細なエラー・コード情報は、Entrust API によって戻されます。

注意： 次のディレクトリは、`sqlnet.ora` ファイルの `TRACE_DIRECTORY_CLIENT` パラメータまたは `TRACE_DIRECTORY_SERVER` パラメータを設定する際に有効なクライアント・ディレクトリ名の例です。

- (UNIX の場合) `/tmp`
 - (Windows の場合) `C:\TEMP`
-
-

ORA-28890 委任ログインに失敗しました。

(クライアントには表示されません)

原因： `WALLET_LOCATION` パラメータに、クライアント側の `sqlnet.ora` ファイルの Entrust 初期化ファイルの位置が指定されていません。

処置： Entrust 初期化ファイルの位置が、クライアント上の `sqlnet.ora` ファイルの `WALLET_LOCATION` パラメータに指定されていることを確認してください。

関連項目：

- 8-10 ページ [「UNIX クライアントにおける Entrust の構成」](#)
- 8-10 ページ [「Windows クライアントでの Entrust の構成」](#)

Windows プラットフォームで Entrust 実行時に戻るエラー・メッセージ

Windows プラットフォームで Entrust を実行している場合、次のエラー・メッセージが表示される可能性があります。

The software authentication failed. (error code - 162).

原因: 既知の FIPS モードの非互換性が原因で Entrust へのログインが失敗し、このエラー・メッセージが戻る可能性があります。

処置: Entrust 社のサポートに連絡を取り、この問題を解決してください。

Algorithm self-test failed. (error code - 176).

原因: Entrust と Oracle ライブラリ間での既知の記号の競合が原因で Entrust へのログインが失敗し、このエラー・メッセージが戻る可能性があります。

処置: Entrust 社のサポートに連絡を取り、この問題を解決してください。

TNS-12560 TNS: プロトコル・アダプタ・エラー

TNS-00558 委任ログインに失敗しました。

ORACLE SERVER (*host_name*)

このエラーは、Entrust へのログインを試みているときに、サーバー側の listener.log ファイルで発生する可能性があります。

原因: 次の各変更を実行してクライアントを構成している場合があります。

- .ual ファイルを削除
- Server Login をアンインストール
- クライアントの sqlnet.ora ファイルの SSL_ENTRUST_INI_FILE パラメータに、Entrust 初期化ファイルの場所を指定

この場合、次のコマンドを入力すると、サーバーではクライアントを認証できない可能性があります。

```
sqlplus/@tns_service_name
```

処置: サーバー上で、次のタスクを実行してトレースを使用可能にします。

1. 「コントロール パネル」→「サービス」を選択します。
2. 「サービス」ダイアログ・ボックスで「OracleTNSListener」をダブルクリックし、「ログオン」で「システム アカウント」を現在ログオンしているアカウントに変更します。これによって、サーバー・プロセスが .ual ファイルを読み込めるようになります。「OK」をクリックして変更を反映すると、「サービス」ダイアログ・ボックスに戻ります。

「サービス」ダイアログ・ボックスで、OracleService についても同様の変更を行います。

3. listener.ora ファイルに対して、次の変更を行います。

- リスナーの ADDRESS に、PROTOCOL として TCPS のみを指定します。たとえば、次のようにすべての PROTOCOL 定義を TCPS に変更します。

```
listener_name=
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=TCPS) (KEY=extproc0))
    (ADDRESS=(PROTOCOL=TCPS) (HOST=sales-pc) (PORT=1521)))
```

TCPS を使用している場合のみ、トレースをオンにすると、リスナーは Entrust プロファイルへのアクセスに問題があるかどうかを提示します。

- 次のように、SSL_CLIENT_AUTHENTICATION パラメータを FALSE に設定します。

```
SSL_CLIENT_AUTHENTICATION=FALSE
```

- 次のパラメータを設定して、トレースをオンにします。

```
TRACE_LEVEL_LISTENER=16
TRACE_DIRECTORY_LISTENER=C:¥temp
```

トレース・ファイルは、C:¥temp ディレクトリに作成されます。

4. sqlnet.ora ファイルに対して次の変更を行い、トレースをオンにします。

```
TRACE_LEVEL_SERVER=16
TRACE_DIRECTORY_SERVER=C:¥temp
```

トレース・ファイルは、C:¥temp ディレクトリに作成されます。

5. Entrust/Entelligence がサーバー上にインストールされていないことを確認します。

文字列「fail」または「ntz*」のファンクション・コールを検索します。エラー・メッセージは、これらの文字列の付近にあり、発生した問題の詳細を確認できます。

Entrust を実行するための一般的なチェックリスト（すべてのプラットフォームに共通）

次の各項目は、すべてのプラットフォームに当てはまります。

1. Entrust/Authority がオンラインであることを確認します。
2. .ual ファイルが生成されていることを確認します。これらのファイルは、自動ログイン資格証明のために作成されます。

注意： 自動ログイン資格証明ファイル（.ual）は、サーバーに対してのみ作成することをお薦めします。サーバーのみに .ual ファイルを生成すると、ユーザーがログインを試みる際に、ユーザーのパスワードと Entrust プロファイル名の入力を求めるプロンプトが表示されます。これらの情報を指定すると、接続要求が Entrust サーバーに転送され、取消しファイルおよび .ual ファイルが検索され、要求を許可するための権限が判断されます。

3. Entrust 初期化ファイルの最初のセクション（Entrust Settings）に次のエントリがあることを確認します。

```
IdentityLibrary=location
```

libidapi.so ファイルの位置を示す完全パスを IdentityLibrary パラメータに指定してください。このパラメータ設定によって、サーバーで .ual ファイルを生成できるようになります。

4. Entrust/IPSEC Negotiator Toolkit や Server Login Toolkit など、Entrust のすべてのツールキットが互換性を持つように同じバージョンであることを確認します。
5. 次の例のように、sqlnet.ora ファイルの SQLNET.AUTHENTICATION_SERVICES パラメータに SSL 付き TCP/IP が指定されていることを確認します。

```
SQLNET.AUTHENTICATION_SERVICES=(tcps, authentication_type1, authentication_type2)
```

Windows NT の場合のインストール・チェックリスト

次のチェックリストの項目は、Windows NT プラットフォームでの Entrust インストールに対してのみ適用されます。

1. Entrust/Entelligence にログインしていることを確認します。
2. Windows の「コントロールパネル」→「サービス」を選択し、Entrust Login Interface サービスが起動され、実行中であることを確認します。

3. sqlnet.ora ファイルの SSL_ENTRUST_INI_FILE パラメータに、Entrust 初期化ファイルの位置が指定されていることを確認します。ただし、このパラメータに位置を指定しない場合は、Entrust 初期化ファイルを c:\¥WINNT の位置に配置する必要があります。
4. データベースが Microsoft 社のプラットフォームで実行されている場合は、Entrust/Entelligence が実行されていないことを確認します。この場合は、自動ログインを使用可能にする .ual ファイルのみが必要となります。

関連項目： Entrust binder コマンドを使用して .ual ファイルを作成する方法は、8-12 ページの「[Windows サーバーでの Entrust の構成](#)」の手順 4 を参照してください。

5. Entrust 初期化ファイルで指定されている Entrust/Authority がアクセス可能で、動作していることを確認します。
6. 入力したプロファイルのパスワードが間違っていないことを確認します。
7. Oracle データベース・サーバーから Entrust へのログインが失敗した場合は、自動ログイン資格証明ファイル (.ual) が有効なパスワードを使用して生成されたことを確認します。また、Entrust/Server Login Toolkit と Entrust/IPSEC Negotiator Toolkit のバージョンが一致していることを確認します（つまり、IPSEC Toolkit 6.0 と Server Login Toolkit 6.0 が使用されていれば問題ありません）。
8. Entrust 初期化ファイルの最初のセクション（Entrust Settings）に次のエントリがあることを確認します。

```
IdentityLibrary = location
```

location は libidapi.so の位置（ファイル名を含む）です。

複数の認証方式の構成

この章では、Oracle Advanced Security で複数の認証方式を構成する方法、および別の認証方式を構成している場合に従来のユーザー名とパスワードの認証を使用する方法について説明します。また、Oracle クライアントが特定の認証方式を使用し、Oracle サーバーが任意の認証方式を受け入れられるようにネットワークを構成する方法についても説明します。

次の項目について説明します。

- ユーザー名とパスワードによる接続
- Oracle Advanced Security 認証を使用禁止にする方法
- 複数の認証方式の構成
- 外部認証を使用する場合の Oracle9i の構成

ユーザー名とパスワードによる接続

Oracle Advanced Security の認証方式が構成されている場合にユーザー名とパスワードを使用して Oracle データベース・サーバーに接続するには、外部認証を使用禁止にします（9-2 ページの「[Oracle Advanced Security 認証を使用禁止にする方法](#)」を参照してください）。

外部認証を使用禁止にすると、ユーザーは次のように入力してデータベースに接続できます。

```
% sqlplus username/password@net_service_name
```

たとえば、次のように指定します。

```
% sqlplus scott/tiger@emp
```

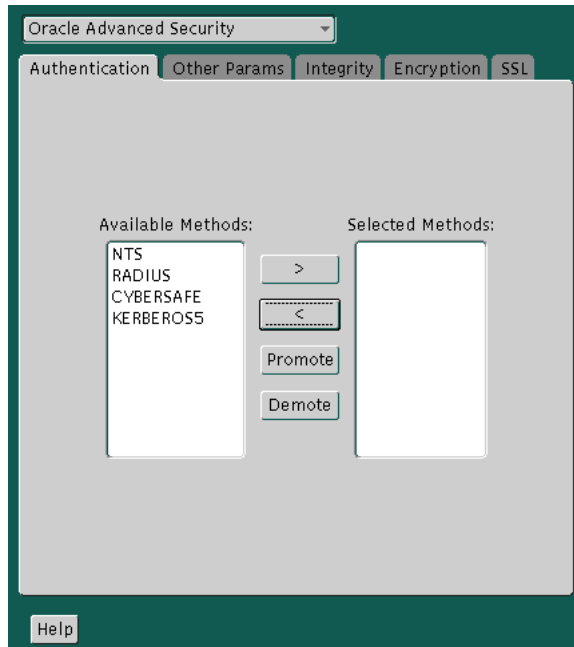
注意： 1つのデータベースで、外部認証ユーザーとパスワード認証ユーザーをどちらも含む複数の認証方式を構成できます。

Oracle Advanced Security 認証を使用禁止にする方法

複数の認証方式を使用する手順は次のとおりです。

1. Oracle Net Manager を起動します。
 - UNIX の場合
\$ORACLE_HOME/bin から netmgr を実行します。
 - Windows NT の場合
「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Network Administration」→「Oracle Net Manager」の順に選択します。
2. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。
3. 右側のウィンドウ・ペインのリストから、「Oracle Advanced Security」を選択します。「Oracle Advanced Security」タブ・ウィンドウが表示されます（[図 9-1](#)）。

図 9-1 Oracle Advanced Security の「Authentication」ウィンドウ



4. 「Authentication」タブを選択します。
5. 認証方式を選択し、左矢印「<」をクリックして、「Selected Methods」リストのすべての認証方式を「Available Methods」リストに順に移動します。
6. 「File」→「Save Network Configuration」を選択します。
`sqlnet.ora` ファイルが更新され、次のエントリが追加されます。
`SQLNET.AUTHENTICATION_SERVICES = (NONE)`

複数の認証方式の構成

多くのネットワークは、1つのセキュリティ・サーバー上で複数の認証方式を使用しています。そのため、Oracle Advanced Security では、Oracle クライアントが特定の認証方式を使用し、Oracle データベース・サーバーが任意の認証方式を受け入れられるようにネットワークを構成できます。

クライアント・システムとサーバー・システムの両方に複数の認証方式を設定するには、Oracle Net Manager を使用するか、テキスト・エディタを使用して `sqlnet.ora` ファイルを変更します。

クライアントとサーバーの両方に認証方式を追加する手順は次のとおりです。

1. Oracle Net Manager を起動します。
 - UNIX の場合
`$ORACLE_HOME/bin` から `netmgr` を実行します。
 - Windows NT の場合
「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Network Administration」→「Oracle Net Manager」の順に選択します。
2. 「Navigator」ウィンドウで、「Local」→「Profile」を展開します。
3. 右側のペインのリストから、「Oracle Advanced Security」を選択します。
「Oracle Advanced Security」タブ・ウィンドウが表示されます (図 9-1)。
4. 「Authentication」タブを選択します。
5. 「Available Methods」リストで方式を選択します。
6. 次に、右矢印「>」をクリックして、選択した方式を「Selected Methods」リストに順に移動します。
7. 選択した方式を使用優先順位の高い順に並べます。「Selected Methods」リストで方式を選択し、「Promote」または「Demote」をクリックして並べ替えます。
8. 「File」→「Save Network Configuration」を選択します。

`sqlnet.ora` ファイル中の次のエントリが更新され、選択した認証方式が左から順にリストされます。

```
SQLNET.AUTHENTICATION_SERVICES = (RADIUS|CYBERSAFE|KERBEROS5)
```

注意：

- SecurID 機能は、RADIUS を介して使用できます。RADIUS のサポートは、RSA ACE/Server に組み込まれています。
 - 第 4 章「RADIUS 認証の構成」を参照してください。
-

外部認証を使用する場合の Oracle9i の構成

この項では、Oracle9i でネットワーク認証を使用するために設定するパラメータについて説明します。関連するタスクは、次のとおりです。

- [sqlnet.ora](#) での `SQLNET.AUTHENTICATION_SERVICES` パラメータの設定
- `REMOTE_OS_AUTHENT` が `TRUE` に設定されていないことを確認
- `OS_AUTHENT_PREFIX` を `NULL` 値に設定

関連項目：

- 特定の認証方式の構成方法は、このマニュアルの対応する章を参照してください。
- 付録 B「認証パラメータ」

`sqlnet.ora` での `SQLNET.AUTHENTICATION_SERVICES` パラメータの設定

すべてのクライアントとサーバーで、それぞれがサポートされている認証方式を使用するには、`sqlnet.ora` ファイルで次のパラメータが設定されている必要があります。

```
SQLNET.AUTHENTICATION_SERVICES=(oracle_authentication_method)
```

たとえば、Kerberos 認証を使用するすべてのクライアントとサーバーに対して、`sqlnet.ora` パラメータが次のように設定されている必要があります。

```
SQLNET.AUTHENTICATION_SERVICES=(KERBEROS5)
```

REMOTE_OS_AUTHENT が TRUE に設定されていないことを確認

REMOTE_OS_AUTHENT が TRUE に設定されていないことを確認するには、認証方式を構成するときに、各データベース・インスタンスで使用する初期化ファイルに次のパラメータを追加します。

```
REMOTE_OS_AUTHENT=FALSE
```

注意： REMOTE_OS_AUTHENT を TRUE に設定すると、非保護プロトコル（TCP など）を使用するユーザーがオペレーティング・システム許可ログイン（以前の OPS\$ ログイン）を実行できるので、セキュリティが侵害されるおそれがあります。

REMOTE_OS_AUTHENT を FALSE に設定した場合に、サーバーがクライアントの要求した認証方式を提供できないと、認証サービスの折衝が失敗して、接続が終了します。

クライアント側またはサーバー側の sqlnet.ora ファイルにパラメータが次のように設定されている場合、データベースは入力されたユーザー名とパスワードを使用して、ユーザーのログインを許可しようとします。

```
SQLNET.AUTHENTICATION_SERVICES= (NONE)
```

しかし、REMOTE_OS_AUTHENT が FALSE に設定されている場合、接続は失敗します。

OS_AUTHENT_PREFIX を NULL 値に設定

認証サービスでは長いユーザー名を使用できますが、Oracle ユーザー名は 30 文字に制限されています。したがって、データベース・インスタンス用の初期化ファイルで、次のように OS_AUTHENT_PREFIX パラメータを NULL 値に設定することをお勧めします。

```
OS_AUTHENT_PREFIX=""
```

注意： OS_AUTHENT_PREFIX のデフォルト値は OPS\$ ですが、このパラメータは任意の文字列に設定できます。

注意： データベースで OS_AUTHENT_PREFIX がすでに NULL ("") 以外の値に設定されている場合は、その値を変更しないでください。変更すると、すでに作成されている外部識別ユーザーが Oracle サーバーに接続できなくなります。

ユーザーを作成するには、SQL*Plus を起動し、次のように入力します。

```
SQL> CREATE USER os_authent_prefix username IDENTIFIED EXTERNALLY;
```

OS_AUTHENT_PREFIX が NULL 値 ("") に設定されているときは、次のように入力してユーザー king を作成します。

```
SQL> CREATE USER king IDENTIFIED EXTERNALLY;
```

この方法でユーザーを作成すると、外部的に識別されるユーザーに対して、異なるユーザー名を管理する必要がありません。これは、サポートされているすべての認証方式に適用されます。

関連項目：

- 『Oracle9i データベース管理者ガイド』
- 『Oracle9i Heterogeneous Connectivity Administrator's Guide』

第 IV 部

Oracle DCE Integration

第 IV 部では、Oracle DCE（分散コンピューティング環境）Integration について説明します。次の章で構成されています。

- 第 10 章「Oracle DCE Integration の概要」
- 第 11 章「Oracle DCE Integration を使用する DCE の構成」
- 第 12 章「Oracle DCE Integration を使用する Oracle9i の構成」
- 第 13 章「DCE 環境の Oracle データベースへの接続」
- 第 14 章「DCE 環境と非 DCE 環境の相互運用性」

注意： オペレーティング・システム固有の Oracle インストレーション・ガイドを確認し、Oracle Advanced Security が使用しているオペレーティング・システムの Oracle DCE Integration をサポートしていることを確認してください。

Oracle DCE Integration の概要

Oracle **DCE** Integration を使用すると、Oracle アプリケーションとツール製品から分散コンピューティング環境（DCE）の Oracle9i サーバーにアクセスできます。この章では、**分散コンピューティング環境**（DCE）と Oracle DCE Integration 製品について簡単に説明します。次の項目について説明します。

- [Oracle DCE Integration の要件](#)
- [分散コンピューティング環境](#)
- [Oracle DCE Integration の構成要素](#)
- [DCE の柔軟な配置方法](#)
- [リリース制限](#)

関連項目： [xxiv ページ「関連文書」](#)

Oracle DCE Integration の要件

システム要件

Oracle DCE Integration を使用するには、Oracle Net Services と Oracle9i が必要です。Oracle DCE Integration は、Open Software Foundation (OSF) の DCE プロトコル (バージョン 1.1 以上) をベースにした製品です。

OSF は別の標準グループである X/OPEN に統合され、オープン・グループを形成しています。このグループが引き続き DCE をサポートします。

下位互換性

DCE Integration 2.3.2 以上のリリースが稼働している Oracle サーバーは、SQL*Net/DCE 2.1.6 または 2.2.3 が稼働しているクライアントと下位互換性があります。ただし、リリース 2.1.6 が稼働しているクライアントは、外部ロールを利用できません。

DCE Integration 2.3.2 以上のリリースを稼働しているクライアントは、SQL*Net/DCE 2.1.6 または 2.2.3 を稼働しているサーバーに接続できません。DCE Integration 2.3.2 以上のリリースが稼働しているクライアントがデータベースに接続するには、2.3.2 以上のリリースが稼働しているサーバーが必要です。

分散コンピューティング環境

オープン・グループの**分散コンピューティング環境** (DCE) は、複数のシステムで機能して分散環境を提供する一連の統合ネットワーク・サービスです。ネットワーク・サービスには、リモート・プロシージャ・コール (RPC)、ディレクトリ・サービス、セキュリティ・サービス、スレッド、分散ファイル・サービス、ディスクレス・サポート、分散タイム・サービスなどがあります。

DCE は分散アプリケーションとオペレーティング・システム / ネットワーク・サービスの間のミドルウェアで、コンピューティングのクライアント / サーバー・モデルに基づいています。ユーザーは DCE が提供するサービスとツールを使用して、異機種環境で動作する分散アプリケーションを作成、使用および管理できます。

Oracle DCE Integration の構成要素

Oracle の DCE Integration には、DCE 通信 / セキュリティと DCE CDS Native Naming の 2 つの構成要素があります。

- [DCE 通信 / セキュリティ](#)
- [DCE Cell ディレクトリ・サービス Native Naming](#)

DCE 通信 / セキュリティ

この構成要素には、次の 3 つの重要な機能があります。

認証された RPC

Oracle DCE Integration は、複数ベンダー間での相互運用性を実現するトランスポート・メカニズムとして、認証されたリモート・プロシージャ・コール (RPC) を採用しています。RPC は他の DCE サービス (ディレクトリ・サービスやセキュリティ・サービスなど) を使用して、位置の透過性と安全な分散コンピューティングも実現します。

統合セキュリティとシングル・サインオン

Oracle DCE Integration は DCE セキュリティ・サービスと連携して、DCE セル内のセキュリティを確立します。このため、DCE にログインしたユーザーは、ユーザー名またはパスワードを指定しなくても、任意の Oracle データベースに安全にアクセスできます。これは、データベースに対する[外部認証](#)または[シングル・サインオン](#)と呼ばれる場合があります。DCE 認証サービスを使用しないクライアントとサーバーは、Oracle パスワードを指定することによって、DCE セキュリティが確立されているシステムにアクセスできます。

データ・プライバシーと整合性

Oracle DCE Integration は、DCE が提供する複数のレベルのセキュリティを使用して、データの確実性、プライバシーおよび整合性を保証しています。ユーザーは接続ごとに「保護なし」から「完全な暗号化」までの様々なレベルを選択して、転送中のデータが変更されないようにすることができます。

注意： DCE を使用しないネットワークの部分については、Oracle Advanced Security を構成する他のセキュリティおよび認証サービスを使用できます。これらのサービスは、SQL*Net リリース 2.1 以上または Oracle Net Services に対応しています。これらのサービスを使用すると、非 DCE 環境でメッセージ整合性サービスとデータ暗号化サービスを利用できるので、ネットワーク接続のスタートポイントまたはエンドポイントに関係なく、すべてのネットワーク通信が許可なしに表示または変更される脅威から防御できます。

DCE Cell ディレクトリ・サービス Native Naming

DCE [Cell ディレクトリ・サービス](#) Native Naming コンポーネントには命名機能と位置透過性の機能があります。

DCE Integration では、Oracle9i の接続記述子を DCE CDS に登録することにより、DCE 環境内のどこからでも接続記述子に透過的にアクセスできます。ユーザーは使い慣れた Oracle サービス名で、DCE 環境内の Oracle データベース・サーバーに接続できます。

DCE Cell ディレクトリ・サービスは、ネットワークに存在するオブジェクトの名前、アドレスおよび属性を登録する分散型の複製リポジトリ・サービスです。サーバーの名前とアドレス情報が CDS に登録されているので、Oracle クライアントは位置に依存せずに Oracle9i サーバーに接続できます。クライアントの構成に変更を加えずに、サービスを再配置できます。Oracle ユーティリティを使用して、Oracle サービス名および対応する接続記述子を CDS にロードします。Oracle サービス名を CDS にロードした後、標準 DCE ツールを使用して集中管理された Oracle 接続識別子を表示できます。

サービスの位置が複数のセルにまたがっている場合は、次のサービスを使用できます。

- DCE [グローバル・ディレクトリ・サービス](#)
- インターネット・ドメイン・ネーム・サービス (DNS)

関連項目：

- CDS ネーミングを使用する DCE の構成については、[第 11 章「Oracle DCE Integration を使用する DCE の構成」](#)を参照してください。
- CDS を使用する Oracle クライアントと Oracle サーバーの構成については、[第 12 章「Oracle DCE Integration を使用する Oracle9i の構成」](#)を参照してください。
- Oracle Native Naming と他の Oracle ネーム・サービスの連携については、『Oracle9i Net Services 管理者ガイド』を参照してください。

DCE の柔軟な配置方法

Oracle Advanced Security では、DCE のサービス使用方法を柔軟に選択できます。次のオプションを選択できます。

- DCE Integration 全体を環境に配置して、すべての DCE Secure Core サービス（RPC、ディレクトリ、セキュリティ、スレッド）と統合できます。
- DCE CDS Native Naming アダプタと、TCP/IP のような従来型のプロトコル・アダプタを使用すると、DCE のディレクトリ・サービスのみを使用できます。

リリース制限

Oracle Advanced Security リリース 2 (9.2) には次の制限があります。

- DCE プロトコルを使用するリスナー・アドレスは、各ノードで 1 つしか認められていません。
- データベース・リンクで接続するには、ユーザー名とパスワードを指定する必要があります。
- このリリースの DCE Integration は、Oracle Multi-Protocol Interchange をサポートしていません。
- このリリースは、Oracle 共有サーバーで動作しません。

Oracle DCE Integration を使用する DCE の構成

この章では、Oracle DCE Integration をインストールした後で、DCE Integration を使用できるように分散コンピューティング環境（DCE）を構成する方法について説明します。

関連項目： [第 10 章「Oracle DCE Integration の概要」](#)

Oracle DCE Integration を使用する DCE の構成

DCE セル管理者が実行する次の作業では、DCE セルが構成済みで、システムがそのセルの一部であると想定しています。

- [タスク 1: 新規のプリンシパルとアカウントの作成](#)
- [タスク 2: サーバーのキーをキータブ・ファイルにインストール](#)
- [タスク 3: Oracle DCE Integration で使用する DCE CDS を構成](#)

タスク 1: 新規のプリンシパルとアカウントの作成

次のプロシージャ・モデルを使用して、サーバー・プリンシパルを追加します。

```
% dce_login cell_admin password
% rgy_edit
Current site is: registry server at ../../cell1/subsys/dce/sec/master
rgy_edit=>do p
Domain changed to: principal
rgy_edit=> add oracle
rgy_edit=> do a
Domain changed to: account
rgy_edit=> add oracle -g none -o none -pw oracle_password -mp cell_admin_password
rgy_edit=> quit
bye
```

これで、oracle という DCE プリンシパルが作成されました。このプリンシパルには、oracle_password というパスワードを持つアカウントが対応付けられています。このアカウントはどの DCE グループや DCE プロファイルにも属していません。

注意： DCE Integration のインストール後、この作業をサーバーに対して一度のみ実行します。クライアントに対しては実行しません。

タスク 2: サーバーのキーをキータブ・ファイルにインストール

サーバーのキーをキータブ・ファイル dcepa.key にインストールします。このファイルにはプリンシパルのパスワードが入っており、Oracle Net リスナーはこのプリンシパルのもとで起動します。Oracle Net リスナーはこのファイルを読み取り、DCE に対してリスナー自身を認証します。キータブ・ファイルを生成するには、次のように入力します。

```
% dce_login cell_admin password
% rgy_edit
Current site is: registry server at ../../cell1/subsys/dce/sec/master
rgy_edit=> ktadd -p oracle -pw Oracle_password -f
$ORACLE_HOME/dcepa/admin/dcepa.key
rgy_edit=>quit
bye
```

注意:

- DCE Integration のインストール後、この作業をサーバーに対して一度のみ実行します。クライアントに対しては実行しないでください。
- ORACLE_HOME 変数を完全パス名で置き換えてください。存在していないディレクトリを指定する場合は、コマンドを実行する前にそのディレクトリを作成する必要があります。ディレクトリを作成するには、次のように入力します。

```
mkdir $ORACLE_HOME/dcepa
mkdir $ORACLE_HOME/dcepa/admin
```

タスク 3: Oracle DCE Integration で使用する DCE CDS を構成

手順 1: CDS ネームスペースに Oracle ディレクトリを作成する

DCE Integration をセルに初めてインストールした後、次のように入力して、Oracle ディレクトリをすべての CDS レプリカ上で作成します。

```
% dce_login cell_admin

Enter Password:(password not displayed)
$ cdscp
cdscp> create dir ./subsys/oracle
cdscp> create dir ./subsys/oracle/names
cdscp> create dir ./subsys/oracle/service_registry
cdscp> exit
```

注意：

- ディレクトリ `././subsys/oracle/names` には、Oracle Net サービス名を接続記述子にマップするオブジェクトがあります。この接続記述子は、CDS ネーミング・アダプタによって使用されます。
 - ディレクトリ `././subsys/oracle/service_registry` にも、DCE アドレス内のサービス名をネットワーク・エンドポイントにマップするオブジェクトが入っています。DCE プロトコル・アダプタのクライアントとサーバーが、このネットワーク・エンドポイントを使用します。
-
-

手順 2: CDS ネームスペースでのオブジェクト作成権限をサーバーに付与する

次のように入力して、oracle プリンシパルを CDS-server グループに追加します。

```
$ dce_login cell_admin
Enter Password: (password not displayed)
$ rgy_edit
rgy_edit=> domain group
Domain changed to: group
rgy_edit=> member subsys/dce/cds-server -a oracle
rgy_edit=> exit
```

手順 3: Oracle サービス名を CDS にロードする

第 12 章「[Oracle DCE Integration を使用する Oracle9i の構成](#)」で説明しているように、Oracle サービス名を Cell ディレクトリ・サービスにロードします。

Oracle DCE Integration を使用する Oracle9i の構成

この章では、Oracle DCE Integration を正常にインストールした後で、DCE Integration を使用できるように Oracle9i と Oracle Net Services を構成する方法について説明します。

次の項目について説明します。

- [DCE アドレス・パラメータ](#)
- [Oracle9i と Oracle Net Services の構成](#)

DCE アドレス・パラメータ

listener.ora 構成ファイルと tnsnames.ora 構成ファイルの DCE アドレスは、次に示すような DCE パラメータによって定義されます。

```
ADDRESS=(PROTOCOL=DCE) (SERVER_PRINCIPAL=server_name) (CELL_NAME=cell_name)
(SERVICE=dce_service_name))
```

これらのパラメータについて表 12-1 で説明します。

表 12-1 DCE アドレス・パラメータと定義

構成要素	説明
PROTOCOL	DCE RPC プロトコルを識別する必須フィールドです。
SERVER_PRINCIPAL	サーバーの必須フィールドで、クライアントのオプション・フィールドです。サーバーはこのプリンシパルでサーバー自身を DCE に対して認証します。このフィールドは、リスナー構成ファイル (listener.ora) の必須フィールドで、サーバーが起動する際のプリンシパルを指定します。このフィールドは、ローカル・ネーミング構成ファイル (tnsnames.ora) のオプション・フィールドで、クライアントが接続する必要があるサーバーのプリンシパルを指定します。このフィールドを指定しないと、1 方向の認証が使用されます。この場合、クライアントはサーバーのプリンシパルを意に介しません。
CELL_NAME	オプション・パラメータです。このパラメータを設定して、データベースの DCE セル名を指定します。このパラメータを設定しないと、セル名はデフォルトのローカル・セルになります（これは、単一セル環境で役立ちます）。また、次の項で説明する SERVICE パラメータでサービスの完全パス（セル名を含むパス）を指定すれば、CELL_NAME パラメータを設定する必要はありません。
SERVICE	サーバーとクライアントの両方での必須フィールドです。サーバーでは、CDS に登録されているサービスを指定します。クライアントでは、CDS に Oracle DCE サーバーの場所を問い合わせるときに使用するサービス名を指定します。サービス名を CDS に格納するためのデフォルト・ディレクトリは、 /.../cellname/subsys/oracle /service_registry です。このサービス名で、CDS 内の完全パスを指定できます。

サービスを次のように指定します。

```
SERVICE=../../cell_name/subsys/oracle/service_registry/dce_service_name
```

また、次のように指定することもできます。

```
SERVICE=dce_service_name
```

これは、CELL_NAME=cell_name も指定されている場合です。

この場合、セル名はローカル・セルにデフォルト設定されています。ただし、この方法でサービス名を指定できるのは、1つのセル内で操作をしているときのみです。

注意： SERVICE フィールドで指定する `dce_service_name` は、Oracle Net Services で使用されているサービス名と同じでなくてもかまいません。Oracle Net で使用されているサービス名は、ローカル・ネーミング構成ファイル (`tnsnames.ora`) 内の接続記述子にマップされます。`dce_service_name` は、接続記述子内のアドレス部です。

Oracle9i と Oracle Net Services の構成

Oracle DCE Integration を使用できるように Oracle9i と Oracle Net Services を構成するには、次のタスクを実行します。

- [タスク 1: サーバーの構成](#)
- [タスク 2: 外部的に認証されるアカウントの作成と命名](#)
- [タスク 3: DCE Integration の外部ロールの設定](#)
- [タスク 4: SYSDBA および SYSOPER で Oracle データベースに接続するための DCE の構成](#)
- [タスク 5: クライアントの構成](#)
- [タスク 6: DCE CDS ネーミングを使用するクライアントの構成](#)

タスク 1: サーバーの構成

次の手順に従って、DCE Integration を使用するサーバーを構成します。

1. リスナー構成ファイル (`listener.ora`) で、すべてのサーバーに対する DCE のアドレス情報を設定します。
2. 他のサーバーへのデータベース・リンク接続が必要な分散環境内のサーバーに対して、`sqlnet.ora` および `protocol.ora` ファイルで DCE アドレス情報を設定する必要があります。

注意： このリリースでは、listener.ora、sqlnet.ora、tnsnames.ora、protocol.ora の各構成ファイルは、\$ORACLE_HOME/network/admin ディレクトリにあります。

DCE 環境でデータベース・サーバーが Oracle Net クライアントからの接続を受け入れるには、サーバー・プラットフォーム上で Oracle Net リスナーがアクティブになっている必要があります。このプロセスによって、listener.ora で定義されているネットワーク・アドレス上で接続をリスニングします。

SERVER_PRINCIPAL パラメータは、リスナーが動作する DCE プリンシパルを指定します。次のサンプルでは、リスナーが oracle プリンシパルで動作しています。

次に示すのは、listener.ora ファイルで設定する DCE アドレスのサンプルです。

```
LSNR_DCE=
  (ADDRESS=
    (PROTOCOL=DCE)
    (SERVER_PRINCIPAL=oracle)
    (CELL_NAME=cell1)
    (SERVICE=dce_svc))
SID_LIST_LISTENER_DCE=
  (SID_DESC=
    (SID_NAME=ORASID)
    (ORACLE_HOME=/private/oracle9))
```

タスク 2: 外部的に認証されるアカウントの作成と命名

DCE 認証を使用して Oracle データベースにログオンするには、外部的に認証されるデータベース・アカウントを作成する必要があります。次の手順に従って、安全性の高い外部認証を使用可能にします。

注意： この項で示す権限は、最低限必要なアクセス権限です。実際に必要な権限は、インスタンスまたはアプリケーションによって異なります。

1. 初期化パラメータ・ファイルに次の行があるかどうか確認します。

```
REMOTE_OS_AUTHENT=FALSE
OS_AUTHENT_PREFIX=""
```

2. DCE に対するマルチスレッド・サーバー (MTS) のエントリが、初期化パラメータ・ファイルに存在しないことを確認します。たとえば、次のようなエントリを指定することはできません。

```
mts_dispatchers="(PROTOCOL=dce) (DISPATCHERS=3) "
```

3. DBA グループのメンバーとしてログインしていることを確認します。データベース・インスタンスを再起動して、変更内容を有効にします。
4. SQL*Plus プロンプトでユーザーを定義します。その前に、すでにマルチセル DCE 環境でデータベースを運用しているのか、または今後そのように運用していくのかについて判断します。マルチセル DCE 環境では、セルの境界を越えて Oracle データベースにアクセスできます。ユーザーの定義方法は、ユーザーが 1 つのセル内でデータベースに接続するか、セルの境界を越えて接続するかによって異なります。

ローカル・セル:

ユーザーがローカル・セル内で接続する場合は、次の書式を使用します。

```
SQL> CREATE USER server_principal IDENTIFIED EXTERNALLY;
SQL> GRANT CREATE SESSION TO server_principal;
```

例:

```
SQL> CREATE USER oracle IDENTIFIED EXTERNALLY;
SQL> GRANT CREATE SESSION TO oracle;
```

CELL_NAME/SERVER_PRINCIPAL 文字列の長さは、全体で 30 文字以内に収める必要があります（これは DCE アダプタの制限ではなく、Oracle9i の制限です）。

例:

```
SQL> CREATE USER "CELL1/ORACLE" IDENTIFIED EXTERNALLY;
SQL> GRANT CREATE SESSION TO "CELL1/ORACLE";
```

複数のセル:

複数のセルにまたがってデータベースに接続する場合は、次に示すように、*cell_name* と *server_principal* の両方を指定します。

```
SQL> CREATE USER "CELL_NAME/SERVER_PRINCIPAL" IDENTIFIED EXTERNALLY;
SQL> GRANT CREATE SESSION TO "CELL_NAME/SERVER_PRINCIPAL";
```

スラッシュは予約文字なので、外部的に識別されるアカウントの名前を二重引用符で囲む必要があります。また、アカウント（ユーザー）名を二重引用符で囲む場合は、アカウント（ユーザー）名を大文字で入力する必要があります。

例:

```
SQL> CREATE USER "CELL1/ORACLE" IDENTIFIED EXTERNALLY;
SQL> GRANT CREATE SESSION TO "CELL1/ORACLE";
```

この書式を使用するときは、`protocol.ora` 構成ファイルで次のパラメータを FALSE に設定します。

```
dce.local_cell_usernames=false
```

この方法で作成した Oracle アカウントを参照するときは、スキーマ / アカウントを適切な書式で指定する必要があります。別のアカウントの表にアクセスを要求する場合について考えてみます。ローカル・セル内で作成した別のアカウントの表を参照するときは、次のようなコマンドを使用します。

```
SQL> SELECT * FROM oracle.emp
```

複数のセルにまたがる接続用に作成した別のアカウントの表にアクセスするときは、次のようなコマンドを使用します。

```
SQL> SELECT * FROM "CELL1/ORACLE" .emp
```

関連項目： 外部認証の詳細は、『Oracle9i Heterogeneous Connectivity Administrator’s Guide』を参照してください。

タスク 3: DCE Integration の外部ロールの設定

次の手順に従って、DCE Integration に外部ロールを設定し、DCE 資格証明を使用して SYSOPER または SYSDBA で Oracle データベースに接続します。

- 1. 次のパラメータを初期化パラメータ・ファイルに設定します。
`OS_ROLES=TRUE`
- 2. データベースを再起動します。
- 3. Oracle ロールにマップする DCE グループが、次の構文になっているかどうかを確認します。

```
ORA_global_name_role[_[a][d]]
```

表 12-2 は、この構文の構成要素を示しています。

表 12-2 外部ロール構文の構成要素の設定

構成要素	定義
ORA	このグループを Oracle 用に使用することを指定します。
GLOBAL_NAME	データベースのグローバル名。
ROLE	データ・ディクショナリで定義されているロール名。
A または a	ユーザーがこのロールの admin 権限を持つことを示すオプション文字。
D または d	接続時にデフォルトでロールを使用可能にすることを示すオプション文字。

関連項目： 外部ロールの詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

4. 次のコマンドを実行して、DCE グループのメンバーである DCE ユーザーを DCE に対して認証します。

```
dce_login
klist
```

サンプル出力は次のとおりです。

```
% dce_login oracle
```

Enter Password:

```
% klist
dce identity information:
Warning: Identity information is not certified
Global Principal: ../../ilab1/oracle
Cell:          001c3f90-01f5-1f72-ba65-02608c2c84f3 ../../ilab1
Principal: 00000068-0568-2f72-bd00-02608c2c84f3 oracle
Group:      0000000c-01f5-2f72-ba01-02608c2c84f3 none
Local Groups:
0000000c-01f5-2f72-ba01-02608c2c84f3 none
0000006a-0204-2f72-b901-02608c2c84f3 subsys/dce/cds-server
00000078-daf4-2fe1-a201-02608c2c84f3 ora_dce222_dba
00000084-89c8-2fe8-a201-02608c2c84f3 ora_dce222_connect_d
00000087-8a13-2fe8-a201-02608c2c84f3 ora_dce222_resource_d
00000080-f681-2fe1-a201-02608c2c84f3 ora_dce222_role1_ad
.
.
.
```

5. 通常の方法でデータベースに接続します。

次のサンプル出力は、DCE グループにマップされた外部ロール (DBA、CONNECT、RESOURCE、ROLE1) のリストを示しています。

```
SQL> SELECT * FROM session_roles;
```

```
ROLE
-----
CONNECT
RESOURCE
ROLE1
```

```
SQL> SET ROLE all;
```

```
Role set.
```

```
SQL> SELECT * FROM session_roles;
```

```
ROLE
-----
DBA
EXP_FULL_DATABASE
IMP_FULL_DATABASE
CONNECT
RESOURCE
ROLE1
```

```
6 rows selected.
```

```
SQL> EXIT
```

タスク 4: SYSDBA および SYSOPER で Oracle データベースに接続するための DCE の構成

次の手順に従って、DCE 資格証明を使用して SYSOPER または SYSDBA で Oracle データベースに接続できるように DCE を構成します。

1. Oracle DBA ロールと OPERATOR ロールにマップする DCE グループを作成します。DCE グループ名は、12-6 ページの「[タスク 3: DCE Integration の外部ロールの設定](#)」で説明している構文に準拠している必要があります。外部的に認証されるユーザー `oracle` をグループのメンバーとして追加します。

```
$ dce_login cell_admin cell_admin_password
$ rgy_edit
rgy_edit=> domain group
Domain changed to: group
rgy_edit=> add ora_dce222_dba_ad
rgy_edit=> add ora_dce222_operator_ad
rgy_edit=> member ora_dce222_dba_ad -a oracle
rgy_edit=> member ora_dce222_operator_ad -a oracle
```

2. GLOBAL_NAME パラメータを DCE アドレスに追加するか、TNS サービス名をローカル構成ファイル `tnsnames.ora` に追加します。

```
ORADCE=
  (ADDRESS=
    (PROTOCOL=DCE)
    (SERVER_PRINCIPAL=oracle)
    (CELL_NAME=cell1)
    (SERVICE=dce_svc))
  (CONNECT_DATA=
    (SID=ORASID)
    (GLOBAL_NAME=dce222)))
```

3. 12-4 ページの「[タスク 2: 外部的に認証されるアカウントの作成と命名](#)」で説明している方法で、データベース・ユーザー `oracle` を作成します。
4. 外部的に認証されるユーザーの DCE 資格証明を取得します。

```
$ dce_login oracle oracle_password
$ klist
DCE Identity Information:
Warning: Identity information is not certified
Global Principal: ../../dce.dlsun685.us.oracle.com/oracle
Cell:           00af8052-7e94-11d2-b261-9019b88baa77
../../dce.dlsun685.us.oracle.com
Principal: 0000006d-88b9-21d2-9300-9019b88baa77 oracle
Group:     0000000c-7e94-21d2-b201-9019b88baa77 none
```

```
Local Groups:
    0000000c-7e94-21d2-b201-9019b88baa77 none
    0000006a-7e94-21d2-ad01-9019b88baa77 subsys/dce/cds-server
    00000076-8b53-21d2-9301-9019b88baa77 ora_dce222_dba_ad
    00000077-8b53-21d2-9301-9019b88baa77 ora_dce222_operator_ad

Identity Info Expires: 1999-12-04-10:28:22
Account Expires:      never
Passwd Expires:      never

Kerberos Ticket Information:
Ticket cache: /opt/dcelocal/var/security/creds/dcecred_43ae2600
Default principal: oracle@dce.dlsun685.us.oracle.com
Server: krbtgt/dce.dlsun685.us.oracle.com@dce.dlsun685.us.oracle.com
        valid 1999-12-04-00:28:22 to 1999-12-04-10:28:22
Server: dce-rgy@dce.dlsun685.us.oracle.com
        valid 1999-12-04-00:28:22 to 1999-12-04-10:28:22
Server: dce-ptgt@dce.dlsun685.us.oracle.com
        valid 1999-12-04-00:28:26 to 1999-12-04-02:28:26
Client: dce-ptgt@dce.dlsun685.us.oracle.com      Server:
krbtgt/dce.dlsun685.us.o
racle.com@dce.dlsun685.us.oracle.com
        valid 1999-12-04-00:28:26 to 1999-12-04-02:28:26
Client: dce-ptgt@dce.dlsun685.us.oracle.com      Server:
dce-rgy@dce.dlsun685.us.
oracle.com
        valid 1999-12-04-00:28:27 to 1999-12-04-02:28:26
```

注意： リスト出力は、Oracle の DCE グループのメンバーシップです。

5. Oracle データベースに SYSDBA または SYSOPER として接続します。

例：

```
SQL> connect /@oradce as SYSDBA
```


タスク 5: クライアントの構成

DCE Integration を使用するクライアントを構成するには、次の Oracle Net ファイルに DCE のアドレス情報とパラメータ情報を構成する必要があります。

- protocol.ora
- sqlnet.ora

通常は、CDS を使用して名前を解決します。したがって、名前とアドレスを CDS にロードする場合を除いて、ローカル・ネーミング構成ファイル (tnsnames.ora) は使用しません。

protocol.ora のパラメータ

4 つの DCE パラメータが protocol.ora ファイルにあります。これらのパラメータは、他のプロトコルに関連するパラメータと区別するために DCE. で始まります。これら 4 つのパラメータでデフォルト値を使用する場合、DCE Integration では、protocol.ora ファイルは必要ありません。パラメータと現行のデフォルトは、次のとおりです。

- DCE.AUTHENTICATION=*dce_secret*
- DCE.PROTECTION=*pkt_integ*
- DCE.TNS_ADDRESS_OID=1.3.22.1.5.1
- DCE.LOCAL_CELL_USERNAMES=TRUE

構成パラメータに大 / 小文字の区別はありません。構成パラメータを入力する際は、大文字でも小文字でもかまいません。

DCE.AUTHENTICATION

DCE.AUTHENTICATION パラメータはオプションです。このパラメータは、それぞれの DCE RPC で使用する認証値を指定します。クライアント側の DCE_AUTHENTICATION の値は、サーバー側の DCE_AUTHENTICATION の値と同じであることが必要です。エントリを指定しないと、デフォルトのセル内認証レベルが使用されます。オプションは次のとおりです。

オプション	説明
NONE	認証はありません。
DCE_SECRET	DCE 共用秘密鍵認証 (Kerberos)。
DCE_SECRET	デフォルトの認証レベルと推奨値。
DEFAULT	セルのデフォルト。

DCE.PROTECTION

DCE.PROTECTION は、データ送信のためのデータの整合性保護レベルを指定するオプション・フィールドです。クライアント側で指定する DCE_PROTECTION のレベルは、サーバー側で指定する DCE_PROTECTION のレベル以上であることが必要です。エントリを指定しないと、デフォルトのセル内保護レベルが使用されます。オプションは次のとおりです。

オプション	説明
NONE	現行の接続でデータの整合性を保護しません。
DEFAULT	デフォルトのセル内保護レベルを使用します。
CONNECT	クライアントがサーバーと関係を確認するときのみ、データの整合性を保護します。
CALL	サーバーが要求を受け取る各リモート・プロシージャ・コールの最初のみ、データの整合性を保護します。
PKT	すべてのデータが所定のクライアントから受け取られることを保証します。
PKT_INTEG	クライアントとサーバーの間で転送されるデータが変更されていないことを保証します。
PRIVACY	前述のすべてのレベルで指定した保護を実行し、それぞれのリモート・プロシージャ・コールの引数値とすべてのユーザー・データを暗号化します。

DCE.TNS_ADDRESS_OID

DCE.TNS_ADDRESS_OID はオプション・パラメータです。このパラメータによって次のようにデフォルト値にかわるものを指定できます。

DCE.TNS_ADDRESS_OID=1.3.22.1.x.x

関連項目： 12-14 ページ「[手順 2: CDS 属性ファイルの変更と CDS の再起動](#)」

DCE.LOCAL_CELL_USERNAMES

DCE.LOCAL_CELL_USERNAMES はオプション・パラメータです。このパラメータはプリンシパル名 (username) を、セル名とともにまたはセル名なしで指定するときに使用する書式を定義します。このパラメータに対して指定する値は、ユーザーが一意の名前を使用して、複数のセルにまたがって接続するかどうかによって異なります。DCE.LOCAL_CELL_USERNAMES のデフォルトは TRUE です (DCE Integration 2.1.6 では FALSE に設定されていました)。

対応するオプションは次のとおりです。

オプション	説明
TRUE	デフォルト値。CELL_NAME を指定せずに SERVER_PRINCIPAL 書式のみを使用する場合は、TRUE を選択します。(デフォルト) たとえば、この書式を使用して次のようにユーザーを指定します。 oracle TRUE を選択するのは、単一のセル内で接続を行う場合、またはネットワーク内のネーミング規則によって、セルの異なるユーザーの名前が重複しないように規定されている場合です。
FALSE	CELLNAME/SERVER_PRINCIPAL 書式を使用するときは、FALSE を選択します。たとえば、この書式を使用して次のようにユーザーを指定します。 CELL1/ORACLE FALSE を選択するのは、セル間で接続を行う場合、または別のセルに同一の名前のユーザーが存在している場合です。

タスク 6: DCE CDS ネーミングを使用するクライアントの構成

通常、クライアントは **Cell ディレクトリ・サービス** (CDS) を使用して Oracle サービス名をアドレスに解決します。次の手順を実行して CDS を構成します。

- 手順 1: 名前参照で CDS を使用
- 手順 2: CDS 属性ファイルの変更と CDS の再起動
- 手順 3: Oracle 接続記述子を CDS にロードするのに必要な tnsnames.ora ファイルの作成
- 手順 4: Oracle 接続記述子の CDS へのロード
- 手順 5: tnsnames.ora ファイルの削除または改名
- 手順 6: CDS で名前を解決するために sqlnet.ora ファイルを変更

注意： この作業を完了すると、DCE 環境の Oracle データベースに接続できます。

手順 1: 名前参照で CDS を使用

名前を解決するときに CDS を使用するには、CDS を使用するすべてのクライアントとサーバーに DCE Integration CDS ネーミング・アダプタをインストールする必要があります。また、DCE Integration が使用する CDS ネームスペースを構成しておく必要があります。

関連項目： DCE Integration のインストールの説明および 11-3 ページの「[タスク 3: Oracle DCE Integration で使用する DCE CDS を構成](#)」を参照してください。

たとえば、ORADCE などのサービス名とそのネットワーク・アドレスを、DCE CDS に格納できます。

次の例で示すようにドメインがない場合、またはデータベースがユーザーのデフォルト・ドメインにある場合、通常、ユーザーは使い慣れた Oracle サービス名で Oracle サービスに接続できます。

```
sqlplus /@ORADCE
```

この例では、DCE の外部認証アカウントを使用していることを想定しています。

CDS にアクセスできないときは、別の名前解決サービスとしてローカル・ネーミング構成ファイル `tnsnames.ora` を使用します。このファイルを使用するには、すべての Oracle サーバーの名前とアドレスをローカル・ネーミング構成ファイル `tnsnames.ora` で指定する必要があります。

手順 2: CDS 属性ファイルの変更と CDS の再起動

CDS ネーミングを使用するすべての DCE マシン上で、CDS 属性 `TNS_Address` のオブジェクト ID を CDS 属性ファイルに追加します（オブジェクト ID は、すべてのマシンで同じにする必要があります）。

1. `/opt/dcelocal/etc/cds_attributes` ファイルに、次に示す書式の行を追加します。

```
1.3.22.1.5.1    TNS_Address    char
```

DCE ネーミング規則では、この `TNS_Address` 属性値 `1.3.22.1.x.y` の最初の 4 桁が固定されています。`TNS_Address` のデフォルトのオブジェクト ID の値 `1.3.22.1.5.1` がすでに `cds_attributes` ファイルに存在している場合は、使用中でないオブジェクト ID の値を指定する必要があります。

オブジェクト ID のデフォルト値を使用できない場合は、クライアント上の `protocol.ora` ファイルでオブジェクト ID を指定する必要があります。

デフォルト値 `1.3.22.1.5.1` 以外の値を指定する必要がある場合は、次のパラメータを `protocol.ora` ファイルに追加してください。

```
DCE.TNS_ADDRESS_OID=1.3.22.1.x.y
```

cds_attributes ファイルで指定するオブジェクト ID の値は、protocol.ora ファイルの DCE.TNS_ADDRESS_OID パラメータで指定した値と一致している必要があります。

2. システム上で CDS を再起動します

CDS を再起動するためのコマンドは、オペレーティング・システムによって異なります。たとえば、Solaris プラットフォームでは、次のコマンドを使用して CDS を再起動します。

```
/opt/dcelocal/etc/rc.dce restart
```

手順 3: Oracle 接続記述子を CDS にロードするのに必要な tnsnames.ora ファイルの作成

Oracle サービス名とアドレスを CDS にロードするために、ローカル・ネーミング構成ファイル tnsnames.ora を作成または変更します。このファイルを使用して、サービス名を Oracle Net で使用するアドレスにマップします。

この項では、tnsnames.ora ファイルに指定する必要があるパラメータについて説明します。このファイルには、ネットワーク内の宛先またはエンドポイントを示す接続記述子にマップされている Oracle サービス名のリストが含まれます。次の項の例にある DCE アドレスは、Oracle サービス名が ORADCE の Oracle サーバーのネットワーク・アドレスを示しています。この DCE アドレスを使用して、CDS ディレクトリの、DCE-SVC と登録されているサービスに接続します。

```

/.../cell_name/subsys/oracle/names.
ORADCE=(DESCRIPTION=(ADDRESS=(PROTOCOL=DCE) (SERVER_PRINCIPAL=oracle) (CELL_
NAME=cell11) (SERVICE=DCE_SVC)) (CONNECT_DATA=(SID=ORASID)))
```

注意： この例では、Oracle サービス名と DCE サービス名が異なりますが、同じサービス名を使用する場合もよくあります。

パラメータ			
名前	タイプ	必須かどうか	説明
PROTOCOL=DCE	キーワード 値ペア	必須	このキーワード値は、リスナー構成ファイル listener.ora のアドレス・セクションと、ローカル・ネーミング構成ファイル tnsnames.ora のアドレス・セクションにあります。
SERVER_PRINCIPAL	DCE パラ メータ	オプション	このパラメータは、tnsnames.ora にあります。

パラメータ			
名前	タイプ	必須かどうか	説明
SERVICE	DCE パラメータ	必須	DCE パラメータの値 (SERVICE=dce_service_name) は、listener.ora と tnsnames.ora で同じにする必要があります。
SID	Oracle パラメータ	必須	このパラメータで Oracle システム ID を指定します。SID の値はノード上で一意である必要があります。SID はローカルのみで使用し、DCE CDS では使用しません。

関連項目： ローカル・ネーミング構成ファイル tnsnames.ora の詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。

手順 4: Oracle 接続記述子の CDS へのロード

Oracle DCE Integration には、接続記述子を CDS にロードするためのユーティリティ tnnfg があります。tnsnames.ora ファイルで新しいサービス名とアドレスを構成すると、tnnfg が新しいサービス名とアドレスを CDS に追加します。特定のサービス名に対するアドレスを変更すると、tnnfg が特定のサービス名に対するアドレスを更新します。

Oracle サービス名または別名を tnsnames.ora から CDS にロードするには、コマンド・プロンプトで次のように入力します。

```
% dce_login cell_admin
% tnnfg dceload full_pathname_to_tnsnames.ora
% Enter Password:(password will not display)
```

tnsnames.ora ファイルの完全パス名を入力してください。また、sqlnet.ora ファイルが tnsnames.ora ファイルと同じディレクトリにあることも確認してください。

手順 5: tnsnames.ora ファイルの削除または改名

CDS が使用不能になった場合のバックアップとして tnsnames.ora ファイルを残しておくことができます。tnsnames.ora ではなく CDS が正しく検索されるように、「[手順 6: CDS で名前を解決するために sqlnet.ora ファイルを変更](#)」(次の項)の説明に従って、プロファイル (sqlnet.ora) の NAMES.DIRECTORY_PATH パラメータを設定します。

手順 6: CDS で名前を解決するために sqlnet.ora ファイルを変更

プロファイル (sqlnet.ora) に必要なパラメータは、使用している SQL*Net または Oracle Net Services のバージョンによって異なります。

DCE CDS ネーミングを使用するクライアントまたはサーバーに対して、管理者は次の作業を行う必要があります。

1. CDS ネーミング・アダプタがそのノード上にインストールされていることを確認します。
2. 次のパラメータを sqlnet.ora ファイルに追加します。

```
NAMES.DIRECTORY_PATH=(cds, tnsnames, onames)
```

このパラメータの値として最初にリストされている名前解決サービスが使用されます。このサービスがなんらかの理由で使用できない場合は、次の名前解決サービスが使用されます。

DCE 環境の Oracle データベースへの接続

この章では、Oracle DCE Integration をインストールし、Oracle DCE Integration を使用できるように DCE と Oracle を構成した後で、Oracle データベースに接続する方法について説明します。

次の項目について説明します。

- [リスナーの起動](#)
- [DCE 環境の Oracle データベース・サーバーに接続](#)

リスナーの起動

次の手順に従って、リスナーを起動します。

1. 次のコマンドを入力します。

```
% dce_login principal_name password
% lsnrctl start listener_name
```

たとえば、`listener.ora` ファイルで設定されている `LSNR_DCE` がリスナー名の場合は、次のように入力します。

```
% dce_login oracle orapwd
% lsnrctl start LSNR_DCE
```

2. 次のように、サーバーがそのバインド・ハンドラを `rpcd` に登録していることを確認します。

```
% rpccp show mapping
```

リスナー・アドレスの一部を構成する `dce_service_name` が含まれている行を探します。

3. 次のように、`dce_service_name` を検索することによって、サービスが作成されていることを確認します。

```
% cdscp show object "/./subsys/oracle/service_registry/dce_service_name"
```

例：

次のコマンドによって、リスナーがエンドポイントとして選択した `CDS` ネームスペース内のマッピングが表示されます。

```
% cdscp show object "/./subsys/oracle/service_registry/dce_svc"

      SHOW
OBJECT    /.../subsys/oracle/service_registry/dce_svc
      AT   1999-05-15-17:10:52
RPC_ClassVersion = 0100
      CDS_CTS = 1999-05-16-00:05:01.221106100/aa-00-04-00-3e-8c
      CDS_UTS = 1999-05-16-00:05:01.443343100/aa-00-04-00-3e-8c
      CDS_Class = RPC_Server
CDS_ClassVersion = 1.0
      CDS_Towers = :
      Tower = ncacn_ip_tcp:144.25.23.57 []
```

DCE 環境の Oracle データベース・サーバーに接続

次のいずれかの方法を使用して、DCE 環境の Oracle サーバーに接続します。

- [方法 1](#)
- [方法 2](#)

方法 1

外部的に識別されるアカウントをセットアップすると、ユーザー名 / パスワード情報を入力しなくても、DCE 認証を利用して Oracle にログインできます。次のようなコマンドで DCE にログインするのみで、このシングル・サインオン機能を使用できます。

```
% dce_login principal_name password
```

例：

```
% dce_login oracle orapwd
```

注意： dce_login コマンドを入力する必要があるのは 1 度のみです。すでに DCE にログインしている場合は、再びログインする必要はありません。

これで、ユーザー名またはパスワードを使用しないで Oracle サーバーに接続できます。次のようなコマンドを入力します。

```
% sqlplus /@net_service_name
```

net_service_name はデータベース・サービス名です。

例：

```
% sqlplus /@ORADCE
```

方法 2

クライアントからユーザー名 / パスワードを使用して接続できます。

```
% sqlplus username/password@net_service_name
```

net_service_name は Oracle Net のネット・サービス名です。

例：

```
% sqlplus scott/tiger@ORADCE
```

DCE 環境と非 DCE 環境の相互運用性

この章では、非 DCE 環境のクライアントが DCE 環境の Oracle サーバーに接続する方法、および CDS にアクセスできないときにローカル・ネーミング構成ファイル `tnsnames.ora` を使用して名前を参照する方法について説明します。

次の項目について説明します。

- 非 DCE 環境のクライアントから DCE 環境の Oracle サーバーに接続
- サンプル・パラメータ・ファイル
- CDS にアクセスできないときに、`tnsnames.ora` を使用して名前を検索

非 DCE 環境のクライアントから DCE 環境の Oracle サーバーに接続

クライアントは DCE と CDS にアクセスできなくても、TCP/IP またはその他のプロトコルを使用して、DCE の Oracle サーバーに接続できます（リスナーを適切に構成してある場合）。サーバー上の `listener.ora` ファイルでリスナーが構成されている場合、非 DCE 環境のクライアントは Oracle*i* および Oracle Net Services の通常の手順を使用して、DCE の Oracle サーバーに接続できます。

注意： この場合、クライアントは DCE のセキュリティ機能を利用できません。また、サービス名はネットワーク・アドレスに解決され、クライアント上の `tnsnames.ora` ファイルで位置が特定されます。CDS ネーム・サーバーは使用されません。

次の項に、非 DCE 環境のクライアントが DCE 環境の Oracle データベース・サーバーに接続する場合に構成する `listener.ora` ファイルと `tnsnames.ora` ファイルのサンプルを示します。

サンプル・パラメータ・ファイル

クライアント / サーバー間で正しく通信を行うためには、少なくとも次の 2 つの Oracle パラメータ・ファイルが必要です。テキスト・エディタを使用して、これらのファイルを作成および変更できます。

これらのパラメータ・ファイルについて次の項で説明します。

- [listener.ora ファイル](#)
- [tnsnames.ora ファイル](#)

listener.ora ファイル

`listener.ora` ファイルはリスナー・ノード上にあります。このファイルは、リスナー特性およびリスニングが行われる場所のアドレスを定義します。

次の例では、各要素を別々の行に配置してファイルの構造をわかりやすくしています。この書式を使用することをお薦めしますが、各要素は必ずしも別々の行に配置する必要はありません。ただし、適切なカッコをすべて記述し、次の行に要素が続く場合は字下げしてください。

この例では、一方のリスナーで UNIX オペレーティング・システムと TCP/IP プロトコルを想定し、別のリスナーで DCE プロトコルを想定しています。1 つのリスナーが複数のアドレスを持つことができます。たとえば、サーバー・ノード上の異なるデータベース・インスタンスに対して 2 つのリスナーを定義するかわりに、両方のデータベース・インスタンスに

対して1つのリスナーを定義して、TCP/IP と DCE でリスニングできます。ただし、パフォーマンスについては個々のリスナーごとに向上されます。

```
LSNR_TCP=
  (ADDRESS_LIST=
    (ADDRESS=
      (PROTOCOL=IPC)
      (KEY=DB1)
    )
    (ADDRESS=
      (PROTOCOL=tcp)
      (HOST=rose)
      (PORT=1521)
    )
  ))

SID_LIST_LSNR_TCP=
  (SID_DESC=
    (SID_NAME=ORASID)
    (ORACLE_HOME=/usr/jprod/Oracle9i)
  )

LSNR_DCE=
  (ADDRESS=
    (PROTOCOL=DCE)
    (SERVER_PRINCIPAL=oracle)
    (CELL_NAME=cell1)
    (SERVICE=dce_svc)
  )
  SID_LIST_LSNR_DCE=
    (SID_DESC=
      (SID_NAME=ORASID)
      (ORACLE_HOME=/usr/prod/oracle8))
  )

#For all listeners, the following parameters list sample
#default values.

PASSWORDS_LISTENER=
STARTUP_WAIT_TIME_LISTENER=0
CONNECT_TIMEOUT_LISTENER=10
TRACE_LEVEL_LISTENER=OFF
TRACE_DIRECTORY_LISTENER=/usr/prod/Oracle9i/network/trace
TRACE File_LISTENER=listener.trc
LOG_DIRECTORY_LISTENER=/usr/prod/Oracle9i/network/log
LOG_FILE_LISTENER=listener.log
```

tnsnames.ora ファイル

このファイルは、クライアント・ノード上とサーバー・ノード上の両方にあります。このファイルには、ネットワーク上のすべてのサービスのサービス名とアドレスのリストが入っています。

次に示すサンプル `tnsnames.ora` ファイルは、TCP/IP アドレスが入っている接続記述子に `ORATCP` サービス名をマップし、DCE アドレスが入っている接続記述子に `ORADCE` サービス名をマップします。

```
ORATCP = (DESCRIPTION=
  (ADDRESS=
    (PROTOCOL=TCP)
    (HOST=rose)
    (PORT=1521)
  )
  (CONNECT_DATA=
    (SID=DB1)
  )
)
ORADCE=(DESCRIPTION=
  (ADDRESS=
    (PROTOCOL=DCE)
    (SERVER_PRINCIPAL=oracle)
    (CELL_NAME=cell1)
    (SERVICE=dce_svc)
  )
  (CONNECT_DATA=
    (SID=ORASID)
  )
)
```

DB1 データベースにアクセスする場合は、適切な接続記述子が指定されている `ORATCP` を使用します。

例：

```
sqlplus scott/tiger@oratcp
```


CDS にアクセスできないときに、tnsnames.ora を使用して名前を検索

通常は、CDS によって名前がネットワーク・アドレスに解決されます。(Native Naming アダプタと関連して) tnsnames.ora を使用する主な目的は、Oracle サービス名とネットワーク・アドレスを CDS にロードすることですが、CDS にアクセスできない場合は、予備の名前解決サービスとして tnsnames.ora を一時的に使用できます。

SQL*Net 2.2 以前のリリース

tnsnames.ora を使用して名前を検索し解決するには、クライアント上の sqlnet.ora ファイルから「固有名」パラメータを削除（またはコメント・アウト）します。行をコメント・アウトするには、次のように各行の先頭にポンド記号（#）を追加します。

例：

```
#native_names.use_native=true
#native_names.directory_path=(dce)
```

SQL*Net リリース 2.3 と Oracle Net Services

クライアント上の sqlnet.ora ファイルで NAMES.DIRECTORY_PATH パラメータの値として TNSNAMES を指定してある場合は、DCE CDS が使用できないときに tnsnames.ora を使用して名前を参照し解決できます。

例：

```
names.directory_path=(dce, tnsnames)
```

このパラメータでは、複数の名前解決メソッドを指定できます。指定した順序で名前解決メソッドが使用されます。上の例では、最初に DCE が使用され、DCE が失敗すると次に TNSNAMES が使用されます。

第 V 部

Oracle9i エンタープライズ・ユーザー・セキュリティ

第V部では、クライアント / サーバー環境においてシングル・サインオンを使用可能にする、Oracle9i のディレクトリとセキュリティを統合した機能について説明します。

次の各章で、使用している Oracle データベース環境でエンタープライズ・ユーザー・セキュリティを設定する方法を説明します。

- 第 15 章「エンタープライズ・ユーザー・セキュリティの管理」
- 第 16 章「ローカルまたは外部ユーザーからエンタープライズ・ユーザーへの移行」
- 第 17 章「Oracle Wallet Manager の使用方法」
- 第 18 章「Oracle Enterprise Login Assistant の使用方法」
- 第 19 章「Oracle Enterprise Security Manager の使用方法」

エンタープライズ・ユーザー・セキュリティの管理

エンタープライズ・ユーザー・セキュリティでは、安全な **LDAP** 準拠のディレクトリ・サービスを使用して、多数のユーザーを作成および管理できます。この章では、エンタープライズ・ユーザー・セキュリティの構成と設定について説明します。

次の項目について説明します。

第 I 部：概要 / 概念：

- エンタープライズ・ユーザー・セキュリティの概要
- 共有スキーマ
- カレント・ユーザー・データベース・リンク
- エンタープライズ・ユーザー・セキュリティのツール
- 配置に関する考慮事項

第 II 部：SSL 認証とパスワード認証の初期構成

第 III 部：SSL 認証の最終構成

第 IV 部：パスワード認証の最終構成

第 V 部：エンタープライズ・ユーザー・セキュリティのトラブルシューティング

関連項目： 第 19 章「Oracle Enterprise Security Manager の使用方法」

第 I 部：概要 / 概念

第 I 部では、Oracle エンタープライズ・ユーザー・セキュリティの概要を示し、その基本的な概念について説明します。

第 I 部の内容は、次のとおりです。

- [エンタープライズ・ユーザー・セキュリティの概要](#)
- [共有スキーマ](#)
- [カレント・ユーザー・データベース・リンク](#)
- [エンタープライズ・ユーザー・セキュリティのツール](#)
- [配置に関する考慮事項](#)

エンタープライズ・ユーザー・セキュリティの概要

この項では、エンタープライズ・ユーザー・セキュリティに関する基本的な概念について説明します。

- [エンタープライズ・ユーザー・セキュリティ](#)
- [エンタープライズ・ユーザーと認証方式](#)
- [エンタープライズ・ユーザーとパスワード認証](#)
- [エンタープライズ・ユーザー・セキュリティのディレクトリ・エントリ](#)
- [SSL 使用時のエンタープライズ・ユーザー・セキュリティのプロセス](#)
- [パスワード使用時のエンタープライズ・ユーザー・セキュリティのプロセス](#)

エンタープライズ・ユーザー・セキュリティ

管理者は、企業全体の複雑なユーザー情報を管理し、それらを最新かつ安全な状態に保つ必要があります。このタスクは、テクノロジーの利用が増加し、企業内でのユーザーの異動が頻繁になるにつれて、ますます難しいものとなっています。一般的な企業では、個々のユーザーが複数のデータベース上に複数のアカウントを持ち、それらのアカウントごとにパスワードを覚えておくよう求められています。そのため、ユーザーはパスワードの数が多すぎて覚えられず、管理者はアカウント数が多すぎて効率的に管理できないという事態が起こっています。

管理者は、何千ものデータベース・アカウントにアクセスする何千ものユーザーを管理するために、多くのリソースを集中させる必要があります。ユーザー名、電話番号、システム・ロール、システム権限など、複数のアプリケーションで使用される共通の情報は、ほとんどの場合、企業内に分散して存在するため、データが冗長で一貫性のないものとなり、データの管理が困難になる原因となっています。

また、セキュリティ上の問題もあります。たとえば、あるユーザーが退職するか、業務が変更になった場合は、そのユーザーの権限が悪用されることを防ぐため、その日のうちにユーザーの権限を変更する必要があります。しかし、大規模な企業では、ユーザーのアカウントとパスワードが複数のデータベースに分散しており、管理者は適時に変更できない場合があります。また、ユーザーがパスワードをメモに書き留めたり（他者がパスワードを簡単に書き写すことができる）、覚えやすいパスワードを使用したり（他者がパスワードを簡単に推測できる）、複数のアプリケーションで同じパスワードを使用したりする場合があります。複数のパスワードを覚えておくためのユーザーの工夫はすべて、企業のセキュリティを危険にさらす可能性があります。

エンタープライズ・ユーザー・セキュリティでは、ユーザー関連情報の格納や管理を LDAP 準拠のディレクトリ・サービスに一元化することで、これらのユーザー、管理者およびセキュリティの問題に対応しています。このような環境では、従業員の業務が変更された場合でも、管理者は1つの場所、つまりディレクトリに格納されている情報を変更するだけで、複数のデータベースおよびシステムに有効な変更を加えることができます。また、一元化によって管理コストが大幅に削減されるとともに、企業のセキュリティも向上します。

エンタープライズ・ユーザーも同様に、エンタープライズ・ユーザー・セキュリティの利点を活用できます。エンタープライズ・ユーザーは、管理者が選択した構成に従って、**シングル・サインオン**または**単一パスワード認証**を使用できます。

シングル・サインオンを使用すると、ユーザーは1度のみ認証され、それ以降の認証は透過的に行われます。これにより、複数パスワードの問題が解消され、より厳密な公開鍵インフラストラクチャ（PKI）ベースの認証が可能になるとともに、ユーザーの操作性が向上します。

単一パスワード認証では、エンタープライズ・ユーザーは、単一のグローバルなパスワードで複数のデータベースに対して認証されます。ただし、各データベースでは個別に認証を行う必要があります。パスワードは、中央に位置する LDAP 準拠のディレクトリに安全に格納され、暗号化や**アクセス制御リスト**（ACL）などのセキュリティ・メカニズムによって保護されます。このアプローチによって、覚えておく必要のあるパスワードの数が減り、

Secure Sockets Layer (SSL) の設定に伴うオーバーヘッドが不要になるため、操作性の向上につながります。

Oracle9i リリース 2 (9.2) エンタープライズ・ユーザー・セキュリティは、次の LDAP 準拠のディレクトリ・サービスをサポートしています。

- Oracle Internet Directory リリース 9.2
- Microsoft Active Directory
(Oracle8i の機能のみをサポート)

Oracle Advanced Security には、ディレクトリにユーザー・エントリを作成し、それらのユーザーの認可を管理するためのツールとして、Oracle Enterprise Security Manager も用意されています。

関連項目： [第 19 章「Oracle Enterprise Security Manager の使用方法」](#)

注意： Microsoft Active Directory は、Windows プラットフォーム上の Oracle データベースに対してのみサポートされます。

エンタープライズ・ユーザーと認証方式

エンタープライズ・ユーザー・セキュリティでは、1 つの LDAP 準拠のディレクトリで次の 2 種類のユーザーを管理できます。

- パスワード認証エンタープライズ・ユーザー
- SSL 認証エンタープライズ・ユーザー

パスワード認証エンタープライズ・ユーザーは、[単一パスワード認証](#)を使用します。

SSL 認証ユーザーは、SSL v3 を介して業界標準の相互運用可能な X.509 v3 証明書を使用し、[シングル・サインオン](#)と厳密認証を利用できます。

各認証方式には、固有の選択条件があります。それをまとめたものが[表 15-1](#)です。

表 15-1 エンタープライズ・ユーザー認証：選択条件

適用可能な選択条件	
パスワード認証	SSL 認証
パスワード・ベースのログインをサポートします。	より強力な、システム全体のセキュリティを提供します。
PKI をサポートしません。つまり、PKI 証明書ベースのクライアント認証は使用できません。	PKI 証明書ベースの認証をサポートします。

表 15-1 エンタープライズ・ユーザー認証：選択条件（続き）

適用可能な選択条件	
パスワード認証	SSL 認証
SSL、Wallet または X.509 証明書を使用しません。	PKI、SSL および業界標準の X.509 証明書をサポートします。
単一のグローバルなユーザー・パスワードをサポートします。データベースおよびアプリケーションごとに個別の認証が必要です（ 単一パスワード認証 ）。	SSL を使用した シングル・サインオン をサポートします。
クライアントとサーバー間で SSL の処理が不要なので、処理が高速です（Oracle Advanced Security 固有の暗号化をサポートしています）。	接続処理に多少時間がかかりますが、ネットワーク・セキュリティは強固です。
ユーザー / クライアントにとって、パスワード認証は使いやすく、特にノートブック PC やホーム・ワークステーションに適していると考えられます。	SSL は、PKI 資格証明をすべてのユーザーに対して生成する必要があるため、初期構成がパスワード認証よりも複雑ですが、より強力なセキュリティを提供します。
パスワード認証は、2 層環境と 3 層環境のいずれとも互換性があります。	SSL 認証は、2 層環境と 3 層環境のいずれとも互換性があります。
パスワード認証は、Oracle9i データベースを使用する Oracle リリース 7.3 または 8.0 のクライアントをサポートします。	SSL 認証は、Oracle9i データベースを使用する Oracle8i または Oracle9i のクライアントをサポートします。

注意： エンタープライズ・ユーザー・セキュリティは 3 層環境をサポートしています。Oracle9i **プロキシ認証**の機能により、1) 複数層を介したユーザー名とパスワードのプロキシ、および 2) 複数層を介した X.509 証明書と識別名のプロキシが使用可能です。

関連項目：『Oracle9i アプリケーション開発者ガイド - 基礎編』

エンタープライズ・ユーザーとパスワード認証

Oracle Advanced Security の Oracle8i のリリースでは、SSL 処理を使用して、1) クライアントとサーバー間、および 2) データベース・サーバーと **LDAP** 準拠のディレクトリ間で保護されたチャネルを確立します。クライアント認証メカニズムによって SSL と X.509 v3 証明書が使用されるため、クライアントとサーバーの両方に Oracle Wallet をインストールする必要があります。

Oracle9i では、エンタープライズ・ユーザーのパスワード・ベース認証が可能になり、それによって証明書ベースの認証が補完されています。パスワード・ベース認証には、主に次のような機能が含まれます。

- **エンタープライズ・ユーザーがパスワードを使用してログインできます。** エンタープライズ・ユーザーは、単一のエンタープライズ・ユーザー名とパスワードを使用して、複数のデータベースに接続できます。
- **ユーザー資格証明が LDAP 準拠のディレクトリに格納されます。** エンタープライズ・ユーザーの資格証明と認可は、集中管理された LDAP 準拠のディレクトリに格納されます。これには、ユーザー名と暗号化されたパスワードのベリファイア、エンタープライズ・ロール、およびユーザーがアクセス権を付与されている各データベースの個々のスキーマへのマッピングが含まれます。
- **クライアント側に SSL と Wallet は必要ありません。** エンタープライズ・ユーザーのパスワード・ベース認証が実装されたことにより、クライアントに SSL と資格証明管理ツールをインストールする必要がなくなりました。

注意： サーバー側では以前のリリースと同様、SSL と Oracle Wallet を両方ともインストールし、使用する必要があります。これは、データベースと LDAP 準拠のディレクトリ間、およびカレント・ユーザー・データベース・リンクを使用するデータベース間で保護されたチャネルを確立するためです。

エンタープライズ・ユーザー・セキュリティのディレクトリ・エントリ

エンタープライズ・ユーザー・セキュリティには、次のディレクトリ・エントリが関係しています。

- [エンタープライズ・ユーザー](#)
- [エンタープライズ・ロール](#)
- [エンタープライズ・ドメイン](#)
- [データベース・サーバー・エントリ](#)
- [ユーザー / スキーマのマッピング](#)
- [管理グループ](#)

エンタープライズ・ユーザー

[エンタープライズ・ユーザー](#)は、ディレクトリ内で定義および管理されるユーザーです。各エンタープライズ・ユーザーは、企業内で固有の識別情報を持っています。各エンタープライズ・ユーザーのエントリは、ディレクトリ内のどの位置にも配置できます。

次の項で説明するエントリは、[Oracle コンテキスト](#)内にのみ配置できます。

エンタープライズ・ロール

エンタープライズ・ユーザーに[エンタープライズ・ロール](#)を割り当てることができ、このエンタープライズ・ロールによってデータベースに対するアクセス権限が判別されます。これらのエンタープライズ・ロールもディレクトリ内に格納され、管理されます。[図 15-1](#) の例では、「OracleDefaultDomain」の下に「Manager」と呼ばれるエンタープライズ・ロールがあります。

エンタープライズ・ロールは1つ以上の[グローバル・ロール](#)から構成され、それぞれのグローバル・ロールは特定のデータベースで定義されています。グローバル・ロールの権限はデータベースに格納されますが、グローバル・ロール自体はディレクトリで管理されます。したがって、エンタープライズ・ロールはグローバル・ロールのコンテナといえます。たとえば、エンタープライズ・ロール USER には、人事管理データベースに対する権限を持つグローバル・ロール HRCLERK と、給与データベースに対する権限を持つロール ANALYST を含めることができます。

[エンタープライズ・ロール](#)は、1つ以上のエンタープライズ・ユーザーに割り当てるができます。たとえば、エンタープライズ・ロール USER を、同じ職責を持つ多数のエンタープライズ・ユーザーに付与できます。この情報はディレクトリ内で保護されており、管理者のみがユーザーの管理とロールの付与を実行できます。ユーザーは、エンタープライズ・ロール以外に、データベース内のローカル・ロールや権限も得ることができます。

[エンタープライズ・ドメイン](#)・サブツリーには、エンタープライズ・ロールのエントリがあります。各エントリには、各サーバーの対応するグローバル・ロールと、ロールが付与され

るエンタープライズ・ユーザーの情報が含まれています。これらの情報は、ドメイン管理者が Oracle Enterprise Security Manager を使用して作成します。

関連項目： 19-38 ページ「[エンタープライズ・ロールの管理](#)」

注意： データベースでは、ユーザーがログインするときに、そのユーザーのグローバル・ロールを取得します。ディレクトリ内のユーザーのグローバル・ロールを変更しても、次にユーザーがログインするまでその変更は適用されません。

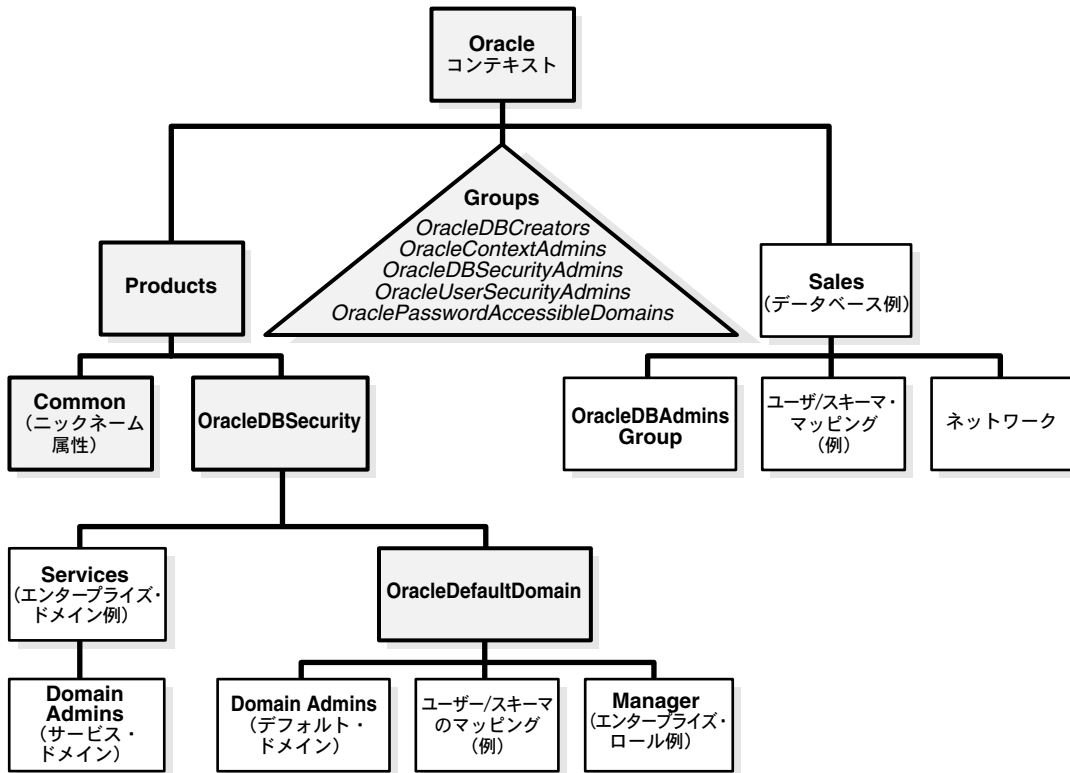
エンタープライズ・ドメイン

エンタープライズ・ドメインとは、データベースおよびエンタープライズ・ロールのグループのことです。ドメインの例としては、企業の技術部門または小規模の企業などがあります。[図 15-1](#) の例では、Oracle コンテキストの「OracleDBSecurity」の下に常駐する「Services」と呼ばれるエンタープライズ・ドメインがあります。ドメイン管理者が Oracle Enterprise Security Manager を使用して、エンタープライズ・ロールをユーザーに割り当て、エンタープライズ・セキュリティを管理するのは、このエンタープライズ・ドメイン・レベルです。ディレクトリ内のエンタープライズ・ドメイン・サブツリーには、エンタープライズ・ロール・エントリ（15-7 ページの「[エンタープライズ・ロール](#)」を参照）、[ユーザー / スキーマのマッピング](#)およびそのドメインのドメイン管理者グループという 3 種類のエントリがあります。

データベース・サーバー・エントリ

データベース・サーバー・エントリ（[図 15-1](#) で「Sales」として表されている）には、データベース・サーバーに関する情報が含まれます。データベース・サーバー・エントリは、データベースの登録時に Database Configuration Assistant または Oracle Enterprise Security Manager によって作成されます。データベース・サーバー・エントリは、完全な DN または部分的な DN と Oracle 共有スキーマ名との間のマッピング情報を含むデータベース・レベルのマッピング・エントリの親です。データベース・レベルのマッピング・エントリは、**データベース管理者**が Oracle Enterprise Security Manager を使用して作成します。そのデータベースの管理者を含む**データベース管理者**のグループは、サーバー・エントリの下に配置されます。

図 15-1 Oracle コンテキスト内の関連エントリ



ユーザー / スキーマのマッピング

ユーザー / **スキーマ・マッピング**・エントリには、DN と Oracle データベース・ユーザー名との間のマッピング情報が含まれています。マッピングで参照されているユーザーは、データベースに接続したときに特定のスキーマに接続されます。ユーザー / スキーマ・マッピング・エントリは、特定のデータベースにのみ、またはドメイン内のすべてのデータベースに適用されます。

関連項目： 15-21 ページ「[エンタープライズ・ユーザーのスキーマへのマッピング](#)」

管理グループ

Oracle コンテキストには、エンタープライズ・ユーザー・セキュリティに関連する管理グループが含まれています。図 15-1 は、Oracle コンテキスト内のこれらの管理グループを示しています。各管理グループには、グループ自体に対するアクセスを制御する[アクセス制御リスト](#)があります。ディレクトリ内の別の場所にある ACL が、これらの管理グループを参照することもあります。これによって、ディレクトリ管理者が別のディレクトリにアクセスして、必要な管理タスクを実行できるようになります。Oracle Net Configuration Assistant を使用して Oracle コンテキストを作成したユーザーは、自動的に各グループの最初のメンバーとなります。

単一のエンタープライズ・ドメインの管理をドメイン管理者に委任することもできます。同様に、単一データベースのディレクトリ・エントリの管理を[データベース管理者](#)に委任することもできます。これらの管理グループは、関連するデータベース・エントリまたはドメイン・エントリの直下に配置されます。

Oracle コンテキストの作成者はデフォルトで各グループのメンバーになりますが、削除することもできます。LDAP の制限によって、各グループには最低 1 人のメンバーが必要であることに注意してください。

注意： Oracle コンテキスト内に格納されているオブジェクトの ACL は変更しないでください。Oracle Enterprise Security Manager や Database Configuration Assistant などの Oracle のツール製品は、エンタープライズ・ユーザー・セキュリティのディレクトリ・エントリを変更する目的にのみ使用してください。その他の目的に使用すると、これらのオブジェクトのセキュリティ構成が壊れ、エンタープライズ・ユーザー機能も壊れる可能性があります。

Oracle コンテキスト内の関連する管理グループを 15-11 ページの表 15-2 に示します。

表 15-2 Oracle コンテキスト内の管理グループ

管理グループ	説明
OracleDBCreators	<p>OracleDBCreators グループのメンバー (cn=OracleDBCreators,cn=OracleContext...) はデータベースの新規作成を担当しますが、その中には、Database Configuration Assistant または Oracle Enterprise Security Manager を使用して各データベースをディレクトリに登録する作業も含まれます。このグループのメンバーは、データベース・サービス・オブジェクトと属性に対する作成および変更のアクセス権を持ちます。このグループのメンバーは、デフォルト・ドメインを変更することもできます。</p> <p>これらのアクセス権は、Oracle コンテキストの作成時に Oracle Net Configuration Assistant によって設定されます。</p> <p>OracleContextAdmins グループのメンバーは、Oracle Enterprise Security Manager を使用してユーザーをこのグループに追加できます。</p> <p>Oracle Enterprise Security Manager では、このグループを Database Registration Admins と呼びます。</p>
OracleContextAdmins	<p>OracleContextAdmins は、あらゆるグループを変更できます。</p> <p>Oracle Enterprise Security Manager では、このグループを Full Context Management と呼びます。</p>
OracleDBSecurityAdmins	<p>OracleDBSecurityAdmins のメンバー (cn=OracleDBSecurityAdmins,cn=OracleContext...) は、OracleDBSecurity サブツリーに対するルート権限を持っています。このグループのメンバーは、エンタープライズ・ユーザー・セキュリティに対する作成、変更および読込みのアクセス権を持ちます。エンタープライズ内のすべてのドメインについて許可を持ち、次のような責任を担っています。</p> <ul style="list-style-type: none"> ■ OracleDBSecurityAdmins グループの管理。 ■ 新規のエンタープライズ・ドメインの作成。 ■ 企業内で、データベースをあるドメインから別のドメインに移動。 <p>これらのアクセス権は、Oracle コンテキストの作成時に Oracle Net Configuration Assistant によって設定されます。</p> <p>OracleContextAdmins 以外に、このグループのメンバーも、Oracle Enterprise Security Manager を使用して他のユーザーをこのグループに追加できます。</p> <p>Oracle Enterprise Security Manager では、このグループを Database Security Management と呼びます。</p>

表 15-2 Oracle コンテキスト内の管理グループ（続き）

管理グループ	説明
OracleUserSecurity-Admins	OracleUserSecurityAdmins のメンバー (cn=OracleUserSecurityAdmins,cn=Groups,cn=OracleContext...) は、Oracle ユーザーのセキュリティを担当します。たとえば、このグループのメンバーは、デフォルトで Wallet のパスワード・ヒントを読み込み、ユーザーのパスワードを変更できます。関連する ACL は、デフォルトで Oracle Internet Directory のルート・ディレクトリに設定されています。 Oracle Enterprise Security Manager では、このグループを Directory User Management と呼びます。
OraclePasswordAccessibleDomains	このグループのメンバーは、パスワード認証エンタープライズ・ユーザーが使用可能なデータベースを含むエンタープライズ・ドメインです。

ユーザーのデータベース・ログイン情報のセキュリティ

概要 パスワード・ベースの安全な認証方式では必ず、サーバーがパスワード・ベリファイア（通常はパスワードがハッシュ化されたもの）を使用してクライアントを認証します。このパスワード・ベリファイアは厳重に保護する必要があります。**Oracle** データベースでのパスワード認証もこれと同じです。**Oracle** 独自のパスワード・ベリファイアが用意されており、同様に保護する必要があります。これは、ベリファイアがデータベース内にローカルに格納されている場合、またはディレクトリで集中管理されている場合のどちらにも当てはまります。パスワード・ベリファイアを使用して、元のパスワードを取得することはできません。

現行のリリースでは、エンタープライズ・ユーザーのデータベース・パスワードを、複数のデータベースからアクセスできるように中央のディレクトリ・サービスに格納し、そのユーザーがアクセス可能な信頼できるすべてのデータベースでデータベース・パスワードを参照および共有できます。ディレクトリに格納されているパスワード・ベリファイアは**クリアテキスト**のパスワードではありませんが、偶発的なアクセスや不正なアクセスから保護する必要があります。したがって、ディレクトリ内にパスワードに関連した **ACL** を定義することが非常に重要になります。この **ACL** には、必要なアクセスと操作性を確保しながら、可能なかぎり制限を課すようにします。

パスワード・ベリファイアを保護するためにディレクトリ内に **ACL** を設定するには、**Oracle** のツール製品が役立ちます。オラクル社がお勧めするアプローチは、セキュリティと操作性のバランスを保つことを目的としています。最大限のセキュリティが必要であり、すべてのユーザーの **Wallet** を設定可能な場合は、ユーザーからデータベースへの接続に **SSL** のみを使用します。**SSL** のみのアプローチを使用すると、ディレクトリ全体のパスワードを保護する問題について検討する必要がなくなります。

ACL の設定 各 **Oracle** コンテキストの **OraclePasswordAccessibleDomains** グループは、コンテキストが作成されるときに自動的に作成され、**Oracle Enterprise Security Manager** を使用して管理できます。このグループには、ディレクトリに格納されているユーザーのデータ

ベース・パスワード・ベリファイアを参照する必要があるデータベースをメンバーとして含むエンタープライズ・ドメインを配置します。Enterprise Security Manager を使用して、ユーザーのサブツリーに適切な ACL を配置すると、このグループに所属するデータベースから、そのサブツリー内にあるユーザーのパスワード・ベリファイアを読み込むことができます。

このアプローチでは、次の 2 つの手順を実行する必要があります。

1. 選択した Oracle コンテキストに対して、パスワード認証接続を受け入れるデータベースを決定します。Oracle Enterprise Security Manager を使用して、それらのデータベースを含むドメインを OraclePasswordAccessibleDomains グループに配置します。
2. 選択した Oracle コンテキストに対して、Oracle Enterprise Security Manager を使用し、このコンテキスト内のデータベースにパスワード認証方式で接続するユーザーを含むユーザー検索ベースを選択します。この検索ベース内のユーザー・エントリのパスワード・ベリファイアへのアクセス権を、許可されたドメインのみに付与するために、「Allow Logon」を選択します。

この手順を実行すると、ディレクトリ内で選択したユーザー検索ベース・エントリに対する ACL を設定できます。この ACL によって、Oracle データベースのパスワード・ベリファイアを保持する属性に対して、次のアクセス権が許可されます。

- 選択した Oracle コンテキスト内の OraclePasswordAccessibleDomains グループのメンバーには、属性に対する読み込みアクセス権が付与されます。したがって、メンバー・ドメインに属するすべてのデータベースで、エンタープライズ・ユーザーを認証するためにパスワード・ベリファイアを読み込むことができます。OraclePasswordAccessibleDomains グループに属していないドメイン内のデータベースでは、パスワード認証接続を受け入れることはできません。
- ユーザー検索ベースが 2 つの異なる Oracle コンテキストによって参照され、両方のコンテキスト内のドメインでパスワード認証ユーザーからのアクセスが必要な場合、ACL には、両方のコンテキストの OraclePasswordAssessibleDomains グループが含まれます。
- 他のユーザーがこの属性にアクセスしようとすると、必ず拒否されます。これを規定する ACL は、実際にはディレクトリ・ツリーのルートにあります。

これらの ACL によってアクセスが許可されるのは、Oracle データベースのパスワード・ベリファイア値 (orcldbpassword 属性) のみです。ユーザー・エントリ内のその他の属性には影響しません。

関連項目： エンタープライズ・ユーザー・セキュリティの LDAP スキーマの詳細は、『Oracle9i Directory Service 統合および配置ガイド』を参照してください。

エンタープライズ・ユーザー・セキュリティの要素

図 15-2 は、SSL 認証の場合のエンタープライズ・ユーザー・セキュリティの要素を示しています。

図 15-2 エンタープライズ・ユーザー・セキュリティの要素 (SSL 認証)

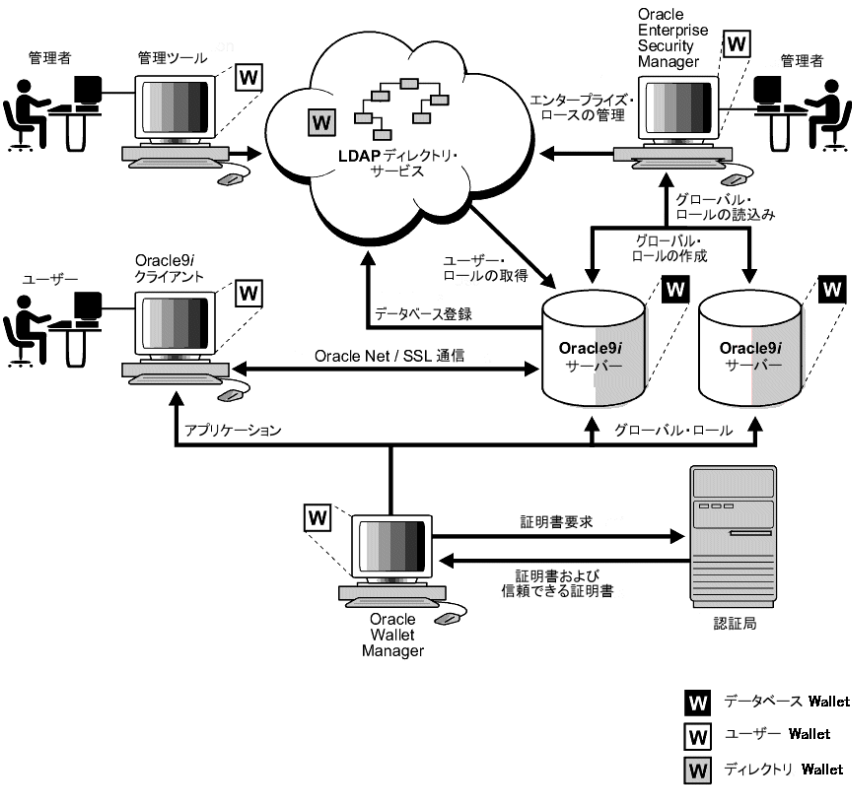
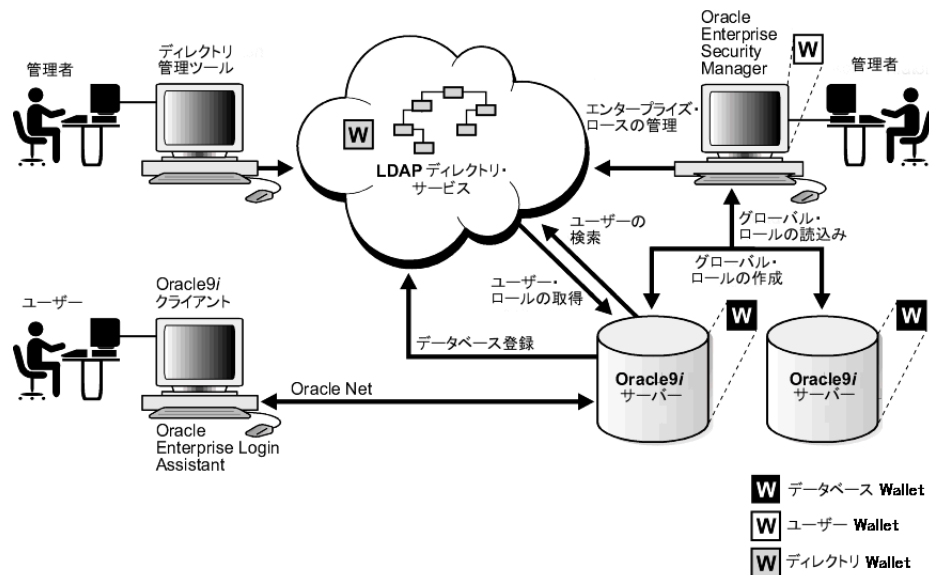


図 15-3 は、パスワード認証の場合のエンタープライズ・ユーザー・セキュリティの要素を示しています。

図 15-3 エンタープライズ・ユーザー・セキュリティの要素（パスワード認証）

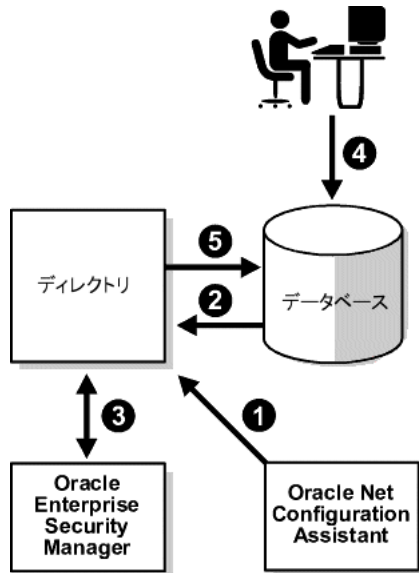


SSL 使用時のエンタープライズ・ユーザー・セキュリティのプロセス

図 15-4 は、エンタープライズ・ユーザー・セキュリティ・プロセスの動作を示しています。SSL 処理環境では、次のことを想定しています。

- ユーザーの Wallet が設定および構成済みであること
- ユーザーのデータベースに対する認証が SSL を使用して行われること

図 15-4 エンタープライズ・ユーザー・セキュリティの動作の仕組み



1. 管理者は Oracle Net Configuration Assistant を使用して、1) ディレクトリで Oracle コンテキストを選択するか、2) 必要に応じて Oracle コンテキストを作成します。
2. OracleDBCreators グループのメンバーは、Database Configuration Assistant または Oracle Enterprise Security Manager を使用してディレクトリにデータベースを登録します。
3. 管理者は、Oracle Enterprise Security Manager を使用して、ディレクトリと関連ドメインにエンタープライズ・ユーザーとエンタープライズ・ロールの両方を設定します。
4. ユーザーが「connect /」を使用してログオンし、データベースへの SSL 接続を開始すると、データベースは SSL を使用してユーザーを認証します。
5. データベースは、このユーザーが所有する専用のスキーマをデータベース上でローカルに検索します。

6. 適切なユーザー・スキーマがローカルで見つからない場合、データベースはディレクトリ内でスキーマを検索します（図 15-4 の手順 2 を参照してください）。スキーマが見つかり、データベースはディレクトリからユーザーのエンタープライズ・ロールを取り出し、対応付けられているグローバル・ロールの中からそのデータベースに適用されるものをすべて使用可能にします。

パスワード使用時のエンタープライズ・ユーザー・セキュリティのプロセス

この項では、パスワード・ベースの認証を使用したエンタープライズ・ユーザー・セキュリティ・プロセスの動作について説明します。次の手順の番号は、15-16 ページの図 15-4 の番号に対応しています。

1. 管理者は Oracle Net Configuration Assistant を使用して、ディレクトリで Oracle コンテキストを選択するか、必要に応じて Oracle コンテキストを作成します。
2. OracleDBCreators グループのメンバーは、Database Configuration Assistant を使用してディレクトリにデータベースを登録します。
3. 管理者は、Oracle Enterprise Security Manager を使用して、ディレクトリと関連ドメインにエンタープライズ・ユーザーとエンタープライズ・ロールを設定し、コンテキスト内の属性を構成します。
4. ユーザーは、ユーザー名とパスワードを使用してログオンすることによって、データベースに対する認証を行います。
5. 認証手続きの過程で、データベースは次の手順を実行します。
 - データベースはユーザー名を検索して、ディレクトリから**識別名**とパスワード・ベリファイアを取り出します。
 - データベースは、取り出したパスワード・ベリファイアに基づいてユーザーを認証し、このユーザーが所有する専用のスキーマをデータベース上でローカルに検索します。
 - データベース上に専用のスキーマが見つからない場合は、ディレクトリ内の共有スキーマを検索します（前述の項の手順 6 と同じです）。
 - データベースは、ユーザーのエンタープライズ・ロールを取り出し、対応付けられているグローバル・ロールの中からそのデータベースに適用されるものをすべて使用可能にします。

注意： 3 層環境では、エンタープライズ・ユーザーは、**プロキシ認証**を使用した中間層を介して、データベースに対する認証を行います。

共有スキーマ

次の各項で、共有スキーマの機能とその設定方法について説明します。

- [概要](#)
- [共有スキーマの構成](#)
- [共有スキーマの作成](#)
- [共有スキーマ](#)
- [エンタープライズ・ユーザーのスキーマへのマッピング](#)

概要

ユーザーは、アクセスするデータベース内に、必ずしも個々のアカウントまたはスキーマを持つ必要はありません。そのかわりに、ターゲット・アプリケーションに関連付けられている [共有スキーマ](#) へのアクセス権を付与できます。たとえば、Tom、Dick および Harriet の 3 人のユーザーが、財務データベース上の給与アプリケーションに対するアクセス権を必要としているとします。これらのユーザーは、データベースに一意のオブジェクトを作成する必要がないため、独自のスキーマを必要としません。ただし、これらのユーザーには、給与スキーマに対するアクセス権が必要です。

Oracle9i リリース 2 (9.2) では、エンタープライズ・ディレクトリに格納されている複数のユーザーを個々のデータベース上の共有スキーマにマップできます。このようにユーザーとスキーマを分離すると、データベース上のユーザー・アカウントの数が減るため、管理コストが削減されます。このことは、ディレクトリにユーザーを作成する以外に、ユーザー・スキーマとも呼ばれる各ユーザーのアカウントを作成する必要がないことを意味します。かわりに、ユーザーを 1 つの場所、つまり、エンタープライズ・ディレクトリに作成し、そのユーザーを、他のエンタープライズ・ユーザーもマッピングできる共有スキーマにマッピングできます。たとえば、Tom、Dick および Harriet 全員が売上データベースと財務データベースの両方にアクセスする場合は、それぞれのデータベース上に各ユーザーのアカウントを作成する必要はありません。かわりに、各データベースにそれぞれ SALES_APPLICATION や FINANCE_APPLICATION など、3 人のユーザー全員がアクセスできる 1 つの共有スキーマを作成できます。一般的な環境では、最大 5,000 のエンタープライズ・ユーザーを 1 つの共有スキーマにマップして、各ユーザーにエンタープライズ・ロールのセットを割り当てることができます。

エントリ・ポイントとして使用するオブジェクトが含まれていない、独立した共有スキーマを作成することをお勧めします。その後、エンタープライズ・ロールを介して、他のスキーマ内のアプリケーション・オブジェクトに対するアクセス権を付与してください。このようにしないと、アプリケーション・オブジェクトが不注意に削除される可能性があります。

要約すると、共有スキーマには次の利点があります。

- 共有スキーマによって、各エンタープライズ・ユーザーに対してデータベースごとに専用のデータベース・スキーマを設定する必要がなくなります。
- 各エンタープライズ・ユーザーを、そのユーザーがアクセスする必要のある各データベース上の共有スキーマにマッピングできます。ユーザーは、データベースに接続するときに、共有スキーマに接続します。
- 共有スキーマによって、エンタープライズ内のユーザーの管理に必要なコストが削減されます。

共有スキーマの構成

共有スキーマを構成するには、ローカルのデータベース管理者（DBA）が、データベースに1つ以上のデータベース・スキーマを作成する必要があります。このスキーマにエンタープライズ・ユーザーをマッピングできます。

次の例では、管理者が共有スキーマを作成し、そのスキーマにユーザーをマッピングします。

- 管理者は、HR というデータベース上に EMPLOYEE というグローバル共有スキーマと HRMANAGER というグローバル・ロールを作成します。
- 管理者は Oracle Enterprise Security Manager を使用して、ディレクトリにエンタープライズ・ユーザーとエンタープライズ・ロールを作成し、管理します。たとえば、Harriet というエンタープライズ・ユーザーと MANAGER というエンタープライズ・ロールを作成します。その後、HR データベースのグローバル・ロール HRMANAGER をエンタープライズ・ロール MANAGER に割り当てます。
- 管理者は、ディレクトリにおいてエンタープライズ・ロールをエンタープライズ・ユーザーに割り当てます。たとえば、エンタープライズ・ロール MANAGER を Harriet に割り当てます。
- 管理者は、Oracle Enterprise Security Manager を使用して、ディレクトリ内のユーザー Harriet を HR データベース上の共有スキーマ EMPLOYEE にマッピングします。

Harriet がその HR データベースに接続すると、EMPLOYEE スキーマに自動的に接続され、グローバル・ロール HRMANAGER が与えられます。1つの共有スキーマに対して複数のエンタープライズ・ユーザーをマッピングできます。たとえば、エンタープライズ・セキュリティ管理者は、別のエンタープライズ・ユーザー Scott を作成し、Scott を EMPLOYEE スキーマにマッピングできます。この場合、HR データベースに接続すると、Harriet と Scott はいずれも EMPLOYEE スキーマを自動的に使用することになりますが、それぞれ異なるロールを持つことができ、個々に監査されます。

共有スキーマの作成

共有スキーマを作成するための構文は次のとおりです。

```
CREATE USER [shared schema name] IDENTIFIED GLOBALLY AS ''
```

たとえば、HR データベースの管理者は、ユーザーの共有スキーマ `EMPLOYEE` を次のように作成します。

```
CREATE USER employee IDENTIFIED GLOBALLY AS ''
```

注意： 共有スキーマを作成するための構文では、引用符で囲まれた部分には空白を入れません。

共有スキーマ

スキーマの識別は次のように行われます。

- データベースは、内部データベース表をチェックして、その DN を持つユーザーが（共有のスキーマではなく）専用のスキーマを所有しているかどうかを確認します。
- 専用のスキーマがローカルで見つからない場合は、ディレクトリ内でユーザーとスキーマの適切なマッピングを検索します。ディレクトリ内のユーザーとスキーマのこのマッピング・オブジェクトによって、ユーザー DN はデータベース・スキーマに関連付けられています。データベースは、次の 1 つ以上の DN マッピング・エントリを検出します。
 - 全 DN（エントリ・レベル） マッピング
 - 部分 DN（サブツリー・レベル） マッピング

エンタープライズ・ユーザーのスキーマへのマッピング

グローバル・スキーマ (CREATE USER IDENTIFIED GLOBALLY AS ' ' を使用して作成されたスキーマ) は、1 人のエンタープライズ・ユーザーが所有 (プライベート・スキーマ)、または複数のエンタープライズ・ユーザーが共有 (共有スキーマ) できます。単一のエンタープライズ・ユーザーとそのユーザーのプライベート・スキーマ間のマッピングは、ユーザー DN とスキーマ名の対応付けとして、データベースに格納されます。ディレクトリ内でのエンタープライズ・ユーザーと共有スキーマ間のマッピングは、1 つ以上のマッピング・オブジェクトによって行われます。マッピング・オブジェクトを使用して、ユーザーの **識別名** を、そのユーザーがアクセスするデータベース・スキーマにマップします。マッピング・オブジェクトの作成には、Oracle Enterprise Security Manager を使用します。マッピングには次のタイプがあります。

- 全 DN (エントリ・レベル) マッピング

このメソッドによって、単一のディレクトリ・ユーザーの DN がデータベース上の特定のスキーマに関連付けられます。その結果、1 人のユーザーにつき 1 つのマッピング・エントリが作成されます。

- 部分 DN (サブツリー・レベル) マッピング

このメソッドによって、複数のエンタープライズ・ユーザーが DN の一部を共有でき、1 つの共有スキーマにアクセスできます。このメソッドは、ディレクトリ・ツリー内の共通ルートの下に複数のエンタープライズ・ユーザーがすでにグループ化されている場合に役立ちます。これらのユーザーが共有するサブツリーは、データベース上の共有スキーマにマッピングできます。たとえば、技術部門のサブツリー内のすべてのエンタープライズ・ユーザーをバグ・データベース上の 1 つの共有スキーマ BUG_APPLICATION にマッピングできます。サブツリーのルートは、指定されたスキーマにはマップされません。

エンタープライズ・ユーザーがデータベースに接続する場合、データベースは、ネットワーク (SSL の場合) またはディレクトリ (パスワード認証エンタープライズ・ユーザーの場合) からユーザーの DN を取得します。

データベースでは、ユーザーの接続先スキーマを判断するときに、ユーザー DN および次の優先順位のルールを使用します。

- 最初に、ローカル (データベース内) でプライベート・スキーマを検索します。
- ローカルでプライベート・スキーマが見つからない場合は、ディレクトリ内を検索します。ディレクトリ内では、サーバー・エントリの下で全 DN マッピングを検索し、次に部分 DN マッピングを検索します。
- サーバー・エントリの下でマッピング・オブジェクトが見つからない場合は、エンタープライズ・ドメイン・エントリの下で全 DN マッピングを検索し、次に部分 DN マッピングを検索します。

- ローカルでプライベート・スキーマが見つからない場合、またはデータベース内に適切なマッピング・エントリがない場合は、接続を拒否します。それ以外の場合は、ユーザーを適切なスキーマに接続します。

たとえば、Harriet が HR データベースに接続しようとしているが、データベースが Harriet のプライベート・スキーマをデータベース内で検出できないとします。この場合、HR データベースはディレクトリ内で Harriet の DN を持つユーザー / スキーマのマッピングを調べます。このディレクトリには、共有スキーマ EMPLOYEE に対する Harriet のマッピングがあり、このスキーマが戻されます。データベースは Harriet のログインを許可し、Harriet を EMPLOYEE スキーマに接続します。

- データベースは、ディレクトリからこのデータベースに対するこのユーザーのグローバル・ロールを取り出します。
- また、データベースは、ユーザーがマッピングされているデータベース・スキーマに関連付けられているローカル・ロールと権限を、その表から取り出します。
- データベースは、グローバル・ロールとローカル・ロールの両方を使用して、そのユーザーがアクセスできる情報を判別します。

さらに例を続けて、エンタープライズ・ロール MANAGER に、HR データベース上のグローバル・ロール ANALYST と、給与データベース上のグローバル・ロール USER が含まれているとします。MANAGER というエンタープライズ・ロールを持つ Harriet が HR データベースに接続する場合は、データベース上のスキーマ EMPLOYEE を使用します。

- HR データベースに対する Harriet の権限は、次のものによって判断されます。
 - グローバル・ロール ANALYST
 - HR データベース上の EMPLOYEE スキーマに関連付けられているすべてのローカル・ロールと権限
- Harriet が給与データベースに接続する場合、その権限は次の内容によって判別されます。
 - グローバル・ロール USER
 - 給与データベース上の EMPLOYEE スキーマに関連付けられているすべてのローカル・ロールと権限

データベース・スキーマに対してロールと権限を付与することで、特定のユーザー・グループに権限を付与できます。このようなスキーマを共有するユーザーは、個別のエンタープライズ・ロールの他に、これらのローカル・ロールと権限を取得します。ただし、この共有スキーマにマッピングされるユーザーはすべて、このスキーマに割り当てられた権限を実行できるため、これを行う場合は十分な注意が必要です。そのため、共有スキーマにロールと権限を付与することはお薦めしていません。

カレント・ユーザー・データベース・リンク

Oracle9i では、SSL 認証エンタープライズ・ユーザーとパスワード認証エンタープライズ・ユーザーの両方に対してカレント・ユーザー・データベース・リンクをサポートしています。カレント・ユーザー・データベース・リンクを使用すると、ユーザー自身として、または別のユーザーが所有するストアード・プロシージャ内で使用される場合はその別のユーザーとして、2 番目のデータベースに対する接続を確立できます。この場合のアクセスは、プロシージャの範囲内に制限されています。カレント・ユーザー・データベース・リンクのセキュリティ上の利点は、別のユーザーの資格証明がデータベース・リンク定義に格納されず、データベース間のネットワーク接続を介して送信されないことです。かわりに、これらのリンクのセキュリティは、データベース間の相互信頼、相互認証および安全なネットワーク接続に基づいています。

たとえば、財務データベースのユーザーである Harriet は、プロシージャ内のカレント・ユーザー・データベース・リンクによって Scott として接続し、Scott の資格証明を使用して、買掛管理データベースにアクセスできます。

Harriet がカレント・ユーザー・データベース・リンクにアクセスしてスキーマ Scott に接続するには、Scott が両方のデータベースで IDENTIFIED GLOBALLY として作成されたスキーマであることが必要です。しかし、Harriet は次の 3 つのいずれかの方法で識別されるユーザーになることができます。

- パスワードの使用
- GLOBALLY
- EXTERNALLY

Scott を買掛管理データベースと財務データベースの両方でグローバル・ユーザーとして作成するには、各データベースで次のコマンドを入力する必要があります。

```
CREATE USER Scott IDENTIFIED GLOBALLY as 'CN=Scott,O=nmt'
```

この種のスキーマを作成するための構文は、15-20 ページの「[共有スキーマの作成](#)」で説明した共有スキーマを作成するための構文と多少異なります。この場合、スキーマは Scott のみです。カレント・ユーザー・データベース・リンクを使用できるようにするために、Scott 用に作成されたスキーマは他のユーザーと共有することはできません。

カレント・ユーザー・データベース・リンクは、1 つのエンタープライズ・ドメイン内の信頼できるデータベース間でのみ機能します。この場合、ドメイン内のデータベースは相互に信頼してユーザーを認証します。信頼できるエンタープライズ・ドメインを指定するには、Oracle Enterprise Security Manager を使用します。デフォルトでは、Enterprise Security Manager を使用して、あるドメインのカレント・ユーザー・データベース・リンクを使用可能にした場合、カレント・ユーザー・データベース・リンクはそのドメイン内部のすべてのデータベースに対して機能します。信頼できるエンタープライズ・ドメインの一部であるデータベースを信頼できないものとして指定するには、PL/SQL パッケージ DBMS_DISTRIBUTED_TRUST_ADMIN を使用します。信頼できるサーバーのリストを取得するには、TRUSTED_SERVERS ビューを使用します。

関連項目：

- カレント・ユーザー・データベース・リンクの詳細は、『Oracle9i Heterogeneous Connectivity Administrator's Guide』を参照してください。
- 構文の詳細は、『Oracle9i SQL リファレンス』を参照してください。
- PL/SQL パッケージ DBMS_DISTRIBUTED_TRUST_ADMIN の詳細は、『Oracle9i PL/SQL パッケージ・プロシージャおよびタイプ・リファレンス』を参照してください。
- TRUSTED_SERVERS ビューの詳細は、『Oracle9i データベース・リファレンス』を参照してください。
- [第 7 章「Secure Sockets Layer 認証の構成」](#)
- [第 17 章「Oracle Wallet Manager の使用方法」](#)

エンタープライズ・ユーザー・セキュリティのツール

エンタープライズ・ユーザー・セキュリティ機能では、次の管理ツールを使用します。

- [Oracle Enterprise Security Manager](#)
- [Oracle Enterprise Login Assistant](#)
- [Oracle Wallet Manager](#)

Oracle Enterprise Security Manager

Oracle Enterprise Security Manager は、次の構成要素の管理に役立つ Graphical User Interface (GUI) を備えた管理ツールです。

- エンタープライズ・ユーザー
- エンタープライズ・ドメイン
- エンタープライズ・ロール
- SSL ベース方式およびパスワード・ベース方式のユーザー認証と認可
- Oracle コンテキスト、データベース、セキュリティおよびエンタープライズ・ドメインの管理者
- ユーザー / スキーマのマッピング

関連項目： [第 19 章「Oracle Enterprise Security Manager の使用方法」](#)

Oracle Enterprise Login Assistant

Oracle Enterprise Login Assistant は、自動ログインの使用可能の設定、Wallet のディレクトリへのアップロードとディレクトリからのダウンロード、および Wallet、ディレクトリ、データベース・パスワードの変更に使用します。エンタープライズ・ユーザーは、このツールを使用することにより、SSL を介して複数のサービスにシングル・サインオンで接続できます。Oracle Enterprise Login Assistant は、SSL、Wallet、エンタープライズ・ユーザーおよび複数データベースに対する認証手続きの複雑さをユーザーに意識させません。

また、Oracle Enterprise Login Assistant はパスワード認証をサポートします。これにより、ユーザーはセッションごとに 1 度のみ単一のパスワードを入力して、複数のデータベースとアプリケーションに安全にアクセスできます。

関連項目： [第 18 章「Oracle Enterprise Login Assistant の使用方法」](#)

Oracle Wallet Manager

Oracle Wallet Manager は、Wallet の所有者およびセキュリティ管理者がその Oracle Wallet 内のセキュリティ資格証明を管理および編集するために使用するスタンドアロン型の Java アプリケーションです。

注意： パスワード認証エンタープライズ・ユーザーにはクライアント側の Wallet は必要ありませんが、Oracle Wallet Manager は、データベースとディレクトリ間の安全な接続をサポートするために必要です（サーバー間接続には SSL とサーバー側 Wallet が必要です）。

関連項目： このアプリケーションの使用方法は、[第 17 章「Oracle Wallet Manager の使用方法」](#)を参照してください。

配置に関する考慮事項

エンタープライズ・ユーザー・セキュリティを配置する前に、次の点について検討してください。

- [セキュリティ資格証明の集中管理に伴うセキュリティ](#)
- [エンタープライズ・ドメイン内のデータベースのメンバーシップ](#)

セキュリティ資格証明の集中管理に伴うセキュリティ

エンタープライズ・ユーザーとそれらに対応する資格証明を集中管理すると一般的なメリットが得られますが、その他にも検討が必要なセキュリティ関連のメリットと危険性が生じます。

集中管理に伴うセキュリティ上のメリットとして、ユーザー、資格証明およびロールの管理が容易かつ短時間で行えるようになり、企業内のすべてのアプリケーションおよびデータベースに対するユーザーの権限を迅速に削除できることがあげられます。集中管理された環境では、管理者が 1 箇所でユーザーを削除してすべての権限を取り消すことができるので、不必要な権限が残ってしまうという危険性を最小限に抑えることができます。

もう 1 つのセキュリティ上のメリットは、セキュリティ情報を集中管理したほうが、安全性が高くなることです。これは、セキュリティについて組織が所有する専門知識を集中化できるためです。セキュリティに詳しい専門家が管理者となり、ディレクトリのセキュリティ、ユーザーのロールと権限、データベースへのアクセスなど、エンタープライズ・ユーザー・セキュリティに関係するあらゆる側面を管理できます。これは、一般に[データベース管理者](#)が、セキュリティも含め、管理を担当しているデータベースに関するすべての作業の責任を担っている従来のモデルからの大きな進歩です。

デメリットは、[Oracle Internet Directory](#) がどれほど安全なリポジトリであっても、公的にアクセス可能なリポジトリで資格証明を集中管理するという状況にセキュリティ上の問題と固有の危険性が存在することです。集中管理された資格証明は、少なくとも分散管理された資格証明と同程度に安全に保護できますが、集中化本来の性質のために、不注意で無許可の第三者に資格証明が開示されてしまう可能性が高くなります。したがって、管理者の権限を制限し、ディレクトリ内に制限された ACL を設定するとともに、管理者が一時的にディレクトリから離れる際にセキュリティ資格証明を保護するための適切なセキュリティ・プラクティスを実装することが不可欠です。

エンタープライズ・ドメイン内のデータベースのメンバーシップ

ドメインにおけるデータベースのメンバーシップを定義する際は、次の基準について検討してください。

- カレント・ユーザー・[データベース・リンク](#)は、1 つの[エンタープライズ・ドメイン](#)内にあるデータベースの間でのみ機能します。
- エンタープライズ・ユーザーの適切な認証タイプは、ドメイン・レベルで定義されます。そのため、ドメインにおけるデータベースのメンバーシップもそれに従って定義する必要があります。1 つ以上のデータベースで SSL ベースの証明書認証のみをサポートする場合、それらのデータベースを同じドメイン内でパスワード認証データベースと組み合わせることはできません。
- エンタープライズ・ロールは、ドメイン・レベルで定義されます。複数のデータベース間で[エンタープライズ・ロール](#)を共有するには、そのデータベースが同じドメインのメンバーであることが必要です。

第 II 部 : SSL 認証とパスワード認証の初期構成

第 II 部では、SSL 認証とパスワード認証に関するエンタープライズ・ユーザー・セキュリティの初期構成タスクについて説明します。

第 II 部のタスクは、次のとおりです。

- [前提条件](#)
- [タスク 1: SSL 用のデータベースの構成](#)
- [タスク 2: Wallet の作成とリスナーの起動](#)
- [タスク 3: データベースのインストールの検証](#)
- [タスク 4: グローバル・スキーマとグローバル・ロールの作成](#)

前提条件

次の前提条件を満たしていることが必要です。

- 前提条件 A: 認証局のインストールまたは識別
- 前提条件 B: ディレクトリ・サービスのインストールと構成
- 前提条件 C: ディレクトリ使用構成の完了

前提条件 A: 認証局のインストールまたは識別

有効な Wallet を作成するには、使用している環境で **認証局** (CA) を持つ必要があります。CA ベンダーの証明書を使用するか、または Base 64 形式で PKCS#10 証明書要求を処理し、Base 64 形式で X509v3 証明書を戻すことができる独自の CA を使用できます。

関連項目： 認証局と Oracle Wallet Manager の詳細は、[第 17 章「Oracle Wallet Manager の使用方法」](#)を参照してください。

前提条件 B: ディレクトリ・サービスのインストールと構成

Oracle9i エンタープライズ・ユーザー・セキュリティ リリース 2 (9.2) には、Oracle Internet Directory リリース 9.2 が必要です。これによって、必要なバージョンの Oracle スキーマがインストールされます。Oracle Internet Directory をインストールおよび構成します。エンタープライズ・ユーザー・セキュリティには、Oracle Internet Directory の SSL インスタンスが必要です。また、ディレクトリに Wallet があることが必要です。

注意： セキュリティ上の理由のため、Oracle Internet Directory の構成セットには Wallet のパスワードを格納しないでください。かわりに、ディレクトリ Wallet の自動ログインを使用可能にしてください。

ディレクトリ内に Oracle9i リリース 2 (9.2) のスキーマがあり、適切な Oracle コンテキストがインストールされていることを確認します。Oracle9i リリース 2 (9.2) のスキーマには下位互換性があります。

Oracle コンテキストのアップグレード Oracle Internet Directory は、ディレクトリのルートに Oracle コンテキストが事前にインストールされた状態で出荷されています。Oracle Net Configuration Assistant を使用すると、追加の Oracle コンテキストを作成できます。

ディレクトリ内の Oracle コンテキストに Oracle9i データベースを登録する前に、Oracle8i の Oracle コンテキストをアップグレードする必要があります。このアップグレードした Oracle コンテキストを使用して、今後作成するすべての Oracle8i データベースを登録できます。Oracle8i データベースと Oracle9i データベースを併用している場合は、Oracle Enterprise Security Manager を使用して、VERSIONCOMPATIBILITY パラメータを 8i および 9i に設定します。Oracle9i データベースのみを配置している場合は、このパラメータを 9i に設定することをお勧めします。Oracle8i データベースと Oracle9i データベースの両方をサ

ポートするために、一部のデータベース・セキュリティ属性をディレクトリ内の 2 箇所に表示する必要があるかどうかは、このパラメータによって決まります。Oracle8i データベースのみを配置している場合は、Oracle コンテキストをアップグレードする必要はなく、このパラメータを設定する必要もありません。

注意：

- Oracle8i の Oracle コンテキストを使用して Oracle エンタープライズ・ユーザー・セキュリティがすでに設定されている場合は、Oracle8i コンテキストをアップグレードしても、対応付けられているセキュリティ情報（エンタープライズ・ユーザー、ロール、権限、ドメインなど）はアップグレードできません。この場合は、Oracle9i の Oracle コンテキストを新規に作成して、セキュリティ情報を再作成する必要があります。
 - Oracle8i の Oracle コンテキストで Oracle エンタープライズ・ユーザー・セキュリティを一度も使用したことがない場合は、その Oracle コンテキストを Oracle9i コンテキストにアップグレードして、設定と構成を続行してください。
-
-

管理ユーザーの作成（必要な場合） 管理ユーザーがまだ存在しない場合は、ディレクトリ内に、次の機能の実行を許可されたエンタープライズ・ユーザーを作成します。

- データベースの登録
- データベース・セキュリティの管理
- エンタープライズ・ドメインの作成と管理

注意： 1 人の管理者で、エンタープライズ・ユーザー・セキュリティのすべての管理機能を実行できます。この方法を選択する場合は、管理ユーザー・エントリを 1 つのみ作成します。

ただし、Oracle エンタープライズ・ユーザー・セキュリティには、セキュリティ機能を異なる担当者に割り当てることができるように、様々な管理者を作成する機能があります。このようにセキュリティ機能を分離することによって、より安全なエンタープライズ環境が実現します。

関連項目：

- 『Oracle9i Directory Service 統合および配置ガイド』
- [第 17 章「Oracle Wallet Manager の使用方法」](#)
- Microsoft Active Directory をエンタープライズ・ユーザー・セキュリティの LDAP ディレクトリとして使用する方法は、[付録 E「Microsoft Active Directory でのエンタープライズ・ユーザー・セキュリティの使用」](#)を参照してください。なお、Active Directory は、Oracle8i の機能に対してのみサポートされます。

前提条件 C: ディレクトリ使用構成の完了

Oracle Net Configuration Assistant を使用して Oracle ホームに対するディレクトリ・アクセスを設定し、Database Configuration Assistant または Oracle Enterprise Security Manager を使用してデータベースをディレクトリに登録します。

データベースのディレクトリへの登録 データベースがディレクトリに正しく登録されると、次の変更が発生します。

- 新規データベース・サービス・オブジェクトが作成され、指定の Oracle コンテキストの下にあるディレクトリに DN が割り当てられます。さらに、データベースはデフォルトのエンタープライズ・ドメインのメンバーとなります。
- 新しく作成されたデータベース DN が、RDBMS_SERVER_DN 初期化パラメータの値として、データベースの spfile.ora ファイルに追加されます。

Oracle Enterprise Security Manager または Database Configuration Assistant を使用して、データベースをディレクトリに登録できます。ただし、それぞれのツールは異なる設定で自動構成を実行します。[表 15-3](#) に、これらの相違をまとめます。

表 15-3 Oracle Enterprise Security Manager または Database Configuration Assistant によるデータベースのディレクトリへの登録における相違点

Oracle のツール製品	ディレクトリ へのデータ ベース DN エントリの 作成	デフォルト・ ドメインへの データベース の追加	ディレクトリ へのブレース ホルダ・ データベース Wallet の作成	RDBMS_ SERVER_DN パラメータの 設定	有効な データベース Wallet の 作成
Oracle Enterprise Security Manager	○	○	○	×	×
Database Configuration Assistant	○	○	×	○	×

注意： いずれか一方のツールを使用してデータベースを登録できますが、一方のツールを使用してデータベースを途中まで登録し、もう一方のツールを使用して登録処理を完了することはできません。たとえば、Oracle Enterprise Security Manager を使用してデータベースを登録する場合は、Database Configuration Assistant を使用して同じデータベースを登録したり、spfile.ora ファイルの RDBMS_SERVER_DN パラメータを設定することはできません。

Database Configuration Assistant によるデータベースのディレクトリへの登録 このツールを使用する場合は、Oracle Enterprise Security Manager で DBA を Database Registration Admins グループに追加することによって、ディレクトリへのアクセス権を個々のローカル DBA に付与する必要があります。

関連項目： Database Configuration Assistant を使用してデータベースをディレクトリに登録する方法は、『Oracle9i Directory Service 統合および配置ガイド』を参照してください。

Oracle Enterprise Security Manager によるデータベースのディレクトリへの登録 このツールを使用すると、ディレクトリへのアクセス権を個々のローカル DBA に付与することなく、集中化された位置からデータベースをディレクトリに登録できます。Oracle Enterprise Security Manager を使用してデータベースをディレクトリに登録する場合は、登録後に次の作業が必要です。

- ALTER SYSTEM コマンドを使用して、データベースの spfile.ora ファイルに RDBMS_SERVER_DN パラメータを設定します。
- ディレクトリに登録されている各データベースの証明書を格納および管理するために、Oracle Wallet Manager でデータベース Wallet を作成します。

関連項目： Oracle Enterprise Security Manager を使用してデータベースをディレクトリに登録する方法は、[第 19 章「Oracle Enterprise Security Manager の使用方法」](#)を参照してください。

注意：

- spfile.ora ファイルの RDBMS_SERVER_DN パラメータは、ディレクトリの実際の DN と一致する必要があります。
 - ディレクトリ管理ツールを使用して、ディレクトリの Oracle コンテキストの下にデータベース・エントリとサブツリーが存在していることを確認してください。
 - エラー・メッセージが表示された場合は、Oracle Enterprise Security Manager を使用して、資格証明を提供したユーザーが Database Registration グループに登録されていることをチェックしてください。
 - Oracle Enterprise Security Manager を使用してデータベースを登録する場合は、データベースの SID とデータベース短縮名が等しいことが必要です。等しくない場合は、Database Configuration Assistant を使用してデータベースをディレクトリに登録してください。
-
-

タスク 1: SSL 用のデータベースの構成

SSL 用のデータベースを構成する手順は次のとおりです。

- [手順 1: リスナーおよびデータベースで SSL をサポートするために Oracle Net を構成](#)
- [手順 2: SSL サービス名の構成](#)
- [手順 3: リスナーの構成](#)
- [手順 4: .ORA ファイルの確認](#)

手順 1: リスナーおよびデータベースで SSL をサポートするために Oracle Net を構成

[リスナー](#)とデータベースの両方で、SSL を使用できるように Oracle Net を構成します。リスナーは、SSL 付き TCP/IP プロトコル用に構成されているリスニング・エンドポイントを持っている必要があり、データベースの Wallet の位置を指定する必要があります。Oracle Net Manager を使用して、これを実行します (7-12 ページの「[SSL を使用可能にする](#)」を参照してください)。

1. Oracle Net Manager を起動します。
2. プロファイルを構成します。
 - 「SSL」タブで、「Server」ボタンを選択します。
 - 「Wallet Directory」フィールドにデータベース用の Wallet の位置を入力します。
 - 「File」→「Save the network configuration」を選択します。sqlnet.ora ファイルが更新されます。

注意： 必要な場合は、データベースのデフォルト以外の位置を選択できます。

お薦めするデータベースの Wallet の位置：

- Windows の場合 : <userprofile>\ORACLE\WALLETS
 - UNIX の場合 : /etc/ORACLE/WALLETS/DATABASES/database_name
-
-

手順 2: SSL サービス名の構成

Oracle Net Manager を使用して SSL サービス名を構成します。

この接続をテストするかどうか尋ねられたときに、テストは実行しないでください。1) SSL 接続をリスニングするリスナーを設定していないこと、および 2) Database Configuration Assistant を使用してデータベースをディレクトリに登録した場合は、データベース用の Wallet を設定していないことから、接続テストは失敗となります。

関連項目： Oracle Net Manager を使用して SSL サービス名を構成（手順 2）およびリスナーを構成（手順 3）する方法は、7-12 ページの「[SSL を使用可能にする](#)」を参照してください。

手順 3: リスナーの構成

Oracle Net Manager を使用して、SSL リスニング・エンドポイントを保持する [リスナー](#) を構成します。

注意：

- SSL 接続用の Wallet を設定するまでリスナーを起動しないでください。
 - listener.ora の SSL_CLIENT_AUTHENTICATION の値は変更しないでください。この値は FALSE であることが必要です。リスナーは認証を行いません。データベースが SSL を使用してクライアントを認証します。
-
-

手順 4: .ORA ファイルの確認

.ora ファイルを簡単に確認できるように、Windows NT の例をいくつか示します。

例 : SQLNET.ORA ファイル

```
NAMES.DEFAULT_DOMAIN = WORLD
WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = C:\WINNT\Profiles\DATABASES\oe)
    )
  )

SQLNET_AUTHENTICATION_SERVICES = (TCPS,NTS)
SSL_CLIENT_AUTHENTICATION = TRUE

SSL_VERSION = 0
```

注意： Wallet 位置は、Oracle Net Manager でデータベース用に入力した値と一致します。

例 : TNSNAMES.ORA ファイル

```
OESSL.WORLD =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCPS) (HOST = host1) (PORT = 5000))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = finance)
    )
    (SECURITY = (AUTHENTICATION_SERVICE = TCPS)
      (SSL_SERVER_CERT_DN="cn=finance,cn=OracleContext,o=Oracle,c=us")
    )
  )

OE.WORLD =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = host1) (PORT = 1521))
    )
  )
```

```
(CONNECT_DATA =  
  (SERVICE_NAME = oe.world)  
)  
)
```

例 : LISTENER.ORA ファイル

```
WALLET_LOCATION =  
  (SOURCE =  
    (METHOD = FILE)  
    (METHOD_DATA =  
      (DIRECTORY = C:\WINNT\Profiles\DATABASES\oe)  
    )  
  )  
  
LISTENER =  
  (DESCRIPTION_LIST =  
    (DESCRIPTION =  
      (ADDRESS = (PROTOCOL = TCP) (host = HOST1) (port = 1521))  
    )  
    (DESCRIPTION =  
      (ADDRESS = (PROTOCOL = TCPS) (HOST = host1) (PORT = 5000))  
    )  
  )  
  
SID_LIST_LISTENER =  
  (SID_LIST =  
    (SID_DESC =  
      (GLOBAL_DBNAME = oe.world)  
      (ORACLE_HOME = D:\Oracle\Ora81)  
      (SID_NAME = oe)  
    )  
  )  
  
SSL_CLIENT_AUTHENTICATION = FALSE
```

タスク 2: Wallet の作成とリスナーの起動

データベース Wallet を作成して構成する手順は、次のとおりです。

- [手順 1: データベース用の Wallet の作成](#)
- [手順 2: 自動ログインを使用可能に設定](#)
- [手順 3: リスナーの起動](#)
- [手順 4: セキュリティのためのデータベース・ログアウトの実行 \(オプション\)](#)

手順 1: データベース用の Wallet の作成

データベース用の Wallet を作成します。

関連項目： データベース用の Wallet の作成方法は、17-10 ページの「[Wallet の管理](#)」を参照してください。

「Wallet」メニューから「New」を選択したときに、デフォルトのディレクトリを新規に作成するかどうかを尋ねられても、作成しないでください。これはユーザー Wallet 用です。証明書要求の作成時に、データベースの**識別名**を正確に入力します。

```
cn=simple_database_name, cn=OracleContext,<location of Oraclecontext>
```

この DN は初期化パラメータ・ファイルの次のパラメータにあります。

```
RDBMS_SERVER_DN
```

注意： 識別名は大 / 小文字が区別されます。

例：

インストール時に選択したグローバル・データベース名が `sales.us.nmt.com` で、Oracle Net Configuration Assistant で Oracle コンテキストに対して選択した位置が `o=nmt` の場合、Oracle Wallet Manager に入力する完全な DN は次のようになります。

```
cn=sales,cn=OracleContext,o=nmt
```

注意： `cn=OracleContext` は、DN の単純データベース名の直後に入れる必要があります。

手順 2: 自動ログインを使用可能に設定

データベースが Oracle Internet Directory と安全に通信するには、データベース Wallet に対して自動ログインが使用可能であることが必要です。ユーザーが SSL で認証される場合は、データベース・リスナーが起動されている必要があります。リスナーは、データベースの自動ログインが使用可能な場合に作成される `cwallet.sso` ファイルを読み込みます。データベースの自動ログインを使用可能にする手順は、次のとおりです。

1. Oracle Wallet Manager を使用します。
2. 確認のために、Wallet ディレクトリに `cwallet.sso` ファイルが存在していることをチェックしてください。
3. リスナーを停止します。

リスナーを停止するには、コマンドラインで次のように入力します。

- Windows の場合 :`lsnrctl stop`
- UNIX の場合 :`lsnrctl stop`

注意 :

- 管理者は、Oracle Wallet Manager を使用してデータベース Wallet の自動ログインを使用可能にする必要があります。Oracle Enterprise Login Assistant は使用できません。
 - エンド・ユーザーは、Oracle Enterprise Login Assistant を使用して自動ログインを使用可能にできるため、Oracle Wallet Manager を使用する必要はありません。
-

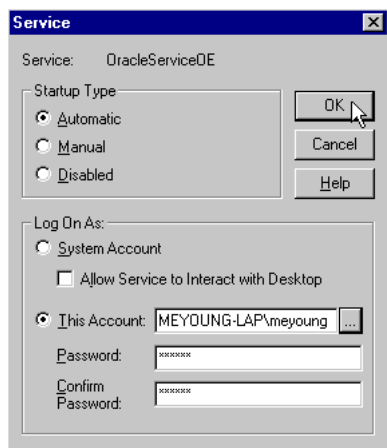
4. Oracle サービス・ログインを変更します (Windows のみ)。データベースとリスナー・サービスはシステム (NT におけるいくつかの権限を所有) として実行され、Wallet はユーザー名でオープンされるため、データベースとリスナーは Wallet を読み込むことができません。データベースとリスナーが Wallet を読み込めるようにするには、データベース Wallet に対する自動ログインが使用可能に設定されているユーザーとしてログインするように変更する必要があります。

Oracle サービス・ログインを変更する手順は次のとおりです。

- 「コントロール パネル」の「サービス」を選択し、「OracleService <database name>」を選択してデータベースをシャットダウンします。「停止」ボタンをクリックし、「はい」を選択して確定します。
- 「スタートアップ」ボタンをクリックします。
- 「サービス」ダイアログ・ボックスの「ログオン」で現在ログオンしているアカウントを選択し、データベース Wallet に対する自動ログインが使用可能に設定されているユーザーの <domain>¥<NT user login> を入力します。

または、ブラウズ・ボタン（「...」）をクリックして、リストから選択することもできます。「パスワード」フィールドと「パスワードの確認入力」ボックスにパスワードを入力し、「OK」をクリックします。15-38 ページの図 15-5 は、Oracle の「サービス」ダイアログ・ボックスを示しています。

図 15-5 「OracleService」ダイアログ・ボックス



- 「開始」をクリックして、Oracle データベースを起動します。
- 「スタートアップ」ボタンのクリックからここまでの手順を「Oracle-
<ORACLE_HOME_NAME>」→「TNSListener」について繰り返します。リスナー・サービスは開始しないでください。
- 「サービス」ダイアログ・ボックスをクローズします。

手順 3: リスナーの起動

注意：

- リスナーが正しく起動されると、コマンドラインで、TCPS 上でリスニングしていることが確認されます。
 - リスナーの起動時にエラーが発生した場合は、自動ログインが選択されていないか、または **Wallet** の位置が正しくない可能性があります。
 - Windows の場合は、「コントロール パネル」の「Service」から **OracleListener Service** を実行して、リスナーを起動および停止することもできます。ただし、リスナーのアクティビティを確認するコマンドラインの応答は表示されません。
-
-

これでデータベース **Wallet** がオープンしたので、データベースは SSL を使用した認証通信に参加できます。Windows では、**OracleTNSListener** サービスも起動されています。

関連項目： **Wallet** の作成方法の詳細は、[第 17 章「Oracle Wallet Manager の使用方法」](#)を参照してください。

手順 4: セキュリティのためのデータベース・ログアウトの実行（オプション）

データベースを長時間停止する場合は、セキュリティのためにデータベース **Wallet** を使用禁止（**Oracle Wallet Manager** で自動ログインを使用禁止）にしてください。

タスク 3: データベースのインストールの検証

データベースが正常に構成されていることを検証する手順は、次のとおりです。

1. データベース **Wallet** ディレクトリに **cwallet.sso** ファイルがあるか検証します。ファイルがない場合は、自動ログインは正常に使用可能に設定されていません。このような事態が発生した場合は、**Oracle Wallet Manager** に戻って **Wallet** をオープンし、「Autologin」チェック・ボックスを選択します。
2. **ldap.ora** ファイルが次の場所にあるか検証します。

`$ORACLE_HOME/network/admin`

ldap.ora ファイルが存在しない場合は、**Oracle Net Configuration Assistant** によってディレクトリのアクセスが正常に構成できていません。**ORACLE_HOME** が設定されていることを確認して、**Oracle Net Configuration Assistant** を再実行してください。

3. ディレクトリ管理ツールを使用して、Oracle Net Configuration Assistant の実行時に指定した Oracle コンテキストの下にデータベース・エントリとサブツリーがあることを確認します。データベース・エントリが見つからない場合は、ディレクトリが実行されているか、Oracle コンテキストが設定されているか、また、ldap.ora ファイルがあり、それが正しいものであるかを検証します。次に、Database Configuration Assistant または Oracle Enterprise Security Manager を使用して再びデータベースを登録します。

タスク 4: グローバル・スキーマとグローバル・ロールの作成

このタスクは Oracle Enterprise Manager を使用して実行できますが、次の例では SQL*Plus を直接使用しています。

グローバル・スキーマとグローバル・ロールを作成する手順は次のとおりです。

- 手順 1: グローバル・スキーマの作成
- 手順 2: セッションの作成権限の付与
- 手順 3: グローバル・ロールの作成
- 手順 4: 権限の関連付け

手順 1: グローバル・スキーマの作成

SQL*Plus を使用して、エンタープライズ・ユーザーの共有スキーマを作成します。たとえば、guest という共有スキーマを作成するには、次のように入力します。

```
CREATE USER guest IDENTIFIED GLOBALLY AS ''
```

行の最後の 2 つの一重引用符の間に空白がないことに注意してください。引用符の間に特定の **識別名** を入力すると、そのユーザーのみがそのスキーマに接続でき、スキーマは共有されません。

手順 2: セッションの作成権限の付与

このスキーマに接続するユーザーには、CREATE SESSION 権限が必要です。CREATE SESSION 権限は、グローバル・スキーマに付与するか、またはエンタープライズ・ロールを介して特定のユーザーに付与される **グローバル・ロール** に対して付与できます。

手順 3: グローバル・ロールの作成

関連する権限を保持するデータベースのグローバル・ロールを作成します。これらのロールは、後で作成されるエンタープライズ・ロールと関連付けられます。エンタープライズ・ロールはユーザーに割り当てられます。

例：

```
CREATE ROLE emprole IDENTIFIED GLOBALLY;  
CREATE ROLE custrole IDENTIFIED GLOBALLY;
```

手順 4: 権限の関連付け

新規グローバル・ロールに権限を関連付けます。

例 :

```
GRANT select ON products TO custrole, emprole;
```

注意： Oracle Advanced Security は、SSL 認証またはパスワード認証（あるいはその両方）を使用してエンタープライズ・ユーザーを認証するように構成できます。

- SSL 認証を構成するには、「[第 III 部 : SSL 認証の最終構成](#)」に進んでください。
 - パスワード認証を構成する場合は、15-49 ページの「[第 IV 部 : パスワード認証の最終構成](#)」に進んでください。
-
-

第 III 部 : SSL 認証の最終構成

この項では、エンタープライズ・ユーザー・セキュリティで SSL 認証を使用するためのインストールおよび構成を完了する最終手順について説明します。必要なタスクは、次のとおりです（各タスクの番号は第 II 部からの連番）。

- [タスク 5: データベース・クライアントの構成](#)
- [タスク 6: エンタープライズ・ドメインの構成](#)
- [タスク 7: エンタープライズ・ユーザーの構成](#)
- [タスク 8: エンタープライズ・ユーザーとしてのログイン](#)

注意： エンタープライズ・ユーザー・セキュリティの構成タスクには連番が付けられており、この連番は第 II 部から続いています。第 III 部では、タスク 5～8 について説明します。

- タスク 1～4 は、15-27 ページの「[第 II 部 : SSL 認証とパスワード認証の初期構成](#)」を参照してください。
 - タスク 9～12 は、15-49 ページの「[第 IV 部 : パスワード認証の最終構成](#)」を参照してください。
-
-

タスク 5: データベース・クライアントの構成

Oracle9i クライアントをインストールした後、Oracle Net Manager を使用して、クライアントの Oracle Net を構成します。この操作は、Oracle9i リリース 2 (9.2) のインストール時またはインストール後に実行できます。

LDAP ディレクトリ・サービスを使用してエンタープライズ・セキュリティを構成しようとしているので、Oracle Net ディレクトリ・ネーミングを使用することもできます。Oracle Net ディレクトリ・ネーミングを使用すると、クライアントは、Database Configuration Assistant によってディレクトリに登録されたデータベース・エントリを使用してデータベースに接続できるようになります。または、ローカル・ネーミング (tnsnames.ora ファイル) など、他の Oracle Net ネーミング・メソッドを使用して、データベースのネット・サービス名を構成することもできます。

データベース・クライアントを構成する手順は、次のとおりです。

1. Oracle Net Manager を使用して SSL ネット・サービス名を構成する方法は、7-12 ページの「[SSL を使用可能にする](#)」を参照してください。
2. クライアント・プロファイルを構成します。クライアント・プロファイルの構成時に Wallet の位置は入力しないでください。Wallet 位置が指定されていない場合、SSL は現行オペレーティング・システム・ユーザーのデフォルト Wallet を検索します。このように、sqlnet.ora ファイルをエンタープライズ・ユーザーで共有でき、管理と配置が簡略化されます。デフォルト以外の Wallet 位置を使用するユーザーは、そのユーザーの Wallet 位置を含む別の sqlnet.ora ファイルを持つ必要があります。

注意： クライアントをインストールしておらず、ORACLE_HOME をデータベース・サーバーの ORACLE_HOME に設定し、その ORACLE_HOME に Wallet 位置を含む sqlnet.ora ファイルが存在する場合は、新しい TNS_ADMIN ディレクトリを 1 つ以上作成し、そこに Wallet 位置を含まない sqlnet.ora ファイルを配置する必要があります。これによって、オペレーティング・システム・ユーザーに対してその SSL で Wallet のデフォルト位置が使用されるようになります。

ユーザー Wallet のデフォルト Wallet 位置

■ **Windows の場合：**

<userprofile>%ORACLE%WALLETS

■ **UNIX の場合：**

/etc/ORACLE/WALLETS/<operating_system_username>

注意： エンタープライズ・ユーザーを作成するときに、特定のユーザー用の Wallet が設定されます。エンタープライズ・ユーザーの作成方法は、[第 19 章「Oracle Enterprise Security Manager の使用方法」](#)を参照してください。

関連項目：

- 『Oracle9i Net Services 管理者ガイド』
- SSL の構成方法は、[第 7 章「Secure Sockets Layer 認証の構成」](#)を参照してください。

タスク 6: エンタープライズ・ドメインの構成

Oracle Enterprise Security Manager は、Oracle9i のインストール時に自動的にインストールされ、エンタープライズ・ドメインの構成に使用します。Oracle デフォルト・ドメインは Oracle コンテキストがディレクトリに作成されるときにデフォルトで作成され、データベースは、Database Configuration Assistant または Oracle Enterprise Security Manager によって登録されるときにそのドメインのメンバーとして自動的に追加されます。[表 15-4](#) は、エンタープライズ・ドメインの設定に必要な手順と関連情報の参照箇所のリストです。Oracle デフォルト・ドメインを使用する場合は、手順 1 と 4 をスキップできます。

表 15-4 エンタープライズ・ドメインの設定

手順	関連情報
1. エンタープライズ・ドメイン を作成します。	19-31 ページ「 エンタープライズ・ドメインの管理 」
2. メンバー・データベース間のカレント・ユーザー・データベース・リンクを使用可能または使用禁止にします。	19-31 ページ「 エンタープライズ・ドメインの管理 」
3. ドメインの認証タイプを選択します。「SSL Only」または「Password and SSL」のいずれかを選択する必要があります。	19-35 ページ「 エンタープライズ・ドメインで使用するデータベース・セキュリティ・オプションの管理 」
4. 対象のエンタープライズ・ドメインのメンバーとして、データベースを登録します。 ¹	19-32 ページ「 エンタープライズ・ドメイン内のデータベースのメンバーシップの定義 」
5. (オプション) ドメインのドメイン管理者を追加します。	第 19 章「Oracle Enterprise Security Manager の使用方法」
6. ドメイン内でのユーザー / スキーマのマッピングを構成します。また、データベース固有のユーザー / スキーマのマッピングを構成することもできます。	第 19 章「Oracle Enterprise Security Manager の使用方法」

表 15-4 エンタープライズ・ドメインの設定（続き）

手順	関連情報
7. エンタープライズ・ドメインにエンタープライズ・ロールを作成します。	19-38 ページ「 エンタープライズ・ロールの管理 」
8. データベース上にグローバル・ロールを作成します。SQL*Plus コマンドは次のようになります。 CREATE ROLE rolename IDENTIFIED GLOBALLY	<ul style="list-style-type: none">■ 『Oracle9i SQL リファレンス』■ 15-40 ページ「手順 3: グローバル・ロールの作成」
9. 各エンタープライズ・ロールにグローバル・ロールを割り当てます。	19-38 ページ「 エンタープライズ・ロールの管理 」

¹ データベースがメンバーとして同時に所属できるドメインは 1 つのみです。データベースのドメイン・メンバーシップを変更した場合は、データベースを再起動する必要があります。

タスク 7: エンタープライズ・ユーザーの構成

新規エンタープライズ・ユーザーを作成する手順は次のとおりです。

- [手順 1: 新規に作成したエンタープライズ・ユーザーのディレクトリへの追加](#)
- [手順 2: ユーザー用の Wallet の作成](#)
- [手順 3: ユーザーの許可](#)
- [手順 4: ユーザーのスキーマへのマップ](#)

手順 1: 新規に作成したエンタープライズ・ユーザーのディレクトリへの追加

ディレクトリ・ユーザーはすべて、エンタープライズ・ユーザーとなり得ます。次のいずれかのツールを使用して、ユーザーをディレクトリに追加できます。

- Oracle Enterprise Security Manager
- ディレクトリ・サービス用の管理ツール
- 標準 LDAP コマンドライン・ツール

ディレクトリへのユーザーの移入に Oracle Enterprise Security Manager を使用しない場合は、orcluser オブジェクト・クラスをディレクトリのユーザー・エントリに追加する必要があります。

Oracle Enterprise Security Manager を使用して、既存のユーザー・エントリを Oracle で使用できるように準備する場合は、orcluser オブジェクト・クラスを既存のエントリに追加します。

注意： Oracle Enterprise Security Manager または Oracle Enterprise Login Assistant を使用して、ユーザーのデータベース・パスワードをディレクトリに設定する必要があります。ディレクトリ・マネージャ・ツールまたは LDAP コマンドライン・ツールを使用して、`orclpassword` 属性のデータベース・パスワードを正確に設定することはできません。

関連項目：

- Oracle Enterprise Security Manager を使用して、新規に作成したエンタープライズ・ユーザーをディレクトリに追加する方法は、19-6 ページの「[エンタープライズ・ユーザーの新規作成](#)」を参照してください。
- ディレクトリ管理ツールの使用方法の詳細は、使用しているディレクトリ・サービスのマニュアルを参照してください。

手順 2: ユーザー用の Wallet の作成

ユーザー用の Wallet を作成するには、第 17 章「[Oracle Wallet Manager の使用方法](#)」を参照してください。

注意： ユーザー Wallet は、デフォルトのユーザー Wallet の位置またはディレクトリに格納してください（ディレクトリだけに格納する場合は、使用する前にユーザー Wallet をクライアントにダウンロードする必要があります）。

- **Windows の場合：**`<userprofile>%ORACLE%WALLETS`

Windows の場合、デフォルトの Wallet 位置は常にユーザー・プロファイルの下です。ユーザー・プロファイルの位置は、`set userprofile=location` を使用して構成できます。

- **UNIX の場合：**`/etc/ORACLE/WALLETS/<os user>`
-

手順 3: ユーザーの許可

次のいずれかまたは両方を行うことができます。

- ローカル Oracle ロールの許可

Oracle Enterprise Manager または SQL*Plus を使用して、ローカル・ロールと権限をデータベース・スキーマに付与します。この手順はオプションです。

注意： これらのロールおよび権限は、そのスキーマに接続している（またはマップされている）すべてのユーザーに対して有効になるため、共有スキーマには、これらのロールおよび権限を付与しないことをお勧めします。

- エンタープライズ・ロールの許可

Oracle Enterprise Security Manager を使用して、エンタープライズ・ロールをディレクトリ内のエンタープライズ・ユーザーに付与します。

手順 4: ユーザーのスキーマへのマップ

共有スキーマを使用する場合は、Oracle Enterprise Security Manager を使用してユーザーをスキーマにマッピングします。次のマッピング・オプションから選択できます。

- データベース：

1 つのデータベースに適用されます。

- エンタープライズ・ドメイン：

エンタープライズ・ドメイン内のすべてのデータベースに適用されます。

例：

3 人のユーザーから **guest** という名前の共有スキーマへのドメイン・マッピングを作成し、そのドメイン内に複数のデータベースがある場合は、各データベースに、3 人のユーザー全員がアクセスできる共有スキーマ（名前は **guest**）が必要です。この 3 人のユーザーは、**guest** という共有スキーマがないドメイン内のデータベースには接続できません。

または、特定のデータベースの下にマッピングを作成することもできます。この方法を使用すると、マッピングはそのデータベースにのみ適用され、ドメイン内のすべてのデータベースには適用されません。両方の場所にマッピングがある場合は、データベース・マッピングが優先されます。

関連項目 :

- エンタープライズ・ドメインの設定方法は、「[タスク 6: エンタープライズ・ドメインの構成](#)」を参照してください。
- 19-6 ページ「[エンタープライズ・ユーザーの管理](#)」
- 15-21 ページ「[エンタープライズ・ユーザーのスキーマへのマッピング](#)」

タスク 8: エンタープライズ・ユーザーとしてのログイン

エンタープライズ・ユーザーとしてログインする手順は、次のとおりです。

- [手順 1: ユーザー Wallet のダウンロード](#)
- [手順 2: 自動ログインを使用可能に設定](#)
- [手順 3: データベースへの接続](#)

手順 1: ユーザー Wallet のダウンロード

ディレクトリからユーザー Wallet をダウンロードする手順は、次のとおりです。

1. 適切なユーザーでオペレーティング・システムにログインします。
2. Oracle Enterprise Login Assistant を使用して Wallet をダウンロードします。

関連項目 : Oracle Enterprise Login Assistant を使用してディレクトリから Wallet をダウンロードする方法は、18-6 ページの「[LDAP ディレクトリへの接続と新規 Wallet のダウンロード](#)」を参照してください。

手順 2: 自動ログインを使用可能に設定

エンタープライズ・ユーザーは、データベースに接続するために、(タスク 10 で作成した) ユーザー Wallet の自動ログインを使用可能にする必要があります。自動ログインを使用可能にすると、シングル・サインオン・ファイルが生成され、SSL アダプタに対する認証が可能になります。

自動ログインを使用可能にするには、Oracle Enterprise Login Assistant を使用します。

関連項目 : ユーザー Wallet のダウンロードおよび自動ログインを使用可能にする方法は、[第 18 章「Oracle Enterprise Login Assistant の使用方法」](#)を参照してください。

手順 3: データベースへの接続

1. ORACLE_HOME を設定します。

ORACLE_HOME がサーバーの ORACLE_HOME に設定されている場合は、環境変数 TNS_ADMIN によって、15-42 ページの「[タスク 5: データベース・クライアントの構成](#)」で作成したクライアントの sqlnet.ora ファイルが存在するディレクトリを指定する必要があります。

クライアントの ORACLE_HOME を別に設定している場合は、環境変数 TNS_ADMIN を設定する必要はありません。

2. SQL*Plus を起動し、次のように入力します。

```
sqlplus/@connect_identifier
```

connect_identifier は 15-42 ページの「[タスク 5: データベース・クライアントの構成](#)」で設定したネット・サービス名です。

接続が成功した場合は、Connected to:... というメッセージが表示されます。このメッセージによって、接続と設定が正常に行われていることを確認します。エラー・メッセージが表示された場合は、15-49 ページの「[第 V 部: エンタープライズ・ユーザー・セキュリティのトラブルシューティング](#)」を参照してください。

正常に接続された場合は、次のように入力して、ディレクトリから適切なグローバル・ロールが取り出されたことをチェックします。

```
select * from session_roles
```

Oracle Enterprise Login Assistant で「Logout」ボタン ([図 18-2](#)) を選択し、SSL アダプタによる認証を使用禁止にします。

関連項目： Oracle Enterprise Login Assistant の使用方法は、[第 18 章「Oracle Enterprise Login Assistant の使用方法」](#)を参照してください。

注意： これで、SSL 認証を使用するためのエンタープライズ・ユーザー・セキュリティの構成が完了しました。第 IV 部はパスワード認証専用の構成を説明しているため、第 IV 部には進まないでください。

第 IV 部 : パスワード認証の最終構成

この段階では、SSL を使用可能なディレクトリ・サーバーおよびデータベースがすでに存在しています。また、データベース上にグローバル・スキーマおよびロールがすでに作成されています。

この項では、エンタープライズ・ユーザーのパスワード認証の設定方法について説明します。必要なタスクは、次のとおりです（各タスクの番号は第 III 部からの連番です）。

- [タスク 9: エンタープライズ・ドメインの構成](#)
- [タスク 10: Oracle コンテキストの構成](#)
- [タスク 11: エンタープライズ・ユーザーの構成](#)
- [タスク 12: パスワード認証を受けたエンタープライズ・ユーザーでの接続](#)

注意： エンタープライズ・ユーザー・セキュリティの構成タスクには連番が付けられており、この連番は第 III 部から続いています。第 IV 部では、タスク 9～12 について説明します。

- タスク 1～4 は、15-27 ページの「[第 II 部 : SSL 認証とパスワード認証の初期構成](#)」を参照してください。
 - タスク 5～8 は、15-41 ページの「[第 III 部 : SSL 認証の最終構成](#)」を参照してください。
 - エンタープライズ・ユーザーの SSL 認証を構成する場合は、第 II 部と第 III 部のタスクのみを実行してください。エンタープライズ・ユーザーのパスワード認証を構成する場合は、第 II 部と第 IV 部のタスクのみを実行してください。
-

タスク 9: エンタープライズ・ドメインの構成

パスワード認証を使用するエンタープライズ・ドメインを構成します。

Oracle Enterprise Security Manager を使用して、エンタープライズ・ドメイン、ユーザー、ロールおよびデータベースを管理します。Oracle Enterprise Security Manager は、エンタープライズ・ドメインの構成に使用できます。Oracle デフォルト・ドメインはディレクトリに Oracle コンテキストを作成するときにデフォルトで作成されます。Database Configuration Assistant または Oracle Enterprise Security Manager を使用してデータベースを登録すると、そのデータベースがデフォルト・ドメインのメンバーとして自動的に追加されます。

表 15-5 は、エンタープライズ・ドメインの設定に必要な手順と関連情報の相互参照を示しています。Oracle デフォルト・ドメインを使用する場合は、手順 1 と 4 をスキップできます。

表 15-5 エンタープライズ・ドメインの設定

手順	関連情報
1. エンタープライズ・ドメインを作成します。	19-31 ページ「 エンタープライズ・ドメインの管理 」
2. メンバー・データベース間のカレント・ユーザー・データベース・リンクを使用可能または使用禁止にします。	19-31 ページ「 エンタープライズ・ドメインの管理 」
3. 「Enterprise Domain Administration」タブを選択し、「Enterprise User Authentication」ドロップダウン・リストから「Oracle Wallet (SSL) And Password」を選択します。	19-25 ページ「 パスワード・アクセシブル・ドメインの管理 」
4. Oracle Enterprise Security Manager を使用して、データベースを希望するエンタープライズ・ドメインのメンバーにします。	19-32 ページ「 エンタープライズ・ドメイン内のデータベースのメンバーシップの定義 」
5. (オプション) ドメインのドメイン管理者を追加します。	第 19 章「 Oracle Enterprise Security Manager の使用方法 」
6. ドメインのユーザーとスキーマのマッピングを構成します。または、データベース固有のマッピングを構成することもできます。	15-9 ページ「 ユーザー / スキーマのマッピング 」
7. エンタープライズ・ドメインにエンタープライズ・ロールを作成します。	19-38 ページ「 エンタープライズ・ロールの管理 」
8. データベース上にグローバル・ロールを作成します。SQL*Plus コマンドは次のようになります。 CREATE ROLE rolename IDENTIFIED GLOBALLY	<ul style="list-style-type: none">■ 『Oracle9i SQL リファレンス』■ 15-40 ページ「手順 3: グローバル・ロールの作成」

表 15-5 エンタープライズ・ドメインの設定 (続き)

手順	関連情報
9. 各エンタープライズ・ロールにグローバル・ロールを割り当てます。	19-38 ページ「 エンタープライズ・ロールの管理 」

タスク 10: Oracle コンテキストの構成

- [手順 1: ユーザー検索ベースの構成](#)
- [手順 2: データベース・アクセスを可能にする](#)
- [手順 3: UserID 属性の構成](#)
- [手順 4: 管理者の構成](#)
- [手順 5: パスワード・アクセシブル・ドメインの構成](#)

手順 1: ユーザー検索ベースの構成

「Oracle Context Properties」ウィンドウの「General」タブで、データベースがユーザー・エントリを検索するためのユーザー検索ベースを追加します。ユーザー検索ベースは、ディレクトリ内でエンタープライズ・ユーザーのエントリが格納されているサブツリーのルートです。

注意： ユーザー検索ベースを入力すると、Oracle Enterprise Security Manager は、適切なデータベースがユーザーのログイン資格証明を読み込めるように、ユーザー検索ベースのサブツリーに対するアクセス権の付与を試みます。

現行のエンタープライズ管理者（Oracle Enterprise Security Manager のユーザー）に、ディレクトリ内でユーザー検索ベース・エントリのアクセス制御リスト（ACL）を変更する権限がない場合、Oracle Enterprise Security Manager は、ユーザー検索ベースを設定するこの段階でエラー・メッセージを表示します。このエラーが発生した場合は、適切なディレクトリ権限を持つエンタープライズ管理者が Oracle Enterprise Security Manager を使用して、このユーザー検索ベースに対するデータベース・アクセスを使用可能にする必要があります。15-52 ページの「[手順 2: データベース・アクセスを可能にする](#)」を参照してください。

手順 2: データベース・アクセスを可能にする

ユーザー・エントリは、すでに Oracle データベースへのアクセスが可能であるユーザーのディレクトリ・サブツリー内に存在する必要があります。選択したサブツリーに対して、Oracle データベースへのアクセス権を設定できます。これにより、特定の Oracle コンテキストのパスワード・アクセシブル・ドメイン・グループに属するドメイン内のデータベースがユーザーのログイン資格証明を読み込むことができるようになります。

データベースへのアクセスを可能にする手順は、次のとおりです。

Oracle コンテキストの下で、選択したディレクトリ・ユーザーのサブツリーでユーザー検索ベースを選択して、Oracle データベースへのアクセス権を設定し、パスワード・アクセシブル・ドメイン・グループに属するデータベースがユーザーのデータベース・ログイン資格証明にアクセスできるようにします。

- 「Users, by Search Base」の下にあるターゲット・ユーザーのサブツリーを選択します。
- 「Enable Database Access」タブを選択します。
- そのサブツリーの「Enable Logon to Authorized Enterprise Domains」を選択します。

手順 3: UserID 属性の構成

「Root Oracle Context Properties」ウィンドウの「General」タブを選択します。「Context Attribute Settings」で、各エンタープライズ・ユーザーを一意に識別する UserID (ニックネーム) を保持するユーザー・エントリ属性の名前を入力します。

ユーザーのニックネーム (UserID) をデフォルトの cn 属性に格納しない場合は、ルートの Oracle コンテキストのディレクトリ全体に対する UserID を構成する必要があります。

例:

- 組織内のエンタープライズ・ユーザーがすべて、各自の employeeid で一意に識別できるとします。
 - employeeid の値が eid という属性に格納されている場合は、「UserID」フィールドに eid と入力します。
-

手順 4: 管理者の構成

「Root Oracle Context Properties」ウィンドウの「Administrators」タブを選択します。管理者をまだ設定していない場合は、この Oracle コンテキストに必要な管理者を設定します。

- コンテキスト管理者は、対象となる Oracle コンテキストに対する権限をすべて持っています。
- データベース・セキュリティ管理者は、エンタープライズ・ドメインとエンタープライズ・ロールを作成および削除し、データベースをドメインに割り当てることができます。
- ユーザー・セキュリティ管理者は、パスワードの設定やパスワード・ヒントの参照などのアクションによって、ディレクトリ内のユーザー・エントリのセキュリティを管理します。

手順 5: パスワード・アクセシブル・ドメインの構成

パスワード認証接続を受け入れるには、データベースがパスワード・アクセシブル・ドメイン・グループ内のドメインに所属している必要があります。

選択した Oracle9i Oracle コンテキストで、データベースをパスワード・アクセシブル・ドメイン・グループに配置します。「Add」ボタンをクリックし、表示されたダイアログから現行のエンタープライズ・ドメインの 1 つを選択します。グループからエンタープライズ・ドメインを削除するには、「Accessible Domains」ウィンドウで対象のドメインを選択して、「Remove」ボタンをクリックします。

関連項目 :

- Oracle Enterprise Security Manager を使用してパスワード・アクセシブル・ドメインを構成する方法は、19-25 ページの「[パスワード・アクセシブル・ドメインの管理](#)」を参照してください。
- 15-12 ページ「[ユーザーのデータベース・ログイン情報のセキュリティ](#)」

タスク 11: エンタープライズ・ユーザーの構成

- [手順 1: エンタープライズ・ユーザーの作成](#)
- [手順 2: ユーザーの許可](#)
- [手順 3: エンタープライズ・ユーザー ID の作成](#)
- [手順 4: エンタープライズ・ユーザーのパスワードの作成](#)

手順 1: エンタープライズ・ユーザーの作成

ディレクトリ・ユーザーはすべて、エンタープライズ・ユーザーとなり得ます。次のいずれかのツールを使用して、ユーザーをディレクトリに追加できます。

- Oracle Enterprise Security Manager
- ディレクトリ・サービス用の管理ツール
- 標準 LDAP コマンドライン・ツール

注意 :

- Oracle Enterprise Security Manager を使用する前に、ディレクトリにユーザーを移入する場合は、`orcluser objectclass` をそのユーザー・エントリに追加する必要があります。
 - Oracle Enterprise Security Manager または Oracle Enterprise Login Assistant を使用して、ユーザーのデータベース・パスワードをディレクトリに設定する必要があります。ディレクトリ・マネージャ・ツールまたは LDAP コマンドライン・ツールを使用して、`orclpassword` 属性のデータベース・パスワードを正確に設定することはできません。
-
-

関連項目 :

- Oracle Enterprise Security Manager を使用して、新規に作成したエンタープライズ・ユーザーをディレクトリに追加する方法は、19-6 ページの「[エンタープライズ・ユーザーの管理](#)」を参照してください。
- ディレクトリ管理ツールの使用方法の詳細は、使用しているディレクトリ・サービスのマニュアルを参照してください。

手順 2: ユーザーの許可

次のいずれかを行うことができます。

- ローカル Oracle ロールの付与

Oracle Enterprise Manager または SQL*Plus を使用して、ローカル・ロールと権限を、データベース・ユーザーまたはスキーマに付与します。この手順はオプションです。

- エンタープライズ・ロールの付与

Oracle Enterprise Security Manager を使用して、エンタープライズ・ロールをディレクトリ内のエンタープライズ・ユーザーに付与します。ユーザー認証は、ローカルのデータベース・ロールとエンタープライズ・ロールをあわせたものになります。

- ユーザーのスキーマへのマップ

共有スキーマを使用する場合は、Oracle Enterprise Security Manager を使用してユーザーをスキーマにマッピングします。次のマッピング・オプションから選択できます。

- データベース :

1 つのデータベースに適用されます。

- ドメイン :

ドメイン内のすべてのデータベースに適用されます。

例 :

3 人のユーザーから **guest** という名前の共有スキーマへのドメイン・マッピングを作成し、そのドメイン内に複数のデータベースがある場合は、各データベースに、3 人のユーザー全員がアクセスできる共有スキーマ（名前は **guest**）が必要です。この 3 人のユーザーは、**guest** という共有スキーマがないドメイン内のデータベースには接続できません。

または、特定のデータベースの下にマッピングを作成することもできます。この方法を使用すると、マッピングはそのデータベースにのみ適用され、ドメイン内のすべてのデータベースには適用されません。両方の場所にマッピングがある場合は、データベース・マッピングが優先されます。

関連項目 :

- エンタープライズ・ドメインの設定方法は、15-43 ページの「[タスク 6: エンタープライズ・ドメインの構成](#)」を参照してください。
- 19-6 ページ「[エンタープライズ・ユーザーの管理](#)」
- 15-21 ページ「[エンタープライズ・ユーザーのスキーマへのマッピング](#)」

手順 3: エンタープライズ・ユーザー ID の作成

ユーザーごとに、企業内で一意の UserID を定義します。デフォルトの UserID は、LDAP ディレクトリで定義されている共通名 (cn) 属性の値です。

次の条件に従って、UserID を選択します。

- UserID はディレクトリ内で一意であるか、または少なくとも手順 1 で入力したユーザー検索ベースの下の子ツリー内で一意にします。たとえば、UserID 属性として cn を選択し、Scott の cn 属性値が Scott.us の場合は、ユーザー検索ベース・フィールドで指定したユーザーのいずれかに cn=Scott.us を定義することはできません。
- UserID は短くて入力しやすいものにします。UserID の目的は、**識別名**の便利な略称として使用することです。

手順 4: エンタープライズ・ユーザーのパスワードの作成

各エンタープライズ・ユーザーのパスワードを作成するために、「Create User」ウィンドウの「Password」タブを選択します。Oracle Enterprise Security Manager では、対応するパスワード・ペリファイアが自動的に作成されて、ユーザー・エントリの orclPassword 属性に格納されます。

注意： Oracle Enterprise Login Assistant を使用して、いつでもパスワードを変更できます。ただし、ディレクトリ・マネージャ・ツールまたは LDAP コマンドライン・コールを使用して、ユーザーのデータベース・パスワードを設定することはできません。

関連項目： 19-10 ページ「[新規エンタープライズ・ユーザーのパスワードの定義](#)」

タスク 12: パスワード認証を受けたエンタープライズ・ユーザーでの接続

エンタープライズ・ユーザーの UserID が hscortea、パスワードが welcome の場合は、sqlplus で次のコマンドを入力して接続します。

```
SQL>connect hscortea/welcome@<TNS Service Name>
```

データベースは、エンタープライズ・ユーザー（hscortea）を認証するために、このユーザーに対応付けられているディレクトリ・エントリに対してユーザー名 / パスワードの組合せを検証します。検証が成功すると、データベースへの接続が確立されます。

注意： パスワード認証を使用するエンタープライズ・ユーザー・セキュリティの構成はすでに完了しています。

第 V 部：エンタープライズ・ユーザー・セキュリティのトラブルシューティング

この項では、発生する可能性のある問題とその対処方法を次の項目に従って説明します。

- データベースへの接続時の [ORA-# エラー](#)
- ユーザー・スキーマ・エラーのチェックリスト
- DOMAIN-READ-ERROR のチェックリスト
- 暗号化された秘密鍵の復号化に関する障害（Windows のみ）
- トレース機能の有効化

データベースへの接続時の ORA-# エラー

ORA-# エラーを受け取った場合は、[表 15-6](#) でエラーを検索して適切な処置を行ってください。

表 15-6 データベースへの接続時の ORA-# エラー

エラー	処置
ORA-01017: ユーザー名 / パスワードが無効です。ログオンは拒否されました。	15-60 ページの「 ユーザー・スキーマ・エラーのチェックリスト 」を参照してください。
ORA-28271: LDAP ディレクトリ・サービスのユーザー・エントリを読み込む権限がありません。	<ol style="list-style-type: none"> 1. Oracle Enterprise Security Manager を使用して、このユーザーが含まれるユーザー検索ベースが、使用している Oracle コンテキストにリストされていることをチェックします。 2. ユーザー・エントリを公的に読み込めないようにするユーザー定義の ACL が、ディレクトリ・ツリーでユーザーよりも上にないことをチェックします。 次の構文を入力します。 <code>ldapsearch -h <directory_host> -p <directory_port> -b <user_search_base_DN> "objectclass=person"</code> ユーザー・エントリが表示されない場合は、コンテキスト内のすべてに対して、または最低でもデータベース DN またはパスワード・アクセシブル・ドメイン・グループに対して、ユーザー・エントリの読み込みアクセスを許可するようにユーザー検索ベースの ACL を変更します。 3. エンタープライズ・ドメインが、その Oracle コンテキストのパスワード・アクセシブル・ドメイン・グループにあることをチェックします。
ORA-28272: ドメイン・ポリシーがパスワード・ベースの GLOBAL ユーザー認証を制限します。	Oracle Enterprise Security Manager を使用して、このエンタープライズ・ドメインのユーザー認証ポリシーを「Oracle Wallet (SSL) And Password」に設定します。
ORA-28273: LDAP 識別名へのユーザー・ニックネームのマッピングが存在しません。	<ol style="list-style-type: none"> 1. Oracle Enterprise Security Manager を使用して、このユーザーが含まれるユーザー検索ベースが、使用している Oracle コンテキストにリストされていることをチェックします。 2. ユーザーの Oracle Internet Directory にユーザー・エントリがあることをチェックします。 3. ユーザー・エントリに正しいユーザー ID が含まれていることをチェックします。 A. ルート・コンテキストでディレクトリに対して構成されている UserID 属性を検索します。 B. ユーザー・データベース・ログインの試行で指定された名前が、ユーザー・ディレクトリ・エントリのその属性の値であることをチェックします。

表 15-6 データベースへの接続時の ORA-# エラー (続き)

エラー	処置
ORA-28274: ユーザー・ニックネームに対応する ORACLE パスワード属性が存在しません。	<ol style="list-style-type: none"> 1. Oracle Enterprise Security Manager を使用して、このユーザーが含まれるユーザー検索ベースが、使用している Oracle コンテキストにリストされていることをチェックします。 2. Oracle コンテキストの下ユーザー検索ベースに対して、「allow DB access」チェックボックスがチェックされていることを確認します。 3. ディレクトリのユーザー・エントリに orcluser オブジェクト・クラスがあることをチェックします。 orcluser オブジェクト・クラスがない場合は、次の確認を行います。 A. Oracle Enterprise Security Manager を使用してユーザーを作成した場合は、次のことをチェックします。 1. ルートの Oracle コンテキストのバージョンが 90000 を超えていること。 2. ベース・スキーマのバージョン (cn=OracleSchemaVersion の下) が 90000 を超えていること。 B. ユーザー・エントリの作成に Oracle Enterprise Security Manager を使用しなかった場合は、Oracle Enterprise Security Manager または LDAP コマンドライン・ツールを使用して、そのオブジェクト・クラスを追加する必要があります。Oracle Enterprise Security Manager で作成したユーザーには、このオブジェクト・クラスが自動的に取得されます。 orcluser オブジェクト・クラスがある場合は、次の確認を行います。 A. エンタープライズ・ドメインが、その Oracle コンテキストの Password Accessible Domains グループにあることをチェックします。 B. ディレクトリ内のそのユーザーのデータベース・パスワードの値セットが orclpassword 属性に設定されていることをチェックします。
ORA-28275: LDAP 識別名へのユーザー・ニックネームのマッピングが複数存在します。	これは、ユーザー検索ベース内のディレクトリに、データベースへの接続に指定したユーザー・ニックネーム /UserID と一致するユーザー DN が複数存在することを意味します。Oracle Enterprise Security Manager を使用して、ldap.ora ファイルの中でリストされている Oracle コンテキストに関連するすべてのユーザー検索ベース内で、ニックネーム /UserID の値が一意となるように (2 人のユーザーが同じニックネームを共有しないように) してください。
ORA-28277: グローバル・ユーザーのパスワードでの認証中に、LDAP 検索が失敗しました。	ディレクトリの SSL インスタンスが起動され、実行中であることをチェックします。

表 15-6 データベースへの接続時の ORA-# エラー (続き)

エラー	処置
ORA-28278: パスワード・ベースの GLOBAL ユーザーのためのドメイン・ポリシーが登録されていません。	これは、データベースが必要とするエンタープライズ・ドメイン情報が読み込めないことを意味します。15-62 ページの「 DOMAIN-READ-ERROR のチェックリスト 」を参照してください。
ORA-28279: INIT.ORA の rdbms_server_dn パラメータの読み込みでエラーが発生しました。	RDBMS_SERVER_DN パラメータが、spfile.ora に指定されていることをチェックします (init.ora は関係ありません)。指定されている場合は、パラメータを読み込めるようにデータベースを再起動します。パラメータが spfile.ora に指定されていない場合は、データベース登録が失敗したことを意味します。その場合は、Database Configuration Assistant または Oracle Enterprise Security Manager を使用して、データベースをディレクトリに登録します。
ORA-28030: LDAP ディレクトリ・サービスへアクセス中にサーバーに問題が発生しました。	考えられる原因 : <ul style="list-style-type: none"> ■ データベースが複数のドメインに属しています (通常は Oracle Enterprise Security Manager を使用することによって防止できます)。この場合、データベース・ログ・ファイルに警告が出力されます。 ■ データベース Wallet の DN が、spfile.ora の RDBMS_SERVER_DN パラメータに指定されている値と異なります。これらは一致する必要があります。

ユーザー・スキーマ・エラーのチェックリスト

ユーザー・スキーマ・エラーを受け取った場合は、次の確認を行います。

1. SSL ユーザーかどうか

- SSL ユーザーの場合は、次の確認を行って適切な Wallet が使用されていることをチェックします。
 - * クライアントの sqlnet.ora ファイルに WALLET_LOCATION パラメータ値がないこと。
 - * 適切な sqlnet.ora ファイルが使用されるように、TNS_ADMIN パラメータが正しく設定されていること。正しい場合は次へ進みます。
- SSL ユーザーでない場合は次へ進みます。

2. データベースにスキーマをグローバル・ユーザーとして作成したかどうか

- グローバル・ユーザーとして作成した場合は次へ進みます。
- グローバル・ユーザーでない場合は、CREATE USER...IDENTIFIED GLOBALLY 構文を使用して、データベースにグローバル・ユーザー / スキーマを作成します。次へ進みます。

3. データベース・スキーマは、プライベート・スキーマ（共有でない）かどうか（つまり、IDENTIFIED GLOBALLY AS '<full_DN_of_user>' として作成したスキーマであるかどうか）
 - プライベート・スキーマの場合は、ユーザー **Wallet** の DN が、CREATE USER 文で使った DN と一致することを確認します。次の項目は共有スキーマを取り扱うため、ここで終了してください。
 - プライベート・スキーマでない場合は次へ進みます。
4. 関連するユーザー / スキーマのマッピングが、ディレクトリ内のデータベース・エントリの下にあるかどうか（つまり、マッピングは、データベースのエンタープライズ・ドメイン全体に対するものではなく、このデータベースに対して適用されるかどうか）
 - データベース・エントリの下にある場合は、データベースが、データベース自体のエントリおよびディレクトリ内のサブツリーを読み込めることをチェックします。次の構文を使用してチェックします。

```
ldapsearch -h <directory_host> -p <directory_SSLport> -U 3 -W
"file:<database_wallet_path>" -P <database_wallet_password> -b "<database_
DN>" "objectclass=*
```

少なくともデータベース・エントリおよび関連するマッピングを確認してください。ここで終了してください。

- データベース・エントリの下にない場合は次へ進みます。
5. 関連するユーザー / スキーマのマッピングが、ディレクトリ内のドメイン・エントリの下にあるかどうか（つまり、マッピングは、データベースのエンタープライズ・ドメイン全体に対して適用されるかどうか）
 - ドメイン・エントリの下にある場合は、15-62 ページの「[DOMAIN-READ-ERROR のチェックリスト](#)」を参照してください。
 - ドメイン・エントリの下にない場合は、ユーザー / スキーマのマッピングがないことが問題です。Oracle Enterprise Security Manager を使用してマッピングを作成します。または、次の構文を使用してプライベート・スキーマが含まれるようにユーザーを変更します。

```
ALTER USER <username> IDENTIFIED GLOBALLY AS '<full_DN_of_user>'
```

DOMAIN-READ-ERROR のチェックリスト

DOMAIN-READ-ERROR を受け取った場合は、次の確認を行います。

1. データベース Wallet を使用してディレクトリにバインドし、データベースとディレクトリ間の SSL 接続をチェックします。次のコマンドを使用します。

```
ldapbind -h <directory_host> -p <directory_SSLport> -U 3 -W "file:<wallet_location>" -P <wallet_password>
```

このコマンドが失敗した場合は、ディレクトリに実行中の SSL インスタンスがあることを確認します。

関連項目： ディレクトリの SSL インスタンスを管理する方法は、『Oracle Internet Directory 管理者ガイド』を参照してください。

2. spfile.ora ファイルの RDBMS_SERVER_DN パラメータが、データベース Wallet の DN と一致していることをチェックします。等号 (=) の左側に記述する属性識別子は、大 / 小文字の区別がありませんが、等号 (=) の右側に記述する属性値には大 / 小文字の区別があります。たとえば、cn=database1 と CN=database1 は同じです（一致します）が、cn=database1 と cn=DATABASE1 は異なります（不一致となります）。これらが一致しない場合は、データベース Wallet に新しい証明書を取得する必要があります。
3. データベースがエンタープライズ・ドメインのメンバーであることを Oracle Enterprise Security Manager でチェックします。データベースがエンタープライズ・ドメインのメンバーでない場合は、Oracle Enterprise Security Manager を使用して、データベースをエンタープライズ・ドメインに追加します。
4. Oracle Enterprise Security Manager を使用して、このドメインのユーザー認証ポリシーをセットまたはリセットします。
5. 次のコマンドを使用して、データベースがそのドメインを参照できることをチェックします。

```
ldapsearch -h <directory_host> -p <directory_SSLport> -U 3 -W "file:<database_wallet_path>" -P <database_wallet_password> -b "cn=OracleContext, <admin_context>" "objectclass=orclDBEnterpriseDomain"
```

このコマンドが失敗した場合は、データベースを再起動してエンタープライズ・ドメインに関してキャッシュされている値を更新してください。

6. データベースがエンタープライズ・ドメインのサブツリーを読み込むことができ、その結果、エンタープライズ・ロールを読み込めることをチェックします。次のコマンドを使用します。

```
ldapsearch -h <directory_host> -p <directory_SSLport> -U 3 -W "file:<database_wallet_path>" -P <database_wallet_password> -b "cn=OracleContext, <admin_context>" "objectclass=orclDBEnterpriseRole"
```

このドメインに対して作成したエンタープライズ・ロールすべてを参照できるようにしてください。

7. これらすべてのチェックで問題がなくても DOMAIN-READ-ERROR を受け取る場合は、ディレクトリの SSL インスタンスおよびデータベースを再起動します。

暗号化された秘密鍵の復号化に関する障害（Windows のみ）

Windows NT の場合のみ適用されます。

このエラーは、オープンすることを許可されていない Wallet をオープンしようとしたときに発生します。

例：

- user-x でシステムにログインしましたが、
<userprofile>%ORACLE%WALLETS を Wallet 位置として識別するローカルの sqlnet.ora ファイルがありません。
- SSL はデフォルト位置の sqlnet.ora ファイルを使用して Wallet 位置を検索し、データベース用の Wallet をオープンしてユーザーのログイン資格証明を取得しようとします。
- この試みは、user-x にデータベース用の Wallet をオープンする許可が与えられていないため失敗します。

トレース機能の有効化

トレース機能を使用すると、デバッグが容易になります。ldapbind に失敗し、ディレクトリの SSL インスタンスが正常に動作していない場合に適しています。

Oracle Internet Directory

Oracle Internet Directory を LDAP ディレクトリとして使用する場合は、次のトレース方法を使用します。

1. Oracle Internet Directory で、デバッグ・フラグをオンにします。

注意：『Oracle Internet Directory 管理者ガイド』を参照してください。

2. Oracle Internet Directory の SSL インスタンスを完全デバッグ・モードで起動します。
ログ・ファイルは \$ORACLE_HOME/ldap/log に作成されます。ファイル名に SSL ディレクトリのインスタンス番号と s が含まれるファイルを探します。s がないログ・ファイルは、モニター・プロセス (oidmon) とディスパッチャ用です。connect /@connect_identifier を試行した直後に、ログ・ファイルの最後を調べます。文字列 Distinguished Name を探して、自分のユーザー DN と一致していることを確認します。
3. Oracle Internet Directory のトレース機能をオフにします。

ローカルまたは外部ユーザーからエンタープライズ・ユーザーへの移行

この章では、ユーザー移行ユーティリティについて説明します。このユーティリティを使用すると、データベース・ユーザーを LDAP ディレクトリに一括して移行し、エンタープライズ・ユーザーとして格納および管理できます。次の項目について説明します。

- ローカルまたは外部ユーザーからエンタープライズ・ユーザーへの移行による利点
- ユーザー移行ユーティリティの概要
- 移行処理を実行するための前提条件
- ユーザー移行ユーティリティのコマンドライン構文
- ユーザー移行ユーティリティ・ヘルプの表示
- ユーザー移行ユーティリティ・パラメータのリスト
- ユーザー移行ユーティリティの使用例
- ユーザー移行ユーティリティでのトラブルシューティング

ローカルまたは外部ユーザーからエンタープライズ・ユーザーへの移行による利点

データベース・ユーザー・モデルからエンタープライズ・ユーザー・モデルへの移行によって、エンタープライズ環境における管理、セキュリティおよび操作性の問題に対するソリューションが提供されます。エンタープライズ・ユーザー・モデルでは、ユーザー情報はすべて LDAP ディレクトリ・サービスに移動します。

エンタープライズ・ユーザー・セキュリティが提供する次の利点によって、企業全体のユーザーを簡単かつ安全に管理できます。

- ユーザーの資格証明、ロールおよび権限を LDAP バージョン 3 準拠のディレクトリ・サーバーに集中して格納します。
- X.509 v3 準拠の証明書を使用したシングル・サインオンを可能にするインフラストラクチャを提供します（通常は、エンド・トゥ・エンドの SSL が必要な場合に使用されます）。
- セキュリティが向上します。

エンタープライズ・ユーザー・モデルは管理しやすいため、セキュリティ管理者はユーザー情報のメンテナンスに必要な変更を即時に実行し、重要なネットワーク・リソースへのアクセスを制御できます。エンタープライズ・ユーザー・モデルでは、覚えておく必要のあるパスワードの数が少ないため、ユーザーにとっても使いやすいものとなります。この結果、ユーザーが、推測しやすいパスワードを選択したり、他者が簡単に書き写すことができる場所にパスワードを書き留める傾向が減ります。

関連項目： エンタープライズ・ユーザー・セキュリティの概念の詳細は、15-2 ページの「[エンタープライズ・ユーザー・セキュリティの概要](#)」を参照してください。

ユーザー移行ユーティリティの概要

ユーザー移行ユーティリティは、ユーザーのローカル・データベース・モデルからエンタープライズ・ユーザー・モデルへの移行を決定する際に、エンタープライズ・ユーザーの管理者が使用するコマンドライン・ユーティリティです。このユーティリティを使用すると、何千ものローカルと外部データベースのユーザーを LDAP ディレクトリ内のエンタープライズ・ユーザー環境に簡単に移行し、集中管理できます。

エンタープライズ・ユーザーの管理者は、データベース内の次のユーザー・サブセットを移行対象として選択できます。

- コマンドラインまたはファイルで指定したユーザーのリスト
- すべての外部ユーザー
- すべてのグローバル・ユーザー

また、ユーザーの移行方法を決定するためにユーティリティ・パラメータ値を指定することもできます。次に例を示します。

- 移行したユーザーを LDAP ディレクトリ・ツリー内のどこに配置するか
- 各種データベース上に複数のアカウントを持つユーザーを単一のディレクトリ・ユーザー・エントリにマップする
- 移行したユーザーの新規パスワードをランダムに生成する
- **データベース・パスワード・ベリファイア¹** をデータベースからローカル・ユーザーのディレクトリにコピーして、ユーザーが各自のデータベース・パスワードを保持できるようにする

次の各項では、移行処理およびユーザーのスキーマに対して行われる変更について説明します。

注意： 移行された外部ユーザーの外部認証および認可メカニズムは、ディレクトリ・ベースのメカニズムに置換されます。

¹ データベース・パスワード・ベリファイアは、ユーザーのデータベース・パスワードから導出される不可逆的な値です。この値は、データベースに対するパスワード認証時に、接続ユーザーの識別情報を証明するために使用されます。

一括ユーザー移行処理の概要

一括ユーザー移行処理では、2 フェーズの処理を行います。フェーズ 1 では、ユーザー情報をインタフェース・データベース表に移入することによって移行処理を開始します。ここで、エンタープライズ・ユーザーの管理者は、情報が正確であることを検証した後、フェーズ 2 でデータベースとディレクトリに対する変更をコミットできます。この処理について、次の各手順で説明します。

- [手順 1: フェーズ 1 – 移行の準備](#)
- [手順 2: ユーザー情報の検証](#)
- [手順 3: フェーズ 2 – 移行の完了](#)

手順 1: フェーズ 1 – 移行の準備

移行処理の最初のフェーズでは、ユーティリティによって、エンタープライズ・ユーザーの管理者のスキーマに `ORCL_GLOBAL_USR_MIGRATION_DATA` インタフェース表が存在するかどうかチェックされます。インタフェース表が存在する場合、管理者は、その表の再利用（内容は消去）、表とその内容の再利用、または表の再作成を選択できます。このユーティリティのフェーズ 1 を複数回実行すると、その都度インタフェース表への追加を実行できます。表が存在しない場合は、管理者のスキーマに表が作成されます。インタフェース表には、データベースおよびディレクトリからの移行ユーザーに関する情報が移入されます。この表に移入される情報の種類は、使用するコマンドライン・オプションによって決まります。

`EXTERNAL` または `GLOBAL` ユーザーの場合、パスワードはランダムに生成され、インタフェース表に移入されます。`LOCAL` ユーザーの場合は、データベース・パスワード・ベリファイアがデータベースからディレクトリにコピーされます。ユーザー用の既存のディレクトリ・エントリにデータベース・パスワード・ベリファイアがすでに存在している場合は、両方のデータベース・パスワード・ベリファイアが一致したときにユーティリティが正常に実行されます。一致しない場合は、そのユーザーに対するエラーが生成されます。

注意： このユーティリティによって、`SYS` スキーマにインタフェース表が作成されることはありません。

手順 2: ユーザー情報の検証

この中間の手順によって、エンタープライズ・ユーザーの管理者は、データベースとディレクトリに対する変更をコミットする前に、インタフェース表の情報が適切であることを検証できます。

手順 3: フェーズ 2 – 移行の完了

インタフェース表にユーザー情報が移入され、エンタープライズ・ユーザーの管理者が情報を検証した後、フェーズ 2 では、ユーティリティによってインタフェース表の情報が取り出され、ディレクトリとデータベースが更新されます。外部およびグローバル・ユーザーの場合は、ランダムなデータベース・パスワードが生成され、ディレクトリ・エントリ用のランダムなパスワードも生成されます。これらのパスワードは、インタフェース表の DBPASSWORD 列と DIRPASSWORD 列に格納されます。エンタープライズ・ユーザーの管理者はこれらのパスワードをインタフェース表から読み込んで、移行ユーザーに通知できます。

関連項目： コマンドライン・オプションとその説明のリストは、16-13 ページの「[ユーザー移行ユーティリティ・パラメータのリスト](#)」を参照してください。

ORCL_GLOBAL_USR_MIGRATION_DATA 表

この表は、一括ユーザー移行処理のフェーズ 1 で、移行ユーザーに関する情報が移入されるインタフェース表です。この表に移入される情報はデータベースから取り出され、ディレクトリ内の既存のエントリに対してチェックされます。ディレクトリ内に対応する情報がある場合は、表内のそのユーザーに対する情報にマークが付けられます。エンタープライズ・ユーザーの管理者がこの表の情報を検証した後、フェーズ 2 で、ディレクトリとデータベースに対する変更が行われます。

注意： ORCL_GLOBAL_USR_MIGRATION_DATA インタフェース表には、機密情報が含まれています。この表へのアクセスは、データベース権限を使用して厳重に管理してください。

この表の列を[表 16-1](#) にリストします。

表 16-1 ORCL_GLOBAL_USR_MIGRATION_DATA 表のスキーマ

列名	データ型	NULL かどうか	説明
USERNAME (主キー)	VARCHAR2(30)	NOT NULL	データベース・ユーザー名。
OLD_SCHEMA_TYPE	VARCHAR2(10)	-	移行前のデータベース内の元のスキーマ・タイプ。
PASSWORD_VERIFIER	VARCHAR2(30)	-	ローカル・ユーザーの場合は、データベースの データベース・パスワード・ベリファイア が含まれます。
USERDN	VARCHAR2(4000)	-	ディレクトリ内のユーザーの識別名 (DN) (新規または既存)

表 16-1 ORCL_GLOBAL_USR_MIGRATION_DATA 表のスキーマ (続き)

列名	データ型	NULL かどうか	説明
USERDN_EXIST_FLAG	CHAR(1)	-	ディレクトリに DN がすでに存在しているかどうかを示すフラグ。
SHARED_SCHEMA	VARCHAR2(30)	-	共有スキーマ名 (フェーズ 2 で共有スキーマにマップされるユーザーの場合)。
MAPPING_TYPE	VARCHAR2(10)	-	マッピング・タイプ (データベースまたはドメイン)。
MAPPING_LEVEL	VARCHAR2(10)	-	マッピング・レベル (エントリまたはサブツリー)。
CASCADE_FLAG	CHAR(1)	-	ユーザーを削除するときに使用するカスケード・フラグ (共有スキーマ・マッピングの場合のみ)。
DBPASSWORD_EXIST_FLAG	CHAR(1)	-	このユーザー用のデータベース・パスワードがディレクトリにすでに存在しているかどうかを示すフラグ。
DBPASSWORD	VARCHAR2(30)	-	ランダムに生成され、ディレクトリに格納されるデータベース・パスワード・ベリファイア。
DIRPASSWORD	VARCHAR2(30)	-	新規エントリ用にランダムに生成されたディレクトリ・パスワード。
PHASE_COMPLETED	VARCHAR2(10)	-	正常に完了したフェーズに関する情報。
NEEDS_ATTENTION_FLAG	CHAR(1)	-	管理者の確認が必要な異常が、行に含まれているかどうかを示すフラグ。
ATTENTION_DESCRIPTION	VARCHAR2(100)	-	管理者に対する説明的なヒント (ATTENTION フラグが設定されている場合)。

フェーズ1とフェーズ2の間に変更できるインタフェース表の列

このユーティリティのフェーズ1の実行後、エンタープライズ・ユーザーの管理者は、必要に応じて、表 16-2 に示すインタフェース表の列を変更できます。

表 16-2 フェーズ1とフェーズ2の間に変更できるインタフェース表の列の値

列名	有効な値	制限事項
USERDN	ユーザーの DN	この値を変更した場合は、USERDN_EXIST_FLAG と DBPASSWORD_EXIST_FLAG の値が適切に設定されていることを検証してください。
USERDN_EXIST_FLAG	T または F	USERDN 列の値を変更した場合は、この列の値も変更して、新しい USERDN のステータスを反映する必要があります。
DBPASSWORD_EXIST_FLAG	T または F	USERDN 列の値を変更した場合は、この列の値も変更して、新しい USERDN に対するデータベース・パスワードが存在しているかどうかを反映する必要があります。
SHARED_SCHEMA	共有スキーマ名	データベースに共有スキーマが存在している場合のみ指定します。
MAPPING_TYPE	DB または DOMAIN	この値は、SHARED_SCHEMA が NULL 以外に設定されている場合のみ設定します。
MAPPING_LEVEL	ENTRY または SUBTREE	この値は、SHARED_SCHEMA が NULL 以外に設定されている場合のみ設定します。
CASCADE_FLAG	T または F	この値は、SHARED_SCHEMA が NULL 以外に設定されている場合のみ設定します。この列に TRUE (T) を設定した場合、ユーザーのスキーマ・オブジェクトは強制的に削除されます。この列に FALSE (F) を設定した場合は、フェーズ2の実行前に、ユーザー・スキーマ・オブジェクトをすべて削除する必要があります。
PHASE_COMPLETED	ZERO、ONE または TWO	NEEDS_ATTENTION_FLAG で示された競合やあいまい性を解決できた場合は、この列の値を ONE に変更して、ユーティリティでフェーズ2を実行できます。

移行によるユーザーの元のデータベース・スキーマへの影響

共有スキーマ・マッピングを使用しない場合、各ユーザーの元のデータベース・スキーマは保持されます。共有スキーマ・マッピングを使用する場合、ユーザーのローカル・スキーマはデータベースから削除され、移行の実行前にエンタープライズ・ユーザーの管理者が作成した共有スキーマにマップされます。移行ユーザーが元のローカル・データベース・スキーマのデータベース・オブジェクトを所有している場合は、CASCADE パラメータを NO に設定して、そのスキーマとオブジェクトが削除されないように指定できます。CASCADE パラメータを NO に設定すると、元のローカル・スキーマのデータベース・オブジェクトを所有しているユーザーの移行は成功しないため、そのユーザーのオブジェクトは削除されません。

各自のローカル・データベース・スキーマのオブジェクトを保持する必要があるユーザーを共有スキーマにマップする場合は、一括ユーザー移行の実行前に、必要なオブジェクトを共有スキーマに手動で移行できます。ただし、オブジェクトを共有スキーマに移行すると、そのオブジェクトは、新しいスキーマを共有するすべてのユーザー間で共有されます。

表 16-3 に、MAPSCHEMA パラメータと CASCADE パラメータの設定による影響をまとめます。

表 16-3 共有スキーマ・マッピングの選択と CASCADE オプションの指定による影響

MAPSCHEMA パラメータ設定	CASCADE パラメータ設定	ユーザーの移行は 成功するか	ユーザー・スキーマ・ オブジェクトは削除されるか
PRIVATE	NO（デフォルト設定）	はい	いいえ
SHARED	NO	はい ¹	いいえ
SHARED	YES	はい ²	はい

¹ ユーザーの移行は、そのユーザーが元のデータベース・スキーマのオブジェクトを所有していない場合のみ成功し、それ以外の場合は失敗します。

² ユーザーの移行は成功し、そのユーザーの元のデータベース・スキーマは削除されます。

関連項目： MAPSCHEMA、CASCADE およびこのユーティリティで使用可能なその他のパラメータの詳細は、16-13 ページの「[ユーザー移行ユーティリティ・パラメータのリスト](#)」を参照してください。

移行処理

ディレクトリに定義および管理されているエンタープライズ・ユーザーは、パスワードまたは証明書のいずれかを使用してデータベースに対する認証を行うことができます。パスワードで認証を行うユーザーには、ディレクトリに格納された **Oracle** のデータベース・パスワードが必要です。証明書で認証を行うユーザーには、有効な X.509 v3 証明書が必要です。

このユーティリティは、移行時に次の手順を実行します。

1. 移行対象ユーザーをデータベースから選択します。
2. 対応するユーザー・エントリを作成するか、またはディレクトリの既存エントリを使用します。
3. データベース・パスワード・ベリファイアをデータベースから取得するか、または新規パスワードを作成し、それらを移行ユーザー用のディレクトリにコピーします。
4. 移行ユーザーのエントリに関するスキーマ・マッピング情報をディレクトリに配置します（オプション）。
5. 移行ユーザーのローカル・データベース・スキーマを削除または変更します（オプション）。

注意： 現行のリリースでは、このユーティリティは、証明書ベースの認証を使用しているユーザーを移行し、それらのユーザーによるパスワード認証を使用可能にします。以前に SSL ベースで認証されたユーザーは、**Oracle** のデータベース・パスワードを再設定する必要があります。この処理の過程でユーザー **Wallet** は作成されません。

関連項目：

- **Oracle Wallet** の作成方法と管理方法は、[第 17 章「Oracle Wallet Manager の使用方法」](#)を参照してください。
- **Wallet** の使用法は、[第 18 章「Oracle Enterprise Login Assistant の使用法」](#)を参照してください。

移行処理を実行するための前提条件

ユーザー移行ユーティリティは、Oracle9i Client のインストール時に次の位置に自動的にインストールされます。

```
$ORACLE_HOME/rdbms/bin/umu
```

次の各項では、ユーザー移行ユーティリティを使用してユーザーを正常に移行するために、実行する必要があるプログラムと必須のユーザー権限について説明します。

必須のデータベース権限

このユーティリティを正常に使用するには、エンタープライズ・ユーザーの管理者が次のデータベース権限を持っている必要があります。

- ALTER USER
- DROP USER
- CREATE TABLE
- SELECT_CATALOG_ROLE

これらの権限によって、ユーザーの変更、ユーザーの削除、ディクショナリ・ビューの参照、およびこのユーティリティで使用するインタフェース表の作成が可能となります。

必須のディレクトリ権限

必須のデータベース権限に加えて、エンタープライズ・ユーザーの管理者には、次の作業を実行するためのディレクトリ権限が必要です。

- 指定のユーザー・ベースおよび Oracle コンテキストの位置の下にあるディレクトリにエントリを作成する
- 検索ベースの下にあるユーザー・エントリをブラウズする

ユーザー移行ユーティリティの実行に必要な設定

ユーザー移行ユーティリティを使用する前に、次の手順を実行してください。

1. ディレクトリ・サーバーが、認証に対応していない SSL の状態で実行されていることを確認します。
2. データベース・サーバーが、暗号化と整合性が可能な状態で実行されていることを確認します。
3. データベース・リスナーに TCP リスニング・エンドポイントが指定されていることを確認します。
4. Oracle コンテキストをディレクトリに作成します（存在していない場合）。

5. ユーザー・エントリの親コンテキストをディレクトリに作成します（存在していない場合）。
6. Oracle Net Configuration Assistant を使用してデータベースの Oracle ホームに対するディレクトリ・アクセスを設定します（デフォルトの `ldap.ora` パラメータを使用する場合）。

注意：

- ユーザーの移行時に共有スキーマ・マッピングを使用する場合は、このユーティリティの実行前に共有スキーマを作成する必要があります。
 - ユーザー移行処理のフェーズ 1 とフェーズ 2 の両方で、同じ `ldap.ora` ファイルを使用する必要があります。
-

関連項目：

- 『Oracle Internet Directory 管理者ガイド』
- ユーザー移行処理の完了後にエンタープライズ・ユーザー認証を設定する方法は、[第 15 章「エンタープライズ・ユーザー・セキュリティの管理」](#)を参照してください。

ユーザー移行ユーティリティのコマンドライン構文

データベース・ユーザーからエンタープライズ・ユーザーへの一括移行処理を実行するには、次の構文を使用します。

```
umu parameter1 parameter2 ...
```

単一の値を指定するパラメータの場合は、次の構文を使用します。

```
keyword=value
```

複数の値を指定するパラメータの場合は、次のように、コロン (:) を使用して値を区切ります。

```
keyword=value1:value2:...
```

[例 16-1](#) に、一括ユーザー移行処理の両方のフェーズでユーティリティを実行するための構文を示します

例 16-1 ユーザー移行ユーティリティのコマンドライン構文

```
umu PHASE=ONE
DBADMIN=dba_username:password
ENTADMIN=enterprise_admin_DN:password
USERS=[ALL_GLOBAL | ALL_EXTERNAL | LIST | FILE]
DBLOCATION=database_host:database_port:database_sid
DIRLOCATION=ldap_directory_host:ldap_directory_port
USERSLIST=username1:username2:username3:...
USERSFILE=filename
MAPSCHEMA=[PRIVATE | SHARED]:schema_name
MAPTYPE=[DB | DOMAIN]:[ENTRY | SUBTREE]
CASCADE=[YES | NO]
CONTEXT=user_entries_parent_location
LOGFILE=filename
PARFILE=filename

umu PHASE=TWO
DBADMIN=dba_username:password
ENTADMIN=enterprise_admin_DN:password
DBLOCATION=database_host:database_port:database_sid
DIRLOCATION=ldap_directory_host:ldap_directory_port
LOGFILE=filename
PARFILE=filename
```

注意： コマンドラインで必須パラメータを指定しなかった場合は、これらのパラメータの入力を求めるプロンプトが対話形式で表示されます。

関連項目：

- 使用可能なパラメータの完全なリストとその説明は、16-13 ページの「[ユーザー移行ユーティリティ・パラメータのリスト](#)」を参照してください。
- このユーティリティの代表的な使用例については、16-21 ページの「[ユーザー移行ユーティリティの使用例](#)」を参照してください。

ユーザー移行ユーティリティ・ヘルプの表示

ユーザー移行ユーティリティの使用方法に関するコマンドライン構文を表示するには、システム・プロンプトで次のコマンドを入力します。

```
umu HELP=YES
```

HELP パラメータを YES に設定すると、ユーティリティは実行されません。

ユーザー移行ユーティリティ・パラメータのリスト

次の各項では、このユーティリティの実行時に使用できるパラメータのキーワードと値を示します。キーワードに大 / 小文字の区別はありません。

キーワード : HELP

有効な値:	YES または NO （これらの値に大 / 小文字の区別はありません。）
デフォルト設定:	NO
構文例:	HELP=YES
説明:	このキーワードは、ユーティリティのヘルプを表示するために使用します。値を YES に設定すると、コマンドライン構文の詳細が表示されます。コマンドを実行するには、値を NO に設定するか、値を指定せずにデフォルトを受け入れます。
制限事項:	なし

キーワード : PHASE

有効な値:	ONE または TWO （これらの値に大 / 小文字の区別はありません。）
デフォルト設定:	ONE
構文例:	PHASE=ONE PHASE=TWO
説明:	ユーティリティのフェーズを示します。ONE に設定すると、コマンドライン引数に指定された情報およびディレクトリ内の既存のユーザー・エントリがインタフェース表に移入されます。TWO に設定すると、インタフェース表の有効な情報によってディレクトリとデータベースが更新されます。
制限事項:	なし

キーワード : DBLOCATION

有効な値:	<i>host: port: sid</i>
デフォルト設定:	デフォルト設定はありません。
構文例:	<code>DBLOCATION=my_oracle.us.oracle.com:7777:ora902</code>
説明:	ホスト名、ポート番号およびデータベース・インスタンスの SID を指定します。
制限事項:	<ul style="list-style-type: none">■ このパラメータは必須です。■ このパラメータの値は、フェーズ 1 とフェーズ 2 の両方で同じである必要があります。■ データベースには暗号化と整合性に備えた構成が必要です。

キーワード : DIRLOCATION

有効な値:	<i>host: port</i>
デフォルト設定:	この値は、デフォルトで <code>ldap.ora</code> ファイルから自動的に移入されます。
構文例:	<code>DIRLOCATION=my_oracle.us.oracle.com:636</code>
説明:	ディレクトリ・サーバーのホスト名とポート番号を指定します。このディレクトリ・サーバーでは、LDAP サーバーが認証を行わない SSL 上で実行されています。
制限事項:	このパラメータの値は、フェーズ 1 とフェーズ 2 の両方で同じである必要があります。

キーワード : DBADMIN

有効な値:	<i>username:password</i>
デフォルト設定:	デフォルト設定はありません。
構文例:	<code>DBADMIN=system:manager</code>
説明:	データベースへの接続に必要な権限を持つ、データベース管理者のユーザー名とパスワード。
制限事項:	<ul style="list-style-type: none">■ このパラメータは必須です。■ このパラメータの <code>username</code> の値は、フェーズ 1 とフェーズ 2 の両方で同じである必要があります。

キーワード: ENTADMIN

有効な値:	<code>userDN: password</code>
デフォルト設定:	デフォルト設定はありません。
構文例:	<code>ENTADMIN=cn=janeadmin,dc=acme,dc=com:welcome</code>
説明:	ディレクトリへのログインに必要な権限を持つ、エンタープライズ・ディレクトリ管理者のユーザー識別名 (UserDN) とディレクトリ・パスワード。UserDN は二重引用符で囲んで ("...") 指定することもできます。
制限事項:	このパラメータは必須です。

キーワード: USERS

有効な値:	<code>value1:value2...</code> 次の値を指定できます。 <ul style="list-style-type: none">■ Kerberos 認証および RADIUS 認証を使用するユーザーも含めて、すべての外部ユーザーを選択する場合は、<code>ALL_EXTERNAL</code> を指定します。■ すべてのグローバル・ユーザーを選択する場合は、<code>ALL_GLOBAL</code> を指定します。■ 「キーワード: USERSLIST」を使用してコマンドラインでユーザーを指定する場合は、<code>LIST</code> を指定します。■ 「キーワード: USERSFILE」で指定したファイルからユーザーを選択する場合は、<code>USERSFILE</code> を指定します。 複数の値を指定するには、値をコロン (:) で区切ります。 (これらの値に大 / 小文字の区別はありません。)
デフォルト設定:	デフォルト設定はありません。
構文例:	<ul style="list-style-type: none">■ <code>USERS=ALL_EXTERNAL:ALL_GLOBAL</code> この使用例は、すべての外部ユーザーとすべてのグローバル・ユーザーを移行することを示しています。■ <code>USERS=ALL_EXTERNAL:FILE</code> この使用例は、すべての外部ユーザーと <code>USERSFILE</code> に指定されたすべてのユーザーを移行することを示しています。

- 説明：** 移行するユーザーを指定します。このパラメータに複数の値を指定すると、複数のユーザー・セットの組合せが使用されます。
- 制限事項：** このパラメータはフェーズ 1 の場合のみ必須で、フェーズ 2 では無視されます。

キーワード : USERSLIST

- 有効な値：** `user1:user2:...`
ユーザー名はコロン (:) で区切ります。
- デフォルト設定：** デフォルト設定はありません。
- 構文例：** `USERSLIST=jdoe:tchin:adesai`
- 説明：** 移行対象データベース・ユーザーのリストを指定します。このリスト内のユーザーは、USERS パラメータで指定したその他のユーザーとともに移行されます。
- 制限事項：** このオプション・パラメータは、USERS パラメータに LIST を指定した場合のみ有効となります。

キーワード : USERSFILE

- 有効な値：** ファイル名とパス。
- デフォルト設定：** デフォルト設定はありません。
- 構文例：** `USERSFILE=/home/orahome/userslist/hr_users.txt`
- 説明：** 移行対象データベース・ユーザーのリスト（各行に 1 ユーザーがリストされた）を含むファイルを指定します。このファイル内のユーザーは、USERS パラメータで指定したその他のユーザーとともに移行されます。
- 制限事項：** このオプション・パラメータは、USERS パラメータに FILE を指定した場合のみ有効となります。

キーワード: MAPSCHEMA

有効な値:	<code>schema_type:schema_name</code> 次のスキーマ・タイプを指定できます。 <ul style="list-style-type: none">■ PRIVATE ユーザーの元のローカル・スキーマを保持します。スキーマ・タイプが PRIVATE の場合、スキーマ名は無視されます。ディレクトリにマッピング・エントリは作成されません。■ SHARED ユーザーを共有スキーマにマップします。ディレクトリにマッピング・エントリが作成されます。スキーマ名は共有スキーマ名を示します。共有スキーマへのマッピング時にユーザーのローカル・スキーマがデータベースから削除されるかどうかは、「キーワード:CASCADE」の設定によって決まります。 (これらの値に大 / 小文字の区別はありません。)
デフォルト設定:	PRIVATE
構文例:	MAPSCHEMA=SHARED:HR_ALL
説明:	インタフェース表にスキーマ・マッピング情報を移入するかどうかを指定します。
制限事項:	<ul style="list-style-type: none">■ 「有効な値」の SHARED オプションを参照してください。■ このパラメータはフェーズ 1 の場合のみ有効です。

キーワード : MAPTYPE

有効な値: `mapping_type: mapping_level`

次のマッピング・タイプを指定できます。

- DB
- DOMAIN

次のマッピング・レベルを指定できます。

- ENTRY
- SUBTREE

コロン (:) を使用してマッピング・タイプとマッピング・レベルを区切ります。

(これらの値に大 / 小文字の区別はありません。)

デフォルト設定: `DB:ENTRY`

構文例: `MAPTYPE=DOMAIN:SUBTREE`

説明: 「[キーワード: MAPSCHEMA](#)」を SHARED に設定した場合に適用するスキーマ・マッピングのタイプを指定します。DB をマッピング・タイプに指定した場合は、データベースのディレクトリにマッピングが作成されます。DOMAIN をマッピング・タイプに指定した場合は、データベースが含まれるドメインのディレクトリにマッピングが作成されます。ドメイン・マッピングの場合は、関連する Oracle コンテキスト内での LDAP 検索によって、データベースが含まれるドメインが判別されます。

制限事項: このパラメータは、MAPSCHEMA を SHARED に設定した場合のみ有効となります。

関連項目: マッピング・レベル・オプションの使用の詳細は、16-25 ページの「[SUBTREE マッピング・レベル・オプションの使用](#)」を参照してください。

キーワード : CASCADE

有効な値 :

- NO

ユーザーを共有スキーマにマッピングするとき、ユーティリティはユーザーのローカル・スキーマをデータベースから削除しようとしています。このパラメータを **NO** に設定すると、ユーザーがローカル・スキーマのオブジェクトを所有していない場合のみ移行が行われます。元のローカル・スキーマのオブジェクトを所有しているユーザーは移行されず、移行ログ・ファイルにエラー・メッセージが記録されます。

- YES

このパラメータを **YES** に設定すると、ユーザーのローカル・スキーマとともにスキーマ・オブジェクトをすべて削除することによって、ユーザーが移行されます。以前にそのユーザーに付与された権限とロールも取り消されます。

(これらの値に大 / 小文字の区別はありません。)

デフォルト設定 :

NO

構文例 :

CASCADE=YES

説明 :

ユーザーの共有スキーマへのマップ時に、ユーザーのローカル・スキーマを削除するかどうかを指定します。

制限事項 :

このパラメータは、MAPSCHEMA を SHARED に設定した場合のみ有効となります。

キーワード : CONTEXT

有効な値 :

ユーザー・エントリの親の識別名 (DN)。

親 DN は二重引用符で囲んで ("...") 指定することもできます。

デフォルト設定 :

この値は、デフォルトで ldap.ora ファイルの DEFAULT_ADMIN_CONTEXT 設定から自動的に移入されます。この設定によって、新規ユーザー・エントリは Oracle コンテキストの親のすぐ下に配置されます。Oracle コンテキストが含まれるディレクトリ情報ツリーの図は、15-9 ページの図 15-1 「Oracle コンテキスト内の関連エントリ」を参照してください。

構文例 :

CONTEXT="c=us"

説明 :

指定したユーザーのユーザー ID に一致するディレクトリ・エントリがない場合に、ユーザー・エントリを作成するディレクトリ内の親エントリの DN を指定します。

- 制限事項:**
- UNIX では、`ldap.ora` ファイルの `DEFAULT_ADMIN_CONTEXT` 設定から管理コンテキスト値 (`c=us`) を取得できるように、Oracle ホーム環境変数を設定する必要があります。
 - このパラメータはフェーズ 1 の場合のみ有効です。

キーワード : LOGFILE

- 有効な値:** ファイル名とパス。
- デフォルト設定:** `$ORACLE_HOME/network/log/umu.log`
- 構文例:** `LOGFILE=home/orahome/network/log/filename.log`
- 説明:** 各ユーザーの移行に関する詳細を書き込むログ・ファイルを指定します。
- 制限事項:** なし

キーワード : PARFILE

- 有効な値:** ファイル名とパス。
- デフォルト設定:** デフォルト設定はありません。
- 構文例:** `PARFILE=home/orahome/network/usr/par.txt`
- 説明:** ユーザーの移行で使用するパラメータのリストが含まれたテキスト・ファイルを指定します。各パラメータは、ファイル内の別々の行に記述する必要があります。パラメータ・ファイルとコマンドラインの両方にパラメータを指定した場合は、コマンドラインに指定したパラメータが優先されます。
- 制限事項:** なし

ユーザー移行ユーティリティの使用例

次の各項では、このユーティリティの代表的な使用例について、その構文をいくつか示します。

ユーザーの移行時にユーザー所有スキーマを保持

ユーザーの移行時に各ユーザーの元のデータベース・スキーマを保持する場合は、MAPSCHEMA パラメータを PRIVATE に設定します（これがデフォルトの設定です）。たとえば、ユーザー scott1、scott2 およびすべての外部データベース・ユーザーを移行するときに、各ユーザーの元のスキーマを保持しながら、生成した新規ディレクトリ・パスワードとともに c=us にあるディレクトリに移行するには、例 16-2 に示す構文を使用します。

例 16-2 ユーザーの移行（MAPSCHEMA=PRIVATE（デフォルト）を使用）

```
umu PHASE=ONE
  DBLOCATION=machine1:1521:ora_sid
  DBADMIN=system:manager
  USERS=ALL_EXTERNAL:LIST
  USERSLIST=scott1:scott2
  DIRLOCATION=machine2:636
  CONTEXT="c=us"
  ENTADMIN="cn=janeadmin":welcome

umu PHASE=TWO
  DBLOCATION=machine1:1521:ora_sid
  DBADMIN=system:manager
  DIRLOCATION=machine2:636
  ENTADMIN="cn=janeadmin":welcome
```

フェーズ 1 が正常に完了した後、インタフェース表にユーザー移行情報が移入されます。その後、エンタープライズ・ユーザーの管理者はこの表を調べて、内容を確認できます。MAPSCHEMA パラメータに値が指定されていないため、デフォルト値の PRIVATE を使用してフェーズ 1 が実行されます。この結果、ユーザーの元のデータベース・スキーマとオブジェクトはすべて保持されます。

ユーザーの移行および共有スキーマへのマッピング

ユーザーを移行して新しい共有スキーマにマップし、各ユーザーの元のデータベース・スキーマを削除する場合は、MAPSCHEMA パラメータを SHARED に設定します。共有スキーマがすでに存在している必要があります。存在していない場合は、エンタープライズ・ユーザーの管理者が共有スキーマを作成してから、このパラメータ設定を使用してユーティリティを実行する必要があります。次の例では、ユーザー scott1、scott2 およびすべての外部データベース・ユーザーを、生成した新規ディレクトリ・パスワードとともに c=us にあるディレクトリに移行し、移行したすべてのユーザーをデータベースの新しい共有スキーマにマップします。

MAPSCHEMA を SHARED に設定して移行処理を実行するには、例 16-3 に示す構文を使用します。

例 16-3 ユーザーの移行 (MAPSCHEMA=SHARED を使用)

```
umu PHASE=ONE
    DBLOCATION=machine1:1521:ora_sid
    DBADMIN=system:manager
    USERS=ALL_EXTERNAL:LIST
    USERSLIST=scott1:scott2
    MAPSCHEMA=SHARED:schema_32
    DIRLOCATION=machine2:636
    CONTEXT="c=us"
    ENTADMIN="cn=janeadmin":welcome

umu PHASE=TWO
    DBLOCATION=machine1:1521:ora_sid
    DBADMIN=system:manager
    DIRLOCATION=machine2:636
    ENTADMIN="cn=janeadmin":welcome
```

フェーズ 1 が正常に完了した後、インタフェース表にユーザー移行情報が移入されます。その後、管理者はこの表を調べて、内容を確認できます。ユーザー scott1 および scott2 の元のデータベース・パスワードは保持されますが、元のデータベース・パスワードを持たない外部ユーザーには、ランダムに生成された新規パスワードが割り当てられます。CASCADE パラメータに値が指定されていないため、デフォルト値の NO を使用してフェーズ 1 が実行されます。この値は、元のデータベース・スキーマのデータ・オブジェクトを所有するユーザーの移行は失敗し、そのユーザーのスキーマが自動的に削除されないことを示します。失敗したユーザーを判別するには、デフォルトで \$ORACLE_HOME/network/log/umu.log にあるログ・ファイルを確認します。

異なる CASCADE オプションを使用したユーザーの共有スキーマへのマッピング

CASCADE パラメータ設定によって、移行時に共有スキーマにマッピングするときに、ユーザーの元のデータベース・スキーマを自動的に削除するかどうかが決まります。CASCADE は、MAPSCHEMA を SHARED に設定した場合のみ使用できます。

ユーザーの共有スキーマへのマッピング (CASCADE=NO を使用)

デフォルトでは、CASCADE パラメータは NO に設定されています。この設定は、移行ユーザーを共有スキーマにマッピングするときに、元のスキーマのデータベース・オブジェクトを所有しているユーザーは移行されないことを示します。データベース・オブジェクトを所有していないユーザーの元のデータベース・スキーマは自動的に削除され、これらのユーザーは新しい共有スキーマにマップされます。

関連項目： CASCADE を NO に設定して、ユーザーを共有スキーマにマップする構文例は、16-22 ページの [例 16-3](#) を参照してください。CASCADE のデフォルト設定は NO であるため、ユーティリティのコマンド構文にこのパラメータを指定する必要はありません。

ユーザーの共有スキーマへのマッピング (CASCADE=YES を使用)

移行ユーザーがデータベース・オブジェクトを所有していないか、元のデータベース・スキーマのオブジェクトを保持する必要がないことがわかっている場合は、CASCADE パラメータを YES に設定します。この設定によって、ユーザーのスキーマとスキーマ・オブジェクトはすべて削除され、これらのユーザーは新しい共有スキーマにマップされます。[例 16-4](#) は、CASCADE を YES に設定する場合に使用する構文を示しています。この例では、ユーザー scott1、scott2 およびすべての外部データベース・ユーザーを c=us にあるディレクトリに移行し、移行したすべてのユーザーをデータベースの新しい共有スキーマにマップします。

例 16-4 ユーザーの移行と共有スキーマへのマッピング (CASCADE=YES を使用)

```
umu PHASE=ONE
  DBLOCATION=machine1:1521:ora_sid
  DBADMIN=system:manager
  USERS=ALL_EXTERNAL:LIST
  USERSLIST=scott1:scott2
  MAPSCHEMA=SHARED:schema_32
  CASCADE=YES
  DIRLOCATION=machine2:636
  CONTEXT="c=us"
  ENTADMIN="cn=janeadmin":welcome
```

```
umu PHASE=TWO
    DBLOCATION=machine1:1521:ora_sid
    DBADMIN=system:manager
    DIRLOCATION=machine2:636
    ENTADMIN="cn=janeadmin":welcome
```

フェーズ 1 が正常に完了した後、インタフェース表にユーザー移行情報が移入されます。その後、管理者はこの表を調べて、内容を確認できます。CASCADE パラメータが YES に設定されているため、移行したユーザーの元のデータベース・スキーマは、ユーザーが所有するデータベース・オブジェクトも含めて、すべて自動的に削除されます。

注意： CASCADE パラメータを YES に設定する場合は、移行対象ユーザーのデータベースのバックアップまたはエクスポート・ダンプを作成してから、このユーティリティを実行することをお勧めします。移行したユーザーの元のデータベース・オブジェクトが後で必要になった場合は、エクスポート・ダンプから取得できます。

異なる MAPTYPE オプションを使用したユーザーの共有スキーマへのマッピング

MAPSCHEMA を SHARED に設定した場合は、MAPTYPE パラメータに値を指定して、マッピングのタイプを設定できます。このパラメータには、2 つの値（マッピングのタイプとマッピングのレベル）を指定します。

マッピング・タイプには、DB（データベースの場合）または DOMAIN（エンタープライズ・ドメインの場合）を設定できます。マッピング・タイプ DB を指定すると、共有スキーマが格納されているデータベースにのみマッピングが適用されます。DOMAIN をマッピング・タイプに指定すると、共有スキーマが格納されているデータベースが含まれるエンタープライズ・ドメインとそのドメイン内のすべてのデータベースにマッピングが適用されます。

マッピング・レベルには、ENTRY または SUBTREE を設定できます。ENTRY を指定すると、ユーザーの完全識別名（DN）を使用して、ユーザーが共有スキーマにマップされます。その結果、1 人のユーザーにつき 1 つのマッピングが作成されます。SUBTREE を指定すると、ユーザー DN の一部を共有するユーザー・グループが一緒にマップされます。その結果、ディレクトリ・ツリーの特定の共通ルートの下にすでにグループ化されているユーザー・グループにつき 1 つのマッピングが作成されます。例 16-5 は、MAPTYPE パラメータを設定する場合に使用する構文を示しています。この例では、ユーザー scott1、scott2 およびすべての外部データベース・ユーザーを c=us にあるディレクトリに移行し、移行したすべてのユーザーをデータベースの新しい共有スキーマにマップします。この例では、マッピングは、データベースが含まれるエンタープライズ・ドメインに適用され、エントリ・レベルで実行され、1 人のユーザーにつき 1 つのマッピングが作成されます。

例 16-5 ユーザーの移行と共有スキーマへのマッピング (MAPTYPE パラメータを使用)

```
umu PHASE=ONE
  DBLOCATION=machine1:1521:ora_sid
  DBADMIN=system:manager
  USERS=ALL_EXTERNAL:LIST
  USERSLIST=scott1:scott2
  MAPSCHEMA=SHARED:schema_32
  MAPTYPE=DOMAIN:ENTRY
  DIRLOCATION=machine2:636
  CONTEXT="c=us"
  ENTADMIN="cn=janeadmin":welcome

umu PHASE=TWO
  DBLOCATION=machine1:1521:ora_sid
  DBADMIN=system:manager
  DIRLOCATION=machine2:636
  ENTADMIN="cn=janeadmin":welcome
```

SUBTREE マッピング・レベル・オプションの使用 移行対象のユーザー (例: scott) に、その下のサブツリーにユーザー・エントリを設定する予定がある場合は、このユーザー・エントリ (cn=scott) からスキーマへのサブツリー・レベルのマッピングを作成することに意味があります。ただし、データベースは、このユーザーをサブツリー内のユーザーとして解析しないため、マッピングは scott 自体には適用されません。たとえば、DN cn=scott,o=acme を持つユーザー scott を移行し、ユーティリティの実行時に SUBTREE をマッピング・レベルとして選択した場合、cn=scott,o=acme から共有スキーマへの新規マッピングは作成されますが、ユーザー scott はそのスキーマにマップされません。共有スキーマにマップされるのは、scott ディレクトリ・エントリの下に作成された新規ユーザーのみです。したがって、SUBTREE マッピング・レベルは、ユーザー・ディレクトリ・エントリが他のユーザー・ディレクトリ・エントリの下に配置されている場合에만指定してください (ただし、これは例外的なディレクトリ構成となります)。

任意のサブツリー・ユーザーを 1 つのマッピング・エントリのみを使用して単一の共有スキーマにマップする場合、そのマッピングの作成には Oracle Enterprise Security Manager を使用する必要があります。

関連項目： Oracle Enterprise Security Manager の使用方法は、19-28 ページの「[データベース・スキーマ・マッピングの管理](#)」を参照してください。

PARFILE、USERSFILE および LOGFILE パラメータを使用したユーザーの移行

ユーザー情報とユーザー移行ユーティリティ・パラメータをテキスト・ファイルに入力し、PARFILE および USERSFILE パラメータを使用して、その情報とパラメータをユーティリティに渡すことができます。LOGFILE パラメータは、各ユーザーの移行に関する詳細を書き込むログ・ファイルのディレクトリ・パスを設定します。

PARFILE パラメータは、一括ユーザー移行処理に関するパラメータが含まれたテキスト・ファイルの場所をユーティリティに指示します。USERSFILE パラメータは、PARFILE パラメータと同様に機能します。指定ファイルにパラメータを記述するかわりに、データベース・ユーザーのリストを含めます。このパラメータとユーザーのリストでは、それぞれのテキスト・ファイルの 1 行につき 1 つのパラメータまたはユーザーを記述します。LOGFILE パラメータは、ユーザーの移行時に発生したエラーなどのシステム・イベントを書き込む場所をユーティリティに指示します。USERSFILE パラメータは、移行処理のフェーズ 1 で使用されます。PARFILE パラメータと LOGFILE パラメータは、両方のフェーズで使用できます。

例 16-6 は、ユーザー `scott1`、`scott2` およびすべての外部データベース・ユーザーを、各ユーザーの元のスキーマを保持しながら `c=us` にあるディレクトリに移行するための、典型的なパラメータ・テキスト・ファイルの構文を示しています。この例では、移行イベントのログは、ユーティリティが実行されるディレクトリにあるファイル `errorfile1` に書き込まれます。別の場所書き込む場合は、パスとファイル名を指定してください。

例 16-6 PARFILE パラメータでのパラメータ・テキスト・ファイル (`par.txt`) の使用

```
DBLOCATION=machine1:1521:ora_sid
DBADMIN=system:manager
USERS=ALL_EXTERNAL:LIST:FILE
USERSLIST=scott1:scott2
USERSFILE=usrs.txt
DIRLOCATION=machine2:636
CONTEXT="c=us"
ENTADMIN="cn=janeadmin":welcome
LOGFILE=errorfile1
```

例 16-7 は、典型的なユーザー・リスト・テキスト・ファイルの構文を示しています。

例 16-7 USERSFILE パラメータでのユーザー・リスト・テキスト・ファイル (`usrs.txt`) の使用

```
user1
user2
user3
```

これらのパラメータとユーザー・リストのテキスト・ファイルを指定して移行処理のフェーズ 1 を実行するには、例 16-8 に示す構文を使用します。

例 16-8 PARFILE、USERSFILE および LOGFILE パラメータを使用したユーザーの移行

```
umu PHASE=ONE
    DBADMIN=system:manager
    PARFILE=par.txt
    LOGFILE=errorfile2
```

注意： LOGFILE パラメータが 2 回指定されています。つまり、1 回目はパラメータ・テキスト・ファイルに `errorfile1` として（例 16-6 を参照）、2 回目はコマンドラインで `errorfile2` として（例 16-8 を参照）指定されています。この場合は、コマンドラインのパラメータがパラメータ・ファイル内のパラメータより優先されます。したがって、例 16-8 のログ・ファイルは、コマンドラインに指定された値である `errorfile2` に書き込まれます。

ユーザー移行ユーティリティでのトラブルシューティング

移行時に発生した障害は、エラー・メッセージとログ・メッセージによってエンタープライズ・ユーザーの管理者に報告されます。次の各項では、一般的なエラー・メッセージとログ・メッセージを記載し、これらを解決する方法について説明します。

関連項目： エラー・メッセージとログ・メッセージの一覧およびその説明の記載箇所については、16-37 ページの「[ユーザー移行ユーティリティのエラー・メッセージとログ・メッセージの要約](#)」を参照してください。

ユーザー移行ユーティリティの一般的なエラー・メッセージ

ユーティリティの実行中にエラーが発生した場合は、エラー・メッセージが表示され、実行が停止します。次の各項では、これらのメッセージを記載し、エラーの解決方法を説明します。

- [両方のフェーズで表示されるエラー・メッセージの解決方法](#)
- [フェーズ 1 で表示されるエラー・メッセージの解決方法](#)

両方のフェーズで表示されるエラー・メッセージの解決方法

次のエラー・メッセージは、移行処理のフェーズ 1 またはフェーズ 2 の実行中に表示される可能性があります。

- 属性値が不明です :: orclCommonNicknameAttribute
- データベース接続の失敗
- データベース・エラー : < database_error_message >
- データベースがドメインに存在しません :: DB-NAME = < database_name >
- データベースがディレクトリに登録されていません :: DB-NAME = < dbName >
- ディレクトリ接続の失敗
- ディレクトリ・エラー :: < directory_error_message >
- 複数のエントリが見つかりました :: uniqueMember = < database_DN >

属性値が不明です :: orclCommonNicknameAttribute

原因: ニックネーム属性がディレクトリのルート・コンテキストに設定されていません。

処置: Enterprise Security Manager を使用して、ルート・コンテキストのニックネーム属性を設定してください。

データベース接続の失敗

原因: データベースに接続できませんでした。

処置: 次の手順を実行してください。

1. データベースのステータスをチェックして、暗号化と整合性に関する構成が行われているかどうかを確認します。
2. ユーティリティを実行しているエンタープライズ・ユーザーの管理者の権限および資格証明をチェックします。

データベース・エラー : < database_error_message >

原因: データベース・エラーが発生しました。

処置: データベース・エラー・メッセージの詳細をチェックしてください。

関連項目: データベース・エラー・メッセージの解決方法は、『Oracle9i データベース・エラー・メッセージ』を参照してください。

データベースがドメインに存在しません :: DB-NAME = < database_name >

原因: このデータベースはエンタープライズ・ドメインのメンバーではありません。

処置: Enterprise Security Manager を使用して、データベースをディレクトリ内のエンタープライズ・ドメインに追加してください。

データベースがディレクトリに登録されていません:: DB-NAME = < dbName >

原因: ldap.ora ファイルが指し示す Oracle コンテキストにデータベースのエントリがありません。

処置: Database Configuration Assistant または Enterprise Security Manager を使用して、データベースをディレクトリに登録してください。

ディレクトリ接続の失敗

原因: ディレクトリに接続できませんでした。

処置: 次の手順を実行してください。

1. ディレクトリ・サーバーのステータスをチェックして、SSL（認証なし）用にディレクトリ・サーバー・ポートが構成されているかどうかを確認します。
2. ユーティリティを実行しているエンタープライズ・ユーザーの管理者の権限および資格証明をチェックします。

ディレクトリ・エラー:: < directory_error_message >

原因: ディレクトリ・エラーが発生しました。

処置: ディレクトリ・エラー・メッセージの詳細をチェックしてください。

関連項目: Oracle Internet Directory のエラー・メッセージの解決方法は、『Oracle Internet Directory 管理者ガイド』を参照してください。

複数のエントリが見つかりました:: uniqueMember = < database_DN >

原因: このデータベースはディレクトリ内の複数のエンタープライズ・ドメインに属しています。

処置: Enterprise Security Manager または Oracle Directory Manager を使用して、データベースが 1 つのエンタープライズ・ドメインにのみ属するようにしてください。

フェーズ 1 で表示されるエラー・メッセージの解決方法

移行処理のフェーズ 1 の実行中に、構文エラーやその他のエラーが発生する場合があります。次のエラー・メッセージは、移行処理のフェーズ 1 の実行中に表示される可能性があります。

- 引数が不明または重複しています:: < parameter >
- データベース・オブジェクトが不明です:: SHARED-SCHEMA = <shared_schema_name >
- ファイル読取り中のエラー:: ldap.ora:: DEFAULT_ADMIN_CONTEXT
- ファイル読取り中のエラー:: ldap.ora:: DIRECTORY_SERVERS
- ファイル読取り中のエラー:: < file_name >:: < io_error_message >

- ファイル読取り中のエラー :: PARFILE = <file_name> :: <io_error_message>
- ローカル・ホスト名の取得に失敗しました
- SYS スキーマでのインタフェース表の作成はできません
- 無効な引数または値です :: <argument>
- フェーズ用の引数が無効です
- 無効な値です :: <user> [USERSFILE]
- 無効な値です :: <user> [USERSFILE] { = DBADMIN }
- 無効な値です :: <user> [USERSLIST]
- 無効な値です :: <user> [USERSLIST] { = DBADMIN }
- ログインに失敗しました :: <io_error_message>
- ニックネーム属性が存在します :: CONTEXT = <context> :
orclCommonNicknameAttribute = <nickname_attribute>
- エントリが見つかりませんでした :: CONTEXT = <context>
- 検索ベースではありません :: CONTEXT = <context> : orclCommonUserSearchBase = <user_search_bases>

引数が不明または重複しています :: <parameter>

原因: 構文エラーです。パラメータが欠落しているか、複数回入力されました。

処置: 構文の使用方法をチェックしてください。

データベース・オブジェクトが不明です :: SHARED-SCHEMA = <shared_schema_name>

原因: 共有スキーマがデータベースに存在していません。

処置: 共有スキーマを作成してください。

ファイル読取り中のエラー :: ldap.ora :: DEFAULT_ADMIN_CONTEXT

原因: 構文エラーです。ldap.ora ファイル内の DEFAULT_ADMIN_CONTEXT パラメータ値の書式が無効です。

処置: Oracle Net Configuration Assistant を使用して、新しい ldap.ora ファイルを作成してください。

ファイル読取り中のエラー :: ldap.ora :: DIRECTORY_SERVERS

原因: 構文エラーです。ldap.ora ファイル内の DIRECTORY_SERVERS パラメータ値の書式が無効です。

処置: Oracle Net Configuration Assistant を使用して、新しい ldap.ora ファイルを作成してください。

ファイル読取り中のエラー :: <file_name> :: <io_error_message>

原因: 構文エラーです。USERSFILE パラメータに指定されたユーザー・リストを含むファイルを読み込むことができません。

処置: 次の手順を実行してください。

1. ファイルが存在していることを確認します。
2. 正しい読み込み許可がファイルに設定されていることを確認します。

ファイル読取り中のエラー :: PARFILE = <file_name> :: <io_error_message>

原因: 構文エラーです。PARFILE パラメータに指定されたパラメータ・リストを含むファイルを読み込むことができません。

処置: 次の手順を実行してください。

1. ファイルが存在していることを確認します。
2. 正しい読み込み許可がファイルに設定されていることを確認します。

ローカル・ホスト名の取得に失敗しました

原因: 構文エラーです。データベース位置またはディレクトリ位置に対するローカル・ホスト名を読み込むことができません。

処置: DBLOCATION パラメータおよび DIRLOCATION パラメータを使用して、ホスト名情報を明示的に入力してください。

関連項目:

- 16-14 ページ [「キーワード: DBLOCATION」](#)
- 16-14 ページ [「キーワード: DIRLOCATION」](#)

各項目に、それぞれのパラメータの使用方法が記載されています。

SYS スキーマでのインタフェース表の作成はできません

原因: SYS スキーマにインタフェース表を作成することはできません。

処置: DBADMIN パラメータに別のユーザーを指定してください。

関連項目: DBADMIN パラメータの設定方法は、16-14 ページの [「キーワード: DBADMIN」](#) を参照してください。

無効な引数または値です :: <argument>

原因: 構文エラーです。引数名または値が正しく入力されていません。

処置: 構文の使用方法をチェックしてください。

関連項目：

- 16-11 ページ「ユーザー移行ユーティリティのコマンドライン構文」
- 16-13 ページ「ユーザー移行ユーティリティ・ヘルプの表示」
- 16-13 ページ「ユーザー移行ユーティリティ・パラメータのリスト」

各項目に、このユーティリティのコマンドライン構文の使用 방법이記載されています。

フェーズ用の引数が無効です

原因：構文エラーです。このエラーは、フェーズ 1 専用のコマンドライン引数を使用してフェーズ 2 を実行している場合に発生します。

処置：構文の使用方法をチェックしてください。

無効な値です::`<user> [USERSFILE]`

原因：構文エラーです。このエラー・メッセージに示されたユーザーは、DBLOCATION パラメータに指定されているデータベースのユーザーでないため、無効です。

処置：無効なユーザーを USERSFILE パラメータで指定したファイルから削除してください。

無効な値です::`<user> [USERSFILE] { = DBADMIN }`

原因：構文エラーです。USERSFILE パラメータに指定したファイルに、この移行ユーティリティを実行しているユーザーが含まれています。

処置：そのユーザーをファイルから削除してください。

無効な値です::`<user> [USERSLIST]`

原因：構文エラーです。このエラー・メッセージに示されたユーザーは、DBLOCATION パラメータに指定されているデータベースのユーザーでないため、無効です。

処置：無効なユーザーを USERSLIST パラメータから削除してください。

無効な値です::`<user> [USERSLIST] { = DBADMIN }`

原因：構文エラーです。USERSLIST パラメータに、この移行ユーティリティを実行しているユーザーが含まれています。

処置：そのユーザーを USERSLIST から削除してください。

ログインに失敗しました::`<io_error_message>`

原因：構文エラーです。ログ・ファイルが見つからないか、書き込むファイルをオープンできません。

処置：次の手順を実行してください。

1. ログ・ファイルが存在していることを確認します。

2. 正しい書込み許可がログ・ファイルに設定されていることを確認します。

ニックネーム属性が存在します:: CONTEXT = < context >:

orclCommonNicknameAttribute = <nickname_attribute>

原因: CONTEXT 値に、不要なニックネーム属性設定が含まれています。ルート Oracle コンテキストにのみこの設定を含めます。

処置: ルート Oracle コンテキストではない別の CONTEXT 値を指定し、ニックネーム属性設定は含めないでください。

エントリが見つかりませんでした:: CONTEXT = < context >

原因: このディレクトリに CONTEXT エントリは存在しません。

処置: 次のオプションのいずれかを実行してください。

- ディレクトリ管理ツールまたは LDAP コマンドライン・ユーティリティを使用して、ディレクトリにコンテキスト値に対応するエントリを作成します。
- 他の有効なコンテキスト値を指定します。

検索ベースではありません:: CONTEXT = < context >: orclCommonUserSearchBase = < user_search_bases >

原因: この CONTEXT 値は検索ベースの下にありません。

処置: Enterprise Security Manager を使用して適切なユーザー検索ベースを追加するか、または適切な CONTEXT 値を指定してください。

フェーズ 2 で表示されるエラー・メッセージの解決方法

ほとんどのエラー・メッセージは、フェーズ 1 の実行中に発生します。フェーズ 1 が正常に完了した後、フェーズ 2 の実行中に次のエラーが発生する場合があります。

データベース・オブジェクトが不明です:: TABLE = ORCL_GLOBAL_USR_MIGRATION_DATA

原因: インタフェース表が見つかりません。

処置: 次のオプションのいずれかを実行してください。

- このユーティリティのフェーズ 1 を実行して、インタフェース表を作成します。
- DBADMIN パラメータに指定されているユーザーが、フェーズ 1 でこのパラメータに指定したユーザーと同じであることを確認します。

ユーザー移行ユーティリティの一般的なログ・メッセージ

通常、移行対象の各ユーザーが正常に移行されたかどうかについて、ログ・メッセージがログ・ファイルに書き込まれます。次の各項では、これらのメッセージを記載し、エラーの解決方法を説明します。

フェーズ 1 での一般的なログ・メッセージ

移行処理のフェーズ 1 の実行中に、ユーザーの情報がインタフェース表に正常に移入されていないことを示すメッセージが、ログ・ファイルに書き込まれる場合があります。フェーズ 1 の完了後に、ログ・ファイルを調べて、次のメッセージの有無を確認してください。

- 複数のエントリが見つかりました :: < nickname_attribute > = < username >
- エントリが見つかりませんでした :: < nickname_attribute > = < username > :: エントリが見つかりました : DN = < dn >
- 値が一致しません : orclPassword : PASSWORD_VERIFIER

複数のエントリが見つかりました :: < nickname_attribute > = < username >

原因：ニックネーム属性が複数のユーザーと一致しているか、またはユーザーが複数のニックネーム属性と一致しています。

処置：複数の一致を解決し、このメッセージがログ・ファイル・エントリに表示されたユーザーに対して、ユーティリティを再度実行してください。

エントリが見つかりませんでした :: < nickname_attribute > = < username > :: エントリが見つかりました : DN = < dn >

原因：ニックネームが一致するエントリは見つかりませんが、この DN のエントリはディレクトリに存在しています。

処置：このユーザーに別の DN を指定してください。

値が一致しません : orclPassword : PASSWORD_VERIFIER

原因：このローカル・ユーザー用に移入された orclPassword を持つ DN は存在しますが、その値がデータベースのペリファイアと一致しません。

処置：ディレクトリのユーザー・エントリとユーザーのデータベース・アカウントをチェックして、値の不一致を解決してください。

フェーズ 2 での一般的なログ・メッセージ

移行処理のフェーズ 2 の実行中に、ユーザーが正常に移行されていないことを示すメッセージが、ログ・ファイルに書き込まれる場合があります。フェーズ 2 の完了後に、ログ・ファイルを調べて、次のメッセージの有無を確認してください。

- 属性が存在します :: orclPassword
- 属性値が不明です :: orclPassword
- データベース・オブジェクトが不明です :: SHARED-SCHEMA =< shared_schema >
- エントリが見つかりました :: DN =< user_DN >
- 値が無効です :: <interface_table_column_name> =< interface_table_column_value >
- エントリが見つかりませんでした :: DN =< user_DN >
- 値が一致しません :: <nickname_attribute> :: USERNAME

属性が存在します :: orclPassword

このメッセージは通常、メッセージ「値が無効です ::<column_name>=<column_value>」とともに発生します。

原因: このエントリには、orclPassword 属性の値がすでに含まれています。

処置: インタフェース表の DBPASSWORD_EXIST_FLAG 列をチェックし、T または F によって、このユーザーに対するデータベース・パスワードの有無が正しく反映されていることを確認してください。

属性値が不明です :: orclPassword

このメッセージは通常、メッセージ「値が無効です ::<column_name>=<column_value>」とともに発生します。

原因: このユーザーのエントリの orclPassword 属性が NULL に設定されています。

処置: インタフェース表の DBPASSWORD_EXIST_FLAG 列をチェックし、T または F によって、このユーザーに対するデータベース・パスワードの有無が正しく反映されていることを確認してください。

データベース・オブジェクトが不明です :: SHARED-SCHEMA =< shared_schema >

原因: このユーザーに指定された共有スキーマがデータベースに存在しません。

処置: 次のオプションのいずれかを実行してください。

- このユーザーに正しい共有スキーマが指定されていることを確認します。誤った共有スキーマ名が指定されている場合は、インタフェース表の SHARED_SCHEMA 列を編集し、このユーザーに対してユーティリティのフェーズ 2 を再度実行します。
- データベースに共有スキーマを作成し、このユーザーに対してユーティリティのフェーズ 2 を再度実行します。

エントリが見つかりました :: DN = < user_DN >

このメッセージは通常、メッセージ「値が無効です :: <column_name>=<column_value>」とともに発生します。

原因: 指定したユーザー DN に対応するエントリがすでに存在しています。

処置: インタフェース表の USERDN_EXIST_FLAG 列をチェックし、T または F の値によって、この DN に対応するユーザー・エントリがディレクトリに存在しているかどうか为正しく反映されていることを確認してください。

値が無効です :: <interface_table_column_name> = < interface_table_column_value >

原因: このユーザーに対するインタフェース表の値が無効です。このメッセージは通常、このユーザーに関するその他のログ・メッセージを伴います。

処置: このユーザーに正しい値が入力されていることを確認してください。

エントリが見つかりませんでした :: DN = < user_DN >

このメッセージは通常、メッセージ「値が無効です :: <column_name>=<column_value>」とともに発生します。

原因: この DN のエントリがディレクトリにありません。

処置: インタフェース表の USERDN_EXIST_FLAG 列をチェックし、T または F の値によって、この DN に対応するユーザー・エントリがディレクトリに存在しているかどうか为正しく反映されていることを確認してください。

値が一致しません :: < nickname_attribute > :: USERNAME

原因: インタフェース表の USERNAME 列の値と、ディレクトリの USERDN エントリに対するニックネーム属性の値が一致しません。

処置: 次の手順を実行してください。

1. Oracle Enterprise Security Manager を使用して、ディレクトリのユーザー・エントリをチェックし、適切な変更を加えてこの値の不一致を解決します。
2. このユーザーに対してユーティリティのフェーズ 2 を再度実行します。

ユーザー移行ユーティリティのエラー・メッセージとログ・メッセージの要約

表 16-4 および表 16-5 に、すべてのエラー・メッセージとログ・メッセージの一覧と、メッセージの説明と解決方法が記載された参照先の情報を示します。

表 16-4 ユーザー移行ユーティリティのエラー・メッセージ

ユーザー移行ユーティリティのエラー・メッセージ	フェーズ
引数が不明または重複しています :: <parameter> (16-30 ページ)	1
属性値が不明です :: orclCommonNicknameAttribute (16-28 ページ)	両方
データベース接続の失敗 (16-28 ページ)	両方
データベース・エラー : <database_error_message> (16-28 ページ)	両方
データベースがドメインに存在しません :: DB-NAME = <database_name> (16-28 ページ)	両方
データベースがディレクトリに登録されていません :: DB-NAME = <dbName> (16-29 ページ)	両方
データベース・オブジェクトが不明です :: SHARED-SCHEMA = <shared_schema_name> (16-30 ページ)	1
データベース・オブジェクトが不明です :: TABLE = ORCL_GLOBAL_USR_MIGRATION_DATA (16-33 ページ)	2
ディレクトリ接続の失敗 (16-29 ページ)	両方
ディレクトリ・エラー :: <directory_error_message> (16-29 ページ)	両方
ファイル読取り中のエラー :: <file_name> :: <io_error_message> (16-31 ページ)	1
ファイル読取り中のエラー :: ldap.ora :: DEFAULT_ADMIN_CONTEXT (16-30 ページ)	1
ファイル読取り中のエラー :: ldap.ora :: DIRECTORY_SERVERS (16-30 ページ)	1
ファイル読取り中のエラー :: PARFILE = <file_name> :: <io_error_message> (16-31 ページ)	1
ローカル・ホスト名の取得に失敗しました (16-31 ページ)	1
SYS スキーマでのインタフェース表の作成はできません (16-31 ページ)	1
無効な引数または値です :: <argument> (16-31 ページ)	1
フェーズ用の引数が無効です (16-32 ページ)	1
無効な値です :: <user> [USERSFILE] (16-32 ページ)	1
無効な値です :: <user> [USERSFILE] { = DBADMIN } (16-32 ページ)	1
無効な値です :: <user> [USERSLIST] (16-32 ページ)	1
無効な値です :: <user> [USERSLIST] { = DBADMIN } (16-32 ページ)	1

表 16-4 ユーザー移行ユーティリティのエラー・メッセージ（続き）

ユーザー移行ユーティリティのエラー・メッセージ	フェーズ
ログインに失敗しました :: <code><io_error_message></code> (16-32 ページ)	1
複数のエントリが見つかりました :: <code>uniqueMember = <database_DN></code> (16-29 ページ)	両方
ニックネーム属性が存在します :: <code>CONTEXT = <context> : orclCommonNicknameAttribute = <nickname_attribute></code> (16-33 ページ)	1
エントリが見つかりませんでした :: <code>CONTEXT = <context></code> (16-33 ページ)	1
検索ベースではありません :: <code>CONTEXT = <context> : orclCommonUserSearchBase = <user_search_bases></code> (16-33 ページ)	1

表 16-5 ユーザー移行ユーティリティのログ・メッセージ

ユーザー移行ユーティリティのログ・メッセージ	フェーズ
属性が存在します :: <code>orclPassword</code> (16-35 ページ)	2
属性値が不明です :: <code>orclPassword</code> (16-35 ページ)	2
データベース・オブジェクトが不明です :: <code>SHARED-SCHEMA = <shared_schema></code> (16-35 ページ)	2
エントリが見つかりました :: <code>DN = <user_DN></code> (16-36 ページ)	2
値が無効です :: <code><interface_table_column_name> = <interface_table_column_value></code> (16-36 ページ)	2
複数のエントリが見つかりました :: <code><nickname_attribute> = <username></code> (16-34 ページ)	1
エントリが見つかりませんでした :: <code>DN = <user_DN></code> (16-36 ページ)	2
エントリが見つかりませんでした :: <code><nickname_attribute> = <username> :: エントリが見つかりました : DN = <dn></code> (16-34 ページ)	1
値が一致しません :: <code><nickname_attribute> :: USERNAME</code> (16-36 ページ)	2
値が一致しません : <code>orclPassword : PASSWORD_VERIFIER</code> (16-34 ページ)	1

Oracle Wallet Manager の使用方法

セキュリティ管理者は、Oracle Wallet Manager を使用して、Oracle クライアントおよびサーバー上の公開鍵のセキュリティ資格証明を管理します。作成された Wallet は、Oracle Enterprise Login Assistant または Oracle Wallet Manager のいずれかを使用してオープンします。

この章では、Oracle Wallet Manager について説明します。項目は、次のとおりです。

- [概要](#)
- [PKCS #12 サポート](#)
- [複数証明書サポート](#)
- [LDAP ディレクトリのサポート](#)
- [Wallet の管理](#)
- [証明書の管理](#)

関連項目： Oracle Enterprise Login Assistant を使用して安全な SSL 通信を行うために Wallet をオープンおよびクローズする方法は、[第 18 章「Oracle Enterprise Login Assistant の使用方法」](#)を参照してください。

概要

従来の秘密鍵または対称鍵の暗号化では、安全な通信を行うために、2 者以上で共有される単一の秘密鍵が必要です。この鍵は、関係者間で送信される安全なメッセージの暗号化と復号化に使用されます。そのため、各関係者にあらかじめ鍵が安全に配布されている必要があります。この方法の問題点は、鍵を安全に送信して格納することが困難なことです。

公開鍵暗号では、**公開鍵と秘密鍵のペア**と鍵を安全に配布する方法を使用することによって、この問題の解決策を提供しています。自由に使用できる**公開鍵**を使用して暗号化されたメッセージは、対応する**秘密鍵**の保持者のみが復号化できます。秘密鍵は、他のセキュリティ資格証明とともに **Wallet** と呼ばれる暗号化されたコンテナに安全に格納されます。

公開鍵アルゴリズムはメッセージの機密性を保証できますが、通信関係者の識別情報は検証しないため、安全な通信は必ずしも保証されません。安全な通信を確立するには、メッセージの暗号化に使用される公開鍵が、実際に宛先の受信者に属していることを検証することが重要です。この検証を行わないと、第三者が通信を盗聴して公開鍵要求を傍受し、その公開鍵を合法的な別のキーに置き換える可能性があります (**介在者**による攻撃)。

このような攻撃を回避するために、**認証**と呼ばれるプロセスで公開鍵の所有者を検証する必要があります。認証は、通信する双方の関係者が信頼している第三者機関である**認証局** (CA) を介して実行できます。

CA は、エンティティの名前、公開鍵およびその他の特定のセキュリティ資格証明が含まれている公開鍵証明書を発行します。この資格証明には通常、CA の名前、CA の署名および証明書の有効日 (開始日、終了日) が含まれています。

CA はその秘密鍵を使用してメッセージを暗号化し、公開鍵はメッセージを復号化するために使用されます。この方法で、そのメッセージが CA によって暗号化されたことが確認されます。CA の公開鍵は広く公開されているため、アクセスするたびに認証する必要はありません。このような CA の公開鍵は、Oracle Wallet に格納されます。

Wallet パスワード管理

Oracle Wallet Manager には、高度な Wallet パスワード管理モジュールが組み込まれており、次のようなパスワード管理方針が規定されています。

- パスワードの最小の長さ (8 文字)
- パスワードの最大の長さ (無制限)
- 英数字の混在 (必須)

強力な Wallet 暗号化

Oracle Wallet Manager は、X.509 証明書に対応付けられた秘密鍵を格納するため、強力な暗号化を必要とします。そのため、リリース 2 (9.2) では、暗号化の方式が DES 暗号化からさらに強力な暗号化アルゴリズムである、3 つの鍵を使用するトリプル DES に置き換えられています。

Microsoft Windows レジストリ

Oracle Wallet Manager では、必要に応じて、Microsoft Windows システム・レジストリ (Windows 95/98/ME/NT 4.0/2000) のユーザー・プロファイル領域または Windows ファイル管理システムの中に複数の Oracle Wallet を格納できます。レジストリに Wallet を格納すると、次のような利点があります。

- **アクセス制御の向上** レジストリのユーザー・プロファイル領域に格納された Wallet には、対応するユーザーのみがアクセスできます。そのため、システムのユーザー・アクセス制御がそのまま Wallet のアクセス制御になります。また、ユーザーがシステムからログアウトすると、そのユーザーの Wallet には実質的にアクセスできなくなります。
- **容易な管理** Wallet が特定のユーザー・プロファイルに対応付けられているため、アクセス権を管理する必要がありません。また、ユーザー・プロファイルを削除すると、プロファイルに格納されている Wallet も自動的に削除されます。レジストリ内の Wallet は、Oracle Wallet Manager を使用して作成および管理し、Oracle Enterprise Login Assistant を使用して同様にアクセスできます。
- **セキュリティの向上** Wallet がレジストリ内に埋め込まれているため、特定のユーザー・プロファイルに対応付けられている Wallet に対して他のすべてのユーザーはアクセスできません。「アクセス制御の向上」および「容易な管理」と組み合わせて考えると、これは結果的に Oracle Wallet の新たなセキュリティ・レイヤーとなります。

サポートされているオプション

- レジストリからの Wallet のオープン
- レジストリへの Wallet の保存
- 異なるレジストリ位置に名前を付けて保存
- レジストリからの Wallet の削除
- ファイル・システムから Wallet をオープンして、レジストリに保存
- レジストリから Wallet をオープンして、ファイル・システムに保存

関連項目：

- 『Oracle9i Database for Windows 管理者ガイド』
- 『Oracle9i for Windows セキュリティおよびネットワーク統合ガイド』

Oracle Wallet の機能

Oracle Wallet Manager は、Wallet の所有者が各自の Oracle Wallet 内のセキュリティ資格証明を管理および編集するために使用するスタンドアロン型の Java アプリケーションです。そのタスクには、次のものがあります。

- 公開鍵と秘密鍵のペアを生成し、CA に提出する証明書要求を作成します。
- エンティティの証明書をインストールします。
- エンティティの信頼できる証明書を構成します。
- Wallet をオープンして PKI ベースのサービスにアクセスできるようにします。
- Wallet を作成します。Wallet は、Oracle Enterprise Login Assistant または Oracle Wallet Manager のいずれかを使用してアクセスできます。
- LDAP ディレクトリへ Wallet をアップロードします。
- LDAP ディレクトリから Wallet をダウンロードします。
- Wallet をインポートします。
- Wallet をエクスポートします。

下位互換性

Oracle Wallet Manager は、リリース 8.1.5 に対する下位互換性があります。

PKCS #12 サポート

Oracle Wallet Manager は、X.509 証明書と **秘密鍵** を業界標準の PKCS #12 形式で格納します。これにより、Oracle Wallet が、サポートされているサード・パーティ製 PKI アプリケーションと相互運用可能な構造になり、オペレーティング・システム間での Wallet の移植が可能になります。

注意： Oracle Advanced Security と Oracle Wallet Manager は PKCS #12 形式に完全に準拠していますが、Netscape Communicator や Microsoft Internet Explorer などのサード・パーティ製品を使用する際に互換性に関するいくつかの問題があります。

サード・パーティー製 Wallet のインポート

Oracle Wallet Manager では、次の PKCS #12 形式の Wallet を、各製品に固有の手順と制限に従ってインポートおよびサポートできます。

- Netscape Communicator 4.x
- Microsoft Internet Explorer 5.x
- OpenSSL

サード・パーティ製 Wallet をインポートする手順は、次のとおりです。

1. 製品固有の手順に従って、Wallet をエクスポートします。
2. エクスポートした Wallet を、Oracle Advanced Security で設定しているディレクトリにオペレーティング・システム固有のファイル名で保存します。

UNIX と Windows NT の場合、ファイル名は ewallet.p12 にします。

その他のオペレーティング・システムについては、Oracle オペレーティング・システム固有のマニュアルを参照してください。

関連項目： 17-22 ページ [「信頼できる証明書のインポート」](#)

注意：

- PKCS #12 形式のサード・パーティ製 Wallet を、Oracle Wallet Manager で設定しているディレクトリにそのままのファイル名でコピーしてから、ファイル名を変更してください。UNIX/NT での Wallet のファイル名は ewallet.p12 です。
 - 一般に、ブラウザでは、(署名者自身の証明書以外の) **信頼できる証明書**が PKCS #12 形式でエクスポートされないため、SSL 接続の通信先を認証するためにトラスト・ポイントを追加が必要となる場合があります。この操作には、Oracle Wallet Manager を使用できます。
-
-

Oracle Wallet のエクスポート

Oracle Wallet Manager では、Oracle Wallet をサード・パーティ環境にエクスポートできません。Wallet をエクスポートする手順は、次のとおりです。

1. Oracle Wallet Manager を使用して、Wallet ファイルを保存します。
2. サード・パーティ製品に固有の手順に従って、Oracle Wallet Manager で作成したオペレーティング・システム固有の PKCS #12 Wallet ファイル (UNIX または NT プラットフォームでは ewallet.p12 ファイル) をインポートします。

注意：

- Oracle Wallet Manager では、Wallet ごとに複数の証明書をサポートしています。しかし、現行のブラウザでは、一般に単一証明書を含む Wallet のみのインポートをサポートしています。したがって、これらのブラウザにインポートする場合は、単一のキーのペアを含む Oracle Wallet をエクスポートする必要があります。
 - Wallet のエクスポートは、Netscape Communicator 純正のバージョンと OpenSSL に対してのみサポートされています。
-
-

複数証明書サポート

Oracle Wallet ツール（Oracle Wallet Manager、Enterprise Login Assistant）は、Wallet ごとに複数の**証明書**をサポートしており、各証明書で次の **Oracle PKI 証明書使用方法**をサポートしています。

- SSL
- S/MIME 署名
- S/MIME 暗号化
- コード署名
- CA 証明書署名

Oracle Wallet Manager では、1 つのデジタル・エンティティに対して複数の証明書をサポートしています。各証明書は、Oracle PKI 証明書使用方法のセットに対して使用できますが、すべての使用方法に対して同じ証明書を使用することはできません（使用方法の有効な組合せは、[表 17-2](#)と[表 17-3](#)を参照してください）。証明書要求と証明書は、必ず 1 対 1 で対応する必要があります。1 つの証明書要求を使用して複数の証明書を取得できます。1 つの Wallet 内に同時に複数の証明書をインストールすることはできません。

Oracle Wallet Manager では、X.509 v3 拡張 KeyUsage を使用して、Oracle PKI 証明書使用方法を定義しています（[表 17-1](#)）。

表 17-1 KeyUsage の値

値	使用方法
0	digitalSignature
1	nonRepudiation
2	keyEncipherment
3	dataEncipherment
4	keyAgreement
5	keyCertSign
6	cRLSign
7	encipherOnly
8	decipherOnly

証明書（ユーザー証明書、**信頼できる証明書**）をインストールすると、Oracle Wallet Manager は[表 17-2](#)と[表 17-3](#)を使用して、KeyUsage 拡張値を Oracle PKI 証明書使用方法にマップします。

表 17-2 Oracle Wallet Manager による Oracle Wallet へのユーザー証明書のインポート

KeyUsage の値	重要かどうか ¹	使用方法
なし	N/A	証明書は、SSL または S/MIME 暗号化で使用するためにインポートできます。
0 のみ、または 0 を含み、5 と 2 を除く任意の組合せ	N/A	S/MIME 署名またはコード署名で使用するために証明書を受け入れます。
1 のみ	重要	インポートできません。
	重要ではない	S/MIME 署名またはコード署名で使用するために証明書を受け入れます。
2 のみ、または 2 との任意の組合せ（5 を除く）	N/A	SSL または S/MIME 暗号化で使用するために証明書を受け入れます。
5 のみ、または 5 を含む任意の組合せ	N/A	CA 証明書署名で使用するために証明書を受け入れます。
その他の任意の設定	重要	インポートできません。
	重要ではない	証明書は、SSL または S/MIME 暗号化で使用するためにインポートできます。

¹ KeyUsage 拡張が「重要」な場合、証明書は他の目的で使用できません。

表 17-3 Oracle Wallet Manager による Oracle Wallet への信頼できる証明書のインポート

KeyUsage の値	重要かどうか ¹	使用方法
なし	N/A	インポートできます。
5 を除く任意の組合せ	重要	インポートできません。
	重要ではない	インポートできます。
5 のみ、または 5 を含む任意の組合せ	N/A	インポートできます。

¹ KeyUsage 拡張が「重要」な場合、証明書は他の目的で使用できません。

必要な Oracle PKI 証明書使用方法に対応する正しい KeyUsage 値を使用して、CA から証明書を取得します。1 つの Wallet には、同じ使用方法で使用する複数の鍵のペアを含めることができます。各証明書は、表 17-2 と表 17-3 に示されている複数の Oracle PKI 証明書使用方法をサポートできます。Oracle PKI アプリケーションは、必要な PKI 証明書使用方法が含まれる最初の証明書を使用します。

例: 使用方法が SSL の場合は、SSL Oracle PKI 証明書使用方法が含まれる最初の証明書が使用されます。

注意: SSL Oracle PKI 証明書使用方法は、Oracle PKI アプリケーションでサポートされている唯一の使用方法です。

LDAP ディレクトリのサポート

Oracle Wallet Manager では、Wallet を LDAP 準拠のディレクトリにアップロードし、LDAP 準拠のディレクトリから取り出すことができます。

集中化された LDAP 準拠のディレクトリに Wallet を格納すると、ユーザーは複数の場所やデバイスから Wallet にアクセスできるので、信頼性の高い一貫したユーザー認証が保証されます。また、Wallet のライフ・サイクル全体を通じて Wallet を集中管理できます。有効な Wallet を不注意で上書きしてしまうことを防ぐために、インストールされた証明書を含む Wallet のみをダウンロードできます。

関連項目:

- 17-13 ページ [「LDAP ディレクトリへの Wallet のアップロード」](#)
- 17-14 ページ [「LDAP ディレクトリからの Wallet のダウンロード」](#)

Oracle Wallet Manager では、Wallet をアップロードまたはダウンロードできるように、エンタープライズ・ユーザーがあらかじめ LDAP ディレクトリで定義および構成されている必要があります。ディレクトリに Oracle8i（またはそれ以前の）ユーザーが登録されている場合、それらのユーザーは、Wallet のアップロード / ダウンロード機能を初めて使用する際に自動的にアップグレードされます。

関連項目: 15-54 ページ [「タスク 11: エンタープライズ・ユーザーの構成」](#)

Oracle Wallet Manager は、LDAP ディレクトリへの単一パスワード・ベースの接続を使用して、ユーザー Wallet をダウンロードします。しかし、アップロードする際、オープンしている Wallet に SSL Oracle PKI 証明書使用方法を含む証明書が格納されている場合は、SSL 接続が使用されます。

関連項目: Oracle PKI 証明書ユーザーの詳細は、17-7 ページの [「複数証明書サポート」](#) を参照してください。

Wallet 内に SSL 証明書が存在しない場合は、パスワード・ベースの認証が使用されます。

注意： ディレクトリのパスワードと Wallet のパスワードは互いに独立しており、異なる値を設定できます。これらのパスワードをメンテナンスする際は、常に異なるものにするをお勧めします。「異なる」というのは、一方から他方を論理的に導出できないという意味を含みます。

Wallet の管理

この項では、Wallet を新規に作成する方法、および証明書要求の生成やエクスポート、証明書の Wallet へのインポートなどに関連する Wallet 管理タスクの実行方法について、次の各項で説明します。

- [Oracle Wallet Manager の起動](#)
- [Wallet の新規作成](#)
- [既存の Wallet のオープン](#)
- [Wallet のクローズ](#)
- [LDAP ディレクトリへの Wallet のアップロード](#)
- [LDAP ディレクトリからの Wallet のダウンロード](#)
- [変更内容の保存](#)
- [開いている Wallet を新しい位置に保存](#)
- [システム・デフォルトへの保存](#)
- [Wallet の削除](#)
- [パスワードの変更](#)
- [自動ログインの使用方法](#)

Oracle Wallet Manager の起動

Oracle Wallet Manager を起動する手順は、次のとおりです。

- Windows NT の場合：「スタート」→「プログラム」→「Oracle-<ORACLE_HOME_NAME>」→「Network Administration」→「Wallet Manager」の順に選択します。
- UNIX の場合：コマンドラインで「owm」と入力します。

Wallet の新規作成

次のように Wallet を新規に作成します。

1. メニュー・バーから「Wallet」→「New」を選択します。「New Wallet」ダイアログ・ボックスが表示されます。
2. パスワードの作成に関する必須ガイドラインに従って、「Wallet Password」フィールドにパスワードを入力します。

Oracle Wallet には、複数のデータベースに対してユーザーを認証するために使用されるユーザーの資格証明が含まれているため、Wallet に対して強力なパスワードを選択することが特に重要となります。Wallet のパスワードを突き止めた悪意あるユーザーは、その Wallet の所有者がアクセス可能なすべてのデータベースにアクセスできます。

パスワードには、英文字と数字または特殊文字を組み合わせて 8 文字以上を指定する必要があります。

注意：「admin0」、「oracle1」または「2135551212A」など、ユーザー名、電話番号または公的機関の識別番号に基づく、容易に推測されるパスワードは選択しないことをお勧めします。容易に推測できるパスワードを使用しないことで、潜在的な攻撃者によって個人情報を使用され、ユーザーのパスワードが割り出されることを予防できます。また、セキュリティの習慣として、月に 1 度または四半期に 1 度など、定期的にパスワードを変更することをお勧めします。

関連項目： 17-2 ページ「[Wallet パスワード管理](#)」

3. 「Confirm Password」フィールドにそのパスワードを再入力します。
4. 「OK」をクリックして処理を継続します。
5. 入力したパスワードが必須ガイドラインに準拠していない場合は、次のメッセージが表示されます。

Password must have a minimum length of eight characters, and contain alphabetic characters combined with numbers or special characters. Do you want to try again?

6. 警告が表示され、空の Wallet が新規に作成されたことが通知されます。証明書要求を作成するかどうかを決定するように求められます。17-18 ページの「[証明書要求の追加](#)」を参照してください。

「Cancel」を選択すると、Oracle Wallet Manager のメイン・ウィンドウに戻ります。新規に作成した Wallet が左側のウィンドウのペインに表示されます。証明書の状態は「Empty」で、Wallet によってデフォルトの信頼できる証明書が表示されます。

7. 「Wallet」→「Save In System Default」を選択して、新しい Wallet を保存します。

システム・デフォルトに Wallet を保存する許可を得ていない場合は、別の場所に保存できます。

ウィンドウの最下部のメッセージにより、Wallet が正常に保存されたことが通知されます。

既存の Wallet のオープン

次のように、ファイル・システム・ディレクトリ上にすでに存在している Wallet をオープンします。

1. メニュー・バーから「Wallet」→「Open」を選択します。「Select Directory」ダイアログ・ボックスが表示されます。
2. Wallet が存在するディレクトリ位置に移動し、そのディレクトリを選択します。
3. 「OK」をクリックします。「Open Wallet」ダイアログ・ボックスが表示されます。
4. 「Wallet Password」フィールドに Wallet パスワードを入力します。
5. 「OK」をクリックします。
6. ウィンドウの最下部に「Wallet opened successfully」というメッセージが表示され、Oracle Wallet Manager のメイン・ウィンドウに戻ります。Wallet の証明書とその信頼できる証明書が左側のウィンドウのペインに表示されます。

Wallet のクローズ

現在選択しているディレクトリで開いている Wallet を閉じる手順は次のとおりです。

- 「Wallet」→「Close」を選択します。
- ウィンドウの最下部に「Wallet closed successfully」というメッセージが表示され、Wallet が閉じられたことが通知されます。

LDAP ディレクトリへの Wallet のアップロード

ターゲットの Wallet に SSL 証明書が含まれている場合、Oracle Wallet Manager は、その Wallet を LDAP ディレクトリにアップロードする際に SSL 接続を使用します。SSL 証明書が含まれていない場合は、ユーザーがディレクトリのパスワードを入力します。Oracle Wallet Manager と Enterprise Login Assistant のどちらでも、区別なく Wallet をアップロードおよびダウンロードできます。

Wallet を不注意で壊してしまうことがないように、Oracle Wallet Manager では、ターゲットの Wallet が現在オープンしており、少なくとも 1 つのユーザー証明書を含んでいる場合を除いて、アップロード・オプションを実行できないようになっています。

Wallet をアップロードする手順は、次のとおりです。

1. 「Wallet」→「Upload into the Directory Service」を選択します。現在オープンしている Wallet が保存されていない場合は、次のメッセージを含むダイアログ・ボックスが表示されます。

Wallet needs to be saved before uploading.

Choose Yes to proceed.

2. Wallet の証明書に、SSL のキー使用方法が含まれているかチェックされます。少なくとも 1 つの証明書に SSL のキー使用方法が含まれている場合は、サーバーとポートの入力を求めるダイアログ・ボックスが表示されます。LDAP ディレクトリに対応付けられているサーバーとポートの情報を入力し、「OK」をクリックします。Oracle Wallet Manager は、SSL を使用して LDAP ディレクトリ・サーバーへの接続を試みます。

3. アップロードが失敗すると、次のメッセージが表示されます。

Upload wallet failed

アップロードが成功すると、次のメッセージが表示されます。

Wallet uploaded successfully.

4. ターゲットの Wallet に SSL のキー使用方法を含む証明書が格納されていない場合は、ユーザーの**識別名**、および LDAP サーバーとポート情報の入力を求めるダイアログ・ボックスが表示されます。これらの情報を入力し、「OK」をクリックします。Oracle Wallet Manager は、Wallet のパスワードとディレクトリのパスワードが同じであると想定し、簡易パスワード認証モードを使用して LDAP ディレクトリ・サーバーへの接続を試みます。
5. 前の手順が失敗すると、ディレクトリのパスワードの入力を求めるダイアログ・ボックスが表示されます。入力されたパスワードを使用して LDAP ディレクトリ・サーバーへの接続が試行され、失敗した場合は警告メッセージが表示されます。成功すると、ウィンドウの最下部に成功を示すメッセージが表示されます。

LDAP ディレクトリからの Wallet のダウンロード

LDAP ディレクトリから Wallet をダウンロードすると、その Wallet は作業メモリー内に格納されます。次の項で説明する保存オプションを使用して明示的に保存しないかぎり、Wallet はファイル・システムに保存されません。

関連項目：

- 「変更内容の保存」
- 「開いている Wallet を新しい位置に保存」
- 「システム・デフォルトへの保存」

LDAP ディレクトリから Wallet をダウンロードする手順は、次のとおりです。

1. 「Wallet」→「Download from the Directory Service」を選択します。
2. ユーザーの DN と、LDAP ディレクトリに対応付けられているディレクトリ・パスワード、サーバーおよびポートの情報の入力を求めるダイアログ・ボックスが表示されます。Oracle Wallet Manager は、簡易パスワード認証を使用して、LDAP ディレクトリに接続します。
3. ダウンロードが失敗すると、次の警告メッセージが表示されます。

Download wallet failed

4. ダウンロードが成功し、オープンしている既存の Wallet がある場合は、次のメッセージが表示されます。

An opened wallet already exists in memory.Do you wish to overwrite it with the downloaded wallet?

「OK」をクリックして、ダウンロードした Wallet をオープンします。

5. Oracle Wallet Manager は、ディレクトリのパスワードを使用して Wallet をオープンしようとします。
6. (ディレクトリのパスワードを使用した) Wallet のオープンが失敗すると、Wallet のパスワードの入力を求めるダイアログ・ボックスが表示されます。
7. Wallet のパスワードを使用してもターゲットの Wallet をオープンできない場合は、次のメッセージが表示されます。

Open downloaded wallet failed

Wallet のオープンに成功した場合は、次のメッセージがウィンドウの最下部にあるメッセージに表示されます。

Wallet downloaded successfully

変更内容の保存

現在開いている Wallet に対する変更を保存する手順は次のとおりです。

- 「Wallet」→ 「Save」を選択します。
- ウィンドウの最下部にあるメッセージで、Wallet の変更が選択したディレクトリ位置の Wallet に正常に保存されたことを確認します。

開いている Wallet を新しい位置に保存

現在オープンしている Wallet を新しいディレクトリ位置に保存するには、「Save As」オプションを使用します。

1. 「Wallet」→ 「Save As」を選択します。「Select Directory」ダイアログ・ボックスが表示されます。
2. Wallet を保存するディレクトリ位置を選択します。
3. 「OK」をクリックします。

選択したディレクトリに Wallet がすでに存在している場合は、次のメッセージが表示されます。

A wallet already exists in the selected path.Do you want to overwrite it?

既存の Wallet を上書きする場合は「Yes」、Wallet を別のディレクトリに保存する場合は「No」をクリックします。

ウィンドウの最下部にあるメッセージで、Wallet が選択したディレクトリ位置に正常に保存されたことを確認します。

システム・デフォルトへの保存

「Save in System Default」メニュー・オプションを使用して、現在開いている Wallet をシステム・デフォルトのディレクトリ位置に保存します。

- 「Wallet」→ 「Save in System Default」を選択します。
- ウィンドウの最下部にあるメッセージで、Wallet の変更がシステム・デフォルト Wallet の位置に正常に保存されたことを確認します。

注意： Oracle アプリケーションの中には、システム・デフォルト以外の位置にある Wallet を使用できないものがあります。

Wallet の削除

現在開いている Wallet を削除する手順は次のとおりです。

1. 「Wallet」→「Delete」を選択します。「Delete Wallet」ダイアログ・ボックスが表示されます。
2. 表示されている Wallet 位置をもう一度よく見て、正しい Wallet を削除しようとしているか確認します。
3. Wallet のパスワードを入力します。
4. 「OK」をクリックします。Wallet が正常に削除されたことを通知するダイアログ・パネルが表示されます。

注意： アプリケーション・メモリー内で開いている Wallet はすべて、アプリケーションを終了するまでメモリー内に残ります。したがって、現在使用中の Wallet を削除しても、システムの操作がただちにその影響を受けることはありません。

パスワードの変更

パスワードを変更すると、その変更がただちに有効となります。Wallet は、新しく暗号化されたパスワードで、現在選択しているディレクトリに保存されます。現在開いている Wallet のパスワードを変更する手順は次のとおりです。

1. 「Wallet」→「Change Password」を選択します。「Change Wallet Password」ダイアログ・ボックスが表示されます。
2. 既存の Wallet パスワードを入力します。
3. 新しい Wallet パスワードを入力します。

関連項目： パスワード・ポリシーに関する制限は、17-2 ページの「[Wallet パスワード管理](#)」を参照してください。

4. 新しい Wallet パスワードをもう一度入力します。
5. 「OK」をクリックします。

ウィンドウの最下部にメッセージが表示され、パスワードが正常に変更されたことが通知されます。

自動ログインの使用法

Oracle Wallet Manager の自動ログイン機能は、Wallet の不明瞭化されたコピーを作成して、パスワードなしでサービスに対する PKI ベースのアクセスを可能にします。これは、Wallet の自動ログイン機能が使用禁止にされるまで有効です。Wallet の自動ログインを使用可能にすると、その Wallet を作成したオペレーティング・システム・ユーザーに対してのみ使用可能になります。

複数の Oracle データベースに対してシングル・サインオン・アクセスを希望する場合は、自動ログインを使用可能にする必要があります（デフォルトでは使用禁止になっています）。

自動ログインを使用可能にする

自動ログインを使用可能にする手順は次のとおりです。

1. メニュー・バーから「Wallet」を選択します。
2. 「Autologin」メニュー項目の横にあるチェックボックスを選択します。ウィンドウの最下部に「Autologin enabled」というメッセージが表示されます。

自動ログインを使用禁止にする

自動ログインを使用禁止にする手順は次のとおりです。

1. メニュー・バーから「Wallet」を選択します。
2. 「Auto Login」メニュー項目の横にあるチェックボックスの選択を解除します。ウィンドウの最下部に「Autologin disabled」というメッセージが表示されます。

証明書の管理

Oracle Wallet Manager では、ユーザー証明書と信頼できる証明書の 2 種類の証明書を使用します。この項では、両方の種類の証明書を管理する方法について、次の各項で説明します。

- [ユーザー証明書の管理](#)
- [信頼できる証明書の管理](#)

注意： 認証局によって発行されるユーザー証明書をインストールするには、最初にその認証局からの信頼できる証明書をインストールする必要があります。Wallet を新規に作成すると、いくつかの信頼できる証明書がデフォルトでインストールされます。

ユーザー証明書の管理

ユーザー証明書の管理には次の作業があります。

- [証明書要求の追加](#)
- [Wallet へのユーザー証明書のインポート](#)
- [Wallet からユーザー証明書を削除する](#)
- [証明書要求の削除](#)
- [ユーザー証明書のエクスポート](#)
- [ユーザー証明書要求のエクスポート](#)

証明書要求の追加

このタスクは、複数の証明書要求を追加するときに行います。Oracle Wallet Manager では、複数の要求を作成すると、以降の要求ダイアログ・ボックスに初期要求の内容が自動的に表示されます。ユーザーはこの内容を編集できます。

実際の証明書要求が Wallet の一部になります。いずれかの証明書を再利用して、新たに証明書を取得できます。ただし、既存の証明書要求を編集することはできません。正しく入力された証明書要求のみ Wallet に格納してください。

PKCS #10 証明書要求を作成する手順は次のとおりです。

1. 「Operations」→「Add Certificate Request」を選択します。「Add Certificate Request」ダイアログ・ボックスが表示されます。
2. 次の情報（[表 17-4](#)）を入力します。

表 17-4 証明書要求：フィールドと説明

フィールド名	説明
Common Name	必須。ユーザーの識別情報またはサービスの識別情報の名前を入力します。名 / 姓の形式でユーザー名を入力します。
Organizational Unit	オプション。識別情報の組織単位の名前を入力します。 例：Finance（財務）
Organization	オプション。識別情報の組織の名前を入力します。例：XYZ Corp
Locality/City	オプション。識別情報が常駐する地方または市の名前を入力します。
State/Province	オプション。識別情報が常駐する州または省の完全名を入力します。 2 文字からなる省略形を受け入れない認証局もあるため、州は完全名で入力してください。

表 17-4 証明書要求：フィールドと説明（続き）

フィールド名	説明
Country	必須。ドロップ・ダウン・リストをクリックして、国の省略形のリストを表示します。その組織の所在地である国を選択します。
Key Size	必須。ドロップダウン・ボックスをクリックして、公開鍵と秘密鍵のペアを作成するときに使用するキー・サイズのリストを表示します。キー・サイズの値は、表 17-5 を参照してください。
Advanced	オプション。「Advanced」を選択して、「Advanced Certificate Request」ダイアログ・パネルを表示します。このフィールドを使用して、識別情報の識別名（DN）を編集またはカスタマイズします。たとえば、州の完全名と、地方を編集できます。

表 17-5 使用可能なキー・サイズ

キー・サイズ	相対的なセキュリティ・レベル
512	安全とはみなされない
768	中程度のセキュリティを提供する
1024	安全

- 「OK」をクリックします。「Oracle Wallet Manager」ダイアログ・ボックスに、証明書要求が正常に作成されたことを示すメッセージが表示されます。このダイアログ・パネルの本文から証明書要求テキストをコピーして、それを認証局に送信する電子メール・メッセージに貼り付けるか、またはその証明書要求をファイルにエクスポートすることもできます。

関連項目： 17-22 ページ「ユーザー証明書要求のエクスポート」

- 「OK」をクリックします。Oracle Wallet Manager のメイン・ウィンドウに戻り、証明書の状態が「Requested」に変わります。

Wallet へのユーザー証明書のインポート

認証局から、証明書要求が受け入れられたことを示す電子メール通知を受け取ります。認証局から受信した電子メールから証明書をコピーして貼り付けるか、ファイルからユーザー証明書をインポートするかどちらかの方法で、その証明書を Wallet にインポートします。

証明書の貼り付け

証明書を貼り付ける手順は次のとおりです。

1. 認証局から受信した電子メール・メッセージまたはファイルから証明書テキストをコピーします。「Begin Certificate」から「End Certificate」までをコピーしてください。
2. メニュー・バーから「Operations」→「Import User Certificate」を選択します。「Import Certificate」ダイアログ・ボックスが表示されます。
3. 「Paste the Certificate」ボタンをクリックしてから、「OK」をクリックします。「Import Certificate」ダイアログ・ボックスに次のメッセージが表示されます。

Please provide a base64 format certificate and paste it below.

4. ダイアログ・ボックスに証明書を貼り付けて、「OK」をクリックします。ウィンドウの最下部のメッセージで、証明書が正常にインストールされたことを確認します。Oracle Wallet Manager のメイン・ウィンドウに戻り、左パネルのサブツリー内にある対応するエントリの状態が「Ready」に変わります。

証明書をコピーおよび貼り付けるショートカット・キー：

- (UNIX の場合) コピーは [Control] + [Insert] キーを押し、貼り付けは [Shift] + [Insert] キーを押します。
 - (Windows の場合) コピーは [Ctrl] + [C] キーを押し、貼り付けは [Ctrl] + [V] キーを押します。
-

証明書を含むファイルの選択

ファイルを選択する手順は次のとおりです。

1. メニュー・バーから「Operations」→「Import User Certificate」を選択します。
2. 「Select a file...」ボタンをクリックしてから、「OK」をクリックします。「Import Certificate」ダイアログ・ボックスが表示されます。
3. 証明書がある位置のパスまたはフォルダ名を入力します。
4. 証明書ファイルの名前（例：cert.txt）を選択します。
5. 「OK」をクリックします。ウィンドウの最下部にメッセージが表示され、証明書が正常にインストールされたことが通知されます。Oracle Wallet Manager のメイン・

ウィンドウに戻り、左パネルのサブツリー内にある対応するエントリの状態が「Ready」に変わります。

Wallet からユーザー証明書を削除する

1. 左パネルのサブツリーで、削除する証明書を選択します。
2. 「Operations」→「Remove User Certificate」を選択します。ダイアログ・パネルが表示され、Wallet からそのユーザー証明書を削除するかどうか確認を求められます。
3. 「Yes」を選択します。Oracle Wallet Manager のメイン・ウィンドウに戻り、証明書の状態が「Requested」に変わります。

証明書要求の削除

証明書要求を削除する手順は、次のとおりです。

1. 左パネルのサブツリーで、削除する証明書要求を選択します。
2. メニュー・バーから「Operations」を選択します。
3. メニュー項目「Remove Certificate Request」を選択します。

注意： 対応する要求を削除する前に、証明書を削除する必要があります。

ユーザー証明書のエクスポート

証明書をエクスポートする場合は、次の手順で証明書をファイル・システム・ディレクトリに保存します。

1. 左パネルのサブツリーで、エクスポートする証明書を選択します。
2. メニュー・バーから「Operations」→「Export User Certificate」を選択します。「Export Certificate」ダイアログ・ボックスが表示されます。
3. 証明書を保存するファイル・システム・ディレクトリを入力するか、「Folders」の下ディレクトリ構造にナビゲートします。
4. 「Enter File Name」フィールドに、証明書を保存するファイル名を入力します。
5. 「OK」をクリックします。ウィンドウの最下部に、証明書が指定したファイルに正常にエクスポートされたことを通知するメッセージが表示されます。Oracle Wallet Manager のメイン・ウィンドウに戻ります。

ユーザー証明書要求のエクスポート

証明書要求をエクスポートする場合は、証明書要求をファイル・システム・ディレクトリに保存します。

1. 左パネルのサブツリーで、エクスポートする証明書要求を選択します。
2. メニュー・バーから「Operations」→「Export Certificate Request」を選択します。「Export Certificate Request」ダイアログ・ボックスが表示されます。
3. 証明書要求を保存するファイル・システム・ディレクトリを入力するか、「Folders」の下ディレクトリ構造にナビゲートします。
4. 「Enter File Name」フィールドに、証明書要求を保存するファイル名を入力します。
5. 「OK」をクリックします。ウィンドウの最下部のメッセージで、証明書要求がそのファイルに正常にエクスポートされたことを確認します。Oracle Wallet Manager のメイン・ウィンドウに戻ります。

信頼できる証明書の管理

信頼できる証明書の管理には次の作業があります。

- [信頼できる証明書のインポート](#)
- [信頼できる証明書の削除](#)
- [信頼できる証明書のエクスポート](#)
- [すべての信頼できる証明書のエクスポート](#)
- [Wallet のエクスポート](#)

信頼できる証明書のインポート

認証局から受信する電子メールから貼り付けるか、ファイルからインポートするかのいずれかの方法で、信頼できる証明書を Wallet にインポートできます。

Oracle Wallet Manager では、新規 Wallet の作成時に、VeriSign、RSA、Entrust および GTE CyberTrust の信頼できる証明書が自動的にインストールされます。

信頼できる証明書の貼り付け

証明書を貼り付ける手順は次のとおりです。

1. メニュー・バーから「Operations」→「Import Trusted Certificate」を選択します。「Import Trusted Certificate」ダイアログ・パネルが表示されます。

2. 「Paste the Certificate」 ボタンをクリックしてから、「OK」をクリックします。「Import Trusted Certificate」 ダイアログ・パネルに次のメッセージが表示されます。

Please provide a base64 format certificate and paste it below.

3. ユーザー証明書が含まれている受信した電子メール・メッセージの本文から信頼できる証明書をコピーします。「Begin Certificate」から「End Certificate」までをコピーしてください。
4. ダイアログ・パネルに証明書を貼り付けて、「OK」をクリックします。ウィンドウの最下部のメッセージで、信頼できる証明書が正常にインストールされたことが通知されます。
5. 「OK」をクリックします。Oracle Wallet Manager のメイン・ウィンドウに戻り、貼り付けた信頼できる証明書が「Trusted Certificates」ツリーの最下部に表示されます。

証明書をコピーおよび貼り付けるショートカット・キー：

- (UNIX の場合) コピーは [Control] + [Insert] キーを押し、貼り付けは [Shift] + [Insert] キーを押します。
 - (Windows の場合) コピーは [Ctrl] + [C] キーを押し、貼り付けは [Ctrl] + [V] キーを押します。
-

信頼できる証明書を含むファイルの選択

ファイルを選択する手順は次のとおりです。

1. メニュー・バーから「Operations」→「Import Trusted Certificate」を選択します。「Import Trusted Certificate」ダイアログ・パネルが表示されます。
2. 信頼できる証明書がある位置のパスまたはフォルダ名を入力します。
3. 信頼できる証明書ファイルの名前（例：cert.txt）を選択します。
4. 「OK」をクリックします。ウィンドウの最下部にメッセージが表示され、信頼できる証明書が Wallet に正常にインストールされたことが通知されます。
5. 「OK」をクリックして、ダイアログ・パネルを閉じます。Oracle Wallet Manager のメイン・ウィンドウに戻り、インポートした信頼できる証明書が「Trusted Certificates」ツリーの最下部に表示されます。

信頼できる証明書の削除

Wallet から信頼できる証明書を削除する手順は次のとおりです。

1. 信頼できる証明書ツリー内にリストされている信頼できる証明書を選択します。
2. メニュー・バーから「Operations」→「Remove Trusted Certificate」を選択します。

ダイアログ・パネルに警告が表示され、署名に使用した信頼できる証明書を削除すると、受信者側でユーザー証明書を確認できなくなることが通知されます。
3. 「Yes」をクリックします。「Trusted Certificates」ツリーから、選択した信頼できる証明書が削除されます。

注意： Wallet から信頼できる証明書を削除すると、その信頼できる証明書によって署名されている証明書は確認できなくなります。

また、信頼できる証明書が Wallet 内に依然として存在するユーザー証明書に署名するために使用されている場合は、その信頼できる証明書を削除することはできません。このような信頼できる証明書を削除するには、その前に、署名された証明書を削除する必要があります。

信頼できる証明書のエクスポート

信頼できる証明書を別のファイル・システム位置にエクスポートする手順は、次のとおりです。

1. 左パネルのサブツリーで、エクスポートする信頼できる証明書を選択します。
2. 「Operations」→「Export Trusted Certificate」を選択します。「Export Trusted Certificate」ダイアログ・ボックスが表示されます。
3. 信頼できる証明書を保存するファイル・システム・ディレクトリを入力するか、「Folders」の下ディレクトリ構造にナビゲートします。
4. 信頼できる証明書の保存先のファイル名を入力します。
5. 「OK」をクリックします。Oracle Wallet Manager のメイン・ウィンドウに戻ります。

すべての信頼できる証明書のエクスポート

信頼できるすべての証明書を別のファイル・システム位置にエクスポートする手順は、次のとおりです。

1. 「Operations」→「Export All Trusted Certificates」を選択します。「Export Trusted Certificate」ダイアログ・ボックスが表示されます。
2. 信頼できる証明書を保存するファイル・システム・ディレクトリを入力するか、「Folders」の下のディレクトリ構造にナビゲートします。
3. 信頼できる証明書の保存先のファイル名を入力します。
4. 「OK」をクリックします。Oracle Wallet Manager のメイン・ウィンドウに戻ります。

Wallet のエクスポート

テキスト・ベースの PKI 形式に Wallet をエクスポートできます。個々の構成要素は、次の規格（表 17-6）に基づいてフォーマットされます。Wallet 内部では、SSL キー使用方法を含む証明書のみが Wallet とともにエクスポートされます。

表 17-6 PKI Wallet エンコーディング規格

構成要素	エンコーディング標準
証明連鎖	X509v3
信頼できる証明書	X509v3
秘密鍵	PKCS #8

Oracle Enterprise Login Assistant の使用方法

Oracle Enterprise Login Assistant を使用すると、ローカルまたは LDAP ディレクトリに格納されている Wallet やパスワードを管理したり、安全な Secure Sockets Layer (SSL) 接続を使用可能または使用禁止にできます。

Oracle Enterprise Login Assistant は、1) SSL 認証エンタープライズ・ユーザー、および 2) パスワード認証エンタープライズ・ユーザーの両方に使用できます。

この章では、Oracle Enterprise Login Assistant について説明します。項目は、次のとおりです。

- [Oracle Enterprise Login Assistant について](#)
- [証明書認証エンタープライズ・ユーザーに対する資格証明の管理](#)
- [パスワード認証エンタープライズ・ユーザーに対する資格証明の管理](#)

Oracle Enterprise Login Assistant について

Oracle Enterprise Login Assistant は、証明書ベースとパスワード・ベースの両方のエンタープライズ・ユーザーが使用できるクライアント側のツールです。

証明書ベースのユーザーに対して、根本的な複雑さを意識せずに既存の Wallet と PKI 証明書への簡単なアクセスを提供します。ユーザーは、Enterprise Login Assistant を使用して安全に Wallet をオープンした後、シングル・サインオン (SSO) を使用して中央に位置する LDAP ディレクトリ・サービスに対して認証を行い、追加のデータベース・パスワードを指定せずに複数のデータベースに接続できます。また、Enterprise Login Assistant を使用して、暗号化された Wallet を LDAP ディレクトリとの間でアップロードまたはダウンロードしたり、ディレクトリ・パスワード (Oracle Internet Directory のみ)、データベース・パスワードおよび Wallet パスワードを更新できます。

Enterprise Login Assistant は、パスワード・ベースのユーザーに対して、複数のデータベースにアクセスするための単一のグローバル・パスワードを設定および管理する機能を提供します。この機能によって、Wallet と証明書を設定および管理する必要がなくなります。パスワード・ベースのユーザーは、このパスワードをそれぞれのデータベース接続に入力する必要があります。

Enterprise Login Assistant は、すべてのエンタープライズ・ユーザーに対して、厳密な認証、安全な接続および優れた操作性を提供します。

Oracle Enterprise Login Assistant の起動

Oracle Enterprise Login Assistant の起動手順は、Oracle オペレーティング・システム固有のマニュアルを参照してください。

証明書認証エンタープライズ・ユーザーに対する資格証明の管理

ここでは、次の項目について説明します。

- ローカル・システム上の既存 Wallet のオープン
- LDAP ディレクトリへの接続と新規 Wallet のダウンロード
- パスワードの変更
- LDAP ディレクトリへの Wallet のアップロード
- ログアウトと SSL 接続を使用禁止にする方法

関連項目： Oracle Wallet Manager を使用した Wallet の管理方法は、
第 17 章「[Oracle Wallet Manager の使用方法](#)」を参照してください。

ローカル・システム上の既存 Wallet のオープン

Oracle Enterprise Login Assistant は、起動時に、デフォルトのシステム位置にあるインストール済みの Wallet を検索します（デフォルトのシステム位置は、使用している Oracle オペレーティング・システム固有のマニュアルを参照してください）。インストール済みの Wallet が見つかったら、ログイン・ウィンドウが表示されます（[図 18-1](#)）。

図 18-1 「Enterprise Login Assistant Login」ウィンドウ（Wallet が見つかった場合）



ローカルの Wallet を使用して安全な SSL 接続を確立する手順は、次のとおりです。

1. 「Local Copy」ボタンをクリックします。
2. Wallet のパスワードを入力します。
3. パスワードを変更するには、「Change passwords」ボタンをクリックします。図 18-5 のウィンドウが表示されます。18-7 ページの「パスワードの変更」を参照してください。
4. 「Login」ボタンをクリックします。

Enterprise Login Assistant は、ローカル・ファイル・システム内に Wallet のコピーを作成し、ユーザーはログインされた状態になります。「Logged-In」ウィンドウが表示されます（図 18-2）。これによって、自動ログインが使用可能になります。

注意： Oracle Wallet は、LDAP ディレクトリとローカル・ファイル・システムのいずれで作成された場合でも常に暗号化されます。ただし、自動ログインを使用可能にすると、自動ログオン・アクセスを可能にするために不明瞭化されます。

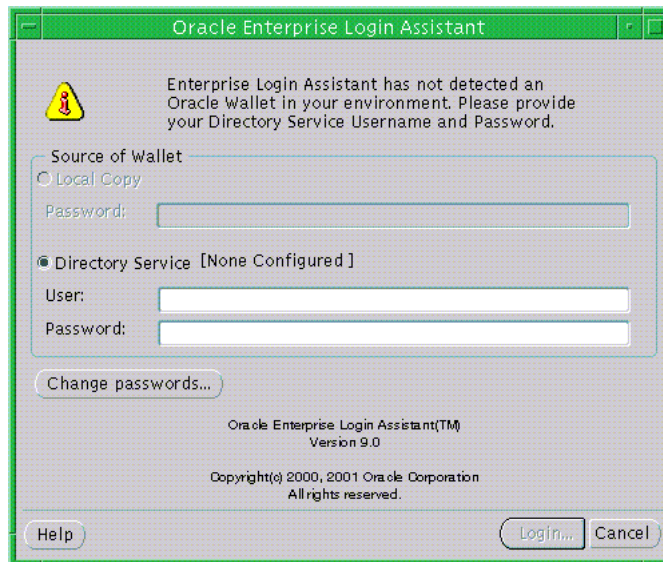
関連項目：「[不明瞭化](#)」

図 18-2 「Enterprise Login Assistant Logged-In」 ウィンドウ



ローカル・システムでインストール済みの Wallet が見つからない場合は、次のウィンドウが表示されます (図 18-3)。

図 18-3 「Enterprise Login Assistant Login」 ウィンドウ (Wallet が見つからない場合)



LDAP ディレクトリから新規 Wallet をダウンロードするには、次の項を参照してください。

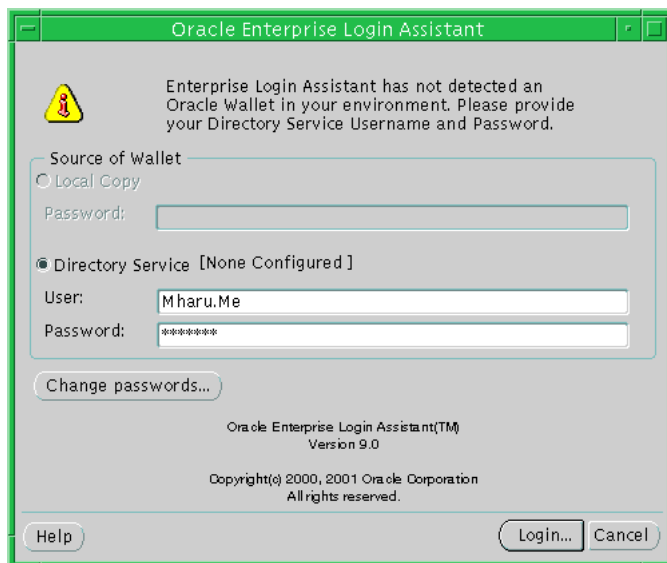
LDAP ディレクトリへの接続と新規 Wallet のダウンロード

Enterprise Login Assistant を使用して、LDAP ディレクトリからローカル・システムに Wallet をダウンロードできます。

LDAP ディレクトリに接続し、Wallet をダウンロードする手順は、次のとおりです。

1. 「Directory Service」 ボタンをクリックします (図 18-3)。

図 18-4 「Enterprise Login Assistant Directory Login」 ウィンドウ



2. 識別名またはディレクトリのユーザー ID とパスワードを入力します。
3. 「Login」 ボタンをクリックします。

Enterprise Login Assistant は、ディレクトリに接続して Wallet をダウンロードしようとします。ディレクトリ・サービスがない場合は、ディレクトリ・サービスのホスト名とポート番号の入力を求めるプロンプトが表示されます (詳細は、システム管理者にお問い合わせください)。

4. Enterprise Login Assistant は、ローカル・システムのデフォルトの位置に Wallet を格納し、ディレクトリのパスワードを使用して Wallet を復号化しようとします。ディレクトリのパスワードが Wallet のパスワードと一致しない場合は、Wallet 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。

注意： Wallet のデフォルト位置は、次のとおりです。

- UNIX の場合： `/etc/oracle/wallets/<username>`
 - Windows の場合： `<userprofile>\ORACLE\WALLETS`
-

5. Enterprise Login Assistant は、ローカル・ファイル・システム内に Wallet の不明瞭化されたコピーを作成し、ユーザーは **logged-in** 状態になります。「**Logged-In**」ウィンドウが表示されます (図 18-2)。これによって、Wallet がローカル・システムに正常にコピーされ、自動ログインが使用可能であることを確認できます。

パスワードの変更

Enterprise Login Assistant を使用して、次のパスワードを変更できます。

- Wallet パスワード
このパスワードは、ローカル Wallet へのアクセスに使用します。
- ディレクトリ・パスワード
このパスワードは、Oracle Internet Directory へのバインドに使用します。
- データベース・パスワード
これは、エンタープライズ・ユーザーが複数のデータベースに対する認証に使用する単一のグローバルなパスワードです。

パスワードを変更する手順は、次のとおりです。

1. 「**Logged In**」ウィンドウ (図 18-2) で、「**Change password**」ボタンをクリックします。「**Change Enterprise Password**」ウィンドウが表示されます (図 18-5)。

図 18-5 「Enterprise Login Assistant Change Password」 ウィンドウ

2. 次の「Password Change Options」の1つを選択します。
 - Directory and Oracle Database Password
 - Directory Password Only
 - Oracle Database Password Only
 - Local Wallet Password Only
3. 「User」フィールドに識別名（DN）またはディレクトリのユーザー ID を入力します。
4. 「Old password」フィールドに、既存のパスワードを入力します。
5. 「New password」フィールドに（パスワード・ポリシーに従って）新しいパスワードを入力し、確認のために「Confirm password」フィールドに同じパスワードを再入力します。
6. （オプション）「Reminder」フィールドにパスワード・ヒントを入力し（次の注意を参照）、「OK」をクリックします。

「Old password」フィールドに入力したパスワードと既存のパスワードが一致した場合は、選択されたパスワードが新しいパスワードとオプションのヒントに更新されます。新しいパスワードに更新されたことを確認する次のメッセージが表示されます。

Password changed successfully.

「OK」をクリックして、ダイアログ・ボックスを閉じます。

注意： Oracle Enterprise Login Assistant は、Wallet を消失した場合のリカバリに役立つリマインダ（ヒント）を提供しますが、このリマインダは暗号化されません。したがって、アクセス制御リスト（ACL）による制限と併用する場合にのみ使用してください。ACL の構成方法は、『Oracle Internet Directory 管理者ガイド』を参照してください。

リマインダについて、次の事項を考慮してください。

- リマインダは必須ではありません。必要時以外は使用しないでください。システムはリマインダを使用しない方が安全です。暗号化されていないリマインダの存在は、パスワードを不正に取り出す可能性を無許可の第三者に提供することになります。
 - ただし、Wallet のパスワードを忘れた場合に Wallet をリカバリするというエンタープライズ要件があり、ACL によってアクセスが制限されている場合（必須条件）は、このリマインダが役立ちます。
 - リマインダは、LDAP ディレクトリに格納されている Wallet に対してのみ入力できます。「Local Wallet Password Only」を選択した場合、識別名およびユーザー ID は必要ありません。また、「Reminder」フィールドには入力できません。
 - リマインダを使用する場合は、リマインダの品質を確実なものとするために、エンタープライズ全体のルールを作成してください。適切なリマインダの最終テストでは、リマインダが意図されたユーザーのみに対して意味を成し、対応するパスワードのセキュリティを危険にさらさないことを確認する必要があります。
-
-

注意： エンタープライズのインストールによっては、特別なセキュリティ要件が必要な場合があります。セキュリティ管理者は、ユーザーが特定のパスワードを更新しないように、またはユーザーにすべてのパスワードを同一にすることを強制するように、LDAP ディレクトリのアクセス制御を設定できます。

LDAP ディレクトリへの Wallet のアップロード

LDAP ディレクトリに Wallet をアップロードする手順は、次のとおりです。

1. 「Logged-In」 ウィンドウ (図 18-2) で、「Upload Wallet」 ボタンをクリックします。
2. 現行のセッションで LDAP ディレクトリ・サービスに対してすでに認証されている場合は、Wallet のコピーがディレクトリにアップロードされ、既存の Wallet と置き換えられます。
3. 現行のセッションで LDAP ディレクトリ・サービスに対してまだ認証されていない場合は、ディレクトリに接続するため、手順 2 が実行される前に識別名 (DN) またはディレクトリのユーザー ID とパスワードの入力を求めるプロンプトが表示されます。

ログアウトと SSL 接続を使用禁止にする方法

Oracle Enterprise Login Assistant を使用して、サーバー側アプリケーションからのシングル・サインオン通信を使用禁止にできます。

ログアウトして SSL 接続を使用禁止にする手順は、次のとおりです。

1. 「Logged-In」 ウィンドウ (図 18-2) で、「Logout」 ボタンをクリックします。
次の警告メッセージが表示されます。

If you log out, your applications will no longer use the security credentials of your wallet.
2. 「Yes」 をクリックして処理を継続します。「Login」 ウィンドウに戻ります (図 18-1)。

パスワード認証エンタープライズ・ユーザーに対する資格証明の管理

ここでは、次の項目について説明します。

- [パスワードの変更](#)

パスワードの変更

Enterprise Login Assistant を使用して、次のパスワードを変更できます。

- ディレクトリ・パスワード

このパスワードは、Oracle Internet Directory へのバインドに使用します。

- データベース・パスワード

これは、エンタープライズ・ユーザーが複数のデータベースに対する認証に使用する単一のグローバルなパスワードです。

Oracle Enterprise Security Manager の使用方法

この章では、Oracle Enterprise Security Manager を使用して、Oracle9i データベースのエンタープライズ・ユーザー・セキュリティを管理する方法について説明します。次の項目について説明します。

- [概要](#)
- [Oracle Enterprise Security Manager のインストールと構成](#)
- [エンタープライズ・ユーザーの管理](#)
- [Oracle コンテキストの管理](#)

関連項目：『Oracle Internet Directory 管理者ガイド』

概要

Oracle Enterprise Manager のコンポーネントの 1 つである Oracle Enterprise Security Manager は、LDAP 準拠のディレクトリ・サービスに格納されている **エンタープライズ・ユーザー**、**エンタープライズ・ドメイン**、データベースおよび **エンタープライズ・ロール** を管理するために Oracle Advanced Security で使用される管理ツールです。

ディレクトリ・サービスは、ネットワーク内のユーザー情報およびサーバー・アクセス情報を定義する中央リポジトリとして使用します。ディレクトリ・サービスには、ディレクトリで定義されているユーザーのネーミング情報、グローバル・パスワードの定義、公開鍵インフラストラクチャ (PKI) 資格証明およびアプリケーションへのアクセス認可が格納されています。このようにエンタープライズ・ユーザーとそのアクセス権限を 1 箇所に格納することにより、シングル・サインオン機能のサポートと、安全でスケーラブルなユーザー管理が可能になります。

Oracle Enterprise Security Manager のインストールと構成

次のタスクでは、Oracle Enterprise Security Manager を使用して、Oracle Management Server と Oracle Enterprise Manager をインストールする方法について説明します。

- **タスク 1: Oracle Internet Directory の構成**
- **タスク 2: Oracle Enterprise Manager のインストール**
- **タスク 3: Oracle Enterprise Security Manager の起動**
- **タスク 4: ディレクトリへのログイン**

タスク 1: Oracle Internet Directory の構成

Oracle9i のエンタープライズ・ユーザー・セキュリティは、LDAP 準拠のディレクトリを基盤としています。そのため、Oracle Enterprise Manager を使用してエンタープライズ・ユーザー・セキュリティを管理するには、ディレクトリ・サーバーを正しくインストールし、構成する必要があります。タスクを継続する前に完了しておく必要のあるディレクトリの構成作業は、次のとおりです。

- 互換性のある LDAP 準拠のディレクトリをインストールし、実行して、標準 LDAP と Secure Sockets Layer LDAP (LDAP/SSL) のどちらかを介してもアクセス可能にする必要があります。

関連項目：

- 『Oracle Internet Directory 管理者ガイド』
- 付録 E 『Microsoft Active Directory でのエンタープライズ・ユーザー・セキュリティの使用』
- Oracle9i のディレクトリ・スキーマ・オブジェクトをサポートするために Oracle Internet Directory を構成し、ルート of **Oracle コンテキスト** を定義する必要があります。ディレクトリ・サーバーでこれら両方を構成するには、Oracle Net Configuration Assistant を使用します。

関連項目：『Oracle9i Net Services 管理者ガイド』

タスク 2: Oracle Enterprise Manager のインストール

Oracle Enterprise Manager には、エンタープライズ・ユーザー・セキュリティをサポートするために必要な機能がすべて含まれています。このツールは、Oracle9i Enterprise Edition サーバーのインストール時に自動的にインストールされます。また、Oracle9i インフラストラクチャ・インストールを使用すると、デフォルトで Oracle Enterprise Manager と Oracle Internet Directory が同時にインストールされます。カスタム・インストール・オプションを使用して、Oracle Enterprise Manager を固有の ORACLE_HOME とは別の場所にインストールすることもできます。

関連項目：

- 『Oracle Enterprise Manager 管理者ガイド』

注意： Oracle Enterprise Security Manager を実行するために、特別な構成作業は必要ありません。ただし、Oracle Enterprise Security Manager を使用するエンタープライズ内のすべての Oracle データベースに対して、Oracle Enterprise Manager の ORACLE_HOME から Oracle Net を介してアクセスする必要があります。

タスク 3: Oracle Enterprise Security Manager の起動

Oracle Enterprise Security Manager を起動するには、次のいずれかを使用します。

- (UNIX の場合)

Enterprise Manager の ORACLE_HOME から、コマンドラインで次のように入力します。

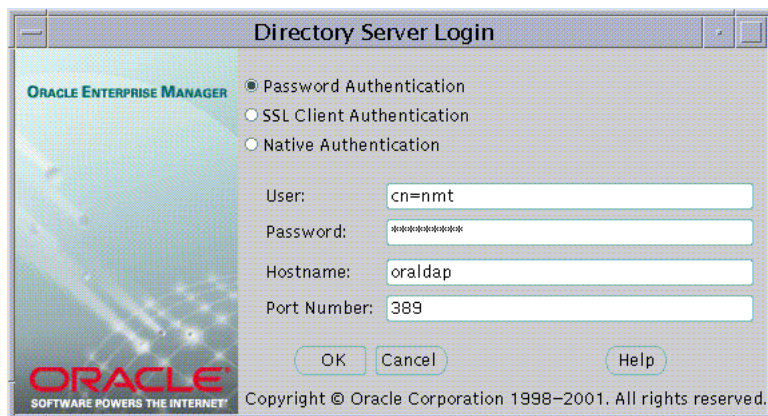
```
esm
```

- (Windows の場合)

「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Integrated Management Tools」→「Enterprise Security Manager」を順に選択します。

ディレクトリ・ログイン・ボックスが表示されます (図 19-1)。

図 19-1 「Directory Server Login」ウィンドウ



注意： Oracle Enterprise Security Manager のすべての機能は、コマンドラインで次の構文を入力することによっても利用できます。

```
esm -cmd <options>
```

オプションの完全なリストを取得するには、コマンドラインで次の構文を入力します。

```
esm -cmd
```

タスク 4: ディレクトリへのログイン

Oracle Enterprise Security Manager を使用してディレクトリ・サーバーに接続するには、3通りの方法があります。それらをまとめたものが表 19-1 です。

表 19-1 Oracle Enterprise Security Manager の認証方法

認証方式	説明
パスワード認証	識別名 またはディレクトリの既知のユーザー ID とパスワード（つまり、ユーザー名とパスワード）を必要とする簡易認証を使用します。
SSL クライアント認証	クライアントとサーバーの両方がデジタル証明書（つまり、ユーザー名と証明書）を含んだ Oracle Wallet を使用する双方向の SSL 認証を使用します。以降の接続は暗号化されます。
システム固有の認証	Microsoft Windows NT と Windows 2000 のみに適用される方法です。オペレーティング・システム・レベルの認証を使用して、Microsoft Active Directory にログインします。

認証方式を選択するには、「Directory Server Login」ウィンドウ（[図 19-1](#)）で適切なオプションを選択します。

エンタープライズ・ユーザーの管理

Oracle Enterprise Security Manager では、メイン・アプリケーション・ツリーの最上部で識別される 1 つのディレクトリ・サーバーを管理します。管理者は、そのディレクトリ内のユーザーおよび **Oracle コンテキスト** を管理します。Oracle コンテキストとは、Oracle 製品が認識できるディレクトリ内のサブツリーです。Oracle コンテキストは、Oracle データの管理階層を示します。これには、ディレクトリにアクセスするインストール済みの Oracle 製品も含まれています。

この項では、Oracle Enterprise Security Manager を使用してエンタープライズ・ユーザーを管理する方法について説明します。次の項目について説明します。

- [エンタープライズ・ユーザーの新規作成](#)
- [ディレクトリ・ベースの定義](#)
- [新規エンタープライズ・ユーザーのパスワードの定義](#)
- [初期エンタープライズ・ロール割当ての定義](#)
- [Wallet の作成](#)
- [ディレクトリ内のユーザーのブラウズ](#)
- [データベース・アクセスを使用可能にする方法](#)

エンタープライズ・ユーザーの新規作成

Oracle Enterprise Security Manager を使用して、ディレクトリのユーザーを作成できます。

新規ユーザーを作成するには、「Operations」メニューから「Create Enterprise User...」を選択します。「Create User」ウィンドウが表示されます (図 19-2)。

図 19-2 Oracle Enterprise Security Manager: 「Create User」 ウィンドウ (「User Naming」 タブ)

The screenshot shows the 'Create User' dialog box with the 'User Naming' tab selected. The fields are as follows:

- Base: dc=com (with a 'Browse..' button)
- First Name: Richard
- Surname: Bentoni
- User ID: Richard.Bentoni
- User ID Suffix: (empty)
- Email Address: Richard.Bentoni@mycompany.com
- Common Name: cn= Richard Bentoni

Buttons at the bottom: OK, Cancel, Help.

表 19-2 を参照して、「User Naming」タブ・ウィンドウに必要なとされる適切なユーザー情報を入力します。「OK」をクリックし、新規のエンタープライズ・ユーザーを作成します。

表 19-2 「Create User」 ウィンドウのフィールド

フィールド名	必須かどうか	説明
Base	必須	新規ユーザーを作成するディレクトリ内のエントリ。
First Name	必須	名。
Surname	必須	姓。
UserID	必須	ネットワーク、データベースおよびアプリケーションに接続するためにユーザーが使用できるユーザー名 (ログイン識別子)。
UserID Suffix	オプション	ユーザー ID の後に追加される共通のユーザー ID 接尾辞の現行値。たとえば、次のように指定します。 <userID>.us.acme.com
Email Address	オプション	新規ユーザーの電子メール・アドレス。

表 19-2 「Create User」 ウィンドウのフィールド（続き）

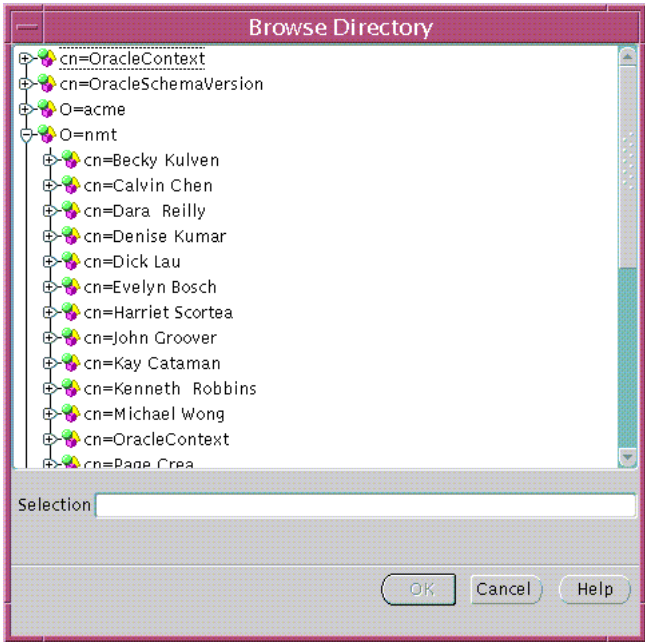
フィールド名	必須かどうか	説明
Common Name: cn=	必須	ディレクトリにおける、新規ユーザーの識別名（DN）の共通名コンポーネント（cn=）。デフォルトでは、新規ユーザーの完全名に設定されます。この値を変更して、DN の cn 部分に特定の値を設定できます。

ディレクトリ・ベースの定義

エンタープライズ・ユーザーのエントリは、ディレクトリ内のどの**ベース**にも配置できます。国エントリ（c=us）や組織エントリ（o=acme、c=us）など、既存のどのようなディレクトリ・エントリでもベースにすることができます。通常は、複数のユーザーが1つのディレクトリ・ベースを共有します。このベースによって、その下に含まれるすべてのユーザーが、階層内の同じレベルの構造に対応付けられます。

ベースは、「Create User」ウィンドウの「Base」フィールドに入力できます（図 19-2）。また、同じウィンドウの「Browse...」ボタンをクリックしてディレクトリ全体をブラウズし、適切なベースを選択することもできます。「Browse Directory」ウィンドウが表示されます（図 19-3）。

図 19-3 Oracle Enterprise Security Manager: 「Browse Directory」ウィンドウ

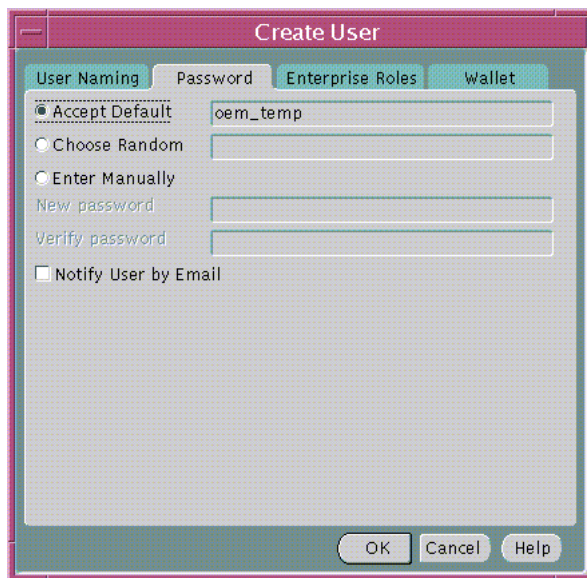


「Browse Directory」ウィンドウでは、ディレクトリ・ツリーの最上部から各エントリにドリルダウンして、ディレクトリをナビゲートできます。ディレクトリの**エントリ**を選択すると、その**識別名**が「Selection」フィールドに表示されます。選択した DN を使用する場合は、「OK」をクリックします。この値は、新規のディレクトリ・ユーザーに対して選択されたベースとして返されます。また、この値は保存されて、それ以降、ディレクトリ内でユーザーを作成または検索する際に使用されます。ただし、この値はその時々に応じて変更できます。

新規エンタープライズ・ユーザーのパスワードの定義

「Create User」ウィンドウ (図 19-4) の「Password」タブを使用して、エンタープライズ・ユーザーのパスワードを定義およびメンテナンスできます。

図 19-4 Oracle Enterprise Security Manager: 「Create User」ウィンドウ (「Password」タブ)



エンタープライズ・ユーザーのパスワードは、次の用途で使用されます。

- ディレクトリへのログイン
- グローバル・ユーザーのパスワード認証をサポートするデータベースへのログイン
- この時点で新規ユーザーに対して作成した新規の Oracle Wallet

新規パスワードを作成する場合は、次のいずれかの方法を選択できます。

- 表示されているデフォルトのパスワードをそのまま使用
- ランダムに生成されたパスワードを選択
- 手動でパスワードを入力

新規ユーザーにパスワードを電子メールで送信するには、「Notify User by Email」を選択します。これによって、最初に使用した後にパスワードを変更するように新規ユーザーに連絡できます。「User Naming」タブ (図 19-2) の電子メール・アドレスが使用されます。

注意： デフォルトの場合、Oracle Enterprise Security Manager は、指定のユーザーに対するディレクトリ、データベースおよび Wallet のパスワードに同一の値を設定します。それらのパスワードに異なる値を設定することによって、セキュリティを強化できます。その場合は、Oracle Enterprise Login Assistant を使用してユーザーのパスワードをリセットします。

関連項目： 第 18 章「Oracle Enterprise Login Assistant の使用方法」

初期エンタープライズ・ロール割当ての定義

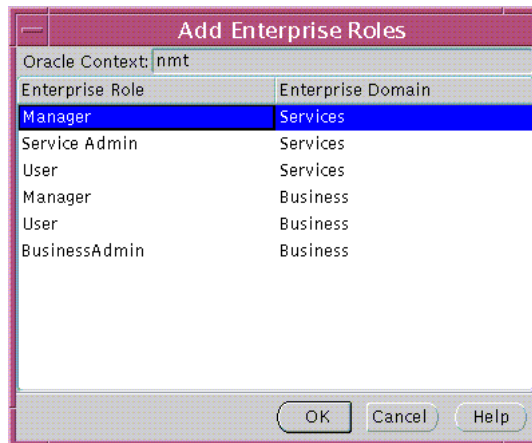
新規のエンタープライズ・ユーザーを作成するときは、すでに構成済みのエンタープライズ・ロールを新規ユーザーに付与できます。

関連項目： 19-38 ページ「エンタープライズ・ロールの管理」

1 つ以上のエンタープライズ・ロールを選択して新規ユーザーに付与するには、「Create User」ウィンドウの「Enterprise Roles」タブで、「Add...」ボタンをクリックします。

「Add Enterprise Roles」ウィンドウが表示されます（図 19-5）。

図 19-5 Oracle Enterprise Security Manager: 「Add Enterprise Roles」ウィンドウ



適切な Oracle コンテキストを選択し、新規ユーザーに割り当てる Oracle コンテキストのエンタープライズ・ロールを選択して「OK」をクリックします。

Wallet の作成

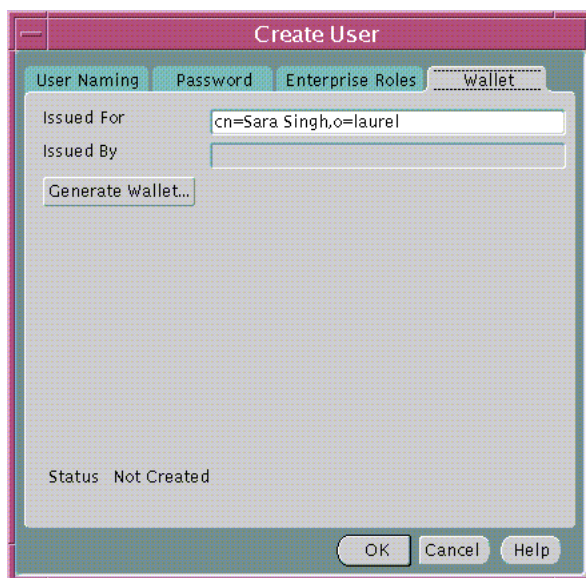
新規デジタル証明、秘密鍵および証明書トラスト・ポイントが含まれるサンプル Oracle Wallet を、新規ユーザー用に暗号化されたバイナリ・フォーマットで作成できます。Oracle Wallet は、ユーザーのディレクトリ・エントリの一部として、新規ユーザーとともにディレクトリ・サーバーに格納されます。新規ユーザーにサンプル Wallet を作成するには、「Create User」ウィンドウの「Wallet」タブを選択します (図 19-6)。

注意：「Wallet」タブは、Oracle Enterprise Security Manager でローカル認証局が構成されている場合にのみ表示されます。ローカル擬似認証局を作成するには、次のツールを実行します。

```
esm -genca
```

表示されたプロンプトに従います。このツールによって、Oracle Wallet ディレクトリに擬似認証局が作成されます。

図 19-6 Oracle Enterprise Security Manager: 「Create User」ウィンドウ (「Wallet」タブ)



新規ユーザーがその下に作成される識別名 (DN) は、デフォルトで、新規ユーザーの Oracle Wallet に含まれるデジタル証明書の DN として使用されます。ユーザー証明書の DN がディレクトリ内の対応する DN と等しくない場合、ユーザーはデータベースに接続できません。ただし、「Issued For」フィールドの内容を編集することによって、Wallet を生成する前に証明書に使用する DN を編集できます。

「Generate Wallet...」ボタンをクリックすると、サンプル Oracle Wallet が作成されます。「Edit User」ウィンドウでユーザーを選択すると (図 19-8 を参照)、そのユーザーの属性リストに `userpkcs12` 属性が表示されるようになります。`userpkcs12` 属性は、ここで作成される Wallet を示します。

注意： この擬似認証局で生成される Wallet はサンプルです。データベースとディレクトリ間の SSL 接続に有効なデータベース Wallet が必要な場合は、Oracle Wallet Manager を使用して Wallet を作成する必要があります。

関連項目： 第 17 章「Oracle Wallet Manager の使用方法」

ディレクトリ内のユーザーのブラウズ

Oracle Enterprise Security Manager では、現在格納されているすべてのユーザーのディレクトリをブラウズできます。

エンタープライズ・ユーザーをブラウズするには、メイン・ウィンドウの「All Users」タブを選択します (図 19-7)。

図 19-7 Oracle Enterprise Security Manager: メイン・ウィンドウ (「All Users」タブ)



ディレクトリ内でユーザーを検索するには、検索基準を定義して「Search Now」ボタンをクリックします。検索結果ウィンドウが表示されます。表 19-3 は、検索基準と、それによる検索結果への影響をまとめたものです。

表 19-3 ディレクトリ検索基準

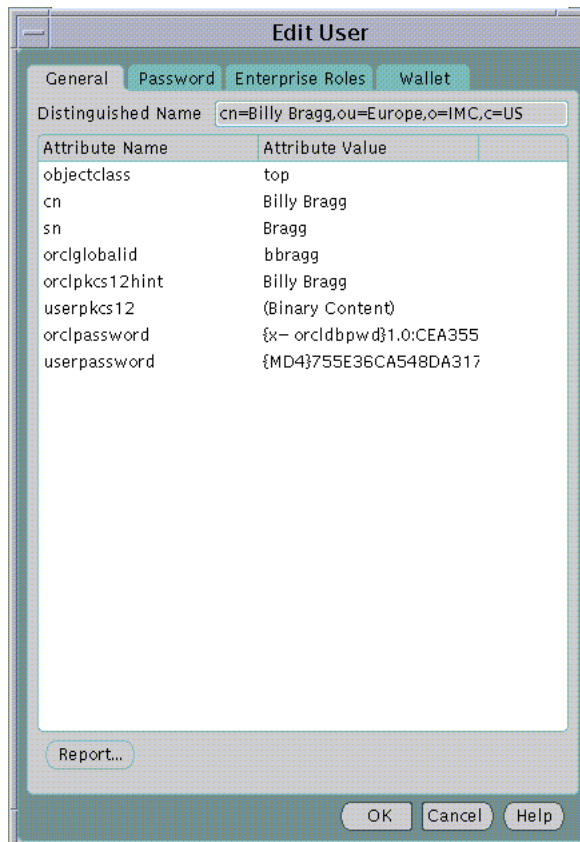
検索基準	検索への影響
Base	これは、検索が実行されるディレクトリ内のベース・エン트리・ポイントです。このベースの下にあるユーザーのみが検索結果として返されます。
Include Subtrees	この検索基準によって、選択したベースの下にあるサブツリー全体で見つかったすべてのユーザーを表示するか、またはベース位置の直下（1 レベルのみ）に存在するユーザーのみを表示するかが決まります。
Show names containing	この検索基準を指定すると、指定した文字で始まる共通名が含まれたディレクトリ・エントリを持つユーザーのみに検索対象が限定されます。これは、ターゲット・ユーザーの正確な名前またはベースがわからない場合に役立ちます。

例：

検索結果からユーザーを選択して編集します。

検索結果として返されたユーザー名の1つを編集するには、ターゲットのユーザー名を選択して「Edit...」ボタンをクリックするか、リスト上でターゲットのユーザー名をダブルクリックします (図 19-8)。

図 19-8 Oracle Enterprise Security Manager: 「Edit User」 ウィンドウ



ディレクトリ・ユーザーを選択して「Edit...」ボタンをクリックすると、パスワードとエンタープライズ・ロールの割当てを変更できます。また、初期作成時と同じ方法でユーザー Wallet を変更できます。

関連項目：

- 19-6 ページ「[エンタープライズ・ユーザーの新規作成](#)」
- 19-13 ページ「[ディレクトリ内のユーザーのブラウズ](#)」

データベース・アクセスを使用可能にする方法

ユーザー・エントリは、すでに Oracle データベースへのアクセスが許可されているユーザーのディレクトリ・サブツリー内に存在する必要があります。選択したサブツリーに対して、Oracle データベースへのアクセス権を設定できます。これにより、パスワード・アクセシブル・ドメイン・グループに属するドメイン内のデータベースがユーザーのログイン資格証明を読み込むことができるようになります。

データベースへのアクセスを可能にする手順は、次のとおりです。

選択したディレクトリ・ユーザーのサブツリーで、Oracle データベースへのアクセス権を設定し、パスワード・アクセシブル・ドメイン・グループに属するデータベースがユーザーのデータベース・ログイン資格証明にアクセスできるようにします。

- 「Users, by Search Base」の下にあるターゲット・ユーザーのサブツリーを選択します。
- 「Allow logon to Databases in Authorized Enterprise Domains」を選択します。

Oracle コンテキストの管理

Oracle コンテキストとはディレクトリ内の1つのサブツリーであり、ディレクトリを使用するインストール済みの Oracle 製品で使用されるデータが含まれています。Oracle Enterprise Security Manager は、ディレクトリを使用する製品の1つです。Oracle Enterprise Security Manager を使用すると、ディレクトリ内のデータベースとセキュリティ関連情報を Oracle コンテキストで管理できます。

注意： Oracle コンテキスト内にユーザーを作成しないことをお勧めします。

関連項目： [第 15 章「エンタープライズ・ユーザー・セキュリティの管理」](#)

Oracle コンテキストのバージョン

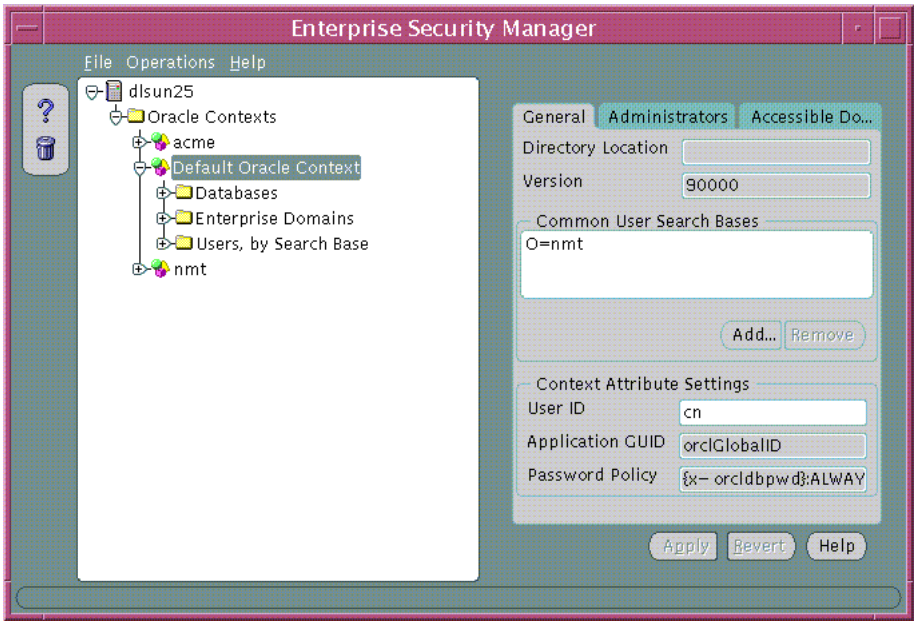
Oracle Enterprise Security Manager では、1つのディレクトリ内で、Oracle8i バージョンや Oracle9i バージョンなどの複数の Oracle コンテキストをサポートできます。ただし、Oracle9i のエンタープライズ・ユーザー・セキュリティを管理できるのは、Oracle9i の Oracle コンテキストを使用している場合のみです。Oracle9i 用の Oracle Enterprise Security Manager を使用すると、ディレクトリ内のバージョン 9i の Oracle コンテキストとバージョン 8i の Oracle コンテキストを管理できます。

Oracle Enterprise Security Manager のメイン・アプリケーション・ツリーには、Oracle8i と Oracle9i の各バージョンを含む既存の Oracle コンテキストがすべて表示されます。次の例 (図 19-9) では、Oracle Enterprise Security Manager を使用して、Oracle9i のディレクトリ・スキーマと Oracle9i のルート Oracle コンテキストをサポートするよう構成されている Oracle ディレクトリに接続しています。

Oracle コンテキストのプロパティの定義

Oracle コンテキストには、「Enterprise Security Manager」ウィンドウで表示および管理できるいくつかのプロパティがあります（[図 19-9](#)、[表 19-4](#)）。

図 19-9 Oracle Enterprise Security Manager: 「General」 タブ



注意： [図 19-9](#) で Default Oracle Context と表示されている部分は、Root Oracle Context に読み替えてください。

Oracle コンテキストのプロパティを理解するには、[表 19-4](#) を参照してください。

表 19-4 Oracle コンテキストのプロパティ

プロパティ	説明
Directory Location	Oracle コンテキストの親。ルート Oracle コンテキストの場合は、コンテキストがディレクトリ・ツリーのルートにあるので、この値は空になります。
Version	Oracle コンテキストのバージョンが、Oracle8i と Oracle9i のいずれであるかを定義します。

表 19-4 Oracle コンテキストのプロパティ（続き）

プロパティ	説明
Versioncompatibility	Oracle コンテキストが、Oracle8i、Oracle9i またはその両方のいずれをサポートするかを定義します。
Common User Search Bases	ユーザーが通常存在するディレクトリ内のベース位置のリスト。ユーザー検索ベースのリストを指定すると、そのディレクトリ位置でユーザーを迅速にブラウズできます。また、Oracle コンテキスト内の Oracle9i データベースに対して、そのデータベースに接続するディレクトリ・ユーザーを検索する場所を指示できます。
UserID	UserID 属性は、エンタープライズ内のユーザーを一意に識別します。つまり、各ユーザーのグローバルに一意な識別子です。ユーザーは、UserID 属性の値を使用して、Oracle9i データベース、ディレクトリ・サービスまたはディレクトリ対応アプリケーションに対する認証を行います。デフォルト値は cn（ディレクトリ・ユーザーの共通名）です。
Application GUID	一意のアプリケーション GUID の値が存在するユーザー・エントリ内の属性の名前。このリリースでは、このプロパティは変更できません。
Password Policy	パスワード認証方式のグローバル・ユーザーを認証する際に Oracle9i データベースで使用されるパスワード・ポリシーの構文。このリリースでは、このプロパティは変更できません。

データベースのディレクトリへの登録

Oracle Enterprise Security Manager を使用したデータベースのディレクトリへの登録は、このリリースの新機能です。Database Configuration Assistant を使用して、データベースをディレクトリに登録することもできます。表 19-5 は、これら 2 種類の Oracle ツール製品の使用上の相違点を示しています。

表 19-5 Oracle Enterprise Security Manager および Database Configuration Assistant によるデータベースのディレクトリへの登録における相違点

Oracle のツール製品	ディレクトリ へのデータ ベース DN エントリの 作成	デフォルト・ ドメインへの データベース の追加	ディレクトリ へのプレース ホルダ・ データベース Wallet の作成	RDBMS_ SERVER_DN パラメータ の設定	有効な データベース Wallet の 作成
Oracle Enterprise Security Manager	○	○	○	×	×
Database Configuration Assistant	○	○	×	○	×

関連項目： データベースのディレクトリへの登録方法は、15-30 ページの「データベースのディレクトリへの登録」を参照してください。

前提条件

プレースホルダ・データベース Wallet を生成する場合は、最初にコマンドラインで次のツールを実行する必要があります。

```
esm -genca
```

表示されるプロンプトに従います。このツールによって、Oracle Wallet ディレクトリに擬似認証局が作成されます。

データベースのディレクトリへの登録手順：

1. 「Enterprise Security Manager」メイン・ウィンドウの「Operations」メニューから「Register Database」を選択します。「Database Registration」ウィンドウが表示されます。
2. 登録するデータベースについて、各フィールドに適切な値を入力します。Oracle Enterprise Security Manager を使用してデータベースを登録する場合は、データベースの SID とデータベース短縮名が等しい必要があることに注意してください。

接続文字列の編集が必要な場合は、「Store TNS Connect String」を選択してフィールドを編集可能にします。

3. 登録しているデータベースにプレースホルダ **Wallet** を生成する場合は、「**Generate Wallet**」を選択して **Wallet** のパスワードを入力します。

「**Generate Wallet**」オプションが表示されない場合は、19-20 ページの「**前提条件**」に説明されている **esm -genca** ツールを実行済みであることを確認します。
4. すべての情報を記入後、「**OK**」をクリックしてデータベース・エントリをディレクトリに作成します。
5. **SQL*Plus** プロンプトで次のコマンドを入力することによって、サーバー・パラメータ・ファイル (**spfile.ora**) の **RDBMS_SERVER_DN** パラメータの設定を説明するダイアログ・ボックスが表示されます。


```
ALTER SYSTEM SET RDBMS_SERVER_DN=SERVER_DN SCOPE SPFILE
```
6. このコマンドを入力後、新規パラメータ設定をシステムに読み込むためにデータベースを再起動します。

ユーザー検索ベースの定義

「**General**」タブ・ウィンドウ (図 19-9) を使用して、Oracle9i の Oracle コンテキストに共通ユーザー検索ベースを追加または削除できます。

注意： この機能は、Oracle8i の Oracle コンテキストでは使用できません。

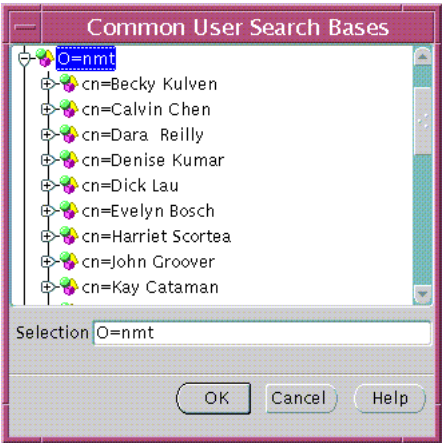
Oracle コンテキストからユーザー検索ベースを削除する手順は、次のとおりです。

1. Oracle Enterprise Security Manager の「**General**」タブ・ウィンドウ (図 19-9) を使用して、「**Common User Search Bases**」リストから検索ベースを選択し、「**Remove...**」ボタンをクリックします。
2. 「**Apply**」ボタンをクリックします。ディレクトリ内の Oracle コンテキストからユーザー検索ベースが削除されます。

Oracle コンテキストに新規のユーザー検索ベースを追加する手順は、次のとおりです。

1. Oracle Enterprise Security Manager の「**General**」タブ・ウィンドウ (図 19-9) で、「**Add...**」ボタンをクリックします。「**Browse Directory**」ウィンドウが表示されます (図 19-10)。

図 19-10 Oracle Enterprise Security Manager: ディレクトリのブラウズ（ユーザー検索ベース）



- 2. ディレクトリ・ツリーをナビゲートして、ユーザー検索ベースのエントリを選択します。また、「Browse Directory」ウィンドウの「Selection」フィールドの内容を編集して、ユーザー検索ベースを手動で定義することもできます。
- 3. 「OK」をクリックします。選択したエントリが、「General」タブ・ウィンドウ（図 19-9）のユーザー検索ベースのリストに追加されます。
- 4. 「Apply」ボタンをクリックします（図 19-9）。ディレクトリ内の Oracle コンテキストにユーザー検索ベースが追加されます。

Oracle コンテキスト管理者の定義

Oracle コンテキストには、Oracle コンテキスト内での操作に対して様々なレベルの権限を持つ管理グループが含まれています。管理グループには、Oracle9i の Oracle コンテキストのみで使用できるものと、Oracle8i と Oracle9i の両方の Oracle コンテキストで使用できるものがあります。Oracle コンテキストの管理グループの定義を表 19-6 に示します。

表 19-6 Oracle コンテキスト管理者

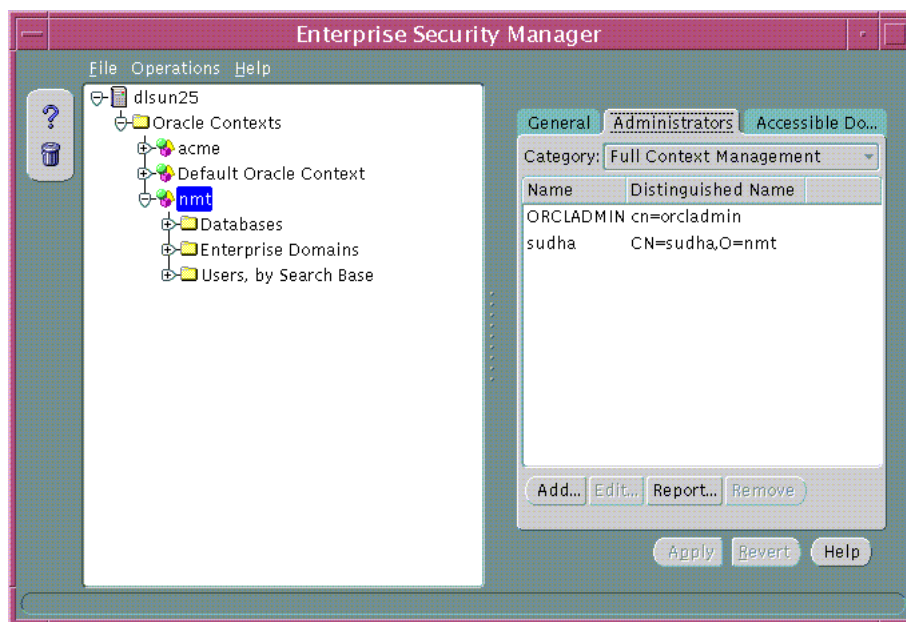
管理グループ	定義	Oracle9i バージョン	Oracle8i バージョン
Full Context Management	Oracle コンテキスト内のすべての製品領域に対して有効なすべての管理者権限を持ちます。	○	×
Directory User Management	ディレクトリ・ユーザーのパスワード・リマインダを表示し、パスワードを更新できます。	○	×

表 19-6 Oracle コンテキスト管理者（続き）

管理グループ	定義	Oracle9i バージョン	Oracle8i バージョン
Database Security Management	Oracle コンテキスト内のすべてのエンタープライズ・ドメインとロールを管理できます。	○	○
Database Registration	Oracle コンテキストに新規データベースを登録できます。	○	○
Oracle Net Management	Oracle コンテキスト内の Oracle Net オブジェクトを管理できます。	○	○

Oracle コンテキスト管理者を管理するには、Oracle Enterprise Security Manager のメイン・ウィンドウの「Administrators」タブを使用します（表 19-11）。

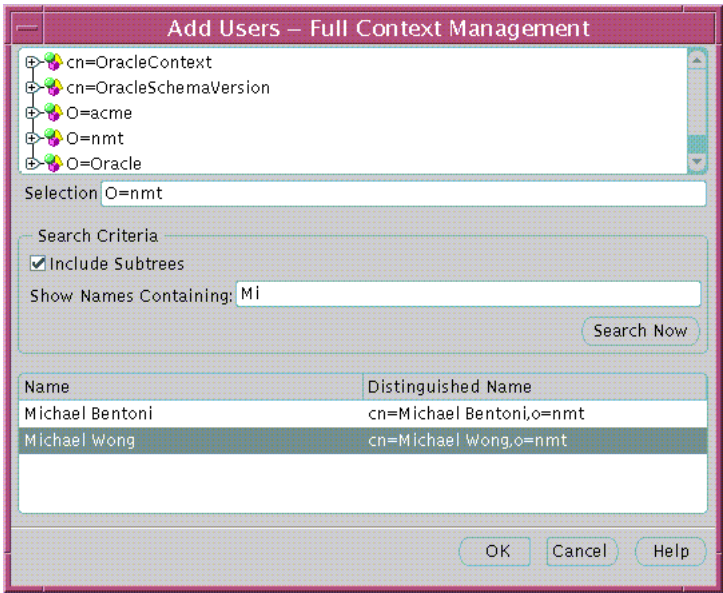
図 19-11 Oracle Enterprise Security Manager の「Administrators」タブ



Oracle コンテキスト管理者のリストからユーザーを削除する手順は、次のとおりです。

1. 管理者の「Category」（表 19-6）を選択します。このカテゴリ内に管理者のリストが表示されます。
2. リストからユーザー名を選択します。
3. 「Remove」ボタンをクリックします。選択したユーザーがリストから削除されます。
4. 「Apply」ボタンをクリックします。選択したユーザーが、Oracle コンテキスト管理者の管理者カテゴリから削除されます。

図 19-12 Oracle Enterprise Security Manager: 「Add Users」ウィンドウ



Oracle コンテキスト管理者のリストに新規ユーザーを追加する手順は、次のとおりです。

1. 図 19-11 のウィンドウで、「Add...」ボタンをクリックします。「Add Users」ウィンドウが表示されます（図 19-12）。

このウィンドウを使用して、ディレクトリのユーザーの位置を特定し、選択します。「Add Users」ウィンドウには、次の 3 つのパネルがあります。

- **上部パネル:** ディレクトリ検索ツリー。
- **中央パネル:** 検索結果として返されるユーザーを指定するための検索基準。
- **下部パネル:** 検索結果—ディレクトリ内で見つかった検索基準に一致しているユーザー。

2. 上部パネルでディレクトリをナビゲートして、ユーザー検索ベースとなるディレクトリ・エントリを選択します。このウィンドウの「**Selection**」フィールドの内容を編集して、ユーザー検索ベースを手動で定義することもできます。
3. 中央パネル（「**Search Criteria**」）で、「**Include Subtrees**」オプションを選択します。このオプションを選択すると、サブツリーも含めて、検索ベース内のすべてのユーザーが検索対象になります。
4. 「**Show Names Containing**」フィールドに、既知のユーザー名を入力します。このユーザー名に一致するユーザーが検索結果として返されます。この検索基準を指定すると、指定した文字と完全に一致する共通名またはそれらの文字で始まる共通名が含まれたディレクトリ内のユーザーのみに検索対象が限定されます。
5. 「**Search Now**」ボタンをクリックします（中央パネル）。選択したベース内に選択基準と一致するディレクトリ・ユーザーが存在する場合は、そのユーザーがウィンドウにリストされます。
6. 対象のユーザー名をリストから選択して「**OK**」をクリックするか、またはリスト上でダブルクリックして、そのユーザー名を選択します。ユーザーの範囲を選択して「**OK**」をクリックすると、リストから複数のユーザーを選択できます。選択したカテゴリの管理者リストに新規ユーザーが表示されます。

注意： このウィンドウは、ディレクトリからユーザーを選択する必要がある場合に、Oracle Enterprise Security Manager 全体を通じて共通に使用されます。

パスワード・アクセシブル・ドメインの管理

データベースでパスワード認証ユーザーからの接続を受け入れるには、次の3つの要件を満たす必要があります。

- データベースがパスワードと SSL 認証を受け入れるように構成されているドメインのメンバーであることが必要です（表 19-8 を参照してください）。
- ドメインが、パスワード・アクセシブル・ドメイン・グループ（**パスワード・アクセシブル・ドメイン・リスト**と呼ばれる）のメンバーであることが必要です。この作業は、Oracle コンテキスト管理者またはデータベース・セキュリティ管理者が行います。このリストのドメイン・メンバーは、ディレクトリ内のユーザーのパスワード・ベリファイアを読み込むことができます。このリストに含まれていないドメインは、読み込むことができません。また、ドメインは、Oracle9i 以上の Oracle コンテキストの一部である必要があります。
- ユーザー・エントリは、すでに Oracle データベースへのアクセスが可能であるユーザーのディレクトリ・サブツリー内に存在する必要があります。選択したサブツリーに対して、Oracle データベースへのアクセス権を設定できます。これにより、パスワード・アクセシブル・ドメイン・リストに属するデータベースがユーザーのログイン情報を読み込むことが可能になります。

パスワードによるアクセスを構成する手順は、次のとおりです。

1. ターゲット・データベースを、パスワードおよび SSL ユーザー認証を受け入れるように構成されているエンタープライズ・ドメインに追加します。

関連項目：

- 19-32 ページ「エンタープライズ・ドメイン内のデータベースのメンバシップの定義」
 - 19-35 ページ「エンタープライズ・ドメインで使用するデータベース・セキュリティ・オプションの管理」
2. 選択した Oracle9i 以上の Oracle コンテキストで、ドメインをパスワード・アクセシブル・ドメイン・リストに追加します。「Add」ボタンをクリックし、表示されたダイアログから現行のエンタープライズ・ドメインの 1 つを選択します。リストからエンタープライズ・ドメインを削除するには、「Accessible Domains」ウィンドウで対象のドメインを選択して、「Remove」ボタンをクリックします。
 3. 選択したディレクトリ・ユーザーのサブツリーで、Oracle データベースへのアクセス権を設定します。これにより、パスワード・アクセシブル・ドメイン・リストに属するデータベースがユーザーのデータベース・ログイン情報にアクセスできるようになります。
 - 「Users, by Search Base」の下にあるターゲット・ユーザーのサブツリーを選択します。
 - そのサブツリーの「Allow Logon to Database in Authorized Enterprise Domain」を選択します。

関連項目： 15-12 ページ「ユーザーのデータベース・ログイン情報のセキュリティ」

注意： パスワード・アクセシブル・ドメインには、Oracle9i の Oracle コンテキストが必要です。

データベース・セキュリティの管理

データベースをディレクトリに登録した後は、Oracle Enterprise Security Manager を使用して、そのデータベースへのユーザー・アクセスを管理できます。そのためには、Oracle コンテキスト内の次のオブジェクトを使用します (表 19-7)。

表 19-7 Oracle Enterprise Security Manager: Oracle コンテキストのオブジェクト

オブジェクト	説明
データベース	登録されたデータベースを表すディレクトリ・エントリ。
エンタープライズ・ドメイン	ディレクトリに登録されているデータベースのグループ。このグループに対して、データベース・セキュリティの共通ユーザー・アクセス・モデルを実装できます。
エンタープライズ・ロール	エンタープライズ・ドメイン 内の複数のデータベースにわたる認可。 エンタープライズ・ロール には、エンタープライズ・ドメイン内の各データベース上にある個々のロールを付与できます。
マッピング	マッピング・オブジェクトを使用して、ユーザーの 識別名 を、そのユーザーがアクセスするデータベース・スキーマにマップします。

関連項目：

- 『Oracle9i データベース管理者ガイド』
- [第 15 章「エンタープライズ・ユーザー・セキュリティの管理」](#)
- [第 19 章「Oracle Enterprise Security Manager の使用方法」](#)

データベース管理者の管理

データベース管理者とは、Oracle コンテキスト内のデータベースやそのサブツリーを変更する権限を与えられたディレクトリ・ユーザーです。データベース管理者を管理するには、メイン・アプリケーション・ツリーの Oracle コンテキストの下でデータベースを選択してから (図 19-11)、「Administrators」タブ・ウィンドウを使用します。

データベース管理者のリストからユーザーを削除する手順は、次のとおりです。

1. 管理者のリストからユーザーを選択します。
2. 「Remove」ボタンをクリックします。選択したユーザーがリストから削除されます。
3. 「Apply」ボタンをクリックします。選択したユーザーが、Oracle コンテキストで定義されているデータベースのデータベース管理者から削除されます。

データベース管理者のリストに新規ユーザーを追加する手順は、次のとおりです。

1. 「Add」ボタンをクリックします。「Add Users」ウィンドウが表示されます（[図 19-12](#)）。このウィンドウを使用して、ディレクトリのユーザーの位置を特定し、選択します。
2. ディレクトリから、データベース管理者として追加するユーザー（複数も可）を選択します。「Administrators」タブ・ウィンドウに新規ユーザーが表示されます（[図 19-11](#)）。
3. 「Apply」ボタンをクリックします。Oracle コンテキスト内のデータベースに新規の管理者が追加されます。

関連項目：

- 19-6 ページ [「エンタープライズ・ユーザーの新規作成」](#)
- 19-13 ページ [「ディレクトリ内のユーザーのブラウズ」](#)

データベース・スキーマ・マッピングの管理

データベース・[スキーマ・マッピング](#)を使用すると、ディレクトリに登録されているデータベースがユーザーからの接続を受け入れる際に、そのユーザー専用のデータベース・[スキーマ](#)が不要になります。たとえば、ローカル・ユーザー Scott がデータベースに接続し、正常にログインするには、Scott というデータベース・スキーマが存在する必要があります。しかし、何千ものユーザーが数百のデータベースを使用している大規模な企業では、このようにデータベース・スキーマを維持することは困難です。

LDAP 準拠のディレクトリに定義されているユーザーには、接続する Oracle8i 以上のデータベース上に専用のスキーマは必要ありません。

データベースでは、スキーマ・マッピングを使用して、複数のディレクトリ・ユーザー間で 1 つのデータベース・スキーマを共有できます。スキーマ・マッピングは、ユーザーが存在するディレクトリ内のベースと、そのユーザーが使用するデータベース・スキーマの名前という 2 つの値のペアです。

データベース・スキーマ・マッピングを管理するには、メイン・アプリケーション・ツリーの Oracle コンテキストの下でデータベースを選択してから、「Database Schema Mappings」タブ・ウィンドウを使用します。このウィンドウには、データベース・スキーマ名とディレクトリ・ベースのペアのリストが表示されます（[図 19-13](#)）。

図 19-13 Oracle Enterprise Security Manager: 「Database Schema Mappings」 タブ



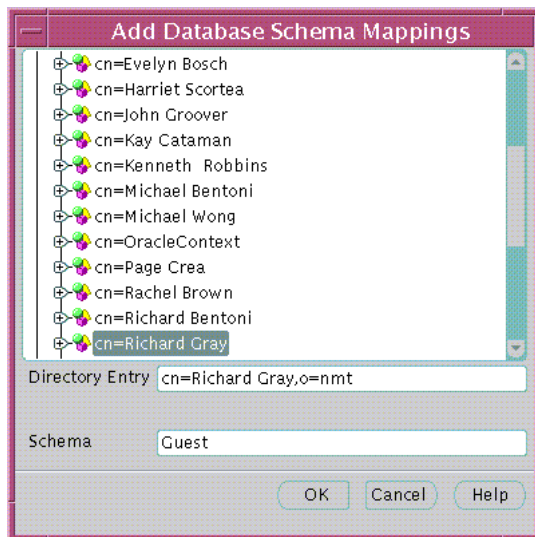
エンタープライズ・ドメイン内のデータベース・スキーマ・マッピングのリストからマッピングを削除する手順は、次のとおりです。

1. 「Database Schema Mappings」タブ・ウィンドウで、マッピングを選択します。
2. 「Remove」ボタンをクリックします。選択したマッピングがリストから削除されます。
3. 「Apply」ボタンをクリックします。選択したマッピングがエンタープライズ・ドメインから削除されます。

エンタープライズ・ドメイン内のデータベース・スキーマ・マッピングのリストに新規のマッピングを追加する手順は、次のとおりです。

1. 「Add...」ボタンをクリックします。「Add Database Schema Mappings」ウィンドウが表示されます (図 19-14)。

図 19-14 Oracle Enterprise Security Manager: 「Add Database Schema Mappings」 ウィンドウ



このウィンドウを使用して、ディレクトリ内のベースの位置を特定して選択し、それとデータベース・スキーマ名をペアにして、データベース・スキーマ・マッピングを作成します。このウィンドウには2つの構成要素があります。ベースを選択するためのディレクトリ検索ツリーと、スキーマ名を入力するフィールドです。

2. ディレクトリをナビゲートして、データベース・スキーマ・マッピングに使用するベースとして必要なエントリを選択します。任意のディレクトリ・エントリを選択できますが、マップするユーザーのサブツリーより上に位置する必要があります。また、このウィンドウの「Directory Entry」フィールドの内容を編集して、ベースを手動で定義することもできます。
3. 「Schema」フィールドに、このマッピングを構成するデータベース・スキーマの名前を入力し、「OK」をクリックします。このスキーマ名は、対象のデータベース上にすでに存在しているスキーマの有効な名前であることが必要です。「Database Schema Mappings」ウィンドウに新規のデータベース・スキーマ・マッピングが表示されます(図 19-13)。
4. 「Apply」ボタンをクリックします。Oracle コンテキスト内の選択したデータベースに新規のデータベース・スキーマ・マッピングが追加されます。

エンタープライズ・ドメインの管理

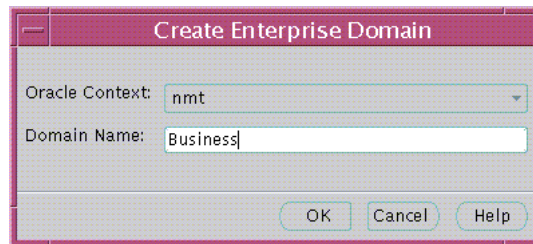
Oracle コンテキストには、OracleDefaultDomain と呼ばれるエンタープライズ・ドメインが少なくとも 1 つ含まれています。OracleDefaultDomain は、Oracle コンテキストがディレクトリ内に最初に作成されたときから、Oracle コンテキストの一部として存在しています。Oracle コンテキストに新規データベースを登録すると、そのデータベースは、自動的に Oracle コンテキスト内の OracleDefaultDomain のメンバーになります。独自に作成したエンタープライズ・ドメインを削除することは可能ですが、Oracle コンテキストから OracleDefaultDomain を削除することはできません。

Oracle コンテキストに新規のエンタープライズ・ドメインを作成するには、次の方法のいずれかを使用します。

- 「Operations」メニューから「Create Enterprise Domain」を選択します (図 19-13)。
- メイン・アプリケーション・ツリーから Oracle コンテキストを選択し、マウスの右ボタンをクリックします。

「Create Enterprise Domain」ウィンドウが表示されます (図 19-15)。

図 19-15 Oracle Enterprise Security Manager: 「Create Enterprise Domain」ウィンドウ



新規のエンタープライズ・ドメインを作成する手順は、次のとおりです。

1. ドロップダウン・リストから適切な Oracle コンテキストを選択します (図 19-15)。

注意： メイン・アプリケーション・ツリーで Oracle コンテキストを右クリックして「Create Enterprise Domain」ウィンドウを起動した場合は、Oracle コンテキストの名前がすでに選択されています。

2. 「Domain Name」フィールドに、新規のエンタープライズ・ドメインの名前を入力します。
3. 「OK」をクリックします。Oracle コンテキストに新規のエンタープライズ・ドメインが作成され、メイン・アプリケーション・ツリーに表示されます。

エンタープライズ・ドメインを削除する手順は、次のとおりです。

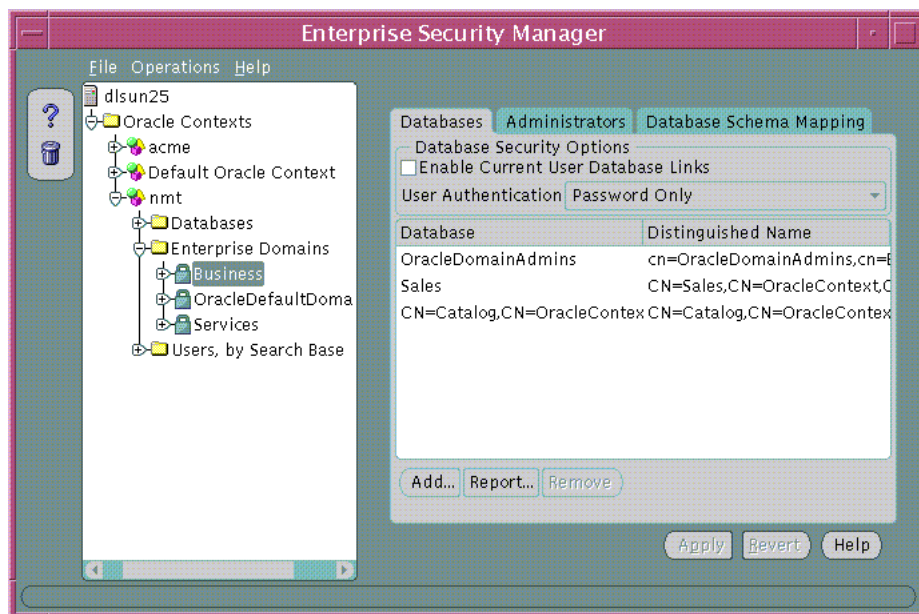
1. メイン・アプリケーション・ツリーからターゲットのエンタープライズ・ドメインを選択します (図 19-13)。
2. 次の方法のいずれかを使用します。
 - 「Operations」メニューから「Remove Enterprise Domain」を選択します。
 - メイン・アプリケーション・ツリーからエンタープライズ・ドメインを選択し、マウスの右ボタンをクリックします。
3. Oracle コンテキストからエンタープライズ・ドメインを削除することを確認するダイアログが表示されます。「OK」をクリックすると、エンタープライズ・ドメインが削除されます。

注意： エンタープライズ・ドメインにエンタープライズ・ロールが含まれている場合は、そのエンタープライズ・ドメインを Oracle コンテキストから削除することはできません。

エンタープライズ・ドメイン内のデータベースのメンバーシップの定義

Oracle Enterprise Security Manager のメイン・ウィンドウのアプリケーション・ツリーを使用して、ターゲットのエンタープライズ・ドメインを選択します。次に、「Databases」タブを使用して、Oracle コンテキスト内に格納されているエンタープライズ・ドメインのデータベースのメンバーシップを管理します (図 19-16)。

図 19-16 Oracle Enterprise Security Manager: 「Databases」タブ (データベースのメンバーシップ)



エンタープライズ・ドメインからデータベースを削除する手順は、次のとおりです。

1. 削除するターゲット・データベースを選択して、「Remove...」ボタンをクリックします。選択したデータベースがリストから削除されます。
2. 「Apply」ボタンをクリックします。選択したデータベースが Oracle コンテキスト内のエンタープライズ・ドメインから削除されます。

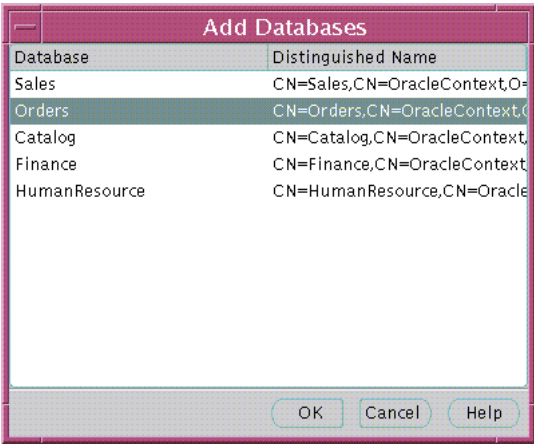
エンタープライズ・ドメインにデータベースを追加する手順は、次のとおりです。

注意： データベースとエンタープライズ・ドメインが同じ Oracle コンテキスト内に存在している場合のみ、エンタープライズ・ドメインにデータベースを追加できます。したがって、次の条件が成り立ちます。

- エンタープライズ・ドメインには、別の Oracle コンテキストのデータベースを含めることはできません。
- データベースは、2 つの異なるエンタープライズ・ドメインのメンバーとして追加することはできません。

1. 「Add...」 ボタンをクリックします (図 19-16)。「Add Databases」 ウィンドウが表示されます。このウィンドウには、Oracle コンテキストに対応付けられているデータベースがすべてリストされます (図 19-17)。

図 19-17 Oracle Enterprise Security Manager: 「Add Databases」 ウィンドウ



2. エンタープライズ・ドメインに追加する新規のターゲット・データベースを選択します。
3. 「OK」 をクリックします。選択したデータベースが、「Databases」 タブ・ウィンドウ (図 19-16) のデータベースのリストに追加されます。
4. 「Apply」 ボタンをクリックします (図 19-16)。Oracle コンテキスト内のエンタープライズ・ドメインに新規のデータベースが追加されます。

エンタープライズ・ドメインで使用するデータベース・セキュリティ・オプションの管理

「Databases」タブ・ウィンドウ (図 19-16) を使用して、エンタープライズ・ドメインに属するすべてのデータベースに適用されるデータベース・セキュリティ・オプションを管理します。

表 19-8 は、データベース・セキュリティ・オプションをまとめたものです。

表 19-8 Oracle Enterprise Security Manager のデータベース・セキュリティ・オプション

データベース・セキュリティ・オプション	説明
Enable current user database links	この設定が有効なエンタープライズ・ドメイン内に、両方のデータベースが存在している場合にかぎり、それらのデータベースの間でカレント・ユーザー・データベース・リンクの使用を許可できます。
User authentication	エンタープライズ・ドメイン内のすべてのデータベースで、次のいずれかのタイプのクライアント認証が行われます。 <ul style="list-style-type: none">■ Oracle Wallet を使用した Oracle Net SSL 認証のみ■ パスワード認証または Oracle Net SSL 認証 (デフォルト)

エンタープライズ・ドメイン管理者の管理

エンタープライズ・ドメイン管理者とは、そのドメインの内容を変更する権限を持つディレクトリ・ユーザーです。エンタープライズ・ドメイン管理者を管理するには、メイン・アプリケーション・ツリーの Oracle コンテキストの下でエンタープライズ・ドメインを選択してから、「Administrators」タブ・ウィンドウを使用します (図 19-11)。

エンタープライズ・ドメイン管理者のリストからユーザーを削除する手順は、次のとおりです。

1. 管理者のリストからユーザーを選択します。
2. 「Remove」ボタンをクリックします。選択したユーザーがリストから削除されます。
3. 「Apply」ボタンをクリックします。選択したユーザーが、Oracle コンテキストで定義されているそのドメインのエンタープライズ・ドメイン管理者から削除されます。

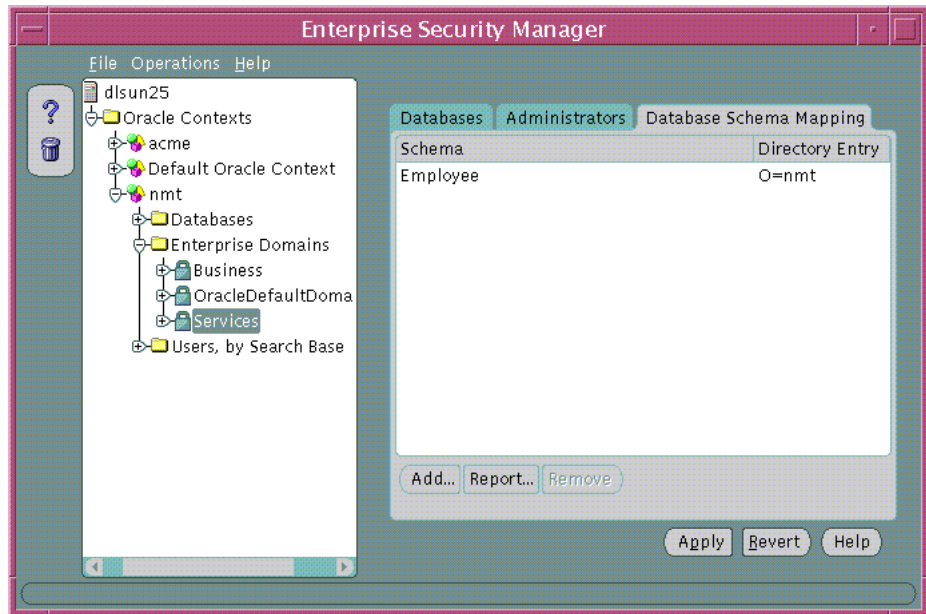
エンタープライズ・ドメイン管理者のリストに新規ユーザーを追加する手順は、次のとおりです。

1. 「Add...」 ボタンをクリックします。「Add Users」 ウィンドウが表示されます。このウィンドウを使用して、エンタープライズ・ドメイン管理者として指定するターゲット・ユーザーの位置を特定し、選択します。「Administrators」 タブ・ウィンドウに、新規ユーザーが表示されます。
2. 「Apply」 ボタンをクリックします。Oracle コンテキスト内のエンタープライズ・ドメインに新規の管理者が追加されます。

エンタープライズ・ドメイン・データベース・スキーマ・マッピングの管理

前述したように、データベース・スキーマ・マッピングは、Oracle コンテキスト内のデータベースごとに管理できます。スキーマ・マッピングは、Oracle コンテキスト内のエンタープライズ・ドメインごとに定義することもできます。そのためには、メイン・アプリケーション・ツリーでエンタープライズ・ドメインを選択してから、「Database Schema Mappings」 タブ・ウィンドウを使用します。このようにして定義したマッピングは、エンタープライズ・ドメインのメンバーであるすべてのデータベースに適用されます。したがって、エンタープライズ・ドメイン内の各データベースでそのマッピングを有効にするために、それぞれのデータベースにマッピングで使用されているものと同じ名前のスキーマを作成する必要があります。

図 19-18 Oracle Enterprise Security Manager: 「Database Schema Mappings」 タブ



エンタープライズ・ドメイン内のデータベース・スキーマ・マッピングのリストからマッピングを削除する手順は、次のとおりです (図 19-18)。

1. 「Database Schema Mappings」のリストからマッピングを選択します。
2. 「Remove」ボタンをクリックします。選択したマッピングがリストから削除されます。
3. 「Apply」ボタンをクリックします。選択したマッピングがエンタープライズ・ドメインから削除されます。

エンタープライズ・ドメイン内のデータベース・スキーマ・マッピングのリストに新規のマッピングを追加する手順は、次のとおりです (図 19-18)。

1. 「Add...」ボタンをクリックします。「Add Database Schema Mappings」ウィンドウが表示されます。前述した手順と同様に、このウィンドウを使用して、新規マッピングに使用するディレクトリ内のベースの位置を特定し、選択します。
2. エンタープライズ・ドメインにマップする新規のデータベース・スキーマを入力します。
3. 「Apply」ボタンをクリックします。Oracle コンテキスト内の選択したエンタープライズ・ドメインに新規のデータベース・スキーマ・マッピングが追加されます。

関連項目：

- 19-28 ページ「データベース・スキーマ・マッピングの管理」
- 19-8 ページ「ディレクトリ・ベースの定義」

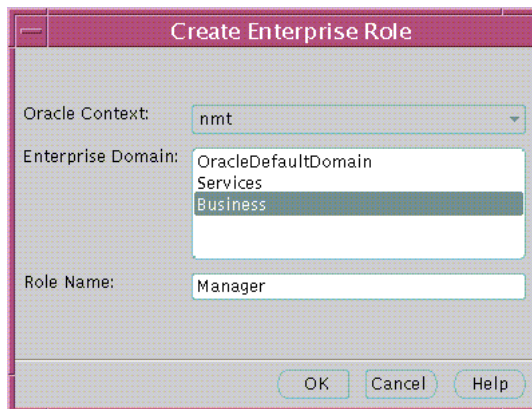
エンタープライズ・ロールの管理

Oracle コンテキスト内のエンタープライズ・ドメインには、複数のエンタープライズ・ロールを含めることができます。エンタープライズ・ロールとは、エンタープライズ・ドメイン内の 1 つ以上のデータベースに關係する Oracle ロール・ベースの認可のセットです。

エンタープライズ・ロールを新規に作成する手順は、次のとおりです。

エンタープライズ・ドメイン内にエンタープライズ・ロールを作成するには、Oracle Enterprise Security Manager のメイン・ウィンドウ (図 19-18) の「Operations」メニューを使用するか、またはメイン・アプリケーション・ツリーでエンタープライズ・ドメインを右クリックします。どちらの方法を使用した場合でも、「Create Enterprise Role」ウィンドウが表示されます (図 19-19)。

図 19-19 Oracle Enterprise Security Manager: 「Create Enterprise Role」ウィンドウ



1. 「Oracle Context」ドロップダウン・リストからターゲットの Oracle コンテキストを選択します。これは、ターゲットのエンタープライズ・ドメイン（新規のエンタープライズ・ロールを保持するエンタープライズ・ドメイン）を含む Oracle コンテキストです。

注意： メイン・アプリケーション・ツリーでエンタープライズ・ドメインを右クリックして「Create Enterprise Role」ウィンドウを起動した場合は、Oracle コンテキストの名前がすでに選択されています。

2. 「Enterprise Domain」リストから、新規のエンタープライズ・ロールを使用する適切なエンタープライズ・ドメインを選択します。

注意： メイン・アプリケーション・ツリーでエンタープライズ・ドメインを右クリックして「Create Enterprise Role」ウィンドウを起動した場合は、エンタープライズ・ドメインの名前がすでに選択されています。

3. 「Role Name」フィールドに、新規のエンタープライズ・ロールの名前を入力します。
4. 「OK」をクリックします。エンタープライズ・ドメイン内に新規のエンタープライズ・ロールが作成され、メイン・アプリケーション・ツリーに表示されます。

エンタープライズ・ロールを削除する手順は、次のとおりです。

1. メイン・アプリケーション・ツリーからターゲットのエンタープライズ・ロールを選択します (図 19-18)。
2. 「Operations」メニューを使用するか、メイン・アプリケーション・ツリーでエンタープライズ・ドメインを右クリックして、「Remove Enterprise Role」を選択します。
3. エンタープライズ・ロールの削除を確認するダイアログが表示されます。「Yes」をクリックします。

エンタープライズ・ロールへのデータベース・グローバル・ロール・メンバーシップの割当て

Oracle Enterprise Security Manager のメイン・ウィンドウの「Database Global Roles」タブ・ウィンドウ（図 19-20）を使用して、エンタープライズ・ロール内のデータベース・グローバル・ロール・メンバーシップを管理します。このウィンドウには、エンタープライズ・ロールに属する各**グローバル・ロール**の名前と、そのグローバル・ロールが存在するデータベースの名前がリストされます。

図 19-20 Oracle Enterprise Security Manager: 「Database Global Roles」タブ



エンタープライズ・ロールに別のデータベース・ロールを移入すると、そのデータベース・ロールがデータベース上でグローバル・ロールとして構成されている場合にかぎり参照可能になります。データベース上のグローバル・ロールは通常のロールとほぼ同じですが、**データベース管理者**によって、ディレクトリを経由しないかぎり認可できないように定義されている点が異なります。データベース管理者は、データベースのユーザーに対してグローバル・ロールをローカルに付与したり、取り消したりすることはできません。

エンタープライズ・ロールからデータベース・グローバル・ロールを削除する手順は、次のとおりです。

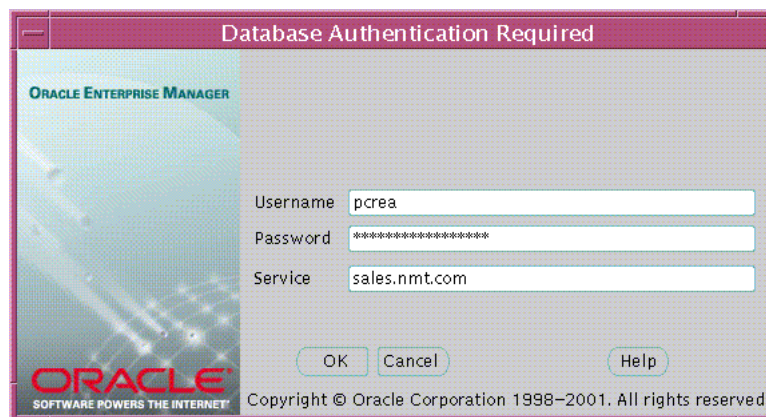
1. メイン・アプリケーション・ツリーのリストからグローバル・ロールを選択して、「Remove...」ボタンをクリックします。選択したグローバル・ロールがリストから削除されます。

2. 「Apply」 ボタンをクリックします。選択したグローバル・ロールがエンタープライズ・ドメイン内のエンタープライズ・ロールから削除されます。

エンタープライズ・ロールにグローバル・ロールを追加する手順は、次のとおりです。

1. 「Add...」 ボタンをクリックします (図 19-20)。「Add Global Database Roles」 ウィンドウが表示されます。このウィンドウには、エンタープライズ・ドメイン内のデータベースがすべてリストされます。これらのデータベースからグローバル・ロールを選択して、エンタープライズ・ロールに追加できます。
2. グローバル・ロールの取得元のデータベースを選択します。データベースに対する認証 (およびグローバル・ロールのフェッチ) を行うために、ログイン情報の入力を求めるウィンドウが表示されます。通常は、そのデータベースの DBA のログイン情報を入力します。

図 19-21 Oracle Enterprise Security Manager: 「Database Authentication Required」 ウィンドウ



注意： データベースの名前は、デフォルトで「Service」フィールドに表示されます。この名前を使用してデータベースに接続できるのは、ORACLE_HOME の Oracle Net ネーミング・メソッドとして LDAP が使用可能になっている場合、またはローカルの Oracle Net 構成でこの名前が TNS 別名として定義されている場合です。いずれのケースにも該当しない場合は、「Service」フィールドの内容を、そのデータベース用に構成されている他の TNS 別名に変更するか、または次の書式の接続文字列を使用します。

<host>:<port>:<oracle sid>

例:cartman:1521:broncos

3. 「OK」をクリックします。所定のデータベースに接続され、そのデータベースでサポートされているグローバル・ロールのリストがフェッチされます。値リストが存在する場合は、「Add Global Database Roles」ウィンドウにそのリストが表示されます。
4. 返されたリストから 1 つ以上のグローバル・ロールを選択して、「OK」をクリックします。「Database Global Roles」タブ・ウィンドウに、選択したグローバル・ロールが表示されます (図 19-20)。
5. 「Apply」ボタンをクリックします。新規のグローバル・ロールがエンタープライズ・ドメイン内のエンタープライズ・ロールに追加されます。

エンタープライズ・ロール権限受領者の管理

エンタープライズ・ロール権限受領者とは、エンタープライズ・ロールが付与されるディレクトリ・ユーザーです。付与されるロールには、そのエンタープライズ・ロールに属するすべてのデータベース・グローバル・ロールが含まれます。エンタープライズ・ロール権限受領者を管理するには、メイン・アプリケーション・ツリーのエンタープライズ・ドメインの下でエンタープライズ・ロールを選択してから、「Enterprise Users」タブ・ウィンドウを使用します (図 19-22)。

エンタープライズ・ロール権限受領者のリスト (図 19-22) からユーザーを削除する手順は、次のとおりです。

1. 権限受領者のリストからユーザーを選択します。
2. 「Remove」ボタンをクリックします。選択したユーザーがリストから削除されます。
3. 「Apply」ボタンをクリックします。選択したユーザーが、エンタープライズ・ドメインで定義されているエンタープライズ・ロールの権限受領者から削除されます。

エンタープライズ・ロール権限受領者のリストに新規ユーザーを追加する手順は、次のとおりです。

- 1. 「Add...」 ボタンをクリックします。「Add Users」 ウィンドウが表示されます (図 19-12)。このウィンドウを使用して、エンタープライズ・ロール権限受領者として追加する 1 人以上のディレクトリ・ユーザーの位置を特定し、選択します。「Enterprise Users」 タブ・ウィンドウに、新規ユーザーが表示されます (図 19-22)。

図 19-22 Oracle Enterprise Security Manager: 「Enterprise Users」 タブ



- 2. 「Apply」 ボタンをクリックします。新規の権限受領者がエンタープライズ・ドメイン内のエンタープライズ・ロールに追加されます。

この新規に作成されたエンタープライズ・ユーザーを選択して、「Enterprise Role」タブを選択し、そのユーザーにエンタープライズ・ロールを割り当てることができます。

関連項目： 19-11 ページ「初期エンタープライズ・ロール割当ての定義」

第 VI 部

付録

第 VI 部では、参照用に次の付録を提供します。

- [付録 A「データの暗号化と整合性のパラメータ」](#)
- [付録 B「認証パラメータ」](#)
- [付録 C「RADIUS による認証デバイスの統合」](#)
- [付録 D「Oracle Advanced Security FIPS 140-1 の設定」](#)
- [付録 E「Microsoft Active Directory でのエンタープライズ・ユーザー・セキュリティの使用」](#)
- [付録 F「Java SSL の Oracle 実装」](#)
- [付録 G「略称と頭字語」](#)

データの暗号化と整合性のパラメータ

この付録では、Oracle Advanced Security でサポートしている[暗号化](#)とデータの[整合性](#)のパラメータについて説明します。この付録には、[第 2 章「データの暗号化および整合性の構成」](#)および[第 7 章「Secure Sockets Layer 認証の構成」](#)で説明したネットワーク構成の実行によって生成された `sqlnet.ora` ファイルの例も含まれています。

項目は、次のとおりです。

- サンプル `sqlnet.ora` ファイル
- データの暗号化と整合性のパラメータ

サンプル sqlnet.ora ファイル

この項では、特性が類似している一連のクライアントとサーバーに対して生成されるサンプル sqlnet.ora 構成ファイルの例を紹介します。このファイルには、**Oracle Advanced Security** 暗号化パラメータおよびデータの整合性パラメータの例が含まれています。

トレース・ファイルの設定

```
#Trace file setup
trace_level_server=16
trace_level_client=16
trace_directory_server=/orant/network/trace
trace_directory_client=/orant/network/trace
trace_file_client=cli
trace_file_server=srv
trace_unique_client=true
```

Oracle Advanced Security 暗号化

```
#ASO Encryption
sqlnet.encryption_server=accepted
sqlnet.encryption_client=requested
sqlnet.encryption_types_server=(RC4_40)
sqlnet.encryption_types_client=(RC4_40)
sqlnet.crypto_seed = "-kdje83kkep39487dvmlqEPtBxxe70273"
```

Oracle Advanced Security 整合性

```
#ASO Checksum
sqlnet.crypto_checksum_server=requested
sqlnet.crypto_checksum_client=requested
sqlnet.crypto_checksum_types_server = (MD5)
sqlnet.crypto_checksum_types_client = (MD5)
```

SSL

```
#SSL
WALLET_LOCATION = (SOURCE=
                    (METHOD = FILE)
                    (METHOD_DATA =
                     DIRECTORY=/wallet)

SSL_CIPHER_SUITES=(SSL_DH_anon_WITH_RC4_128_MD5)
SSL_VERSION= 3
SSL_CLIENT_AUTHENTICATION=FALSE
```


共通

```
#Common
automatic_ipc = off
sqlnet.authentication_services = (beq)
names.directory_path = (TNSNAMES)
```

Kerberos

```
#Kerberos
sqlnet.authentication_services = (beq, kerberos5)
sqlnet.authentication_kerberos5_service = oracle
sqlnet.kerberos5_conf= /krb5/krb.conf
sqlnet.kerberos5_keytab= /krb5/v5srvtab
sqlnet.kerberos5_realms= /krb5/krb.realm
sqlnet.kerberos5_cc_name = /krb5/krb5.cc
sqlnet.kerberos5_clockskew=900
sqlnet.kerberos5_conf_mit=false
```

CyberSafe

```
#CyberSafe
sqlnet.authentication_services = (beq, cybersafe)
sqlnet.authentication_gssapi_service = oracle/cybersaf.us.oracle.com
sqlnet.authentication_kerberos5_service = oracle
sqlnet.kerberos5_conf= /krb5/krb.conf
sqlnet.kerberos5_keytab= /krb5/v5srvtab
sqlnet.kerberos5_realms= /krb5/krb.realm
sqlnet.kerberos5_cc_name = /krb5/krb5.cc
sqlnet.kerberos5_clockskew=900
```

RADIUS

```
#Radius
sqlnet.authentication_services = (beq, RADIUS )
sqlnet.radius_authentication_timeout = (10)
sqlnet.radius_authentication_retries = (2)
sqlnet.radius_authentication_port = (1645)
sqlnet.radius_send_accounting = OFF
sqlnet.radius_secret = /orant/network/admin/radius.key
sqlnet.radius_authentication = radius.us.oracle.com
sqlnet.radius_challenge_response = OFF
sqlnet.radius_challenge_keyword = challenge
sqlnet.radius_challenge_interface =
oracle/net/radius/DefaultRadiusInterface
sqlnet.radius_classpath = /jre1.1/
```

データの暗号化と整合性のパラメータ

サーバー暗号化、クライアント暗号化、サーバー・チェックサムまたはクライアント・チェックサムの値を指定しないと、それぞれに対応する構成パラメータが `sqlnet.ora` ファイルに設定されません。ただし、Oracle Advanced Security では、各パラメータが `ACCEPTED` にデフォルト設定されています。

データの暗号化と整合性のアルゴリズムについて、サーバーは、サーバーの `sqlnet.ora` ファイルにリストされているアルゴリズムの中で、クライアントの `sqlnet.ora` ファイルまたはクライアントのインストール済みリスト（クライアントの `sqlnet.ora` ファイルにアルゴリズムがリストされていない場合）にリストされているアルゴリズムと最初に一致したものを選択します。サーバーの `sqlnet.ora` ファイルにエントリがない場合、そのサーバーは、インストール済みリストを順に検索し、クライアントの `sqlnet.ora` ファイルまたはクライアントのインストール済みリストに含まれているクライアント側の項目と照合します。一致する項目がなく、接続の一方でアルゴリズムのタイプが必須の場合（データの暗号化または整合性パラメータが `REQUIRED` に設定されている場合）、その接続は失敗します。それ以外の場合、接続は成功しますが、そのアルゴリズムのタイプはアクティブになりません。

データの暗号化と整合性のアルゴリズムは、互いに独立して選択されます。表 A-1 で示すように、暗号化は整合性なしで、整合性は暗号化なしでアクティブにすることができます。

表 A-1 アルゴリズムのタイプの選択

暗号化での選択	整合性での選択
選択する	選択しない
選択する	選択する
選択しない	選択する
選択しない	選択しない

データの暗号化と整合性を使用可能にするには、次の 3 つのパラメータ・クラスが必要です。

- 暗号化と整合性のレベル設定
- 暗号化と整合性の選択リスト
- ランダム鍵ジェネレータのシード

関連項目：

- 第 2 章「データの暗号化および整合性の構成」
- 2-6 ページ「暗号化および整合性をアクティブにする」

暗号化と整合性のレベル設定

表 A-2 には、データの暗号化と整合性のレベル設定がまとめてあります。

表 A-2 暗号化と整合性のレベル設定

アルゴリズム のタイプ	プラット フォーム	項目	説明
暗号化	サーバー	用途	クライアントまたはクライアントとして動作するサーバーがこのサーバーに接続するときに必要な暗号化の動作を指定します。サーバーの動作は、接続先で設定されている <code>SQLNET.ENCRYPTION_CLIENT</code> によって多少変化します。
		構文	<code>SQLNET.ENCRYPTION_SERVER = valid_value</code>
		指定できる値	ACCEPTED、REJECTED、REQUESTED、REQUIRED
		デフォルト	ACCEPTED
	クライアント	用途	このクライアントまたはクライアントとして動作するサーバーがサーバーに接続するときに必要な暗号化の動作を指定します。クライアントの動作は、接続先で設定されている <code>SQLNET.ENCRYPTION_SERVER</code> によって多少変化します。
		構文	<code>SQLNET.ENCRYPTION_CLIENT = valid_value</code>
		指定できる値	ACCEPTED、REJECTED、REQUESTED、REQUIRED
		デフォルト	ACCEPTED

表 A-2 暗号化と整合性のレベル設定（続き）

アルゴリズム のタイプ	プラット フォーム	項目	説明
整合性	サーバー	用途	クライアントまたはクライアントとして動作する別のサーバーがこのサーバーと接続するときに必要なデータの整合性の動作を指定します。サーバーの動作は、接続先で設定されている SQLNET.CRYPTO_CHECKSUM_CLIENT によって多少変化します。
		構文	SQLNET.CRYPTO_CHECKSUM_SERVER = valid_value
		指定できる値	ACCEPTED、REJECTED、REQUESTED、REQUIRED
		デフォルト	ACCEPTED
	クライアント	用途	このクライアントまたはクライアントとして動作するサーバーがサーバーと接続するときに必要なデータの整合性の動作を指定します。クライアントの動作は、接続先で設定されている SQLNET.CRYPTO_CHECKSUM_SERVER によって多少変化します。
		構文	SQLNET.CRYPTO_CHECKSUM_CLIENT = valid_value
		指定できる値	ACCEPTED、REJECTED、REQUESTED、REQUIRED
		デフォルト	ACCEPTED

暗号化と整合性の選択リスト

表 A-3 暗号化と整合性の選択リスト

アルゴリズム のタイプ	プラットフォーム	項目	説明
暗号化	サーバー	用途	このサーバーで使用する暗号化アルゴリズムのリストを優先的に使用する順序で指定します。このリストを使用して、接続先のクライアント側と相互に受入れ可能なアルゴリズムを折衝します。一致するアルゴリズムが見つかるまで、サーバー側の各アルゴリズムがクライアント側のアルゴリズム・リストと照合されます。インストールされていないアルゴリズムをこのサーバー側で指定すると、エラー・メッセージ ORA-12650 で接続が終了します。
		構文	<code>SQLNET.ENCRYPTION_TYPES_SERVER = (valid_encryption_algorithm [,valid_encryption_algorithm])</code>
		指定できる値	<ul style="list-style-type: none">■ RC4_256: RSA RC4 (256 ビット鍵サイズ)■ AES256: AES (256 ビット鍵サイズ)■ AES192: AES (192 ビット鍵サイズ)■ 3DES168: 3 つの鍵を使用するトリプル DES (有効な鍵サイズ: 168 ビット)■ RC4_128: RSA RC4 (128 ビット鍵サイズ)■ AES128: AES (128 ビット鍵サイズ)■ 3DES112: 2 つの鍵を使用するトリプル DES (有効な鍵サイズ: 112 ビット)■ RC4_56: RSA RC4 (56 ビット鍵サイズ)■ DES: 標準 DES (56 ビット鍵サイズ)■ RC4_40: RSA RC4 (40 ビット鍵サイズ)■ DES40: DES40 (40 ビット鍵サイズ)
		デフォルト	ローカルの <code>sqlnet.ora</code> ファイルでアルゴリズムが定義されていない場合は、すべてのインストール済みアルゴリズムが前述の順で折衝に使用されます。

表 A-3 暗号化と整合性の選択リスト（続き）

アルゴリズム のタイプ	プラット フォーム	項目	説明
暗号化	サーバー	使用上の注意	<p>複数の暗号化アルゴリズム、つまり、アルゴリズム名の1つの値またはリストを指定できます。たとえば、次に示す暗号化パラメータはどちらも有効です。</p> <p>SQLNET.ENCRYPTION_TYPES_SERVER=(RC4_40)</p> <p>SQLNET.ENCRYPTION_TYPES_SERVER=(DES,RC4_56,RC4_128,DES40)</p>
	クライアント	用途	このクライアントまたはクライアントとして動作するサーバーで使用する暗号化アルゴリズムのリストを指定します。このリストを使用して、接続先と相互に受入れ可能なアルゴリズムを折衝します。インストールされていないアルゴリズムをこのサーバー側で指定すると、エラー・メッセージ ORA-12650 で接続が終了します。
		構文	SQLNET.ENCRYPTION_TYPES_CLIENT = (valid_encryption_algorithm [,valid_encryption_algorithm])
		指定できる値	<ul style="list-style-type: none"> ■ RC4_256: RSA RC4 (256 ビット鍵サイズ) ■ AES256: AES (256 ビット鍵サイズ) ■ AES192: AES (192 ビット鍵サイズ) ■ 3DES168: 3つの鍵を使用するトリプル DES (有効な鍵サイズ: 168 ビット) ■ RC4_128: RSA RC4 (128 ビット鍵サイズ) ■ AES128: AES (128 ビット鍵サイズ) ■ 3DES112: 2つの鍵を使用するトリプル DES (有効な鍵サイズ: 112 ビット) ■ RC4_56: RSA RC4 (56 ビット鍵サイズ) ■ DES: 標準 DES (56 ビット鍵サイズ) ■ RC4_40: RSA RC4 (40 ビット鍵サイズ) ■ DES40: DES40 (40 ビット鍵サイズ)
		デフォルト	ローカルの sqlnet.ora ファイルでアルゴリズムが定義されていない場合は、すべてのインストール済みアルゴリズムが折衝に使用されます。

表 A-3 暗号化と整合性の選択リスト（続き）

アルゴリズム のタイプ	プラット フォーム	項目	説明
暗号化	クライア ント	使用上 の注意	<p>複数の暗号化アルゴリズム、つまり、アルゴリズム名の1つの値またはリストを指定できます。たとえば、次に示す暗号化パラメータはどちらも有効です。</p> <pre>SQLNET.ENCRYPTION_TYPES_CLIENT=(DES,DES40,RC4_56,RC4_40) SQLNET.ENCRYPTION_TYPES_CLIENT=(RC4_40)</pre>
整合性	サーバー	用途	このサーバーまたは別のサーバーに対するクライアントで使用するデータの整合性アルゴリズムのリストを優先的に使用する順序で指定します。このリストを使用して、接続先と相互に受入れ可能なアルゴリズムを折衝します。一致するアルゴリズムが見つかるまで、サーバー側の各アルゴリズムがクライアント側のアルゴリズム・リストと照合されます。インストールされていないアルゴリズムをこのサーバー側で指定すると、エラー・メッセージ ORA-12650 で接続が終了します。
		構文	SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER = (valid_crypto_checksum_algorithm [,valid_crypto_checksum_algorithm])
		指定できる値	<ul style="list-style-type: none"> ■ SHA-1: Secure Hash Algorithm ■ MD5: Message Digest 5
		デフォルト	ローカルの sqlnet.ora ファイルでアルゴリズムが定義されていない場合は、すべてのインストール済みアルゴリズムが前述の順で折衝に使用されます。
	クライア ント	用途	このクライアントまたはクライアントとして動作するサーバーで使用するデータの整合性アルゴリズムのリストを指定します。このリストを使用して、接続先と相互に受入れ可能なアルゴリズムを折衝します。インストールされていないアルゴリズムをこのサーバー側で指定すると、エラー・メッセージ ORA-12650 で接続が終了します。
		構文	SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT = (valid_crypto_checksum_algorithm [,valid_crypto_checksum_algorithm])
		指定できる値	<ul style="list-style-type: none"> ■ SHA-1: Secure Hash Algorithm ■ MD5: Message Digest 5

表 A-3 暗号化と整合性の選択リスト（続き）

アルゴリズム のタイプ	プラット フォーム	項目	説明
整合性	クライア ント	デフォ ルト	ローカルの sqlnet.ora ファイルでアルゴリズムが定義 されていない場合は、すべてのインストール済みアルゴ リズムが折衝に使用されます。

ランダム鍵ジェネレータのシード

SQLNET.CRYPTO_SEED = "10-70 random characters"

このパラメータの値として入力する文字を使用して、暗号鍵が生成されます。このフィールドにランダムな文字を入力するほど、強力な鍵が生成されます。このパラメータを設定するには、前述の文にランダムな文字を 10 ～ 70 文字入力します。

注意： 生成される鍵がランダムで強力になるように、できるだけ多くの文字（最大 70 文字まで）を入力することをお勧めします。

データの暗号化または整合性を使用可能にするときは、このパラメータが必ずその sqlnet.ora ファイルに存在している必要があります。

認証パラメータ

この付録では、プロファイル・ファイル (sqlnet.ora) とデータベース初期化ファイル (init.ora) に必要な認証パラメータを含む構成ファイルのサンプルをいくつか紹介します。これらのパラメータは、CyberSafe、Kerberos、RADIUS または SSL の各認証を使用するときに必要です。

項目は、次のとおりです。

- [CyberSafe 認証を使用したクライアントとサーバーのパラメータ](#)
- [Kerberos 認証を使用するクライアントとサーバーのパラメータ](#)
- [RADIUS 認証を使用するクライアントとサーバーのパラメータ](#)
- [SSL を使用するクライアントとサーバーのパラメータ](#)

CyberSafe 認証を使用したクライアントとサーバーのパラメータ

CyberSafe を使用するクライアントとサーバーの構成ファイルには、次のパラメータを挿入します。

表 B-1 CyberSafe 構成パラメータ

ファイル名	構成パラメータ
sqlnet.ora	SQLNET.AUTHENTICATION_SERVICES=(cybersafe) SQLNET.AUTHENTICATION_GSSAPI_SERVICE= oracle/dbserver.someco.com@SOME.CO.COM SQLNET.AUTHENTICATION_KERBEROS5_SERVICES=oracle SQLNET.KERBEROS5_CONF=/krb5/krb.conf SQLNET.KERBEROS5_REALMS=/krb5/krb.realms SQLNET.KERBEROS5_KEYTAB=/krb5/v5srvtab
初期化パラメータ・ ファイル (init.ora)	REMOTE_OS_AUTHENT=FALSE OS_AUTHENT_PREFIX=""

Kerberos 認証を使用するクライアントとサーバーのパラメータ

Kerberos を使用するクライアントとサーバーの構成ファイルには、次のパラメータを挿入します。

表 B-2 Kerberos 認証パラメータ

ファイル名	構成パラメータ
sqlnet.ora	SQLNET.AUTHENTICATION_SERVICES=(KERBEROS5) SQLNET.AUTHENTICATION_KERBEROS5_SERVICE=oracle SQLNET.KERBEROS5_CC_NAME=/usr/tmp/DCE-CC SQLNET.KERBEROS5_CLOCKSKEW=1200 SQLNET.KERBEROS5_CONF=/krb5/krb.conf SQLNET.KERBEROS5_CONF_MIT=(FALSE) SQLNET.KERBEROS5_REALMS=/krb5/krb.realms SQLNET.KERBEROS5_KEYTAB=/krb5/v5srvtab
初期化パラメータ・ ファイル (init.ora)	REMOTE_OS_AUTHENT=FALSE OS_AUTHENT_PREFIX=""

RADIUS 認証を使用するクライアントとサーバーのパラメータ

次の項では、RADIUS 認証用のパラメータについて説明しています。

- [sqlnet.ora ファイル・パラメータ](#)
- 最小限の RADIUS パラメータ
- 初期化ファイル (init.ora) パラメータ

sqlnet.ora ファイル・パラメータ

SQLNET.AUTHENTICATION_SERVICES

このパラメータは、RADIUS アダプタを使用するためにクライアントまたはサーバーを構成します。表 B-3 に、このパラメータの属性を示します。

表 B-3 SQLNET.AUTHENTICATION_SERVICES パラメータの属性

属性	説明
構文	SQLNET.AUTHENTICATION_SERVICES=radius
デフォルト設定	なし

SQLNET.RADIUS_AUTHENTICATION

このパラメータは、プライマリ RADIUS サーバーの場所を、ホスト名またはドット付き 10 進数の書式で設定します。RADIUS サーバーが Oracle サーバー以外のマシンにあるときは、そのマシンのホスト名または IP アドレスを指定する必要があります。表 B-4 に、このパラメータの属性を示します。

表 B-4 SQLNET.RADIUS_AUTHENTICATION パラメータの属性

属性	説明
構文	SQLNET.RADIUS_AUTHENTICATION=RADIUS_server_IP_address
デフォルト設定	localhost

SQLNET.RADIUS_AUTHENTICATION_PORT

このパラメータは、プライマリ RADIUS サーバーのリスニング・ポートを設定します。表 B-5 に、このパラメータの属性を示します。

表 B-5 SQLNET.RADIUS_AUTHENTICATION_PORT パラメータの属性

属性	説明
構文	SQLNET.RADIUS_AUTHENTICATION_PORT= <i>port_number</i>
デフォルト設定	1645

SQLNET.RADIUS_AUTHENTICATION_TIMEOUT

このパラメータは、応答待ち時間を設定します。表 B-6 に、このパラメータの属性を示します。

表 B-6 SQLNET.RADIUS_AUTHENTICATION_TIMEOUT パラメータの属性

属性	説明
構文	SQLNET.RADIUS_AUTHENTICATION_TIMEOUT= <i>time_in_seconds</i>
デフォルト設定	5

SQLNET.RADIUS_AUTHENTICATION_RETRIES

このパラメータは、再送回数を設定します。表 B-7 に、このパラメータの属性を示します。

表 B-7 SQLNET.RADIUS_AUTHENTICATION_RETRIES パラメータの属性

属性	説明
構文	SQLNET.RADIUS_AUTHENTICATION_RETRIES= <i>n_times_to_resend</i>
デフォルト設定	3

SQLNET.RADIUS_SEND_ACCOUNTING

このパラメータは、アカウントをオンまたはオフに設定します。アカウントを使用可能にすると、パケットはリスニング・ポート +1 のアクティブ RADIUS サーバーに送られます。デフォルトの場合、パケットはポート 1646 に送信されます。この機能をオンにする必要があるのは、RADIUS サーバーでアカウントをサポートしている場合で、システムにログインしたユーザーのログイン回数を記録する場合のみです。表 B-8 に、このパラメータの属性を示します。

表 B-8 SQLNET.RADIUS_SEND_ACCOUNTING パラメータの属性

属性	説明
構文	SQLNET.RADIUS_SEND_ACCOUNTING= <i>on</i>
デフォルト設定	OFF

SQLNET.RADIUS_SECRET

このパラメータは、ファイル名と RADIUS 秘密鍵の場所を指定します。表 B-9 に、このパラメータの属性を示します。

表 B-9 SQLNET.RADIUS_SECRET パラメータの属性

属性	説明
構文	SQLNET.RADIUS_SECRET= <i>path_to_RADIUS_secret_key</i>
デフォルト設定	<i>\$ORACLE_HOME/network/security/radius.key</i>

SQLNET.RADIUS_ALTERNATE

このパラメータは、プライマリ・サーバーがフォルト・トレランスに使用できない場合、かわりに使用する RADIUS サーバーの場所を設定します。表 B-10 に、このパラメータの属性を示します。

表 B-10 SQLNET.RADIUS_ALTERNATE パラメータの属性

属性	説明
構文	SQLNET.RADIUS_ALTERNATE= <i>alternate_RADIUS_server_hostname_or_IP_address</i>
デフォルト設定	OFF

SQLNET.RADIUS_ALTERNATE_PORT

このパラメータは、代替 RADIUS サーバーのリスニング・ポートを設定します。表 B-11 に、このパラメータの属性を示します。

表 B-11 SQLNET.RADIUS_ALTERNATE_PORT パラメータの属性

属性	説明
構文	SQLNET.RADIUS_ALTERNATE_PORT= <i>alternate_RADIUS_server_listening_port_number</i>
デフォルト設定	1645

SQLNET.RADIUS_ALTERNATE_TIMEOUT

このパラメータは、代替 RADIUS サーバーの応答待ち時間を設定します。表 B-12 に、このパラメータの属性を示します。

表 B-12 SQLNET.RADIUS_ALTERNATE_TIMEOUT パラメータの属性

属性	説明
構文	SQLNET.RADIUS_ALTERNATE_TIMEOUT= <i>time_in_seconds</i>
デフォルト設定	5

SQLNET.RADIUS_ALTERNATE_RETRIES

このパラメータは、代替 RADIUS サーバーによるメッセージの再送回数を設定します。表 B-13 に、このパラメータの属性を示します。

表 B-13 SQLNET.RADIUS_ALTERNATE_RETRIES パラメータの属性

属性	説明
構文	SQLNET.RADIUS_ALTERNATE_RETRIES= <i>n_times_to_resend</i>
デフォルト設定	3

SQLNET.RADIUS_CHALLENGE_RESPONSE

このパラメータは、要求 / 応答（つまり、非同期）モードのサポートをオンまたはオフに設定します。表 B-14 に、このパラメータの属性を示します。

表 B-14 SQLNET.RADIUS_CHALLENGE_RESPONSE パラメータの属性

属性	説明
構文	SQLNET.RADIUS_CHALLENGE_RESPONSE= <i>on</i>
デフォルト設定	OFF

SQLNET.RADIUS_CHALLENGE_KEYWORD

このパラメータは、RADIUS サーバーからの要求を求めるキーワードを設定します。クライアント側のユーザーはパスワードを入力しません。表 B-15 に、このパラメータの属性を示します。

表 B-15 SQLNET.RADIUS_CHALLENGE_KEYWORD パラメータの属性

属性	説明
構文	SQLNET.RADIUS_CHALLENGE_KEYWORD= <i>keyword</i>
デフォルト設定	challenge

SQLNET.RADIUS_AUTHENTICATION_INTERFACE

このパラメータは、RADIUS が要求 / 応答（非同期）モードのときに、グラフィカル・ユーザー・インタフェースを持つ Java クラスの名前を設定します。表 B-16 に、このパラメータの属性を示します。

表 B-16 SQLNET.RADIUS_AUTHENTICATION_INTERFACE パラメータの属性

属性	説明
構文	SQLNET.RADIUS_AUTHENTICATION_INTERFACE= <i>Java_class_name</i>
デフォルト設定	DefaultRadiusInterface (oracle/net/radius/DefaultRadiusInterface)

SQLNET.RADIUS_CLASSPATH

要求 / 応答認証モードを使用する場合、RADIUS は、Java ベースのグラフィカル・インタフェースをユーザーに表示して、最初にパスワードを、次にユーザーがトークン・カードから取得する動的パスワードなどの他の追加情報を要求します。SQLNET.RADIUS_CLASSPATH パラメータを sqlnet.ora ファイルに追加して、そのグラフィカル・インタフェースの Java クラスのパスを設定し、JDK Java ライブラリへのパスを設定します。表 B-17 に、このパラメータの属性を示します。

表 B-17 SQLNET.RADIUS_CLASSPATH パラメータの属性

属性	説明
構文	SQLNET.RADIUS_CLASSPATH= <i>path_to_GUI_Java_classes</i>
デフォルト設定	<i>\$ORACLE_HOME/jlib/netradius.jar:</i> <i>\$ORACLE_HOME/JRE/lib/sparc/native_threads</i>

最小限の RADIUS パラメータ

```
sqlnet.authentication_services = (radius)
sqlnet.authentication = IP-address-of-RADIUS-server
sqlnet.radius_challenge_response = ON
```

初期化ファイル (init.ora) パラメータ

```
REMOTE_OS_AUTHENT=FALSE
OS_AUTHENT_PREFIX=""
```


SSL を使用するクライアントとサーバーのパラメータ

パラメータを構成するには、次の 2 通りあります。

- 静的 : `sqlnet.ora` ファイル内にあるパラメータの名前
- 動的 : Oracle Net アドレスのセキュリティ・サブセクションで使用するパラメータの名前

SSL 認証パラメータ

この項では、サーバー上に SSL を構成するための静的パラメータと動的パラメータについて説明します。

パラメータ名 (静的) : `SQLNET.AUTHENTICATION_SERVICES`

パラメータ名 (動的) : `AUTHENTICATION`

パラメータ・タイプ : 文字列 LIST

パラメータ・クラス : 静的

指定できる値 : 使用可能な認証サービスのリストに TCPS を追加します。

デフォルト値 : デフォルト値はありません。

説明 : ユーザーが使用する認証サービスを制御します。

注意 : 動的バージョンでは 1 種類の設定のみサポートされます。

既存 / 新規パラメータ

既存

構文 (静的) : `SQLNET.AUTHENTICATION_SERVICES = (TCPS, selected_method_1, selected_method_2)`

例 (静的) : `SQLNET.AUTHENTICATION_SERVICES = (TCPS, cybersafe)`

構文 (動的) : `AUTHENTICATION = string`

例 (動的) :

`AUTHENTICATION = (TCPS)`

Cipher Suite パラメータ

この項では、Cipher Suite を構成するための静的パラメータと動的パラメータについて説明します。

パラメータ名（静的）： SSL_CIPHER_SUITES

パラメータ名（動的）： SSL_CIPHER_SUITES

パラメータ・タイプ： 文字列 LIST

パラメータ・クラス： 静的

指定できる値： 既知の SSL Cipher Suite

デフォルト値： デフォルトはありません。

説明： SSL で使用する暗号化とデータ整合性の組合せを制御します。

既存 / 新規パラメータ 既存

構文（静的）： SSL_CIPHER_SUITES=(SSL_cipher_suite1[, SSL_cipher_suite2, ...
SSL_cipher_suiteN])

例（静的）： SSL_CIPHER_SUITES=(SSL_DH_DSS_WITH_DES_CBC_SHA)

構文（動的）： SSL_CIPHER_SUITES=(SSL_cipher_suite1
[, SSL_cipher_suite2, ...SSL_cipher_suiteN])

例（動的）： SSL_CIPHER_SUITES=(SSL_DH_DSS_WITH_DES_CBC_SHA)

サポートされている SSL Cipher Suite

Oracle Advanced Security では次の Cipher Suite をサポートしています。

- SSL_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_WITH_RC4_128_SHA
- SSL_RSA_WITH_RC4_128_MD5
- SSL_RSA_WITH_DES_CBC_SHA
- SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
- SSL_DH_anon_WITH_RC4_128_MD5
- SSL_DH_anon_WITH_DES_CBC_SHA
- SSL_RSA_EXPORT_WITH_RC4_40_MD5
- SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
- SSL_DH_anon_EXPORT_WITH_RC4_40_MD5
- SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA

SSL バージョン・パラメータ

この項では、使用する SSL のバージョンを構成するための静的パラメータと動的パラメータについて説明します。

パラメータ名（静的）： SSL_VERSION

パラメータ名（動的）： SSL_VERSION

パラメータ・タイプ： 文字列

パラメータ・クラス： 静的

指定できる値： SSL で有効な任意のバージョン（0、3.0）

デフォルト値： "0"

説明： SSL 接続のバージョンを強制実行します。

既存 / 新規パラメータ 新規

構文（静的）： SSL_VERSION=*version*

例（静的）： SSL_VERSION=3.0

構文（動的）： SSL_VERSION=*version*

例（動的）： SSL_VERSION=3.0

SSL クライアント認証パラメータ

この項では、クライアント上に SSL を構成するための静的パラメータと動的パラメータについて説明します。

パラメータ名 (静的) : SSL_CLIENT_AUTHENTICATION

パラメータ名 (動的) : SSL_CLIENT_AUTHENTICATION

パラメータ・タイプ : ブール値

パラメータ・クラス : 静的

指定できる値 : TRUE/FALSE

デフォルト値 : TRUE

説明 : サーバーに加えて、クライアントを SSL で認証するかどうかを制御します。

既存 / 新規パラメータ : 新規

構文 (静的) : SSL_CLIENT_AUTHENTICATION={TRUE | FALSE}

例 (静的) : SSL_CLIENT_AUTHENTICATION=FALSE

構文 (動的) : SSL_CLIENT_AUTHENTICATION={TRUE | FALSE}

例 (動的) : SSL_CLIENT_AUTHENTICATION=FALSE

SSL X.509 サーバー照合パラメータ

この項では、クライアントの接続先サーバーを識別するためのパラメータについて説明します。

SSL_SERVER_DN_MATCH

パラメータ名	SSL_SERVER_DN_MATCH
格納場所	sqlnet.ora
用途	サーバーの 識別名 (DN) とそのサービス名を照合するために使用します。照合検証を行う場合は、SSL によって、証明書がサーバーのものであることが保証されます。照合検証を行わない場合、SSL は DN とサービス名が一致しているかチェックしますが、一致していない場合でも接続は成功します。照合を行わないと、サーバーが自身の識別情報を偽る可能性があります。
指定できる値	YES ON TRUE – 照合することを指定します。DN とサービス名が一致した場合は接続が成功し、一致しない場合は失敗します。 NO OFF FALSE – 照合しないことを指定します。DN とサービス名が一致しない場合、接続は成功しますが、sqlnet.log ファイルにエラーが記録されます。
デフォルト	Oracle8i および Oracle9i: FALSE。SSL クライアントは常にサーバーの DN をチェックします。DN とサービス名が一致しない場合、接続は成功しますが、sqlnet.log ファイルにエラーが記録されます。
使用上の注意	サーバー DN の照合を有効にするには、tnsnames.ora パラメータの SSL_SERVER_CERT_DN もあわせて構成する必要があります。

SSL_SERVER_CERT_DN

パラメータ名	SSL_SERVER_CERT_DN
格納場所	tnsnames.ora。クライアント上に格納し、接続するサーバーごとにエントリを定義できます。また、LDAP ディレクトリに格納し、接続するサーバーごとにエントリを定義して、集中的に更新することもできます。
用途	サーバーの 識別名 を指定します。クライアントは、この情報を使用して、各サーバーに対して想定している DN のリストを取得し、サーバーの DN とそのサービス名との照合を行います。
指定できる値	サーバーの 識別名 と等しい値を設定します。
デフォルト	N/A

使用上の注意

サーバー DN の照合を有効にするには、sqlnet.ora パラメータの SSL_SERVER_DN_MATCH もあわせて構成する必要があります。

例

dbalias=(description=address_
list=(address=(protocol=tcps) (host=hostname) (port=p
ortnum))) (connect_
data=(sid=Finance)) (security=(SSL_SERVER_
DN="CN=Finance,CN=OracleContext,C=US,O=Acme"))

Wallet の場所

セキュリティ資格証明をプロセス空間にロードするために Wallet にアクセスする必要があるアプリケーションについては、表 B-18 で定義されている Wallet の場所パラメータを次の各構成ファイルに指定する必要があります。

- sqlnet.ora
- listener.ora

表 B-18 Wallet の場所パラメータ

静的構成	動的構成
WALLET_LOCATION = (SOURCE= (METHOD=File) (METHOD_DATA= (DIRECTORY=your wallet location)))	MY_WALLET_DIRECTORY = your_wallet_dir

デフォルトの Wallet の場所は、\$ORACLE_HOME ディレクトリです。

RADIUS による認証デバイスの統合

この付録では、認証デバイスのサード・パーティ・ベンダーが、その認証デバイスに合わせて RADIUS 要求 / 応答ユーザー・インタフェースをカスタマイズする方法について説明します。

項目は、次のとおりです。

- [RADIUS 要求 / 応答ユーザー・インタフェース](#)
- [RADIUS 要求 / 応答ユーザー・インタフェースのカスタマイズ](#)

関連項目： [第 4 章「RADIUS 認証の構成」](#)

RADIUS 要求 / 応答ユーザー・インタフェース

RADIUS 標準をサポートする認証デバイスをセットアップして、Oracle ユーザーを認証できます。認証デバイスで要求 / 応答認証モードを使用する場合、グラフィカル・インタフェースを表示してパスワードを最初に要求し、ユーザーがトークン・カードから取得する動的パスワードなど、他の追加情報を要求します。このインタフェースは、プラットフォームからの最適な独立性を得るために Java ベースになっています。

認証デバイスのサード・パーティ・ベンダーは、そのデバイスに適したグラフィカル・ユーザー・インタフェースにカスタマイズする必要があります。たとえば、スマートカードのベンダーは、スマートカード・リーダーに要求を発行するように、Oracle クライアントをカスタマイズします。次に、スマートカードが要求を受け取ると、PIN などの追加情報をユーザーに入力させて応答します。

RADIUS 要求 / 応答ユーザー・インタフェースのカスタマイズ

このインタフェースをカスタマイズするには、表 C-1 の機能をサポートする独自のクラスを作成します。次に、sqlnet.ora ファイルをオープンして SQLNET.RADIUS_AUTHENTICATION_INTERFACE パラメータを検索し、値としてリストされているクラス名 DefaultRadiusInterface を、作成したクラス名に置き換えます。この変更を sqlnet.ora ファイルで行うと、認証手続きを処理するために、このクラスが Oracle クライアントにロードされます。

サード・パーティは、ORACLE.NET.RADIUS パッケージにある Oracle RADIUS インタフェースを実装する必要があります。

```
public interface OracleRadiusInterface {
    public void radiusRequest();
    public void radiusChallenge(String challenge);
    public String getUsername();
    public String getPassword();
}
```

表 C-1 サーバーの暗号化レベルの設定

パラメータ	説明
radiusRequest	通常、ユーザーに対してユーザー名とパスワードを入力するようプロンプトを表示し、これらの値を getUsername と getPassword で取得します。
getUsername	ユーザーが入力したユーザー名を取得します。このメソッドが空の文字列を戻したときは、ユーザーが操作をキャンセルしていることを意味します。ユーザーは、認証に失敗したというメッセージを受け取ります。

表 C-1 サーバーの暗号化レベルの設定（続き）

パラメータ	説明
getPassword	ユーザーが入力したパスワードを取得します。getUserName が有効な文字列を返し、getPassword が空の文字列を返した場合は、データベースでパスワードとして要求キーワードが設定されます。ユーザーが有効なパスワードを入力した場合は、RADIUS サーバーから要求が返されるときと返されないときがあります。
radiusChallenge	ユーザーがサーバーの要求に応答できるように、RADIUS サーバーから送られてきた要求を表示します。
getResponse	ユーザーが入力した応答を取得します。このメソッドが有効な応答を戻した場合は、新規の Access-Request パケットの User-Password 属性にその情報が入力されます。空の文字列が戻された場合は、対応する値を戻して、両サイドでの処理が停止されます。

Oracle Advanced Security FIPS 140-1 の設定

Oracle Advanced Security リリース 8.1.6 は、[米国連邦情報処理標準 140-1](#) の Level2 セキュリティ・レベルの検証を受けています。この付録では、Oracle Advanced Security が FIPS 140-1 標準に準拠するうえで必要となる正式な構成について説明しています。次の Web サイトで、NIST Cryptographic Modules Validation のリストを参照してください。

<http://csrc.nist.gov/cryptval/140-1/1401val.htm>

項目は、次のとおりです。

- [構成パラメータ](#)
- [インストール後のチェック](#)
- [ステータス情報](#)
- [物理的なセキュリティ](#)

注意： この付録に含まれている情報は、[付録 A「データの暗号化と整合性のパラメータ」](#) で提供している情報とともに使用してください。

構成パラメータ

この付録には、クライアントとサーバーの間で行われたすべての接続がサーバーの制御の下で暗号化されるようにする `sqlnet.ora` ファイルに必要な Oracle Advanced Security パラメータに関する情報が含まれています。

構成パラメータは、各クライアントおよびサーバー・プロセスごとにローカルで保持される `sqlnet.ora` ファイルに格納されています。これらのファイルに対して設定される保護機能は、DBA のレベルと同等である必要があります。

この付録では、次の構成パラメータについて説明しています。

- `ENCRYPTION_SERVER`
- `ENCRYPTION_CLIENT`
- `ENCRYPTION_TYPES_SERVER`
- `CRYPTO_SEED`
- `CRYPTO_SEED_CLIENT`
- `FIPS_140`

サーバーの暗号化レベルの設定

サーバー側の折衝により、概念上の接続の設定が制御されます。サーバー・ファイル内の次のパラメータは必須です。

```
SQLNET.ENCRYPTION_SERVER=REQUIRED
```

サーバー側の接続で暗号化を `REQUIRED` に設定すると、クライアント側のパラメータ値に関係なく、暗号化が使用されている場合にのみ接続が許可されます。

クライアントの暗号化レベルの設定

`ENCRYPTION_CLIENT` パラメータは、クライアントの接続動作を指定します。クライアント・ファイル内の次のパラメータ設定のうちのいずれかが必須です。

```
SQLNET.ENCRYPTION_CLIENT=(ACCEPTED|REQUESTED|REQUIRED)
```

サーバーへの接続は、接続の暗号化についてクライアントとサーバーとの間で合意がある場合にのみ可能です。サーバー側で値を `REQUIRED` に設定すると、接続を有効とするにはクライアント側は暗号化を拒否できません。これらの値のいずれかを指定しないと、FIPS 140-1 準拠のサーバーに接続するときにエラーが発生します。

サーバーの暗号化選択リスト

ENCRYPTION_TYPES_SERVER パラメータには、サーバーが、サーバーとして動作するときに見える暗号化アルゴリズムのリストを、優先的に使用する順序で指定します。指定したアルゴリズムがインストールされているか、あるいは接続が終了している必要があります。FIPS 140-1 準拠の場合、DES 暗号化のみが許可されているため、次のパラメータ設定が必須となります。

```
SQLNET.ENCRYPTION_TYPES_SERVER=(DES|DES40)
```

クライアントの暗号化選択リスト

ENCRYPTION_TYPES_CLIENT パラメータには、サーバーとの接続のためにクライアント側で使用する準備ができていない暗号化アルゴリズムのリストを指定します。接続を成功させるには、まず、アルゴリズムがインストールされる必要があります、暗号化タイプがサーバーに対して相互に受入可能である必要があります。

FIPS 140-1 に構成されるサーバーとの接続を作成するには、次のパラメータ設定が必須となります。

```
SQLNET.ENCRYPTION_TYPES_CLIENT=(DES|DES40)
```

暗号シード値

CRYPTO_SEED パラメータには、ランダム番号ジェネレータのシードを構成する文字が含まれています。FIPS 140-1 標準にはこのパラメータの値に関する明示的な要件はありませんが、次に示すような最大 70 までの大きなランダム文字セットを選択することをお勧めします。

```
SQLNET.CRYPTO_SEED=10_to_70_random_characters
```

FIPS パラメータ

FIPS_140 パラメータのデフォルト設定は FALSE です。Oracle Advanced Security を FIPS 140-1 で定義された標準に準拠させるには、次のようにクライアントとサーバーの両方でこのパラメータを TRUE に設定することが必須となります。

```
SQLNET.FIPS_140=TRUE
```

注意： sqlnet.ora ファイルに FIPS_140 パラメータを設定するには、テキスト・エディタを使用します。Oracle Net Manager を使用してこのパラメータを設定することはできません。

インストール後のチェック

インストール後に、オペレーティング・システムで次の許可を確認する必要があります。

- システムのセキュリティ・ポリシーに従って権限のないユーザーが Oracle Advanced Security を実行しないようにするには、すべての Oracle Advanced Security 実行可能ファイルに対して実行許可を設定する必要があります。
- ユーザーが誤ってまたは故意に Oracle Advanced Security を読み取ったり変更したりしないようにするには、すべての実行可能ファイルに対して読取りおよび書込み権限を設定する必要があります。

FIPS 140-1 Level 2 要件に準拠するには、セキュリティ・ポリシーの中に、権限のないユーザーが Oracle Advanced Security プロセスを読み取ったり、変更したり、実行したりしないようにする手順を組み込む必要があります。

ステータス情報

Oracle Advanced Security のステータス情報は、接続が確立すると使用できます。この情報は、RDBMS 仮想表 v\$session_connect_info に格納されています。

問合せ SELECT *from v\$session_connect_info を実行すると、アクティブな接続のすべての製品バナー情報が表示されます。表 D-1 では、DES 暗号化と MD5 データ整合性の両方が定義されている接続構成の例を示しています。

表 D-1 v\$session_connect_info からのサンプル出力

SID	AUTHENTICATION	OSUSER	NETWORK_SERVICE_BANNER
7	DATABASE	oracle	Oracle Bequeath operating system adapter for Solaris, v8.1.6.0.0
7	DATABASE	oracle	Oracle Advanced Security: encryption service for Solaris
7	DATABASE	oracle	Oracle Advanced Security: DES encryption service adapter
7	DATABASE	oracle	Oracle Advanced Security: crypto-checksumming service
7	DATABASE	oracle	Oracle Advanced Security: MD5 crypto-checksumming service adapter

物理的なセキュリティ

FIPS 140-1 Level2 要件に準拠するには、カバーが外されたことを発見できるように、各マシンのカバーに不正開封防止シールを貼付する必要があります。

Microsoft Active Directory でのエンタープライズ・ユーザー・セキュリティの使用

この付録では、Microsoft Active Directory を Oracle Advanced Security エンタープライズ・ユーザー・セキュリティの LDAP ディレクトリとして構成および使用する方法を説明します。次の項目について説明します。

- [Active Directory をサポートする Oracle9i ディレクトリ・サーバーの機能](#)
- [Active Directory との統合](#)
- [Active Directory で Oracle9i を使用するための要件](#)
- [Active Directory を使用するための Oracle9i の構成](#)
- [接続性のテスト](#)
- [Oracle ディレクトリ・オブジェクトのアクセス制御リストの管理](#)
- [エンタープライズ・ドメインの作成](#)

Active Directory をサポートする Oracle9i ディレクトリ・サーバーの機能

Oracle9i には、ディレクトリ・サーバーを利用する 2 つの機能があります。これらの機能については次の各項で簡単に説明します。

- [ディレクトリ・ネーミング](#)
- [エンタープライズ・ユーザー・セキュリティ](#)

これらの機能は、両方とも Microsoft 社の Active Directory で動作します。

ディレクトリ・ネーミング

この機能によって、クライアントは、Active Directory などの LDAP 準拠のディレクトリ・サーバーに集中的に格納されている情報を利用して、データベース・サーバーに接続できます。たとえば、以前は [tnsnames.ora](#) ファイルに格納されていた [ネット・サービス名](#) を Active Directory に格納できるようになりました。

注意： Oracle Names Server に格納されているデータベース・サービスと ネット・サービス名のエントリは、Oracle Names Server 制御ユーティリティを使用してディレクトリ・サーバーに移行できます。詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。

エンタープライズ・ユーザー・セキュリティ

この機能を使用すると、Oracle9i データベースの情報を LDAP 準拠のディレクトリ・サーバーのディレクトリ・オブジェクトとして作成および格納できます。たとえば、Oracle9i データベースのエンタープライズ・ユーザーおよびロールをディレクトリに作成および格納できます。これによって、管理者は複数のデータベースにわたるユーザーとロールの管理を集中化できます。

この付録では、読者が[エンタープライズ・ユーザー](#)のセキュリティに関する用語および概念について理解していることを前提としています。次の用語について理解していない場合は、[第 15 章「エンタープライズ・ユーザー・セキュリティの管理」](#) および [第 19 章「Oracle Enterprise Security Manager の使用方法」](#) で確認してください。

- エンタープライズ・ユーザー、ロール、ドメインおよび関連する概念
- エンタープライズ・ユーザーのセキュリティと管理
- ディレクトリ・サーバー内のエンタープライズ・ユーザー・セキュリティ・エントリの位置
- エンタープライズ・ユーザー・セキュリティのインストールと構成
- エンタープライズ・ユーザー、ロールおよびドメインの作成と管理

注意： [エンタープライズ・ロール](#)の管理に Active Directory を使用する
には、Oracle Advanced Security のライセンス契約を結ぶ必要があります。

注意： Oracle Enterprise Security Manager で、Windows 2000、
Windows NT、Windows 95 または Windows 98 オペレーティング・シス
テムのユーザー名を作成または削除することはできません。かわりに、
Oracle Enterprise Security Manager では Active Directory に接続名が作成
されます。接続名でログインすることはできません。この名前は、単に外
部の目的のために定義されているものです。このユーザーにロールを割り
当てることはできます。

Active Directory との統合

ディレクトリ・ネーミングやエンタープライズ・ユーザー・セキュリティのディレクトリ・
サーバーとの統合に加え、具体的には、次の Oracle9i の機能が Active Directory と統合され
ています。

- [Active Directory の概要](#)
- [ディレクトリ・サーバーの自動検出](#)
- [Microsoft のツールとの統合](#)
- [Oracle Net ディレクトリ・ネーミングのユーザー・インタフェースの機能拡張](#)
- [ディレクトリ・オブジェクト型記述の拡張機能](#)
- [Windows ログイン資格証明との統合](#)
- [Active Directory の Oracle ディレクトリ・オブジェクト](#)

Active Directory の概要

Active Directory は、Windows 2000 に組み込まれている LDAP 準拠のディレクトリ・サーバーです。Active Directory には、ユーザー、グループおよびポリシーも含めた Windows 2000 の情報がすべて格納されます。また、Active Directory には、ネットワーク・リソース（データベースなど）に関する情報も格納され、アプリケーション・ユーザーやネットワーク管理者にこの情報が提供されます。Active Directory によって、ユーザーは単一のログインでネットワーク・リソースにアクセスできます。Active Directory の対象は、小規模なコンピュータ・ネットワークの全リソースの格納から、複数の Wide Area Network (WAN) に関する全リソースの格納まで広範囲にわたります。

LDAP を使用して Active Directory をサポートする Oracle の機能を使用する場合は、ドメイン・コントローラに接続できる TCP/IP ホスト名形式のすべてを使用して、Active Directory コンピュータに正常に接続できることを確認してください。たとえば、ドメイン・コントローラのホスト名が、ドメイン `acme.com` の `server1` である場合は、次のすべての形式を使用してそのコンピュータに ping できることを確認してください。

- `server1.acme.com`
- `acme.com`
- `server1`

Active Directory ではそれ自体に戻す参照が、これらの 1 つ以上の形式で実行されている操作に応じて頻繁に発行されます。すべての形式が Active Directory コンピュータへの接続に使用できない場合は、一部の LDAP 操作で障害が発生する可能性があります。

ディレクトリ・サーバーの自動検出

Oracle Net Configuration Assistant を使用すると、ディレクトリ・サーバーにアクセスするクライアント・コンピュータおよび Oracle9i データベース・サーバーを構成できます。Oracle Net Configuration Assistant が Oracle9i データベースのインストールの最後に起動した場合、またはインストール後に手動で起動した場合は、使用するディレクトリ・サーバーのタイプを指定するためのプロンプトが表示されます。ディレクトリ・サーバーのタイプに Active Directory を選択すると、Oracle Net Configuration Assistant では次の操作が自動的に行われます。

- Active Directory サーバー位置の検出
- Active Directory サーバーへのアクセスの構成
- Oracle コンテキスト（ドメインとも呼ばれる）の作成

クライアント接続の Oracle9i データベースへのアクセスに使用している Active Directory サーバーが停止すると、別の Active Directory サーバーが自動的に検出され、接続情報の提供を開始します。この自動検出によって、クライアント接続の停止時間が最小化されます。

注意： Oracle Net Configuration Assistant のディレクトリ・サーバー自動検出機能を利用するには、使用している Oracle クライアントとデータベースのリリースに関係なく、Windows 2000 のドメインで実行する必要があります。Windows 2000 のドメインで実行していない場合、Oracle Net Configuration Assistant では、ディレクトリ・サーバーが自動検出されず、Active Directory の位置などの追加情報を要求するプロンプトが表示されます。

Active Directory に対するディレクトリの使用構成を完了するために Oracle Net Configuration Assistant を使用している場合は、Active Directory の表示が 24 種のデフォルト言語すべてに移入されないため、Oracle スキーマの作成に失敗する可能性があります。24 言語すべての表示指定子に移入されていることを、次の構文をコマンド・プロンプトで入力することで確認してから、Oracle Net Configuration Assistant を実行してディレクトリ・アクセス構成を完了してください。

```
ldifde -p OneLevel -d cn=DisplaySpecifiers,cn=Configuration,domain
context -f temp file
```

ここで、

- *domain context* は、Active Directory サーバーのドメイン・コンテキストです (dc=acme、dc=com など)。
- *temp file* は、出力を書き込むファイルです。

このコマンドで、検出されたエントリは 24 未満であったことがレポートされた場合でも、Oracle Net Configuration Assistant を継続して使用できます。ただし、一部の言語の表示指定子が作成されていなかった場合は、Oracle スキーマの作成に失敗したことがレポートされます。

Microsoft のツールとの統合

Active Directory の Oracle9i データベース・サービス、ネット・サービス名およびエンタープライズ・ロールのエントリは、2 つの Windows 2000 ツールで表示およびテストできます。

- Windows エクスプローラ
- Active Directory ユーザーとコンピュータ

Windows エクスプローラは、コンピュータのファイル、ディレクトリ、ローカル・ドライブおよびネットワーク・ドライブを階層構造で表示します。また、Oracle9i データベース・サービスやネット・サービス名オブジェクトを表示およびテストできます。

Active Directory ユーザーとコンピュータは、ドメイン・コントローラとして構成された Windows サーバーにインストールされている管理ツールです。このツールを使用すると、Windows 2000 のアカウントとグループを追加、変更、削除、編成したり、組織のディレクトリにあるリソースを公開できます。また、Windows エクスプローラと同様に、Oracle9i データベース・サービスやネット・サービス名オブジェクトを表示およびテストできます。さらに、アクセス制御を管理することもできます。

関連項目：

- E-18 ページ「[Microsoft のツールによる接続性テスト](#)」
- E-21 ページ「[Oracle ディレクトリ・オブジェクトのアクセス制御リストの管理](#)」

Oracle Net ディレクトリ・ネーミングのユーザー・インタフェースの機能拡張

Windows エクスプローラおよび Active Directory ユーザーとコンピュータの Oracle9i データベース・サービスやネット・サービス名オブジェクトのプロパティ・メニューの機能が拡張されました。これらの Oracle ディレクトリ・オブジェクトを右クリックすると、接続性をテストする 2 つの新しいオプションが表示されるようになりました。

- Test
- Connect with SQL*Plus

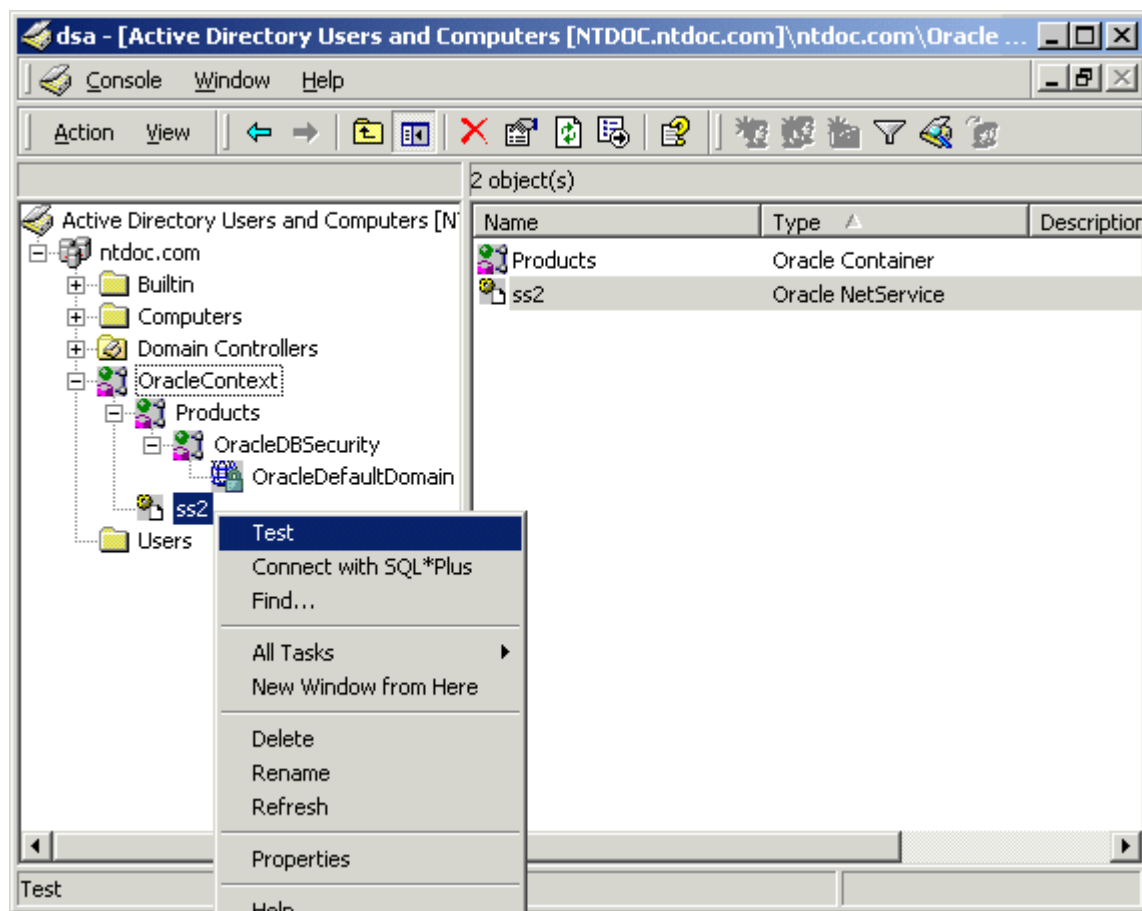
「Test」オプションは、最初に入力したユーザー名、パスワードおよびネット・サービス名で、Oracle9i データベースに実際に接続できるかどうかをテストするアプリケーションを起動します。「Connect with SQL*Plus」オプションは、データベースの管理、スクリプトの実行などを実施できる SQL*Plus を起動します。

関連項目： 詳細は、E-18 ページの「[Microsoft のツールによる接続性テスト](#)」を参照してください。

ディレクトリ・オブジェクト型記述の拡張機能

Active Directory の Oracle ディレクトリ・オブジェクト型記述が、簡単に理解できるように機能拡張されました。たとえば、図 E-1 の右側のペインにある「Type」列は、ss2 が Oracle NetService であることを示しています。

図 E-1 Active Directory のディレクトリ・オブジェクト型記述



Windows ログイン資格証明との統合

Oracle9i データベースと構成ツールでは、現在 Windows にログオンしているユーザーのログイン資格証明を使用して（つまり、ログイン資格証明を再入力せずに）、Active Directory に自動的に接続できます。この機能には、2 つの利点があります。

- Oracle9i のクライアントとデータベースは、Active Directory に安全に接続し、ネット・サービス名、エンタープライズ・ユーザーおよびエンタープライズ・ロールの情報を取得できます。
- Oracle の構成ツールは、Active Directory に自動的に接続して、Oracle9i データベースおよびネット・サービス名オブジェクトを構成できます。使用可能なツールには、Oracle Enterprise Security Manager、Oracle Net Configuration Assistant および Database Configuration Assistant が含まれます。

Active Directory の Oracle ディレクトリ・オブジェクト

Oracle9i データベースと **Oracle Net Services** が Active Directory にアクセスするようにインストールおよび構成されている場合、Active Directory ユーザーとコンピュータでは、Oracle ディレクトリ・オブジェクトが図 E-2 のように表示されます。

図 E-2 Active Directory ユーザーとコンピュータでの Oracle ディレクトリ・オブジェクト

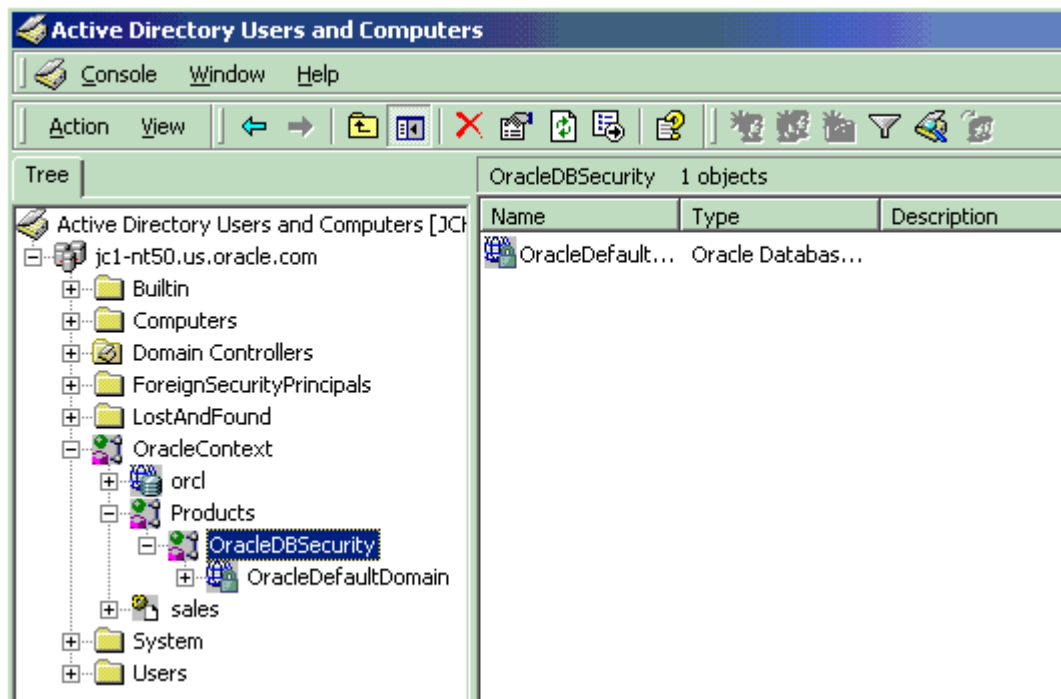


表 E-1 は、図 E-2 に表示されている Oracle ディレクトリ・オブジェクトを示しています。

表 E-1 Oracle ディレクトリ・オブジェクト

オブジェクト	説明
jc1-nt50.us.oracle.com	Oracle コンテキストに作成したドメイン。このドメイン（管理コンテキストとも呼ばれる）には、ディレクトリ・ネーミングやエンタープライズ・ユーザー・セキュリティをサポートするための様々な Oracle エントリが含まれます。この情報は、Oracle9i データベースと Active Directory の統合時に、Oracle Net Configuration Assistant によって自動的に検出されます。
OracleContext	Active Directory ツリーの最上位レベルの Oracle エントリ。Oracle9i データベース・サービスやネット・サービス名オブジェクトの情報が含まれています。Oracle のソフトウェア情報すべてがこのフォルダに配置されます。
orcl	この例における Oracle9i データベース・サービス名。
Products	Oracle 製品情報のフォルダ。

表 E-1 Oracle ディレクトリ・オブジェクト（続き）

オブジェクト	説明
OracleDBSecurity	データベース・セキュリティ情報のフォルダ。
OracleDefaultDomain	作成されたデフォルトのエンタープライズ・ドメイン。Oracle Enterprise Security Manager を使用すると、追加のエンタープライズ・ドメインを作成できます。
sales	この例におけるネット・サービス名オブジェクト。
Users	3 つの Oracle セキュリティ・グループのフォルダ。詳細は、E-21 ページの「Oracle ディレクトリ・オブジェクトのアクセス制御リストの管理」を参照してください。Oracle Enterprise Security Manager で作成されたエンタープライズ・ユーザーやロールも、このフォルダに表示されます。

Active Directory で Oracle9i を使用するための要件

Oracle Net ディレクトリ・ネーミングまたはエンタープライズ・ユーザー・セキュリティを Active Directory で使用する場合は、Microsoft と Oracle の特定のリリースのソフトウェアが必要です。また、Oracle スキーマと Oracle コンテキストも作成する必要があります。これらの要件については、次の各項で説明します。

- Oracle スキーマの作成
- Oracle コンテキストの作成
- ディレクトリ・ネーミング・ソフトウェアの要件
- エンタープライズ・ユーザー・セキュリティソフトウェアの要件

注意： Oracle スキーマと Oracle コンテキストは、Oracle Net Configuration Assistant を実行することで、作成できます。

注意： Oracle Net ディレクトリ・ネーミングおよびエンタープライズ・ユーザー・セキュリティを Active Directory に統合するには、使用している Oracle クライアントおよび Oracle データベース・サーバーのリリースには関係なく、Windows 2000 のドメインで実行する必要があります。

Windows 2000 または Windows NT 上の Oracle で Active Directory を使用している場合は、Windows 2000 ドメインの DNS ドメイン名を ping してください。動作しない場合は、次のいずれかの作業を実行します。

- Windows 2000 プライマリ・ドメイン・コントローラの IP アドレスを DNS として設定します。

たとえば、使用している Windows 2000 ドメインが sales の場合、このドメインの DNS ドメイン名は sales.acme.com です。IP アドレスは 001.002.003.0 の書式です。

- 使用している Windows 2000 ドメインの DNS ドメイン名およびドメイン・コントローラの IP アドレスを hosts ファイルまたは lmhosts ファイルに追加します。

Windows 2000 の場合は、DNS として 001.002.003.0 を設定するか、または 001.002.003.0 sales.acme.com を hosts ファイルまたは lmhosts ファイルに追加できます。

この手順を実行しないと、Active Directory の使用時に、次のようなエラーが発生します。

Cannot Chase Referrals

Windows NT および Windows 2000 の場合、Oracle データベース・サービスは、LocalSystem または特定のローカルまたはドメイン・ユーザーのセキュリティ・コンテキストで実行されます。Oracle8i リリース 8.1.7 を Active Directory で使用する場合、データベース・サービスが LocalSystem のセキュリティ・コンテキストで実行されているときは、データベース・サービスを実行しているコンピュータ名を手動で追加します。この追加によって、OracleDBSecurity コンテナ・オブジェクトに対する読取り権限で、Active Directory の OracleDBSecurity コンテナ・オブジェクトの制御エントリにアクセスできるようになります。

たとえば、データベース・サービス OracleServiceORCL が、コンピュータ mypc1 の LocalSystem のセキュリティ・コンテキストで実行されている場合は、OracleDBSecurity オブジェクトに対する読取り権限を持つ mypc1 を OracleDBSecurity コンテナ・オブジェクトのアクセス制御エントリに追加します。

Oracle スキーマの作成

Oracle Net ディレクトリ・ネーミングおよびエンタープライズ・ユーザー・セキュリティの機能を Active Directory で使用するには、Oracle スキーマを作成する必要があります。スキーマとは、Oracle Net Services および Oracle9i データベースのエントリに対する一連のルールで、その属性は Active Directory に格納されます。Active Directory で使用する Oracle スキーマの作成には、次の制限が適用されます。

- 各フォレストに対して作成できる Oracle スキーマは 1 つのみです。
- スキーマの作成は、Windows 2000 ドメイン・コントローラで行う必要があります。
- Windows 2000 ドメイン・コントローラは、スキーマを更新できる操作マスタである必要があります。詳細は、Windows オペレーティング・システムのマニュアルを参照してください。

Oracle スキーマを作成する手順は、次のとおりです。

1. スキーマ管理者グループのメンバーでログインします。デフォルトの場合、ドメイン管理者はスキーマ管理者グループにいます。
2. Oracle Net Configuration Assistant を使用して、Oracle スキーマを作成します。スキーマは、データベースのインストール時、またはその後で作成できます。

関連項目：

- 構成手順は、『Oracle9i Net Services 管理者ガイド』を参照してください。
- 構成の概要は、『Oracle9i Database for Windows インストレーション・ガイド』を参照してください。

Oracle コンテキストの作成

Oracle Net ディレクトリ・ネーミングおよびエンタープライズ・ユーザー・セキュリティの機能を Active Directory で使用するには、Oracle コンテキストを作成する必要があります。Oracle コンテキストとは、Active Directory ツリーの最上位レベルの Oracle エントリです。Oracle9i データベース・サービスや Oracle Net サービス名オブジェクトの情報が含まれています。

- Windows 2000 の各ドメイン（管理コンテキスト）に対して作成できる Oracle コンテキストは 1 つのみです。
- Oracle Net Configuration Assistant を使用して Active Directory に Oracle コンテキストを作成するには、ドメイン・オブジェクトを作成する権限が必要です。ドメイン管理者であれば、これらの権限は自動的に付与されています。
- Oracle Net Configuration Assistant を使用して、Oracle コンテキストを作成します。Oracle コンテキストは、Oracle9i データベースのカスタム・インストール時またはインストール後に作成できます。

関連項目：

- インストール手順は、『Oracle9i Database for Windows インストレーション・ガイド』を参照してください。
- 構成手順は、『Oracle9i Net Services 管理者ガイド』を参照してください。

ディレクトリ・ネーミング・ソフトウェアの要件

Oracle9i エンタープライズ・ユーザー、ロールおよびドメインを管理するクライアント・コンピュータの場合は、Oracle8i Client リリース 8.1.6 以上および次の Microsoft 社製品のいずれかが必要です。

- Windows 2000
- Windows NT 4.0 および **Active Directory Service Interfaces (ADSI)**
- Windows 95 または 98 および分散システム・クライアントのアップグレード

データベース・サーバーの場合は、Oracle8i Database リリース 8.1.6 以上が必要です。これは、データベース・サービスを Active Directory のオブジェクトとして登録するために必要です。データベース・サーバーには、次の Microsoft 社製品のいずれかを使用できます。

- Windows 2000
- Windows NT 4.0 および ADSI

これらのソフトウェア要件に加え、クライアント・コンピュータとデータベース・サーバーは Windows 2000 ドメインで実行する必要があります。

エンタープライズ・ユーザー・セキュリティソフトウェアの要件

データベース・サーバーの場合は、Oracle8i Database リリース 8.1.6 以上が必要です。これは、データベース・サービスを Active Directory のオブジェクトとして登録するために必要です。データベース・サーバーには、次の Microsoft 社製品のいずれかを使用できます。

- Windows 2000
- Windows NT 4.0 および ADSI

リモート・コンピュータには、リリース 2.1 以上の Oracle Enterprise Manager のコンソールおよび次のコンポーネントが必要です。

- Oracle Enterprise Security Manager
- Oracle Net Services

リモート・コンピュータには、次の Microsoft 社製品のいずれかを使用できます。

- Windows 2000
- Windows NT 4.0 および ADSI

これらのソフトウェア要件に加え、リモート・コンピュータとデータベース・サーバーは Windows 2000 ドメインで実行する必要があります。

注意： エンタープライズ・ユーザー、ロールおよびドメインの作成と管理には、Oracle Enterprise Security Manager が必要です。Oracle Enterprise Security Manager がシステム固有の認証を Active Directory への接続に使用する場合、ホスト・コンピュータは Windows 2000 ドメインにある必要があります。また、ユーザーは Windows 2000 ドメイン・ユーザーでホスト・コンピュータにログインする必要があります。

Active Directory を使用するための Oracle9i の構成

Oracle9i と Active Directory の統合によって、オペレーティング・システムのユーザー認証とロールの認可を利用できます。次のタスクを実行して、Oracle のコンポーネントを Active Directory に統合します。

- [タスク 1: コンポーネントのインストールと構成](#)
- [タスク 2: OSAUTH_X509_NAME レジストリ・パラメータの設定](#)
- [タスク 3: Oracle Enterprise Security Manager の起動と使用](#)

注意： オペレーティング・システムのユーザー認証とロールの認可を使用できるのは、Windows 2000 ドメインで実行している場合のみです。

タスク 1: コンポーネントのインストールと構成

E-3 ページの「[Active Directory との統合](#)」、E-10 ページの「[Active Directory で Oracle9i を使用するための要件](#)」および『Oracle9i Database for Windows インストレーション・ガイド』を参照して、インストール前の問題点と構成に関する問題点を確認してください。

タスク 2: OSAUTH_X509_NAME レジストリ・パラメータの設定

OSAUTH_X509_NAME レジストリ・パラメータを TRUE に設定し、クライアント・ユーザーが X.509 準拠のエンタープライズ・ユーザーで Oracle9i データベースにアクセスできるようにします。クライアントのユーザー名と許可ロールの識別に、Active Directory が使用されます。このパラメータ設定が必要なのは、エンタープライズ・ユーザーとロールを使用する場合のみです。

パラメータが FALSE（デフォルト設定）の場合、クライアント・ユーザーは、外部ユーザーとして識別され、ユーザー・ロールの認可には Oracle9i データベースの[データ・ディクショナリ](#)が使用されます。

OSAUTH_X509_NAME レジストリ・パラメータを設定する手順は、次のとおりです。

1. Oracle9i データベースがインストールされているコンピュータに移動します。
2. 「スタート」→「ファイル名を指定して実行」を選択します。
3. 「名前」フィールドに regedt32 を入力して、「OK」をクリックします。
「レジストリ・エディタ」ウィンドウが表示されます。
4. HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOMEID に移動します。
この ID が、編集対象の Oracle ホームです。

5. レジストリ値 `OSAUTH_X509_NAME` が存在している場合は、`OSAUTH_X509_NAME` をダブルクリックします。

「文字列の編集」ダイアログ・ボックスが表示されます。

レジストリ値がない場合は、`OSAUTH_X509_NAME` を、タイプ `REG_EXPAND_SZ` のレジストリ値として追加します。

6. 「Enter」をクリックします。
7. 「文字列」フィールドに `TRUE` の値を設定します。
8. 「OK」をクリックします。
9. 「レジストリ」メニューの「レジストリ・エディタの終了」をクリックします。

レジストリ・エディタが終了します。

タスク 3: Oracle Enterprise Security Manager の起動と使用

Oracle Enterprise Security Manager は、Oracle Enterprise Manager と統合されたアプリケーションとして組み込まれます。Oracle Enterprise Security Manager を使用すると、エンタープライズ・ユーザー、ロールおよびドメインを作成および管理できます。また、エンタープライズ・ユーザーとグループをエンタープライズ・ロールに割り当てることもできます。

Oracle Enterprise Security Manager を使用する管理者は、セキュリティ・グループ `OracleDBSecurityAdmin` のメンバーである必要があります。デフォルトでは、Oracle コンテキストを作成した（つまり、ディレクトリ・サーバーで動作するように Oracle9i データベースを構成した）管理者が、このセキュリティ・グループのメンバーです。Oracle Enterprise Security Manager の機能すべてを使用する許可が与えられているのは、このセキュリティ・グループのメンバーのみです。他のユーザーを手動で追加するには、E-21 ページの「[Oracle ディレクトリ・オブジェクトのアクセス制御リストの管理](#)」を参照してください。

ディレクトリ・サーバーのメイン・メニューで「Login」を選択し、環境に適した認証プロトコルを選択するダイアログ・ボックスにアクセスします。Windows NT 4.0 または Windows 2000 の Active Directory を備えた Windows 2000 ドメインで Oracle9i データベースを実行している場合は、「NT Native Authentication」を選択します。Windows 2000 ドメインで実行している場合、Oracle Enterprise Security Manager では、Windows システム固有の認証が自動的に使用されます。

利用可能な他の選択が動作しない場合は、「Simple Authentication」を選択します。簡易認証は Oracle Internet Directory または Active Directory のいずれでも使用できますが、安全性は低くなります。

接続性のテスト

この項では、Active Directory を介して Oracle9i データベースに接続する方法について説明します。項目は、次のとおりです。

- クライアント・コンピュータからの接続性テスト
- Microsoft のツールによる接続性テスト

クライアント・コンピュータからの接続性テスト

Oracle Net ディレクトリ・ネーミングを使用している場合、クライアント・コンピュータは、Oracle コンテキストに表示されるデータベースまたはネット・サービス名のエントリを指定することで、データベースに接続します。たとえば、Active Directory の Oracle コンテキストの下にデータベース・エントリが orcl で、クライアントと Oracle9i データベースが同一のドメインにある場合、ユーザーは、次の**接続文字列**を入力することで、SQL*Plus を介してデータベースに接続します。

```
SQL> CONNECT scott/tiger@orcl
```

クライアントと Oracle9i データベースが異なるドメインにある場合、ユーザーは、次の構文を入力することで、SQL*Plus を介してデータベースに接続します。

```
SQL> CONNECT scott/tiger@orcl.domain
```

この domain は、Oracle9i データベースが配置されているドメインです。

これらの接続文字列は、DNS スタイル規則に従います。Active Directory は X.500 ネーミング規則を使用した接続もサポートしていますが、より簡単に使用できる理由から、DNS スタイル規則の使用をお勧めします。

DNS スタイル規則によって、クライアント・ユーザーは、クライアント・コンピュータと Oracle9i データベースが別のドメインにある場合でも、最小の接続情報の入力でディレクトリ・サーバーを介して Oracle9i データベースにアクセスできます。クライアントと Oracle9i データベースが異なるドメイン（管理コンテキストとも呼ばれる）に配置されている場合は特に、X.500 規則に従った名前は、DNS スタイル規則の場合よりも長くなります。

関連項目： X.500 ネーミング規則の詳細は、『Oracle9i Net Services 管理者ガイド』の「構成管理の概念」を参照してください。

Microsoft のツールによる接続性テスト

Active Directory の Oracle ディレクトリ・オブジェクトは、Microsoft の 2 つのツールと統合されています。

- Windows エクスプローラ
- Active Directory ユーザーとコンピュータ

Microsoft のこれらのツールから、次のタスクを実行できます。

- SQL*Plus による Oracle9i データベースへの接続
- Oracle9i データベース接続性のテスト

接続性をテストする手順は、次のとおりです。

1. Windows エクスプローラまたは Active Directory ユーザーとコンピュータを起動します。

Windows エクスプローラを起動する手順は、次のとおりです。

- a. 「スタート」→「プログラム」→「アクセサリ」→「Windows エクスプローラ」を順に選択します。
- b. 「マイ・ネットワーク」を展開します。
- c. 「ネットワーク全体」を展開します。
- d. 「Directory」を展開します。

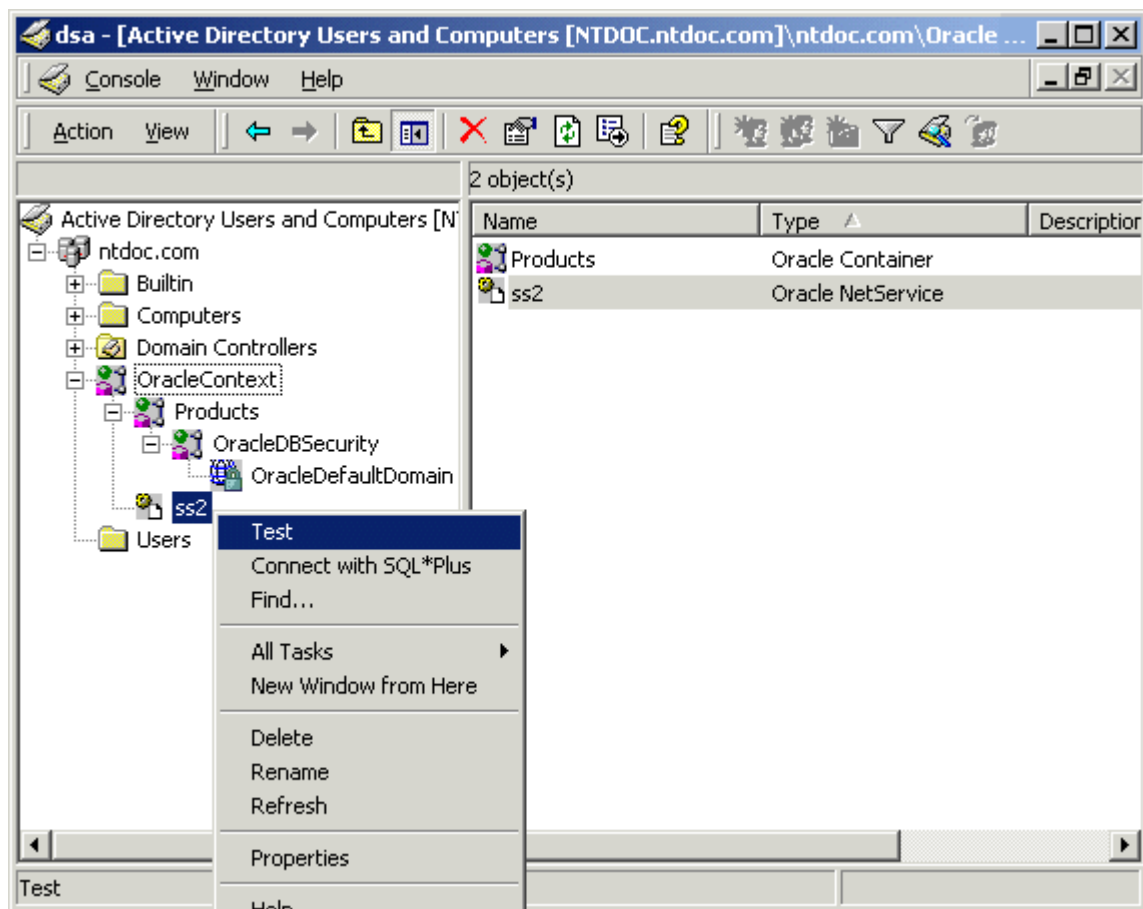
Active Directory ユーザーとコンピュータを起動する手順は、次のとおりです。

「スタート」→「プログラム」→「管理ツール」→「Active Directory ユーザーとコンピュータ」を順に選択します。

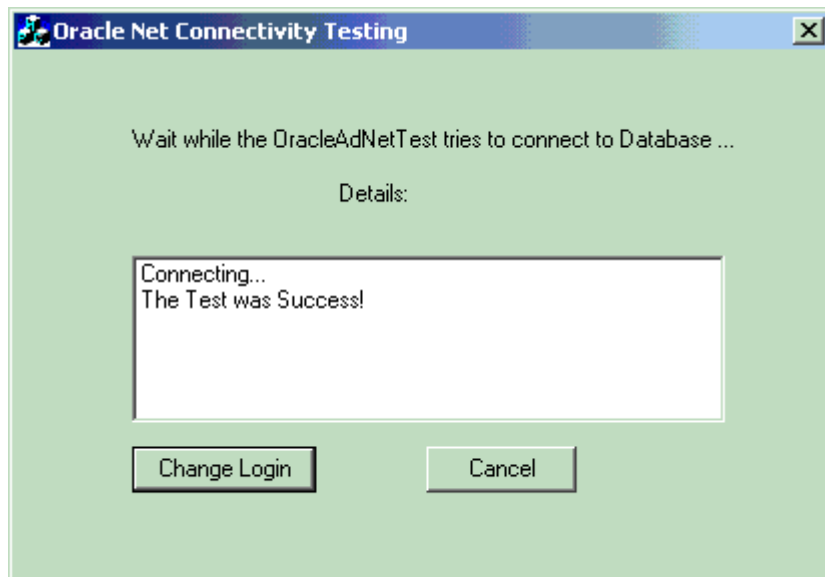
注意： Active Directory を介して Oracle9i データベースにアクセスするすべてのクライアントには、Oracle コンテキストのネット・サービス名オブジェクトすべてに対する読み込みアクセス権限が必要です。また、Active Directory を使用して匿名で認証できる必要もあります。Oracle Net Configuration Assistant では、この設定が自動的に行われます。

2. Oracle コンテキストが配置されているドメインを展開します。
3. Oracle コンテキストを展開します。
4. データベース・サービスまたは Oracle Net サービス名オブジェクトを右クリックします。

様々なオプションとともにメニューが表示されます。ここでは、「Test」と「Connect with SQL*Plus」の 2 つのオプションが関係します。

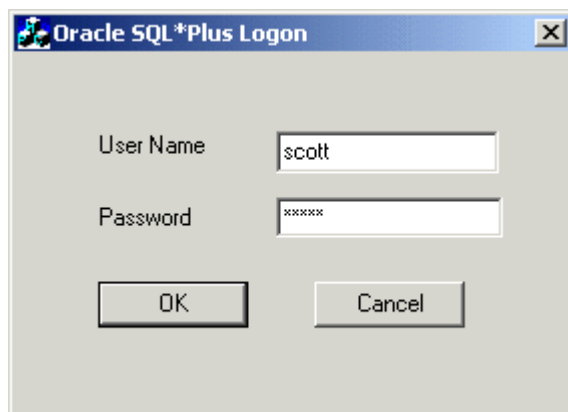


5. データベース接続を実際の接続なしにテストする場合は、「Test」を選択します。
接続テストのステータスを示すステータス・メッセージが表示されます。



6. 実際に接続してデータベース接続をテストする場合は、「Connect with SQL*Plus」を選択します。

「Oracle SQL*Plus Logon」ダイアログ・ボックスが表示されます。



7. ユーザー名とパスワードを入力します。

接続テストのステータスを示すステータス・メッセージが表示されます。

Oracle ディレクトリ・オブジェクトのアクセス制御リストの管理

次の指定を行うことで、アクセス制御リストでは Active Directory のセキュリティが提供されます。

- オブジェクトのオブジェクト属性にアクセスできるユーザー
- エントリにアクセスするための認証方式
- アクセス権限、またはオブジェクトのオブジェクト属性（読取り / 書込み）を使用してユーザーが実行できる内容

セキュリティ・グループ

Oracle コンテキストが Active Directory に作成されると、3 つのセキュリティ・グループが自動的に作成されます。アクセスを構成している（つまり、Oracle コンテキストを作成している）ユーザーは、各グループに自動的に追加されます。グループは、次のとおりです。

- [OracleDBSecurityAdmin](#)
- [OracleDBCcreator](#)
- [OracleNetAdmins](#)

OracleDBSecurityAdmin

OracleDBSecurityAdmin グループは、Oracle コンテキスト作成者用のグループです。このグループのユーザーは、次のことができます。

- 3 つのセキュリティ・グループすべてに関するグループ・メンバーシップの管理
- Oracle コンテキスト内のオブジェクトの管理
- エンタープライズ・ドメインの作成（Oracle Enterprise Security Manager を使用）

OracleDBCcreator

OracleDBCcreator グループは、Oracle9i データベース作成者用のグループです。ドメイン管理者は、自動的にこのグループのメンバーになります。このグループのユーザーは、次のことができます。

- Oracle コンテキストでの Oracle9i データベース・オブジェクトの新規作成
- 作成した Oracle9i データベース・オブジェクトの変更
- このグループのメンバーシップの読込み（変更はできません）

OracleNetAdmins

OracleNetAdmins グループのユーザーは、次のことができます。

- Oracle Net Services のオブジェクトと属性の作成、変更および読み込み
- このグループのグループ・メンバーシップの読み込み

セキュリティ・グループへのアクセス

Active Directory ユーザーとコンピュータを使用すると、3つのセキュリティ・グループすべてについて、ユーザーの追加や削除、または権限の設定内容を変更できます。

OracleDBSecurityAdmin および OracleDBCcreator では Oracle Enterprise Security Manager を使用することもできます (OracleNetAdmins では使用できません)。

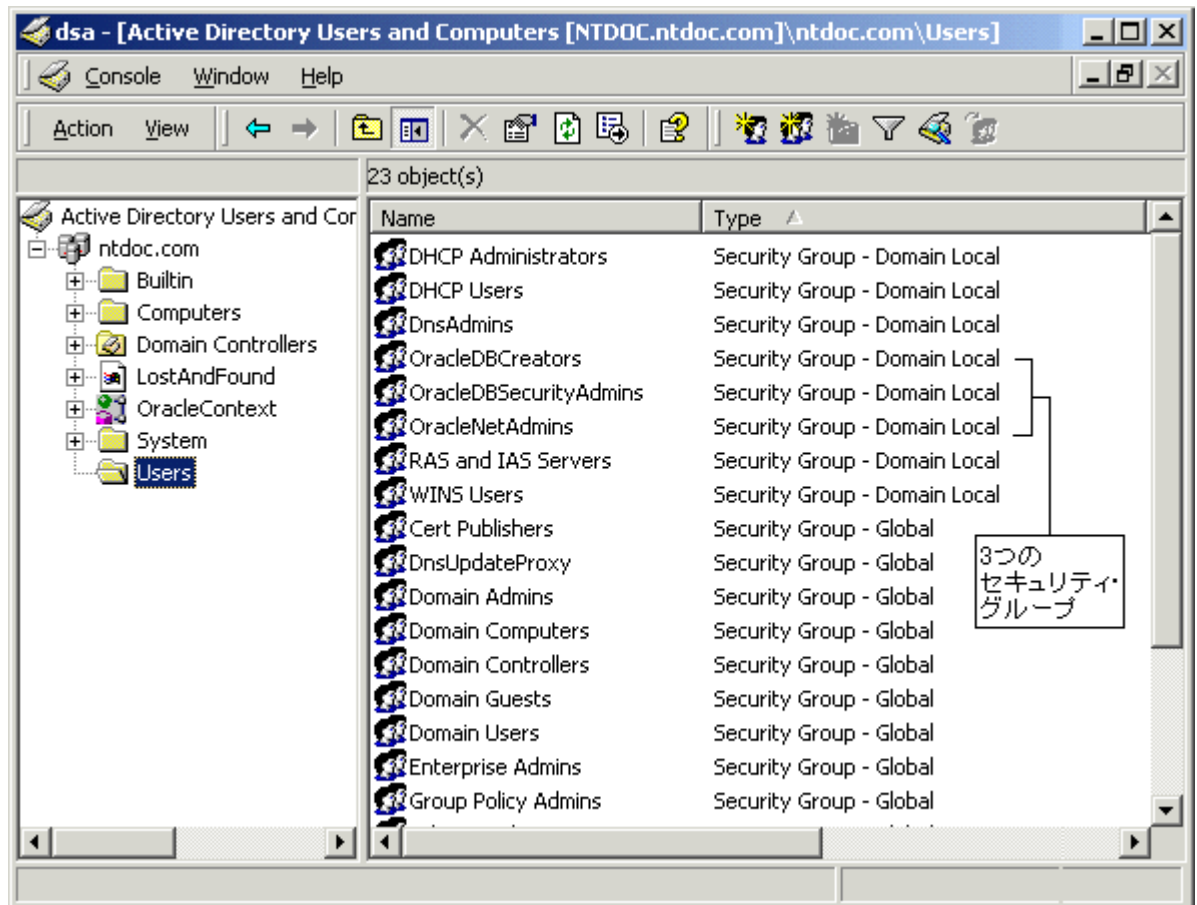
この項では、Active Directory ユーザーとコンピュータの使用方法について説明します。

注意： この項で説明する手順を実行するには、Active Directory ユーザーとコンピュータを使用してください。Windows エクスプローラには、この手順に必要な機能がありません。

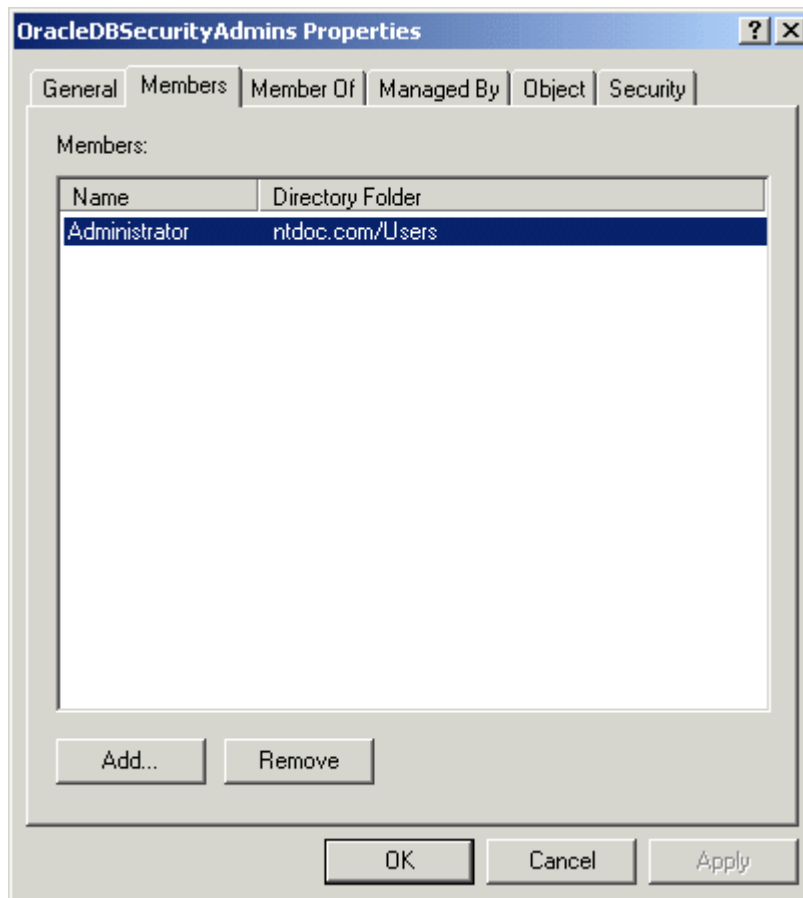
ユーザーの追加や削除、または権限の設定内容を変更する手順は、次のとおりです。

1. 「スタート」→「プログラム」→「管理ツール」→「Active Directory ユーザーとコンピュータ」を順に選択します。
2. 「View」メイン・メニューの「Advanced Features」を選択します。
この選択によって、通常は非表示になっている情報の表示と編集が可能になります。
3. Oracle コンテキストが配置されているドメイン（管理コンテキスト）を展開します。
4. 「Users」を展開します。

ウィンドウ右側のペインに3つのセキュリティ・グループが表示されます。



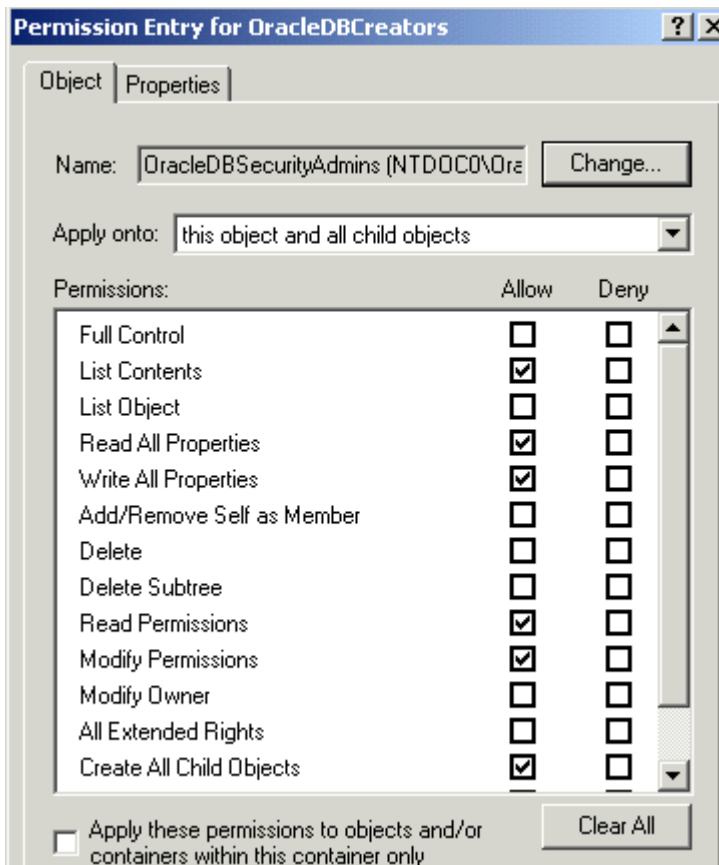
5. 表示または変更する Oracle セキュリティ・グループを右クリックします。
様々なオプションとともにメニューが表示されます。
6. 「Properties」を選択します。
7. 権限を変更する場合は、手順 13 までスキップします。
ユーザーを追加または削除する場合は、手順 8 に進みます。
8. 「Members」タブを選択します。
選択したグループのプロパティ・ダイアログ・ボックスが表示されます（この例では、「OracleDBSecurityAdmins Properties」）。



9. ユーザーを追加するには、「Add」をクリックします。
「Select Users, Contacts, Computers, or Groups」ダイアログ・ボックスが表示されます。
10. 追加するユーザーまたはグループを選択して「Add」をクリックします。
「Select Users, Contacts, Computers, or Groups」ダイアログ・ボックスに選択したユーザーまたはグループが表示されます。
11. ユーザーを削除するには、「Members」リストでユーザー名を選択して「Remove」をクリックします。
12. ユーザーの追加または削除を終了した後、「OK」をクリックします。

13. ユーザーの権限を変更するには、「Properties」ダイアログ・ボックスの「Security」タブを選択します。
14. 「Advanced」をクリックします。
15. 「View/Edit」をクリックします。

選択したセキュリティ・グループの「Permission Entry」ダイアログ・ボックスが表示されます。



16. グループの権限に対して必要な変更を行います。
17. 「OK」をクリックします。

エンタープライズ・ドメインの作成

デフォルトのエンタープライズ・ドメインである `OracleDefaultDomain` は、Oracle コンテキストに作成されます。このドメインを使用しない場合、または別のドメインを作成する場合は、Oracle Enterprise Security Manager を使用して追加のエンタープライズ・ドメインを作成します。これらのドメインは、`OracleDBSecurity` フォルダの下に追加されます。

Java SSL の Oracle 実装

この付録では、Java Secure Socket Extension (JSSE) の Oracle 実装について説明します。項目は、次のとおりです。

- [前提条件](#)
- [Oracle Java SSL 機能](#)
- [Oracle Java SSL の例](#)
- [Oracle Java SSL のトラブルシューティング](#)
- [Oracle Java SSL の API](#)

注意： この付録では、読者が Java ソケット・プログラミングの基礎と SSL プロトコルの基本原理について理解していることを前提としています。

関連項目： Java SSL パッケージの詳細は、Sun Microsystems 社の Web サイトにある Java のマニュアルを参照してください。

<http://java.sun.com/products/jsse>

前提条件

Oracle Java SSL 実装を使用するには、次の作業を行う必要があります。

- JDK のバージョン 1.1 以上をインストールします。
- 環境変数 CLASSPATH に、次の jar ファイルが含まれていることを確認します。
 - JDK1.1 の場合: javax-ssl-1_1.jar、jssl-1_1.jar
 - JDK1.2 以上の場合: javax-ssl-1_2.jar、jssl-1_2.jar
- Oracle Java SSL 共有ライブラリをプラットフォームの共有ライブラリ・パスに追加します。
 - **UNIX の場合:** 環境変数 LD_LIBRARY_PATH で指定されているライブラリ・パスに libnjs19.so を含める必要があります。
 - **Windows の場合:** 環境変数 PATH で指定されているパスに njssl9.dll を含める必要があります。
- CLASSPATH に JSSE バージョン 1.0.2 がある場合は、jssl-1_1.jar を使用して ssl.SocketFactory.provider と ssl.ServerSocketFactory の Java セキュリティ・プロパティを、次のように設定します。

```
ssl.SocketFactory.provider=oracle.security.ssl.OracleSSLSocketFactoryImpl
ssl.ServerSocketFactory.provider=oracle.security.ssl.OracleSSLServerSocketFactoryImpl
```

- jsse.jar または jcert.jar が Java の extensions フォルダにインストールされている場合は、jssl-1_1.jar も extensions フォルダにインストールする必要があります。

関連項目： 使用しているオペレーティング・システム固有のマニュアルを参照してください。

Oracle Java SSL 機能

Oracle Java SSL は、JSSE の商用レベルの実装です。Oracle Java SSL では、安全で高速な SSL 実装を作成するために、ネイティブ・コードを使用してパフォーマンスを改善しています。

JSSE 仕様で規定されている機能の他に、Oracle Java SSL は次のものをサポートしています。

- 複数の暗号化アルゴリズム
- Oracle Wallet Manager による証明書と鍵の管理
- Oracle Java SSL の最上部のアプリケーションで使える SSL セッション機能（認証を含む）

次の項で、Oracle Java SSL の機能の詳細について説明します。

- [Oracle Java SSL でサポートされた SSL Cipher Suite](#)
- [Oracle Wallet Manager での証明書と鍵管理](#)
- [セキュリティを意識したアプリケーションのサポート](#)

Oracle Java SSL でサポートされた SSL Cipher Suite

SSL 接続の中をデータが流れるには、接続の両側で、データ送信に使用する共通のアルゴリズムについて折衝が行われる必要があります。複数のセキュリティ機能を併用するために組み合わせたアルゴリズムのセットのことを、[Cipher Suite](#) と呼びます。SSL 接続に参加しているシステムは、特定の Cipher Suite を選択することで、その通信に適切なレベルを設定できます。

Oracle Java SSL では、次のオプションを含む Cipher Suite をサポートしています。

- 512、768 または 1024 ビットの非対称鍵の鍵交換。次のアルゴリズムを使用できます。
 - RSA
 - Diffie-Hellman
- 40 ビットおよび 128 ビットの対称鍵による NULL 暗号化または対称鍵暗号化。次のアルゴリズムを使用できます。
 - RC4 ストリーム暗号化
 - DES、DES40 および 3DES-EDE ([暗号ブロック連鎖](#) (CBC) モード)

注意： NULL 暗号化における SSL の用途は、認証とデータの整合性のみです。

- MD5 または SHA1 データ整合性を使用したメッセージ認証コード。

Oracle Wallet Manager での証明書と鍵管理

公開鍵と秘密鍵のペアと証明書要求を生成するには、Oracle Wallet Manager を使用します。完全な Oracle Wallet を作成するには、署名された証明書要求と適切な信頼できる証明書を追加する必要があります。

Oracle Wallet Manager のメニューで「**Operation**」→「**ExportWallet**」を選択すると、Ready 状態の証明書を含む完全な Wallet を BASE64 形式のファイルでエクスポートできます。このファイルを使用して、SSL 資格証明を Java SSL ベースのプログラムに追加できます。

Oracle Wallet Manager を使用せずに、個々の構成要素を手動でファイルに追加することもできます。

- 最初に**証明書**を追加し、次に**秘密鍵**を追加します。
- その後で、**認証局**証明書と他の**信頼できる証明書**を追加します。

関連項目：

- Java SSL での資格証明の設定方法は、F-17 ページの「**パブリック・クラス : OracleSSLCredential**」を参照してください。
- 第 17 章「Oracle Wallet Manager の使用方法」

セキュリティを意識したアプリケーションのサポート

セキュリティを意識したアプリケーションの中には、**トラスト・ポイント**を設定しないものがあります。Oracle Java SSL では、このようなアプリケーションが独自の検証を実行できるように、接続先から完全な**証明連鎖**が送られてきている場合は、セキュリティ資格証明がなくてもハンドシェイクを完了できます。この機能は、データベースに数多くのトラスト・ポイントが格納されており、アプリケーションが SSL レイヤーにそれらすべてのトラスト・ポイントを渡す必要がある場合に役立ちます。

ハンドシェイクが完了すると、接続先証明連鎖を取得して、個々の接続先証明書を抽出できます。これらの証明書を使用して、証明書の**識別名**とユーザー・データベースとの照合など、アプリケーション固有の検証を実行できます。

セキュリティを意識していないアプリケーションでトラスト・ポイントのチェックが必要な場合は、アプリケーションでトラスト・ポイントを確実に設定する必要があります。

関連項目： 接続先資格証明のチェック方法は、F-17 ページの「**パブリック・クラス : OracleSSLCredential**」を参照してください。

Oracle Java SSL の例

この項の例では、Oracle Java SSL の使用方法を具体的に示します。これらの例で使用するために、それぞれ `SSLServerExample` および `SSLClientExample` という名前のサーバーとクライアントのモデルを作成しています。この 2 つのシステムによって、Oracle Java SSL の共通機能やソケット通信の基本について説明します。また、`SSLProxyClientExample` によって、ファイアウォール・トンネリング接続の実装方法の一例を示します。

プログラムごとに完全なコードを掲載し、重要な部分には説明を加えています。

注意： この例は、Oracle Java SSL で使用できるすべての機能を網羅するものではありません。

次の項で、Oracle Java SSL の例を説明します。

- [例 : SSLServerExample プログラム](#)
- [例 : SSLClientExample プログラム](#)
- [例 : SSLProxyClientExample プログラム](#)

関連項目：

- このパッケージで利用できる他のセキュリティ・オプションの詳細は、この付録の後半の項を参照してください。
- ソケット・プログラミングに関する全般的な情報は、F-20 ページの「[パブリック・クラス : OracleSSLServerSocketFactoryImpl](#)」を参照してください。
- ソケットとソケット・ストリームの詳細は、`java.net` パッケージの Java マニュアルを参照してください。

例 : SSLServerExample プログラム

SSLServerExample は、単純な SSL サーバーです。SSLServerExample は、Oracle Wallet Manager からエクスポートされた Wallet を使用して、自身のセキュリティ資格証明を設定します。このサーバーは、起動した後、クライアントからの接続開始要求を待ちます。SSL ハンドシェイクが完了すると、SSLServerExample は、クライアントに短いメッセージを送信して、接続をクローズします。

プログラムには、次のコードが含まれています。

```
import oracle.security.ssl.*;
import java.net.*;
import java.io.*;
import java.util.*;
import javax.net.*;
import javax.net.ssl.*;

public class SSLServerExample
{
    private OracleSSLServerSocketFactoryImpl _socketFactory;
    private OracleSSLCredential _credential;
    private SSLServerSocket _svrSoc;

    private void initCredential(String wltPath, String password)
        throws java.io.IOException
    {
        _credential = new OracleSSLCredential();
        _credential.setWallet(wltPath, password);
    }

    private void initSocketFactory()
        throws javax.net.ssl.SSLException
    {
        _socketFactory
            = (OracleSSLServerSocketFactory) SSLServerSocketFactory.getDefault();
        _socketFactory.setSSLProtocolVersion(
            OracleSSLProtocolVersion.SSL_Version_3_0_With_2_0_Hello);
        _socketFactory.setSSLCredentials(_credential);
    }

    private void initServerSocket(int port)
        throws java.io.IOException
    {
        _svrSoc = (SSLServerSocket) _socketFactory.createServerSocket(port);
        _svrSoc.setUseClientMode(false);
        _svrSoc.setNeedClientAuth(false);
        _svrSoc.setEnabledCipherSuites(new String[]{"SSL_RSA_WITH_RC4_128_SHA",
```

```
        "SSL_RSA_WITH_RC4_128_MD5"}));
    }

    public SSLServerExample(String wltPath, String password, int port)
        throws java.io.IOException, javax.net.ssl.SSLException
    {
        initCredential(wltPath, password);
        initSocketFactory();
        initServerSocket(port);
    }

    public void runServer()
    {
        String message = "Hello! Current Server Time is " + new Date() + "\n";
        Socket csocket = null;
        OutputStreamWriter out = null;
        try
        {
            csocket = _svrSoc.accept();
            out = new OutputStreamWriter(csocket.getOutputStream());
            out.write(message);
            System.out.println("Connection Succeeded");
        }
        catch(IOException e)
        {
            System.out.println("Connection Failed");
            e.printStackTrace();
        }
        finally
        {
            try
            {
                if(out != null)
                    out.close();
                if(csocket != null)
                    csocket.close();
                _svrSoc.close();
            }
            catch(IOException e){}
        }
    }

    public static void main(String[] argv)
    {
        System.getProperties().put("SSLServerSocketFactoryImplClass",
            "oracle.security.ssl.OracleSSLServerSocketFactoryImpl");
        try
```

```

    {
        SSLServerExample myServer = new SSLServerExample("mywallet.txt",
            "welcome1", 19978);
        myServer.runServer();
    }
    catch(IOException i)
    {
        System.out.println("Failed to start up server");
        i.printStackTrace();
    }
}
}

```

資格証明の初期化

SSLServerExample は Oracle Wallet Manager で作成された Wallet を使用するの、資格証明オブジェクトの設定ジョブは非常に簡単です。(wltPath) に配置されている Wallet を読み込むには、initCredential() で次のようにコールします。

```

_credential = new OracleSSLCredential();
_credential.setWallet(wltPath, password);

```

接続では、Wallet 内に格納されている **秘密鍵**、ユーザー **証明書**、**証明書**および**トラスト・ポイント**が使用されます。Wallet へのアクセス時にエラーが起これと、IOException が戻ります。

Wallet を使用しない場合は、必要なセキュリティ資格証明を手動でインストールできます。

関連項目： addTrustedCert()、addCertChain() および
setPrivateKey() の詳細は、F-17 ページの「**パブリック・クラス：**
OracleSSLCredential」を参照してください。

ソケット・ファクトリの初期化

SSL ソケットを作成するには、正しいソケット・ファクトリにアクセスする必要があります。Oracle Java SSL で javax.net.ServerSocketFactory を実装しているクラスの名前は、oracle.security.ssl.OracleSSLSocketFactoryImpl です。正しいソケット・ファクトリに確実にアクセスできるように、次の設定を使用して、main() 関数でシステム・プロパティを設定します。

```

System.getProperties().put("SSLServerSocketFactoryImplClass","oracle.security.ssl.Or
acleSSLServerSocketFactoryImpl");

```

システム・プロパティを設定した後、ソケット・ファクトリのインスタンスを取得して、それをカスタマイズできます。initSocketFactory() では、このファクトリによって作成されたソケットがサポートする SSL プロトコルを指定し、このファクトリによって作成されたすべてのソケットで使用するセキュリティ資格証明をインストールしています。

サーバー・ソケットの初期化

メソッド `initServerSocket()` では、次のように、ソケット・ファクトリを使用して、指定したポートをサーバー・モードでリスニングする新規のサーバー・ソケットを作成しています。

```
_svrSoc = (SSLServerSocket)_socketFactory.createServerSocket(port);  
_svrSoc.setUseClientMode(false);
```

ソケットを作成した後は、次のプロパティを変更してその属性の一部を変更できます。

```
_svrSoc.setNeedClientAuth(false);  
_svrSoc.setEnabledCipherSuites(new String[]{"SSL_RSA_WITH_RC4_128_SHA"  
"SSL_RSA_WITH_RC4_128_MD5"});
```

この例の場合、クライアントはサーバーに対してクライアント自身を認証する必要はありません。ただし、デフォルトで使用可能な **Cipher Suite** は使用されず、`SSL_RSA_WITH_RC4_128_SHA` または `SSL_RSA_WITH_RC4_128_MD5` の **Cipher Suite** をサポートするクライアントのみが接続を許可されます。

Java SSL でサポートされている **Cipher Suite** を確認するには、`OracleSSLServerSocketFactory.getSupportedCipherSuites()` を使用します。

関連項目：

- F-3 ページ「[Oracle Java SSL でサポートされた SSL Cipher Suite](#)」
- F-20 ページ「[パブリック・クラス：OracleSSLServerSocketFactoryImpl](#)」

接続の待機とデータの送信

`SSLServerExample` は、クライアントがサーバーに接続するまで待機します。接続が確立されるまで、メソッド `accept()` によってブロックされています。クライアントの接続が確立されると、`getOutputStream()` をコールしてソケットの出力ストリームを取得します。この出力ストリームを使用して、情報をクライアントに送信します。サーバーからクライアントに送信するデータがなくなると、サーバーは対応する出力ストリームとソケットをクローズします。サーバーが接続の受入れを停止するには、対応するサーバー・ソケットをクローズする必要があります。サーバーは、これ以上接続を受け入れることができなくなると、ソケットをクローズします。

関連項目： ソケットとソケット・ストリームの詳細は、`java.net` パッケージの Java マニュアルを参照してください。

例 : SSLClientExample プログラム

SSLClientExample は、SSLServerExample プログラムへの接続に使用する単純なプログラムです。JDK バージョン 1.1 を使用しています。SSLClientExample の初期化は、サーバーの初期化と非常によく似ています。ただし、この例では、Oracle Java SSL の機能の一部を説明するために、多少の相違点が含まれています。必要に応じて、説明の部分でこれらの相違点を中心に上げています。プログラム例には、次のコードが含まれています。

```
import oracle.security.ssl.*;
import java.net.*;
import java.io.*;
import java.util.*;
import javax.net.*;
import javax.net.ssl.*;
import javax.security.cert.*;

public class SSLClientExample
{
    protected OracleSSLSocketFactoryImpl _socketFactory;
    private OracleSSLCredential _credential;
    protected SSLSocket _socket;

    private void initCredential(String wltPath, String password)
        throws java.io.IOException
    {
        _credential = new OracleSSLCredential();
        _credential.setWallet(wltPath, password);
    }

    private void initSocketFactory()
        throws javax.net.ssl.SSLException
    {
        _socketFactory
            = (OracleSSLSocketFactoryImpl) SSLSocketFactory.getDefault();
        _socketFactory.setSSLProtocolVersion(
            OracleSSLProtocolVersion.SSL_Version_3_0);
        _socketFactory.setSSLCredentials(_credential);
    }

    private void initSocket(String host, int port)
        throws java.io.IOException
    {
        _socket = (SSLSocket)_socketFactory.createSocket(host, port);
        _socket.setUseClientMode(true);
    }

    public SSLClientExample(String wltPath, String pass, String host, int port)
        throws java.io.IOException, javax.net.ssl.SSLException
    {
        initCredential(wltPath, pass);
        initSocketFactory();
        initSocket(host, port);
    }
}
```

```
{
    initCredential(wltPath, pass);
    initSocketFactory();
    initSocket(host, port);
}

public void connectSocket()
{
    try
    {
        _socket.startHandshake();
        getData();
    }
    catch(IOException e)
    {
        System.out.println("Connection Failed");
        e.printStackTrace();
    }
    finally
    {
        try
        {
            _socket.close();
        }
        catch(IOException e){}
    }
}

public void getData()
{
    InputStreamReader in = null;
    try
    {
        int ch;
        SSLSession session = _socket.getSession();

        System.out.println("Negotiated Cipher Suite " +
            session.getCipherSuite());
        X509Certificate[] peerCerts = session.getPeerCertificateChain();
        for(int i = 0; i < peerCerts.length; i++)
        {
            System.out.println(peerCerts[i]);
        }
        System.out.println("Server Response:");
        in = new InputStreamReader(_socket.getInputStream());
        ch = in.read();
        while((char)ch != '\n')
```

```
        {
            if(ch != -1)
                System.out.print((char)ch);
            ch=in.read();
        }
        System.out.println();
    }
    catch(IOException e)
    {
        System.out.println("Connection Failed");
        e.printStackTrace();
    }
    finally
    {
        try
        {
            if(in != null)
                in.close();
        }
        catch(IOException e){}
    }
}

public static void main(String[] argv)
{
    System.getProperties().put("SSLSocketFactoryImplClass",
        "oracle.security.ssl.OracleSSLSocketFactoryImpl");
    try
    {
        SSLClientExample myClient = new
            SSLClientExample("mywallet.txt","welcome1","localhost", 19978);
        myClient.connectSocket();
    }
    catch(IOException i)
    {
        System.out.println("Failed to start up client");
        i.printStackTrace();
    }
}
}
```

注意： JDK バージョン 1.2 を使用する場合は、import javax.security.cert.*; を import java.security.cert.*; に変更してください。

資格証明の初期化

クライアントは、サーバーと同じ方法で資格証明を初期化します。ここでは、例を示すという趣旨に沿って、クライアントとサーバーで同じ **Wallet** を使用しています。ただし、本番のアプリケーションでは、クライアントとサーバーは異なるセキュリティ資格証明を持つ必要があります。SSL 接続が正常に完了するためには、**Wallet** 内に正しい**信頼できる証明書**が含まれていることが重要です。

関連項目： 信頼できる証明書の詳細は、[第 17 章「Oracle Wallet Manager の使用方法」](#)を参照してください。

ソケット・ファクトリの初期化

クライアント・ソケットの作成に使用するソケット・ファクトリは、サーバーで使用されているものと似ています。SSLServerExample プログラムと同様に、正しいソケット・ファクトリを取得するためには、`initSocketFactory()` でソケット・ファクトリを構成する前にシステム・プロパティを設定する必要があります。次の構文を使用して、`main()` で正しいソケット・ファクトリを設定します。

```
System.getProperties().put("SSLSocketFactoryImplClass",  
"oracle.security.ssl.OracleSSLSocketFactoryImpl");
```

クライアント・ソケットの初期化と接続

クライアント・ソケットは、サーバー・ソケットがサーバー・ソケット・ファクトリによって作成されるのと同様に、ソケット・ファクトリによって作成されます。ただし、クライアント・ソケットを特定のサーバーに接続するには、作成時にサーバーの名前とポート番号を入力する必要があります。また、ソケットがクライアント・モードで接続されるように、次の設定が指定されていることを確認します。

```
_socket = (SSLSocket)_socketFactory.createSocket(host, port);  
_socket.setUseClientMode(true);
```

ソケットを作成した後は、次のメソッドを使用して、ソケットをサーバーに接続できます。

```
_socket.startHandshake();
```

接続先資格証明の表示

ソケットがサーバーに接続した後は、接続に関する情報にアクセスできます。情報は、`OracleSSLSession` クラスに格納されています。このクラスのインスタンスは、`_socket.getSession()` を使用して取得できます。

この例では、クライアントとサーバーの間で折衝された **Cipher Suite** と、サーバーのセキュリティ資格証明が出力されます。この情報は、接続が信頼できるものであるかどうかを判断するために、セキュリティを意識したアプリケーションで使用できます。たとえば、ほとんどのブラウザでは、サーバー証明書内の共通名とアクセス先の **URL** が一致していることが確認され、一致していない場合は警告が表示されます。ただし、このチェックは、SSL プロトコルにとって必須ではありません。

データの受信

SSL ソケットを介したデータの送受信は、他のソケットを介したデータの受信と特に違いはありません。この例では、ソケットの入力ストリームにアクセスし、行端文字が現れるまで読み込みます。

関連項目： ソケットとソケット・ストリームの詳細は、`java.net` パッケージの Java マニュアルを参照してください。

例 : SSLProxyClientExample プログラム

この例は、ファイアウォールのトンネリングを使用して、サーバーへの安全な接続を確立します。このプログラムは、必ずしもすべてのファイアウォールで動作するとはかぎりません。たとえば、一部のファイアウォールでは、この例で使用されているポート 19978 などの非標準のポートへの接続が許可されていません。この場合は、ポート 443 上で安全なサーバーを設定し、それに応じてクライアントを変更する必要があります。

```
import oracle.security.ssl.*;
import java.net.*;
import java.io.*;
import java.util.*;
import javax.net.*;
import javax.net.ssl.*;
import javax.security.cert.*;

public class SSLProxyClientExample extends SSLClientExample
{
    private String _proxyName;
    private int _proxyPort;

    protected void initSocket(String host, int port)
        throws java.io.IOException
    {
        final String connString = "CONNECT" + host + ":" + port +
            " HTTP/1.0 \n" + "User-Agent: Oracle Proxy Enabled SSL Socket\n\n";
        Socket normalSocket = new Socket(_proxyName, _proxyPort);
        OutputStreamWriter out
            = new OutputStreamWriter(normalSocket.getOutputStream());
        out.write(connString, 0, connString.length());
        _socket = (SSLSocket)_socketFactory.createSocket(normalSocket);
    }

    public SSLProxyClientExample(String wltPath, String password, String host,
        int port, String proxyName, int proxyPort)
        throws java.io.IOException, javax.net.ssl.SSLException
    {
        super(wltPath, password, host, port);
        _proxyName = proxyName;
    }
}
```

```
        _proxyPort = proxyPort;
    }

    public static void main(String[] argv)
    {
        System.getProperties().put("SSLSocketFactoryImplClass",
            "oracle.security.ssl.OracleSSLSocketFactory");
        try{
            SSLClientExample myClient
                = new SSLProxyClientExample("mywallet.txt", "welcome1",
                    "localhost", 19978, "www-proxy", 80);
            myClient.connectSocket();
        }
        catch(IOException i)
        {
            System.out.println("Failed to start up client");
            i.printStackTrace();
        }
    }
}
```

注意： JDK バージョン 1.2 を使用する場合は、`import javax.security.cert.*;` を `import java.security.cert.*;` に変更してください。

クライアント・ソケットの初期化と接続

SSLProxyClientExample とそのスーパークラスである SSLClientExample との間の唯一の重要な相違点は、メソッド `initSocket()` にあります。トンネリング接続を設定するには、プレーンなソケットを作成する必要があります。このソケットを使用して、ファイアウォールに特別なメッセージ `connString` を送信し、実際のサーバーへの接続を設定します。この接続を設定した後は、次の構文を入力して、プレーンなソケットを使用して SSL ソケットを初期化できます。

```
_socketFactory.createSocket(normalSocket)
```

Oracle Java SSL のトラブルシューティング

この項では、いくつかの典型的な Java SSL エラーについて説明します。

SSLException X509CertExpiredErr

ハンドシェイク時に *SSLException* が発生し、メッセージ *X509CertExpiredErr* が戻り、プログラムが失敗します。前回の時点ではプログラムは正常に動作しており、それ以降変更は行っていない。

原因：ユーザー証明書の期限が切れています。

処置：ユーザー証明書を新たに取得する必要があります。

関連項目： [第 17 章「Oracle Wallet Manager の使用方法」](#)

SSLException X509CertChainInvalidErr

ハンドシェイク時に *SSLException* が発生し、メッセージ *X509CertChainInvalidErr* が戻り、クライアント側でハンドシェイクが失敗します。Web ブラウザは、サーバーに正常に接続できます。

原因：サーバーまたはクライアントが正しい資格証明を持っていません。

処置：クライアント・プログラムで信頼できる証明書を設定している場合は、そのリストに、サーバーの証明連鎖内の証明書が少なくとも 1 つ含まれていることを確認します。また、Oracle Java SSL は証明連鎖自体を作成できないため、サーバーからクライアントに完全な証明連鎖が送られていることを確認します。Apache サーバーを使用している場合は、変数 *SSLCertificateChainFile* と *SSLCertificateFile* を正しく設定します。これは、クライアント・プログラムで信頼できる証明書を設定していない場合に特に重要です。

関連項目： 詳細は、Web サーバーのマニュアルを参照してください。

資格証明のないクライアント接続

クライアント・プログラムで *OracleSSLCredentials* が設定されていない場合でも、ハンドシェイクは成功します。

原因：Oracle Java SSL では、セキュリティを意識したアプリケーションが独自の検証を実行できるように、サーバーから完全な証明連鎖が送られてきている場合にのみ、クライアントで資格証明が設定されていなくても接続が許可されます。

処置：このような動作を回避するには、アプリケーションで少なくとも 1 つの信頼できる証明書を設定する必要があります。

関連項目： [F-17 ページ「パブリック・クラス : OracleSSLCredential」](#)

Oracle Java SSL の API

この項では、Oracle Java SSL で使用されるパブリック・クラスとインタフェースについて説明します。Oracle Java SSL は JSSE の実装であるため、JSSE パッケージに追加された Oracle 独自の部分について説明します。

この項では、次の Oracle Java SSL クラスとインタフェースについて説明します。

- パブリック・クラス : `OracleSSLCredential`
- パブリック・インタフェース : `OracleSSLProtocolVersion`
- パブリック・クラス : `OracleSSLServerSocketFactoryImpl`
- パブリック・クラス : `OracleSSLSession`
- パブリック・クラス : `OracleSSLSocketFactoryImpl`
- パブリック・インタフェース : `OracleX509TrustManagerInterface`

関連項目 :

JSSE クラスの詳細は、次の Web サイトを参照してください。

<http://java.sun.com/products/jsse/doc/apidoc/index.html>

パブリック・クラス : `OracleSSLCredential`

このパブリック・クラスは、`java.lang.Object` を拡張しています。

資格証明は、サーバーとクライアントを相互に認証するために使用します。

`OracleSSLCredential` クラスは、BASE64 または DER でエンコードされた証明書から、ユーザー証明書、信頼できる証明書（トラスト・ポイント）および秘密鍵をロードするために使用します。

コンストラクタ

```
public OracleSSLCredential()
```

空の `OracleSSLCredential` 資格証明を作成します。ソケットは、空の資格証明を使用して、ハンドシェイク時に完全な証明連鎖を送信する任意の接続先と接続できます。

メソッド

```
public void addTrustedCert(java.lang.String b64TrustedCert)
```

資格証明に信頼できる証明書を追加します。

パラメータ: b64TrustedCert – BASE64 でエンコードされた X.509 証明書

```
public void addTrustedCert(byte[] trustedCert)
```

資格証明に信頼できる証明書を追加します。

パラメータ: trustedCert – DER でエンコードされた信頼できる X.509 証明書

```
public void setPrivateKey(java.lang.String b64PvtKey, java.lang.String password)
```

資格証明に秘密鍵を追加します。

パラメータ: b64PvtKey – BASE64 でエンコードされた X.509 秘密鍵

password – 秘密鍵の解読に必要なパスワード

```
public void setPrivateKey(byte[] pvtKey, java.lang.String password)
```

資格証明に秘密鍵を追加します。

パラメータ: b64PvtKey – DER でエンコードされた X.509 秘密鍵

password – 秘密鍵の解読に必要なパスワード

```
public void addCertChain(java.lang.String b64certChainCert)
```

証明連鎖に証明書を追加します。証明連鎖は、SSL ハンドシェイク時にユーザー証明書とともに送信されます。そして、ユーザー証明書を検証するために、接続先で使用されます。証明連鎖に最初に追加する証明書は、ルート CA 証明書である必要があります。その後に追加する証明書は、直前の証明書によって署名されている必要があります。

パラメータ: b64certChainCert – BASE64 でエンコードされた X.509 証明書

```
public void addCertChain(byte[] certChainCert)
```

証明連鎖に証明書を追加します。

パラメータ: certChainCert — DER でエンコードされた X.509 証明書

```
public void setWallet(java.lang.String wltPath, java.lang.String password)
    throws java.io.IOException
```

Oracle Wallet Manager を使用して作成した Wallet は、テキスト・フォーマットでエクスポートでき、Oracle Java SSL で使用できます。テキスト・ファイルには、必ずユーザー証明書が含まれ、その後に秘密鍵、証明連鎖およびその他の信頼できる証明書が続きます。Wallet がオープンできない場合は、`java.io.IOException` が発生します。

パラメータ: wltPath — Wallet のパス名

password — 秘密鍵の解読に必要なパスワード

パブリック・インタフェース: `OracleSSLProtocolVersion`

このインタフェースは、使用可能な SSL プロトコルのバージョンを定義します。

フィールド

```
public static final int SSL_Version_Undetermined
```

SSL プロトコルのバージョンが未確定

```
public static final int SSL_Version_3_0_With_2_0_Hello
```

SSL プロトコル・バージョン 3.0 と 2.0 hello

```
public static final int SSL_Version_3_0_Only
```

SSL プロトコル・バージョン 3.0 のみ

```
public static final int SSL_Version_2_0
```

SSL プロトコルバージョン 2.0

```
public static final int SSL_Version_3_0
```

SSL プロトコル・バージョン 3.0

パブリック・クラス : **OracleSSLServerSocketFactoryImpl**

このパブリック・クラスは、`javax.net.ssl.SSLServerSocketFactory` を拡張しています。このクラスを使用して、SSL サーバー・ソケットを作成します。

このクラスは、サーバー・ソケットの作成に必要な

`javax.net.ssl.SSLServerSocketFactory` メソッドを実装しています。また、Oracle Java SSL 固有のオプションを構成するために必要な追加のメソッドを提供します。

コンストラクタ

```
public OracleSSLServerSocketFactoryImpl()
```

ソケットの作成に使用できるソケット・ファクトリを作成します。ただし、ソケット・ファクトリを作成する際は、システム・プロパティ `SSLServerSocketFactoryImplClass` を `oracle.security.sslOracleSSLServerSocketFactoryImpl` に設定する方法の使用をお勧めします。たとえば、次のように指定します。

```
System.getProperties().put("SSLServerSocketFactoryImplClass",  
    "oracle.security.ssl.OracleSSLServerSocketFactoryImpl");  
SSLServerSocketFactory factory = OracleSSLServerSocketFactoryImpl.getDefault();
```

メソッド

```
public void setSSLCredentials(OracleSSLCredential sslCredential) throws  
    javax.net.ssl.SSLException
```

SSL 接続で使用される `OracleSSLCredential`（秘密鍵、証明連鎖および同種のデータを保持する）を設定します。エラーが起きた場合は、`javax.net.ssl.SSLSocketException` が戻ります。

```
public void setSSLProtocolVersion(int version) throws javax.net.ssl.SSLException
```

SSL プロトコルのバージョンを設定します。SSL のバージョンがサポートされていない場合は、`javax.net.ssl.SSLSocketException` が発生します。

パブリック・クラス : OracleSSLSession

このパブリック・クラスは、`java.lang.Object` クラスを拡張しています。このクラスは、`javax.net.ssl.SSLSession` インタフェースを実装しています。

このクラスは `javax.net.ssl.SSLSession` で指定されているほとんどのメソッドを実装しています。ただし、次のメソッドは、このクラスでは実装されていません。

- `getPeerHost()`
- `getValue()`
- `invalidate()`
- `removeValue()`
- `getValueNames()`

このクラスは、Oracle Java SSL 固有の追加メソッドを提供します。これらのメソッドについては、次の各項で説明します。

メソッド

```
public byte[] [] getPeerRawCertificateChain() throws  
javax.net.ssl.SSLPeerUnverifiedException
```

DER 形式の接続先 [X.509](#) 証明書の配列として、接続先から送られてきた [証明連鎖](#) を返します。連鎖内の最初には接続先の証明書があり、最後にはルート CA があります。接続先証明書が検証できない場合は、`javax.net.ssl.SSLPeerUnverifiedException` が戻ります。

```
public java.lang.String getNegotiatedProtocolVersion()
```

このセッションで使用される SSL プロトコルのバージョンを返します。

パブリック・クラス : OracleSSLSocketFactoryImpl

このパブリック・クラスは、`javax.net.ssl.SSLSocketFactory` を拡張しています。

このクラスは、サーバー・ソケットの作成に必要な `javax.net.ssl.SSLSocketFactory` メソッドを実装しています。また、Oracle Java SSL 固有のオプションを構成するために必要な追加のメソッドを提供します。これらのメソッドについては、次の各項で説明します。

コンストラクタ

```
public OracleSSLSocketFactoryImpl()
```

ソケットの作成に使用できるソケット・ファクトリを作成します。ただし、ソケット・ファクトリを作成する際は、システム・プロパティ `SSLSocketFactoryImplClass` を `oracle.security.sslOracleSSLSocketFactoryImpl` に設定する方法の使用をお勧めします。たとえば、次のように指定します。

```
System.getProperties().put("SSLSocketFactoryImplClass",  
    "oracle.security.sslOracleSSLSocketFactoryImpl");  
SSLSocketFactory factory = OracleSSLSocketFactoryImpl.getDefault();
```

メソッド

```
public java.net.Socket createSocket(java.net.Socket socket) throws  
    java.io.IOException
```

既存のソケットを使用して読み書きを行う SSL ソケットの新規インスタンスを戻します。これは、ファイアウォールをトンネリングする場合に特に役立ちます。ソケットの作成時にエラーが起こった場合は、`java.io.IOException` が戻ります。

パラメータ : `socket` データを送信するためのソケット・オブジェクト

```
public void setSSLCredentials(OracleSSLCredential sslCredential) throws  
    javax.net.ssl.SSLException
```

SSL 接続で使用される `OracleSSLCredential` (秘密鍵、証明連鎖および同種のデータを保持する) を設定します。また、このメソッドは `OracleSSLCredential` の場合と同じトラスト・ポイントで、デフォルトの `OracleX509TrustManager` を作成および設定します。エラーが起こった場合は、`javax.net.ssl.SSLSocketException` が発生します。

```
public void setSSLProtocolVersion(int version) throws javax.net.ssl.SSLException
```

SSL プロトコルのバージョンを設定します。SSL のバージョンがサポートされていない場合は、`javax.net.ssl.SSLSocketException` が発生します。

```
public void setTrustManagers(OracleX509TrustManagerInterface[] tm)
```

このファクトリで作成されたソケットに使用する OracleX509TrustManagers を設定します。

パラメータ : tm — トラスト・マネージャの配列

```
public OracleX509TrustManagerInterface[] getTrustManagers()
```

このファクトリに設定されている X509TrustManagers を戻します。

パブリック・インタフェース : OracleX509TrustManagerInterface

このパブリック・インタフェースは `javax.net.ssl.TrustManager` を拡張しています。`javax.net.ssl.X509TrustManager` に基づいていますが、そのインタフェースは継承していません。

このインタフェースは、有効な証明連鎖を構築し、保護ソケットのリモート・サイドの認証に使用できる X.509 証明書を管理します。判断は、信頼できる認証局、証明書失効リスト、オンライン・ステータス・チェックまたは指定された他の手段に基づいて行われます。この関数は、信頼できる証明書が設定され、次の条件の 1 つを満たす場合にコールされます。

- 接続先証明連鎖に信頼できる証明書が含まれている場合
- 接続先証明連鎖に信頼できる証明書が含まれていない場合
- 接続先証明連鎖に期限切れの証明書が含まれている場合

注意： このような状況に対してカスタマイズされた動作が必要な場合は、特定の要件を満たすように設計されたこのインタフェースの実装をお勧めします。

メソッド

```
public abstract void checkClientTrusted(X509Certificate[] chain)
    throws CertificateException
```

接続先が提供する部分的または完全な証明連鎖の場合、このメソッドは、信頼できるルートへの証明書パスを作成し、その証明連鎖が検証できるかどうか、およびクライアントの SSL 認証に対して信頼できるかどうかを戻します。証明連鎖がこのトラスト・マネージャで信頼されない場合は、`javax.net.ssl.CertificateException` が発生します。

パラメータ : `chain` —最初に接続先自身の証明書がリストされ、次に認証局が続くように順序付けられた接続先 X.509 証明書の配列

```
public abstract void checkServerTrusted(X509Certificate[] chain) throws
    CertificateException
```

接続先が提供する部分的または完全な証明連鎖の場合、このメソッドは、信頼できるルートへの証明書パスを作成し、その証明連鎖が検証できるかどうか、およびサーバーの SSL 認証に対して信頼できるかどうかを戻します。証明連鎖がこのトラスト・マネージャで信頼されない場合は、`javax.net.ssl.CertificateException` が発生します。

パラメータ : `chain` —最初に接続先自身の証明書がリストされ、次に認証局が続くように順序付けられた接続先 X.509 証明書の配列

```
public abstract X509Certificate[] getAcceptedIssuers()
```

このメソッドは、認証する接続先に信頼されている認証局の証明書の配列を戻します。受入れ可能な認証局発行人の証明書について、NULL でない（おそらく空）の配列を戻します。

略称と頭字語

この付録では、このマニュアルで使用されている略称と頭字語を定義します（表 G-1）。

表 G-1 略称と頭字語

略称 / 頭字語	説明
3DES	3 重の暗号化を提供する DES 暗号化アルゴリズムのバージョンの 1 つ（トリプル DES を参照）。
ACL	アクセス制御リスト
CA	認証局
CBC	暗号ブロック連鎖
CDS	Cell ディレクトリ・サービス
CORBA	Common Object Request Broker Architecture
DCE	分散コンピューティング環境
DES	米国データ暗号化規格
DES40	40 ビットの暗号化鍵を使用したデータ暗号化規格
DES56	56 ビットの暗号化鍵を使用したデータ暗号化規格
DIT	ディレクトリ情報ツリー
DN	識別名
DNS	ドメイン・ネーム・サービス
FIPS	米国連邦情報処理標準
GDS	グローバル・ディレクトリ・サービス
GSSAPI	汎用セキュリティ・サービス・アプリケーション・プログラミング・インタフェース
HTTP	Hypertext Transfer Protocol
HTTPS	サブレイヤーに SSL レイヤーを組み込んだ HTTP
IIOP	Internet Inter-ORB Protocol
ISM	Bull Integrated System Management
ISP	インターネット・サービス・プロバイダ
JDBC	Java Database Connectivity
JDK	Java Development Kit
JRE	Java Runtime Environment
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol

表 G-1 略称と頭字語（続き）

略称 / 頭字語	説明
MD4	Message Digest 4。128 ビットのハッシュ合計（チェックサム）を生成するチェックサム・アルゴリズム。「MD5」を参照。
MD5	Message Digest 5。128 ビットのハッシュ合計（チェックサム）を生成するチェックサム・アルゴリズム。MD4 の後継であり、より強力なアルゴリズムを提供します。
NIST	米国商務省国立標準技術研究所
OCI	Oracle Call Interface
OSF	Open Software Foundation
PIN	個人識別番号
PKE	公開鍵エンコーディング（Public Key Encoding）
PKI	公開鍵インフラストラクチャ
RADIUS	Remote Authentication Dial-In User Service
RC4	RSA Data Security 社の対称型暗号化アルゴリズム
RPC	リモート・プロシージャ・コール
RSA	RSA Data Security 社（RSA 暗号化モジュールを参照）
SASL	Simple Authentication and Security Layer
SHA	Secure Hash Algorithm
SSL	Secure Sockets Layer
SSO	シングル・サインオン
WAN	Wide Area Network
トリプル DES	3 重の暗号化を提供する DES 暗号化アルゴリズムのバージョンの 1 つ（3DES を参照）。

用語集

Active Directory Service Interfaces (ADSI)

コンポーネント・オブジェクト・モデル (COM) に基づいたクライアント側の製品。ADSI は、ディレクトリ・サービス・モデルおよび一連の COM インタフェースを定義する。このインタフェースによって、Windows 2000、Windows NT、Windows 98 および Windows 95 のクライアント・アプリケーションは、Active Directory も含めた様々なネットワーク・ディレクトリ・サービスにアクセスできる。また、ADSI によって、アプリケーションは Active Directory と通信できる。

Advanced Encryption Standard

Advanced Encryption Standard (AES) は、DES に代わるものとして米国商務省国立標準技術研究所 (NIST) によって承認された新しい暗号化アルゴリズムである。AES 規格は、米国連邦情報処理標準 (FIPS) 刊行物 197 で入手できる。AES アルゴリズムは、128、192 および 256 ビット長の暗号鍵を使用して、128 ビットのデータ・ブロックを処理できる対称型ブロック暗号である。

AES

「[Advanced Encryption Standard](#)」を参照。

CDS

「[Cell ディレクトリ・サービス \(Cell Directory Services: CDS\)](#)」を参照。

Cell ディレクトリ・サービス (Cell Directory Services: CDS)

外部ネーミング・メソッドの 1 つ。これを使用すると、ユーザーは透過的に Oracle のツール製品を使用でき、アプリケーションによって分散コンピューティング環境 (DCE) にある Oracle9i データベースにアクセスできる。

Cipher Suite

ネットワークのノード間でメッセージ交換するのに使用する認証、暗号化、データ整合性アルゴリズムのセット。たとえば、SSL ハンドシェイク時に、2つのノード間で折衝し、メッセージを送受信するときに使用する Cipher Suite を確認する。

Cipher Suite 名 (cipher suite name)

特定のセッションでの接続で使用される暗号化保護の種類を示す Cipher Suite。

CORBA

Common Object Request Broker Architecture。オブジェクトと呼ばれるプログラムで、それが記述されているプログラミング言語や動作するオペレーティング・システムに関係なく、互いにやりとりできるようになるアーキテクチャ。CORBA は OMG (Object Management Group) として知られる業界のコンソーシアムで開発されたアーキテクチャである。

DCE

「[分散コンピューティング環境 \(Distributed Computing Environment: DCE\)](#)」を参照。

DES

「[データ暗号化規格 \(Data Encryption Standard: DES\)](#)」を参照。

Diffie-Hellman 鍵折衝アルゴリズム (Diffie-Hellman key negotiation algorithm)

このアルゴリズムを使用すると、非保護チャネルで通信を行う二者間で、関係者のみが認識する乱数を作成できる。関係者は、Diffie-Hellman 鍵折衝アルゴリズムの実行中は非保護チャネルで情報を交換できるが、攻撃者がネットワーク通信を分析し、関係者が合意した乱数を推定するのはほぼ不可能である。Oracle Advanced Security では、セッション鍵の生成に Diffie-Hellman 鍵折衝アルゴリズムが使用されている。

FIPS

「[米国連邦情報処理標準 \(Federal Information Processing Standard: FIPS\)](#)」を参照。

GDS

「[グローバル・ディレクトリ・サービス \(Global Directory Service: GDS\)](#)」を参照。

HTTP

Hypertext Transfer Protocol。WWW でファイル (テキスト、グラフィック・イメージ、サウンド、ビデオおよび他のマルチメディア・ファイル) を交換する際の規則セット。TCP/IP プロトコルがインターネットでの情報交換の基礎的なプロトコルであるのに対して、HTTP はアプリケーション・プロトコルである。

HTTPS

標準の HTTP アプリケーション・レイヤーのサブレイヤーとして SSL (Secure Sockets Layer) を使用したプロトコル。

IIOP

Internet Inter-ORB Protocol。OMG（Object Management Group）で開発されたプロトコルで、ワールド・ワイド・ウェブで CORBA ソリューションを実装するプロトコル。IIOP では、テキストの転送のみがサポートされる HTTP とは異なり、ブラウザとサーバーで整数、配列、その他の複雑なオブジェクトを交換できる。

Java Database Connectivity (JDBC)

Java プログラムからリレーショナル・データベースに接続するための業界標準の Java インタフェース。Sun Microsystems 社が定義した。

Java コード不明瞭化 (java code obfuscation)

Java コード**不明瞭化**は、Java プログラムをリバース・エンジニアリングから保護するために使用する。特別なプログラム（不明瞭化プログラム）を使用して、コード内の Java シンボルをスクランブルする。これにより、元のプログラムの構造はそのまま、その内容を隠すためにクラス、メソッドおよび変数の名前のみを変えて、プログラムが正しく実行されるようにする。不明瞭化されていない Java コードをデコンパイルして読み取るのは可能だが、不明瞭化された Java コードは、米国政府の輸出規制を満たすために、デコンパイルが困難になっている。

JDBC

「[Java Database Connectivity \(JDBC\)](#)」を参照。

KDC/TGS

Key Distribution Center/Ticket Granting Service。KDC は、Kerberos 認証でユーザー・プリンシパルのリストを管理する。ユーザーは kinit プログラムを実行して KDC とコンタクトをとり、**初期チケット**を取得する。チケット認可サービスはサービス・プリンシパルのリストを管理する。チケット認可サービスなどを提供するサーバーにユーザーが自己認証するとき、チケット認可サービスとコンタクトをとる。

KDC/TGS は、安全なホスト上で実行される必要のある信頼できるサード・パーティで、チケット認可チケットおよびサービス・チケットを作成する。通常、KDC と TGS は同じエンティティである。

Kerberos

分散環境のセキュリティ強化を図るためにマサチューセッツ工科大学での Athena プロジェクトで開発されたネットワーク認証サービス。Kerberos は共有シークレットに依存し、サード・パーティの安全性を前提とした信頼度の高いサード・パーティ認証システムである。

Kerberos には、シングル・サインオン機能とデータベース・リンク認証機能（MIT Kerberos のみ）があり、パスワードを集中的に保管できるため、PC のセキュリティを向上できる。

kinstance

サービスのインスタンス化または位置。kinstance として任意の文字列を指定できるが、通常はサービスのホスト・マシン名を指定する。

kservice

Kerberos サービス・オブジェクトを表す任意の名前。

LDAP

「[Lightweight Directory Access Protocol \(LDAP\)](#)」を参照。

Lightweight Directory Access Protocol (LDAP)

標準的で拡張可能なディレクトリ・アクセス・プロトコル。LDAP クライアントとサーバーが通信で使用する共通言語。Oracle Internet Directory などの業界標準のディレクトリ製品をサポートしている設計規則に関するフレームワークである。

listener.ora ファイル (listener.ora file)

次のものを識別するリスナーの構成ファイル。

- リスナー名
- 接続要求を受け入れるプロトコル・アドレス
- リスニングの対象となるサービス

通常、listener.ora ファイルは、UNIX プラットフォームでは \$ORACLE_HOME/network/admin に、Windows NT では %ORACLE_HOME%\network\admin にある。

MD5

指定されたデータから 128 ビット暗号化メッセージ・ダイジェスト値を生成して、データの整合性を保証するアルゴリズム。データ内のわずか 1 ビットが変更された場合でも、データの MD5 チェックサムが変更される。元のデータと同じ結果を MD5 で生成するようにデータを偽造することはほぼ不可能である。

NIST

「[米国連邦情報処理標準 \(Federal Information Processing Standard: FIPS\)](#)」を参照。

Oracle Net Services

Oracle サーバーまたは Designer/2000 などの Oracle のツール製品を稼働する複数のコンピュータが、サード・パーティ・ネットワークを通じてデータを交換できるようにする Oracle 製品。Oracle Net Services は分散処理と分散データベース機能をサポートする。Oracle Net Services は、通信プロトコルに依存しないオープン・システムで、ユーザーは多くのネットワーク環境に対するインタフェースとして Oracle Net を使用できる。

Oracle PKI 証明書使用方法 (Oracle PKI certificate usages)

証明書でサポートされる Oracle アプリケーション・タイプを定義する。

Oracle コンテキスト (Oracle Context)

LDAP 準拠のインターネット・ディレクトリ内にある cn=OracleContext というエントリ。このディレクトリには、**Oracle Net Services** ディレクトリ・ネーミングおよび**エンタープライズ・ユーザー**のセキュリティのエントリなど、Oracle ソフトウェア関連のすべての情報が格納されている。最上位レベルのディレクトリ・エントリであり、ディレクトリを使用するインストール済の Oracle 製品で使用されるデータが含まれている。

1 つのディレクトリに複数の Oracle コンテキストを設定できる。Oracle コンテキストは、**管理コンテキスト**の下に配置される。

Oracle データベース・メソッド (Oracle database method)

Indextix Biometric 認証の構成時に、指紋テンプレートの保管に Oracle データベースを使用すること。このメソッドのかわりに**ファイル・システム・メソッド**を使用することもできる。

PKCS #12

公開鍵暗号規格 (PKCS)。RSA Data Security, Inc. の PKCS #12 は、個人的な認証資格証明を、通常 **Wallet** と呼ばれる形式で保管および転送するための業界標準である。

PKI

「**公開鍵インフラストラクチャ (public key infrastructure: PKI)**」を参照。

Secure Hash Algorithm (SHA)

指定されたデータから 160 ビット暗号化メッセージ・ダイジェスト値を生成して、データの整合性を保証するアルゴリズム。データ内のわずか 1 ビットが変更された場合でも、データの Secure Hash Algorithm チェックサムが変更される。元のデータと同じ結果を Secure Hash Algorithm で生成するようにデータ・セットを偽造することはほぼ不可能である。

264 ビット長未満のメッセージを扱い、160 ビット・メッセージ・ダイジェストを作成するアルゴリズムである。SHA は MD5 に比べて少し遅くなるが、長いメッセージ・ダイジェストを作成できるので、強引な衝突や反転攻撃をさらに効果的に防御できる。

Secure Sockets Layer (SSL)

ネットワーク接続を保護するために Netscape Communications Corporation が開発した業界標準プロトコル。SSL では公開鍵インフラストラクチャ (PKI) を使用して、認証、暗号化およびデータの整合性を実現している。

SHA

「**Secure Hash Algorithm (SHA)**」を参照。

Sniffer

ネットワークからプライベート・データ通信を不正に傍受または取得するために使用されるデバイス。

sqlnet.ora ファイル (sqlnet.ora file)

次のものを指定するクライアントまたはサーバーの構成ファイル。

- 修飾されていないサービス名またはネット・サービス名に付加されるクライアント・ドメイン
- 名前を解決するときにクライアントが使用する必要があるネーミング・メソッドの順序
- 使用するロギングおよびトレース機能
- 接続のルート
- デフォルトの Oracle Names Server
- 外部ネーミング・パラメータ
- Oracle Advanced Security パラメータ

通常、sqlnet.ora ファイルは、UNIX プラットフォームでは
\$ORACLE_HOME/network/admin に、Windows プラットフォームでは
%ORACLE_HOME%\network\admin にある。

SSO

「[シングル・サインオン \(single sign-on: SSO\)](#)」を参照。

System Global Area (SGA)

Oracle [インスタンス](#)のデータと制御情報が格納される共有メモリー構造のグループ。

tnsnames.ora

接続記述子を含むファイル。各 [接続記述子](#) は [ネット・サービス名](#) にマップされる。このファイルは、すべてまたは個々のクライアントが使用するために、中央またはローカルで保持できる。このファイルは通常、使用しているプラットフォームに従って次の位置に常駐する。

- (UNIX の場合) \$ORACLE_HOME/network/admin
- (Windows の場合) %ORACLE_HOME%\network\admin

Wallet

個々のエンティティのセキュリティ資格証明を格納したり、管理するために使用されるデータ構造。各種暗号化サービスで使用する資格証明の保存と取得を行う。[Wallet Resource Locator](#) (WRL) は、Wallet の位置を特定するために必要な情報をすべて提供する。

Wallet Resource Locator

Wallet Resource Locator (WRL) は、**Wallet** の位置を特定するために必要な情報をすべて提供する。これは、Wallet を含んだオペレーティング・システムのディレクトリへのパスである。

Wallet 不明瞭化 (wallet obfuscation)

Wallet **不明瞭化** を使用すると、ユーザーが Wallet にアクセスする際にパスワードを入力しなくても、Oracle **Wallet** を格納し、Wallet にアクセスできる (**シングル・サインオン** のサポート)。Wallet の暗号化に使用される秘密鍵を生成するために、マシン固有の特定の情報が使用される。

WRL

「**Wallet Resource Locator**」を参照。

X.509

デジタル**証明書**の業界標準仕様。

アクセス制御 (access control)

特定のクライアントまたはクライアントのグループに対して、特定データへのアクセス権限を付与または制限するシステムの機能。

アクセス制御リスト (Access Control List : ACL)

ユーザーが定義するアクセス・ディレクティブのグループ。このディレクティブは、特定のクライアントまたはクライアントのグループ（あるいはその両方）の特定のデータに対するアクセス・レベルの権限を付与する。

暗号化 (cryptography)

データのコード化とデコード化を行い、保護メッセージを生成する作業。

暗号化 (encryption)

メッセージを、宛先の受信者以外の第三者が読むことのできない書式に変換するプロセス。

暗号化テキスト (encrypted text)

暗号化アルゴリズムを使用して暗号化されたテキスト。暗号化プロセスの出力ストリーム。最初に**復号化**しないかぎり、そのままでは読取りまたは解読できない。**暗号文**とも呼ばれる。暗号化テキストは、最終的には**平文**になる。

暗号ブロック連鎖 (Cipher Block Chaining: CBC)

暗号化方式の 1 つ。先行するすべてのブロックに従って暗号ブロックの暗号化を行い、ブロック再生攻撃からデータを保護する。許可されていない復号化が段階的に困難になるように設計されている。Oracle Advanced Security では、外部暗号ブロック連鎖 (CBC) が使用

されている。これは、内部暗号ブロック連鎖（CBC）より安全性が高く、実質的なパフォーマンスの低下を伴わないためである。

暗号文 (ciphertext)

暗号化されたメッセージ・テキスト。

インスタンス (instance)

実行中のすべての Oracle データベースは、Oracle インスタンスに関連付けられている。データベースがデータベース・サーバーで起動されると（コンピュータの種類に関係なく）、Oracle は **System Global Area (SGA)** と呼ばれるメモリ領域を割り当て、Oracle プロセスを開始する。この SGA と Oracle プロセスの組合せをインスタンスと呼ぶ。インスタンスのメモリとプロセスは、関連するデータベースのデータを効率的に管理し、データベースの 1 人以上のユーザーのために機能する。

エンタープライズ・ドメイン (enterprise domain)

データベースと **エンタープライズ・ロール** のグループからなるディレクトリ構造。1 つのデータベースが同時に複数のエンタープライズ・ドメイン内に存在することはない。エンタープライズ・ドメインは Windows 2000 ドメインとは異なり、共通のディレクトリ・データベースを共有するコンピュータの集合である。

エンタープライズ・ドメイン管理者 (Enterprise Domain Administrator)

特定の **エンタープライズ・ドメイン** を管理する権限があるユーザー。新しいエンタープライズ・ドメイン管理者を追加する権限もある。

エンタープライズ・ユーザー (enterprise user)

ディレクトリで定義され管理されるユーザー。各エンタープライズ・ユーザーは企業内で固有の識別情報を持ち、ログイン資格証明の格納に **Wallet** を使用する。

エンタープライズ・ロール (enterprise role)

エンタープライズ・ユーザー に割り当てられるアクセス権限。 **エンタープライズ・ドメイン** 内の 1 つ以上のデータベースに関係する Oracle ロール・ベースの **認可** のセット。エンタープライズ・ロールはディレクトリに格納され、1 つ以上の **グローバル・ロール** が含まれる。

エントリ (entry)

ディレクトリのビルディング・ブロックで、ディレクトリのユーザーにとって関心のあるオブジェクトに関する情報が含まれている。

オブジェクト・クラス (object class)

名前を持った **属性** のグループ。属性をエントリに割り当てるには、その属性を保持しているオブジェクト・クラスをそのエントリに割り当てる。同じオブジェクト・クラスに関連するオブジェクトはすべて、同じ属性を共有する。

介在者 (man-in-the-middle)

第三者によるメッセージの不正傍受などのセキュリティ攻撃。第三者、つまり介在者は、メッセージを復号化して再暗号化し（元のメッセージを変更する場合と変更しない場合がある）、元のメッセージの宛先である受信者に転送する。これらの処理はすべて、正当な送受信者が気付かないうちに行われる。この種のセキュリティ攻撃は、**認証**が行われていない場合にのみ発生する。

外部認証 (external authentication)

Kerberos や RADIUS などのサード・パーティの認証サービスによるユーザー識別情報の検証。

鍵 (key)

データの暗号化時に、指定したアルゴリズムによって指定した平文から生成される暗号文を決定する値。また、データの復号化時に、暗号文を正しく復号化するために必要な値。暗号文は、正しい鍵が提供された場合にのみ正しく復号化される。

対称型暗号化アルゴリズムでは、同一データの暗号化と復号化の両方に同じ鍵が使用される。非対称型暗号化アルゴリズム（公開鍵暗号アルゴリズムまたは公開鍵暗号方式とも呼ばれる）では、同一データの暗号化と復号化に異なる鍵が使用される。

鍵のペア (key pair)

公開鍵とそれに対応する**秘密鍵**のペア。

「**公開鍵と秘密鍵のペア (public/private key pair)**」を参照。

管理コンテキスト (administrative context)

Oracle コンテキストがその下に常駐するディレクトリ・エントリ。管理コンテキストは、**ディレクトリ・ネーミング・コンテキスト**の場合がある。ディレクトリ・アクセス構成時に、クライアントはディレクトリ構成ファイル (ldap.ora) 内の管理コンテキストによって構成される。管理コンテキストは、そのエントリへのアクセスをクライアントが予測するディレクトリ内の Oracle コンテキストの位置を指定する。

機密保護 (confidentiality)

暗号化の機能。機密保護によって、メッセージの本来の受信者のみがメッセージを見る（暗号文を復号化する）ことができる。

共有スキーマ (shared schema)

複数のエンタープライズ・ユーザーが使用できるデータベースまたはアプリケーション・スキーマ。Oracle Advanced Security では、複数のエンタープライズ・ユーザーを1つのデータベース上の同一共有スキーマにマップできる。この機能によって、管理者はそれぞれのデータベースで各ユーザーごとにアカウントを作成する必要がない。かわりに、ユーザーを1つの場所、つまり、エンタープライズ・ディレクトリに作成して、そのユーザーを共有スキーマにマップすることができる。この共有スキーマには他のエンタープライズ・ユーザーもマップできる。**ユーザー / スキーマの分割**と呼ばれることもある。

クライアント (client)

サービスを利用する側。クライアントはユーザーであったり、データベース・リンク中にユーザーとして機能するプロセス（プロキシともいう）であったりする。

クリアテキスト (cleartext)

暗号化されていないメッセージ・テキスト。

グローバル・ディレクトリ・サービス (Global Directory Service: GDS)

DCE CDS と X.500 ディレクトリ・サービス間のエージェントとして動作する **DCE** ディレクトリ・サービス。GDS と **CDS** はともに古く、現在は **DCE** のみが使用される。

グローバル・ロール (global role)

ディレクトリで管理されるが、その権限は 1 つのデータベースに格納されているロール。

公開鍵 (public key)

公開鍵暗号における一般に公開される鍵。主に暗号化に使用されるが、署名の確認にも使用できる。「**公開鍵と秘密鍵のペア (public/private key pair)**」を参照。

公開鍵暗号 (public key encryption)

送信側でメッセージを受信側の公開鍵で暗号化するプロセス。配信されたメッセージは、受信側の秘密鍵で復号化される。

公開鍵インフラストラクチャ (public key infrastructure: PKI)

公開鍵暗号の原理を利用した情報セキュリティ技術。公開鍵暗号では、共有されている公開鍵と秘密鍵のペアを使用して情報を暗号化および復号化する。パブリック・ネットワークにおける安全でプライベートな通信を提供する。

公開鍵と秘密鍵のペア (public/private key pair)

暗号化と**復号化**に使用される 2 つの数字のセット。1 つは**秘密鍵**、もう 1 つは**公開鍵**と呼ばれる。公開鍵は通常広く使用可能であるのに対して、秘密鍵はその各所有者によって保有される。2 つの数字は関連付けられているが、公開鍵から秘密鍵を導出することは一般的にほぼ不可能である。公開鍵と秘密鍵は、非対称型暗号化アルゴリズム（公開鍵暗号アルゴリズムまたは公開鍵暗号方式とも呼ばれる）でのみ使用される。**鍵のペア**の公開鍵または秘密鍵のいずれかで暗号化されたデータは、鍵のペアの関連する鍵で復号化できる。ただし、公開鍵で暗号化されたデータを同じ公開鍵では復号化できず、秘密鍵で暗号化されたデータを同じ秘密鍵では復号化できない。

サーバー (server)

サービスの提供側。

サービス (service)

1. クライアントが使用するネットワーク資源 (Oracle データベース・サーバーなど)。
2. Windows NT の [レジストリ](#) にインストールされ、Windows NT で管理される実行可能プロセス。サービスが作成されて開始されると、ユーザーがコンピュータにログオンしていない場合でも実行できる。

サービス・チケット (service ticket)

クライアントを認証する際に使用する信頼度の高い情報。初期チケットとも呼ばれるチケット認可チケットを取得するには、okinit プログラムを直接または間接的に実行し、パスワードを入力する。チケット認可チケットは、クライアントがサービス・チケットを要求するときに使用される。サービス・チケットは、クライアントがサービスへの認証を受けるときに使用される。

サービス表 (service table)

Kerberos 認証では、サービス表は、*kinstance* 上に存在するサービス・プリンシパルのリスト。Oracle で Kerberos を使用する前に、サービス表を Kerberos から抽出して Oracle サーバー・マシンにコピーする必要がある。

サービス・プリンシパル (service principal)

「[プリンシパル \(principal\)](#)」を参照。

サービス名 (service name)

Kerberos ベースの認証で使用するサービス・プリンシパルの **kservice** 部分。

資格証明 (credentials)

データベースへのアクセスの取得に使用する [ユーザー名](#)、パスワードまたは証明書。

識別情報 (identity)

エンティティの公開鍵と他の公開情報の組合せ。公開情報には、電子メールのアドレスなど、ユーザー識別データが含まれる。宣言通りのエンティティとして証明されているユーザー。

識別名 (distinguished name: DN)

ディレクトリ・エントリの一意名。親エントリの個々の名前がすべて、下からルート方向へ順に結合されて構成されている。

辞書攻撃 (dictionary attack)

パスワードに対する一般的な攻撃。攻撃者は、考えられる多数のパスワードとそれに対応するベリファイアの辞書を作成する。なんらかの手段で、攻撃者は目的のパスワードに対応するベリファイアを取得し、辞書でベリファイアを探し、目的のパスワードを取得する。

システム識別子 (system identifier: SID)

Oracle **インスタンス**の一意名。Oracle データベース間の切替えでは、ユーザーによる SID の指定が必要である。SID は、**tnsnames.ora** ファイルにある**接続記述子**の CONNECT DATA 部分、および **listener.ora** ファイルにある**ネットワーク・リスナー**の定義部分に記述されている。

証明書 (certificate)

公開鍵に対して識別情報を安全にバインドする ITU x.509 v3 の標準データ構造。

証明書は、エンティティの公開鍵が、信頼されている機関（認証局）によって署名されたときに有効となる。この証明書は、そのエンティティの情報が正しいこと、および公開鍵がそのエンティティに実際に属していることを保証する。

証明書にはエンティティの名前、認証情報および公開鍵が含まれる。また、証明書に関連する権利、ユーザーおよび権限についてのシリアル番号、有効期限、その他の情報が含まれる場合もある。さらに、発行元の認証局についての情報も含まれる。

証明連鎖 (certificate chain)

エンドユーザーまたはサブスクライバの証明書と、その認証局の証明書を含む指定順の証明書リスト。

初期チケット (initial ticket)

Kerberos 認証では、初期チケットまたはチケット認可チケット (TGT) によって、その他のサービス・チケットの要求権利を持つユーザーであることが証明される。初期チケットがなければ、他のチケットは取得できない。初期チケットは、kinit プログラムを実行し、パスワードを入力することで取得できる。

シングル・サインオン (single sign-on: SSO)

ユーザーの認証を 1 回のみ行う機能。以降の他のデータベースまたはアプリケーションへの接続において透過的に発生する厳密な認証と組み合わせて認証を行う。シングル・サインオンを使用すると、ユーザーは最初の接続時に入力した単一のパスワードで複数のアカウントとアプリケーションにアクセスできる。単一パスワードによる単一の認証。Oracle Advanced Security は、Kerberos、CyberSafe、DCE および SSL ベースのシングル・サインオンをサポートしている。

信頼できる証明書 (trusted certificate)

一定の信頼度を有すると認定されたサード・パーティの識別情報。（ルート鍵証明書とも呼ばれる）。信頼できる証明書は、エンティティが本人であるという識別情報の確認が行われるときに使用される。通常、信頼できる認証局を信頼できる認証という。いくつかのレベルの信頼できる証明書がある場合、証明連鎖の中でレベルの低い、信頼できる証明書は、それよりもレベルの高い証明書で再確認する必要はない。

信頼できる認証局 (trusted certificate authority)

「**認証局 (certificate authority)**」を参照。

スキーマ (schema)

表、**ビュー**、クラスタ、プロシージャ、パッケージ、**属性**、**オブジェクト・クラス**およびそれらに対応する一致規則など、名前を持つオブジェクトの集合体で、特定のユーザーに関連付けられる。

スキーマ・マッピング (schema mapping)

ユーザーが存在する **LDAP** 準拠のディレクトリ内の**ベース**と、そのユーザーがマップされるデータベース・スキーマの名前を示すデータベース内の値のペア。

スマートカード (smart card)

ユーザー名やパスワードなどの情報を格納するため、また認証交換に関連する計算を実行するための IC が組み込まれた（クレジット・カードに似た）プラスチック製のカード。スマートカードはクライアントまたはサーバーにあるハードウェア・デバイスで読み取る。

スマートカードは、ワンタイム・パスワードとして使用することができる乱数を生成できる。この場合、スマートカードは、サーバー上のサービスと同期化されているので、サーバーはスマートカードによって生成されるパスワードと同じパスワードを要求する。

整合性 (integrity)

受信したメッセージの内容が、送信前のメッセージ内容と変更されていないことの保証。

セッション鍵 (session key)

少なくとも二者間（通常はクライアントとサーバー）で共有され、単一の通信セッション継続中のデータ暗号化に使用される鍵。セッション鍵は通常、ネットワーク通信を暗号化するのに使用される。クライアントとサーバーは、セッションの開始時に使用するセッション鍵を取り決めることができる。セッション継続中は、関係者間の全ネットワーク通信の暗号化にその鍵が使用される。クライアントとサーバーが新しいセッションで再び通信する場合は、新しいセッション鍵を取り決める。

セッション・レイヤー (session layer)

プレゼンテーション・レイヤーのエンティティが必要とするサービスを提供するネットワーク・レイヤー。エンティティでは、対話を構成および同期化することができ、データ交換の管理が有効となる。このレイヤーは、クライアントとサーバー間でネットワーク・セッションを確立、管理および終了する。セッション・レイヤーの例には、ネットワーク・セッションがある。

接続記述子 (connect descriptor)

ネットワーク接続の宛先を示す、特別にフォーマットされた表記。接続記述子には、接続先**サービス**とネットワーク・ルーティング情報が含まれる。宛先サービスは、Oracle9i データベースまたは Oracle8i データベースのサービス名または Oracle8 リリース 8.0 データベースの Oracle **システム識別子**を使用して識別される。ネットワーク・ルーティングは、ネットワーク・アドレスを使用して、少なくとも **リスナー**の位置を提供する。

接続先識別情報 (peer identity)

SSL 接続では、特定のクライアントと特定のサーバーの間にセッションが確立される。接続先の識別情報は、セッションのセットアップ・プロセスの一部として設定される場合がある。接続先は、[X.509 証明連鎖](#)によって識別される。

接続文字列 (connect string)

[ユーザー名](#)、パスワードおよび[ネット・サービス名](#)など、接続するためにユーザーが[サービス](#)に渡す情報。例：

```
CONNECT username/password@net_service_name
```

属性 (attribute)

エントリの性質を説明する断片的な情報項目。1つのエントリは1組の属性から構成され、それぞれが[オブジェクト・クラス](#)に所属する。さらに、各属性は型と値を持つ。型は属性の情報の種類を示し、値には実際のデータが格納される。

単一鍵ペア Wallet (single key-pair wallet)

単一のユーザー[証明書](#)とその関連する[秘密鍵](#)が含まれる [PKCS #12](#) 形式の [Wallet](#)。[公開鍵](#)は証明書に埋め込まれている。

単一パスワード認証 (single password authentication)

ユーザーが単一のパスワードを使用して自己を複数のデータベースに対して認証する機能。ユーザーは1つのデータベースまたはアプリケーションに対して単一のパスワードで自己を認証した後、同じパスワードで他のデータベースやアプリケーションに対しても認証できる。[Oracle Advanced Security](#) 実装では、パスワードはLDAP 準拠のディレクトリに格納され、暗号化と ACL によって保護される。単一パスワード認証を使用すると、ユーザーは単一のパスワードを使用して複数回（複数のデータベースとアプリケーションに対して）自己を認証できる。単一パスワードによる複数の認証。

チェックサム (checksumming)

メッセージ・パケットに含まれているデータに基づいてメッセージ・パケットの値を計算し、その値をデータとともに渡して、データが改ざんされていないことを証明するメカニズム。データ受信側は暗号チェックサムを再計算して、それをデータとともに送られた暗号チェックサムと比較する。これらの暗号チェックサムが一致している場合は、データが転送中に改ざんされなかったことを「高い確率で」証明できる。

チケット (ticket)

所有者を識別するのに役立つ情報。「[サービス・チケット \(service ticket\)](#)」を参照。

データ暗号化規格 (Data Encryption Standard: DES)

米国データ暗号化規格。

データ・ディクショナリ (data dictionary)

データベースに関する情報を提供する一連の読取り専用の表。

データベース・インストール管理者 (Database Installation Administrator)

データベース作成者とも呼ばれる。新しいデータベースの作成を管理する。データベースの管理には、Database Configuration Assistant を使用して、ディレクトリに各データベースを登録する作業が含まれる。この管理者は、データベース・サービス・オブジェクトと属性に対する作成と変更のアクセス権限を所有する。また、デフォルト・**ドメイン**を変更することもできる。

データベース管理者 (Database Administrator)

- (1) Oracle サーバーまたはデータベース・アプリケーションを操作および管理する人。
- (2) DBA 権限を所有し、データベース管理操作を実行できる Oracle ユーザー名。通常、これら 2 つを同時に意味する。多くのサイトでは複数の DBA が配置される。

データベース・セキュリティ管理者 (Database Security Administrator)

エンタープライズ・ユーザーのセキュリティのために、アクセスの作成、変更および読込みを行う管理者。この管理者は企業内のすべてのドメインに対する権限を持ち、次のような責任を担っている。

- Oracle DBSecurityAdmins および OracleDBCreators グループの管理
- 新規のエンタープライズ・ドメインの作成
- 企業内のデータベースの **ドメイン**間での移動

データベース・パスワード・ベリファイア (database password verifier)

ユーザーのデータベース・パスワードから導出される不可逆的な値。この値は、接続するユーザーの識別情報を証明するために、データベースに対するパスワード認証時に使用される。

データベース別名 (database alias)

「**ネット・サービス名 (net service name)**」を参照。

データベース・メソッド (database method)

「**Oracle データベース・メソッド (Oracle database method)**」を参照。

データベース・リンク (database link)

リモート・データベース、およびそのデータベースに至る通信パス、ユーザー名とパスワードなどを識別するために、ローカル・データベースまたはネットワーク定義に格納されているネットワーク・オブジェクト。一度定義すると、リモート・データベースへのアクセスにはデータベース・リンクが使用される。

あるデータベース・リンクから別のデータベースへのパブリックまたはプライベート・データベース・リンクは、DBA またはユーザーによってローカル・データベース上に作成される。

グローバル・データベース・リンクは、Oracle Names でネットワーク内の各データベースから他のすべてのデータベースに自動的に作成される。グローバル・データベース・リンクはネットワーク定義に格納される。

ディレクトリ情報ツリー (directory information tree: DIT)

エントリの識別名 (Distinguished Name: DN) で構成されるツリー形式の階層構造。

ディレクトリ・ネーミング・コンテキスト (directory naming context)

ディレクトリ・サーバー内で重要なサブツリー。通常、ディレクトリ・ネーミング・コンテキストは、組織サブツリーの最上部となっている。一部のディレクトリでは、固定のコンテキストのみが可能である。また、別のディレクトリでは、ディレクトリ管理者による構成の対象をゼロから多数までとすることができる。

デジタル署名 (digital signature)

デジタル署名は、送信者の秘密鍵によって送信者のメッセージを署名するのに公開鍵アルゴリズムが使用されているときに作成される。デジタル署名によって、文書が信頼できるものであること、別のエンティティで偽造されていないこと、変更されていないこと、送信者によって拒否されないことが保証される。

トークン・カード (token card)

ユーザーが容易に認証サービスを利用できるように、数種類のメカニズムを提供するデバイス。一部のトークン・カードは、認証サービスと同期化されているワンタイム・パスワードを提供する。サーバーは認証サービスとやりとりすることによって、トークン・カードが提供するパスワードをいつでも検証できる。要求 / 応答に基づいて機能するトークン・カードもある。この場合は、サーバーが要求 (番号) を提供し、ユーザーがその番号をトークン・カードに入力する。そして、トークン・カードは別の番号 (最初の番号から暗号的に導出される番号) を提供し、それをユーザーがサーバーに渡す。

ドメイン (domain)

ドメイン・ネーム・システムは、ネームスペース内の任意のツリーまたはサブツリー。ドメインは通常、所属するホストの名前が共通の接尾辞、つまりドメイン名を共有しているコンピュータのグループを指す。

ドメイン・ネーム・システム (Domain Name System: DNS)

コンピュータやネットワーク・サービスのネーミング・システムであり、**ドメイン**を階層的に編成している。DNS は、わかりやすい名前でもコンピュータの位置を識別するために TCP/IP ネットワークで使用される。DNS は、ユーザー・フレンドリな名前を、コンピュータが理解できる IP アドレスに変換する。

Oracle Net Services の場合、DNS は、TCP/IP アドレス内のホスト名を IP アドレスに変換する。

トラスト・ポイント (trust point)

「**信頼できる証明書 (trusted certificate)**」を参照。

トランスポート・レイヤー (transport layer)

データ・フローの制御とエラーのリカバリ・メソッドによってエンド間の信頼性を維持するネットワーク・レイヤー。**Oracle Net Services** では、Oracle プロトコル・サポートをトランスポート・レイヤーに使用する。

認可 (authorization)

ユーザー、プログラムまたはプロセスに、オブジェクトへのアクセスを許可すること。**Oracle** では、ロール・メカニズムに基づいて認可が行われる。1 人のユーザーまたはユーザー・グループに、1 つのロールまたは一連のロールを付与できる。また、ロールに他のロールを付与することもできる。認証されたエンティティに対して利用できる権限のセット。

認証 (authentication)

コンピュータ・システム内のユーザー、デバイスまたはその他のエンティティの識別情報を検証するプロセス。多くの場合、システム内のリソースへのアクセスを許可する前提条件として使用される。認証されたメッセージの受信者は、そのメッセージの発信元（送信側）を確認できる。認証では、第三者が送信者になりすます可能性はないと仮定されている。

認証局 (certificate authority)

ユーザー、データベース、管理者、クライアント、サーバーなどの他のエンティティが本当に本人であるかどうかを証明する信頼できるサード・パーティ。ユーザーを証明するとき、認証局では、最初にユーザーが証明書失効リスト (CRL) にないことを確認してからユーザーの識別情報を検証し、認証局の秘密鍵で署名して証明書を付与する。認証局には、認証局が発行する認証局独自の証明書と公開鍵がある。サーバーとクライアントではこれらを使用して、認証局が作成した署名を検証する。認証局は証明書サービスを行う外部の会社であったり、企業の MIS 部門などの内部組織である場合がある。

ネット・サービス名 (net service name)

データベース・サーバーを識別するためにクライアントが使用する名前。ネット・サービス名は、ポート番号とプロトコルにマップされる。**接続文字列**または**データベース別名**とも呼ばれる。

ネットワーク認証サービス (network authentication service)

分散環境で、クライアントをサーバーに対して、サーバーをサーバーに対して、またはユーザーをクライアントとサーバーに対して認証する方法。ネットワーク認証サービスは、ユーザーに関する情報、ユーザーがアクセスする様々なサーバー上のサービスに関する情報、およびネットワーク上のクライアントとサーバーに関する情報を格納するためのリポジトリである。認証サーバーは物理的に異なるマシンであったり、システム内で別のサーバー上に置かれる機能であったりする。可用性を向上させるために、認証サービスを複製して 1 点障害を回避できる場合がある。

ネットワーク・リスナー (network listener)

1 つ以上のプロトコルで 1 つ以上のデータベースの接続要求をリスニングするサーバー上のリスナー。「[リスナー \(listener\)](#)」を参照。

パスワード・アクセシブル・ドメイン・リスト (Password-Accessible Domains List)

パスワード認証ユーザーからの接続を受け入れるように構成された[エンタープライズ・ドメイン](#)のグループ。

否認防止 (non-repudiation)

メッセージの出所、送達、提出または送信に関する自明の証明。

秘密鍵 (private key)

公開鍵暗号における秘密鍵。主に復号化に使用されるが、デジタル署名とともに暗号化にも使用される。「[公開鍵と秘密鍵のペア \(public/private key pair\)](#)」を参照。

ビュー (views)

1 つ以上の表（または他のビュー）の選択的提示。表の構造とデータの両方を表示する。

ファイル・システム・メソッド (file system method)

Identix Biometric 認証の構成時に、指紋テンプレートをファイルに保管すること。このメソッドのかわりに [Oracle データベース・メソッド](#)を使用することもできる。

フォレスト (forest)

相互に信頼する 1 つ以上の Active Directory ツリーのグループ。フォレスト内のすべてのツリーは、共通の[スキーマ](#)、構成およびグローバル・カタログを共有する。フォレストに複数のツリーがある場合、それらのツリーは連続するネームスペースを形成しない。指定フォレスト内のすべてのツリーは、推移的な双方向の信頼関係を介して相互に信頼する。

復号化 (decryption)

暗号化されたメッセージの内容（暗号文）を、元の読取り可能な書式（平文）に戻す変換プロセス。

不明瞭化 (obfuscation)

情報を判読不可能な形式にスクランブルするプロセス。スクランブルに使用したアルゴリズムが不明な場合、スクランブルを解除することは非常に困難である。

不明瞭化プログラム (obfuscator)

Java ソース・コードの不明瞭化に使用される特別なプログラム。「[不明瞭化 \(obfuscation\)](#)」を参照。

プリンシパル (principal)

一意に識別されるクライアントまたはサーバー。 *kservice/kinstance@REALM* からなる Kerberos オブジェクト。「kservice」、「kinstance」および「レルム (realm)」も参照。

プロキシ認証 (proxy authentication)

ファイアウォールなどの中間層を含む環境で一般に使用されるプロセス。エンド・ユーザーは中間層に対して認証を行い、中間層はエンド・ユーザーのプロキシとして、ユーザーのかわりにディレクトリに対して認証を行う。中間層は、プロキシ・ユーザーとしてディレクトリにログインする。プロキシ・ユーザーは ID を切り替えることができ、一度ディレクトリにログインすると、エンド・ユーザーの ID に切り替わる。次に、その特定のエンド・ユーザーに付与されている認可を使用して、エンド・ユーザーのかわりに操作を実行する。

分散コンピューティング環境 (Distributed Computing Environment: DCE)

分散環境を提供するために複数のシステムで動作する、統合化された一連のネットワーク・サービス。分散アプリケーションとオペレーティング・システムまたはネットワーク・サービスとの間にあるミドルウェア。クライアント / サーバー・コンピューティング・モデルに基づいている。DCE はオープン・グループによってサポートされる。

ベース (base)

LDAP 準拠のディレクトリ内でのエントリ・ポイント。

米国商務省国立標準技術研究所 (National Institute of Standards and Technology: NIST)

コンピュータおよびテレコミュニケーション・システム内の暗号ベース・セキュリティの設計、取得および実装に関連するセキュリティ標準の開発を担う米国商務省内の機関。米国連邦機関、米国連邦機関の請負業者または連邦の機能を果たすために米国連邦政府にかかわって情報を処理する他の組織によって運営されている。

米国連邦情報処理標準 (Federal Information Processing Standard: FIPS)

暗号化モジュールのセキュリティ要件を定義する米国連邦政府の標準。コンピュータおよびテレコミュニケーション・システム内の非機密情報を保護するセキュリティ・システムで使用される。米国商務省国立標準技術研究所 (NIST) が発行する。

平文 (plaintext)

暗号化されていないプレーン・テキスト。

メッセージ・ダイジェスト (message digest)

「**チェックサム (checksumming)**」を参照。

メッセージ認証コード (message authentication code)

データ認証コード (DAC) ともいう。秘密鍵を追加した**チェックサム**。鍵を持つ人のみが暗号化チェックサムを検証できる。

ユーザー / スキーマの分割 (user/schema separation)

「[共有スキーマ \(shared schema\)](#)」を参照。

ユーザー名 (username)

データベース内のオブジェクトに接続してアクセスできる名前。

リスナー (listener)

サーバー上で実行される独立したプロセス。クライアントの着信接続要求をリスニングし、サーバーへの通信量を管理する。

クライアントがサーバーとのネットワーク・セッションを要求するたびに、リスナーは実際の要求を受信する。クライアントの情報がリスナーの情報と一致すると、リスナーはサーバーへの接続を認める。

リモート・コンピュータ (remote computer)

ローカル・コンピュータ以外のネットワーク上のコンピュータ。

ルート鍵証明書 (root key certificate)

「[信頼できる証明書 \(trusted certificate\)](#)」を参照。

レジストリ (registry)

コンピュータの構成情報を格納する Windows リポジトリ。

レルム (realm)

Kerberos オブジェクト。1 つの Key Distribution Center/Ticket Granting Service (KDC/TGS) の下で稼働するクライアントとサーバーのセット。名前が同じでも異なるレルムにある *kservices* は一意である。

索引

A

Active Directory

- Microsoft のツールとの統合, E-6
- Microsoft のツールによる接続性テスト, E-18
- Oracle Net ディレクトリ・ネーミングの作成要件, E-13
- Oracle ディレクトリ・オブジェクトの表示方法, E-8
- Oracle の使用要件, E-17
- SQL*Plus を使用した接続性のテスト, E-6
- Windows ログイン資格証明との統合, E-8
- アクセス制御リストの管理, E-21
- エンタープライズ・ユーザー・セキュリティの作成要件, E-14
- クライアント・コンピュータからの接続性テスト, E-17
- セキュリティ・グループ・ディレクトリ・サーバーの管理
 - アクセス制御リストの管理, E-21
- セキュリティ・グループへのアクセス, E-22
- セキュリティ・ドメインの作成, E-26
- 定義, E-2
- ディレクトリ・オブジェクト型記述の拡張機能, E-7
- ディレクトリ・サーバーとの統合, E-17
- ディレクトリ・サーバーの自動検出, E-4
- データベース接続性のテスト, E-6
- データベースへの接続, E-17, E-18
- ユーザー・インタフェースの拡張機能, E-6

Active Directory ユーザーとコンピュータ

- Active Directory での Oracle オブジェクトとの統合, E-6

- ディレクトリ・サーバー・オブジェクトへのアクセス, E-18

ATTENTION_DESCRIPTION 列, 16-6

C

C:¥ORANT、定義, xxx

CASCADE_FLAG 列, 16-6, 16-7

CASCADE パラメータ, 16-8

CDS, 「Cell ディレクトリ・サービス (CDS)」を参照

CDS ネームスペース内のマッピングの表示, リスナー・エンドポイント, 13-2

Cell ディレクトリ・サービス (CDS)

cds_attributes ファイル

- CDS で名前を解決するために変更, 12-14

Oracle サービス名, 10-4

名前の検索を実行, 12-14

ネーミング・アダプタ・コンポーネント, 10-4

ネーミング・アダプタ, 10-4

Cipher Suite

Secure Sockets Layer (SSL), B-11

CyberSafe, 1-10

kinstance, 5-3

sqlnet.ora ファイルのサンプル, A-3

システム要件, 1-16

認証の構成, 5-2

認証パラメータ, B-2

レルム, 5-3

CyberSafe Challenger

システム要件, 1-16

D

Database Configuration Assistant
ディレクトリ・サーバーでのデータベース・オブジェクトの登録, E-8
DBPASSWORD_EXIST_FLAG 列, 16-6, 16-7
DBPASSWORD 列, 16-6
DCE.AUTHENTICATION パラメータ, 12-11
DCE.LOCAL_CELL_USERNAMES パラメータ, 12-11
DCE.PROTECTION パラメータ, 12-11
DCE.TNS_ADDRESS_OID パラメータ, 12-11
DCE.TNS_ADDRESS_OID パラメータ
protocol.ora ファイルでの変更, 12-14
DCE グループのマップ
Oracle ロール, 12-6
DCE, 「分散コンピューティング環境 (DCE)」を参照
DES, 「データ暗号化規格 (DES)」を参照
Diffie-Hellman 鍵折衝アルゴリズム, 2-5
DIRPASSWORD 列, 16-6
DNS スタイル・ネーミング規則, E-17

E

ELA, 「Oracle Enterprise Login Assistant (ELA)」を参照
Enterprise Login Assistant, 「Oracle Enterprise Login Assistant (ELA)」を参照
Entrust Technologies 社, 8-2
Entrust/PKI for Oracle, 8-5
Entrust/PKI ソフトウェア, 1-10, 8-1, 8-2
Authority, 8-5
Entelligence, 8-5
etbinder コマンド, 8-12
IPSEC Negotiator Toolkit, 8-6
RA, 8-5
Toolkit Server Login, 8-6
鍵管理, 8-3
構成
クライアント, 8-10
サーバー, 8-11
構成要素, 8-4
サポートしているバージョン, 8-4
証明書取消し, 8-3
データベース・ユーザーの作成, 8-13
認証, 8-7, 8-8

プロファイル, 8-8
管理者による作成, 8-8
ユーザーによる作成, 8-9
問題点と制限事項, 8-14

esm -genca ツール, 19-12, 19-20
esm -genca ツールによる認証局の作成, 19-20
ESM, 「Oracle Enterprise Security Manager (ESM)」を参照
etbinder コマンド, 8-12

F

FIPS, 「米国連邦情報処理標準 (FIPS)」を参照

G

GDS, 「グローバル・ディレクトリ・サービス (GDS)」を参照

H

HTTPS, 7-6

I

Internet Inter-ORB Protocol (IIOP)
SSL による保護, 7-6

J

Java Database Connectivity (JDBC)
Oracle Advanced Security の実装, 3-2
Oracle O3LOGON プロトコル, 3-3
Oracle 拡張機能, 3-2
Thin ドライバ機能, 3-3
構成パラメータ, 3-5
Java バイトコードの不明瞭化, 3-4
JDBC Thin のサポート, 3-1
JDBC, 「Java Database Connectivity」を参照

K

Kerberos, 1-10
kinstance, 6-2
kservice, 6-2
sqlnet.ora ファイルのサンプル, A-3
システム要件, 1-16

認証アダプタ・ユーティリティ, 6-11
認証の構成, 6-2, 6-5
レルム, 6-3
kinstance (CyberSafe), 5-3
kinstance (Kerberos), 6-2
kservice (Kerberos), 6-2

L

LAN 環境
弱点, 1-2
LDAP, 「Lightweight Directory Access Protocol (LDAP)」を参照
Lightweight Directory Access Protocol (LDAP), 1-12, 18-1, 18-2, 18-5, 18-6, 18-9, 18-10
listener.ora ファイル, 15-35
DCE 用のパラメータ, 12-4

M

MAPPING_LEVEL 列, 16-6, 16-7
MAPPING_TYPE 列, 16-6, 16-7
MD5 メッセージ・ダイジェスト・アルゴリズム, 2-4

N

NAMES.DIRECTORY_PATH パラメータ, 12-17
NEEDS_ATTENTION_FLAG 列, 16-6
Netscape Communications Corporation, 7-2

O

okdstry
Kerberos アダプタ・ユーティリティ, 6-11
okinit
Kerberos アダプタ・ユーティリティ, 6-11
oklist
Kerberos アダプタ・ユーティリティ, 6-11
OLD_SCHEMA_TYPE 列, 16-5
ORA-12650 エラー・メッセージ, A-8
Oracle Advanced Security
Java 実装, 3-2, 3-4
sqlnet.ora ファイルの暗号化のサンプル, A-2
sqlnet.ora ファイルのチェックサムのサンプル, A-2
SSL 機能, 7-2
構成パラメータ, 3-5
認証を使用禁止にする, 9-2

Oracle Advanced Security の利点, 1-4
Oracle Connection Manager, 1-15
Oracle Enterprise Login Assistant (ELA), 15-25, 18-1
LDAP ディレクトリ, 18-6
SSL 接続の使用禁止, 18-10
Wallet のアップロード, 18-10
Wallet のダウンロード, 18-6
起動, 18-2
証明書認証エンタープライズ・ユーザー, 18-3
パスワード認証ユーザー, 18-11
パスワードの変更, 18-7, 18-11
ローカル Wallet のオープン, 18-3
ログアウト, 18-10
Oracle Enterprise Manager, 19-2, 19-3
Oracle Enterprise Security
初期構成, 15-27
Oracle Enterprise Security Manager (ESM), 15-19, 19-1
esm -genca ツール, 19-12, 19-20
Oracle コンテキスト, 19-17, 19-22
Oracle コンテキスト管理者, 19-22
RDBMS_SERVER_DN パラメータ, 19-21
spfile.ora, 19-21
userpkcs12 属性, 19-13
インストール, 19-2
エンタープライズ・ドメイン管理者, 19-35
エンタープライズ・ドメインの管理, 19-31
エンタープライズ・ユーザー・パスワードの定義, 19-10
エンタープライズ・ロール, 19-38
エンタープライズ・ロール権限受領者, 19-42
概要, 15-2, 19-2
起動, 19-4
グローバル・ロール・メンバーシップ, 19-40
構成, 19-2
コマンドライン構文, 19-4
使用, E-16
セキュリティ・ドメインの作成, E-26
ディレクトリ・ベースの定義, 19-8
データベース・アクセス, 19-16
データベース管理者, 19-27
データベース・スキーマ・マッピング, 19-28
データベース・セキュリティ, 19-27
データベース・セキュリティ・オプション, 19-35
データベース・ドメインのメンバーシップ, 19-32
ドメイン・データベース・スキーマ・マッピング, 19-36

- 認証局の作成, 19-12
- パスワード・アクセスブル・ドメイン, 19-25
- ユーザー検索ベース, 19-21
- ユーザーのブラウズ, 19-13
- ロールの定義, 19-11
- Oracle Java SSL
 - Cipher Suite, F-3
 - 機能, F-3
- Oracle Net, 15-32
- Oracle Net Configuration Assistant
 - Oracle スキーマの作成, E-10
 - ディレクトリ・サーバー情報の自動検出, E-9
 - ディレクトリ・サーバーで使用する Oracle ソフトウェアの構成, E-4, E-8
- Oracle Net ディレクトリ・ネーミング
 - Microsoft のツールによる接続性テスト, E-18
 - クライアント・コンピュータからの接続性テスト, E-18
 - 作成要件, E-13
 - ディレクトリ・サーバーを介したデータベースへの接続, E-17
- Oracle Password Protocol, 3-4
- Oracle Wallet Manager, 8-2, 15-25, 15-36, 15-39
 - 鍵管理, F-4
- ORACLE_BASE
 - 説明, xxx
- ORACLE_HOME
 - 説明, xxx
- OracleDBCreators グループ, 15-11
- OracleDBCreator セキュリティ・グループ
 - 定義, E-21
- OracleDBSecurityAdmin セキュリティ・グループ
 - 定義, E-21
- OracleDBSecurity グループ, 15-11
- OracleDefaultDomain
 - ディレクトリ・サーバーのセキュリティ・ドメイン, E-26
- OracleNetAdmins セキュリティ・グループ
 - 定義, E-21
- Oracle コンテキスト, 15-51, 19-6, 19-17, 19-22, 19-27
 - 定義, E-9
- Oracle サービス名, 10-4
 - CDS へのロード, 12-16
- Oracle スキーマ
 - Oracle Net Configuration Assistant による作成, E-10

- Oracle にログイン
 - DCE 認証, 13-3
- Oracle パラメータ
 - 認証, 9-5
- ORCL_GLOBAL_USR_MIGRATION_DATA インタフェース表, 16-4
 - ATTENTION_DESCRIPTION 列, 16-6
 - CASCADE_FLAG 列, 16-6, 16-7
 - DBPASSWORD_EXIST_FLAG 列, 16-6, 16-7
 - DBPASSWORD 列, 16-6
 - DIRPASSWORD 列, 16-6
 - MAPPING_LEVEL 列, 16-6, 16-7
 - MAPPING_TYPE 列, 16-6, 16-7
 - NEEDS_ATTENTION_FLAG 列, 16-6
 - OLD_SCHEMA_TYPE 列, 16-5
 - PASSWORD_VERIFIER 列, 16-5
 - PHASE_COMPLETED 列, 16-6, 16-7
 - SHARED_SCHEMA 列, 16-6, 16-7
 - USERDN_EXIST_FLAG 列, 16-6, 16-7
 - USERDN 列, 16-5, 16-7
 - USERNAME 列, 16-5
 - アクセス, 16-5
- OS_AUTHENT_PREFIX パラメータ, 9-6
 - CyberSafe 認証, 5-8
- OS_ROLES パラメータ
 - 設定, 12-6
- OSS.SOURCE.MY_WALLET パラメータ, 7-17, 7-23

P

- PASSWORD_VERIFIER 列, 16-5
- PHASE_COMPLETED 列, 16-6, 16-7
- PKI, 「公開鍵インフラストラクチャ」を参照
- protocol.ora ファイル
 - CDS 用のパラメータ, 12-12
 - DCE.AUTHENTICATION パラメータ, 12-11
 - DCE.LOCAL_CELL_USERNAMES パラメータ, 12-11
 - DCE.PROTECTION パラメータ, 12-11
 - DCE.TNS_ADDRESS_OID パラメータ, 12-11

R

- RADIUS, 1-10
 - sqlnet.ora ファイルのサンプル, A-3
 - アカウント, 4-20
 - 構成, 4-10

- システム要件, 1-16
- スマートカード, 1-10, 4-8, 4-15, C-2
- 同期認証モード, 4-4
- 認証パラメータ, B-3
- 認証モード, 4-4
- 非同期認証モード, 4-6
- 秘密鍵の場所, 4-15
- 要求 / 応答
 - 認証, 4-6, C-1, D-1
 - ユーザー・インタフェース, C-1, C-2, D-1
- RADIUS サーバーによるロールの管理, 4-22
- RADIUS での非同期認証モード, 4-6
- RADIUS での要求 / 応答認証, 4-6
- RC4 暗号化アルゴリズム, 1-5, 2-3
- RDBMS_SERVER_DN パラメータ, 19-21
- REMOTE_OS_AUTHENT パラメータ
 - CyberSafe 認証, 5-8
- RSA Security 社 (RSA), 1-5

S

- Secure Sockets Layer (SSL), 1-9, 7-1, 8-1, 8-2, 15-32, 18-1, 18-4, 18-10
 - Cipher Suite, B-11
- Oracle 環境での構成要素, 7-3
- Oracle 環境での認証手続き, 7-5
- sqlnet.ora ファイルのサンプル, A-2
- Wallet, 7-3
- Wallet の場所、パラメータ, B-14
- アーキテクチャ, 7-7
- 業界標準プロトコル, 7-2
- 共有スキーマ, 15-20
- クライアントに構成, 8-10
- クライアント認証パラメータ, B-12
- クライアントの構成, 7-12
- 権限, 7-11
- 構成, 7-12
- サーバーの構成, 7-22
- システム要件, 1-16
- 使用可能にする, 7-12, 8-8
- 証明書, 7-3
- 接続の使用禁止, 18-10
- 認可, 7-11
- 認証局, 7-3
- 認証パラメータ, B-9
- バージョン・パラメータ, B-11
- ハンドシェイク, 7-5

- 必要なクライアント認証, 7-28
- 他の認証方式の併用, 7-7
- ロール, 7-11
- SecurID, 4-5
 - トークン・カード, 4-5
- SHARED_SCHEMA 列, 16-6, 16-7
- spfile.ora, 15-30, 15-31, 15-32, 15-60, 15-62, 19-21
- SQL*Plus
 - Active Directory を介したデータベースへの接続, E-6
- SQLNET.AUTHENTICATION_GSSAPI_SERVICE パラメータ, 5-7, B-2
- SQLNET.AUTHENTICATION_KERBEROS5_SERVICE パラメータ, 6-8
- SQLNET.AUTHENTICATION_SERVICES パラメータ, 4-11, 5-7, 6-8, 7-21, 7-22, 7-29, 9-3, 9-4, B-2
- SQLNET.CRYPTO_CHECKSUM_CLIENT パラメータ, 2-13, A-6
- SQLNET.CRYPTO_CHECKSUM_SERVER パラメータ, 2-13, A-6
- SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT パラメータ, 2-13, A-9
- SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER パラメータ, 2-13, A-9
- SQLNET.CRYPTO_SEED パラメータ, 2-11, A-10
- SQLNET.ENCRYPTION_CLIENT パラメータ, 2-11, A-5
- SQLNET.ENCRYPTION_SERVER パラメータ, 2-11, A-5
- SQLNET.ENCRYPTION_TYPES_CLIENT パラメータ, 2-11, A-8
- SQLNET.ENCRYPTION_TYPES_SERVER パラメータ, 2-11, A-7
- SQLNET.FIPS_140 パラメータ, D-3
- SQLNET.KERBEROS5_CC_NAME パラメータ, 6-8
- SQLNET.KERBEROS5_CLOCKSKEW パラメータ, 6-8
- SQLNET.KERBEROS5_CONF_MIT パラメータ, 6-9
- SQLNET.KERBEROS5_CONF パラメータ, 6-9
- SQLNET.KERBEROS5_KEYTAB パラメータ, 6-9
- SQLNET.KERBEROS5_REALMS パラメータ, 6-9
- sqlnet.ora ファイル, 15-34
 - CDS が名前を解決できるように変更, 12-17
 - CyberSafe サンプル, A-3
 - CyberSafe を使用するクライアントとサーバーのパラメータ, B-2
 - FIPS 140-1 パラメータ, D-2

Kerberos サンプル, A-3
Kerberos を使用するクライアントとサーバーのパラメータ, B-2
NAMES.DIRECTORY_PATH パラメータ, 12-17
Oracle Advanced Security 暗号化のサンプル, A-2
Oracle Advanced Security チェックサムのサンプル, A-2
OSS.SOURCE.MY_WALLET パラメータ, 7-17, 7-23
RADIUS のサンプル, A-3
RADIUS を使用するクライアントとサーバーのパラメータ, B-3
SQLNET.AUTHENTICATION_GSAPPI_SERVICE パラメータ, B-2
SQLNET.AUTHENTICATION_GSSAPI_SERVICE パラメータ, 5-7
SQLNET.AUTHENTICATION_KERBEROS5_SERVICE パラメータ, 6-8
SQLNET.AUTHENTICATION_SERVICES パラメータ, 5-7, 6-8, 7-21, 7-22, 7-29, 9-3, 9-4, B-2
SQLNET.CRYPTO_CHECKSUM_CLIENT パラメータ, 2-13, A-6
SQLNET.CRYPTO_CHECKSUM_SERVER パラメータ, 2-13, A-6
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT パラメータ, 2-13, A-9
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER パラメータ, 2-13, A-9
SQLNET.CRYPTO_SEED パラメータ, 2-11, A-10
SQLNET.ENCRYPTION_CLIENT パラメータ, A-5
SQLNET.ENCRYPTION_SERVER パラメータ, 2-11, A-5
SQLNET.ENCRYPTION_TYPES_CLIENT パラメータ, 2-11, A-8
SQLNET.ENCRYPTION_TYPES_SERVER パラメータ, 2-11, A-7
SQLNET.FIPS_140 パラメータ, D-3
SQLNET.KERBEROS5_CC_NAME パラメータ, 6-8
SQLNET.KERBEROS5_CLOCKSKEW パラメータ, 6-8
SQLNET.KERBEROS5_CONF_MIT パラメータ, 6-9
SQLNET.KERBEROS5_CONF パラメータ, 6-9
SQLNET.KERBEROS5_KEYTAB パラメータ, 6-9
SQLNET.KERBEROS5_REALMS パラメータ, 6-9
SSL_CLIENT_AUTHENTICATION パラメータ, 7-29

SSL_CLIENT_AUTHENTICATION パラメータ, 7-17
SSL_VERSION パラメータ, 7-21, 7-27
SSL サンプル, A-2
SSL を使用するクライアントとサーバーのパラメータ, B-9
共通のサンプル, A-3
サンプル, A-2
トレース・ファイルの設定のサンプル, A-2
SQLNET.RADIUS_ALTERNATE_PORT パラメータ, 4-17
SQLNET.RADIUS_ALTERNATE_RETRIES パラメータ, 4-17
SQLNET.RADIUS_ALTERNATE_TIMEOUT パラメータ, 4-17
SQLNET.RADIUS_ALTERNATE パラメータ, 4-17
SQLNET.RADIUS_SEND_ACCOUNTING パラメータ, 4-20
SSL_CLIENT_AUTHENTICATION パラメータ, 7-17, 7-29
SSL_VERSION パラメータ, 7-21, 7-27
SSL, 「Secure Sockets Layer (SSL)」を参照
SSL でのクライアント認証, 7-28
SSO, 「シングル・サインオン (SSO)」を参照
SYS スキーマ, 16-4

T

tnsnames.ora ファイル, 15-34
tnnfg を使用して CDS にロード, 12-16
改名, 12-16
接続記述子を CDS にロードするために変更, 12-15

U

USERDN_EXIST_FLAG 列, 16-6, 16-7
USERDN 列, 16-5, 16-7
UserID 属性, 15-52
USERNAME 列, 16-5
userpkcs12 属性, 19-13

W

Wallet, 7-4
アップロード, 18-10
オープン, 17-12
管理, 17-10

クローズ, 17-12
削除, 17-16
作成, 17-11
証明書の管理, 17-17
信頼できる証明書の管理, 17-22
ダウンロード, 18-6
場所の設定, 7-16, 7-23
パスワードの変更, 17-16, 18-7
保存, 17-15
ローカル・システムでのオープン, 18-3
Windows 2000 ドメイン
ディレクトリ・サーバー機能を使用する Oracle の
クライアントとサーバーに必要なドメイン,
E-10
Windows エクスプローラ
Active Directory での Oracle オブジェクトとの
統合, E-6
ディレクトリ・サーバー・オブジェクトへのアクセ
ス, E-18
Windows システム固有の認証
インストール, E-15
概要, E-15
方式と使用, E-15
利点, E-15

X

X.500 ネーミング規則, E-17
X.509 PKI 証明書標準, 8-3

あ

アカウント、RADIUS, 4-20
アクセス制御リスト
使用可能なセキュリティ・グループ, E-21
セキュリティ・グループへのアクセス, E-22
アダプタ, 1-14
暗号化, 1-15
暗号化とチェックサム
アクティブにする, 2-6
クライアント・プロファイルの暗号化, A-10
サーバーの暗号化選択リスト, A-7
サーバーの暗号化レベルの設定, A-5
折衝, 2-7
パラメータの設定, 2-9
暗号ブロック連鎖 (CBC) モード, 1-5

い

一般的なマニュアル参照
Windows NT 固有の認証方式, E-15
インストール
サーバーのキー, 11-3
インターネット, 7-6

え

エラー・メッセージ
ORA-12650, 2-6, 2-7, 2-8, A-7, A-8, A-9
ORA-28890, 8-15
Oracle Enterprise Security Manager, 15-32, 15-48
トレース, 15-63
エンタープライズ・ドメイン, 15-50, 19-31
エンタープライズ・ドメイン管理者, 19-35
エンタープライズ・ドメインの管理, 19-31
エンタープライズ・ユーザー
Active Directory での表示, E-9
管理, 19-6
作成, 19-6
パスワードの定義, 19-10
エンタープライズ・ユーザー・セキュリティ
Oracle Enterprise Security Manager, 15-4
Oracle コンテキスト, 15-51
SSL, 15-32
SSL サービス名, 15-33
SSL の最終構成, 15-41
UserID 属性, 15-52
エンタープライズ・ドメイン, 15-8, 15-43, 15-50
エンタープライズ・ユーザー, 15-7, 15-44, 15-47
Wallet の作成, 15-45
許可, 15-46, 15-55
構成, 15-54
作成, 15-54
追加, 15-44
データベース・アクセスを可能にする, 15-52
パスワードの作成, 15-56
マッピング, 15-21, 15-46
ユーザー ID の作成, 15-56
エンタープライズ・ロール, 15-7
概要, 15-2
管理者, 15-53

共有スキーマ, 15-18
 SSL, 15-20
 構成, 15-19
 作成, 15-20
グループ
 OracleDBCreators, 15-11
 OracleDBSecurity, 15-11
グローバル・ロール, 15-7, 15-40
権限, 15-41
構成要素, 15-24
作成要件, E-14
自動ログイン, 15-37
スキーマ, 15-40
セッションに対する権限, 15-40
ディレクトリ・エントリ, 15-7
ディレクトリ・サービス, 15-28
データベース・クライアント, 15-42
トラブルシューティング, 15-57
 トレース, 15-63
認証局, 15-28
パスワード・アクセシブル・ドメイン, 15-53
パスワードの最終構成, 15-49
秘密鍵の復号化に失敗した場合, 15-63
ユーザー検索ベース, 15-51
リスナー, 15-33, 15-36, 15-39
ロール, 15-40
エンタープライズ・ドメイン
 Active Directory での表示, E-9
エンタープライズ・ロール, 19-38, 19-40
 Active Directory での表示, E-9
エンタープライズ・ロール権限受領者, 19-42

お

オペレーティング・システム
 認証の概要, E-15

か

管理者, 19-22, 19-35

き

共有スキーマ, 15-19, 15-40
 SSL, 15-20

く

グローバル・スキーマ, 15-40
グローバル・ディレクトリ・サービス (GDS), 10-4
グローバル・ロール, 15-40, 19-40

け

権限, 15-41
権限受領者, 19-42

こ

公開鍵インフラストラクチャ (PKI), 1-9, 1-10, 8-2, 18-2
公開鍵と秘密鍵のペア, 8-2
構成
 CyberSafe 認証サービス・パラメータ, 5-5
 CyberSafe を使用する Oracle サーバー, 5-3
 DCE CDS ネーミングを使用するクライアント, 12-13
 DCE Integration で使用するクライアント, 12-11
 DCE Integration を使用する DCE, 11-2
 JDBC Thin のサポート, 3-1
 Kerberos 認証サービス・パラメータ, 6-5
 Kerberos を使用する Oracle サーバー, 6-2
 Oracle Net/DCE 外部ロール, 12-6
 RADIUS 認証, 4-10
 Secure Sockets Layer (SSL)
 クライアント, 8-10
 SSL, 7-12
 クライアント, 7-12
 サーバー, 7-22
 共有スキーマ, 15-19
構成ファイル
 CyberSafe, B-2
 Kerberos, B-2

さ

作成
 CDS での Oracle ディレクトリ, 11-3
 プリンシパルとアカウント, 11-2

し

- システム要件, 1-16
 - CyberSafe, 1-16
 - DCE Integration, 10-2
 - Kerberos, 1-16
 - RADIUS, 1-16
 - SSL, 1-16
- 自動ログイン, 15-37
- 証明書, 7-3
 - 作成, 8-2
- 証明書認証エンタープライズ・ユーザー, 18-2
- 初期化パラメータ・ファイル
 - CyberSafe を使用するクライアントとサーバーのパラメータ, B-2
 - Kerberos を使用するクライアントとサーバーのパラメータ, B-2
 - RADIUS を使用するクライアントとサーバーのパラメータ, B-3
 - SSL を使用するクライアントとサーバーのパラメータ, B-9
- シングル・サインオン (SSO), 1-10, 8-3, 13-3, 18-2

す

- スキーマ・マッピング, 19-28, 19-36
- スマートカード, 1-10
 - RADIUS, 1-10, 4-8, 4-15, C-2

せ

- 制限, 1-17
- セキュリティ
 - インターネット, 1-2
 - イントラネット, 1-2
 - 脅威, 1-2
 - 識別情報の偽造, 1-3
 - 辞書攻撃, 1-3
 - データの改ざん, 1-3
 - パスワード関連, 1-3
 - 傍受, 1-2
 - クライアント / サーバー間, 7-6
- セキュリティ・オプション, 19-35
- セキュリティ・グループ
 - アクセス, E-22

セキュリティとの関係

- ヒント, 18-9
- リマインダ, 18-9
- セッションに対する権限, 15-40
- 接続
 - DCE, 13-3
 - ユーザー名とパスワードを使用, 13-4
 - ユーザー名とパスワードを使用しない, 13-3
 - Oracle データベース
 - ロールを確認するため, 12-8
 - 複数のセル, 12-5
 - ユーザー名とパスワードを使用, 9-2

ち

- チェックサムと暗号化、アクティブにする, 2-6

て

- ディレクトリ
 - パスワードの変更, 18-7
- ディレクトリ・サーバー
 - Active Directory で Oracle を使用するための要件, E-17
 - Active Directory との統合, E-17
 - Microsoft のツールとの統合, E-6
 - Oracle Net ディレクトリ・ネーミングを Active Directory に作成する場合の要件, E-13
 - Oracle9i に統合されている機能, E-2
 - Oracle ディレクトリ・オブジェクトの Active Directory での表示方法, E-8
 - Windows ログイン資格証明との統合, E-8
 - エンタープライズ・ユーザー・セキュリティを Active Directory に作成する場合の要件, E-14
 - セキュリティ・ドメインの作成, E-26
 - ディレクトリ・オブジェクト型記述の拡張機能, E-7
 - ディレクトリ・サーバーの自動検出, E-4
 - ユーザー・インタフェースの拡張機能, E-6
- ディレクトリ・ベース
 - 定義, 19-8
- データ暗号化規格 (DES), 2-3
 - DES40 暗号化アルゴリズム, 2-3
 - DES 暗号化アルゴリズム, 1-5
 - トリプル DES 暗号化アルゴリズム, 1-5, 2-3
- データの整合性, 1-6
- データ・プライバシー, 1-4

データベース

- パスワードの変更, 18-7
- データベース管理者, 19-27
- データベース・スキーマ・マッピング, 19-28, 19-36
- データベース・セキュリティ, 19-27
- データベース・セキュリティ・オプション, 19-35
- データベース・ドメインのメンバーシップ, 19-32
- データベース・パスワード・ベリファイア, 16-3
- デジタル署名, 8-2

と

- 同期認証モード、RADIUS, 4-4
- トークン・カード, 1-11
- ドメイン, 19-25, 19-31, 19-35
- ドメイン管理者, 19-35
- ドメイン・データベース・スキーマ・マッピング, 19-36
- ドメイン・ネーム・サービス (DNS), 10-4
- ドメインのメンバーシップ, 19-32
- トラスト・ポイント, 8-2
- トラブルシューティング, 5-10, 6-19, 8-15, 15-57
- 取消し, 8-3
- トリプル DES 暗号化アルゴリズム, 1-5
- トレース, 15-63
- トレース・ファイル
 - sqlnet.ora ファイルにサンプルを設定, A-2

に

- 認可, 1-13
- 認証, 1-7, 1-14
 - RADIUS のモード, 4-4
 - Windows システム固有の認証方式の使用, E-15
 - 概要, E-15
 - 複数のメソッドの構成, 9-4
 - 方式, 1-9
- 認証局, 7-3, 8-2
- 認証局、esm-genca ツールによる擬似, 19-12
- 認証された RPC
 - プロトコル・アダプタ, 10-3

ね

- ネットワーク・プロトコル境界, 1-15

は

- パスワード・アクセシブル・ドメイン, 19-25
- パスワード認証エンタープライズ・ユーザー, 18-2, 18-11
 - パスワードの変更, 18-11
- パスワード・ヒント, 18-9
- パスワード・リマインダ, 18-9
- パラメータ
 - JDBC の構成, 3-5
 - 暗号化とチェックサム, 2-9
 - 認証
 - CyberSafe, B-2
 - Kerberos, B-2
 - RADIUS, B-3
 - Secure Sockets Layer (SSL), B-9
- ハンドシェイク
 - SSL, 7-5

ひ

- 秘密鍵
 - RADIUS での場所, 4-15
- ヒント, 18-9

ふ

- 不明瞭化, 3-4
- ブラウズ, 19-13
- 分散コンピューティング環境 (DCE)
 - CDS ネーミング・アダプタ・コンポーネント, 10-4
 - DCE CDS ネーミングを使用するクライアントの構成, 12-13
 - DCE Integration で使用するクライアントの構成, 12-11
 - DCE Integration を使用するための構成, 11-2
 - dce_service_name の確認, 13-2
 - DCE グループのマッピングの確認, 12-8
 - DCE と CDS にアクセスせずにクライアントに接続, 14-2
 - listener.ora パラメータ, 12-2
 - Multi-Protocol Interchange, 10-5
 - Oracle サーバーへの接続, 13-3
 - Oracle ロールへのグループのマッピング、構文, 12-6
 - protocol.ora ファイル・パラメータ, 12-11
 - REMOTE_OS_AUTHENT パラメータ, 12-4

Secure Core サービス, 10-5
tnsnames.ora ファイル, 12-2
tnsnames.ora ファイルのサンプル・アドレス,
12-15
下位互換性, 10-2
外部的に認証されるアカウント, 12-4
外部ロールの設定, 12-6
概要, 10-2
構成ファイルが必要, 12-3
構成要素, 10-3
サーバーの構成, 12-3
サンプル listener.ora ファイル, 14-2
サンプル tnsnames.ora ファイル, 14-2
サンプル・パラメータ・ファイル, 14-2
接続
 Oracle データベース, 13-1
通信およびセキュリティ, 10-3
リスナーの起動, 13-2

へ

米国連邦情報処理標準
 構成, xxiii
米国連邦情報処理標準 (FIPS), 1-6, D-1
 sqlnet.ora パラメータ, D-2

ま

マッピング, 19-36

め

メンバーシップ, 19-32

ゆ

ユーザー移行ユーティリティ
 ATTENTION_DESCRIPTION 列, 16-6
 CASCADE_FLAG 列, 16-6, 16-7
 CASCADE パラメータ, 16-8
 DBPASSWORD_EXIST_FLAG 列, 16-6, 16-7
 DBPASSWORD 列, 16-6
 DIRPASSWORD 列, 16-6
 LOGFILE 優先順位, 16-27
 MAPPING_LEVEL 列, 16-6, 16-7
 MAPPING_TYPE 列, 16-6, 16-7
 MAPSCHEMA パラメータ

 PRIVATE, 16-17
 SHARED, 16-17
 MAPTYPE パラメータ
 DB マッピング・タイプ, 16-18
 DOMAIN マッピング・タイプ, 16-18
 ENTRY マッピング・レベル, 16-18
 SUBTREE マッピング・レベル, 16-18, 16-25
 NEEDS_ATTENTION_FLAG 列, 16-6
 OLD_SCHEMA_TYPE 列, 16-5
 ORCL_GLOBAL_USR_MIGRATION_DATA インタ
 フェース表, 16-4
 PASSWORD_VERIFIER 列, 16-5
 PHASE_COMPLETED 列, 16-6, 16-7
 SHARED_SCHEMA 列, 16-6, 16-7
 SYS スキーマ, 16-4
 USERDN_EXIST_FLAG 列, 16-6, 16-7
 USERDN 列, 16-5, 16-7
 USERNAME 列, 16-5
 USER パラメータ
 ALL_EXTERNAL, 16-15
 ALL_GLOBAL, 16-15
 LIST, 16-15
 USERSFILE, 16-15
 X.509 v3 証明書, 16-9
 インタフェース表へのアクセス, 16-5
 共有スキーマ・マッピング, 16-8
 現行リリースでの SSL 認証, 16-9
 削除されたスキーマ・オブジェクトの取得, 16-24
 証明書認証ユーザー, 16-9
 データベース・パスワード・ベリファイア, 16-3
 パスワード認証ユーザー, 16-9
 ヘルプの表示, 16-13
 ユーティリティのディレクトリ位置, 16-10
 例
 CASCADE=NO の使用, 16-22
 CASCADE=YES の使用, 16-23
 MAPSCHEMA=PRIVATE の使用, 16-21
 MAPSCHEMA=SHARED の使用, 16-22
 MAPTYPE オプションの使用, 16-25
 PARFILE、USERSFILE および LOGFILE パラ
 メータの使用, 16-27
 パラメータ・テキスト・ファイル (par.txt),
 16-26
 ユーザー・リスト・テキスト・ファイル
 (usrs.txt), 16-26
 ユーザー検索ベース, 15-51, 19-21

よ

要件

- Active Directory で Oracle を使用, E-17
- Oracle Net ディレクトリ・ネーミングを Active Directory に作成する場合, E-13
- エンタープライズ・ユーザー・セキュリティを Active Directory に作成する場合, E-14

り

- リスナー, 15-32, 15-33, 15-36, 15-39
 - DCE 環境での起動, 13-2
 - listener.ora ファイル, 15-35
 - エンドポイント
 - SSL の構成, 7-29
- リマインダ, 18-9

れ

- レルム (CyberSafe), 5-3
- レルム (Kerberos), 6-3

ろ

- ロール, 15-40, 19-11, 19-38, 19-40, 19-42
 - RADIUS サーバーによる管理, 4-22
- ロール、外部、DCE グループへのマッピング, 12-6
- ロール権限受領者, 19-42
- ログアウト, 18-10