

# Oracle8*i*

Net8 管理者ガイド

リリース 8.1

2000 年 2 月

部品番号 : J00959-01

ORACLE®



---

Oracle® Net8 管理者ガイド, リリース 8.1

部品番号 : J00959-01

原本名 : Net8 Administrator's Guide, Release 8.1.6

原本部品番号 : A76933-01

原本著者 : Deborah Steiner

原本協力者 : Pierre Baudin, Toby Close, Santanu Datta, Harvey Eneman, Shuvayu Kanjila, Cynthia Kibbe, Scot McKinley, Sergio Mendiola, Ed Miner, Peter Povinec, Radhika Shah, Richard Smith, Juliet Tran, Sandy Venning, and Norman Woo

Copyright © 1996, 1999, 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 .....	xiii
------------	------

## 第 I 部 Net8 の概要と概念

### 1 Net8 の概要

Net8 の概要 .....	1-2
Oracle の接続性の概要 .....	1-4
構成の概要 .....	1-5
ネーミング・メソッドの構成 .....	1-5
サーバー上でのリスナーの構成 .....	1-7

### 2 Net8 の概念

ネットワーク環境 .....	2-2
分散処理 .....	2-2
分散データベース .....	2-2
接続操作 .....	2-2
サーバーへの接続 .....	2-3
サーバーからの切断 .....	2-5
例外操作 .....	2-6
Net8 接続モデル .....	2-6
マルチスレッド・サーバー・モデル .....	2-6
専用サーバー・モデル .....	2-11
クライアント接続オプション .....	2-18
複数のリスナーに対するクライアント・ロード・バランス .....	2-18
複数のリスナーに対する接続時フェイルオーバー .....	2-18



可用性を高めるための透過的アプリケーション・フェイルオーバー .....	2-19
<b>リスナーに対するサービス登録 .....</b>	<b>2-19</b>
利点 .....	2-21
デフォルトのローカル・リスナーへの登録 .....	2-21
デフォルト以外リスナーへの登録 .....	2-21
リモート・リスナーへの登録 .....	2-22
<b>接続時ロード・バランス化 .....</b>	<b>2-23</b>
構成 .....	2-23
<b>SID ではなくサービス名によるデータベースの識別 .....</b>	<b>2-27</b>
SERVICE_NAME および INSTANCE_NAME の使用による接続記述子 .....	2-28
同じサービスに属している複数のインスタンス .....	2-29
複数のサービスに属している単一のインスタンス .....	2-29
移行 .....	2-29
<b>ネーミング・メソッドによるサービスの解決 .....</b>	<b>2-30</b>
ローカル・ネーミング .....	2-30
ディレクトリ・ネーミング .....	2-30
Oracle Names .....	2-31
ホスト・ネーミング .....	2-32
外部ネーミング .....	2-32
<b>Net8 および LDAP 準拠ディレクトリ・サーバー .....</b>	<b>2-33</b>
概要 .....	2-33
ディレクトリ・ネーミングの概要 .....	2-35
ディレクトリ・エントリ .....	2-36
ディレクトリでのエントリの追加または変更 .....	2-38
ディレクトリ・ネーミングを使用したクライアントの接続 .....	2-39
絶対名の指定 .....	2-41
Net8 のネーミングおよびディレクトリ設計 .....	2-43

### 3 Net8 アーキテクチャ

<b>Transparent Network Substrate (TNS) .....</b>	<b>3-2</b>
<b>スタック通信の概要 .....</b>	<b>3-2</b>
<b>代表的な Net8 クライアント / サーバー環境でのスタック通信 .....</b>	<b>3-4</b>
Net8 クライアント側の対話 .....	3-4
サーバー側の対話 .....	3-7
サーバー / サーバー間の対話 .....	3-8



ディレクトリ・サーバーに接続しているクライアントのスタック通信 .....	3-8
JDBC クライアントのスタック通信 .....	3-9
Oracle8i JServer クライアントのスタック通信 .....	3-13

## 4 Net8 製品およびツール

Net8 Client .....	4-2
Net8 Server .....	4-2
Oracle プロトコル .....	4-2
TCP/IP .....	4-2
SSL 付き TCP/IP .....	4-2
SPX .....	4-3
Named Pipes .....	4-3
LU6.2 .....	4-3
Bequeath .....	4-4
Oracle Connection Manager .....	4-4
Oracle Connection Manager プロセス .....	4-5
接続集中化 .....	4-6
Net8 アクセス制御 .....	4-6
複数プロトコル・サポート .....	4-7
Oracle Names .....	4-8
概要 .....	4-9
管理リージョン .....	4-10
ドメイン .....	4-11
データ・リポジトリとしての Oracle Names .....	4-11
Oracle Names サーバーに格納されるデータ .....	4-12
ネットワーク・コンポーネントの編成およびネーミング .....	4-14
管理責任を分散する複数のリージョンの使用 .....	4-16
Oracle Names のバージョン間の相違 .....	4-18
検出の理解 .....	4-20
外部ネーミング・サービス .....	4-21
認証方式 (authentication method) .....	4-21
Oracle Advanced Security .....	4-22
ネットワーク・セキュリティ .....	4-22
シングル・サインオン .....	4-22
Distributed Computing Environment インテグレーション .....	4-23
管理ツール .....	4-23



Net8 Configuration Assistant .....	4-23
Oracle Net8 Assistant .....	4-24
制御ユーティリティ .....	4-24

## 5 ネットワークのプランニング

計画の概要 .....	5-2
ネットワーク・レイアウトの決定 .....	5-2
接続記述子への名前の解決 .....	5-2
大規模ネットワーク・パフォーマンスの向上 .....	5-4
共有サーバーまたは事前生成済専用サーバーの使用 .....	5-4
リスナー・キューサイズの増加 .....	5-5
接続プーリングの使用 .....	5-5
接続集中化機能の使用 .....	5-6
クライアント・ロード・バランスの使用 .....	5-7
データ転送の最適化を目的としたセッション・データ・ユニット (SDU) サイズの調整 .....	5-7
TCP/IP の持続的バッファ・フラッシング .....	5-8
計画のサマリー .....	5-8

## 第 II 部 構成

## 6 ネーミング・メソッドの構成

構成の概要 .....	6-2
接続記述子の理解 .....	6-2
構成モデル .....	6-3
ネーミング・メソッド .....	6-4
ローカル・ネーミング・メソッドの構成 .....	6-4
インストール中の tnsnames.ora ファイル構成 .....	6-5
インストール後の tnsnames.ora ファイル構成 .....	6-5
ディレクトリ・ネーミング・メソッドの構成 .....	6-13
ディレクトリ・ネーミング構成ステップ .....	6-13
データベース・サービス・エントリに対する接続情報の変更 .....	6-26
OracleNetAdmins グループへのユーザーの追加と削除 .....	6-27
tnsnames.ora ファイルからのネット・サービス名のエクスポート .....	6-28
Oracle Names サーバーからのネットワーク・オブジェクトのエクスポート .....	6-31
Oracle Names メソッドの構成 .....	6-43



作業 1: Oracle Names オプションの検討 .....	6-43
作業 2: 必要なコンポーネントのインストール .....	6-44
作業 3: Oracle Names サーバーの作成 .....	6-44
作業 4: Oracle Names サーバーを使用するためのクライアントとデータベース・サーバーの構成 .....	6-55
作業 5: リスナーの構成 .....	6-59
作業 6: Oracle Names サーバーへのデータベースの登録 .....	6-59
作業 7: (オプション) 委任管理リージョンへのドメインの委任 .....	6-66
作業 8: データベースへの接続 .....	6-67
<b>ホスト・ネーミング・メソッドの構成</b> .....	6-68
作業 1: グローバル・データベース名を使用したリスナーの構成 .....	6-68
作業 2: 第 1 番目のネーミング・メソッドとしての HOSTNAME の構成 .....	6-69
作業 3: ホスト名解決の環境の設定 .....	6-70
作業 4: データベースへの接続 .....	6-71
<b>外部ネーミング・メソッドの構成</b> .....	6-71
NIS の構成 .....	6-71
NDS の構成 .....	6-75

## 7 リスナーの構成

<b>構成の概要</b> .....	7-2
デフォルト・リスニング・プロトコル・アドレス .....	7-2
インストール中の構成作業 .....	7-3
Oracle Net8 Assistant による構成 .....	7-4
<b>listener.ora ファイル変更後のリスナーの再起動</b> .....	7-4
<b>リスナー・プロトコル・アドレスの構成</b> .....	7-5
Oracle8i JServer のためのリスニング・アドレスの構成 .....	7-7
<b>デフォルト以外のアドレスを使用するリスナーの構成</b> .....	7-8
LOCAL_LISTENER パラメータの構成 .....	7-8
ネーミング・メソッドの構成 .....	7-9
<b>多数の接続要求の処理</b> .....	7-10
<b>静的サービス情報の構成</b> .....	7-10
<b>事前生成済専用サーバーの構成</b> .....	7-13

## 8 Net8 の拡張機能

拡張ネットワーク・アドレスおよび接続データ情報の構成 .....	8-2
----------------------------------	-----



リスナー・プロトコル・アドレスのリスト作成 .....	8-2
アドレス・リスト・パラメータの構成 .....	8-3
拡張接続データ・パラメータの構成 .....	8-8
<b>透過的アプリケーション・フェイルオーバーの構成 .....</b>	<b>8-11</b>
FAILOVER_MODE パラメータ .....	8-11
TAF 実装 .....	8-12
<b>名前解決のクライアント属性の構成 .....</b>	<b>8-15</b>
クライアントのデフォルト・ドメインの構成 .....	8-15
ネーミング・メソッドの優先順位 .....	8-16
接続要求のルーティング .....	8-18
<b>プロファイルの詳細情報の構成 .....</b>	<b>8-20</b>
<b>優先 Oracle Names サーバーの構成 .....</b>	<b>8-23</b>
<b>Oracle Connection Manager 機能の構成 .....</b>	<b>8-25</b>
接続集中化機能の使用 .....	8-25
複数プロトコル・サポートの使用 .....	8-28
Net8 アクセス制御の使用 .....	8-31
Oracle Connection Manager を使用する場合のクライアントの構成 .....	8-33
<b>非 Oracle データベース・サービスへの接続の構成 .....</b>	<b>8-39</b>
外部プロシージャを使用する場合の Net8 の構成 .....	8-39
Oracle 異種サービスを使用する場合の Net8 の構成 .....	8-43
Oracle Rdb データベースを使用する場合の Net8 の構成 .....	8-46

## 9 マルチスレッド・サーバーの構成

概要 .....	9-2
MTS_DISPATCHERS パラメータを使用した MTS の構成 .....	9-3
ディスパッチャの初期数の設定 .....	9-7
接続プーリングの使用 .....	9-8
リソースの割当て .....	9-9
クライアントでの MTS の使用 .....	9-9
クライアントでの MTS の変更 .....	9-9

## 10 プログラマのための Net8 拡張機能

Net8 OPEN .....	10-2
Net8 OPEN API ファンクション・コール .....	10-3
Net8 OPEN アプリケーション・プログラム・インタフェースのインストール先 .....	10-6



独自のアプリケーションの開発 .....	10-7
Net8 OPEN アプリケーションを使用するためのシステムの構築 .....	10-7
サンプル・プログラム .....	10-8
Net8 OPEN API エラー .....	10-8
<b>UNIX クライアント・プログラミング</b> .....	10-9
シグナル・ハンドラとアラームのプログラミング .....	10-9
Bequeath .....	10-10

## 第 III 部 Net8 テストおよびトラブルシューティング

### 11 接続の確立とネットワークのテスト

<b>データベースへの接続</b> .....	11-2
Net8 制御ユーティリティの使用 .....	11-2
Net8 コンポーネント起動の概要 .....	11-3
接続文字列の入力 .....	11-7
接続識別子の構文特性 .....	11-7
接続の開始 .....	11-8
<b>ネットワークのテスト</b> .....	11-10
Oracle Names サーバーのテスト .....	11-10
Oracle Net8 Assistant または NAMESCTL ユーティリティによるネットワーク・オブジェクト のテスト .....	11-11
リスナーのテスト .....	11-13
Oracle Connection Manager のテスト .....	11-13
サーバーでの構成のテスト .....	11-13
クライアントからのネットワーク接続のテスト .....	11-14

### 12 トラブルシューティング

<b>Net8 の診断</b> .....	12-2
サーバーの診断 .....	12-2
クライアントの診断 .....	12-3
<b>一般的なエラー・メッセージの解決</b> .....	12-5
<b>フィールドからのトラブルシューティングのヒント</b> .....	12-11
トラブルシューティング時のチェックリスト .....	12-12
<b>ログ・ファイルとトレース・ファイルを使用したネットワーク上の問題点の トラブルシューティング</b> .....	12-12



<b>エラー情報のロギング</b> .....	12-13
エラー・スタック .....	12-13
ログ・ファイル名 .....	12-15
ログ・パラメータの理解と設定 .....	12-16
制御ユーティリティ実行時のログ設定 .....	12-19
ログ・ファイルの使用 .....	12-20
リスナー・ログ・ファイルの理解 .....	12-20
Oracle Connection Manager ログの理解 .....	12-24
<b>エラー情報のトレース</b> .....	12-28
トレース・ファイルの名前 .....	12-28
トレース・パラメータの理解と設定 .....	12-29
制御ユーティリティ実行時のトレース設定 .....	12-35
Net8 トレースの評価 .....	12-36
Trace Assistant を使用したトレース・ファイルの検査 .....	12-39
<b>オラクル社カスタマ・サポート・センターへの連絡</b> .....	12-52

## 第 IV 部    リファレンス

### A    制御ユーティリティ

<b>動作モード</b> .....	A-2
<b>リスナー制御ユーティリティ (LSNRCTL)</b> .....	A-2
SET および SHOW コマンド .....	A-3
セキュリティ .....	A-3
LSNRCTL コマンド .....	A-3
<b>Oracle Names 制御ユーティリティ (NAMESCTL)</b> .....	A-25
SET および SHOW コマンド .....	A-26
セキュリティ .....	A-26
NAMESCTL の分散処理 .....	A-26
NAMESCTL の確認モード .....	A-27
NAMESCTL コマンド .....	A-27
<b>Oracle Connection Manager 制御ユーティリティ (CMCTL)</b> .....	A-77
SET および SHOW コマンド .....	A-78
Shutdown コマンド .....	A-78
CMCTL コマンド .....	A-78



## B プロトコル・アドレス

ADDRESS と ADDRESS_LIST .....	B-2
プロトコルのパラメータ .....	B-2
お薦めするポート番号 .....	B-4
ポート番号の制限 .....	B-4

## C 構成パラメータ

構成ファイルの構文規則 .....	C-2
構成ファイルの構文規則の詳細 .....	C-2
ネットワーク・キャラクタ・セット .....	C-3
キャラクタ・セット .....	C-4
プロファイル・パラメータ (sqlnet.ora) .....	C-4
ローカル・ネーミング・パラメータ (tnsnames.ora) .....	C-31
一般構文 .....	C-31
複数の記述 .....	C-32
複数のアドレス・リスト .....	C-32
クライアント・ロード・バランスと DESCRIPTION_LIST の変更された機能 .....	C-33
例 .....	C-35
DESCRIPTIONS および DESCRIPTION_LIST .....	C-36
ADDRESS と ADDRESS_LIST .....	C-37
リストのオプション・パラメータ .....	C-38
CONNECT_DATA セクション .....	C-40
リスナー・パラメータ (listener.ora) .....	C-45
リスナー・アドレス・セクション .....	C-46
SID_LIST_listener_name 静的サービス・セクション .....	C-47
制御パラメータ .....	C-51
Oracle Names パラメータ (names.ora) .....	C-55
Oracle Connection Manager パラメータ (cman.ora) .....	C-65
プロトコル固有のパラメータ (protocol.ora) .....	C-73
ディレクトリ・サーバー・アクセス・パラメータ (ldap.ora) .....	C-74
リリース 8.1 での構成ファイルの変更点 .....	C-75
リリース 8.1 で廃止されたパラメータ .....	C-75
構成ファイルで使用される構文の変更点 .....	C-76
tnsnames.ora の変更点 .....	C-81
cman.ora の変更点 .....	C-83



D Net8 の LDAP スキーマ

構造化オブジェクト・クラス .....	D-2
orclDBServer .....	D-2
orclNetService .....	D-2
orclNetDescription .....	D-2
orclNetDescriptionList .....	D-3
orclNetAddress .....	D-3
orclNetAddressList .....	D-4
属性 .....	D-4

用語集

索引





1-1	基本的なクライアント / サーバー接続 .....	1-2
1-2	接続要求時のリスナー .....	1-3
1-3	Oracle Connection Manager を通した、クライアントとサーバーの接続 .....	1-4
1-4	ネーミング・メソッドを使用して接続識別子を解決するクライアント .....	1-7
2-1	代表的な Net8 接続におけるリスナー .....	2-4
2-2	MTS 接続モデル .....	2-7
2-3	ディスパッチャへの Direct Hand-Off .....	2-9
2-4	ディスパッチャにリダイレクトされた接続 .....	2-10
2-5	接続プーリング .....	2-11
2-6	専用サーバー・モデル .....	2-12
2-7	専用サーバー・プロセスへのリダイレクト接続 .....	2-14
2-8	事前生成済専用サーバーへの接続 .....	2-15
2-9	事前生成済専用サーバー・プロセスへのリダイレクト接続 .....	2-17
2-10	サービス登録 .....	2-20
2-11	ロード・バランス化環境 .....	2-25
2-12	ロード・バランス化の例 .....	2-27
2-13	ディレクトリを使用して接続識別子を解決するクライアント .....	2-34
2-14	ディレクトリでのデータベース・サービス .....	2-35
2-15	ドメイン・コンポーネント DIT .....	2-37
2-16	X.500 形式 DIT .....	2-37
2-17	アプリケーションを使用したディレクトリのエントリの作成 .....	2-38
2-18	相対的ネーミング .....	2-40
2-19	X.500 名前空間の絶対名 .....	2-42
2-20	ドメイン・コンポーネント名前空間の絶対名 .....	2-43
2-21	ディレクトリ・ユーザー・グループ .....	2-45
3-1	OSI 通信スタック .....	3-3
3-2	Oracle 環境での標準的な通信スタック .....	3-5
3-3	通常のクライアントと Net8 LDAP クライアントの通信スタック比較 .....	3-9
3-4	通常のクライアントと JDBC クライアントの通信スタック .....	3-11
3-5	JDBC 環境における Java クライアントとサーバーの通信スタック .....	3-13
3-6	通常の Net8 クライアントと Oracle8i JServer クライアントの通信スタック比較 .....	3-14
3-7	Oracle8i JServer 環境でのクライアント / サーバー通信スタック .....	3-15
4-1	Oracle Connection Manager による接続集中化 .....	4-6
4-2	Oracle Connection Manager による複数プロトコル・サポート .....	4-8
4-3	Oracle Names .....	4-10
4-4	シングル・ドメイン・ネーミング・モデル .....	4-14
4-5	階層ネーミング・モデル .....	4-15
4-6	委任管理リージョン .....	4-17
6-1	シングル・ドメインのエクスポート .....	6-33
6-2	類似 DIT へのルートおよびサブドメインのエクスポート .....	6-34
6-3	1 つの DIT ノードへのシングル・ドメインのエクスポート .....	6-36
6-4	異なる DIT へのルートおよびサブドメインのエクスポート .....	6-37
6-5	1 つの DIT ノードへのルートおよびサブドメインのエクスポート .....	6-39



6-6	2つのサブツリーへの複数ドメインのエクスポート .....	6-41
9-1	MTS アーキテクチャ .....	9-3
10-1	Net8 OPEN .....	10-2
12-1	ネットワーク製品とエラー・スタック・コンポーネント .....	12-14



---

---

# はじめに

『Net8 管理者ガイド』は、Net8 製品とそれに関連するアプリケーションの理解と使用に必要な情報を提供します。

---

---

**注意：** このガイドでは、Oracle8i および Oracle8i Enterprise Edition 製品に搭載されている機能および特徴について説明します。Oracle8i と Oracle8i Enterprise Edition は、同じ基本機能を備えています。ただし、いくつかの高度な Net8 機能は、Oracle8i Enterprise Edition でのみ利用できます。たとえば、Oracle Connection Manager を使用するには、Oracle8i Enterprise Edition が必要です。

---

---

内容は次のとおりです。

- [対象読者](#)
- [このガイドの編成](#)
- [Oracle8i の新機能](#)
- [関連マニュアル](#)
- [表記規則](#)



---

# 対象読者

このガイドの情報は、Oracle8i データベースなどのサービスへの接続やネットワーク構成を行う、ネットワークまたはデータベース管理者（DBA）を対象としています。さらに、Net8 の機能を理解したい人も対象としています。

## このガイドの編成

このガイドは、以下の部、章および付録に分かれています

---

### 第 I 部：Net8 の概要と概念

第 1 章「Net8 の概要」	Net8 の概要を示します。
第 2 章「Net8 の概念」	マルチスレッド・サーバー接続モデル、専用サーバー接続モデル、クライアント接続機能、リスナー機能、ネーミング・メソッドおよび LDAP 準拠ディレクトリ・サービスの Net8 サポートについて説明します。
第 3 章「Net8 アーキテクチャ」	Net8 のアーキテクチャを説明します。
第 4 章「Net8 製品およびツール」	Net8 製品および Net8 管理ツールを説明します。
第 5 章「ネットワークのプランニング」	Net8 を使用してネットワークを計画する際の考慮事項について説明します。さらに、Net8 製品の関係と、将来のネットワーク管理をよりよく行うためのオプションについて説明します。

---

### 第 II 部：構成

第 6 章「ネーミング・メソッドの構成」	Net8 の多様な方法を使用してネットワークを構成する方法を説明します。
第 7 章「リスナーの構成」	リスナーの構成方法を説明します。
第 8 章「Net8 の拡張機能」	高度な Net8 接続機能およびオプションの Net8 接続機能を構成する方法を説明します。これらの機能は、Oracle Connection Manager、接続プーリング、クライアント・ロード・バランス、接続時フェイルオーバー、異種サービス、外部プロシージャおよび Oracle Rdb データベースなどです。
第 9 章「マルチスレッド・サーバーの構成」	マルチスレッド・サーバー・アーキテクチャのコンポーネントをチューニングする方法を説明します。
第 10 章「プログラマのための Net8 拡張機能」	プログラマのための Net8 の拡張機能について説明します。この中には、Net8 Open、UNIX、シグナル処理および bequeath アダプタの復習が含まれています。

---

### 第 III 部：Net8 テストおよびトラブルシューティング

第 11 章「接続の確立とネットワークのテスト」	Net8 コンポーネントの起動方法、接続の確立方法、および接続のテスト方法を説明します。
--------------------------	--

---



---

第 12 章「トラブルシューティング」 Net8 をトラブルシューティングする手順について説明します。この中には、トレースとロギングに関する情報が含まれています。

---

## 第 IV 部：リファレンス

付録 A「制御ユーティリティ」	リスナー制御ユーティリティ（LSNRCTL）、Oracle Names 制御ユーティリティ（NAMECTL）、Oracle Connection Manager 制御ユーティリティ（CMCTL）など、Net8 制御ユーティリティのすべてのコマンドについて説明します。
付録 B「プロトコル・アドレス」	プロトコル・アドレスの構成方法を説明します。
付録 C「構成パラメータ」	ネットワーク構成ファイルの構成パラメータを表示して説明します。
付録 D「Net8 の LDAP スキーマ」	LDAP 準拠ディレクトリ・サービス・スキーマに格納されている Net8 オブジェクト・クラスおよび属性について説明します。

---

## Oracle8i の新機能

この項では、Oracle8i の新機能の一覧を簡単に示し、関連情報記載箇所へのポインタを提供します。新機能を次に示します。

- [ディレクトリ・ネーミング](#)
- [リスナーの新機能](#)
- [クライアント接続の新機能](#)
- [新製品](#)
- [Net8 に影響のある Oracle8i データベース機能](#)

### ディレクトリ・ネーミング

ネットワーク情報は、Oracle Internet Directory、Microsoft の Active Directory および Novell Directory Services などの LDAP 準拠ディレクトリ・サービスに一括して格納できるようになりました。

**関連項目：** 2-33 ページの「[Net8 および LDAP 準拠ディレクトリ・サーバー](#)」を参照。

### リスナーの新機能

機能	関連項目
サービス・ネーミング —Oracle8i データベース・サービスは、Oracle システム識別子（SID）ではなく、サービス名で識別されます。これは、接続記述子の定義方法に影響を与えます。	2-27 ページの「 <a href="#">SID ではなくサービス名によるデータベースの識別</a> 」



機能	関連項目
ダイレクト・ハンドオフ — リスナーには、クライアントにリダイレクト・メッセージを戻すことなく、要求を直接 MTS ディスパッチャに渡す機能があります。	2-19 ページの「 <a href="#">リスナーに対するサービス登録</a> 」
サービス登録 — Oracle8i データベース・インスタンスは、データベースの起動時にリスナーによって情報を登録します。これにより、サービス情報を使用して listener.ora ファイルを構成する必要がなくなります。	2-6 ページの「 <a href="#">マルチスレッド・サーバー・モデル</a> 」
接続時ロード・バランス化 — リスナーは、同じサービスに対する様々なインスタンスと MTS ディスパッチャ間のアクティブな接続の数を均衡化できます。	2-23 ページの「 <a href="#">接続時ロード・バランス化</a> 」

## クライアント接続の新機能

機能	詳細
クライアント・ロード・バランス — 複数のリスナーが1つのサービスをサポートしている場合、クライアントは様々なリスナーへの要求をランダム化できます。	2-18 ページの「 <a href="#">複数のリスナーに対するクライアント・ロード・バランス</a> 」
接続時フェイルオーバー — 複数のリスナーが1つのサービスをサポートしている場合、最初のリスナーが失敗すると、異なるリスナーへのクライアント要求をフェイルオーバーするようにクライアントを構成できます。	2-18 ページの「 <a href="#">複数のリスナーに対する接続時フェイルオーバー</a> 」

## Oracle Connection Manager の新機能

機能	詳細
Oracle Connection Manager 制御ユーティリティ (CMCTL) では、いくつかの新機能が拡張され、これにより制御能力が向上しました。	A-77 ページの「 <a href="#">Oracle Connection Manager 制御ユーティリティ (CMCTL)</a> 」



---

## 新製品

機能	詳細
Net8 Configuration Assistant— インストール後に基本的な構成作業を実行するためのツールです。インストールの終了後に、このツールによって自動的にデフォルトの構成ファイルが設定されます。	4-23 ページの「 <a href="#">Net8 Configuration Assistant</a> 」およびオペレーティング・システムのインストール・ガイド
Secure Sockets Layer (SSL) プロトコル付き TCP/IP—TCP/IP および Secure Sockets Layer (SSL) を使用して、ネットワーク上でクライアント / サーバー認証を行うためのプロトコルです。	4-2 ページの「 <a href="#">SSL 付き TCP/IP</a> 」

## Net8 に影響のある Oracle8i データベース機能

機能	詳細
データベース・サービスは、システム識別子 (SID) ではなく、サービス名で識別されます。	2-27 ページの「 <a href="#">SID ではなくサービス名によるデータベースの識別</a> 」

## 関連マニュアル

- 『Oracle8i 概要』
- 『Oracle8i リファレンス・マニュアル』
- 『Oracle8i 分散システム』
- 『Oracle8i パフォーマンスのための設計およびチューニング』
- 『Oracle Enterprise Manager 管理者ガイド』
- 『Oracle8i Advanced Security 管理者ガイド』



# 表記規則

このガイドでは次の表記規則を使用しています。

表記規則	意味
大文字	SQL コマンド、キーワードおよび初期化パラメータを示します。
太字	太字は、用語集で定義されている用語を示します。
固定幅フォント の小文字	ファイル名、ディレクトリおよびコマンドを示します。
固定幅文字	この字体は、ユーザー入力およびコード例で使用されます。
イタリック体	イタリック体の用語は、変数を示します。
UNIX では \$ORACLE_ HOME、Windows で は ORACLE_HOME	この Optimal Flexible Architecture (OFA) 互換のリリースでは、すべてのサブディレクトリを最高位の ORACLE_HOME ディレクトリ以下に配置する必要はありません。複数の Oracle ホーム・ディレクトリを含めた、ORACLE_BASE ディレクトリを設定できます。
< >	山カッコで囲む対象は、ユーザーが指定する名前です。
[ ]	大カッコで囲む対象は、ユーザーの選択元となる 1 つまたは複数のオプション項目の選択肢です。
{ }	中カッコで囲む対象は、必須項目です。
...	コード・サンプルにある水平方向の省略記号は、サンプルの一部を省略していることを意味します。



# 第 I 部

---

## Net8 の概要と概念

第 I 部では、Net8 の概念、製品およびツールの概要を示します。次の章があります。

- 第 1 章「Net8 の概要」
- 第 2 章「Net8 の概念」
- 第 3 章「Net8 アーキテクチャ」
- 第 4 章「Net8 製品およびツール」
- 第 5 章「ネットワークのプランニング」







---

# Net8 の概要

この章では、Net8 について紹介し、その主なアプリケーション、機能および機能性の概要を示します。内容は次のとおりです。

- [Net8 の概要](#)
- [Oracle の接続性の概要](#)
- [構成の概要](#)

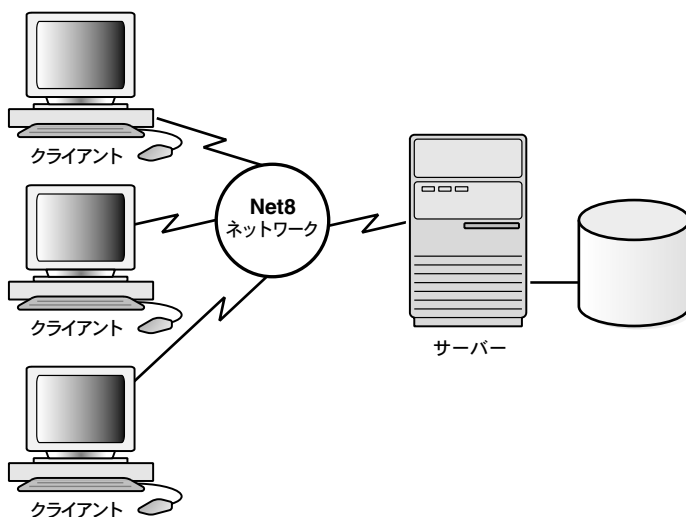


## Net8 の概要

Net8 を使用すると、サービスおよびそのアプリケーションを様々なコンピュータに置いて、ピア・アプリケーションとして通信することが可能になります。Net8 の主要な機能は、ネットワーク・セッションを確立し、クライアント・マシンとサーバー間、または 2 つのサーバー間でデータを転送することです。Net8 はネットワークの各マシンに配置されます。ネットワーク・セッションが確立されると、Net8 はクライアントやサーバーへのデータの案内役として動作します。

図 1-1 は、クライアント / サーバー接続の様子を示しています。

図 1-1 基本的なクライアント / サーバー接続



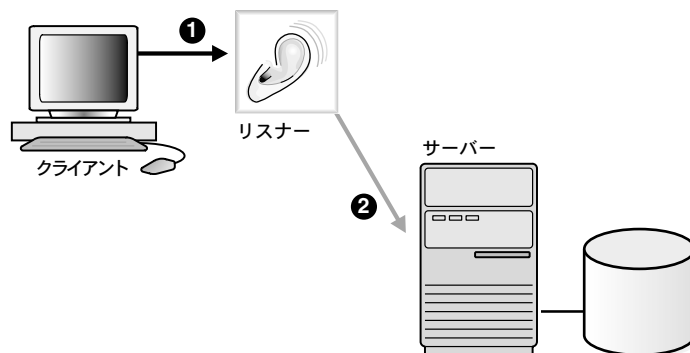
ネットワーク・セッションは、**リスナー (listener)** の支援を受けて確立されます。リスナーは、サーバー上に存在する別のプロセスです。リスナーは、クライアントの着信要求を受信して、それをサーバーに渡します。

リスナーは、クライアント要求を受け取って、それをサーバーに渡します。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のネットワーク・セッションを要求するたびに、リスナーが実際の要求を受信します。

図 1-2 は、接続要求時のリスナーの様子を示しています。



図 1-2 接続要求時のリスナー



同じサービスに多数の接続が必要な環境の場合、Net8 では、**Oracle Connection Manager** と呼ばれるルーティング・プロセスが提供されます。このプロセスは、通常、クライアントおよびサーバー以外のマシンに存在します。

Net8 では、次のような環境で接続ルーティングを行います。

- 多数のユーザーが単一プロトコルを通して単一のサーバーにアクセスする必要がある環境
- クライアントとサーバーに異なるプロトコルがインストールされ、一般的な接続が失敗する環境

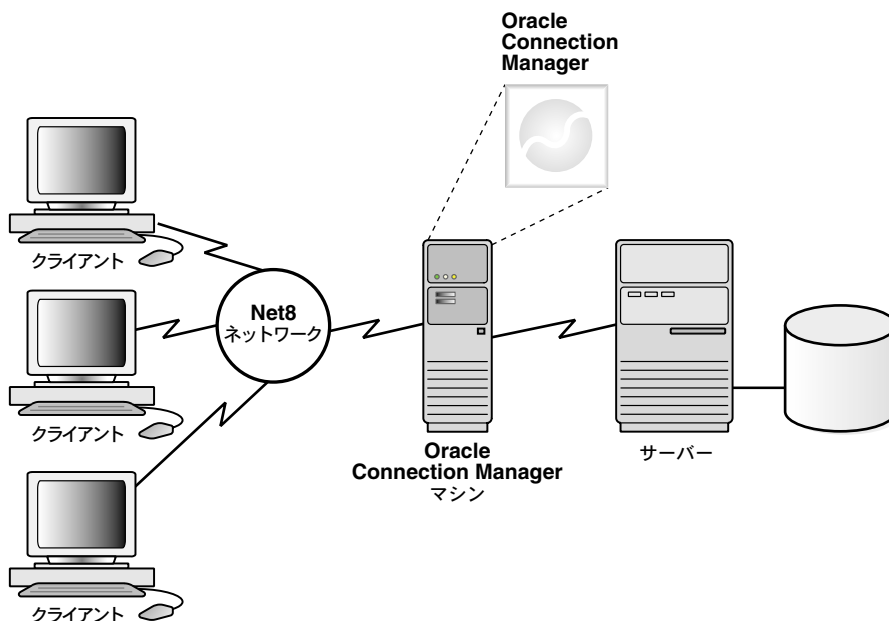
標準的な Net8 接続では、クライアントとサーバーに同じプロトコルがインストールされている必要があります。

- TCP/IP 環境で、指定したサーバーへのクライアント・アクセスの制御が必要な環境

図 1-3 は、クライアント接続が、ネットワーク中の別のマシンに存在する Oracle Connection Manager にルーティングされる様子を示したものです。



図 1-3 Oracle Connection Manager を通した、クライアントとサーバーの接続



**関連項目：** 製品情報は、4-4 ページの「[Oracle Connection Manager](#)」を参照してください。

## Oracle の接続性の概要

Net8 では、Oracle データベースや Oracle 以外のデータベースなどの様々なサービスに接続できます。Oracle データベースは、Oracle 環境で最も一般的なサービスなので、この項ではデータベース・サービスを中心に説明します。

ユーザーがネットワークを介してデータベース・サービスに接続する場合、宛先サービスに関するネットワーク情報が記述されている[接続記述子 \(connect descriptor\)](#) がリスナーに渡されます。

接続記述子の種類は、次のとおりです。

- **プロトコル・アドレス (protocol address)** によるリスナーの位置情報など、サービスへのネットワーク・ルート。
- 通常は**グローバル・データベース名 (global database name)** であるサービス名。データベース名およびデータベース・ドメインから構成され、データベース・サービスの識別に使用します。



完全な接続記述子を使用して接続すると、次の `sales.us.acme.com` という名前のデータベース・サービスへの接続例に示すように、冗長な**接続文字列 (connect string)** が作成されます。

```
CONNECT
scott/tiger@(description=(address=(protocol=tcp)(host=sales-server)(port=1521))
(connect_data=(service_name=sales.us.acme.com)))
```

冗長な接続文字列を回避するために、接続記述子を**接続識別子 (connect identifier)** にマップすることができます。接続識別子は、**ネット・サービス名 (net service name)** または実際のサービス名のどちらでも構いません。次にこの情報は、少なくとも1つの**ネーミング・メソッド (naming method)** に格納されます。クライアントに必要なことは、接続文字列に接続識別子を使用することのみです。たとえば、`sales` という名前のネット・サービス名が前述の例に使用されている接続記述子にマップされている場合、クライアントは次の接続文字列を使用できます。

```
CONNECT scott/tiger@sales
```

接続要求時に、クライアントはネーミング・メソッドに接続して、`sales` を接続記述子に解決します。クライアントは、次に、接続記述子で指定したリスナー・アドレスに接続要求を転送します。

リスナーは、プロトコルを通してクライアントの接続を受け付けます。リスナーは、クライアントの情報を、データベースから受信した情報および構成ファイル `listener.ora` に格納した情報と比較します。この情報が一致すると、接続が許可されます。

## 構成の概要

接続のためのネットワーク構成は、次の項で説明する主な2つの作業から構成されています。

- **ネーミング・メソッドの構成**
- **サーバー上でのリスナーの構成**

## ネーミング・メソッドの構成

ネーミング・メソッドは、接続識別子を接続記述子にマップするために、必ず構成する必要があります。

Net8 は、次に示す複数のカテゴリのネーミング・メソッドをサポートしています。

- **ローカル・ネーミング**

ネット・サービス名は、ネットワーク上の各クライアントにあるローカル構成ファイル `tnsnames.ora` に格納できます。`tnsnames.ora` ファイルは各クライアントで構成できるので、特定のクライアントの必要性に応じてチューニングできます。



- ディレクトリ・ネーミング

サービス・アドレスおよびネット・サービス名は、Lightweight Directory Access Protocol (LDAP) 準拠のディレクトリ・サーバーに格納できます。

- Oracle Names

サービス・アドレスおよびネット・サービス名は、Oracle Names サーバーのシステムに格納できます。Oracle Names は、Oracle 固有のネーム・サービスで、サービス・アドレスの一元的な格納をメンテナンスするサービスです。

- ホスト・ネーミング

サービス名は、既存の IP アドレス解決メカニズムへ格納できます。

- 外部ネーミング

使用中の環境でサードパーティ製のネーミング・サービスがすでに構成されている場合、Net8 では、追加構成を回避するために、その環境へのサービス・アドレス情報の格納がサポートされます。

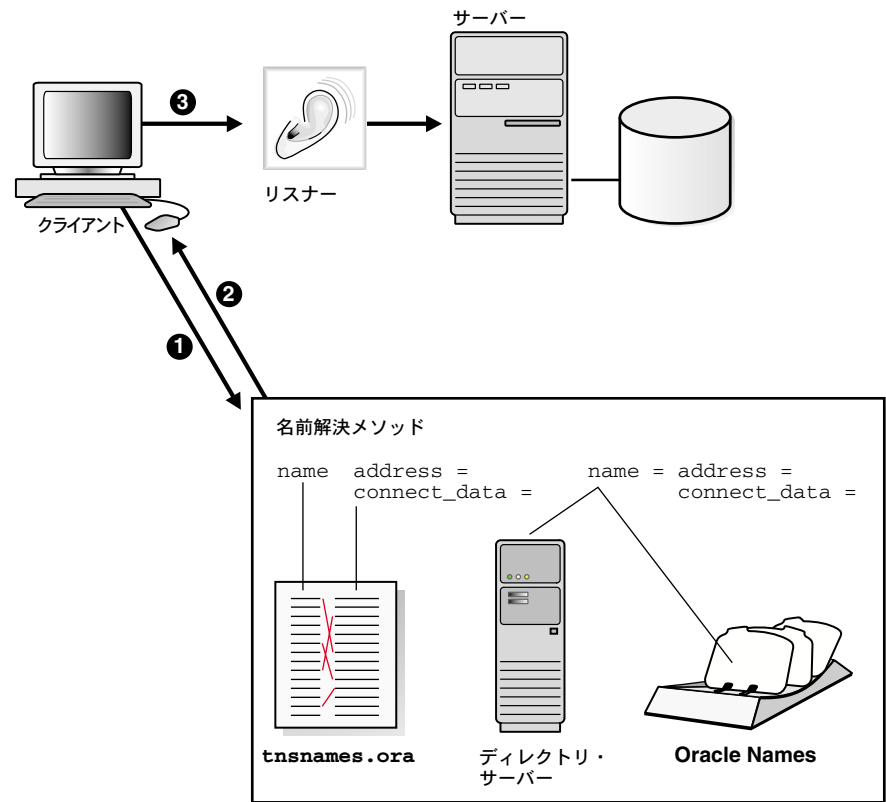
ネーミング・メソッドが構成されると、クライアントがそのネーミング・メソッドにアクセスできるようにする必要があります。

図 1-4 は、tnsnames.ora ファイル、ディレクトリ・サーバーおよび Oracle Names サーバーを介して接続識別子を解決するクライアントを示しています。

1. クライアントは、接続識別子を接続記述子に解決するために、ネーミング・メソッドに接続します。
2. 接続識別子は、ネーミング・メソッドによって接続記述子に解決されます。
3. クライアントは、この接続記述子を使用して、要求をリスナーのアドレスに転送します。



図 1-4 ネーミング・メソッドを使用して接続識別子を解決するクライアント



**関連項目：** 構成の詳細は、[第6章](#)を参照してください。

## サーバー上でのリスナーの構成

リスナーは、1つまたは複数のリスニング・プロトコル・アドレスおよび宛先サービスに関する情報を使用して構成されます。

プロトコル・アドレスは、リスナー構成ファイル `listener.ora` で構成されます。サービス情報は、`listener.ora` ファイルで構成される場合とそうでない場合があります。

- Oracle8i データベースでは、リスナーによって特定の情報が自動的に登録されます。たとえば、サービス名、インスタンス名、サービス・ハンドラおよびロード情報などです。サービス登録と呼ばれるこの機能では、`listener.ora` ファイルでの構成は不要です。



- この他のサービス、たとえば Oracle7 データベースや Oracle8 リリース 8.0 データベース、Oracle Enterprise Manager 管理ツールなどでは、`listener.ora` ファイルでのサービス構成が必要です。

**関連項目：** 構成の詳細は、[第 7 章](#)を参照してください。



---

## Net8 の概念

この章では、サーバーへの接続時に、クライアントが Net8 を使用方法の概要を説明します。この章の内容は次のとおりです。

- ネットワーク環境
- 接続操作
- 例外操作
- Net8 接続モデル
- クライアント接続オプション
- リスナーに対するサービス登録
- 接続時ロード・バランス化
- SID ではなくサービス名によるデータベースの識別
- ネーミング・メソッドによるサービスの解決
- Net8 および LDAP 準拠ディレクトリ・サーバー



## ネットワーク環境

Oracle ネットワーク環境は、次の2つの概念に基づいています。

- [分散処理](#)
- [分散データベース](#)

### 分散処理

Oracle データベースとクライアント・アプリケーションは、分散処理環境として知られている環境で動作します。分散処理または協同処理は、1回のデータ・トランザクションを完了するために、複数のコンピュータが関与します。Oracle Tool などのアプリケーションは、特定の操作を行うためのデータを必要とするクライアントの役割を果たします。データベース・サーバーは、データの格納や提供を行います。

代表的なネットワーク構成では、クライアントとサーバーが別の物理マシン上の論理エンティティとして存在する場合があります。この構成では、負荷の分割が可能で、クライアント・ワークステーションとサーバー・マシンとの間で、リソースが効果的に割り当てられます。通常、クライアントはユーザーが使用するアプリケーションの実行に適したメモリーを搭載したデスクトップ・コンピュータ上に存在するのに対し、サーバーにはデータベースを実行、管理するための、多くのメモリー、ディスク・ストレージおよび処理能力を備えています。

### 分散データベース

このタイプのクライアント / サーバー・アーキテクチャでは、ネットワーク間にデータベースを分散できます。分散データベースは、ユーザーに対しては1つの論理データベースとして表示される、複数のコンピュータ上に格納されたデータベースのネットワークです。分散データベース・サーバーは、データベース・リンク、またはあるデータベースから別のデータベースへのパスで接続されます。1つのサーバーがデータベース・リンクを使用して、第2サーバーにある情報を必要に応じて問合せや変更を行うことで、第2サーバーに対するクライアントの役割を果たします。

### 接続操作

Net8 は、2種類の接続操作をサポートします。

- [サーバーへの接続](#)
- [サーバーからの切断](#)



## サーバーへの接続

ユーザーは、接続するサービスの識別子とともにユーザー名およびパスワードを渡すことによって接続要求を開始します。**接続識別子 (connect identifier)** と呼ばれる識別子によって識別される対象は、次のとおりです。

- 宛先サービス
- 宛先サービスにアクセスするためのパスまたはネットワーク・ルート

接続識別子は、複数の方法で指定されます。最も一般的な方法の1つは、このサービスの別名である**ネット・サービス名 (net service name)** を使用する方法です。この方法では、**接続記述子 (connect descriptor)** へのマッピングが行われます。接続記述子には、宛先サービスおよびネットワーク・ルートの情報が含まれています。宛先サービスは、Oracle8i データベースの場合はその**サービス名 (service name)**、Oracle8 リリース 8.0 または Oracle7 の場合はその **Oracle システム識別子 (Oracle System Identifier: SID)** を使用して表示します。ネットワーク・ルートは、少なくとも、ネットワーク・アドレスによってリスナーの位置を提供します。

次に示すネット・サービスは、sales と呼ばれ、接続記述子にマップされます。

```
sales=
(description=
  (address=(protocol=tcp) (host=sales-server) (port=1521))
  (connect_data=
    (service_name=sales.us.acme.com))
```

sales-server 上のリスナーは、ポート 1521 の TCP/IP を使用して、sales.us.acme.com と呼ばれるデータベース・サービスの接続要求をリスニングします。

クライアントは、接続文字列に完全な接続記述子または接続識別子を使用できます。次の例では、完全な接続記述子を使用する接続文字列と、sales というネット・サービス名を使用する別の接続文字列を示しています。

```
CONNECT
scott/tiger@(description=(address=(protocol=tcp) (host=sales-server1) (port=1521))
(connect_data=(service_name=sales.us.acme.com))
```

```
CONNECT scott/tiger@sales
```

ネット・サービス名を使用すると、sales から接続記述子への最初のマッピングによって、接続処理が実行されます。接続記述子は、リスナーの検索に使用され、その後、データベースを検索するためにリスナーに渡されます。クライアントの要求に対して最適なサービスを提供できる**サービス・ハンドラ (service handler)** の種類を判別することは、リスナーのジョブになります。サービス・ハンドラは、マルチスレッド・サーバー・**ディスパッチャ**



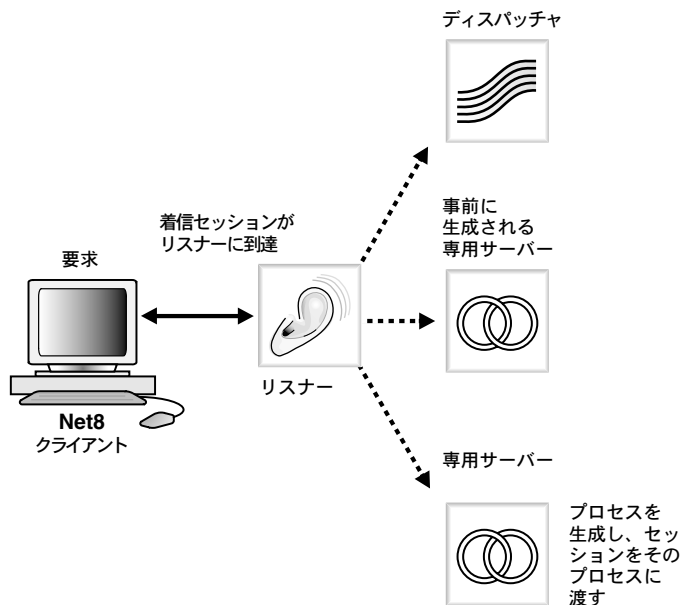
(dispatcher)、専用サーバー (dedicated server) または事前生成済専用サーバー (prespawnd dedicated server) のいずれでも構いません。

**関連項目：**

- サービス名および SID の概要は、2-27 ページの「[SID ではなくサービス名によるデータベースの識別](#)」を参照してください。
- サービス・ハンドラは、2-6 ページの「[Net8 接続モデル](#)」を参照してください。

図 2-1 は、代表的な Net8 接続におけるリスナーの役割を示したものです。

**図 2-1 代表的な Net8 接続におけるリスナー**



リスナーは、構成内容に基づいて、次のいずれかの処理を実行します。

- 要求を直接ディスパッチャに渡します。
- ディスパッチャまたは専用サーバー・プロセスの位置情報が記録されたリダイレクト・メッセージをクライアントに戻します。続いてクライアントが、ディスパッチャまたは専用サーバー・プロセスに接続します。



- 専用サーバー・プロセスを生成して、クライアント接続を専用サーバー・プロセスに渡します。

## サーバーからの切断

サーバーからの切断要求は、次の項で説明する方法で開始できます。

- [ユーザーによる切断](#)
- [追加接続要求](#)
- [接続の異常終了](#)
- [タイマーによる切断またはデッド接続検出](#)

### ユーザーによる切断

ユーザーは、クライアント / サーバーのトランザクションが終了したときに、サーバーからの接続の切断を要求できます。サーバーはすべてのサーバー / サーバー間データ転送が終了し、リンクを保持する必要がない場合に、2 番目のサーバーから切断できます。

### 追加接続要求

クライアント・アプリケーションがサーバーに接続されているときに、同じサーバーまたは他のサーバー上の別のユーザー・アカウントにアクセスを要求すると、大半の Oracle Tools は、現在接続されているサーバーからアプリケーションを切断します。接続の切断が終了すると、対応するサーバー上の新規ユーザー・アカウントに対する接続要求が開始されます。

### 接続の異常終了

その他のコンポーネントでは、Net8 に通知せずに通信を切断または異常終了する場合があります。この場合、Net8 は次のデータ処理中に障害を認識し、クライアントとサーバーの各処理をクリーン・アップして、現行の処理を効率よく切断します。

### タイマーによる切断またはデッド接続検出

タイマーによる切断またはデッド接続検出を利用すると、Net8 は、クライアントの異常終了によって中断したままになっている接続を識別できます。この機能は、無効になった接続によるリソースの無駄を最小限に抑えます。また、ブレイクされた接続のユーザーが保持しているコミットされていないトランザクションおよびロックが自動的にロールバックされます。

デッド接続検出を有効にした接続では、ユーザーが定義した間隔（通常は数分）で、サーバーからクライアントに小さなプローブ・パケットが送られます。接続が無効な場合は、送信操作によってエラーが発生し、サーバー・プロセスによって接続が終了します。



## 例外操作

Net8 では、3 種類の例外操作をサポートします。

- 接続を通じてブレークを発行します
- ブレーク後の同期化のために接続をリセットします
- 着信ブレークのために接続状態をテストします

ユーザーは、これら 3 つの操作のうち 1 つのみ、すなわち、ブレークの発行を制御します。ユーザーが割込みキー（マシンによっては [Ctrl] を押しながら [C]）を押すと、アプリケーションはこの操作を呼び出します。また、SQL\*Loader を使用して無効なデータ行のロードを試みた場合など、異常な操作が行われると、データベースはクライアントに対してブレークを発行します。

## Net8 接続モデル

クライアント / サーバー間の接続は、次に示す異なる 2 つのモデルを使用して確立します。

- **マルチスレッド・サーバー・モデル** — リスナーは、接続をディスパッチャに渡すか、またはクライアントをリダイレクトしてディスパッチャを通じて接続します。
- **専用サーバー・モデル** — リスナーは、専用サーバーを起動して接続を専用サーバーに渡すか、またはクライアントを専用サーバーにリダイレクトします。

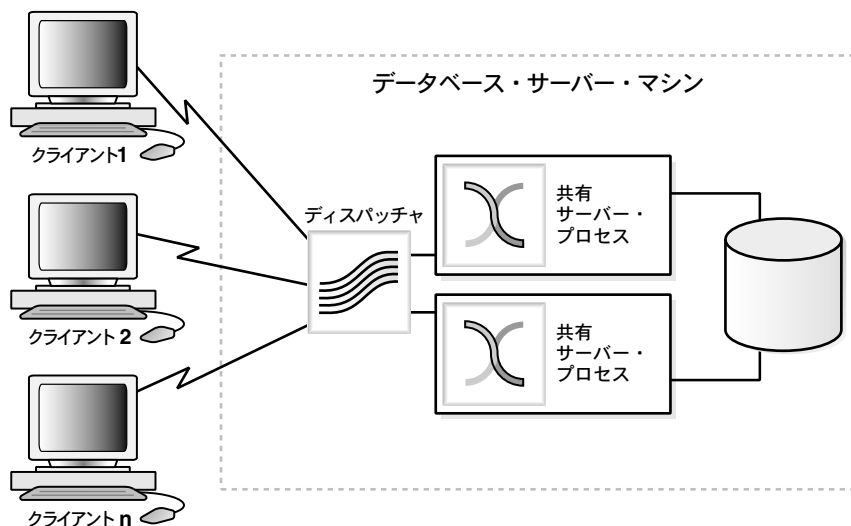
## マルチスレッド・サーバー・モデル

**マルチスレッド・サーバー (multi-threaded server: MTS)** 構成では、クライアント・ユーザー・プロセスがディスパッチャに接続します。**ディスパッチャ (dispatcher)** は、複数のクライアント接続を同時にサポートできます。各クライアント接続は、**仮想サーキット (virtual circuit)** にバインドされます。仮想サーキットは、ディスパッチャが使用する共有メモリーの 1 つで、クライアント・データベースの接続要求および応答を目的としています。ディスパッチャは、要求が到着すると、仮想サーキットを共通キューに置きます。アイドル状態の共有サーバーは、共通キューから仮想サーキットを取り出して要求を処理し、その仮想サーキットが共通キューから別の仮想サーキットを取り出そうとする前に、取り出した仮想サーキットを放棄します。この方法によって、サーバー・プロセスの小規模プールが大量のクライアントを処理できるようになります。専用サーバー・モデルに比した MTS モデルのきわめて有利な点は、システム・リソースの使用が削減されることです。これにより、サポート可能なユーザー数を増加できます。

図 2-2 は、1 つのディスパッチャに送られているクライアント接続要求を示しています。



図 2-2 MTS 接続モデル



データベースが MTS で構成されていると、ネットワーク・セッションが専用サーバーを特に要求している場合、またはディスパッチャが使用できない場合を除き、着信ネットワーク接続要求は必ずディスパッチャに送られます。ディスパッチャで発生するイベントの順序は、次のとおりです。

1. リスナーを起動すると、リスナーは、デフォルトのアドレスか、またはリスナーの構成ファイル listener.ora で指定したアドレスでリスニングを行います。
2. データベース**インスタンス (instance)** が起動します。ディスパッチャは、**初期化パラメータ・ファイル (initialization parameter file)** 内の構成パラメータに従って起動します。各ディスパッチャは、それぞれに割り当てられたアドレスでリスニングを行います。
3. 各ディスパッチャのアドレスが**サービス登録 (service registration)** によってリスナーに登録されます。これには、ディスパッチャがリスナーに登録される前に最大1分ほどかかります。

リスナーがデフォルトのアドレスでリスニングしない場合は、リスナーのネットワーク名を初期化パラメータ・ファイル内で指定できます。複数のリスナーを使用している場合、名前は、このような複数のアドレスに解決できます。

ステップ2をステップ1の前に実行した場合は、サーバーがリスナーに接続しようとするとディレイが発生することがあります。



これで、リスナーとディスパッチャは、着信ネットワーク・セッションを受信できる状態になりました。

---

**注意：**リスナー制御ユーティリティから SERVICES コマンドを発行すると、リスナーに登録されたディスパッチャ・アドレスをチェックできます。詳細は、A-9 ページの「[SERVICES](#)」を参照してください。

---

4. クライアントは、接続記述子のリスナー・アドレスを使用して、リスナーに接続します。

ディスパッチャ登録のない時間枠に接続要求が到着すると、これらの要求は (listener.ora ファイルで構成された) 専用サーバーによって処理されます。あるクライアントに対して必ずディスパッチャを使用させる必要がある場合は、次の接続記述子の例に示すように、接続記述子の接続データ部分に (SERVER=SHARED) を指定することによってこのクライアントを構成することができます。この構成を使用すると、使用できる共有サーバーが存在しない場合、クライアント接続要求は失敗します。

```
sales=
(description=
  (address=(protocol=tcp) (host=sales1-server) (port=1521))
  (connect_data=
    (service_name=sales.us.acme.com)
    (server=shared)))
```

**関連項目：** クライアントでの MTS の構成については、9-9 ページの「[クライアントでの MTS の使用](#)」を参照してください。

5. リスナーは接続要求を受け取り、次に、クライアントの要求を処理できるかどうかを判別します。リスナーは、ネットワーク接続要求を処理できない場合、この要求を拒否した後、ステップ 8 から再開します。
6. クライアントの要求が有効な場合、リスナーは次のいずれかを実行します。
  - リスナーが接続要求を直接ディスパッチャに渡す。
  - リスナーがクライアントに、ディスパッチャのネットワーク・アドレスを含むリダイレクト・メッセージを発行する。次に、クライアントは、リスナーに要求したネットワーク・セッションを取り消し、リダイレクト・メッセージで提供されたネットワーク・アドレスを使用して、ディスパッチャとのネットワーク・セッションを確立する。

ディスパッチャへの Direct Hand-Off がサポートされるのは、ディスパッチャがリスナーに対してローカルで次のプロトコルの場合です。

- IPC プロトコル
- TCP/IP プロトコル



ディスパッチャがリモートの場合は、リダイレクト接続を使用します。

7. ディスパッチャは、重要なロード情報によってリスナーを更新します。これにより、リスナーは、ディスパッチャ間の着信ネットワーク接続要求のバランスをとることができます。
8. リスナーは、着信ネットワーク・セッションのリスニングを再開します。

図 2-3 は、リスナーが接続要求を直接ディスパッチャに渡すことを示しています。

1. リスナーがクライアント接続要求を受け取ります。
2. リスナーは、接続要求を直接ディスパッチャに渡します。
3. クライアントがディスパッチャに接続します。

図 2-3 ディスパッチャへの Direct Hand-Off

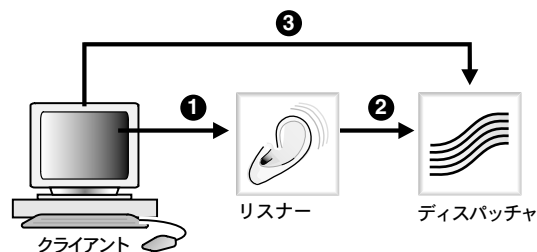
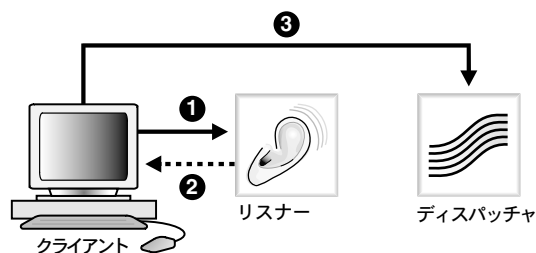


図 2-4 は、リダイレクト接続におけるディスパッチャの役割を示しています。

1. リスナーがクライアント接続要求を受け取ります。
2. リスナーは、ディスパッチャの位置をリダイレクト・メッセージでクライアントに通知します。
3. クライアントがディスパッチャに直接接続します。



図 2-4 ディスパッチャにリダイレクトされた接続



## 構成

MTS を使用可能にするには、MTS\_DISPATCHERS パラメータを初期パラメータ・ファイルに設定する必要があります。

**関連項目：** MTS\_DISPATCHERS パラメータの構成については、[第 9 章](#)を参照してください。

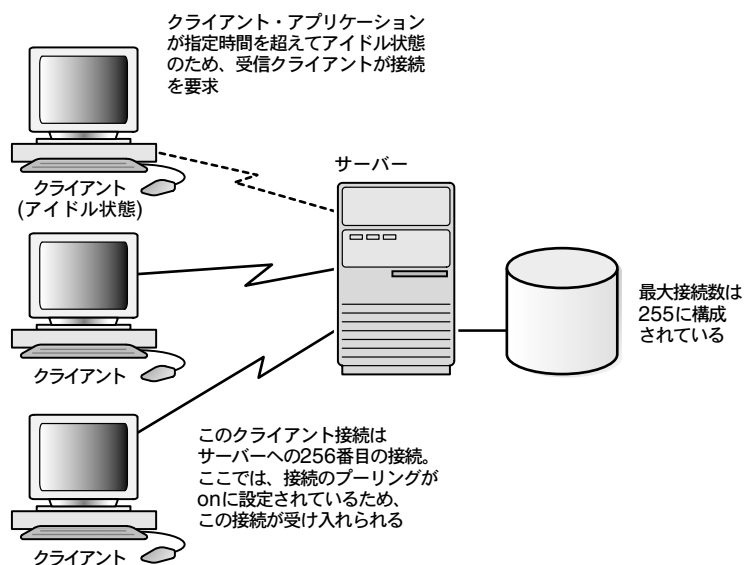
## 接続プーリング

接続プーリングは、MTS のリソース利用機能およびユーザー・スケーラビリティ機能で、これを利用すると、ディスパッチャに対する接続の数が制限された状態でも、ネットワーク・セッションの数を最大にすることができます。これは接続を共有またはプーリングすることによって行われます。

[図 2-5](#) は、接続数を 255 に設定した場合の接続プーリングの動作を示しています。



図 2-5 接続プーリング



タイムアウト・メカニズムを使用して、指定された期間アイドル状態であったネットワーク接続を一時的に解放することによって、接続プーリングは、アイドル状態のネットワーク接続とのネットワーク・セッションを維持しながら、これらのネットワーク接続を着信クライアントに使用できるようにします。アイドル状態のクライアントに実行対象の作業が残っている場合、ネットワーク接続は、ディスパッチャによって再確立されます。

**関連項目：** 構成に関する情報は、9-8 ページの「[接続プーリングの使用](#)」を参照してください。

## 専用サーバー・モデル

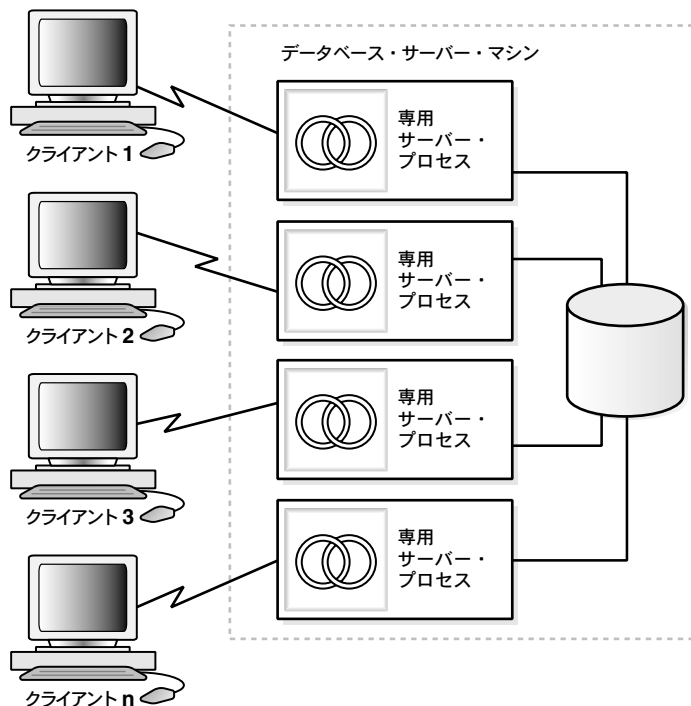
Net8 は、MTS のかわりに、クライアント接続要求を**専用サーバー (dedicated server)** 宛てに送ることができます。このモデルでは、クライアントごとに 1 つのサーバー・プロセスが存在します。クライアントを専用サーバーに接続するためには、リスナーおよび**インスタンス (instance)** は、必ず同じマシン上で稼動している必要があります。

**注意：** 専用サーバー・プロセスでは、MTS よりも大量のメモリが必要になります。



図 2-6 は、専用サーバーに送られているクライアント要求を示しています。

図 2-6 専用サーバー・モデル



次の例に示すように、接続記述子の接続データ部分にある (SERVER=DEDICATED) を設定すれば、MTS の構成をオーバーライドするようにクライアントを構成できます。

```
sales=
(description=
  (address=(protocol=tcp) (host=sales1-server) (port=1521))
  (connect_data=
    (service_name=sales.us.acme.com)
    (server=dedicated)))
```

**関連項目：** クライアントでの MTS 構成のオーバーライドは、9-9 ページの「[クライアントでの MTS の変更](#)」を参照してください。



専用サーバーとの間に確立される接続の順序は、次のとおりです。

1. リスナーを起動すると、リスナーは、デフォルトのアドレスか、またはリスナーの構成ファイル `listener.ora` で指定したアドレスでリスニングを行います。
2. データベース・インスタンスが起動します。
3. リスナーは、このインスタンスの情報で更新されます。

これで、リスナーは、着信ネットワーク接続を受け取る準備ができました。

---

**注意：**リスナー制御ユーティリティから `SERVICES` コマンドを発行すると、リスナーに登録された専用サーバー・アドレスをチェックできます。詳細は、A-9 ページの「[SERVICES](#)」を参照してください。

---

4. クライアントは、接続記述子のリスナー・プロトコル・アドレスを使用して、リスナーに接続します。
5. リスナーは接続要求を受け取り、次に接続のハンドシェイクを実行して、クライアントの要求を処理できるかどうかを判別します。リスナーは、ネットワーク接続要求を処理できない場合、その要求を拒否した後、ステップ7から再開します。
6. リスナーは専用サーバーを生成して、次のいずれかの処理を実行します。
  - リスナーが、生成した専用サーバーにクライアント接続を渡す。
  - 専用サーバーが、リスニング対象となっているアドレスのリスナーに通知する。リスナーが、クライアントに戻すリダイレクト・メッセージを発行して、接続を取り消す。クライアントが、リダイレクト・メッセージに記述されたアドレスを使用して、専用サーバーに直接接続する。

どちらが選択されるかは、オペレーティング・システムおよび使用中のプロトコルによって決まります。

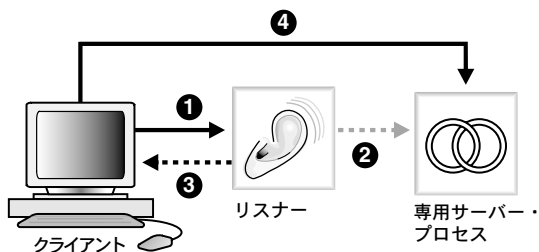
7. リスナーは、着信ネットワーク・セッションのリスニングを再開します。



図 2-7 は、リダイレクト接続における専用サーバーの役割を示しています。

1. リスナーがクライアント接続要求を受け取ります。
2. リスナーが専用サーバーを生成します。
3. リスナーが専用サーバーの位置をリダイレクト・メッセージでクライアントに通知します。
4. クライアントが専用サーバーに直接接続します。

図 2-7 専用サーバー・プロセスへのリダイレクト接続



## 事前生成済専用サーバー

Net8 には、接続要求を受け取る前に、専用サーバー・プロセスを自動的に作成するオプションがあります。これらのプロセスはリスナーが稼働している間存在しており、それ以降の接続要求で再使用できます。

---

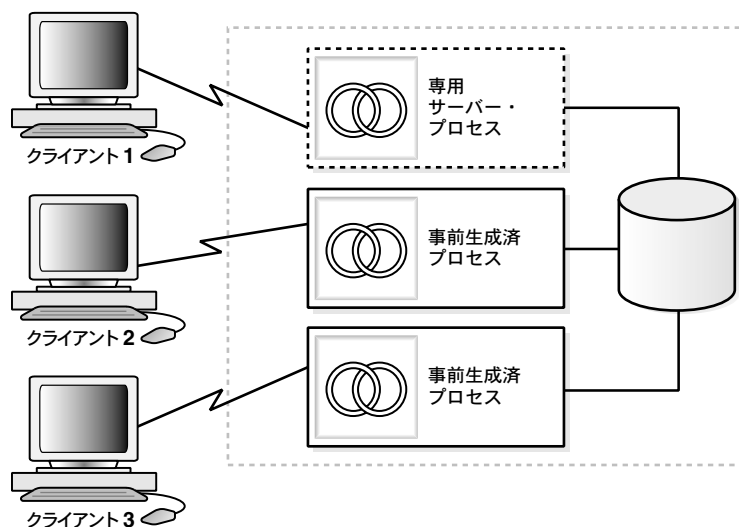
**注意：** 事前生成済専用サーバーには、SQL\*Net バージョン 2.1 以降、および Oracle Server リリース 7.1 以降が必要です。

---

図 2-8 は、事前生成済専用サーバーに送られているクライアント要求を示しています。



図 2-8 事前生成済専用サーバーへの接続



事前生成済専用サーバー・プロセスを使用してクライアント接続要求を処理するときが発生するイベントの順序は、次のとおりです。

1. リスナーを起動すると、リスナーは、その構成ファイル `listener.ora` で指定されたアドレスでリスニングを行います。
2. リスナーは、その構成ファイルで定義されているプール・サイズに達するまで、一連の専用サーバーを生成します。
3. 生成された各サーバーは、部分アドレスのリスニングを実行して、リスニング対象の部分アドレスをリスナーに通知します。部分アドレスのリスニングでは、サーバー・プロセスはリスニングを行います、その基礎となるプロトコル・スタックに対して、リスニング対象の特定のアドレスについて優先のないことを通知します。このため、多くのプロトコル・スタックは空いているリスニング・アドレスを選択し、要求側のサーバー・プロセスにそのアドレスを自動的に割り当てます。リスナーはこの時点で、事前生成済サーバーすべてをアイドルとしてマークします。
4. クライアントが接続要求をリスナーに送ります。
5. リスナーは、ネットワーク接続要求を受け取って、クライアントの要求を処理できるかどうかを判別します。リスナーは、ネットワーク接続を処理できない場合、その接続を拒否した後、ステップ9から再開します。
6. リスナーは、事前生成済サーバーのいずれかの完全なネットワーク・アドレスを含むリダイレクト・メッセージをクライアントに発行します。リスナーは、そのサーバー・プロセスをアクティブとして記録します。



7. クライアントは、リスナーに要求したネットワーク接続を取り消し、リダイレクト・メッセージで提供されたアドレスを使用して、事前生成済専用サーバーとのネットワーク接続を確立します。
8. リスナーは、別のサーバー・プロセスを生成して、アクティブ状態の事前生成済専用サーバーと置き換えます。ただし、`listener.ora` ファイルにある `PRESPAWN_MAX` の値が（アクティブ状態またはアイドル状態の）既存の事前生成済専用サーバー・プロセス数の合計よりも大きく、かつ `POOL_SIZE`（アイドル状態の事前生成済プロセスの数）が達していないことを前提とします。
9. リスナーは、着信ネットワーク・セッションのリスニングを続行します。

前述の一連のイベントは、事前に生成された制限の最大値に到達するまで続行されます。リスナーは、この時点で新規の専用サーバー・プロセスの事前生成を停止します。

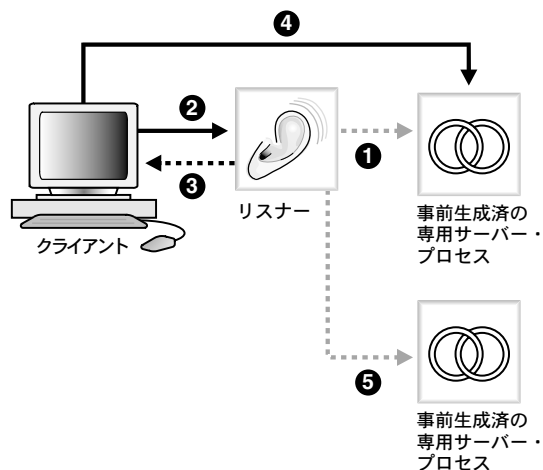
クライアントが接続を切断すると、そのクライアントに関連付けられている事前生成済専用サーバー・プロセスは、アイドル・プールに戻って、別のクライアントに割り当てられるのを指定された時間待機します。事前生成済サーバーにクライアントが渡されずにタイムアウトが満了すると、事前生成済サーバーが停止します。

[図 2-9](#) は、接続要求の前に、事前生成済専用サーバーを生成し、接続をクライアントにリダイレクトして戻しているリスナーを示しています。

1. リスナーが、事前生成済専用サーバーを生成します。
2. リスナーがクライアント接続要求を受け取ります。
3. リスナーは、事前生成済専用サーバーの位置をリダイレクト・メッセージでクライアントに通知します。
4. クライアントは、事前生成済専用サーバーに直接接続します。
5. リスナーは、別の事前生成済専用サーバーを生成します。



図 2-9 事前生成済専用サーバー・プロセスへのリダイレクト接続



## 構成

事前生成済専用サーバーを使用するには、リスナー構成ファイル内のパラメータで指定する必要があります。

**関連項目：** 構成に関する情報は、7-13 ページの「[事前生成済専用サーバーの構成](#)」を参照してください。

## 専用サーバーへ直接伝送されたネットワーク・セッション

クライアントとサーバーが同じマシン上に存在する場合、クライアント接続は、リスナーを経由せずに直接専用サーバー・プロセスに伝送する（渡す）ことができます。セッションを開始するアプリケーションは、Bequeath プロトコルを使用して、接続要求に対する専用サーバー・プロセスを生成します。

データベースを同じマシン上で起動するのにアプリケーションを使用する場合、この処理は次に示すようなコマンドで自動的に実行されます。

```
sqlplus
```

```
SQL> connect internal/password
```



## クライアント接続オプション

クライアント接続は、次の項で説明しているオプションを使用してカスタマイズできます。

- 複数のリスナーに対するクライアント・ロード・バランス
- 複数のリスナーに対する接続時フェイルオーバー
- 可用性を高めるための透過的アプリケーション・フェイルオーバー

### 複数のリスナーに対するクライアント・ロード・バランス

複数のリスナーが1つのサービスをサポートしている場合、クライアントは様々なリスナーへの要求をランダム化できます。この機能は[クライアント・ロード・バランス \(client load balancing\)](#)と呼ばれ、1つのリスナーにロードが集中しないようにロードを分散します。

#### 構成

対象のクライアントがリスナーからの選択をランダムに実行できるようにするには、次に示すように、接続記述子を複数のリスニング・アドレスで構成して、LOAD\_BALANCE パラメータをオンに設定します。

```
sales=
(description=
  (load_balance=on)
  (address=(protocol=tcp) (host=sales1-server) (port=1521))
  (address=(protocol=tcp) (host=sales2-server) (port=1521))
  (connect_data=
    (service_name=sales.acme.com)))
```

クライアント・ロード・バランスは、接続時フェイルオーバーと同時に使用できます。

**関連項目：** 構成に関する情報は、8-3 ページの「[アドレス・リスト・パラメータの構成](#)」を参照してください。

### 複数のリスナーに対する接続時フェイルオーバー

複数のリスナーが1つのサービスをサポートしている場合、最初のリスナーが失敗すると、異なるリスナーへのクライアント要求をフェイルオーバーするようにクライアントを構成できます。再接続の試行は、クライアントが正常にリスナーに接続するまで継続されます。

#### 構成

これらの接続試行をクライアントが実行する方法を制御するには、次に示すように、接続記述子を複数のリスニング・アドレスで構成して、FAILOVER パラメータをオンに設定します。



```
sales=
  (description=
    (failover=on)
    (address=(protocol=tcp) (host=sales1-server) (port=1521))
    (address=(protocol=tcp) (host=sales2-server) (port=1521))
    (connect_data=
      (service_name=sales.acme.com)))
```

接続時フェイルオーバーは、クライアント・ロード・バランスと同時に使用できます。

**関連項目：** 構成の詳細は、8-3 ページの「[アドレス・リスト・パラメータの構成](#)」を参照してください。

## 可用性を高めるための透過的アプリケーション・フェイルオーバー

**透過的アプリケーション・フェイルオーバー (Transparent Application Failover:TAF)** は、Oracle Parallel Server や Oracle Fail Safe など、可用性の高い環境を目的としたランタイム・フェイルオーバーです。これは、アプリケーションとサービス間接続のフェイルオーバーおよび再確立を指しています。これを利用すると、クライアント・アプリケーションは、接続が失敗した場合データベースに自動的に再接続できます。また、オプションで進行中の SELECT 文を再開できます。この再接続は、Oracle Call Interface (OCI) ライブラリの内部から自動的に実行されます。

**関連項目：** 『Oracle8i 概要』

### 構成

TAF を使用可能にするには、接続記述子の接続データ部分にある接続記述子を FAILOVER\_MODE パラメータで設定します。

**関連項目：** 構成の詳細は、8-11 ページの「[透過的アプリケーション・フェイルオーバーの構成](#)」

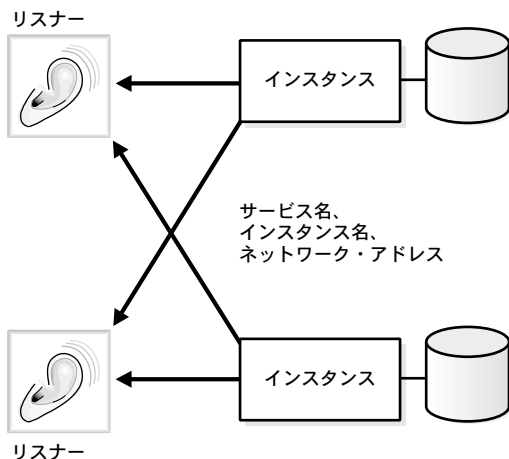
## リスナーに対するサービス登録

**PMON プロセス (PMON process)** は、インスタンスのバックグラウンド・プロセスであり、インスタンスの情報、インスタンスの現在の状態およびロード、MTS ディスパッチャをリスナーに登録します。この登録プロセスは、**サービス登録 (service registration)** と呼ばれます。この登録情報によって、リスナーはクライアント接続要求を適切なサービス・ハンドラに転送できます。

図 2-10 は、情報をリスナーに登録しているインスタンスを示しています。



図 2-10 サービス登録



サービス登録によってリスナーに登録される情報は、次のとおりです。

- データベースにおける稼動中の各インスタンスのサービス名
- データベースのインスタンス名
- 各インスタンスで使用可能なサービス・ハンドラ（ディスパッチャおよび専用サーバー）

これにより、リスナーは、クライアントの要求を適切に送ることができます。

- ディスパッチャ、インスタンスおよびノードのロード情報

このロード情報により、リスナーは、クライアント接続の要求を最適に処理できるディスパッチャを判別できます。すべてのディスパッチャがブロックされている場合、リスナーは、この接続に対する専用サーバーを生成できます。

インスタンスが起動すると、初期化パラメータ・ファイルから初期化パラメータが読み取られます。デフォルトでは、PMON は、予約済アドレスで、ローカル・マシン上にあるリスナーに情報を登録します。

インスタンスの起動時にリスナーが起動していない場合は、PMON は情報を登録できません。ただし、PMON は、リスナーに定期的に接続を試みます。リスナーが着信要求を受け取るのが、対応するインスタンスが登録されるより前の場合は、リスナーは要求を拒否します。



## 利点

サービス登録には、次の利点があります。

利点	説明
構成	<p><code>listener.ora</code> の必要性が削減されます。</p> <p><b>注意：</b> リスナーによって処理されるデータベースの管理に Oracle Enterprise Manager を使用している場合は、データベースについての情報を指定する <code>SID_LIST_listener_name</code> パラメータが依然として必要になります。</p>
MTS に対する接続時フェイルオーバー	<p>サービス登録を利用すると、リスナーは、インスタンスの起動の有無を常に認識できます。このため、あるインスタンスがダウンした場合、クライアント接続要求に対して、別のインスタンスへの自動フェイルオーバーを行うことが容易になります。</p> <p>静的構成モデルでは、リスナーは、クライアント要求を受け取ったときに専用サーバーを起動します。専用サーバーは、インスタンスがまだ起動していないために、"Oracle not available" というエラー・メッセージが発生していることを後で認識する場合があります。</p> <p><b>関連項目：</b> 2-18 ページの「複数のリスナーに対する接続時フェイルオーバー」</p>
接続時ロード・バランス化	<p>サービス登録を利用すると、リスナーは、クライアント接続要求をロード量の最も少ないディスパッチャに転送できます。これにより、リスナーは、サービス・ハンドラ間およびノード間で負荷の均衡化を図ることができます。</p> <p>Oracle Parallel Server 環境では、サービス登録によってロード量の最も少ないインスタンスを検出できます。</p> <p><b>関連項目：</b> 2-23 ページ「接続時ロード・バランス化」</p>

## デフォルトのローカル・リスナーへの登録

デフォルトでは、PMON は、TCP/IP のデフォルトのローカル・アドレスであるポート 1521 にあるサーバーにローカル・リスナーを登録します。

## デフォルト以外のリスナーへの登録

TCP/IP またはポート 1521 を使用しないサーバーのローカル・リスナーへ PMON によって登録する必要がある場合は、初期化ファイルの `LOCAL_LISTENER` パラメータを構成して、ローカル・リスナーを指定します。`LOCAL_LISTENER` パラメータは、次のように構成します。

```
local_listener=listener_alias
```



*listener\_alias* は、次に、サーバー上にある `tnsnames.ora` ファイルなどのネーミング・メソッドによって解決されます。

たとえば、LOCAL\_LISTENER パラメータは、次のように設定できます。

```
local_listener=listener1
```

`listener1` は、ローカルの `tnsnames.ora` ファイルによって、次のように解決できます。

```
listener1=
(description=
  (address=(protocol=tcp) (host=sales-server) (port= 1421)))
```

**関連項目：** LOCAL\_LISTENER の設定については、7-8 ページの  
「[LOCAL\\_LISTENER パラメータの構成](#)」を参照してください。

## リモート・リスナーへの登録

リモート・リスナーへの登録は、たとえば Oracle Parallel Server の場合、MTS\_DISPATCHERS パラメータの LISTENER 属性を使用すれば、MTS 環境用に構成できます。LISTENER 属性は、次のように設定します。

```
mts_dispatchers="(protocol=tcp) (listener=listener_alias)"
```

`listener_alias` は、`tnsnames.ora` ファイルなどのネーミング・メソッドによって解決されます。

たとえば、MTS\_DISPATCHERS パラメータは、初期パラメータ・ファイルに次のように設定できます。

```
mts_dispatchers="(protocol=tcp) (listener=listeners_sales)"
```

`listeners_sales` は、次に、ローカルの `tnsnames.ora` ファイルによって、次のように解決できます。

```
listeners_sales=
(description=
  (address_list=
    (address=(protocol=tcp) (host=sales1-server) (port= 1521))
    (address=(protocol= tcp) (host=sales2-server) (port=1521))))
```

## 構成

サービス登録を使用可能にするには、初期パラメータ・ファイルに必ず次のパラメータが含まれている必要があります。

- データベース・サービス名の SERVICE\_NAMES
- インスタンス名の INSTANCE\_NAME



たとえば、次のように指定します。

```
service_names=sales.us.acme.com
instance_name=sales
```

#### 関連項目：

- SERVICE\_NAMES パラメータおよび INSTANCE\_NAME パラメータの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。
- LOCAL\_LISTENER パラメータの詳細は、7-8 ページの「[デフォルト以外のアドレスを使用するリスナーの構成](#)」を参照してください。

## 接続時ロード・バランス化

接続時ロード・バランス化を利用すると、複数のディスパッチャ間のアクティブな接続数を均衡化することによって、接続時のパフォーマンスが向上します。Oracle Parallel Server 環境では、接続時ロード・バランス化は、複数のインスタンス間のアクティブな接続数を均衡化する機能も備えています。

PMON はリモート・リスナーを登録できるので、リスナーは、すべてのインスタンスおよびディスパッチャを、その場所にかかわらず常に認識することができます。リスナーは、着信クライアント要求の送信先となるインスタンスおよびディスパッチャを、ロード情報に応じて判別します。したがって、リスナーは着信要求を受け取ると、次の順序でディスパッチャを選択します。a) ロード量が最小のノード、b) ロード量が最小のインスタンス、c) そのインスタンスについて、ロード量が最小のディスパッチャ。

データベース・サービスが複数のノード上に複数のインスタンスである場合、リスナーはロード量が最小のノード上にあるロード量が最小のインスタンスを選択します。次に、選択されたインスタンスに基づいて、ロード量が最小のディスパッチャが選択されます。

## 構成

接続時ロード・バランス化を使用可能にするには、初期パラメータ・ファイルに必ず次のパラメータを指定しておく必要があります。

- データベース・サービス名の SERVICE\_NAMES
- インスタンス名の INSTANCE\_NAME
- MTS を使用可能にして、インスタンスが使用するディスパッチャを指定するための MTS\_DISPATCHERS

たとえば、次のように指定します。

```
service_names=sales.us.acme.com
instance_name=sales
mts_dispatchers="(protocol=tcp) (dispatchers=2) (listener=listeners_sales)"
```



Oracle Parallel Server 環境では、各インスタンスのディスパッチャは、他のノード上にある他のリスナーによって相互に登録する必要があります。これを行うには、MTS\_DISPATCHERS パラメータの LISTENER 属性を使用します。

### 関連項目：

- 相互登録の詳細は、2-22 ページの「[リモート・リスナーへの登録](#)」を参照してください。
- SERVICE\_NAMES パラメータおよび INSTANCE\_NAME パラメータの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。
- LISTENER 属性の詳細は、[第9章](#)を参照してください。

---

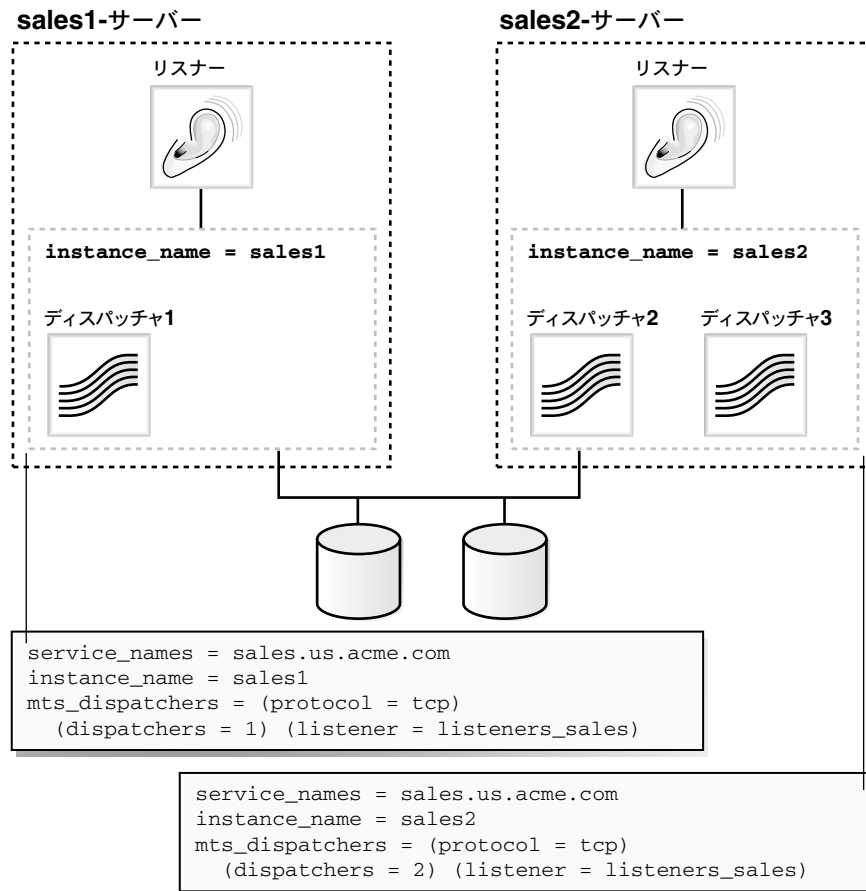
**注意：** 接続時ロード・バランス化で最適な結果を得るためには、同じデータベース・サービスに属するインスタンスは、等価なハードウェアおよびソフトウェア構成上に存在するようにしてください。

---

**例** [図 2-11](#) は、同じサービスである sales.us.acme.com の 2 つのインスタンス sales1 および sales2 を持つ Oracle Parallel Server データベースを示しています。これらのインスタンス sales1 および sales2 は、それぞれ sales1-server および sales2-server というマシン上に存在します。sales1 にはディスパッチャが 1 つあり、sales2 にはディスパッチャが 2 つあります。LISTENER という名前のリスナーは、それぞれノード 1 および 2 上で稼動しています。MTS\_DISPATCHER パラメータの LISTENER 属性が構成され、2 つのリスナーに対して情報のサービス登録が可能になっています。



図 2-11 ロード・バランス化環境



(listener=listeners\_sales) の listeners\_sales の値は、次に、両方のサーバー上に存在するローカルの tnsnames.ora ファイルによって、次のように解決できます。

```
listeners_sales=
(description=
(address_list=
(address=(protocol=tcp) (host=sales1-server) (port= 1521))
(address=(protocol= tcp) (host=sales2-server) (port=1521))))
```

環境に応じて、次のような処理が実行されます。次の各処理に付いている番号は 2-27 ページの、[図 2-12](#) に示す矢印の番号に対応しています。



1. PMON は、インスタンス sales1 および sales2 を、両方のリスナーに登録します。リスナーは、インスタンスおよびディスパッチャのロードについて動的に更新され、次のロード情報が登録されます。

	dispatcher1	dispatcher2	dispatcher 3
接続数	200	100	200

	sales1	sales2
接続数	200	300

	sales1-server	sales2-server
1 分当りのロード 量平均	600	400

sales1 の接続数（200）は、その唯一のディスパッチャである dispatcher1 の接続数と同じです。しかし、sales2 の接続数（300）は、その 2 つのディスパッチャである dispatcher2 の接続数（100）と dispatcher3 の接続数（200）の合計になります。したがって、sales2 には sales1 より多くの接続が存在します。ただし、sales2-server のロード量平均（400）は、sales1-server のロード量平均（600）より少なくなります。これは、sales1-server に必要な処理が多い場合に発生する可能性があります。

2. クライアントが接続要求を送信します。

接続記述子が構成されリスナー・アドレスが 1 つ成功するまで、各リスナー・アドレスがランダムに試行されます。

```
sales.us.acme.com=
  (description=
    (load_balance=on)
    (failover=on)
    (address=(protocol=tcp) (host=sales1-server) (port=1521))
    (address=(protocol=tcp) (host=sales2-server) (port=1521))
    (connect_data=(service_name=sales.acme.com)))
```

sales1-server 上のリスナーがランダムに選択されて、クライアント接続要求を受信します。

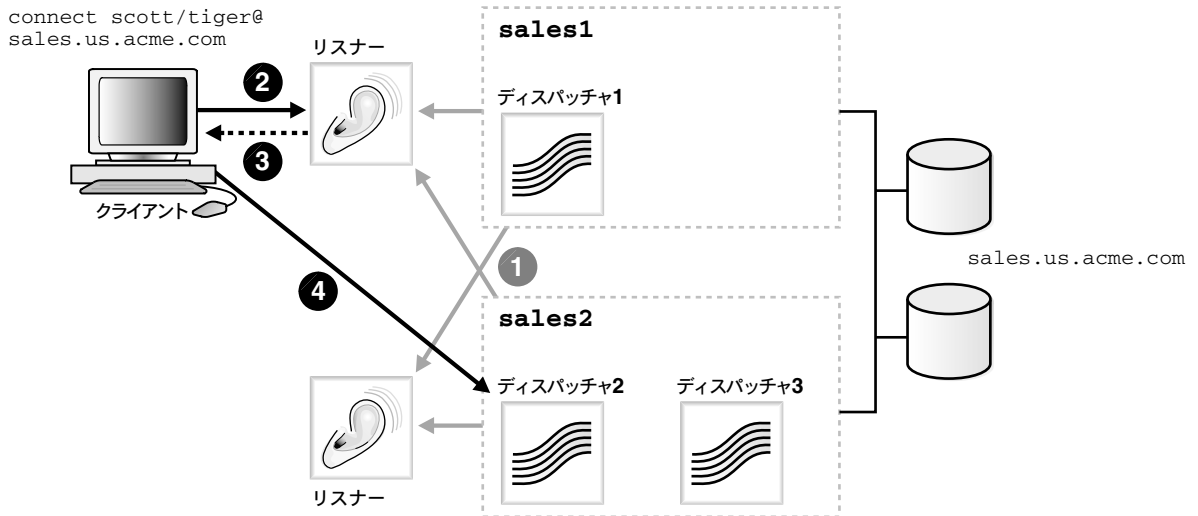
sales1-server 上のリスナーは、インスタンス sales1 および sales2 のロード量を比較します。この比較では、ノード sales1-server および sales2-server 上のロード量がそれぞれ考慮されます。sales2-server のロード量は sales1-server のロー



ド量より少ないため、リスナーは、sales1-server より sales2-server を選択します。

3. リスナーは、ディスパッチャ dispatcher2 のロード量と dispatcher3 のロード量を比較します。dispatcher2 のロード量は dispatcher3 のロード量より少ないため、リスナーはクライアント接続要求を dispatcher2 にリダイレクトします。
4. クライアントは、dispatcher2 に直接接続します。

図 2-12 ロード・バランス化の例



## SID ではなくサービス名によるデータベースの識別

Oracle データベースの対象範囲は、複数のコンピュータ上の複数のインスタンスに及ぶため、データベース・サービスは分散サービスになります。

Oracle8i より前では、Oracle データベース・サービスは、**Oracle システム識別子 (Oracle System Identifier: SID)**、つまりあるデータベースの固有インスタンスを識別する名前で識別されていました。クライアントをデータベース・インスタンスに接続するときは、接続記述子に SID を指定していました。この命名方法では、インスタンスとサービスを区別していませんでした。

Oracle8i では、新しい方法がインプリメントされました。Oracle データベースの対象範囲は、複数のコンピュータに及ぶため、サービス全体とその各インスタンスの両方が指定されます。



データベース・サービスは、初期化パラメータ・ファイルの `SERVICE_NAMES` パラメータを指定することによって、**サービス名 (service name)** で識別されるようになりました。`SERVICE_NAMES` パラメータでは、対象範囲の複数のインスタンスおよびノードに及ぶことが可能な Oracle データベース・サービスについて最高レベルの表示名が指定されます。`SERVICE_NAMES` のデフォルト値は **グローバル・データベース名 (global database name)** です。これは、データベース名 (`DB_NAME`) およびドメイン名 (`DB_DOMAIN`) から構成される名前です。

データベース・インスタンスは、初期化パラメータ・ファイルの `INSTANCE_NAME` パラメータを指定することによって、インスタンス名で識別されます。`INSTANCE_NAME` は、インスタンスの `SID` に対応しています。

この項の内容は次のとおりです。

- **`SERVICE_NAME` および `INSTANCE_NAME` の使用による接続記述子**
- **同じサービスに属している複数のインスタンス**
- **複数のサービスに属している単一のインスタンス**

## SERVICE\_NAME および INSTANCE\_NAME の使用による接続記述子

クライアントが使用する接続記述子は、`SERVICE_NAME` パラメータ (S は付けません) で構成して、Oracle8i データベースに接続します。たとえば、次の接続記述子には、`sales.us.acme.com` と呼ばれるデータベース・サービスの接続要求をリスニングしている `sales1-server` 上にあるリスナーのアドレスが格納されています。

```
sales=
(description=
  (address=(protocol=tcp) (host=sales1-server) (port=1521))
(connect_data=
  (service_name=sales.us.acme.com)))
```

接続記述子は、オプションとして、(`SERVICE_NAME` パラメータに加えて) `INSTANCE_NAME` パラメータで構成して、データベースの特定のインスタンスに接続できます。これは、複数のインスタンスを持つ Oracle Parallel Server を保有している場合に役立ちます。

たとえば、次の接続記述子には、`sales.us.acme.com` データベースに関連付けられている `sales1` と呼ばれるインスタンスに対する接続要求をリスニングしている `sales1-server` 上にあるリスナーのアドレスが格納されています。

```
sales=
(description=
  (address=(protocol=tcp) (host=sales1-server) (port=1521))
(connect_data=
  (instance_name=sales1)
  (service_name=sales.us.acme.com)))
```



## 同じサービスに属している複数のインスタンス

複数のデータベース・インスタンスは、Oracle Parallel Server 環境の場合のように、異なるマシン上で動作でき、同じデータにアクセスできますが、これらのインスタンスは同じサービスに属しています。

複数のインスタンスの場合、リスナーは、接続ロード・バランス化を実行して、ロード量が最小のインスタンスおよびロード量が最小のディスパッチャを検索します。接続記述子の SERVICE\_NAME パラメータを指定するクライアントは、この最小のロード量のインスタンスおよびディスパッチャに送られます。

特定のインスタンスの使用が必要な場合、クライアントは、接続記述子にある特定のインスタンスで INSTANCE\_NAME を指定できます。クライアントは、インスタンスが指定されたサービスに属している限りは、このようにしてそのインスタンスに接続されます。

## 複数のサービスに属している単一のインスタンス

データベース・サービスは、初期化パラメータ・ファイルにある SERVICE\_NAMES パラメータの複数のサービス・エントリで表現できます。この機能によって可能なことは、次のとおりです。

- 単一のインスタンスを、様々な方法で様々なクライアントによって識別できます。
- データベース管理者は、これらのサービスの 1 つを要求しているクライアントにサービスを提供するために割り当てたり予約したりするリソースを制限できます。

たとえば、初期化パラメータ・ファイルは次のように構成できます。

```
service_names=sales1.us.acme.com, sales2.us.acme.com
```

この構成を使用すると、MTS ディスパッチャのプールを、sales1.us.acme.com を要求しているクライアントのみに割り当てることができます。sales2.us.acme.com を要求しているその他のクライアントが、要求を処理するためには別のディスパッチャの使用を強制することができます。このようにすると、ミッション・クリティカルな要求に対してより多くのリソースを割り当てたり、優先順位を高くしたりすることができます。

**関連項目：** リソース割当ての例は、9-9 ページの「[リソースの割当て](#)」を参照してください。

## 移行

古い命名方法から新しい命名方法への移行は、できるだけ早急に行い、現行の機能の利点を十分活用するようにお勧めします。将来の新機能では、クライアント接続記述子で SID を使用できなくなります。

**関連項目：** 『Oracle8i 移行ガイド』を参照してください。



## ネーミング・メソッドによるサービスの解決

接続記述子は、1つまたは複数の**ネーミング・メソッド (naming method)** で解決されます。ネーミング・メソッドとは、Net8が使用する解決方法のことで、名前をネットワーク記述子に解決します。Net8には、次のネーミング・メソッドが用意されています。

- ローカル・ネーミング
- ディレクトリ・ネーミング
- Oracle Names
- ホスト・ネーミング
- 外部ネーミング

### ローカル・ネーミング

ローカル・ネーミングでは、`tnsnames.ora` と呼ばれるローカル・ネーミング構成ファイルで構成された情報を使用して、ネット・サービス名をネットワーク・アドレスに解決します。

**関連項目：** 構成の詳細は、6-4 ページの「[ローカル・ネーミング・メソッドの構成](#)」を参照してください。

#### ローカル・ネーミングを使用した接続の確立

ローカル・ネーミングを使用してクライアント・セッションを確立するためのプロセスは、次のとおりです。

1. クライアントは、ネット・サービス名を指定して接続要求を開始します。
2. ネット・サービス名は、`tnsnames.ora` ファイルで構成された接続記述子に解決されます。
3. クライアントは、接続記述子内に存在するアドレスに対して、接続要求を実行します。
4. リスナーは要求を受け取り、それを該当するサーバーに送ります。
5. サーバーによって、接続が受け入れられます。

### ディレクトリ・ネーミング

**ディレクトリ・ネーミング (directory naming)** を使用すると、クライアントは、集中化された LDAP 準拠ディレクトリ・サーバーに格納されたネットワーク情報を使用して、データベース・サービスにアクセスできます。



**関連項目：**

- 概念的な情報は、2-33 ページの「[Net8 および LDAP 準拠ディレクトリ・サーバー](#)」を参照してください。
- 構成に関する情報は、6-13 ページの「[ディレクトリ・ネーミング・メソッドの構成](#)」を参照してください。

**ディレクトリ・ネーミングを使用した接続の確立**

ディレクトリ・ネーミングを使用してクライアント・セッションを確立するためのプロセスは、次のとおりです。

1. クライアントは、接続識別子を指定して接続要求を開始します。
2. 接続識別子は、ディレクトリ・サーバーによって接続記述子に解決されます。この情報は、クライアントに戻ります。
3. クライアントは、接続記述子内に存在するアドレスに対して、接続要求を実行します。
4. リスナーは要求を受け取り、それを該当するサーバーに送ります。
5. サーバーによって、接続が受け入れられます。

**Oracle Names**

[Oracle Names](#) は、Oracle の専用ソフトウェアを使用して、ネットワーク上のすべてのデータベース・サービスの名前およびアドレスを格納します。サーバーに接続するクライアントは、Oracle Names サーバーに対して自分たちの接続要求を送ります。Oracle Names サーバーは、この名前をネットワーク・アドレスに解決し、その情報をクライアントに戻します。

**関連項目：**

- 製品情報は、4-8 ページの「[Oracle Names](#)」を参照してください。
- 構成の詳細は、6-43 ページの「[Oracle Names メソッドの構成](#)」を参照してください。

**Oracle Names を使用した接続の確立**

Oracle Names オプションを使用してクライアント・セッションを確立するためのプロセスは、次のとおりです。

1. クライアントは、接続識別子を指定して接続要求を開始します。
2. 接続識別子は、Oracle Names サーバーによって接続記述子に解決されます。この情報は、クライアントに戻ります。
3. クライアントは、接続記述子内に存在するアドレスに対して、接続要求を実行します。



4. リスナーは要求を受け取り、それを該当するサーバーに送ります。
5. サーバーによって、接続が受け入れられます。

## ホスト・ネーミング

**ホスト・ネーミング**では、ユーザーはホスト名エイリアスを使用して Oracle Server に接続できます。ホスト名は、ドメイン名システム (DNS)、ネットワーク・インフォメーション・サービス (NIS)、または集中管理された `/etc/hosts` ファイルのセットなどの既存の名前解決サービスに存在する、サーバーのグローバル・データベース名にマップされます。

接続は、リスナーのデフォルトの TCP/IP ポートである 1521 を使用して確立されます。ノードごとの複数のデータベースおよびデータベース位置の透過性は、グローバル・データベース名とホスト名の別名を一致させることによってサポートされます。

ホスト・ネーミングでは、データベースとの単純な接続を必要とする環境で使用される `tnsnames.ora` ファイルが不要になります。このことは、**Oracle Connection Manager** を使用する環境、TCP/IP を使用しない環境または透過的アプリケーション・ファイルオーバーを必要とする環境には該当しません。

**関連項目：** 構成の詳細は、6-68 ページの「**ホスト・ネーミング・メソッドの構成**」を参照してください。

## ホスト・ネーミングを使用した接続の確立

ホスト・ネーミング・オプションを使用してクライアント・セッションを確立するためのプロセスは、次のとおりです。

1. クライアントは、TCP/IP ホスト名別名を指定して接続要求を開始します。
2. ホスト・ネーミングでは、TCP/IP ホスト名とサービス名を兼ねるネット・サービス名を使用してネットワーク・アドレスを生成することによって、このホスト名または別名が解決されます。TCP/IP ポートは 1521 にデフォルト設定されています。
3. Net8 は、作成されたアドレスに対して接続要求を送信します。
4. リスナーは、登録された TCP/IP ポート 1521 でリスニングし、クライアントの要求を受け取って、該当するサーバーへの接続を確立します。
5. 該当するサーバーによって、接続が受け入れられます。

## 外部ネーミング

外部ネーミングとは、サポートされている Oracle 以外のネーミング・サービスを使用して、ネットワーク・アドレスにネット・サービス名を解決する方法です。外部ネーミングでは、顧客サイトの Oracle 以外の外部ネーミング・サービスに格納されているネット・サービス名が解決されます。外部ネーミングの対象は、次のとおりです。

- ネットワーク情報サービス (NIS) 外部ネーミング



- Novell Directory Services (NDS) 外部ネーミング
- 分散コンピューティング環境 (DCE) セル・ディレクトリ・サービス (CDS)

**関連項目：** 構成の詳細は、6-71 ページの「[外部ネーミング・メソッドの構成](#)」を参照してください。

## 外部ネーミングを使用した接続の確立

外部ネーミングを使用してクライアント・セッションを確立するためのプロセスは、次のとおりです。

1. クライアントは、ネット・サービス名を指定して接続要求を開始します。
2. 外部ネーミング・メソッドによって外部ネーミング・システムに要求が送られ、そこでネット・サービス名がネットワーク・アドレスに解決されます。アドレスがクライアントに戻ります。
3. Net8 は、指定されたアドレスに接続要求を送信します。
4. リスナーは要求を受け取り、それを該当するデータベースに送ります。
5. サーバーによって、接続が受け入れられます。

## Net8 および LDAP 準拠ディレクトリ・サーバー

この項の内容は、次のとおりです。

- [概要](#)
- [ディレクトリ・ネーミングの概要](#)
- [ディレクトリ・エントリ](#)
- [ディレクトリでのエントリの追加または変更](#)
- [ディレクトリ・ネーミングを使用したクライアントの接続](#)
- [絶対名の指定](#)
- [Net8 のネーミングおよびディレクトリ設計](#)

### 概要

今日では、ネットワーク情報は複数のシステムに複数のディレクトリ形式で格納されています。インターネット・コンピューティングや新しい e-business テクノロジーの新しい要求によって、すべてのデータおよびリソースの管理および構成の基盤として機能する共通のリポジトリ・インフラストラクチャの必要性が増加しています。このような共通の基盤によって、異機種間ネットワークでのリソースの管理および構成のコストが削減されます。



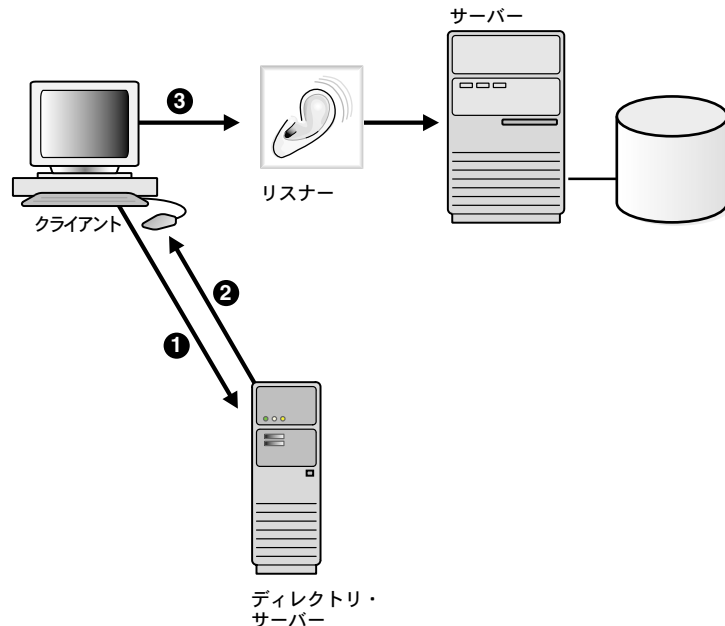
LDAP 準拠ディレクトリ・サーバーのサポートによって、分散 Oracle ネットワークの管理および構成に関する集中化された媒体が提供されます。このディレクトリは、データベース・ネットワーク・コンポーネント、ユーザー・ポリシー、企業ポリシー、ユーザー認証およびユーザー・セキュリティに関するすべてのデータの中央リポジトリとして機能できるので、クライアント側およびサーバー側に局在化している `tnsnames.ora` ファイルに置き換わります。

Net8 では、集中化されたディレクトリ・サーバーが**接続識別子 (connect identifier)** を格納する主な方法の 1 つとして使用されます。クライアントは、このディレクトリにアクセスするよう構成されていて、その接続文字列に接続識別子を使用します。このディレクトリは、この接続識別子をクライアントに戻される接続記述子に解決します。

図 2-13 は、ディレクトリによって接続識別子を解決しているクライアントを示しています。

1. クライアントは、接続識別子を接続記述子に解決するために、ディレクトリに接続します。
2. ディレクトリは、この接続識別子を解決して、クライアントから接続記述子を取り出します。
3. クライアントは、この接続記述子を使用して接続要求を送信します。

図 2-13 ディレクトリを使用して接続識別子を解決するクライアント





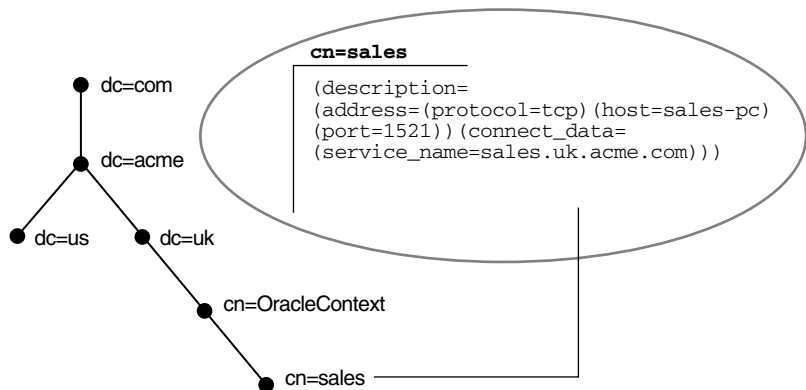
Net8 では、次のディレクトリがサポートされます。

- Oracle Internet Directory
- Microsoft の Active Directory
- Novell Directory Services

## ディレクトリ・ネーミングの概要

Net8 によるディレクトリの使用形態を理解するには、[図 2-14](#) を考察します。

図 2-14 ディレクトリでのデータベース・サービス



このツリーの各ノードは、**エントリ (entry)** を表しています。右側の分岐は、sales と呼ばれるデータベース・サービスのエントリを表しており、dc=uk、dc=acme、dc=com という階層的なドメイン構造の下に存在します。

このエントリの階層は、**ディレクトリ情報ツリー (Directory Information Tree: DIT)** と呼ばれます。各エントリは、**識別名 (Distinguished Name: DN)** によって一意に識別されます。DN は、ディレクトリ階層内に対象のエントリが存在する場所を正確に示します。sales の DN は、(dn:cn=sales,cn=OracleContext,dc=uk,dc=acme,dc=com) です。DN の形式では、DIT の最下位コンポーネントを左側に置いてから、DIT を上方向に徐々に移動することに注意してください。各 DN は、**Relative Distinguished Names (RDN)** の列で構成されています。これは、ディレクトリ・パスにディレクトリの列が格納されているのと似ています。sales のエントリでは、RDN は cn=sales です。RDN は、**属性 (attribute)** の集合から構成されています。たとえば、cn=sales では、cn は、このエントリの属性の 1 つです。属性は、その値を伴って、エントリを一意に識別します。



この章で参照される共通の属性は、次のとおりです。

属性	説明
commonName、cn	エントリの共通名
country、c	国の名前
domain component、dc	ドメイン・コンポーネント
organization、o	組織の名前
organizationalUnitName、ou	組織内の単位の名前

sales が cn=OracleContext の下に存在することに注意してください。  
cn=OracleContext は、**Oracle コンテキスト**と呼ばれるディレクトリに存在する特殊なエントリで、ここには、ディレクトリのネーミングや**エンタープライズ・ユーザー (enterprise user)** のセキュリティをサポートするために、様々な Oracle エントリが格納されています。

**関連項目：** エンタープライズ・ユーザーのセキュリティの詳細は、『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。

この例では、Oracle コンテキストに sales データベース・サービスのエントリが格納されています。

インストール時またはインストール後に Net8 Configuration Assistant が実行されディレクトリ・アクセスの構成時に、**管理コンテキスト (administrative context)** が選択されます。管理コンテキストとは、Oracle コンテキスト (cn=OracleContext) が格納されているディレクトリ・エントリのことです。クライアントは、この管理コンテキストをデフォルトの場所として使用して、このディレクトリ内で接続識別子を検索します。

管理コンテキストの処理対象は、接続文字列です。たとえば、クライアントが sales エントリに頻繁にアクセスする必要がある場合、管理コンテキストは、dc=uk,dc=acme,dc=com などが妥当と思われます。cn=OracleContext は必要ないことに注意してください。これにより、クライアントは次の接続文字列を使用できます。

CONNECT scott/tiger@sales

sales は、Oracle コンテキストに相対しています。

クライアントの管理コンテキストが、サービスの存在する管理コンテキストと一致しない場合、2-39 ページの「**ディレクトリ・ネーミングを使用したクライアントの接続**」で説明するように、接続文字列には、必ずエントリの絶対名を指定する必要があります。

## ディレクトリ・エントリ

DIT は、次の基準を使用して、多種類の構造化が可能です。



- 既存のドメイン名前空間 (DNS)
- インターネット命名規則
- 地理的境界および組織的境界

図 2-15 は、DNS ドメイン・コンポーネントに従って構築された DIT を示します。

図 2-15 ドメイン・コンポーネント DIT

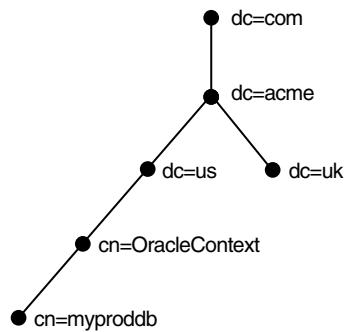
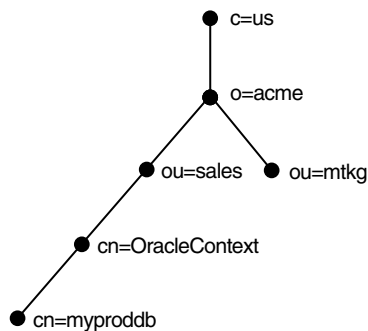


図 2-16 は、国、組織および組織単位に従って構築された DIT を示します。この構造は、一般に X.500 DIT と呼ばれています。

図 2-16 X.500 形式 DIT

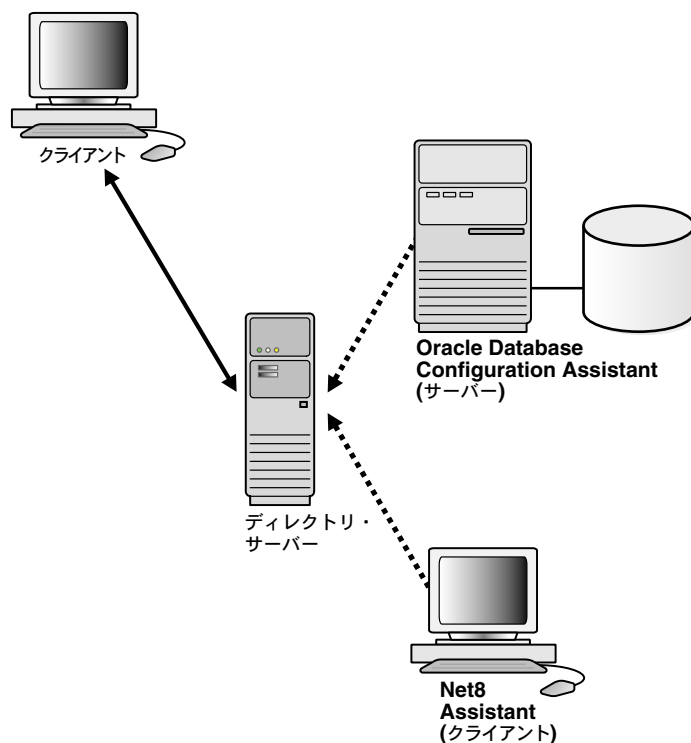




## ディレクトリでのエントリの追加または変更

データベース・サービスのエントリは、インストール時に、**Oracle Database Configuration Assistant** を使用して作成されます。ネット・サービス名のエントリは、**Oracle Net8 Assistant** を使用して作成できます。Oracle Net8 Assistant を使用すると、データベース・サービスやネット・サービスのエントリについて Net8 属性を変更することもできます。[図 2-17](#) は、Oracle Database Configuration Assistant および Oracle Net8 Assistant がディレクトリと連携する様子を示しています。

図 2-17 アプリケーションを使用したディレクトリのエントリの作成



ディレクトリ・アクセスのために構成されたクライアントは、2-39 ページの「[ディレクトリ・ネーミングを使用したクライアントの接続](#)」で説明するように、これらのアプリケーションによって作成されたエントリを使用して、データベースに接続できます。

これらのアプリケーションを使用してエントリを追加するには、必ず Oracle コンテキスト (cn=OracleContext) を作成する必要があります。Oracle コンテキストは、サーバー上でのディレクトリ・アクセス構成時に **Net8 Configuration Assistant** によって作成されます。



Oracle Database Configuration Assistant を使用する管理者は、データベース・サービスのエントリを作成するために、必ず OracleDBCreators グループ (cn=OracleDBCreators,cn=OracleContext) のメンバーになる必要があります。同様に、Oracle Net8 Assistant を使用して、ネット・サービス名、データベース・サービスの Net8 属性またはサービス名のエントリを追加する管理者は、必ず OracleNetAdmins グループ (cn=OracleNetAdmins,cn=OracleContext) のメンバーになる必要があります。Oracle コンテキストを作成したディレクトリ・ユーザーは、これらの両方のグループに自動的に追加されます。これにより、このユーザーは Oracle コンテキストの管理者になります。この他のユーザーは、ディレクトリ管理者によってこのグループに追加できます。

## ディレクトリ・ネーミングを使用したクライアントの接続

大半のクライアントが必要とするのは、ディレクトリ内での名前参照のみです。参照を実行するには、クライアントは、次に示す最低条件を満たす必要があります。

- 対象のディレクトリでの無名認証が可能なこと。
- Oracle コンテキストに存在する Net8 エントリへの読み込みアクセスが可能なこと。
- ディレクトリ・サービスへのアクセスを対象として構成されていること。

**Net8 Configuration Assistant** は、通常、クライアントのインストール時に必要なディレクトリ・アクセス構成を実行して、次の情報を読み取り専用の `ldap.ora` ファイルに格納します。

- ディレクトリのタイプ
- ディレクトリの位置
- 接続識別子の参照、作成および変更の元となる管理コンテキスト

クライアントは、このファイルを読み込んで、アクセス対象のディレクトリおよび Net8 エントリを検索します。

**関連項目：** ディレクトリ・アクセスの構成は、6-14 ページの「[作業 1: サーバーおよびクライアントでのディレクトリ・アクセスの構成](#)」を参照してください。

他のネーミング・メソッドを使用する場合と同様に、クライアントは、ディレクトリに格納できる接続識別子の 2 種類の 1 つを使用して、データベースに接続できます。データベース・サービスおよびネット・サービス・エントリは、それぞれの共通名で参照できます。または、接続文字列内に追加のディレクトリ位置情報が必要な場合があります。接続文字列内での接続識別子の指定方法は、クライアントの管理コンテキストの構成形態によって異なります。

ディレクトリ・ネーミングを使用した場合、エントリを識別する方法は 2 つあります。

- [エントリの相対名の使用](#)



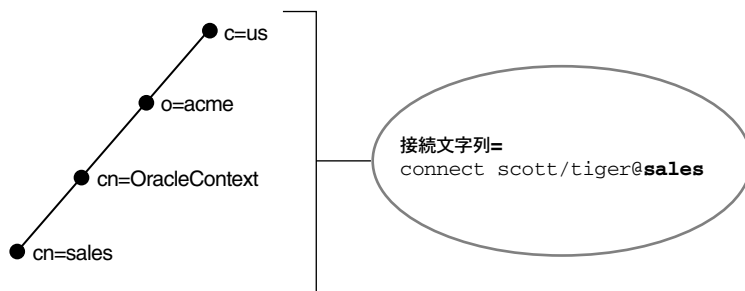
## ■ エントリの絶対名の使用

### エントリの相対名の使用

次の例は、エントリの相対名の使用例を示しています。これによって、サービスはその共通名で参照されます。相対名が使用できるのは、クライアントの構成に使用する管理コンテキストが、ディレクトリ内のエントリの管理コンテキストに一致する場合です。

図 2-18 に示すように、sales というデータベースのエントリが格納されているディレクトリと、DN (dn:cn=sales,cn=OracleContext,o=acme,c=us) について考えます。クライアントが o=acme,c=us という管理コンテキストで構成されている場合、sales が使用されている接続文字列が使用できます。

図 2-18 相対的ネーミング



### エントリの絶対名の使用

2-40 ページの図 2-18 と同じディレクトリ構造について考えます。ただし、クライアントは、dc=uk,dc=acme,dc=com という管理コンテキストで構成されています。

クライアントは、ディレクトリ内の sales の位置とは一致しない管理コンテキストで構成されているので、sales を使用する接続文字列は機能しません。そのかわりに、クライアントは必ず sales の位置を特定する必要があります。sales は、次の 2 つの方法で特定できます。

- エントリの完全な DN を接続文字列に使用します。例を次に示します。

```
CONNECT scott/tiger@cn=sales,cn=OracleContext,o=acme,c=us
```

- エントリをその絶対名で参照します。絶対名は、オブジェクトの名前とそのディレクトリ内の位置が含まれる名前、絶対パスを指定する場合と同様です。例を次に示します。

```
CONNECT scott/tiger@sales.acme.us
```

オラクル社では、エントリの DN ではなく、絶対名の使用をお勧めします。



**関連項目：** 絶対名の詳細は、「[絶対名の指定](#)」を参照してください。

## 絶対名の指定

この項では、次の名前空間の絶対名の構成方法について説明します。

- [X.500 名前空間の絶対名](#)
- [ドメイン・コンポーネント名前空間の絶対名](#)

### X.500 名前空間の絶対名

X.500 名前空間の場合、管理コンテキストは、必ず次のいずれかの形式にする必要があります。

`[ou],o`

`[ou],o,c`

`[ou]` は、オプションの組織単位名を表しています。

クライアントが接続識別子として使用する絶対名は、必ず次のいずれかの形式である必要があります。

`cn[.ou].o`

`cn[.ou].o.c`

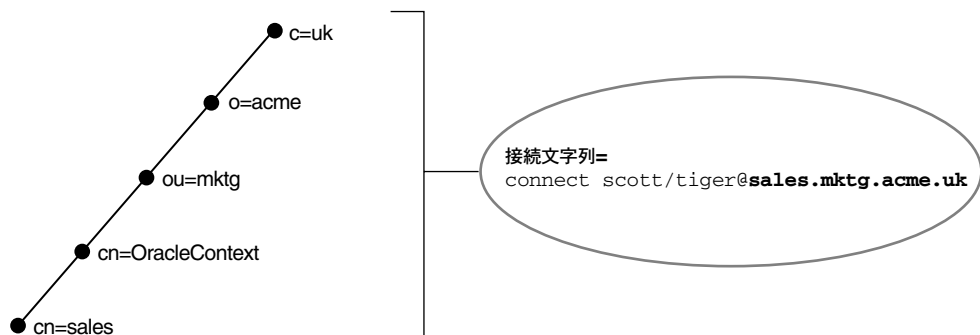
`[cn]` は、Net8 エントリを表しています。

たとえば、`ou=acctg,o=acme,c=us` という管理コンテキストで構成したクライアントについて考えます。

`sales` というデータベースのエントリが格納されているディレクトリと、DN (`dn:cn=sales,cn=OracleContext,ou=mktg,o=acme,c=uk`) について考えます。この例では、クライアントに `sales.mktg.acme.uk` (`cn.ou.o.c`) の接続識別子が必要です。[図 2-19](#) にこの例を示します。



図 2-19 X.500 名前空間の絶対名



## ドメイン・コンポーネント名前空間の絶対名

ドメイン・コンポーネント名前空間の場合、クライアントに定義された管理コンテキストは、必ず次のいずれかの形式にする必要があります。

`dc[,dc]*`

`ou,dc[,dc]*`

[dc] は、オプションのドメイン・コンポーネントを表しており、\* は、追加のドメイン・コンポーネントを表します。

クライアントが接続識別子として使用する必要がある絶対名は、必ず次のいずれかの形式である必要があります。

`cn.dc[.dc]*`

`cn[.ou]@dc[,dc]*`

[cn] は、Net8 エントリを表しています。

**例 1** `dc=us,dc=acme,dc=com` という管理コンテキストで構成したクライアントについて考えます。

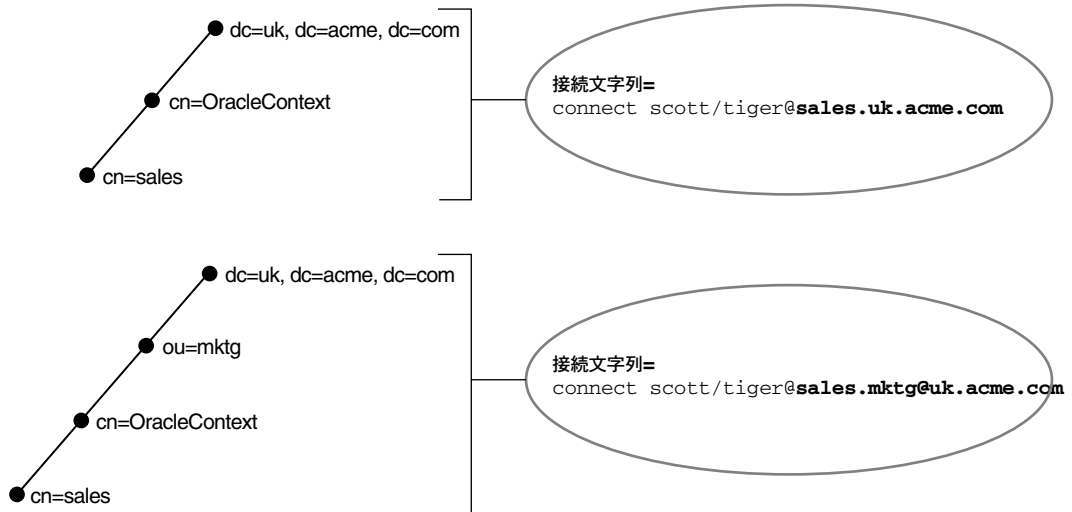
`sales` というデータベースのエントリが格納されているディレクトリと、DN (`dn:cn=sales,cn=OracleContext,dc=uk,dc=acme,dc=com`) について考えます。この例では、クライアントに `sales.uk.acme.com` (`cn.dc.dc.dc`) の接続識別子が必要です。2-43 ページの図 2-20 にこの例を示します。

**例 2** 例 1 と同じ管理コンテキストについて考えます。ディレクトリには、`sales` というデータベースのエントリと、DN (`dn:cn=sales,cn=OracleContext,ou=mktg,dc=uk,dc=acme,dc=com`) が格納され



ています。ou=marketing に注意してください。ドメイン・コンポーネントは必ず組織単位から分離する必要がありますので、クライアントに sales.marketing@uk.acme.com (cn.ou@dc.dc.dc) の接続識別子が必要です。2-43 ページの図 2-20 にこの例を示します。

図 2-20 ドメイン・コンポーネント名前空間の絶対名



## Net8 のネーミングおよびディレクトリ設計

Net8 のディレクトリの設計を担当している場合は、次の問題について検討してください。

- パフォーマンス
- セキュリティ
- スキーマ

### パフォーマンス

接続識別子は、アクセス対象のすべてのクライアントに対して、1つのディレクトリに格納されます。Net8 では、要求された名前についてディレクトリ参照を実行するため、その結果としてディレクトリが大量に使用されます。

参照時には、名前は特定の Oracle コンテキストのもとで検索されます。参照の有効範囲を考慮すると、ユーザーが比較的迅速なパフォーマンスで検索できる必要があると考えられます。パフォーマンスが迅速でない場合は、データベースの接続時間に影響が生じます。参照時間が1秒以上になると、ユーザーは接続時間が長く感じるようになります。



パフォーマンスの問題は、次の方法で解決できます。

- トポロジーの変更
- レプリケーション

**関連項目：** パフォーマンス問題の解決の詳細は、ディレクトリのドキュメントを参照してください。

セキュリティ

管理クライアントはディレクトリのエントリを作成したり変更できるため、セキュリティは必要不可欠です。ここでは、次の2つの項でのセキュリティ関連項目について説明します。

- [認証方式](#)
- [アクセス制御リスト](#)

**認証方式** ディレクトリ内の情報について通常の参照を実行するクライアントは、通常、無名認証を使用します。

あるディレクトリでエントリの追加または変更を行うクライアントは、Oracle Database Configuration Assistant または Oracle Net8 Assistant を使用する場合、必ずそのディレクトリで認証する必要があります。適正な権限を持つ認証ユーザーのみがエントリを変更できます。次の認証方式のいずれかが使用できます。

認証方式	説明
システム固有認証	このディレクトリでは、オペレーティング・システムのユーザー資格証明が使用されます。
単純認証	クライアントは、DN およびパスワードを平文で使用して、クライアント自体をディレクトリ・サーバーに認証要求します。サーバーは、クライアントが送信した DN およびパスワードが、ディレクトリに格納されている DN およびパスワードと一致していることを確認します。
強固な認証	ディレクトリでは、Secure Sockets Layer（SSL）で公開鍵暗号を使用することによって、強固な認証が得られます。公開鍵暗号では、メッセージの送信者は受信者の公開鍵を使用してメッセージを暗号化します。メッセージが送付されると、受信者は受信者の秘密鍵を使用して、このメッセージを復号化します。

**アクセス制御リスト** クライアントがディレクトリ内の情報に対して変更または追加を行うかどうかを判断するときは、認証と**アクセス制御リスト（ACL）**を併用します。ACL は、サーバー上でのディレクトリ・アクセス構成時に、Oracle コンテキストおよび Net8 Configuration Assistant を使用して作成します。

ACL の指定内容は、次のとおりです。



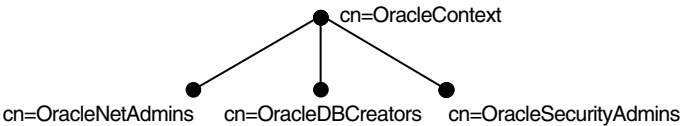
- ユーザーがアクセス可能なエントリ
- エントリをアクセスする認証方式
- アクセス権、またはユーザーがオブジェクトに対して可能な処理（読み込み / 書き込み）

ACL は、ユーザーのグループに対して作成されます。Oracle コンテキストの作成時には、次のグループが作成されます。

- OracleDBCreators
- OracleNetAdmins
- OracleSecurityAdmins

これらのグループは、図 2-21 に示すように、Oracle コンテキストの下に直接作成されます。

図 2-21 ディレクトリ・ユーザー・グループ



Net8 Configuration Assistant を使用して Oracle コンテキストを作成するユーザーは、これらのグループの先頭に自動的に追加されます。

次の表では、これらのグループおよび無名ユーザーに対する ACL 要件について、ディレクトリ内の Net8 エントリに対する関連内容に応じて説明しています。

グループ	ACL 要件
OracleNetAdmins グループ・ユーザー	<p>OracleNetAdmins (cn=OracleOracleNetAdmins,cn=OracleContext) のメンバーは、Net8 のオブジェクトおよび属性に対する作成、変更および読み込みアクセスが可能です。Net8 Configuration Assistant は、Oracle コンテキストの作成時に、このグループに対するこれらのアクセス権を作成します。</p> <p>Oracle コンテキストの作成者以外に、ディレクトリ管理者によってこの他のユーザーもこのグループに追加できます。</p> <p><b>関連項目：</b> OracleNetAdmins グループへユーザーを追加するには 6-27 ページの、「<a href="#">OracleNetAdmins グループへのユーザーの追加と削除</a>」を参照してください。</p>



グループ	ACL 要件
無名ユーザー	<p>無名ユーザーはすべての Net8 属性およびディレクトリ内のすべてのオブジェクトに、対する読み込みアクセスが可能です。無名ユーザーのこれらのオブジェクトに対する読み込みアクセスは、Oracle コンテキストにも適用されます。これにより無名ユーザーは、cn=OracleContext という RDN で格納されたエントリをブラウズできます。これには、エンタープライズ・ユーザー・セキュリティを対象とするオブジェクトは含まれていません。</p> <p>Net8 Configuration Assistant は、クライアントのインストール時にこのアクセス権をセットアップします。</p>
OracleDBCreators グループ・ユーザー	<p>OracleDBCreators (cn=OracleDBCreators,cn=OracleContext) のメンバーは、データベース・サービスのオブジェクトおよび属性に対する作成および読み込みアクセスが可能です。Net8 Configuration Assistant は、Oracle コンテキストの作成時に、このグループに対するこれらのアクセス権を作成します。</p> <p>Oracle コンテキストの作成者以外に、この他のユーザーも、ディレクトリ管理者が Oracle Enterprise Security Manager を使用してこのグループに追加できます。</p> <p><b>関連項目：</b> OracleDBCreators グループの詳細は、『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。</p>
OracleSecurityAdmins グループ・ユーザー	<p>OracleSecurityAdmins (cn=OracleOracleSecurityAdmins,cn=OracleContext) 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、変更および読み込みアクセスが可能です。Net8 Configuration Assistant は、Oracle コンテキストの作成時にこれらのアクセス権をセットアップします。</p> <p>Oracle コンテキストの作成者以外に、この他のユーザーも、ディレクトリ管理者が Oracle Enterprise Security Manager を使用してこのグループに追加できます。</p> <p><b>関連項目：</b> OracleSecurityAdmins グループの詳細は、『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。</p>

## スキーマ

ディレクトリは、Net8 クライアントがサービスのネットワーク情報を参照するために、Oracle スキーマの正しいバージョンによって作成される必要があります。Oracle スキーマは、ディレクトリおよびその属性に格納できるオブジェクトのタイプを定義します。このタイプは、**オブジェクト・クラス**と呼ばれます。Oracle スキーマは、次に示す Net8 参照のオブジェクト・クラスをサポートします。



オブジェクト・クラス	説明
orclDbServer	データベース・サービス・エントリの属性を定義します。
orclNetService	ネット・サービス名エントリの属性を定義します。

orclDbServer および orclNetService は、次のオブジェクト・クラスを使用します。

オブジェクト・クラス	説明
orclNetAddress	リスナー・プロトコル・アドレスを定義します。
orclNetAddressListorclNetAddressList	アドレスのリストを定義します。
orclNetDescription	データベースのリスナー・アドレスおよびサービスに対する接続情報が記述されている接続記述子を指定します。
orclNetDescriptionList	接続記述子のリストを定義します。

**関連項目：** これらのオブジェクト・クラスの詳細は、[付録 D](#) を参照してください。







---

## Net8 アーキテクチャ

この章では、Net8 のアーキテクチャについて説明します。この章の内容は次のとおりです。

- [Transparent Network Substrate \(TNS\)](#)
- [スタック通信の概要](#)
- [代表的な Net8 クライアント / サーバー環境でのスタック通信](#)
- [ディレクトリ・サーバーに接続しているクライアントのスタック通信](#)
- [JDBC クライアントのスタック通信](#)
- [Oracle8i JServer クライアントのスタック通信](#)



## Transparent Network Substrate (TNS)

**Transparent Network Substrate (TNS)** は、すべての業界標準プロトコルに対して1つの共通インタフェースを提供するために、Net8 に構築されている基礎技術です。

TNS を利用すると、peer-to-peer 接続が可能になります。peer-to-peer アーキテクチャでは、複数のコンピュータ（ネットワーク化された環境で使用される場合は**ノード (node)** と呼ばれます）が相互に直接交信でき、中間のデバイスは必要ありません。peer-to-peer システムでは、ノードはクライアントとサーバーの両方になり得ます。

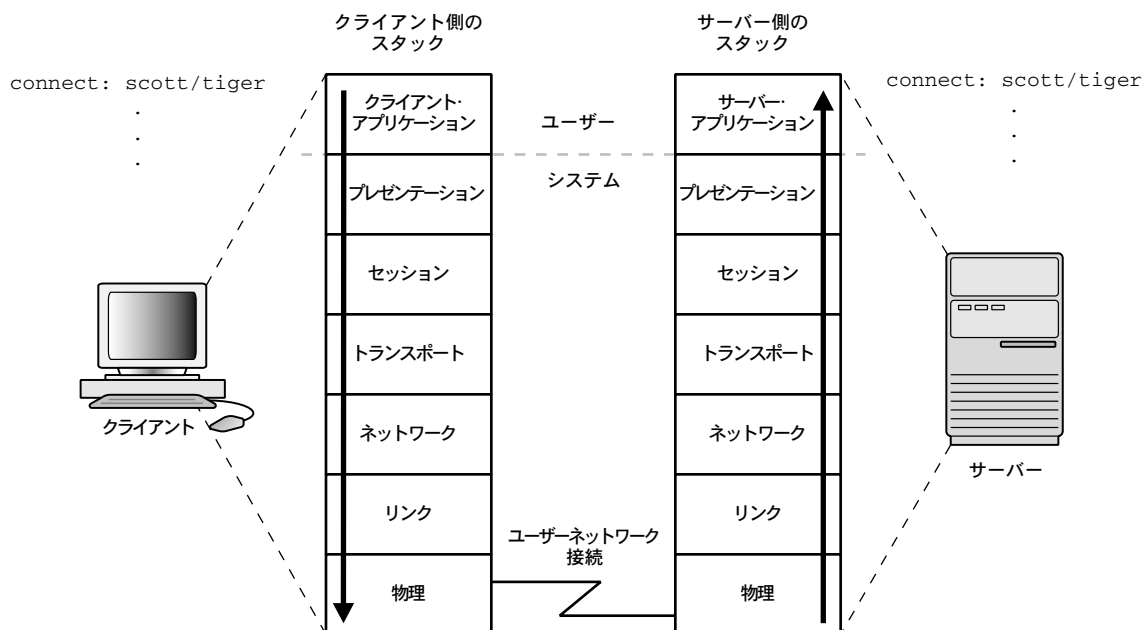
## スタック通信の概要

分散処理の概念は、設計と物理的な位置の両方で区別されているコンピュータ相互の通信と対話を行うための能力に依存しています。これは、スタック通信と呼ばれるプロセスで行われます。

スタック通信は、**Open Systems Interconnection (OSI)** モデルを参照することで説明できます。OSI モデルでは、いくつかのコードのレイヤーを通じてあるノードから別のノードに情報が渡されるスタックのような形式で、コンピュータ間の通信が行われます。3-3 ページの図 3-1 は、一般的な OSI プロトコル通信スタックを示したものです。



図 3-1 OSI 通信スタック



情報はクライアント側のレイヤーを通して降りて行き、トランスポート用にパッケージされ、ネットワーク・メディアを通じて運ばれ、その情報がサーバー側の対応するレイヤーで変換、解釈されます。

代表的な OSI プロトコル通信スタックには、こうしたレイヤーが7つあります。

**クライアント・アプリケーション** ユーザーに最も近い OSI レイヤーであるため、ユーザーが要求した機能により異なります。たとえば、データベース環境では、フォーム・アプリケーションがサーバーからのデータにアクセスするために通信を開始しようとする場合があります。

**プレゼンテーション** データの表現が、アプリケーション・レイヤーとセッション・レイヤーが処理できる形式になっていることを保証します。このレイヤーでは、クライアントとサーバー間で転送されたデータの構文と意味を追跡します。必要であれば、プレゼンテーション・レイヤーが共通のデータ形式を使用して、複数のデータ表現形式間で変換を行います。

**セッション** 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、管理および終了します。これは、データの要求と応答を伝送する仮想パイプです。セッション・レイヤー



は、データ通信を同時に両方向に送るか（非同期伝送）、一度に 1 つの方向のみに送るか（同期伝送）を制御します。

**トランスポート** データが確実に転送されるようにするデータ・トランスポートをインプリメントします。

**ネットワーク** 一連の相互接続サブネットワークによって、最適なパスでデータ・トランスポートが送られることを保証します。

**リンク** 物理リンクを通じて、信頼性の高いデータ転送を行います。

**物理** クライアントとサーバーの間の物理リンクをアクティブにし、メンテナンスし、非アクティブにするための、電氣的、機械的、手順的な仕様を定義します。

## 代表的な Net8 クライアント / サーバー環境でのスタック通信

スタック通信により、Oracle のクライアントとサーバーは、それらの間でデータを共有、変更、処理できます。代表的な Oracle 通信スタック内のレイヤーは、標準 OSI 通信スタックのレイヤーに類似しています。この項の内容は、次のとおりです。

- [Net8 クライアント側の対話](#)
- [サーバー側の対話](#)

### Net8 クライアント側の対話

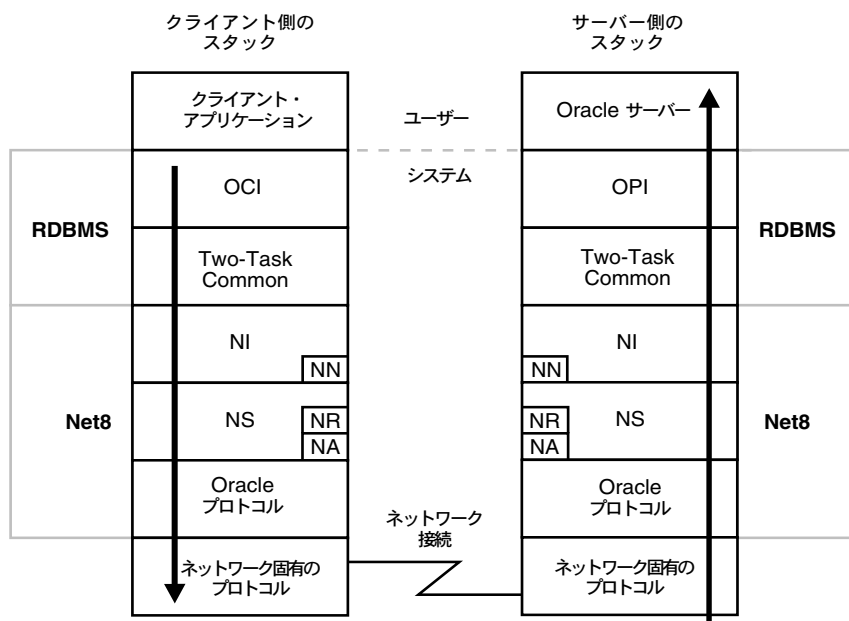
Oracle クライアント / サーバーのトランザクションでは、情報が次のレイヤーを通過します。

- [クライアント・アプリケーション](#)
- [Oracle コール・インタフェース \(OCI\)](#)
- [Two-Task Common \(TTC\)](#)
- [Net8](#)
- [ネットワーク固有のプロトコル](#)

図 3-2 は、Oracle ネットワーク環境でのクライアント / サーバー通信スタックを示したものです。



図 3-2 Oracle 環境での標準的な通信スタック



## クライアント・アプリケーション

Oracle Client Applications は、文字またはグラフィック表示、スクリーン制御、データ・プレゼンテーション、アプリケーション・フローなど、すべてのユーザー指向アクティビティを提供します。アプリケーションは、サーバーに送るデータベース操作を識別し、Oracle コール・インタフェース（OCI）を通じてその操作を渡します。

## Oracle コール・インタフェース（OCI）

**Oracle コール・インタフェース（Oracle Call Interface : OCI）** コードには、クライアントとサーバーの間で SQL ダイアログを開始するために必要な情報がすべて含まれています。OCI コードは、次のことをサーバーに対して行うコールを定義します。

- SQL 文を構文解析する
- SQL 文用にカーソルをオープンする
- クライアント・アプリケーション変数をサーバー共有メモリにバインドする
- サーバーのデータ・ディクショナリ内の値に基づいて、戻されるフィールドの内容を詳解する



- カーソル・メモリ領域内で SQL 文を実行する
- 1 行以上のデータをクライアント・アプリケーションにフェッチする
- カーソルをクローズする

クライアント・アプリケーションは、これらのコールの組合せを使用して、サーバー内のアクティビティを要求します。クライアント・アプリケーションの性質に応じて、OCI コールはサーバーに対する 1 つのメッセージに組み合わせたり、複数のメッセージで一度に 1 つずつ処理することができます。Oracle の製品は、多数の OCI コールを 1 つのメッセージに組み合わせて、サーバーに送信されるメッセージ数を最小限に抑えようとします。コールが実行されると、制御が Net8 に渡され、接続を確立して要求をサーバーに送信します。

**関連項目：** 『Oracle8i コール・インタフェース・プログラマーズ・ガイド』

Two-Task Common (TTC)

**Two-Task Common (TTC)** とは、プレゼンテーション・レイヤーの Oracle による実装です。TTC は、クライアントとサーバー間のキャラクタ・セットの相違または形式の相違に対して、キャラクタ・セットとデータ型の変換を行います。このレイヤーは、接続ごとに要求されたときのみ変更を行うように最適化されています。

初期の接続時に、TTC は内部データとキャラクタ・セットの表現の違いを評価したり、2 つのコンピュータが通信するために変換が必要かどうかを判断します。

Net8

Net8 は、Oracle 通信スタックにおけるセッション・レイヤーおよびトランスポート・レイヤーの機能をすべて備えています。Net8 は、クライアント・アプリケーションとサーバー間における接続の確立および維持、両者間のメッセージ交換の役割を担っています。Net8 には、セッション機能を業界標準プロトコルにマッピングする役割もあります。

Net8 には、セッション・レイヤーとトランスポート・レイヤーの機能を促進する、次の 3 つのコンポーネント・レイヤーがあります。

コンポーネント	説明
ネットワーク・インタフェース (Network Interface: NI)	<p>このレイヤーには、Oracle のクライアントまたはサーバー、外部プロセスが Net8 の機能にアクセスするための汎用インタフェースがあります。NI レイヤーは、接続のブレイク要求およびリセット要求を処理します。</p> <p>NI は、ネットワーク・ネーミング (NN) を使用して、名前を接続記述子に解決します。</p>



コンポーネント	説明
ネットワーク・セッション (Network Session: NS)	<p>このレイヤーは NI から要求を受け取り、サーバーまたは接続先の位置（オープン、クローズの各機能）、接続に 1 つ以上のプロトコルを必要とするかどうか（オープン、クローズの各機能）、クライアントとサーバー間の割込みをそれぞれの機能に基づいて処理する方法（送信、受信の各機能）など、マシン・レベルの一般的な接続に関する問題をすべて解決します。</p> <p>NS は、ネットワーク・ルート（NR）を使用してネットワーク・セッションを接続先に転送し、ネットワーク認証（NA）を使用して接続先との認証要求のネゴシエーションを行います。</p>
Oracle プロトコル (Oracle Protocol)	<p>Oracle プロトコルとは、Oracle によるトランスポート・レイヤーの実装です。Oracle プロトコルは、クライアント / サーバー接続で使用する業界標準プロトコルに NS の機能をマッピングする役割を担っています。各プロトコルは、NS と特定のプロトコルとの間で等価な機能をマップします。Oracle プロトコルには次のプロトコルが含まれます。</p> <ul style="list-style-type: none"><li>■ TCP/IP</li><li>■ SSL 付き TCP/IP</li><li>■ SPX</li><li>■ Named Pipes</li><li>■ LU6.2</li></ul>

## ネットワーク固有のプロトコル

すべての Oracle ソフトウェアは、クライアント / サーバー接続処理の際、2 台のマシンのトランスポート・レイヤー間でマシン・レベルの接続を確立するために、既存のネットワーク・プロトコル・スタックを必要とします。ネットワーク・プロトコルは、クライアント・マシンからサーバー・マシンまで、データを送ることによりのみ責任があります。その時点で、データがサーバー側の Oracle プロトコルに渡されます。

## サーバー側の対話

ネットワーク・プロトコルを通じてクライアント・アプリケーションから渡された情報は、サーバー側にある同様の通信スタックで受信されます。サーバー側でのプロセス・スタックは、クライアント側でのプロセス・スタックの逆になり、情報は通信レイヤーを通してさかのぼります。サーバー側に固有な操作の 1 つは、ネットワーク・リスナーを通じて初期の接続を受信することです。

ネットワーク・セッション・レイヤーの上位にある次のコンポーネントは、クライアント側のコンポーネントとは異なります。



- Oracle プログラム・インタフェース (OPI)
- Oracle サーバー

### Oracle プログラム・インタフェース (OPI)

**Oracle プログラム・インタフェース (Oracle Program Interface: OPI)** は、OCI の機能に対して補完的な機能を実行します。OPI は、OCI が送った各メッセージに応答します。たとえば、OCI が 25 行のデータのフェッチを要求すると、OPI はフェッチした 25 行のデータを OCI に戻します。

### Oracle サーバー

接続の Oracle サーバー側は、クライアント OCI からの要求を受信したり、クライアント・アプリケーションのかわりに SQL 文を解析します。要求を受信すると、処理されたデータが応答をフォーマットしてクライアント・アプリケーションに戻すため OPI に渡されます。

## サーバー / サーバー間の対話

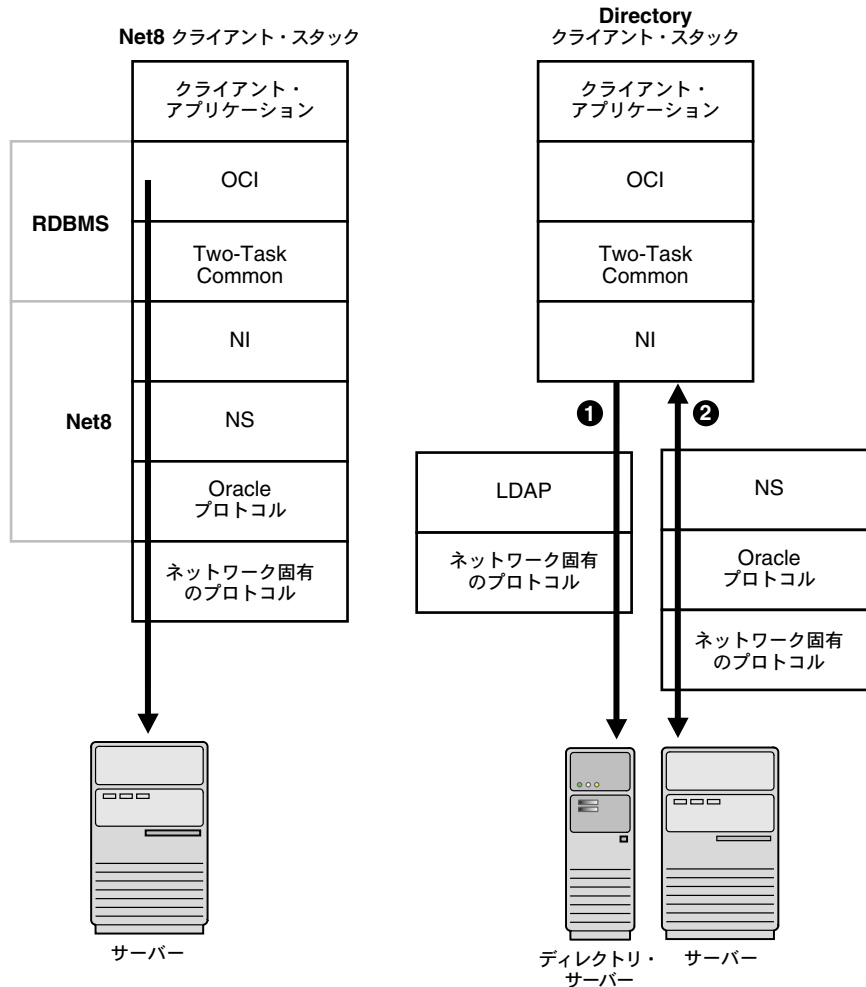
分散トランザクションを実行するために 2 つのサーバーが通信する場合、サーバーはクライアント・アプリケーションとして機能します。サーバーには、**ネットワーク・プログラム・インタフェース (Network Program Interface: NPI)** と呼ばれるサーバー専用の OCI パージョンがあります。NPI インタフェースは、OCI がクライアントに対して実行する機能をすべて実行でき、調整サーバーが追加サーバーに対する SQL 要求を構築できるようにします。

## ディレクトリ・サーバーに接続しているクライアントのスタック通信

LDAP 準拠のディレクトリ・サーバーにアクセスするクライアントは、[図 3-3](#) に示すように、新たに追加される LDAP レイヤーを通じてディレクトリに到達します。ディレクトリ・サーバーは、従来のクライアント・スタックを通じて情報を戻します。



図 3-3 通常のクライアントと Net8 LDAP クライアントの通信スタック比較



## JDBC クライアントのスタック通信

Oracle の Java Database Connectivity (JDBC) ドライバを利用すると、Java プログラムは Oracle データベースにアクセスできます。Oracle には、2 つの JDBC ドライバが用意されています。



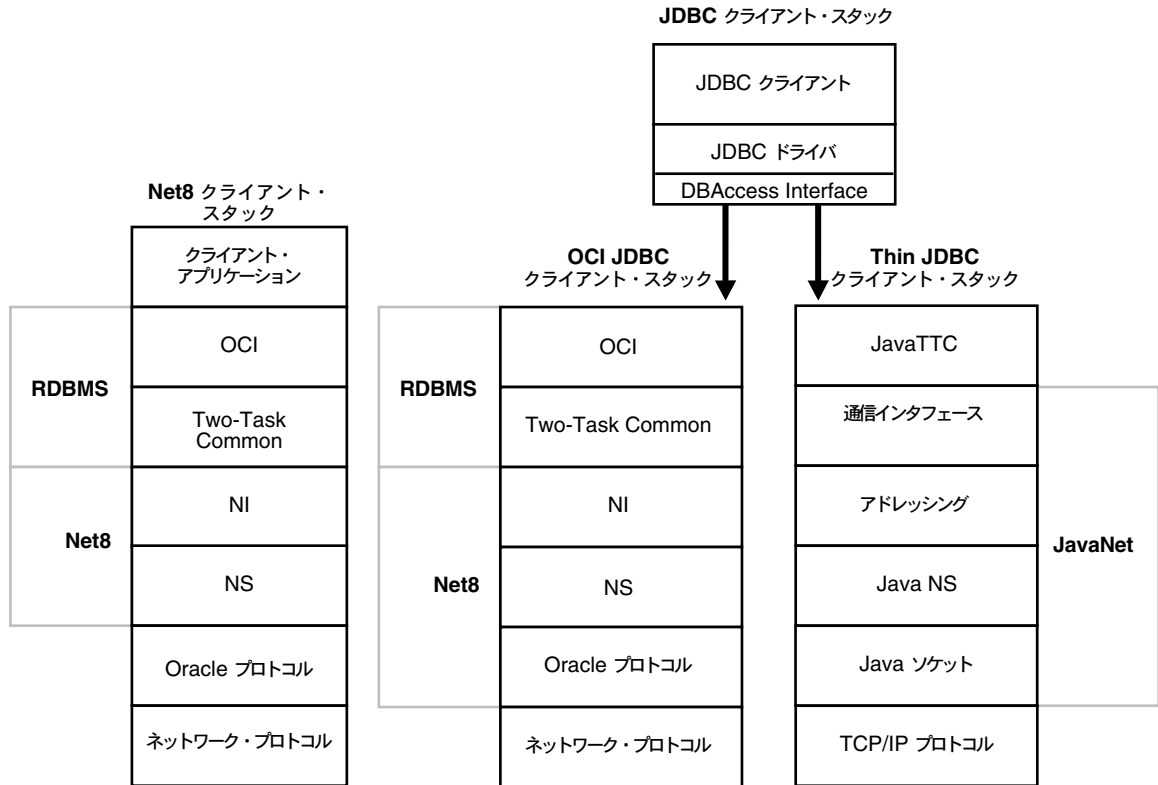
- JDBC/OCI は、Oracle の Level 2 JDBC ドライバです。これは、クライアント / サーバー型 Java アプリケーションのプログラマと、Java ベースの中間層の開発者を対象としています。JDBC/OCI ドライバは、JDBC の呼び出しを Oracle Call Interface (OCI) の呼び出しに変換します。変換された呼び出しは、Net8 を通して Oracle データベース・サーバーに送信されます。
- Thin JDBC ドライバは、Oracle の Level 4 ドライバです。これは、Java アプレットおよび Java アプリケーションの開発者を対象に設計されています。JDBC ドライバは、Java ソケットを通して Oracle データベース・サーバーに直接接続を確立します。データベースへのアクセスは、Two-Task Common (TTC) と Net8 の軽量実装に支援されます。

**関連項目：**『Oracle8i JDBC 開発者ガイドおよびリファレンス』

JDBC クライアント通信スタック内のレイヤーは、図 3-4 に示すように、標準の Oracle 通信スタックのレイヤーに類似しています。



図 3-4 通常のクライアントと JDBC クライアントの通信スタック



**注意：** サーバー側のスタックは、3-5 ページの図 3-2 と同じです。

JDBC/OCI クライアント・スタックは、標準的な Net8 クライアント・スタックに類似しています。Thin JDBC クライアント・スタックには、別にインプリメントされた類似コンポーネントがあります。Java クライアント・アプレット / サーバーのトランザクションでは、情報が次のレイヤーを通過します。

- DBAccess Interface
- JavaTTC
- JavaNet



- TCP/IP プロトコル

### DBAccess Interface

Java DBAccess Interface レイヤーを利用すると、クライアントは、標準的なクライアント通信スタックにおける **Oracle コール・インタフェース (OCI)** の場合と同様に SQL セッションを開始できます。

### JavaTTC

JavaTTC は Java クライアントとデータベース間で情報を交換する際に必要な、**Two-Task Common (TTC)** の実装のサブセットを提供します。JavaTTC は次の動作を行います。

- プロトコルのバージョンとデータ型を折衝する
- Java クライアントとサーバー間でキャラクタ・セットの変換が必要かどうかを判断する
- SQL 文を実行する

### JavaNet

JavaNet は JDBC クライアントが Oracle データベースに接続できるよう通信基盤を提供します。Net8 における NI、NR/NN/NA および TNS に相当します。

- 通信インタフェース – **JavaTTC** と他のネットワーク・レイヤー間のインタフェースを提供します。標準クライアント通信スタックにおける NI と同様に働きます。
- アドレッシング – Net8 の接続先アドレスを処理します。標準クライアント通信スタックにおける NN と同様に働きます。
- Java NS – TCP/IP に対する共通のインタフェースを提供し、サーバーまたは接続先の位置（オープン、クローズの各機能）、接続時に 1 つ以上のプロトコルを必要とするかどうか（オープン、クローズの各機能）、クライアントとサーバー間の割込みをそれぞれの機能に基づいて処理する方法（送信、受信の各機能）など、マシン・レベルの一般的な接続に関する問題をすべて解決します。
- Java Sockets – Java NS と TCP/IP 間の接続を提供します。

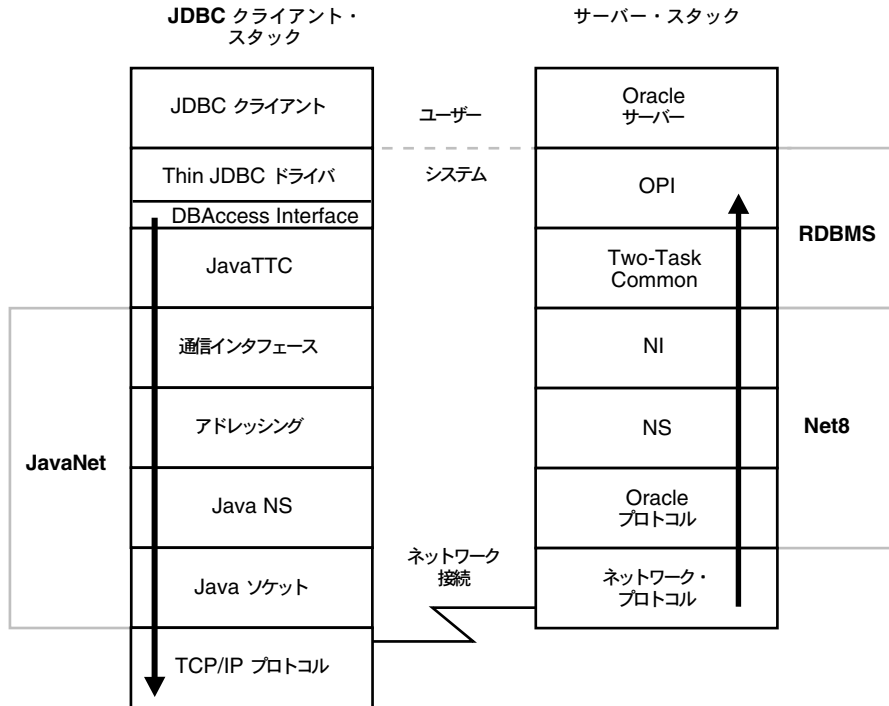
### TCP/IP プロトコル

TCP/IP プロトコル – Java NS は TCP/IP を使用するため、TCP/IP 上でしか動作しません。

**図 3-5** は、Java クライアント・アプレットと Oracle データベース間の通信スタックを示したものです。



図 3-5 JDBC 環境における Java クライアントとサーバーの通信スタック



**注意：** JDBC クライアントは、『Oracle8i JDBC 開発者ガイドおよびリファレンス』で説明するように、[ディレクトリ・ネーミング \(directory naming\)](#) を使用するよう構成できます。

## Oracle8i JServer クライアントのスタック通信

Oracle8i では、**Oracle8i JServer** によって Java をサポートしています。Oracle8i JServer のサポート対象には、Java ストアド・プロシージャ、JDBC、SQLJ、Common Object Request Broker Architecture (CORBA) および Enterprise JavaBeans (EJB) があります。

**関連項目：** Oracle8i JServer の概要については、『Oracle8i Java 開発者ガイド』を参照してください。

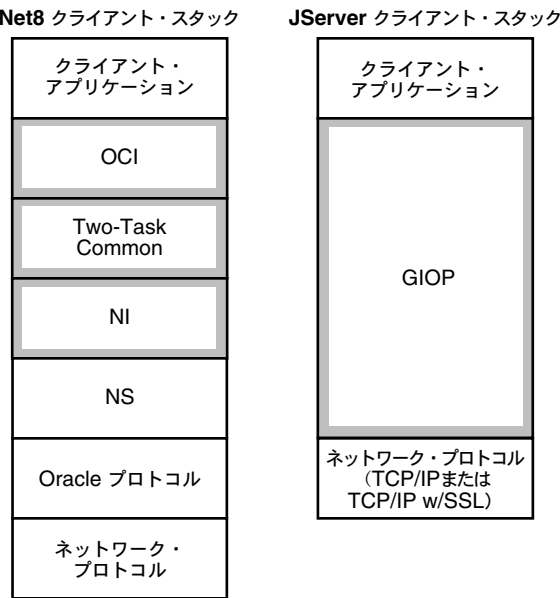


Oracle8i JServer では、Java Virtual Machine (VM) 要求に対して General Inter-Orb Protocol (GIOP) プレゼンテーションをサポートしています。GIOP は、Java VM 中の Enterprise JavaBeans および CORBA Servers にアクセスしているクライアントが使用します。

図 3-6 に示すように、EJB クライアントおよび CORBA クライアントは、標準の Net8 クライアント・スタックとは異なる通信スタックを使用します。差異は次のとおりです。

- プレゼンテーション・レイヤーとしての GIOP
- セッション・レイヤーを使用しない

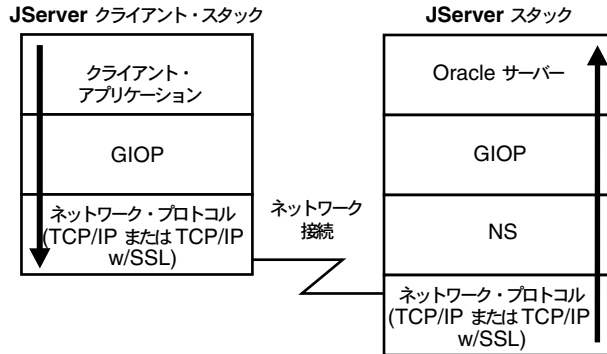
図 3-6 通常の Net8 クライアントと Oracle8i JServer クライアントの通信スタック比較



サーバー側では、通常の Net8 接続で必要とされる多くの Net8 通信レイヤーを必要としません。かわりにサーバー側では、TCP/IP のネットワーク・プロトコルと TCP/IP または SSL 付き TCP/IP の Oracle プロトコルを必要とします。Net8 が必要とするコンポーネントは、ネットワーク・セッション (Network Session: NS) のみです。図 3-7 は、クライアントと Oracle データベース内の Oracle8i JServer オプション間の通信スタックを示したものです。



図 3-7 Oracle8i JServer 環境でのクライアント / サーバー通信スタック

**関連項目：**

- EJB アプリケーションおよび CORBA アプリケーションに対するクライアント接続の構成については、『Oracle8i Enterprise JavaBeans と CORBA 開発者ガイド』を参照してください。
- Java ストアド・プロシージャに対するクライアント接続の構成については、『Oracle8i Java ストアド・プロシージャ開発者ガイド』を参照してください。







---

## Net8 製品およびツール

この章では、Net8 について紹介し、その主なアプリケーション、機能および機能性の概要を示します。内容は次のとおりです。

- [Net8 Client](#)
- [Net8 Server](#)
- [Oracle プロトコル](#)
- [Oracle Connection Manager](#)
- [Oracle Names](#)
- [外部ネーミング・サービス](#)
- [認証方式 \(authentication method\)](#)
- [Oracle Advanced Security](#)
- [管理ツール](#)

---

**注意：** 製品が使用できるかどうかは、インストールの種類やオペレーティング・システムによって異なります。詳細は、使用しているオペレーティング・システムに固有のドキュメントを参照してください。

---



## Net8 Client

Net8 Client を使用すると、データベースへのクライアント接続がネットワークを介してできるようになります。クライアント側アプリケーションから Net8 に送られた接続要求は、ネットワークを経由してサーバーに渡されます。

## Net8 Server

Net8 Server を使用すると、リスナーは、ネットワーク上のクライアント側アプリケーションの接続を、プロトコルを介して受け入れることができます。

## Oracle プロトコル

Oracle では、次のプロトコルがサポートされています。

- TCP/IP
- SSL 付き TCP/IP
- SPX
- Named Pipes
- LU6.2
- Bequeath

## TCP/IP

Transmission Control Protocol/Internet Protocol (TCP/IP) は、ネットワークを介したクライアント / サーバー対話に使用される、事実上標準のイーサネット・プロトコルです。TCP/IP により、クライアント上の Oracle アプリケーションは、TCP/IP を介してリモートの Oracle データベースと通信できるようになります (TCP/IP を使用したネットワーク通信をサポートするホスト・システム上で Oracle データベースが動作している場合)。

## SSL 付き TCP/IP

**Secure Sockets Layer (SSL)** プロトコル付き TCP/IP により、クライアント上の Oracle アプリケーションは、SSL 付き TCP/IP を介してリモートの Oracle データベースと通信できるようになります (TCP/IP および SSL を使用したネットワーク通信をサポートするホスト・システム上で Oracle データベースが動作している場合)。SSL 付き TCP/IP を使用するためには、**Oracle Advanced Security** が必須となります。

SSL は、証明書や秘密鍵などの認証データを Oracle Wallet に格納します。クライアントがサーバーと Net8 接続を開始すると、SSL は (証明書を使用して) 両者間のハンドシェイクを実行します。ハンドシェイクの実行中、次の処理が行われます。



- クライアントとサーバーは、交換するメッセージに適用するための認証、暗号化方法およびデータ整合性タイプのセットである Cipher Suite についてネゴシエーションを行います。
- 使用する構成によっては、サーバーはクライアントの公開鍵で暗号化したメッセージにサーバー自身の証明書を入れてクライアントに送信します。サーバーは、この同じメッセージを使用して、クライアントの証明書を送るように要求する場合があります。クライアントは、それ自身の秘密鍵を使用してこのメッセージの暗号化を解除し、サーバーの証明書に認証局のシグネチャがあることを確認します。
- 必要に応じてクライアントは、このユーザーの証明書を Oracle サーバーに送信します。証明書によって、ユーザーの情報が正しいことと、公開鍵が実際にそのユーザーのものであることが保証されます。サーバーはこのユーザーの証明書を調べて、認証局のシグネチャがあることを確認します。

**関連項目：** 『Oracle8i Advanced Security 管理者ガイド』

## SPX

Sequenced Packet Exchange (SPX) プロトコルでは、SPX/IPX を使用するネットワーク上でクライアント / サーバー対話が可能です。クライアント上の Oracle アプリケーションは SPX/IPX を介してリモートの Oracle データベースと通信できるようになります (Oracle データベースが、SPX/IPX を使用したネットワーク通信をサポートするホスト・システム上で実行されている場合)。このプロトコルは、Novell の NetWare で広く使用されています。

## Named Pipes

**Named Pipes プロトコル**は、クライアント / サーバー (分散アプリケーション) 間でプロセス間通信を提供する高水準のインタフェースです。一方のプロセス (サーバー側アプリケーション) でパイプを作成し、他方のプロセス (クライアント側) が名前を使用してオープンします。これによって互いに、一方が書き込む情報を他方が読み取ることができます。Named Pipes は、特に PC LAN 環境で使用することを念頭に設計されています。

Named Pipes では、Named Pipes を使用するネットワーク上でクライアント / サーバー対話が可能です。クライアント上の Oracle アプリケーションは Named Pipes を介してリモートの Oracle データベースと通信できるようになります (Oracle データベースが、Named Pipes を使用したネットワーク通信をサポートするホスト・システム上で実行されている場合)。

## LU6.2

Logical Unit Type 6.2 (LU6.2) プロトコルは、IBM Advanced Program-to-Program Communication (APPC) アーキテクチャの一部を構成しています。



APPC は、IBM の peer-to-peer（プログラム間）プロトコルで、System Network Architecture (SNA) ネットワークで使われます。SNA は、国際標準化機構 (ISO) が策定した Open Systems Interconnect (OSI) モデルに類似した IBM の参照モデルです。

APPC アーキテクチャでは、(端末 / ホスト間プロトコルのように) クライアントで端末エミュレーションを実行することなく、SNA ネットワークを介してクライアントとホストが通信できます。APPC アーキテクチャでは peer-to-peer 通信が可能なため、サーバーとの通信をクライアント側から開始できます。

LU6.2 プロトコルと Physical Unit Type 2.1 (PU2.1) プロトコルを使用する SNA ネットワークでは、APPC が提供されます。LU6.2 プロトコルは 2 つのアプリケーション・プログラム間のセッションを定義します。LU6.2 は、製品に依存しないタイプの LU です。

LU6.2 を使用すると、PC 上の Oracle アプリケーションは Oracle データベースと通信できるようになります。この通信は、SNA ネットワークを介して、APPC をサポートするホスト・システム上で動作する Oracle データベースとの間で発生します。

## Bequeath

**Bequeath プロトコル**を使用すると、サーバーと同じマシン上に存在するクライアントは、リスナーを使用せずにデータベースから情報を取り出すことができます。Bequeath プロトコルでは、専用サーバー・プロセスがクライアント・アプリケーションごとに内部で生成されます。ある意味で bequeath プロトコルは、リモートのネットワーク・リスナーが接続を処理するのと同じ方法でローカルの接続を処理します。

Bequeath プロトコルは、ローカル接続で使われます。この接続では、SQL\*Plus などの Oracle クライアント・アプリケーションが、同じマシン上で動作している Oracle サーバーと通信します。

## Oracle Connection Manager

---

**注意：** Oracle Connection Manager は、Oracle8i Enterprise Edition をインストールすると使用できるようになります。

---

**Oracle Connection Manager** は、クライアント接続要求をその次のホップに送るか、または直接データベース・サーバーに送る際の経路となるルーターの役割を果たします。Connection Manager を介して接続要求を送信するクライアントは、送信後、その Connection Manager で構成された接続集中化、Net8 アクセス制御、複数プロトコル・サポートの各機能を利用できます。



# Oracle Connection Manager プロセス

Oracle Connection Manager は、クライアントからの着信要求のリスニングを行って、宛先サービスへの接続要求を開始します。Oracle Connection Manager は、次の 2 つのプロセスとの連携により、これらのタスクを実行します。

プロセス	説明
CMGW (Oracle Connection Manager ゲートウェイ・プロセス)	<p>Oracle Connection Manager のハブとして機能するゲートウェイ・プロセスです。このプロセスは、次の役割を担います。</p> <ul style="list-style-type: none"><li>■ CMADMIN プロセスへの登録。</li><li>■ 着信接続要求のリスニング。デフォルトでは、TCP/IP、ポート 1630 でリスニングします。</li><li>■ クライアントのリスナーに対する接続要求の開始。</li><li>■ クライアントとサーバー間のデータ中継。</li><li>■ Oracle Connection Manager 制御ユーティリティ (CMCTL) によって開始された要求への応答。</li></ul>
CMADMIN (Oracle Connection Manager 管理プロセス)	<p>CMADMIN は、Oracle Connection Manager の管理上の問題点すべてに対して責任を果たすマルチスレッド・プロセスです。このプロセスは、次の役割を担います。</p> <ul style="list-style-type: none"><li>■ CMGW 登録の処理。</li><li>■ 1 つ以上のデータベース・インスタンスをサービスする全リスナーの識別。</li><li>■ CMGW とリスナーに関するソース・ルート・アドレス情報の登録。</li><li>■ Oracle Names サーバーに登録されたリスナーの監視。および、ソース・ルート・アドレス。</li><li>■ Oracle Names サーバーの検索。</li><li>■ SQL*Net 2.x クライアントおよび Net8.x クライアントについて、Oracle Names サーバーでのアドレス情報のメンテナンス。</li><li>■ Oracle Names サーバーへ要求を定期的に送信して、Oracle Names サーバーの、使用可能なサービスについてのキャッシュを更新。</li><li>■ CMCTL によって開始された要求への応答。</li></ul>

**関連項目：** CMCTL の使用については、A-77 ページの「[Oracle Connection Manager 制御ユーティリティ \(CMCTL\)](#)」を参照してください。



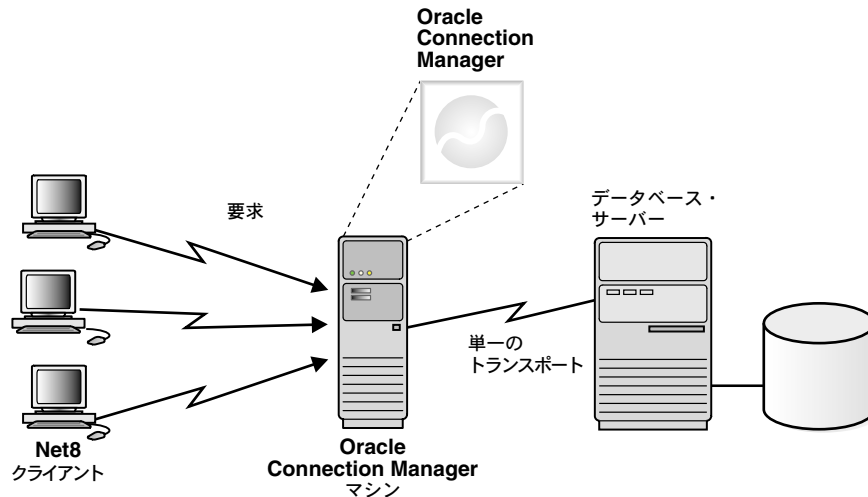
## 接続集中化

Oracle Connection Manager を使用すると、マルチスレッド・サーバー接続先に対する単一トランスポート・プロトコル接続を介して、複数のクライアント・ネットワーク・セッションを多重化または集中化できます。

集中化では、サーバーが着信要求に使用する接続終点の数を少なくすることにより、2つのプロセス間で複数の接続を維持するために必要なリソースを削減します。このため、サーバーが処理できるネットワーク・セッションの総数を増加できます。複数の Connection Manager を使用して、数千のユーザーを同時にサーバーに接続できます。

図 4-1 は、接続集中化の様子を示しています。

図 4-1 Oracle Connection Manager による接続集中化



**関連項目：** 構成に関する情報は、8-25 ページの「[接続集中化機能の使用](#)」を参照してください。

## Net8 アクセス制御

Oracle Connection Manager には、TCP/IP 環境で指定されたサーバーへのクライアント・アクセスを制御するときに使用できる機能も含まれています。次の基準に基づいて、サーバーへの特定のクライアント・アクセスを可能にするか、または制限するフィルタリング・ルールを設定できます。

- クライアントのソース・ホスト名または IP アドレス



- サーバーの接続先ホスト名または IP アドレス
- 接続先データベース・サービス名

**関連項目：** 構成に関する情報は、8-31 ページの「[Net8 アクセス制御の使用](#)」を参照してください。

## Net8 Firewall Proxy

Net8 Firewall Proxy を提供しているファイアウォール・ベンダーもあります。これは、アプリケーションのプロキシを必要とするファイアウォールにインストールされます。Net8 Firewall Proxy には、Oracle Connection Manager と同じアクセス制御機能があります。

---

---

**注意：** Net8 Firewall Proxy がサポートされているかどうかは、ファイアウォールのベンダーに問い合せてください。

---

---

## 複数プロトコル・サポート

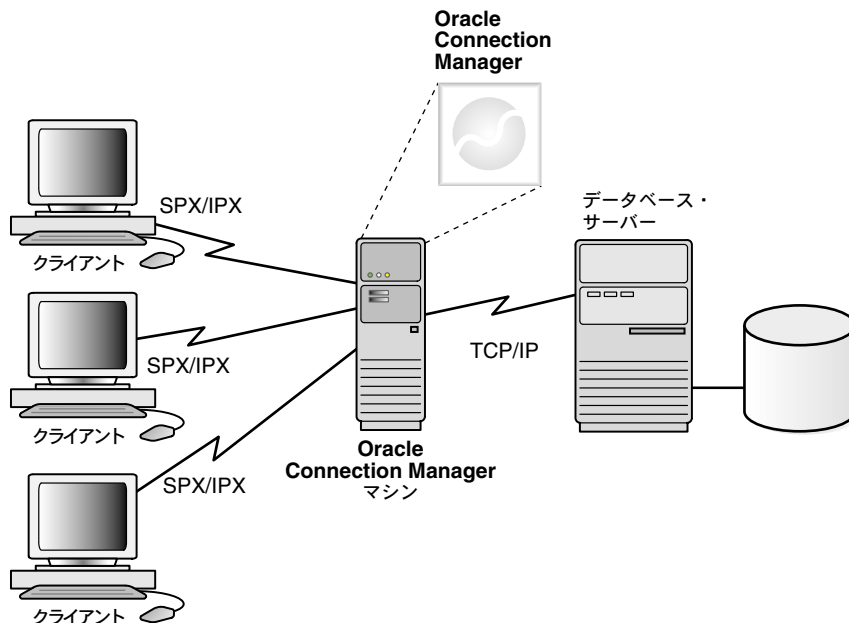
Oracle Connection Manager では、複数プロトコルもサポートされています。これにより、クライアントとサーバーは、異なるネットワーク・プロトコルで相互に通信できます。この機能は、SQL\*Net バージョン 2 Oracle MultiProtocol Interchange ですでに実現している機能を置換したものです。

Net8 は、インストールおよびサポートが可能な限り多くのネットワーク・プロトコル・スタックを横断できます。実際に、サポートされるネットワーク・プロトコル数の制限は、特定のノードのハードウェア、メモリーおよびオペレーティング・システムにある制限のみです。

[図 4-2](#) は、SPX ネットワーク内のクライアントが、Oracle Connection Manager を介して TCP/IP トランスポート上のサーバーにそのクライアントのネットワーク・セッションを送る様子を示しています。



図 4-2 Oracle Connection Manager による複数プロトコル・サポート



**関連項目：** 構成に関する情報は、8-28 ページの「[複数プロトコル・サポートの使用](#)」を参照してください。

## Oracle Names

**Oracle Names** は、グローバルなクライアント / サーバー・コンピューティング・ネットワークを簡単に設定、管理できるように開発された Oracle 環境用分散ネーミング・サービスです。

この項の内容は、次のとおりです。

- [概要](#)
- [管理リージョン](#)
- [ドメイン](#)
- [データ・リポジトリとしての Oracle Names](#)
- [Oracle Names サーバーに格納されるデータ](#)



- ネットワーク・コンポーネントの編成およびネーミング
- 管理責任を分散する複数のリージョンの使用
- Oracle Names のバージョン間の相違
- 検出の理解

## 概要

Oracle Names は、ディレクトリ・サービスのように連携して動作する **Oracle Names サーバー** の統合システムを作成およびメンテナンスします。このシステムは、ネットワーク上のすべてのサービスのアドレスを格納し、接続が必要なクライアントに対してそのアドレスを使用できるようにします。

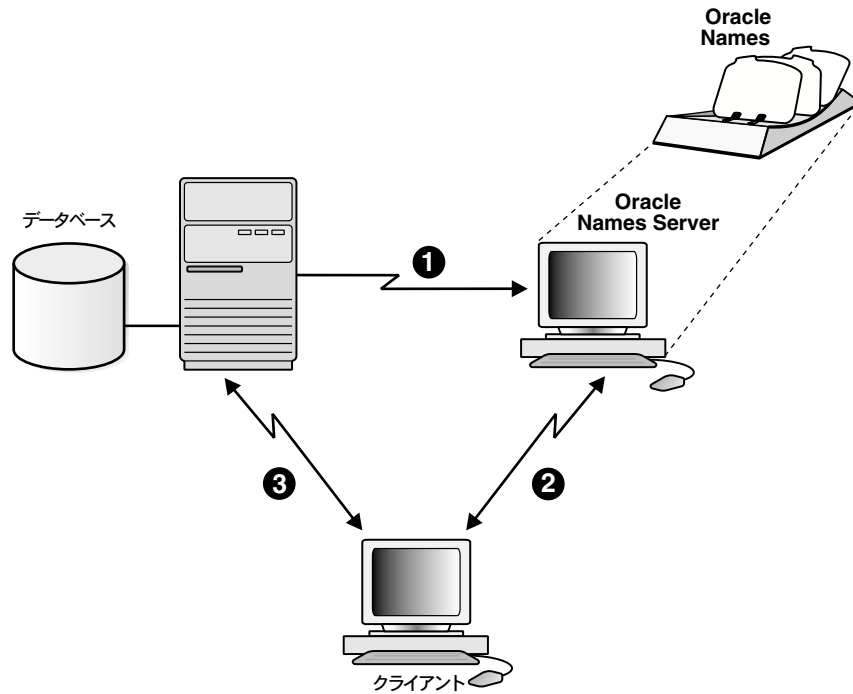
電話をかける人が電話番号を調べるために電話番号案内を利用するように、Oracle Names を使用するよう構成されたクライアントは、その接続要求を Oracle Names サーバーに照会します。Oracle Names サーバーは、クライアントから示されたサービス名をネットワーク・アドレスに解決しようとします。Oracle Names サーバーは、ネットワーク・アドレスを検出すると、その情報をクライアントに戻します。こうしてクライアントはサービスに接続するためにそのアドレスを使用できます。

図 4-3 は、クライアントとサーバー間の接続の確立を支援する Oracle Names の動作を示しています。

1. データベースは、そのサービスを Oracle Names サーバーに登録します。
2. クライアントは、サービスをネットワーク上に見つけるため Oracle Names サーバーに接続してネットワーク・アドレスを取り出します。
3. クライアントは、その後、対象のサービスに自動的に接続されます。



図 4-3 Oracle Names



**関連項目：** 構成に関する情報は、6-43 ページの「[Oracle Names メソッドの構成](#)」を参照してください。

## 管理リージョン

大半のネットワークには、1つの管理中心点、すなわち、1つの**管理リージョン (administrative region)** があります。管理リージョンは、ネットワークのサービスを管理する Oracle Names サーバーの集まりです。接続情報はすべて、単一のデータ・リポジトリに格納されます。データ・リポジトリには、サービス名を解釈する権限があります。管理リージョン内のすべての Oracle Names サーバーは、このデータ・リポジトリから情報を問い合わせます。管理リージョンが格納のためにデータベースを使用する場合、1つの管理リージョンに対して1つのデータベースが存在します。Oracle Names サーバーの数に制限はありません。Oracle Names は、1つ以上の管理リージョンをサポートします。

複数のデータ・センターと多数の Oracle インスタンスが存在する大規模な企業環境では、複数管理リージョンの利用が選択されています。これにより、各データ・センターは、そのセンター専用の環境でサービスを独自に定義、管理できます。同時に、すべてのサービス・



アドレスは、環境全体のすべてのクライアントに対して継続的に使用可能となります。Oracle Names サーバーは、外部の管理リージョンにあるクライアントから適切な Oracle Names サーバーに対して、名前解決要求を透過的に送ります。

## ドメイン

**ドメイン (domain)** は、マシンおよびネットワーク・サービスの論理グループです。各ドメインの中では、すべての名前が一意である必要があります。ただし、異なるドメイン間では名前が重複してもかまいません。

管理リージョンには 1 つ以上のドメインがあり、これにより管理責任が分割されます。

ネットワーク・ドメインは、階層的なものであるという点で、多くのオペレーティング・システムで使用されているファイル・ディレクトリに似ています。ただし、ファイル・システムと違い、ネットワーク・ドメインはデータベースまたはネットワーク内の他のオブジェクトの物理的配置に対応している場合としていない場合があります。ネットワーク・ドメインは、名前空間の矛盾を防ぐために開発された名前空間にすぎません。

---

**注意：** 似ているように見えますが、Oracle のネットワークのドメインと Domain Name System (DNS) の名前空間とは完全に独立しています。便宜的に、Oracle ネットワーク内に DNS ディレクトリ構造を反映するように選択できます。

---

## データ・リポジトリとしての Oracle Names

Oracle Names サーバーのデータは、リージョン内の全 Oracle Names サーバー間にある継続レプリケーションを通じて、または共通の Oracle データベースで書き込み / 読み込みを行うことによって更新されます。

全サービスが動的に登録されている比較的小規模なワークグループ環境では、管理者は個々の Oracle Names サーバーすべてに対しデータを継続的にレプリケートする Oracle Names サーバーを構成できます。リスナーが新しいサービスを登録すると、そのサービスに関する情報が管理リージョン内のその他の Oracle Names サーバーに即座に渡されます。

一方、大規模な環境の管理者は、通常、**リージョン・データベース (region database)** と呼ばれる単一の Oracle データベース内に登録データを格納します。リージョン・データベースは、Oracle Names 情報を格納するテーブルで構成されます。Oracle データベースをリポジトリとして使用するように Oracle Names サーバーが構成されている場合、すべてのサービス登録内容がそのデータベースに書き込まれます。所定の管理リージョンにある各 Oracle Names サーバーは、リージョン・データベースについて更新された登録内容の有無を定期的にポーリングします。この方法によって、所定の管理リージョン内にある全 Oracle Names サーバーに、新しい登録が適切な時期に伝達されます。同時に、Oracle Names サーバー同士が直接交信する必要がなくなり、高い信頼性が確保されます。



## Oracle Names サーバーに格納されるデータ

Oracle Names サーバーに格納されるデータの型を次に示します。

データ	説明
グローバル・データベース名 およびアドレス	Oracle Names サーバーは、グローバル・データベース名（データベース名およびドメイン） およびアドレスなどのデータベースに関する情報をリスナーから取り出します。アドレスは、listener.ora ファイルに構成されます。また、グローバル・データベース名はデータベースの起動時に登録されるか、listener.ora ファイルに静的に構成されます。この情報を登録する必要はありません。
その他の Oracle Names サー バーの名前およびアドレス	Oracle Names サーバーには、同じ管理リージョン内にあるすべての Oracle Names サーバーの名前およびアドレスが格納されます。ネットワーク内に複数の管理リージョンが存在する場合、Oracle Names サーバーには、ルート管理リージョンとその直属の各サブリージョンに存在する 1 つ以上の Oracle Names サーバーの名前およびアドレスが格納されます。この情報を登録する必要はありません。
ネット・サービス名	NAMESCTL 制御ユーティリティや Oracle Net8 Assistant を使用してネット・サービス名を登録すると、Oracle Names サーバーにこれらの名前が格納されます。Oracle Names サーバーには、非 Oracle データベースおよび Oracle Rdb データベースへのゲートウェイも格納されます。



データ	説明
グローバル・データベース・リンク	<p><b>データベース・リンク</b>を使用すると、データベース間の通信が可能になります。データベース・リンクの名前は、リンクが指すデータベースのグローバル・データベース名と同じです。通常、1つのデータベースにはデータベース・リンクは1つしか存在しません。</p> <p>次のタイプのデータベース・リンクが作成できます。</p> <ul style="list-style-type: none"> <li>■ データベースの特定のスキーマにある<b>プライベート・データベース・リンク (private database link)</b>。プライベート・データベース・リンクの所有者のみが使用できます。</li> <li>■ データベースの<b>パブリック・データベース・リンク (public database link)</b>。そのデータベースのすべてのユーザーが使用できます。</li> <li>■ Oracle Names サーバー内の<b>グローバル・データベース・リンク (global database link)</b>。ネットワーク内のすべてのユーザーが使用できます。</li> </ul> <p>Oracle Names はリスナーからグローバル・データベース名を取り出すため、グローバル・データベース名であるグローバル・データベース・リンクは Oracle Names サーバーに自動的に登録されます。したがって、この情報を登録する必要はありません。</p> <p>グローバル・データベース・リンクに対するユーザー名とパスワードからなる資格証明は、Oracle Net8 Assistant を使用して Oracle Names サーバーに登録できます。これらのグローバル・データベース・リンクには、Net8 Assistant を使用して定義されたリンク修飾子を追加できます。</p> <p>グローバル・データベース・リンクは、個々のユーザーが作成したプライベート・データベース・リンクやパブリック・データベース・リンクによって置き換えられることがあります。</p> <p><b>関連項目：</b>『Oracle8i 分散システム』</p>
別名	<p>Oracle Names サーバーには、定義済のネット・サービス名、データベース・サービスまたはグローバル・データベース・リンクに対する別名または代替サービス名が格納されます。別名は、NAMESTL 制御ユーティリティまたは Oracle Net8 Assistant を使用して Oracle Names サーバーに登録されます。</p>
Oracle Connection Manager	<p>Oracle Names サーバーには、ネットワーク上に存在するすべての Oracle Connection Manager の名前およびリスニング・アドレスが格納されます。この情報を登録する必要はありません。</p>



## ネットワーク・コンポーネントの編成およびネーミング

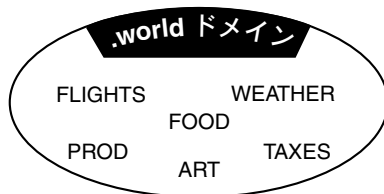
Oracle Names を使用する場合、ネットワーク化された環境にあるデータベースなどのオブジェクトには、ネットワーク内で名前が重複しないような方法でネーミングする必要があります。ネットワーク内のオブジェクトをネーミングする基本モデルは2つあります。

- シングル・ドメイン・モデル
- 階層ネーミング・モデル

### シングル・ドメイン・モデル

ネットワークが小規模で名前の重複がない場合は、シングル・ドメイン・ネーミング・モデルを使用するのが有効です。図 4-4 は、シングル・ドメイン名 .WORLD を使用した、一般的なフラット・ネーミング構造を示しています。

図 4-4 シングル・ドメイン・ネーミング・モデル



この環境では、データベース・サービス名に拡張子 .WORLD が自動的に付けられます（たとえば、PROD.WORLD、FLIGHTS.WORLD など）。

### 階層ネーミング・モデル

階層ネーミング・モデルは、名前を階層構造に分割して、ネットワークの規模拡張に備えたり、名前の独立性を高めることができます。このタイプのネーミング・モデルでは、異なるドメインに同じ単純名を持つ複数のデータベースを実現できます。

図 4-5 は、(ROOT) ドメイン、ACME ドメイン、US.ACME ドメイン、EUROPE.ACME ドメインおよび ROW.ACME (Rest of World) ドメインで構成されるドメインの階層構造を示しています。



図 4-5 階層ネーミング・モデル

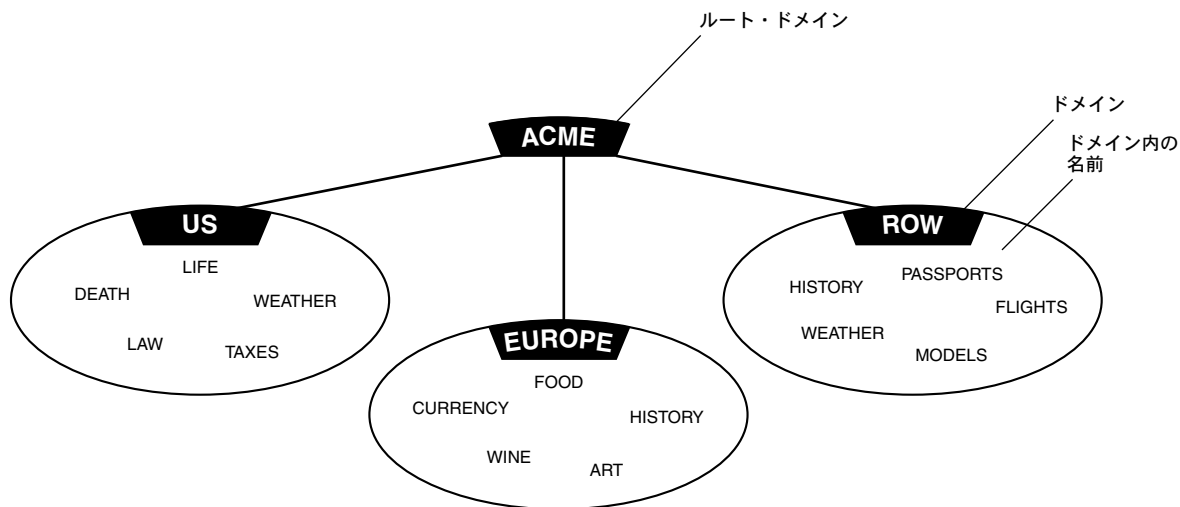


図 4-5 で、WEATHER と HISTORY はどちらも 2 つありますが、名前は一意です（すなわち、HISTORY.ROW.ACME と HISTORY.EUROPE.ACME）。

**デフォルト・ドメイン** デフォルト・ドメインは、クライアントの名前要求の大半を処理するドメインです。通常、これはクライアントが存在するドメインですが、クライアントが最も頻繁にサービスを要求する別のドメインを指定することもできます。クライアントがデフォルト・ドメイン内のネットワーク・サービスを要求するときは、サービスの単純な未修飾名を使用して、つまりドメイン名を指定しないでサービスを要求できます。文字 "." を含まない名前をユーザーが要求すると、要求したデータベース・サービス名またはデータベース・リンク名に、デフォルト・ドメイン名が自動的に追加されます。

たとえば、クライアントはデフォルト・ドメインの EUROPE.ACME.COM で構成されます。クライアントがサービス名 "WINE" を図 4-5 で要求すると、要求した名前にデフォルト・ドメイン名 EUROPE.ACME.COM が付加されるので、名前は WINE.EUROPE.ACME.COM となります。

**複数のドメイン** 複数のドメインは、一連の親子関係で、ルート・ドメイン（階層内の最高レベルのドメイン）に階層的に関連付けられています。たとえば、ルートの下にはいくつかのドメインがあり、その 1 つを COM と呼びます。COM ドメインの下にはいくつかのドメインがあり、その 1 つを ACME と呼びます。ACME ドメインの下には、いくつかのドメイン（US や EUROPE など）があります。



---

**注意：** SQL\*Net と Oracle Names の旧リリースでは、ドメインが 1 つのみのネットワークは、デフォルトで .world と呼ばれていました。これは Net8 と Oracle Names バージョン 8 の要件ではありませんが、旧リリースからのデータベースをすべて改名することを避け、下位互換性を維持するために同じ規約を保つことができます。

---

## 管理責任を分散する複数のリージョンの使用

Oracle Names を使用していて、かつネットワークが大規模であるか、または地理的に広く分散している場合は、管理リージョンを複数にすることを選択できます。たとえば、ネットワークにアメリカとヨーロッパの両方が含まれている場合、そのネットワークについて管理上の判断をローカルに下すことが必要な場合があります。リージョンを再分割するには、親から子またはサブリージョンに、リージョンとドメインを委任する必要があります。

管理リージョンを委任するには、それぞれの管理リージョンが 1 つ以上の異なるドメインを制御するように階層ネーミング・モデルの適用が必要です。

複数の管理リージョンを持つネットワークは、1 つの**ルート管理リージョン (root administrative region)** と 1 つ以上の**委任管理リージョン (delegated administrative region)** で構成されます。

### ルート管理リージョン

ルート管理リージョンには、**ルート・ドメイン (root domain)** が含まれます。ルート管理リージョンには次の情報が含まれています。

- ルート・リージョン内の Oracle Names サーバー。
- このリージョンで管理されるドメイン。ルート・ドメインが必ず含まれ、他のドメインが含まれる場合もあります。
- 委任管理リージョンの Oracle Names サーバー。ルートの直属の子リージョンの役割を果たす代替リージョンにあるドメインおよび Oracle Names サーバーのアドレス。
- ルート・リージョンのデータ定義。ルート管理リージョンに関連するすべてのデータベース・サービス名、データベース・リンクおよび別名。

### 委任管理リージョン

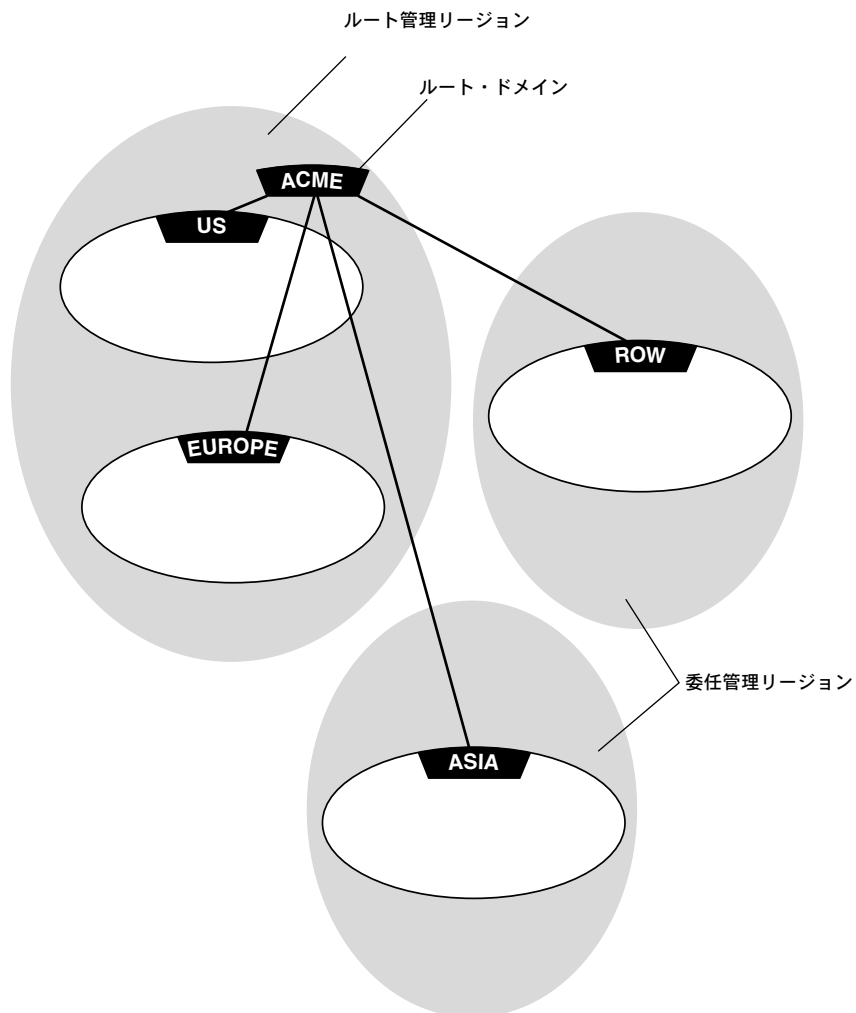
管理リージョンは、ネーミング・モデルの階層の最高位から、その他の下位ドメインに至るまで委任できます。たとえば、ドメイン数が 10 のネットワークでは、1 ～ 10 の管理リージョンで構成できます。

ルート以外の管理リージョンはすべて、階層構造内でルートと直接または間接的に連結している委任リージョンです。



図 4-6 は、5 つのドメインおよび3つの管理リージョンを持つネットワークを示しています。3つの管理リージョンは、ルートである ACME と、2つの委任リージョン（ROW、ASIA）です。

図 4-6 委任管理リージョン





## ルートの下の委任管理リージョン

ルートの下の管理リージョンはすべて委任管理リージョンです。委任管理リージョンは、ルート管理リージョンなどの他のリージョンから、ドメインの管理責任を受け取ります。委任管理リージョンには次の情報が含まれています。

- リージョン内のすべての Oracle Names サーバーとドメイン。
- この管理リージョンの子リージョン内のドメインと Oracle Names サーバーのアドレス。
- ルート・リージョン内の Oracle Names サーバーのアドレス。このデータによって、委任リージョン内の Oracle Names サーバーが（ルート・リージョンを介して）他のリージョンと交信することができます。
- データ定義。このローカル（委任）管理リージョン内のすべてのドメインに対するすべてのデータベース・サービス名、データベース・リンクおよび別名。

## Oracle Names のバージョン間の相違

現在の Oracle Names は、以前のバージョンと大きく異なります。

- [Oracle Names バージョン 1](#)
- [Oracle Names バージョン 2](#)
- [Oracle Names バージョン 8（このリリース）](#)

### Oracle Names バージョン 1

Oracle Names バージョン 1 では、管理者は Oracle Network Manager を使用して Oracle Names を構成し、すべてのトポロジー・データをデータベースに格納していました。リージョン内の Oracle Names サーバーはすべて同じデータベースにアクセスするため、サーバー間で同じ情報が共有されます。

クライアントには、`sqlnet.ora` ファイルで指定された **優先 Oracle Names サーバー (Preferred Oracle Names Server)** のリストがあります。このリストはユーザーによって作成され、接続する Oracle Names の優先順位を表します。リストの先頭にある Oracle Names サーバーが、最初にクライアントから接続されます。

Preferred Oracle Names サーバーは、現在でも構成する場合があります。

**関連項目：** 構成に関する情報は、8-23 ページの「[優先 Oracle Names サーバーの構成](#)」を参照してください。

### Oracle Names バージョン 2

Oracle Names バージョン 2 では、管理者は Oracle Names サーバーの構成をバージョン 1 から引き継ぐか、または動的検出オプションを使用するかを選択できました。動的検出オプションは、単一リージョンおよび単一 DNS ドメインのネットワークの場合にのみ選択することをお勧めしていました。動的検出オプションでは、**予約済 Oracle Names サーバー**



(well-known Oracle Names サーバー) を使用します。これは、DNS にハードコードされた正確な名前か、または Oracle Names サーバーとそのクライアントの両方に存在する hosts ファイルにハードコードされた正確な名前です。

TCP 接続用の予約済ホスト名	Named Pipes 接続用の予約済コンピュータ名 <sup>1</sup>	SPX 接続用の予約済サービス名
oramesrvr0	ORAMESRVR0	oramesrvr
oramesrvr1	ORAMESRVR1	
oramesrvr2	ORAMESRVR2	
oramesrvr3	ORAMESRVR3	
oramesrvr4	ORAMESRVR4	

<sup>1</sup> Named Pipes 用の予約済 Names Server 名は、すべて大文字にする必要があります。

Oracle Names サーバーは、これらの予約済アドレスで使えるようになるので、優先 Oracle Names サーバーのリストによって Oracle Names サーバーの検索先をクライアントに指示する必要はありません。

動的検出オプションが選択された場合、各 Oracle Names サーバーは、管理リージョンにある他のすべての予約済 Oracle Names サーバーにデータを自動的にレプリケートしていました。リスナーは予約済 Oracle Names サーバーに登録するように構成されていました。

**関連項目：** 構成については、『Oracle Names 管理者ガイド』を参照してください。

## Oracle Names バージョン 8 (このリリース)

Oracle Names バージョン 8 では、管理者は Oracle Names サーバーの構成をバージョン 1 またはバージョン 2 から引き継ぐか、または新しい機能を使用するか選択できます。Oracle Names バージョン 8 には、バージョン 2 の動的検出オプション機能が組み込まれ、さらに単一リージョンおよびシングル・ドメインの制約がなくなっています。Oracle Names バージョン 8 の主要な機能は次のとおりです。

- サービスは、検索した Oracle Names サーバーに自身を登録し、名前とアドレスをリージョン内のすべての Oracle Names サーバーで利用可能にします。同様に、管理者がサービスを Oracle Names サーバーに手動で登録する場合も、サービス情報がすべての Oracle Names サーバーで利用可能になります。アドレス情報は、次のいずれかの方法で共有されます。
  - リージョン・データベース。Oracle データベースが登録リポジトリとして使用される場合、登録情報はその Oracle データベースに格納され、すべての Oracle Names サーバーがそのデータベースにアクセスできます。
  - サービス・レプリケーション。サービス・レプリケーションでは、サービス情報は Oracle Names サーバーのキャッシュに格納され、さらにそれ以外のすべての Oracle Names サーバーのキャッシュに即座にレプリケートされます。



- 接続対象の Oracle Names サーバーのリストは、4-20 ページの「[検出の理解](#)」で説明するように、手動による構成ではなく、検出プロセスによってクライアントに作成されます。
- クライアント側のプロセスが、Oracle Names サーバーに格納されたデータのクライアント・キャッシュ・バージョンを作成するために実行されます。ここには、データベース・アドレス、Oracle Names サーバーのアドレス、その他のサービス情報が含まれます。結果として、クライアントは Oracle Names サーバーに接続してアドレス情報を取得する必要がなく、アドレスの検索時間を短縮できます。
- Oracle Names バージョン 8 では、トポロジー情報を維持するためにデータベースを必要としません。ただし、管理者の設定で使用することもでき、こちらをお勧めします。

**関連項目：** 構成に関する情報は、6-43 ページの「[Oracle Names メソッドの構成](#)」を参照してください。

## 検出の理解

Oracle Names サーバーのリストを作成すると、ある Oracle Names サーバーに対して、クライアントや他の Oracle Names サーバーが接続できるようになります。このリストを作成するプロセスを検出と呼びます。

クライアントが NAMESCTL ユーティリティまたは Oracle Net8 Assistant を使用して Oracle Names サーバーを検出しようとする、最初に Oracle Names サーバーが 1 つ検出されます。クライアントは Oracle Names サーバーを 1 つ検出すると、リージョン内の他の Oracle Names サーバーに ping します。これにより、Oracle Names サーバーのリストがクライアントに作成され、プラットフォームが UNIX の場合は `.sdns.ora`、Windows の場合は `sdns.ora` に保存されます。このリストは、応答時間の順に格納されます。

では、次の順で最初の Oracle Names サーバーが検出されます。

1. `sqlnet.ora` ファイルの優先 [Oracle Names サーバー \(Preferred Oracle Names Server\)](#)
2. [予約済 Oracle Names サーバー \(well-known Oracle Names サーバー\)](#)
3. ポート 1575 に TCP/IP で構成されたローカル Oracle Names サーバー

クライアントが上の方法で最初の Oracle Names を検出できなかった場合は、次の処理が行われます。

- Oracle Net8 Assistant を使用している場合は、Oracle Names サーバーのアドレスを指定するよう求められます。
- NAMESCTL 制御ユーティリティが失敗します。指定したアドレスでコマンドを再実行する必要があります。

## 外部ネーミング・サービス

Net8 には、次の外部ネーミング・メソッドが用意されています。



- **Novell ディレクトリ・サービス (Novell Directory Services: NDS)** 外部ネーミング  
NDS 外部ネーミングでは、外部の NDS ネーミング表記法を使用して、Novell NDS 対応のネットワーク上で Oracle データベースに接続できます。
- **Network Information Service (NIS)** 外部ネーミング  
システム基盤の一部としてすでに NIS を使用している組織や会社には、NIS 外部ネーミングを使用して NIS 内に Oracle サービスの別名およびアドレスを格納するオプションがあります。
- **セル・ディレクトリ・サービス (Cell Directory Services: CDS)** 外部ネーミング  
CDS 外部ネーミングを使用すると、ユーザーは各種の Oracle ツールおよびアプリケーションから分散コンピューティング環境 (DCE) での Oracle8i データベースに透過的にアクセスできます。

**関連項目：** NDS および NIS の構成情報は、6-71 ページの「[外部ネーミング・メソッドの構成](#)」を参照してください。

## 認証方式 (authentication method)

Net8 には、次の認証方式 (authentication method) が用意されています。

- **Windows NT のシステム固有な認証 (Windows NT native authentication)**  
Windows NT サーバーおよび Oracle データベースに対してクライアントのシングル・ログインが可能になります。
- **NDS 認証**  
単一の Novell Directory Services (NDS) ディレクトリ・ツリーの下にある複数サーバーおよび複数データベース・ネットワークに対して、クライアントのシングル・ログインが可能になります。

### 関連項目：

- Windows NT のシステムに固有な認証についての構成情報は、『Oracle8i for Windows NT 管理者ガイド』を参照してください。
- NDS 認証の構成情報は、Oracle Novell ドキュメントを参照してください。

その他の認証方式は、4-22 ページの「[シングル・サインオン](#)」で説明するように、Oracle Advanced Security によって使用できます。



## Oracle Advanced Security

**Oracle Advanced Security** は、個別にライセンスの許諾が必要なオプションで、次のコンポーネントから構成されています。

- ネットワーク・セキュリティ
- シングル・サインオン
- Distributed Computing Environment インテグレーション

**関連項目：** これらの機能の詳細は、『Oracle8i Advanced Security 管理者ガイド』を参照してください。

### ネットワーク・セキュリティ

Oracle のネットワーク・データの暗号化とチェックサム・サービスによって、ネットワーク上での安全なデータ送信が保証されます。ネットワーク・セキュリティでは、RSA Data Security 社が提供する暗号化およびチェックサムのエンジンを使用します。

次のアルゴリズムをサポートします。

暗号化

- DES\_40（米国内版および輸出版）
- RC4\_40（米国内版および輸出版）
- RC4\_56（米国内版および輸出版）
- RC4\_128（米国内版）

チェックサム

- MD5

### シングル・サインオン

シングル・サインオンを使用すると、ユーザーは、1 つのパスワードのみで複数のアカウントおよびアプリケーションにアクセスできます。この機能によって、ユーザーは複数のパスワードを維持する必要がなくなり、システム管理者のユーザー・アカウントとパスワードの管理作業を簡素化します。

集中化された、安全な認証サービスによって、分散環境でのユーザー、クライアントおよびサーバーの識別情報に高度な信頼性が与えられます。また、ネットワーク認証サービスによって、シングル・サインオンという便宜がユーザーに与えられます。

次の認証方式をサポートします。

- CyberSAFE
- Identix (Biometric)



- Kerberos
- RADIUS
- SecurID
- SSL

## Distributed Computing Environment インテグレーション

Distributed Computing Environment (DCE) インテグレーションによって、ユーザーは透過的に各種の Oracle ツールとアプリケーションを使用できるようになり、アプリケーションは DCE 環境において Oracle8i データベースにアクセスできるようになります。Oracle DCE インテグレーション製品は、2 つの主要なコンポーネントから構成されます。

- DCE 通信 / セキュリティ method
- DCE CDS (Cell Directory Service) ネーミング method

## 管理ツール

この項では、Net8 で使用できる管理ツールについて説明しています。これらのツールは次のとおりです。

- [Net8 Configuration Assistant](#)
- [Oracle Net8 Assistant](#)
- 制御ユーティリティ

## Net8 Configuration Assistant

[Net8 Configuration Assistant](#) を使用すると、基本的なネットワーク・コンポーネントを構成できます。

Net8 Configuration Assistant は、Oracle インストレーション・ガイドで説明するように、インストール後、自動的に実行されます。クライアントとサーバーのどちらでも使用できます。

このツールは、スタンドアロン・モードで実行して、ネーミング・メソッド、リスナー、tnsnames.ora ファイル内のネット・サービス名およびディレクトリ・サーバー・アクセスを構成できます。

Net8 Configuration Assistant を起動するには、次の手順に従います。

- UNIX の場合、\$ORACLE\_HOME/bin から netca を実行します。
- Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Configuration Assistant」の順に選択します。



**関連項目：** Net8 Configuration Assistant オンライン・ヘルプ

## Oracle Net8 Assistant

**Oracle Net8 Assistant** は、構成機能にコンポーネント制御機能を組み合わせることによって、Net8 の構成および管理のための統合環境を提供するグラフィカル・ユーザー・インタフェース・ツールです。クライアントとサーバーのどちらでも使用できます。

Oracle Net8 Assistant を使用すると、次のネットワーク・コンポーネントを構成できます。

- **ネーミング** – 単純名、接続識別子を定義して接続記述子にマップし、ネットワーク位置およびサービスを識別します。Oracle Net8 Assistant では、ローカルの `tnsnames.ora` ファイル、集中化された LDAP 準拠ディレクトリ・サービスまたは Oracle Names サーバーでの接続記述子の構成がサポートされます。
- **ネーミング・メソッド** – 接続識別子を接続記述子に解決する様々な方法を構成します。
- **リスナー** – リスナーの作成および構成を行って、クライアント接続を受け取ります。

Oracle Names サーバーを構成した場合、その起動、停止、チューニングまたは統計情報収集は、Oracle Net8 Assistant を使用して行います。

Oracle Net8 Assistant を起動するには、次の手順に従います。

- UNIX の場合、`$ORACLE_HOME/bin` から `netasst` を実行します。
- Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」の順に選択します。

**関連項目：** Oracle Net8 Assistant オンライン・ヘルプを参照してください。

## 制御ユーティリティ

Net8 には、リスナー、Oracle Names サーバーおよび Oracle Connection Manager を制御する制御ユーティリティが用意されています。これらのユーティリティの内容は次のとおりです。

- **LSNRCTL 制御ユーティリティ**
- **CMCTL 制御ユーティリティ**
- **NAMESCTL 制御ユーティリティ**

### LSNRCTL 制御ユーティリティ

リスナー制御ユーティリティ（LSNRCTL）はリスナーを管理します。LSNRCTL の一般的な構文は、次のとおりです。

`LSNRCTL command [listener_name]`



*listener\_name* は、*listener.ora* ファイルで定義したリスナーの名前です。LISTENER という名前のデフォルトのリスナーを使用している場合は、リスナーを指定する必要はありません。

LSNRCTL には、次に示すタイプのコマンドが含まれています。

- 操作コマンド (START、STOP など)
- 実行中のリストのための修飾コマンド SET
- 情報コマンド (STATUS、SHOW コマンドなど)
- コマンド・ユーティリティ操作コマンド (EXIT、QUIT、HELP など)

**関連項目：** A-2 ページの「[リスナー制御ユーティリティ \(LSNRCTL\)](#)」

## CMCTL 制御ユーティリティ

Oracle Connection Manager 制御ユーティリティ (CMCTL) は、Oracle Connection Manager を起動し、制御するためにオペレーティング・システム・プロンプトから実行するツールです。CMCTL の一般的な構文は、次のとおりです。

*CMCTL command [process\_type]*

*process\_type* は、コマンドの実行対象となるプロセスの名前です。これらを次に示します。

- *cman* (CMGW ゲートウェイおよび CMADMIN 管理プロセスの両方)
- *adm* (CMADMIN プロセスのみ)
- *cm* (CMGW プロセスのみ)

CMCTL ユーティリティには、次のいくつかのタイプのコマンドがあります。

- 操作コマンド (START、STOP、SHUTDOWN など)
- 実行中の Oracle Connection Manager のための修飾子コマンド SET
- 情報コマンド (STATUS、STATS、SHOW コマンドなど)
- コマンド・ユーティリティ操作コマンド (EXIT、QUIT、HELP など)

**関連項目：** A-77 ページの「[Oracle Connection Manager 制御ユーティリティ \(CMCTL\)](#)」を参照してください。

## NAMESCTL 制御ユーティリティ

Oracle Names 制御ユーティリティ (NAMESCTL) は、Oracle Names サーバーを操作、制御するためにオペレーティング・システム・プロンプトから実行するツールです。NAMESCTL の一般的な構文は、次のとおりです。

*NAMESCTL command*



NAMESCTL には、次のいくつかのタイプのコマンドがあります。

- 操作コマンド (START、STOP、RESTART など)
- 実行中の Names サーバーのための修飾子コマンド SET
- 情報コマンド (STATUS、SHOW コマンド、PING など)
- コマンド・ユーティリティ操作コマンド (EXIT、QUIT、HELP など)

**関連項目：** A-25 ページの「[Oracle Names 制御ユーティリティ \(NAMESCTL\)](#)」を参照してください。



---

## ネットワークのプランニング

Net8 には、柔軟性のある使用しやすいネットワークの設計と管理を支援する各種オプションがあります。Net8 では拡張性と管理容易性が向上したので、様々な環境をサポートするためのネットワークを構築できます。これらの環境が単純なワークグループであるか、大規模でミッション・クリティカルな企業であるかどうかは関係ありません。

この章では、Net8 を使用してネットワークを計画するときに考慮する必要がある点について説明します。また、ネットワーク製品の関係と、今後ネットワークを拡張してさらに効率的に管理するためのオプションについて説明します。内容は次のとおりです。

- [計画の概要](#)
- [ネットワーク・レイアウトの決定](#)
- [接続記述子への名前の解決](#)
- [大規模ネットワーク・パフォーマンスの向上](#)
- [計画のサマリー](#)



## 計画の概要

ネットワークを構成する前に、ネットワークの検討と計画のための時間を取ってください。Oracle ネットワークを計画するときは、将来のニーズと現在の要件を念頭に置いてください。ネットワークに関して行う必要がある重要な決定は、次のとおりです。

- ネットワーク・レイアウトの決定
- 接続記述子への名前の解決
- 大規模ネットワーク・パフォーマンスの向上

## ネットワーク・レイアウトの決定

次のチェックリストは、ネットワークの主なコンポーネントの概要を示すためのものです。

1. ネットワークで行うことは何かを定義します。
2. クライアント・アプリケーションに必要な機能を調べ、これらの要件を満たすためのリソースを評価します。
3. どのマシンまたはノードが、クライアントまたはサーバーのアプリケーションに最適であるかを判断します。
4. 既存または将来のネットワークング要件に最適なネットワーク・プロトコルを選択します。多数のプラットフォーム上でほとんどの主要プロトコルが利用可能です。ネットワーク内のすべてのコンポーネントでうまく機能する1つのトランスポート・レベルのプロトコルを選択できます。使用しているネットワークが、複数のプロトコルを通じて動作するクライアントまたはサーバーに関与している場合があります。
5. ネットワークで複数のプロトコルを使用することに決定した場合は、どのノードが Oracle Connection Manager をインストールするのに最適であるかを判断します。どのノードを選択するかについては、選択したネットワーク・プロトコルと、予想される通信量を処理するマシンの能力によって決定します。

ネットワーク・コンポーネント間の物理的関係とともに論理的关系を示す図を描くと役立つことがあります。

## 接続記述子への名前の解決

ネットワーク・レイアウトを定義した後は、ネットワークの実装を効率的に構成、管理する方法を決定する必要があります。ネーミング・メソッドを選択することが、最も重要で、最初に行う必要のある決定です。

次の表は、各ネーミング・メソッドの相対的な利点および不利な点をまとめ、ネットワーク内でそれらのネーミング・メソッドを使用するためのヒントを示したものです。



ネーミング・メソッド	メリット/デメリット	推奨環境
ローカル・ネーミング	<p><b>メリット:</b></p> <ul style="list-style-type: none"> <li>■ ネット・サービス名アドレスを解決する比較的小さい方法です。</li> <li>■ 異なるプロトコルを実行しているネットワーク間でネット・サービス名を解決します。</li> </ul> <p><b>デメリット:</b> すべてのネット・サービス名とアドレス変更をローカル側で構成する必要があります。</p>	ほとんど変更がなく、サービス数の少ない単純な分散ネットワーク
ディレクトリ・ネーミング	<p><b>メリット:</b></p> <ul style="list-style-type: none"> <li>■ ネットワーク名とアドレスを 1 箇所に集中させるので、名前の変更および更新を管理しやすくなります。このため、管理者は数百、数千のクライアントに対して変更を行う必要がありません。</li> <li>■ ディレクトリは、この他のサービスの名前も格納します。</li> <li>■ ツールを使用して簡単に構成ができます。</li> </ul> <p><b>デメリット:</b> ディレクトリ・サーバーにアクセスする必要があります。</p>	頻繁に変更する大規模で複雑なネットワーク (20 以上のデータベースを持つ)
Oracle Names	<p><b>メリット:</b> ネットワーク名とアドレスを 1 箇所に集中させるので、名前の変更および更新を管理しやすくなります。たとえば、既存のサーバーに対して変更を付加する場合、または新しいサーバーをネットワークに追加する場合、変更は 1 つの Oracle Names サーバー上で 1 回のみ行われます。このため、管理者は数百、数千のクライアントに対して変更を行う必要がありません。</p> <p><b>デメリット:</b></p> <ul style="list-style-type: none"> <li>■ Oracle Names は、Oracle サービス用のネットワーク名とネットワーク・アドレスのみを格納します。</li> <li>■ Oracle Names サーバーの設定と管理が必要。</li> </ul>	頻繁に変更する大規模で複雑なネットワーク (20 以上のデータベースを持つ)



ネーミング・メソッド	メリット/デメリット	推奨環境
ホスト・ネーミング	<p><b>メリット:</b></p> <ul style="list-style-type: none"><li>■ 最小のユーザー構成で済みます。ユーザーは、接続を確立するためにホスト名のみを指定します。</li><li>■ ローカル名構成ファイル (tnsnames.ora) の作成とメンテナンスの必要がありません。</li></ul> <p><b>デメリット:</b> 推奨環境の欄に示すように、使用できる環境に制限があります。</p>	<p>次に示す条件に合致する単純な TCP/IP ネットワーク</p> <ul style="list-style-type: none"><li>■ クライアントおよびサーバーは、TCP/IP を使用して接続していること。</li><li>■ ホスト名は、ドメイン・ネーム・サービス (DNS)、NIS、中央でメンテナンスされる TCP/IP hosts ファイルなどの IP アドレス解決メカニズムによって解決されること。</li><li>■ Oracle Connection Manager の機能は不要。</li></ul>
外部ネーミング	<p><b>メリット:</b> 管理者は、使用方法を熟知しているツールおよびユーティリティを使用して、Oracle ネット・サービス名をサイト固有のネーム・サービスにロードできます。</p> <p><b>デメリット:</b> Net8 製品では管理できない第 3 のネーミング・サービスが必要です。</p>	<p>既存のネーム・サービスを持つネットワーク</p>

## 大規模ネットワーク・パフォーマンスの向上

次のいずれかを実施すると、大規模なネットワークのパフォーマンスを向上できます。

- 共有サーバーまたは事前生成済専用サーバーの使用
- リスナー・キューサイズの増加
- 接続プーリングの使用
- 接続集中化機能の使用
- クライアント・ロード・バランスの使用
- データ転送の最適化を目的としたセッション・データ・ユニット (SDU) サイズの調整
- TCP/IP の持続的バッファ・フラッシング

### 共有サーバーまたは事前生成済専用サーバーの使用

ネットワークに大量の接続通信量が発生すると予想される場合は、リスナーを使用して接続要求を **マルチスレッド・サーバー (multi-threaded server: MTS)** 共有サーバーまたは事前生成済専用サーバーにリダイレクトするように管理できます。

次の表は、各プロセスの相対的な利点をまとめ、ネットワーク内でそれらのプロセスを使用するためのヒントを示したものです。



タイプ	利点	推奨環境
MTS 共有サーバー	<ul style="list-style-type: none"> <li>■ 専用サーバーより効率よくネットワーク・リソースを利用するので、セッションのスループットとパフォーマンスが向上します。</li> <li>■ データベースに対するセッション数が増加しても、サーバー側で必要なメモリー・リソースと処理リソースを最小限に抑えることができます。</li> </ul>	MTS がサポートされているネットワーク、または新しいサーバー・プロセスの作成に時間のかかるリソースに厳しい環境
事前起動済または事前生成済の専用サーバー・プロセス	<ul style="list-style-type: none"> <li>■ 新しい接続要求ごとに専用サーバーを作成する必要がないので、接続時間を短縮できます。</li> <li>■ サーバした・プロセスを停止して再作成しなくても、他の接続で使用したサーバーを再利用できるので、割当て済のメモリーやシステム・リソースを有効に利用できます。</li> </ul>	MTS がサポートされていないネットワーク、または新しいサーバー・プロセスの作成に時間のかかるリソースに厳しい環境

**関連項目：**

- MTS の概念については、2-6 ページの「[マルチスレッド・サーバー・モデル](#)」を参照してください。
- [第9章「マルチスレッド・サーバーの構成」](#)を参照してください。
- 概念的な情報は、2-14 ページの「[事前生成済専用サーバー](#)」を参照してください。
- 構成に関する情報は、7-13 ページの「[事前生成済専用サーバーの構成](#)」を参照してください。

## リスナー・キューサイズの増加

TCP/IP を介してリスニング・プロセス（リスナー、Oracle Connection Manager、Oracle Names など）に対する多数の接続要求を受信することが予想される場合は、Net8 を使用してシステム・デフォルトより高い値になるようにリスニング・キューを設定できます。

**関連項目：** 既存のキューサイズの構成については、7-10 ページの「[多数の接続要求の処理](#)」を参照してください。

## 接続プーリングの使用

接続プーリングは、マルチスレッド・サーバーに対する物理ネットワーク接続数を最大にすることのできるリソース利用機能です。これは、複数のクライアント・プロセス間でディスパッチャの接続の集合を共有またはプールすることにより行います。

タイムアウト・メカニズムを使用して、指定された期間アイドル状態であったトランスポート接続を一時的に解放することによって、接続プーリングは、以前のアイドル接続との論理セッションを維持しながら、これらの物理接続を着信クライアントに使用できるようにします。



す。アイドル・クライアントにさらに多くの作業がある場合、物理接続はディスパッチャで再確立されます。

この機能は、マルチスレッド・サーバーが構成されている場合にのみ動作します。

関連項目：

- 概念的な情報は、2-10 ページの「[接続プーリング](#)」を参照してください。
- 構成の詳細は、9-9 ページの「[クライアントでの MTS の使用](#)」を参照してください。

接続集中化機能の使用

接続集中化は、Oracle Connection Manager を使用することにより利用できる機能です。この機能によって、Oracle Connection Manager の利点を活用し、複数のクライアント・セッションを、1つのトランスポートを介してマルチスレッド・サーバーに多重送信または集中送信できます。接続プーリングと同様に、接続集中化によってネットワーク・リソースが最適化され、一定数の物理サーバー・ポート間で可能なクライアント / サーバー・セッション数が増加します。接続プーリングとは異なり、接続集中化ではトランスポート接続が維持されます。

関連項目：

- 概念的な情報は、4-4 ページの「[Oracle Connection Manager](#)」を参照してください。
- 構成に関する情報は、8-25 ページの「[接続集中化機能の使用](#)」を参照してください。

接続プーリングと接続集中化の使用

次の表は、接続プーリングと接続集中化を使用するときの相対的な利点をまとめ、ネットワーク内でそれらを使用するためのヒントを示したものです。

機能	利点	推奨環境
接続プーリング	<ul style="list-style-type: none"><li>■ 1つのプロセスに対して使用されるネットワーク・リソース数を制限します。</li><li>■ 一定数に制限された物理接続数に対するクライアント / サーバー・セッション数を最大化します。</li><li>■ リソースの利用を最適化します。</li></ul>	メッセージ送信、OLAP など、対話形式の評価 / 検索の時間頻度が高いアプリケーションを多数のクライアントで実行するようなネットワーク



機能	利点	推奨環境
接続集中化	<ul style="list-style-type: none"><li>■ 多数のクライアントがサポートされます。</li><li>■ 実ユーザーの識別および監視が可能になります。</li><li>■ 中間層のアプリケーションによって追加サービスをサポートできるようになります。</li><li>■ 複数のアプリケーションを持つクライアントに対して、必要なトランスポートが1つのみで済みます。</li><li>■ データベース・リンクに対して、必要なネットワーク接続が1つのみで済みます。</li></ul>	持続的接続が必要なネットワーク

## クライアント・ロード・バランスの使用

複数のリスナーが1つのサービスをサポートしている場合、クライアントを構成して様々なリスナーへの要求をランダム化できます。ランダム化の機能は、1つのリスナーにロードが集中しないようにロードを分散することです。ロードのバランスをとることにより、接続パフォーマンスが向上します。

**関連項目：** 構成の詳細は、8-3 ページの「[アドレス・リスト・パラメータの構成](#)」を参照してください。

## データ転送の最適化を目的としたセッション・データ・ユニット（SDU）サイズの調整

ネットワーク間の往復回数を減らすようにアプリケーションをチューニングすることは、ネットワーク・パフォーマンスを向上させる最高の方法です。これを行った後、セッション・データ・ユニット（SDU）のサイズを調整してデータ転送を最適化することもできます。

SDU は、Net8 がネットワーク間でデータを転送する前にデータを配置するバッファです。Net8 がバッファ内のデータを送信するのは、データ送信が要求されたとき、またはバッファがデータでいっぱいになったときです。

次の表は、SDU のサイズ変更が適切な時期と適切でない時期を示したものです。



SDU サイズの変更が適切な時期	SDU サイズの変更が適切でない時期
<ul style="list-style-type: none"><li>■ サーバーから戻されるデータが個別のパケットに別れる場合</li><li>■ 遅延の起こる広域ネットワーク（WAN）上にいる場合</li><li>■ パケット・サイズが一定している場合</li><li>■ 戻されるデータ量が多い場合</li></ul>	<ul style="list-style-type: none"><li>■ 遅延に対処できるようにアプリケーションをチューニングできる場合</li><li>■ データ伝送の影響がほとんどない高速ネットワークを使用している場合</li><li>■ 要求によってサーバーから戻されるデータ量が少ない場合</li></ul>

**注意：** SDU サイズは、通常の伝送フレーム・サイズの倍数に設定する必要があります。イーサネットの通常のフレーム・サイズは 1500 のため、イーサネット・プロトコルでの最も効率的な SDU サイズは 1500 の倍数です。

接続プーリングまたは接続集中化を使用しているときは、これらの機能に対して、1 つのトランスポートについてさらに 16 バイトが必要になることに注意してください。プロトコル・フレーム・サイズの詳細は、使用しているオペレーティング・システムのドキュメントを参照してください。

**関連項目：** 構成の詳細は、8-8 ページの「[拡張接続データ・パラメータの構成](#)」を参照してください。

TCP/IP の持続的バッファ・フラッシング

TCP/IP を使用するいくつかのアプリケーションのある状況では、Net8 のパケットがネットワークにすぐに送出されません。ほとんどの場合、この動作は大量のデータが一方から他方に送られるときに発生します。TCP/IP の実装自体がフラッシュの欠如の原因で、許容範囲を超える遅延を引き起こすおそれがあります。この問題を修正するためには、バッファ・フラッシング・プロセスで遅延なしを指定できます。

**関連項目：** このパラメータについては、C-74 ページの「[TCP.NODELAY](#)」を参照してください。

計画のサマリー

次の表は、ネットワークを計画したときに選択した可能性のある多数のオプションをまとめたものです。

科目	オプション
ネットワーク・レイアウト	<ul style="list-style-type: none"><li>■ 単一または複数のプロトコル</li></ul>



科目	オプション
ネット・サービス名の変換	<ul style="list-style-type: none"><li>■ ローカル・ネーミング</li><li>■ ディレクトリ・ネーミング</li><li>■ Oracle Names</li><li>■ ホスト・ネーミング</li><li>■ 外部ネーミング</li></ul>
接続要求の管理	<ul style="list-style-type: none"><li>■ 共有サーバー</li><li>■ 事前生成済専用サーバー</li></ul>
ネットワーク・パフォーマンス	<ul style="list-style-type: none"><li>■ リスナー・キューサイズの増加</li><li>■ クライアント・ロード・バランスの使用</li><li>■ 接続プーリングの使用</li><li>■ 接続集中化の使用</li><li>■ SDU サイズの調整</li><li>■ 持続的バッファ・フラッシング</li></ul>







# 第 II 部

---

## 構成

第 II 部では、Net8 のセットアップおよび構成方法について記述します。第 II 部には、次の章があります。

- [第 6 章「ネーミング・メソッドの構成」](#)
- [第 7 章「リスナーの構成」](#)
- [第 8 章「Net8 の拡張機能」](#)
- [第 9 章「マルチスレッド・サーバーの構成」](#)
- [第 10 章「プログラマのための Net8 拡張機能」](#)







---

## ネーミング・メソッドの構成

この章では、ネーミング・メソッドの構成方法について説明します。内容は次のとおりです。

- [構成の概要](#)
- [接続記述子の理解](#)
- [構成モデル](#)
- [ローカル・ネーミング・メソッドの構成](#)
- [ディレクトリ・ネーミング・メソッドの構成](#)
- [Oracle Names メソッドの構成](#)
- [ホスト・ネーミング・メソッドの構成](#)
- [外部ネーミング・メソッドの構成](#)



## 構成の概要

サービスへ接続する際、クライアントでは、サービスへ接続する接続文字列において**接続記述子 (connect descriptor)**ではなく、**接続識別子 (connect identifier)**と呼ばれる単純な名前を使用します。接続記述子には次のものが含まれます。

- プロトコル・アドレスによるリスナーの位置情報など、サービスへのネットワーク・ルート。
- Oracle リリース 8.1 データベースのサービス名または Oracle 8.0 またはバージョン 7 データベースの **Oracle システム識別子 (Oracle System Identifier: SID)**。

接続識別子には、**ネット・サービス名 (net service name)** (サービスの単純名) または実際のサービス名を使用できます。接続識別子は、クライアントに対して指定された**ネーミング・メソッド (naming method)**により、接続記述子に解決されます。

いったん接続識別子が解決されると、クライアントにより、接続依頼がリスナーに転送されます。リスナーでは、クライアントの情報が、データベース・サービスから受信した情報および構成ファイル listener.ora に格納された情報と比較されます。情報が一致した場合は接続が許可されます。

ネーミング・メソッドの構成は、サービスに対する接続記述子の作成およびネーミング・メソッドへアクセスするクライアントの構成からなります。

## 接続記述子の理解

接続記述子は、1 つまたは複数のリスナーのプロトコル・アドレスおよび宛先サービスに対する接続データ情報により構成されます。次の例では、代表的な接続記述子を示します。

```
sales=
  (description=
    (address= (protocol=tcp) (host=sales-server) (port=1521))
    (connect_data=
      (service_name=sales.us.acme.com)))
```

ADDRESS 部分には、リスナー・プロトコル・アドレスが含まれ、CONNECT\_DATA 部分には、宛先サービス情報が含まれます。この例では、宛先サービスは、sales.us.acme.com という名前のデータベース・サービスです。

Oracle リリース 8.1 データベース・サービスの接続記述子を作成する際、SERVICE\_NAME パラメータを使用して、そのサービス名を必ず指定する必要があります。オプションで、INSTANCE\_NAME パラメータでインスタンスを次のように特定できます。

```
sales=
  (description=
    (address= (protocol=tcp) (host=sales-server) (port=1521))
    (connect_data=
      (service_name=sales.us.acme.com)
      (instance_name=sales)))
```



これらのパラメータの値は、**初期化パラメータ・ファイル (initialization parameter file)** の SERVICE\_NAMES (S 付き) および INSTANCE\_NAME から取得されます。一般的に初期化パラメータ・ファイルの SERVICE\_NAMES パラメータは、インストールまたはデータベース作成時に入力されたデータベース名とドメイン名から構成される**グローバル・データベース名 (global database name)** です。たとえば、sales.us.acme.com には、sales のデータベース名と us.acme.com のドメインがあります。初期化パラメータ・ファイルの INSTANCE\_NAME パラメータは、インストールまたはデータベース作成中に入力された SID に、デフォルト設定されます。

**関連項目：** 2-27 ページの「SID ではなくサービス名によるデータベースの識別」

Oracle リリース 8.0 またはバージョン 7 データベースの接続記述子を作成する際、SID パラメータを使用して、そのサービスを必ず識別する必要があります。次の例では、sales という SID の Oracle リリース 8.0 データベースへの接続記述子を示します。

```
sales=
(description=
(address= (protocol=tcp) (host=sales-server) (port=1521)
(connect_data=
(sid=sales)))
```

構成モデル

Net8 の構成は、2 つのモデルのいずれかを基盤としています。

ネットワークの構成モデル	適したネットワーク	ネーミング・メソッド
ローカル管理	ネットワーク・アドレスは、各マシン上のファイル tnsnames.ora にマップされます。	ローカル・ネーミング ホスト・ネーミング
集中管理	Oracle Names サーバーまたはディレクトリによってネットワーク・アドレスの解決が実行されるネットワーク	ディレクトリ・ネーミング Oracle Names 外部ネーミング



## ネーミング・メソッド

Net8 では、次のネーミング・メソッドがサポートされます。

ネーミング・メソッド	説明
ローカル・ネーミング	ネット・サービス名をクライアント上に格納される tnsnames.ora ファイルに格納します。  ローカル・ネーミングは、あまり変更されない少数のサービスのある単純な分散ネットワークに最も適しています。  <b>関連項目：</b> 6-4 ページの「 <a href="#">ローカル・ネーミング・メソッドの構成</a> 」
ディレクトリ・ネーミング	集中化された LDAP 準拠のディレクトリ・サービス内にサービスおよびネット・サービス名が格納されます。  <b>関連項目：</b> 6-13 ページの「 <a href="#">ディレクトリ・ネーミング・メソッドの構成</a> 」
Oracle Names	サービスおよびネット・サービス名が、Oracle Names サーバーに格納されます。  <b>関連項目：</b> 6-43 ページの「 <a href="#">Oracle Names メソッドの構成</a> 」
ホスト・ネーミング	ユーザーがホスト名の別名を使用して Oracle Server へ接続できます。別名は、ドメイン名システム（DNS）、NIS または集中管理された /etc/hosts ファイルのセットなどの、既存の名前解決サービス内によって接続記述子にマップされます。  この機能を利用するために、クライアントを構成する必要はありません。この方法は、単純な TCP/IP 環境にお勧めします。  <b>関連項目：</b> 6-68 ページの「 <a href="#">ホスト・ネーミング・メソッドの構成</a> 」
外部ネーミング	サービス情報がサードパーティ・ネーミング・サービスに格納されます。  <b>関連項目：</b> 6-71 ページの「 <a href="#">外部ネーミング・メソッドの構成</a> 」

**関連項目：** 2-30 ページの「[ネーミング・メソッドによるサービスの解決](#)」

## ローカル・ネーミング・メソッドの構成

ローカル・ネーミング・メソッドを使用すると、ネット・サービス名が tnsnames.ora ファイルに追加されます。各ネット・サービス名は、接続記述子にマップされます。接続記述子にマップされたネット・サービス名に対する tnsnames.ora ファイルの基本的な構文例を、次に示します。

```
sales=
(description=
  (address=(protocol=tcp) (host=sales-server) (port=1521))
  (connect_data=
    (service_name=sales.us.acme.com)))
```



この例では、sales は、DESCRIPTION に含まれる接続記述子にマップされるネット・サービス名です。DESCRIPTION には、リスナー・アドレスおよび宛先データベース・サービスが含まれます。

ローカル・ネーミング構成は、次の項で説明するように、インストール中およびインストール後に構成できます。

- インストール中の `tnsnames.ora` ファイル構成
- インストール後の `tnsnames.ora` ファイル構成

## インストール中の `tnsnames.ora` ファイル構成

**Net8 Configuration Assistant** を使用すると、クライアントに対するネット・サービス名の構成が可能となります。これは、ソフトウェアのインストール後、Oracle ユニバーサル・インストーラによって、起動されます。構成は、インストール・モードによって異なります。

- **Typical** または **Minimal** インストール
- **Custom** インストール

### Typical または Minimal インストール

Net8 Configuration Assistant では、Oracle データベース・サービスへ接続するための、`tnsnames.ora` ファイル中のネット・サービス名を構成するよう求められます。

### Custom インストール

Net8 Configuration Assistant では、使用するネーミング・メソッドを選択するよう求められます。ローカル・ネーミングを選択した場合、Net8 Configuration Assistant では、Oracle データベース・サービスへ接続するための、`tnsnames.ora` ファイル中のネット・サービス名を構成するよう求められます。

## インストール後の `tnsnames.ora` ファイル構成

ネット・サービス名は、いつでも `tnsnames.ora` ファイルに追加できます。ローカル・ネーミング・メソッドを構成するには、次の作業を実行します。

作業 1: ネット・サービス名の構成

作業 2: 第 1 番目のネーミング・メソッドとしての `TNSNAMES` の構成

作業 3: 構成情報の配布

作業 4: リスナーの構成

作業 5: データベースへの接続



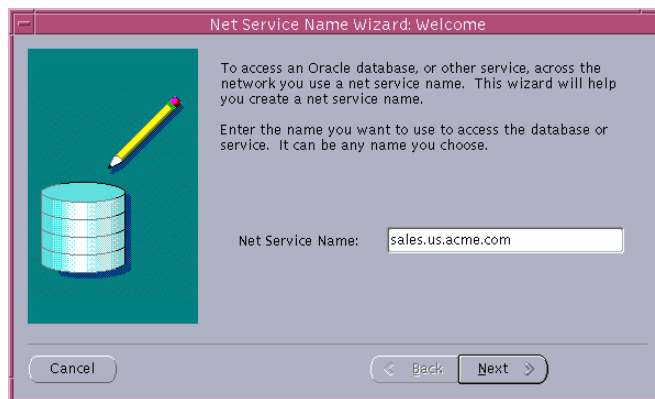
## 作業 1: ネット・サービス名の構成

ローカル・ネーミング・メソッドを使用して構成するには、次のツールのいずれか 1 つを使用します。

- [Net8 Assistant](#)
- [Net8 configuration Assistant](#)

**Net8 Assistant** [Oracle Net8 Assistant](#) を使用して `tnsnames.ora` ファイルのネット・サービス名を構成するには、次のようにします。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、`$ORACLE_HOME/bin` から `netasst` を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Local」 -> 「Service Naming」を展開します。
3. ツールバーで「+」をクリックするか、「編集」メニューから「作成」を選択します。  
Net Service Name Wizard が起動します。
4. 「Net Service Name」フィールドに名前を入力し、「Next」をクリックします。

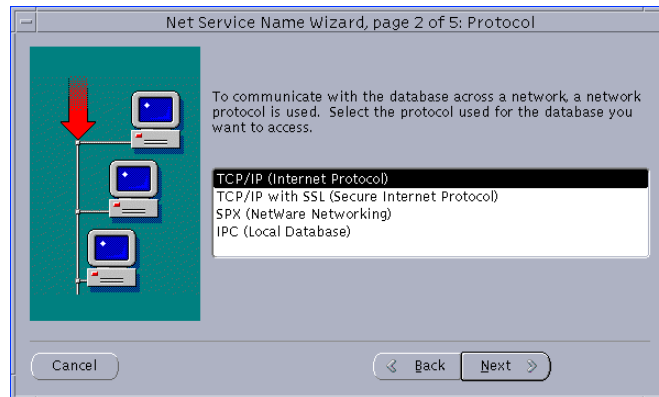


ネット・サービス名は、クライアントのドメインで修飾できます。`sqlnet.ora` ファイルに `NAMES.DEFAULT_DOMAIN` パラメータを指定すると、ネット・サービス名は自動的にドメインによって修飾されます。

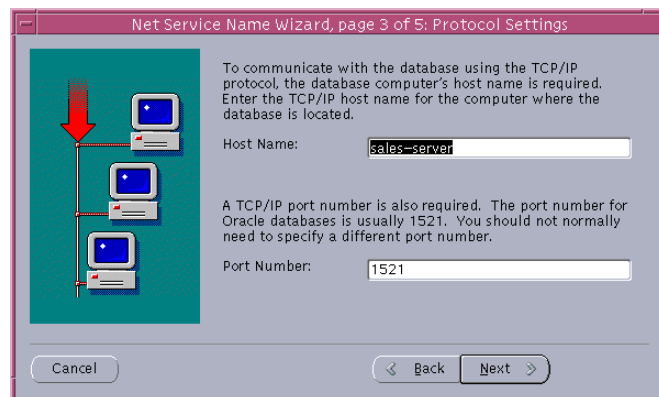
**関連項目 :** 8-15 ページの「[クライアントのデフォルト・ドメインの構成](#)」を参照してください。



- リスナーがリスニングするように構成されているプロトコルを選択します。このプロトコルも、クライアント上にインストールする必要があります。「Next」をクリックします。



- 提供されたフィールドで選択したプロトコルに対して適切なプロトコル・パラメータを入力し、「Next」をクリックします。



**関連項目：**プロトコル・パラメータの設定については、[付録 B](#) を参照してください。

- リリースを選択後、宛先サービスを入力し、「Next」をクリックします。





宛先サービスが Oracle リリース 8.1 である場合、「Oracle8i」をクリックし、「Service Name」フィールドにサービス名を入力します。宛先サービスが Oracle リリース 8.0 またはバージョン 7 データベースの場合、「Oracle8 or Previous」をクリックして、「Database SID」フィールドのインスタンスに対して Oracle システム識別子を入力します。

**関連項目：**使用するサービス名文字列の詳細は、6-2 ページの「[接続記述子の理解](#)」を参照してください。

オプションで、このネット・サービス名の Oracle8i 接続タイプ・リストからデータベース接続タイプを選択できます。オラクル社では、Database Default のデフォルト設定の使用をお勧めします。**マルチスレッド・サーバー (multi-threaded server: MTS)** が、初期化パラメータ・ファイルで構成される場合、専用サーバーを選択して、リスナーに MTS 構成を使用せずに専用サーバーを起動させることができます。MTS が初期化パラメータ・ファイルで構成され、接続時に常に MTS が使用されるようにする場合は、共有サーバーを選択します。

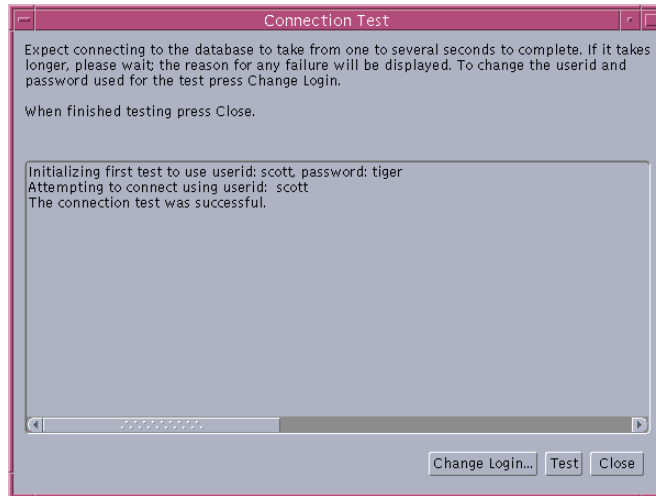
**関連項目：**MTS 構成の詳細は、[第 9 章](#)を参照してください。

8. ネット・サービス名の動作を検証するには「Test」をクリックし、「Next」をクリックします。

このテストでは、データベースとリスナーが実行されている必要があります。実行されていない場合は、11-2 ページの「[Net8 制御ユーティリティの使用](#)」の手順に従ってコンポーネントを起動するか、または「Next」をクリックしてテストを省略し、手順 10 に進んでください。

テストが成功すると、次のメッセージが表示されます。





テストが正常終了しなかった場合は、次の処置を行ってください。

- データベースとリスナーが実行されていることを確認し、「Test」をクリックします。
- 「Change Login」をクリックして接続に使用するユーザー名とパスワードを変更し、「Test」をクリックします。

9. 「Close」をクリックして「Connect Test」ダイアログ・ボックスを閉じます。

10. 「Finish」をクリックして構成を保存し、Net Service Name Wizard を終了します。

#### 関連項目：

- 複数のプロトコル・アドレスの構成方法については、8-2 ページの「[リスナー・プロトコル・アドレスのリスト作成](#)」を参照してください。
- 追加の接続データ・オプションの構成方法については、8-8 ページの「[拡張接続データ・パラメータの構成](#)」を参照してください。

**Net8 configuration Assistant** Net8 Configuration Assistant を使用して `tnsnames.ora` ファイルのネット・サービス名を構成するには、次のようにします。

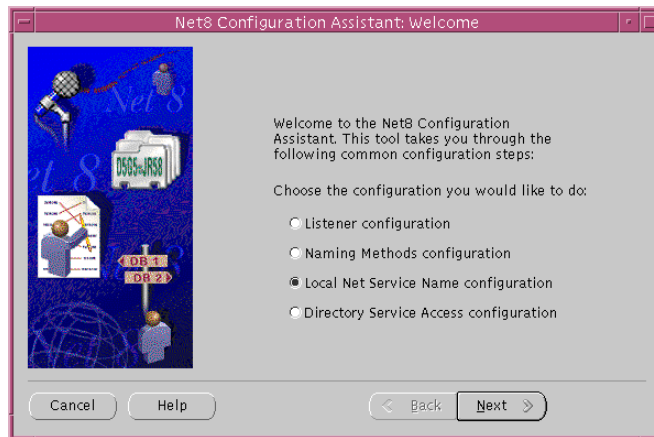
1. Net8 Configuration Assistant を起動します。

- UNIX の場合、`$ORACLE_HOME/bin` から `netca` を実行します。

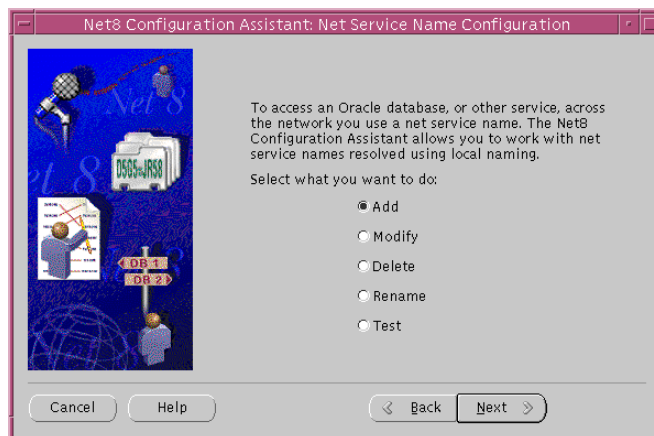


- Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Configuration Assistant」を選択します。

次のような「Welcome」ページが表示されます。



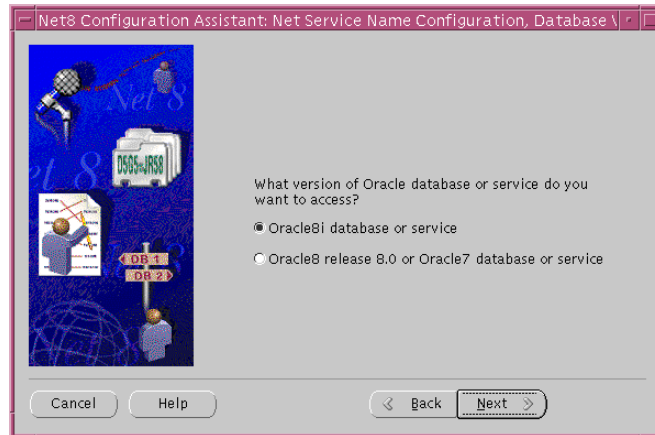
2. 「Local Net Service Name Configuration」を選択し、「Next」をクリックします。  
「Net Service Name Configuration」ページが表示されます。



3. 「Add」をクリックした後、「Next」をクリックします。



「Net Service Name Configuration, Database Version」 ページが表示されます。



4. リリースを選択して、「Next」をクリックします。

宛先サービスが Oracle8 リリース 8.1 データベースにある場合は、「Oracle8i database or service」を選択します。宛先サービスが Oracle8 リリース 8.0 またはバージョン 7 データベースにある場合は、「Oracle8 release 8.0 or Oracle7 database or service」を選択します。

5. ウィザードのプロンプトとオンライン・ヘルプに従って、ネット・サービス名を作成します。

## 作業 2: 第 1 番目のネーミング・メソッドとしての TNSNAMES の構成

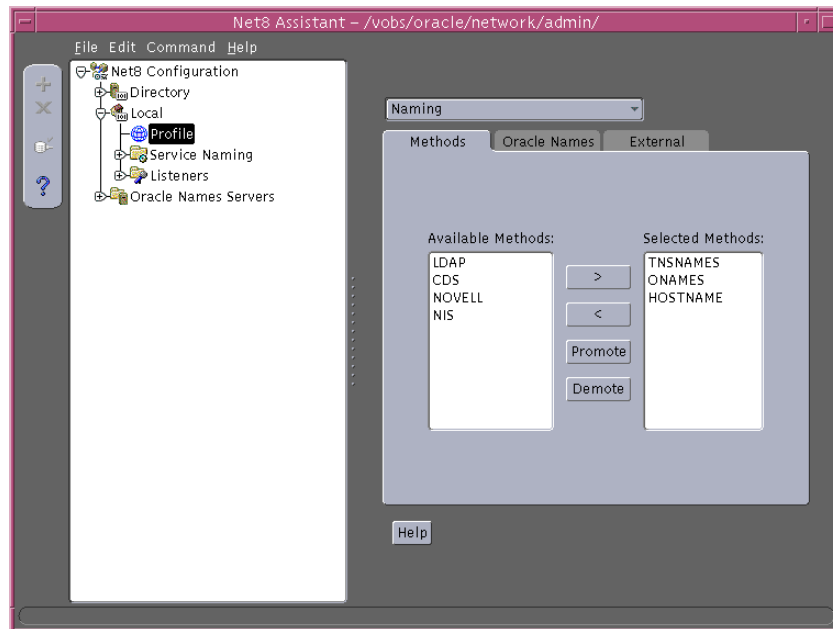
sqlnet.ora ファイルの NAMES.DIRECTORY\_PATH パラメータに指定する第 1 番目のネーミング・メソッドとしてローカル・ネーミングを構成します。このパラメータによって、Net8 が接続記述子の名前解決に使用するネーミング・メソッドの順序が指定されます。

ローカル・ネーミングを第 1 番目のネーミング・メソッドに指定するには、次の手順に従います。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Local」 -> 「Profile」を展開します。
3. 右ペインのリストから、「Naming」を選択します。



4. 「Methods」タブをクリックします。
5. 「Available Methods」リストから「TNSNAMES」を選択し、「>」ボタンをクリックします。
6. 「Selected Methods」リストで「TNSNAMES」を選択し、「Promote」ボタンを使用して選択した項目をリストの最上位に移動します。



7. 「File」 -> 「Save Network Configuration」を選択します。  
sqlnet.ora ファイルで、TNSNAMES がリストの最初にくるように  
NAMES.DIRECTORY\_PATH パラメータが更新されます。  

```
names.directory_path=(tnsnames, onames, hostname)
```

### 作業 3: 構成情報の配布

クライアントを 1 台構成すると、単に tnsnames.ora と sqlnet.ora の構成ファイルをクライアントの同じ場所にコピーする方法が最も確実です。これによって、ファイルの一貫性が保たれます。このようにしないと、Net8 Assistant か Net8 Configuration Assistant をすべてのクライアントで使用することが必要になり、エラーの原因になります。



## 作業 4: リスナーの構成

サーバー上のリスナーは、ネット・サービス名用に構成したものと同一プロトコル・アドレスを使用してリスニングするように構成する必要があります。デフォルトでリスナーは、ポート 1521 の TCP/IP プロトコルを使用するようにすでに構成されています。

**関連項目：** リスナー構成の詳細は、[第 7 章](#)を参照してください。

## 作業 5: データベースへの接続

クライアントは、次の構文を使用してデータベースへ接続できます。

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

# ディレクトリ・ネーミング・メソッドの構成

ディレクトリ・ネーミング・メソッドでは、接続識別子は、Oracle インターネット・ディレクトリ、Microsoft アクティブ・ディレクトリまたは Novell ディレクトリ・サービスを含む LDAP 準拠ディレクトリ・サーバーに含まれる接続記述子にマップされます。ディレクトリでは、データベース・サービスおよびネット・サービス名の集中管理が提供され、サービスの追加または再配置が容易になります。

データベース・サービス・エントリは、[Oracle Database Configuration Assistant](#) により、インストール中に作成され、ネット・サービス名エントリは、[Oracle Net8 Assistant](#) により作成されます。Net8 Assistant は、データベース・サービス・エントリおよびネット・サービス名エントリの Net8 属性変更の際にも使用されます。

ディレクトリにアクセスするよう構成されたクライアントでは、データベースへの接続の際に、これらのエントリを使用できます。

この項の内容は、次のとおりです。

- [ディレクトリ・ネーミング構成ステップ](#)
- [データベース・サービス・エントリに対する接続情報の変更](#)
- [OracleNetAdmins グループへのユーザーの追加と削除](#)
- [tnsnames.ora ファイルからのネット・サービス名のエクスポート](#)
- [Oracle Names サーバーからのネットワーク・オブジェクトのエクスポート](#)

## ディレクトリ・ネーミング構成ステップ

ディレクトリ・ネーミング・メソッドを構成するには、次の作業を実行します。

[作業 1: サーバーおよびクライアントでのディレクトリ・アクセスの構成](#)

[作業 2: ネット・サービス名の作成 \(オプション\)](#)



作業 3: 第 1 番目のネーミング・メソッドとしての LDAP の構成

作業 4: リスナーの構成

### 作業 1: サーバーおよびクライアントでのディレクトリ・アクセスの構成

データベース・サービスまたはネット・サービス名をディレクトリに追加する前に、ディレクトリ・アクセス構成が完了している必要があります。ディレクトリ・アクセスは、次の項で説明するように、インストール中およびインストール後に構成できます。

- インストール中のディレクトリ・アクセス構成
- インストール後のディレクトリ・アクセス構成

#### インストール中のディレクトリ・アクセス構成

**Net8 Configuration Assistant** は、ソフトウェアのインストール後、Oracle ユニバーサル・インストーラによって起動されます。これは、ディレクトリへのアクセスを構成可能にします。ディレクトリ・アクセスの構成は、次の項で説明するように、インストール・モードによって異なります。

- サーバーでの Custom インストール
- クライアント・インストール

**サーバーでの Custom インストール** サーバーでの Custom インストール後、Net8 Configuration Assistant により、ディレクトリへのアクセスを構成するよう求められます。ディレクトリ・アクセス構成により、次のことが可能となります。

- Net8 Configuration Assistant の後に実行される Oracle Database Configuration によるディレクトリでのデータベース・サービス・エントリの作成
- Net8 Assistant による、データベース・サービス・エントリおよびネット・サービス名エントリの Net8 属性の変更およびディレクトリでのネット・サービス名の作成
- サーバーによる、ディレクトリでのデータベース・サービスおよびネット・サービス名エントリの検索

---

**注意:** ディレクトリ・アクセス構成は、サーバーでの Typical または Minimal インストール中は使用不能です。

---

ディレクトリ・アクセスの構成中、Net8 Configuration Assistant により、次のディレクトリ・アクセス設定を構成するよう求められます。

- ディレクトリのタイプ
- ディレクトリの場所
- このサーバーが接続識別子を検索、作成および変更する**管理コンテキスト**



管理コンテキストは、**Oracle コンテキスト** (cn=OracleContext) を含むディレクトリ・エントリです。Oracle コンテキストは、すべての Oracle ソフトウェア間連情報が保管されるディレクトリ・サブツリーのルートです。

この情報は、クライアントが、アクセスするディレクトリおよび Oracle エントリを検索する際に読み込む、読取り専用 ldap.ora ファイルに格納されます。

Oracle コンテキストがディレクトリにない場合、Net8 Configuration Assistant により、それを作成するよう求められます。Oracle コンテキストが正常に作成された場合、認証ユーザーがディレクトリの次のグループに追加されます。

- OracleDBCreators (cn=OracleDBCreators,cn=OracleContext)
- OracleNetAdmins (cn=OracleNetAdmins,cn=OracleContext)
- OracleSecurityAdmins (cn=OracleSecurityAdmins,cn=OracleContext)

OracleDBCreators および OracleNetAdmins グループのメンバーであるので、このユーザーは、Oracle Database Configuration Assistant を使用してデータベース・サービスを作成でき、Net8 Assistant を使用してネット・サービス名の作成およびデータベース・サービスまたはネット・サービス名の Net8 属性変更をできるようになります。ディレクトリ管理者は、これらのグループに他のユーザーを追加できます。

#### 関連項目：

- NetAdmins グループへのユーザーの追加については、6-27 ページの「**OracleNetAdmins グループへのユーザーの追加と削除**」を参照してください。
- ユーザーの OracleDBCreators グループへの追加については、『Oracle8i Advanced Security 管理者ガイド』を参照してください。

また、Net8 Configuration Assistant では、**Oracle スキーマ**が作成されたかどうか検証されます。Oracle スキーマでは、Oracle エントリおよびその属性が定義されます。スキーマがない、あるいはバージョンが古い場合、スキーマの作成が求められます。

Net8 Configuration Assistant による構成が完了した後、Oracle Database Configuration Assistant では、データベースが作成されます。データベースのサービス名は、Oracle コンテキストの下で自動的に作成されます。

**関連項目：** Oracle のインストール・ガイドを参照してください。

**クライアント・インストール** ディレクトリ・ネーミング・メソッドの構成を選択すると、Net8 Configuration Assistant では、ディレクトリへのアクセスの構成が求められます。ディレクトリ・アクセスの構成により、クライアントによるディレクトリ内接続識別子エントリの検索が可能となります。ディレクトリ・アクセスが構成されていない場合、クライアントでは、ディレクトリ・ネーミングを使用できません。



Net8 Configuration Assistant では、通常、必要なディレクトリ・アクセス構成がクライアント・インストール中に実行され、次のものが、読取り専用 `ldap.ora` ファイルに格納されます。

ディレクトリ・アクセスの構成中、Net8 Configuration Assistant により、次のディレクトリ・アクセス設定を構成するよう求められます。

- ディレクトリのタイプ
- ディレクトリの場所
- このクライアントが接続識別子を検索する管理コンテキスト

この情報は、クライアントが、アクセスするディレクトリおよび Oracle エントリを検索する際に読み込む読取り専用 `ldap.ora` ファイルに格納されます。

また、Net8 Configuration Assistant では、Oracle スキーマがインストールされたかどうか検証されます。Oracle コンテキストまたは Oracle スキーマがサーバーにより構成されなかった場合は、クライアントでディレクトリ・アクセス構成を完了できません。

### インストール後のディレクトリ・アクセス構成

ディレクトリ・ネーミング・メソッドの使用を計画している場合、ディレクトリ・アクセスを構成する必要があります。ディレクトリ・アクセスは、Net8 Configuration Assistant を使用して、いつでも構成できます。

ディレクトリ・アクセスを構成するには、次の作業を実行します。

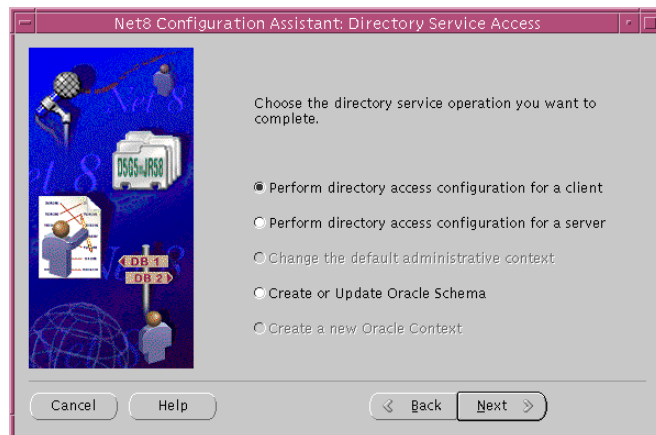
1. Net8 Configuration Assistant を起動します。
  - UNIX の場合、`$ORACLE_HOME/bin` の下にある `netca` を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Configuration Assistant」を選択します。

次のような「Welcome」ページが表示されます。





2. 「Directory Service Access Configuration」を選択し、「Next」をクリックします。  
「Directory Service Access」ページが表示されます。





オプション	説明
Perform directory access configuration for a client	<p>クライアントでディレクトリ・アクセスを構成する際にクリックします。このオプションにより、クライアントからディレクトリ・エントリの検索が可能となります。このオプションの使用により、クライアントでのディレクトリ内エントリの追加および変更が可能とはなりません。クライアントのディレクトリ・アクセス構成によって、次の情報を構成するよう求められます。</p> <ul style="list-style-type: none"><li>■ ディレクトリのタイプ</li><li>■ ディレクトリの場所</li><li>■ このクライアントが接続識別子を検索する管理コンテキスト</li></ul> <p><b>注意：</b> Oracle コンテキストまたは Oracle スキーマがない場合には、ディレクトリ・アクセスのクライアントを構成できません。</p>



オプション	説明
Perform directory access configuration for a server	<p>サーバーでディレクトリ・アクセスを構成する際にクリックします。このオプションにより、サーバーからディレクトリ・エントリの追加、変更および検索が可能となります。サーバーのディレクトリ・アクセス構成によって、次の情報を構成するよう求められます。</p> <ul style="list-style-type: none"><li>■ ディレクトリのタイプ</li><li>■ ディレクトリの場所を指定します。</li><li>■ このサーバーが Oracle エントリにアクセスして作成する元となる管理コンテキスト</li></ul> <p>Oracle コンテキストがディレクトリにない場合、それを作成するよう求められます。</p> <p>Net8 Configuration Assistant でも、Oracle スキーマがディレクトリで作成されているかどうか検証します。スキーマがない、あるいはバージョンが古い場合、スキーマの作成が求められます。</p> <p>Oracle コンテキストが正常に作成された場合、認証ユーザーがディレクトリの次のグループに追加されます。</p> <ul style="list-style-type: none"><li>■ OracleDBCreators (cn=OracleDBCreators,cn=OracleContext)</li><li>■ OracleNetAdmins (cn=OracleNetAdmins,cn=OracleContext)</li><li>■ OracleSecurityAdmins (cn=OracleSecurityAdmins,cn=OracleContext)</li></ul> <p>OracleDBCreators および OracleNetAdmins グループのメンバーであるので、このユーザーは、Oracle Database Configuration Assistant を使用して、データベース・サービスを作成でき、Net8 Assistant を使用してネット・サービス名の作成およびデータベース・サービスまたはネット・サービス名の Net8 属性変更をできるようになります。ディレクトリ管理者は、これらのグループに他のユーザーを追加できます。</p> <p><b>関連項目：</b></p> <ul style="list-style-type: none"><li>■ OracleNetAdmins グループへのユーザーの追加については、6-27 ページの「<a href="#">OracleNetAdmins グループへのユーザーの追加と削除</a>」を参照してください。</li><li>■ ユーザーの OracleDBCreators グループへの追加については、『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。</li></ul>
Create or Update Oracle Schema	ディレクトリで Oracle スキーマを作成または更新する際にクリックします。

3. 適切なオプションを選択した後、ウィザードのプロンプトおよびオンライン・ヘルプに従って、ディレクトリ・アクセス構成を完了します。



## 作業 2: ネット・サービス名の作成（オプション）

Oracle Database Configuration Assistant により作成されたデータベース・サービス・エントリではなく、ネット・サービス名を使用するように、クライアントを構成できます。ネット・サービス名は、Oracle コンテキスト（cn=OracleContext）の下で作成されます。

---

**注意：** OracleNetAdmins グループのメンバーであるユーザーのみが、ディレクトリのネット・サービス・エントリを作成できます。サーバーのディレクトリ・アクセス中に Oracle コンテキストを作成した認証ユーザーは、自動的にこのグループのメンバーとなります。このグループからユーザーを追加または削除するには、6-27 ページの「[OracleNetAdmins グループへのユーザーの追加と削除](#)」を参照してください。

---

---

**注意：** tnsnames.ora ファイルまたは Oracle Names サーバーに格納されるネット・サービス名は、6-28 ページの「[tnsnames.ora ファイルからのネット・サービス名のエクスポート](#)」および 6-31 ページの「[Oracle Names サーバーからのネットワーク・オブジェクトのエクスポート](#)」に説明されているように、ディレクトリ・サーバーにエクスポートできます。

---

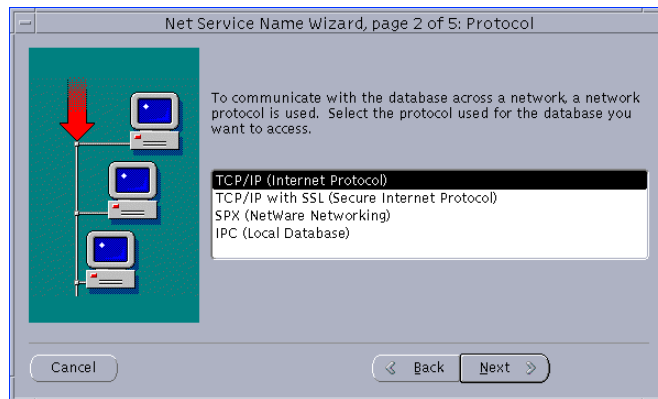
ディレクトリ・サーバーにネット・サービス名を作成するには、次の作業を実行してください。

1. サーバーのディレクトリ・アクセスが構成されたマシンで、Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Directory」->「Service Naming」を展開します。
3. ツールバーで「+」をクリックするか、「Edit」メニューから「Create」を選択します。Net Service Name Wizard が起動します。
4. 「Net Service Name」フィールドに名前を入力し、「Next」をクリックします。



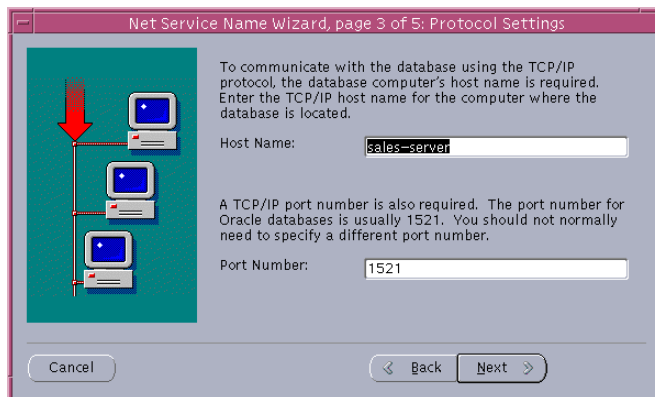


5. リスナーがリスニングするように構成されているプロトコルを選択します。このプロトコルも、クライアント上にインストールする必要があります。「Next」をクリックします。



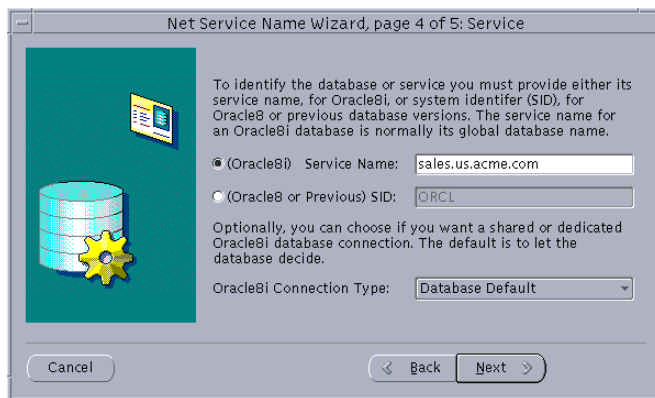
6. 提供されたフィールドで選択したプロトコルに対して適切なプロトコル・パラメータを入力し、「Next」をクリックします。





**関連項目：**プロトコル・パラメータの設定については、[付録 B](#) を参照してください。

7. リリースを選択後、宛先サービスを入力し、「Next」をクリックします。



宛先サービスが Oracle リリース 8.1 データベースである場合、「Oracle8i」をクリックし、「Service Name」フィールドにサービス名を入力します。宛先サービスが Oracle リリース 8.0 またはバージョン 7 の場合、「Oracle8 or Previous」をクリックして、「Database SID」フィールドに Oracle システム識別子 (SID) を入力します。

**関連項目：**使用するサービス名文字列の詳細は、6-2 ページの「[接続記述子の理解](#)」を参照してください。



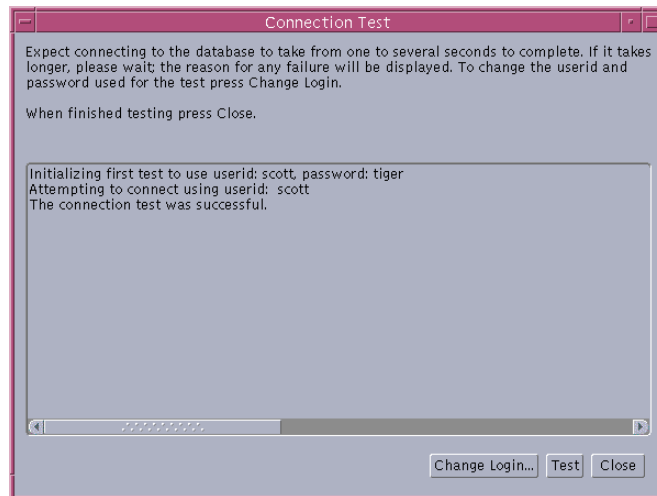
オプションで、このネット・サービス名の Oracle8i 接続タイプ・リストからデータベース接続タイプを選択できます。オラクル社では、Database Default のデフォルト設定の使用をお勧めします。**マルチスレッド・サーバー (multi-threaded server: MTS)** が、初期化パラメータ・ファイルで構成される場合、専用サーバーを選択して、MTS 構成を使用せずにリスナーに専用サーバーを起動させることができます。MTS が初期化パラメータ・ファイルで構成され、接続時に常に MTS が使用されるようにする場合は、共有サーバーを選択します。

**関連項目：**MTS 構成の詳細は、[第 9 章](#)を参照してください。

8. ネット・サービス名の動作を検証するには「Test」をクリックし、「Next」をクリックします。

このテストでは、データベースとリスナーが実行されている必要があります。実行されていない場合は、11-2 ページの「[Net8 制御ユーティリティの使用](#)」の手順に従ってコンポーネントを起動するか、または「Next」をクリックしてテストを省略し、手順 10 に進んでください。

テストが正常終了すると、「Connect Test」ダイアログ・ボックスに「The connect test was successful.」というメッセージが表示されます。



テストが正常終了しなかった場合は、次の処置を行ってください。

- データベースとリスナーが実行されていることを確認し、「Test」をクリックします。
- 「Change Login」をクリックして接続に使用するユーザー名とパスワードを変更し、「Test」をクリックします。



9. 「Close」をクリックして「Connect Test」ダイアログ・ボックスを閉じます。
10. 「Finish」をクリックして構成を保存し、Net Service Name Wizard を終了します。

### 関連項目：

- 複数のプロトコル・アドレスの構成方法については、8-2 ページの「[リスナー・プロトコル・アドレスのリスト作成](#)」を参照してください。
- 追加の接続データオプションの構成方法については、8-8 ページの「[拡張接続データ・パラメータの構成](#)」を参照してください。

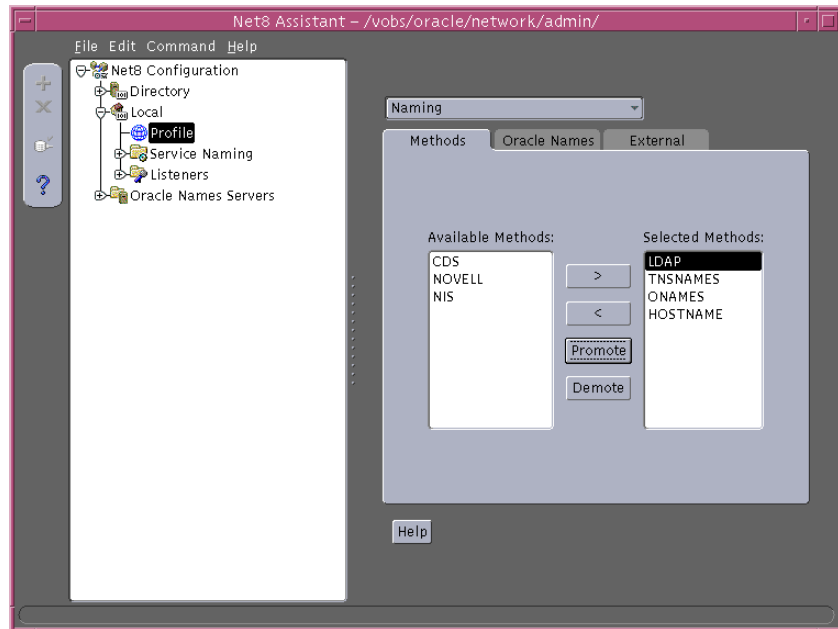
### 作業 3: 第 1 番目のネーミング・メソッドとしての LDAP の構成

データベース・サービスまたはネット・サービス名をディレクトリにエントリとして構成した後、ディレクトリ・ネーミングを `sqlnet.ora` ファイルの `NAMES.DIRECTORY_PATH` パラメータに指定する第 1 番目のメソッドとして設定します。このパラメータによって、Net8 が接続記述子の名前解決に使用するネーミング・メソッドの順序が指定されます。

ディレクトリ・ネーミングを第 1 番目のネーミング・メソッドに指定するには、次の手順に従います。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、`$ORACLE_HOME/bin` から `netasst` を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Local」 -> 「Profile」を展開します。
3. 右ペインのリストから、「Naming」を選択します。
4. 「Methods」タブをクリックします。
5. 「Available Methods」リストから「LDAP」を選択して、「>」ボタンをクリックします。
6. 「Selected Methods」リストで「LDAP」を選択し、「Promote」ボタンを使用して選択した項目をリストの最上位に移動します。





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

sqlnet.ora ファイルは、先頭に LDAP がリストされる NAMES.DIRECTORY\_PATH パラメータにより更新されます。

```
names.directory_path=(ldap, tnsnames, onames, hostname)
```

## 作業 4: リスナーの構成

サーバー上のリスナーは、ネット・サービス名用に構成したものと同一プロトコル・アドレスを使用してリスニングするように構成する必要があります。デフォルトでリスナーは、ポート 1521 の TCP/IP プロトコルを使用するようにすでに構成されています。

**関連項目：** リスナー構成の詳細は、[第 7 章](#)を参照してください。

## 作業 5: データベースへの接続

クライアントが検索中のエントリの管理コンテキストと一致する管理コンテキストが構成されているクライアントでは、次の構文を使用してデータベースに接続できます。

```
CONNECT username/password@connect_identifier
```



エントリの管理コンテキストと一致しない管理コンテキストで構成されているクライアントでは、接続文字列中の接続識別子を使用できません。かわりに、これらの接続では、エントリの識別名またはその絶対名が必要です。

### 関連項目：

- 絶対名の使用法は、2-39 ページの「[ディレクトリ・ネーミングを使用したクライアントの接続](#)」を参照してください。
- 接続識別子の構文規則は、11-7 ページの「[接続識別子の構文特性](#)」を参照してください。

## データベース・サービス・エントリに対する接続情報の変更

---

**注意：** OracleNetAdmins グループのメンバーであるユーザーのみが、データベース・サービス・エントリの Net8 属性を変更できます。サーバーのディレクトリ・アクセス中に Oracle コンテキストを作成した認証ユーザーは、自動的にこのグループのメンバーとなります。このグループからユーザーを追加または削除するには、6-27 ページの「[OracleNetAdmins グループへのユーザーの追加と削除](#)」を参照してください。

---

データベース・サービス・エントリは、ディレクトリ・アクセスがサーバーで構成された後データベース作成中に Oracle Database Configuration Assistant により作成されます。エントリは、Oracle コンテキスト (cn=OracleContext) の下に含まれます。

ディレクトリに格納されるデータベース・サービス・エントリには、ネットワーク・ルート情報は含まれません。接続記述子には、プロトコル・アドレス経由のリスナーの場所を含むネットワーク・ルートが含まれる必要があります。これらが含まれていない場合、クライアントはリスナーの場所を検索できません。

---

**注意：** Oracle Names サーバーや tnsnames.ora に格納されるデータベース・サービスは、6-28 ページの「[tnsnames.ora ファイルからのネット・サービス名のエクスポート](#)」および 6-31 ページの「[Oracle Names サーバーからのネットワーク・オブジェクトのエクスポート](#)」で説明するようにディレクトリにエクスポートされます。

---

データベース・サービスのネットワーク・ルート情報を作成または変更するには、次の作業を実行してください。

1. サーバーのディレクトリ・アクセスが構成されたマシンで、Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。



- Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
- 2. ナビゲータで、「Directory」->「Service Naming」を展開します。
- 3. データベース・サービスを選択します。右ペインには、現行の宛先サービス名が表示されます。
- 4. 「Address Configuration」ボックスの「+」ボタンをクリックします。新規のアドレス・タブが表示されます。
- 5. プロトコルを選択して、そのプロトコルの適切なプロトコル・パラメータ情報を入力します。

**関連項目：**プロトコル・パラメータの設定については、[付録 B](#) を参照してください。

- 6. 右ペインで「Apply」をクリックします。

## OracleNetAdmins グループへのユーザーの追加と削除

Oracle コンテキストを作成したユーザーは、OracleNetAdmins (cn=OracleNetAdmins,cn=OracleContext) グループのメンバーです。ldapmodify などのディレクトリ・ツールを使用して、Oracle コンテキストを作成したユーザーまたはディレクトリ管理者は、このグループにユーザーを追加できます。

ldapmodify ツールを使用して OracleNetAdmins グループへユーザーを追加するには、次の作業を実行してください。

- 1. OracleNetAdmins グループへ追加するユーザーを指定する **LDAP Data Interchange Format (LDIF)** ファイルを作成します。次のサンプル LDIF ファイルで**識別名 (Distinguished Name: DN)** cn=OracleNetAdmins に独自の設定を行い、ユーザーを指定して使用できます。

```
dn: cn=OracleNetAdmins,cn=OracleContext,...
changetype: modify
add: uniquemember
uniquemember: <DN of user being added to group>
```

- 2. ユーザーを追加するには、次の ldapmodify 構文を使用します。

```
ldapmodify -h host -p port -D binddn
-w password -file ldif_file
```

引数	説明
-h host	ディレクトリ・サーバー・ホストを指定します。



引数	説明
-p <i>port</i>	ディレクトリ・サーバーのリスニング TCP/IP ポートを指定します。このオプションを指定しない場合、デフォルト・ポート (389) が使用されます。
-D <i>binddn</i>	ディレクトリ管理者またはユーザー DN を指定します。
-w <i>password</i>	ディレクトリ管理者またはユーザー DN のパスワードを指定します。
-f <i>ldif_file</i>	入力ファイル名を指定します。

ldapmodify ツールを使用して OracleNetAdmins グループからユーザーを削除するには、次の作業を実行してください。

1. OracleNetAdmins グループへ追加するユーザーを指定する LDIF ファイルを作成します。次のサンプル LDIF ファイルで識別名 (DN) cn=OracleNetAdmins に独自の設定を行い、ユーザーを指定して使用できます。

```
dn: cn=OracleNetAdmins,cn=OracleContext,...
changetype: modify
delete: uniquemember
uniquemember: <DN of user being delete from group>
```

2. ユーザーを追加するには、次の ldapmodify 構文を使用します。

```
ldapmodify -h host -p port -D binddn
-w password -file ldif_file
```

## tnsnames.ora ファイルからのネット・サービス名のエクスポート

tnsnames.ora ファイルがすでに存在する場合、そのネット・サービス名は、ディレクトリ・サーバーへエクスポートできます。エクスポートは、一度に1つのドメインで実行されます。

tnsnames.ora ファイルのエクスポートにより、新しいネット・サービス・エントリが現在選択された **Oracle コンテキスト** (cn=OracleContext) に作成されます。Oracle コンテキストを変更する必要がある場合には、Net8 Assistant で「Command」->「Directory Server」->「Change Oracle Context」を選択します。

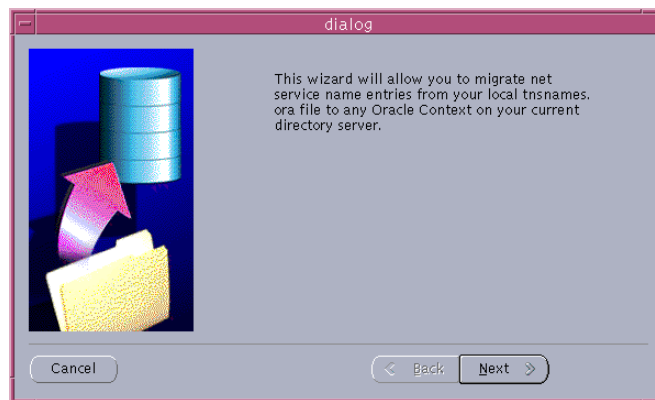
データのエクスポートを実行するコンピュータでは、サーバー・ディレクトリ・アクセスが構成されている必要があります。

**関連項目 :** 6-14 ページの「作業 1: サーバーおよびクライアントでのディレクトリ・アクセスの構成」を参照してください。

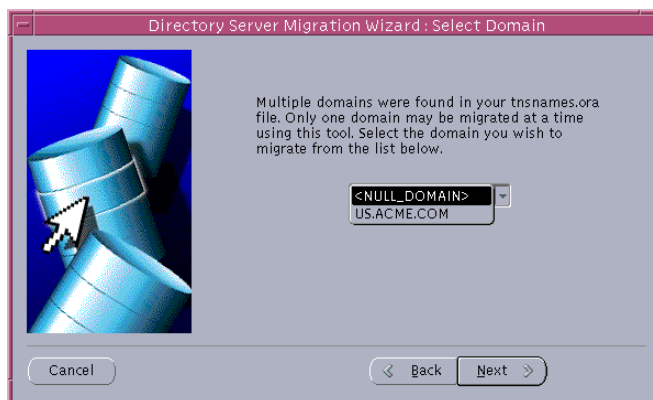


tnsnames.ora ファイルに含まれるネット・サービス名をエクスポートするには、次のようにします。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン->「プログラム」->「Oracle - HOME\_NAME」->「Network Administration」->「Net8 Assistant」を選択します。
2. 「Command」->「Directory」->「Import Net Service Names」を選択してください。「Directory Server Migration Wizard」が起動します。



3. 「Next」をクリックします。  
複数のドメインが検索された場合、「Select Domain」ページが表示されます。

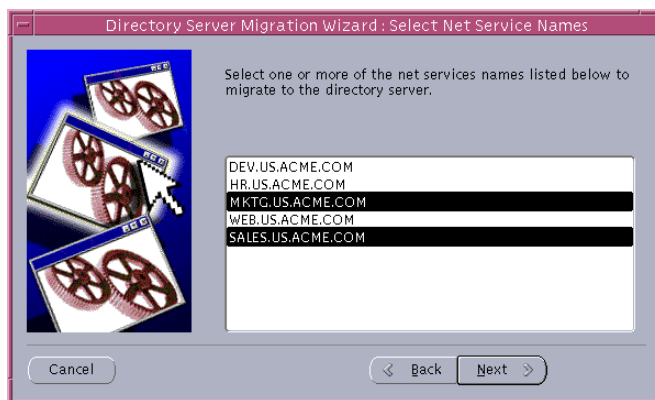




ネット・サービス名がドメインで修飾されていない場合、ステップ 4 で示されるように「Select Net Service Names」ページが表示されます。

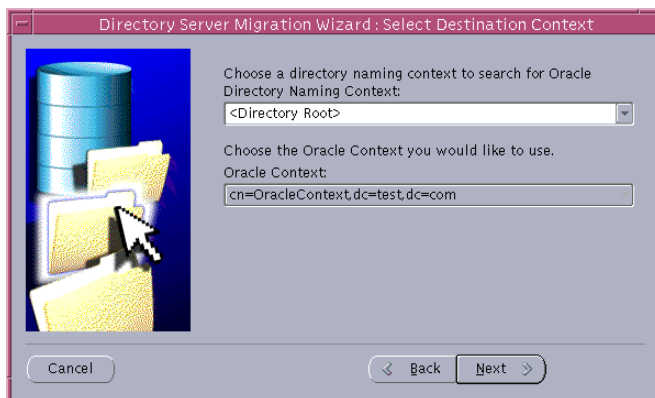
4. リストから、エクスポートするネット・サービス名のネットワーク・ドメインを選択します。

「Select Net Service Names」ページが表示されます。



5. エクスポートするリストからネット・サービス名を選択して、「Next」をクリックします。

「Select Destination Context」ページが表示されます。



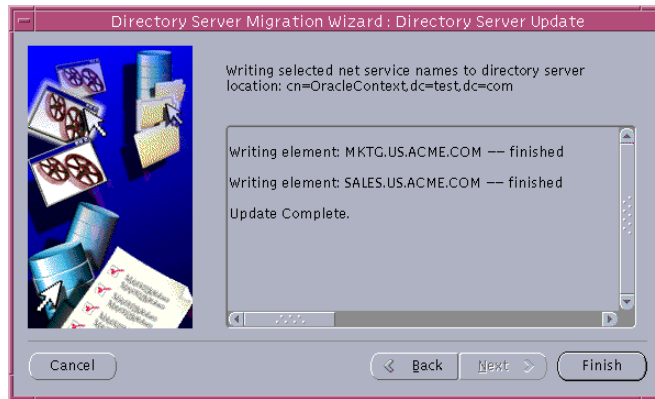
6. ディレクトリ・ネーミング・リストから Oracle コンテキストを含む **ディレクトリ・ネーミング・コンテキスト (directory naming context)** を選択します。ディレクトリ・



ネーミング・コンテキストは、1 つまたは複数の Oracle コンテキストを含むディレクトリ・サブツリーの一部です。

7. 選択したネット・サービス名をエクスポートする先の Oracle コンテキストを Oracle コンテキスト・リストから選択します。
8. 「Next」をクリックします。

ウィザードで、ネット・サービス名がエクスポートされます。



9. Directory Server Migration Wizard を終了するには、「Finish」をクリックします。

## Oracle Names サーバーからのネットワーク・オブジェクトのエクスポート

Oracle Names サーバーに格納されたデータベース・サービスおよびネット・サービス名は、ディレクトリまたは **LDAP Data Interchange Format (LDIF)** ファイルにエクスポートできます。データは指定されたドメインからエクスポートされます。ドメインにサブドメインへの定義権限がある場合には、そのデータもエクスポートできます。

**前提条件：**データのエクスポートを実行するコンピュータでは、6-14 ページの「**作業 1: サーバーおよびクライアントでのディレクトリ・アクセスの構成**」に説明されているように、サーバー・ディレクトリ・アクセスが構成されている必要があります。

データのエクスポート実行に関するステップおよび例は、次の項で説明されます。

- **作業 1: ディレクトリでの構造の作成**
- **作業 2: エクスポートするオブジェクト・リストの取得**
- **作業 3: ディレクトリ・サーバーへのオブジェクトのエクスポート**



- 類似 DIT へのドメインのエクスポート
- 非類似 DIT へのドメインのエクスポート

作業 1: ディレクトリでの構造の作成

Oracle Names オブジェクトの[ディレクトリ情報ツリー \(Directory Information Tree: DIT\)](#)を使用してディレクトリを構成すると Oracle Names で現在使用しているドメイン構造をレプリケートできます。または、まったく別の構造を作成することもできます。

オラクル社では、Net8 オブジェクトを格納する際に、Oracle コンテキスト (cn=OracleContext) を作成することをお勧めします。

**関連項目：** Oracle コンテキストの詳細は、[第 2 章](#)を参照してください。

作業 2: エクスポートするオブジェクト・リストの取得

Oracle Names ドメイン構造およびその構造内のオブジェクトを判断します。NAMESCTL では、この作業の使用に役立つコマンドが 3 つ提供されます。

コマンド	説明
LIST_DOMAINS	すべての管理ドメインをリストします。 <b>関連項目：</b> A-37 ページの <a href="#">LIST_DOMAINS</a>
LIST_DELEGATED	すべての委任されたドメインをリストします。 <b>関連項目：</b> A-36 ページの <a href="#">LIST_DELEGATED</a>
LIST_OBJECTS	すべての許可されたネットワーク・オブジェクトをリストします。 <b>関連項目：</b> A-38 ページの <a href="#">LIST_OBJECTS</a>

作業 3: ディレクトリ・サーバーへのオブジェクトのエクスポート

NAMESCTL ユーティリティでは、ネットワーク・オブジェクトが DUMP\_LDAP コマンドによりディレクトリへエクスポートされます。このコマンドにより、ネットワーク・オブジェクトの LDIF ファイルまたは直接ディレクトリへのエクスポートが可能となります。

LDIF ファイルヘデータをエクスポートする構文は次のとおりです。

```
NAMESCTL
NAMESCTL> dump_ldap [source] [destination] [options] {-f filename}
```

**注意：** 生成された LDIF ファイルは、ldap\_modify コマンドにより、後でディレクトリへロードされます。構文の使用方法は、ディレクトリのマニュアルを参照してください。



ディレクトリヘデータを直接エクスポートする構文は次のとおりです。

```
NAMESCTL
NAMESCTL> dump_ldap [source] [destination] [options] {-h host} {-p port} {-D
user_dn} {-w password}
```

**関連項目：** 引数の説明は、A-29 ページの [DUMP\\_LDAP](#) を参照してください。

## 類似 DIT へのドメインのエクスポート

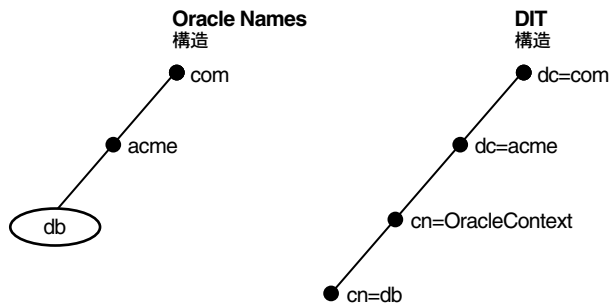
ディレクトリの DIT 構造が、現在の Oracle Names 構造に合致するドメイン・コンポーネント (dc) で設計されている場合、データのエクスポート方法については次の例を参照してください。

- [例 1: シングル・ドメインから同じ DIT ノードへのデータのエクスポート](#)
- [例 2: ドメイン・ツリーから同じ DIT へのデータのエクスポート](#)

### 例 1: シングル・ドメインから同じ DIT ノードへのデータのエクスポート

[図 6-1](#) では、acme.com の Oracle Names ドメイン構造が示されます。これには、db と呼ばれるデータベース・サービスが含まれます。ディレクトリ DIT は、Oracle Names 構造に合致するドメイン構造により設計されています。この DIT 構造が使用される場合、db は、cn=OracleContext,dc=acme,dc=com にエクスポートされます。

図 6-1 シングル・ドメインのエクスポート



Oracle Names から、構成された DIT 構造へデータをエクスポートする際は、次の構文のいずれかを使用できます。

```
NAMESCTL> dump_ldap acme.com (dn:cn=OracleContext,dc=acme,dc=com) -f sample.ldif
NAMESCTL> dump_ldap acme.com -x -f sample.ldif
```



構文の一行目では、宛先 DN が明示的に指定されます。宛先 DN は Oracle Names で使用されるドメイン・モデルに合致するため、構文の 2 行目のように、宛先 DN は指定する必要がありません。-x オプションは、cn=OracleContext を宛先 DN の左側に前方追加する際に使用されます。つまり、DN で cn=OracleContext を指定するかわりに -x を使用できます。このようにして、ディレクトリの cn=OracleContext RDN の下で、全オブジェクトを作成できます。

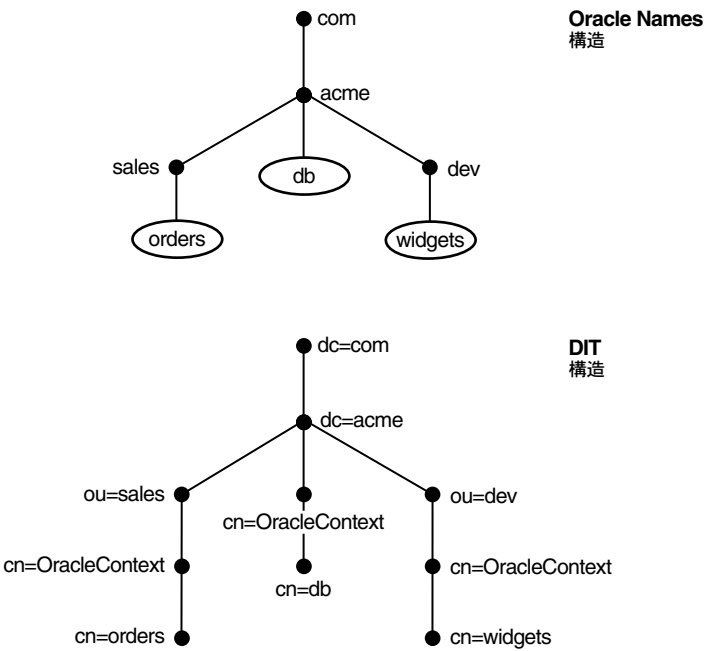
データベース・サービス db は、cn=OracleContext,dc=acme,dc=com にエクスポートされ、DN は (dn:cn=db,cn=OracleContext,dc=acme,dc=com) に位置します。

例 2: ドメイン・ツリーから同じ DIT へのデータのエクスポート

データは、Oracle Names のルート・ドメインおよびそのサブドメインから、類似 DIT が使用されるディレクトリへエクスポートできます。

図 6-2 では、acme.com のルート・ドメイン、sales.acme.com および dev.acme.com のサブドメインを含む Oracle Names 構造が示されています。db、orders および widgets のデータベース・サービスは、それぞれ acme.com、sales.acme.com および dev.acme.com に位置します。ディレクトリ DIT は、Oracle Names 構造と類似しています。

図 6-2 類似 DIT へのルートおよびサブドメインのエクスポート





次の構文は、データを `acme.com` のルート・ドメインおよびそのサブドメインから、構成された DIT 構造へエクスポートする際に使用します。

```
dump_ldap acme.com -R -x -f sample.ldif
```

次のテーブルでは、`acme.com`、`sales.acme.com` および `dev.acme.com` のデータベース・オブジェクトが、ディレクトリで DN へどのようにマップされているか示します。`-x` が使用されるため、全オブジェクトは、ディレクトリの `cn=OracleContext` RDN の下に作成されます。

Oracle Names のデータベース・オブジェクト	ディレクトリの新規 DN
<code>db.acme.com</code>	<code>dn:cn=db,cn=OracleContext,dc=acme,dc=com</code>
<code>orders.sales.acme.com</code>	<code>dn:cn=orders,cn=OracleContext,dc=sales,dc=acme,dc=com</code>
<code>widgets.dev.acme.com</code>	<code>dn:cn=widgets,cn=OracleContext,dc=dev,dc=acme,dc=com</code>

### 非類似 DIT へのドメインのエクスポート

ディレクトリの DIT 構造が、現在の Oracle Names ドメイン構造とは異なる DIT により設計されている場合、データのエクスポート方法については次の例を参照してください。

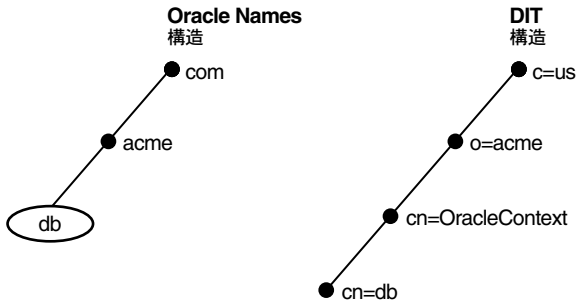
- [例 1: シングル・ドメインから非 DC ノードへのデータのエクスポート](#)
- [例 2: ドメイン・ツリーから非 DC DIT へのデータのエクスポート](#)
- [例 3: 複数ドメインから 1 つの DIT ノードへのデータのエクスポート](#)
- [例 4: エクスポート中のツリー構造の再編成](#)



## 例 1: シングル・ドメインから非 DC ノードへのデータのエクスポート

図 6-3 では、acme.com の Oracle Names ドメイン構造が示されます。これには、db と呼ばれるデータベース・サービスが含まれます。ディレクトリは、cn=OracleContext,o=acme,c=us の DIT によって設計されます。

図 6-3 1 つの DIT ノードへのシングル・ドメインのエクスポート



次の構文は、データを acme.com ドメインから、構成済 DIT 構造へエクスポートする際に使用します。

```
dump_ldap acme.com (dn:c=us,o=acme) -x -f sample.ldif
```

DIT は Oracle Names 構造と異なるため、宛先 DN は明示的に指定される必要があります。-x オプションは、cn=OracleContext を宛先 DN の左側に前方追加する際に使用されます。つまり、DN で cn=OracleContext を指定するかわりに -x を使用できます。このようにして、ディレクトリの cn=OracleContext RDN の下で、オブジェクトを作成できます。

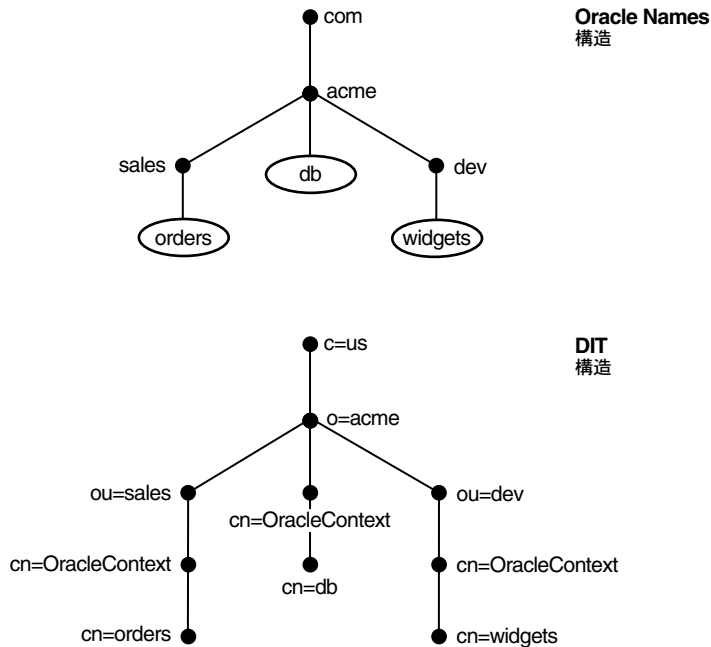
## 例 2: ドメイン・ツリーから非 DC DIT へのデータのエクスポート

データは、Oracle Names のルート・ドメインおよびそのサブドメインから、異なる DIT が使用されるディレクトリへエクスポートできます。宛先 DN の RDN のサブドメインはワイルドカードである（つまり、値が指定されない）必要があります。

図 6-4 では、acme.com のルート・ドメイン、sales.acme.com および dev.acme.com のサブドメインを含む Oracle Names 構造が示されます。db、orders および widgets のデータベース・サービスは、それぞれ acme.com、sales.acme.com および dev.acme.com に存在します。ディレクトリ DIT には、Oracle Names の acme.com に対応する o=acme,c=us の最高位構造があります。サブツリー、ou=sales および ou=dev は、Oracle Names のサブドメイン sales.acme.com および dev.acme.com に対応します。



図 6-4 異なる DIT へのルートおよびサブドメインのエクスポート



次の構文は、データを `acme.com` のルート・ドメインおよびそのサブドメインから、構成された DIT 構造へエクスポートする際に使用します。

```
dump_ldap acme.com (dn:ou,o=acme,c=us) -R -x -f sample.ldif
```

`organizationalUnitName (ou)` は置換え可能である（つまり値がまったく含まれていない）ため、ソース・リージョンにある `acme.com` の `sales` および `dev` サブドメインは、`ou` にマップできます。

次のテーブルでは、`acme.com`、`sales.acme.com` および `dev.acme.com` のデータベース・オブジェクトが、ディレクトリで DN へどのようにマップされているか示します。`-x` が使用されるため、全オブジェクトは、ディレクトリの `cn=OracleContext` RDN の下に作成されます。



Oracle Names のデータベー ス・オブジェクト	ディレクトリの新規 DN
db.acme.com	dn:cn=db,cn=OracleContext,o=acme,c=us
orders.sales.acme.com	dn:cn=orders,cn=OracleContext,ou=sales, o=acme,c=us
widgets.dev.acme.com	dn:cn=widgets,cn=OracleContext,ou=dev, o=acme,c=us

acme.com に mktg.dept.acme.com のサブドメインが含まれている場合、このサブドメインのネットワーク・オブジェクトはエクスポートされません。これは、宛先 DN (dn:ou,o=acme,c=us) では、1 レベルのサブドメインのみが許可されるためです。mktg.dept.acme.com からオブジェクトをエクスポートするには、次の構文が必要です。

```
dump_ldap acme.com (dn:ou,ou,o=acme,c=us) -x -f sample.ldif
```

これによって、サブドメインを最大 2 レベルまでエクスポートできます。追加ワイルド・カードを追加することにより、任意の深さを指定できます。

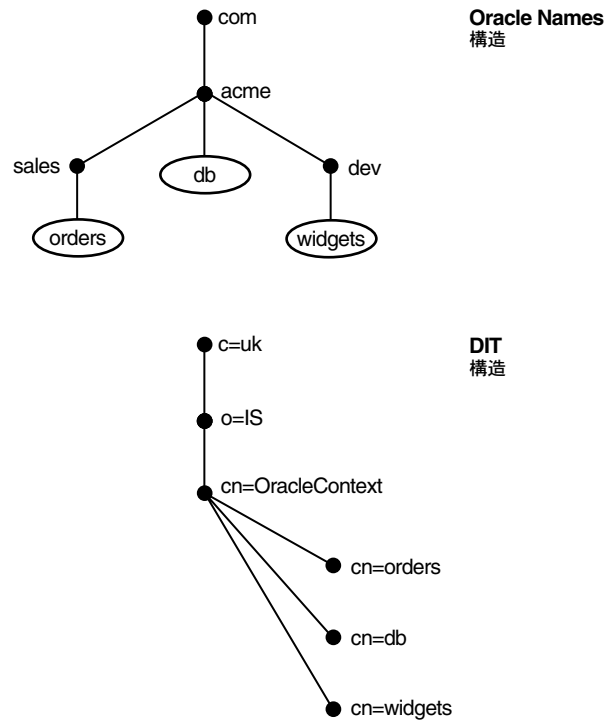
**例 3: 複数ドメインから 1 つの DIT ノードへのデータのエクスポート**

データは、複数のドメインから宛先 DIT の 1 つのノードへエクスポートできます。

図 6-5 では、acme.com のルート・ドメイン、sales.acme.com および dev.acme.com のサブドメインを含む Oracle Names 構造が示されます。db、orders および widgets のデータベース・サービスは、それぞれ acme.com、sales.acme.com および dev.acme.com に存在します。ディレクトリ DIT には、Oracle Names サブドメインに対応するサブツリーを含まない o=IS,c=uk の構造が含まれます。



図 6-5 1 つの DIT ノードへのルートおよびサブドメインのエクスポート



すべてのデータは、ルート・ドメインおよびサブドメインから DIT 内の `cn=OracleContext,o=IS,c=uk` へ次の構文を使用してエクスポートできます。

```
dump_ldap acme.com (dn:o=IS,c=uk) -R -x -f sample.ldif
```

次のテーブルでは、`acme.com`、`sales.acme.com` および `dev.acme.com` のデータベース・オブジェクトが、ディレクトリで DN へどのようにマップされているか示します。`-x` が使用されるため、全オブジェクトは、ディレクトリの `cn=OracleContext` RDN の下に作成されます。

Oracle Names のデータベース・オブジェクト	ディレクトリの新規 DN
db.acme.com	dn:cn=db,cn=OracleContext,o=IS,c=uk
orders.sales.acme.com	dn:cn=orders,cn=OracleContext,o=IS,c=uk
widgets.dev.acme.com	dn:cn=widgets,cn=OracleContext,o=IS,c=uk



サブドメインの1つに db という名前のデータベース・オブジェクトが含まれる場合、その名前が db.acme.com からエクスポートされる db オブジェクトと衝突するため、それはエクスポートされません。

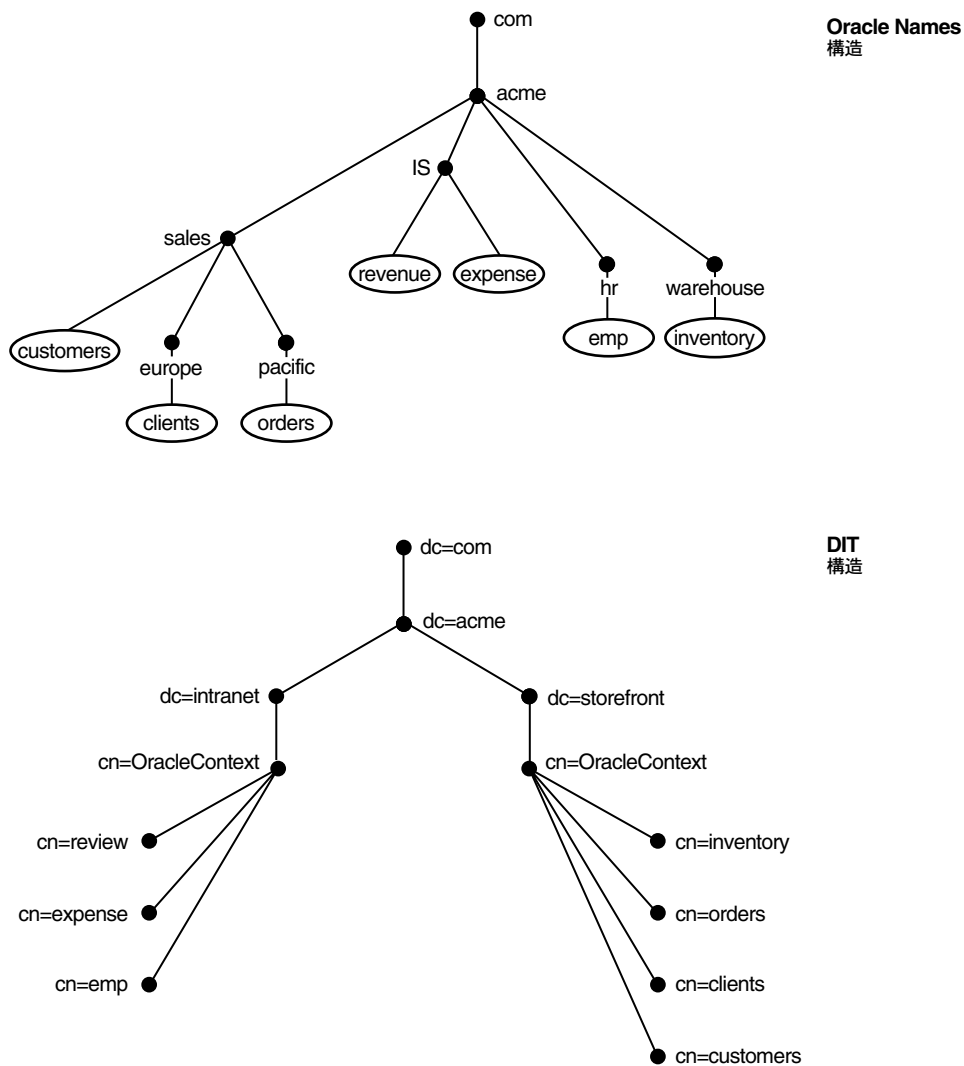
### 例 4: エクスポート中のツリー構造の再編成

前述の例では、非 DC DIT へのデータのエクスポートおよび複数ドメインから DIT 内の1つのノードへのデータのエクスポートについて紹介しました。この例では、これら2つのエクスポート・タイプを組み合わせて、まったく異なる構造を持つ DIT へのデータのエクスポート方法を説明します。

6-41 ページの図 6-6 では、acme.com のルート・ドメインおよび4つのサブドメイン（それぞれ少なくとも1つのデータベース・サービス・オブジェクトを含む）を含む Oracle Names 構造が示されます。ディレクトリ DIT には、Oracle Names の acme.com に対応する最高位構造 dc=acme,dc=com があります。2つのサブツリー dc=intranet および dc=storefront は、Oracle Names のサブドメインとは無関係です。



図 6-6 2つのサブツリーへの複数ドメインのエクスポート





データを Oracle Names 構造から DIT へエクスポートするには、各ドメインを個別にエクスポートする必要があります。

```
dump_ldap IS.acme.com (dn:dc=intranet,dc=com,dc=acme) -x -f sample.ldif
dump_ldap hr.acme.com (dn:dc=intranet,dc=com,dc=acme) -x -f sample.ldif
dump_ldap warehouse.acme.com (dn:dc=storefront,dc=com,dc=acme) -x -f sample.ldif
dump_ldap sales.acme.com (dn:dc=storefront,dc=com,dc=acme) -R -x -f sample.ldif
```

データベース・オブジェクトは、最初の 2 つの dump\_ldap コマンドにより、cn=Oraclecontext,dc=intranet,dc=acme,dc=com にエクスポートされ、最後の 2 つの dump\_ldap コマンドにより、cn=Oraclecontext,dc=storefront,dc=acme,dc=com へエクスポートされます。dump\_ldap sales.acme.com コマンドの -R によって、データベース・サービス・オブジェクトを sales.acme.com、europe.sales.acme.com および pacific.sales.acme.com からエクスポートできます。

次のテーブルでは、Oracle Names ドメインのデータベース・オブジェクトが、ディレクトリの DN にどのようにマップされているかを示します。-x が使用されるため、全オブジェクトは、ディレクトリの cn=OracleContext RDN の下に作成されます。

Oracle Names のデータベース・オブジェクト	ディレクトリの新規 DN
revenue.IS.acme.com	dn:cn=revenue,cn=OracleContext,dc=intra,dc=acme,dc=com
expense.IS.acme.com	dn:cn=expense,cn=OracleContext,dc=intra,dc=acme,dc=com
emp.hr.acme.com	dn:cn=emp,cn=OracleContext,dc=intra,dc=acme,dc=com
inventory.warehouse.com	dn:cn=inventory,cn=OracleContext,dc=storefront,dc=acme,dc=com
customer.sales.acme.com	dn:cn=customer,cn=OracleContext,dc=storefront,dc=acme,dc=com
clients.europe.sales.acme.com	dn:cn=clients,cn=OracleContext,dc=storefront,dc=acme,dc=com
orders.pacific.sales.acme.com	dn:cn=orders,cn=OracleContext,dc=storefront,dc=acme,dc=com



## Oracle Names メソッドの構成

---

**注意：**この項では、リリース 8.1 を対象に構成方法について説明しています。既存のリリース 8.0 またはリリース 7.x の構成がある場合は、『Oracle8i 移行ガイド』を参照してください。

---

Oracle Names を使用すると、クライアント / サーバーによるグローバルなコンピュータ・ネットワークの設定と管理が容易になります。Oracle Names によって、ネットワーク・アドレスとデータベース・リンクの情報がネットワーク全体のすべてのノードから利用できるようになります。各データベース・サーバーのネットワーク・アドレスは、単純なサービス名によって識別されます。これによりクライアント・アプリケーションは、長いアドレスを使用せずに、この単純な名前を使用してデータベース接続を要求できます。Oracle Names では、ネットワーク・インフラストラクチャが変更されても、ユーザーとアプリケーションはその影響を受けずに済みます。Oracle Names は、ネットワーク・サービス名の集中的な管理を提供します。

Oracle Names の構成に必要な作業は、次のとおりです。

作業 1: Oracle Names オプションの検討

作業 2: 必要なコンポーネントのインストール

作業 3: Oracle Names サーバーの作成

作業 4: Oracle Names サーバーを使用するためのクライアントとデータベース・サーバーの構成

作業 5: リスナーの構成

作業 6: Oracle Names サーバーへのデータベースの登録

作業 7: (オプション) 委任管理リージョンへのドメインの委任

作業 8: データベースへの接続

### 作業 1: Oracle Names オプションの検討

Oracle Names サーバーを作成する前に、次の項目について決定する必要があります。

- Oracle Names サーバー間でサービス登録データを継続的にレプリケートするか、または同一リージョン内のすべての Oracle Names サーバーがそれぞれの登録データを Oracle データベースに格納する必要があるかどうか。
- 1 つ以上の**管理リージョン (administrative region)** に対するサポートが必要かどうか。管理リージョンは、1 つ以上の**ドメイン (domain)** に存在する Oracle Names サーバーの集合であり、データベースなどのネットワーク・オブジェクトをグループ化したもの



です。複数の管理リージョンのあるネットワークには、1つの**ルート管理リージョン (root administrative region)** および1つ以上の**委任管理リージョン (delegated administrative region)** が必要です。委任管理リージョンには、ルートの直属の子リージョンの役割を果たす代替リージョン内の Oracle Names サーバーのアドレスとドメインが格納されます。

**関連項目 :** 4-8 ページの「[Oracle Names](#)」を参照してください。

## 作業 2: 必要なコンポーネントのインストール

次の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ます。

- Oracle Names サーバーとして指定するマシン上の Oracle Names
- クライアント上の Net8 Client または SQL\*Net Client
- サーバー上の Net8 Server または SQL\*Net Server

**重要 :** UNIX プラットフォームでは、Oracle Universal Installer によって \$ORACLE\_HOME/network/names ディレクトリは作成されません。このディレクトリは、正しく構成作業を実行するために必要です。このディレクトリが存在しない場合は、作成してください。

## 作業 3: Oracle Names サーバーの作成

Oracle Names サーバーを作成する方法は、リージョン・データの格納方法によって異なります。データは次のようにして格納できます。

- Oracle Names サーバーにアクセス可能な Oracle データベース内の、リージョン・データベースと呼ばれる表に格納します。  
  
Oracle Names サーバーに追加した情報は、データベースに格納される他、Oracle Names サーバーのキャッシュ・ファイルにも格納されます。データベースが利用できない場合は、キャッシュ・ファイル内の情報が使用されます。
- Oracle Names サーバー間でデータをレプリケートして、ディスクへのチェックポイントを実行します。

チェックポイント・ファイルは、UNIX プラットフォームでは \$ORACLE\_HOME/network/names に、Windows NT では ORACLE\_HOME¥network¥names に格納されます。

該当するファイル	説明
ckcfig.ora	リージョン・データベースの ONRS_CONFIG テーブルに格納された構成パラメータのバックアップ・コピー。



該当するファイル	説明
ckpcch.ora	リモート・リージョンから取り出された後、キャッシュされ、まだ期限切れになっていない現行の無保証データを含む。
ckpreg.ora	リージョンに対するすべての正当データを含む。Oracle Names サーバーでリージョン・データベースが使用されている場合、このチェックポイント・ファイルは、最終再ロード時点でのテーブルのリージョン・データのコピーです。このデータは、データベースがアクセス不能の場合、Oracle Names Server の起動時に使用されます。  サーバーでデータベースが使用されていない場合、このファイルのみが持続ストレージであり、起動時に Oracle Names サーバーによりロードされます。Oracle Names サーバーでは、そのリージョンに他の Oracle Names サーバーがない場合、リージョン・チェックポイント・ファイルのデータをカレントと見なします。この場合、ファイルは、リージョンの最終更新時のものでカレントとされます。

次の項では、両方のモードおよび構成が不要なデフォルト Oracle Names サーバーを説明します。

- [デフォルト Oracle Names サーバー](#)
- [データベースでのテーブルの作成](#)
- [チェックポイント実行済ファイルのレプリケート](#)

どちらの方法を選択するにしても、オラクル社では Oracle Names サーバーがダウンした場合に備えて複数の Oracle Names サーバーを作成するようにお勧めしています。

### デフォルト Oracle Names サーバー

Oracle Names サーバーは構成せずに実行できます。名前が `names.ora` ファイルで構成されている場合、名前はデフォルトで `ONAMES_host` と構成されます。リスニング・アドレスは、ローカル・ホスト・ポート 1575 上の TCP/IP または他のプロトコルの予約済アドレスにデフォルトで設定されます。

**関連項目 :** デフォルト・アドレス情報に関する Oracle 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。

`NAMES.DOMAINS` パラメータが、`names.ora` ファイルに構成されていない場合、Oracle Names サーバーでは、ルート・ドメインの認可レベルが仮定されます。

この Oracle Names サーバーを使用する場合は、6-53 ページの「[作業 2: Oracle Names サーバーの検出](#)」に進んでください。



## データベースでのテーブルの作成

Oracle データベースにサービス・データを格納するには、次の作業を実行します。

- 作業 1: データベース・サーバーの構成
- 作業 2: Oracle Names サーバーの作成
- 作業 3: Oracle Names サーバーの起動

**作業 1: データベース・サーバーの構成** データベース・サーバーを構成するには、次の手順に従います。

1. まだデータベースが実行されていない場合は、データベースを起動します。データベースがすでに起動されている場合は、手順 2 に進みます。

- a. INTERNAL としてデータベースに接続します。

```
SQL> CONNECT internal/password
```

INTERNAL ユーザー・アカウントの *password* は、デフォルトで ORACLE です。

- b. データベースを起動します。

```
SQL> STARTUP;
```

2. SYSTEM ユーザーとしてデータベースに接続します。

```
SQL> CONNECT system/password
```

SYSTEM ユーザー・アカウントの *password* は、デフォルトで MANAGER です。

3. ユーザー・アカウントとパスワードを作成します。

```
SQL> CREATE user user  
identified by password  
default tablespace users  
temporary tablespace temp;
```

4. namesini.sql スクリプトを実行します。このファイルは、UNIX では \$ORACLE\_HOME/network/admin に、Windows NT プラットフォームでは ORACLE\_HOME¥network¥admin¥names にあります。このスクリプトによって、Oracle Names が情報を格納するために必要な表が作成されます。オプションで namesupg.sql を実行し、古い表をアップグレードします。

```
SQL> CONNECT user/password  
SQL> @oracle_home/network/admin/namesini.sql;
```

**作業 2: Oracle Names サーバーの作成** Oracle Names がインストールされている各コンピュータに Oracle Names サーバーが必要な場合、Oracle Net8 Assistant を使用して Oracle Names サーバーを作成します。



1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、Oracle Names サーバーを展開します。
3. 「Edit」->「Create」を選択するか、または「+」ボタンをクリックして、新規の Oracle Names サーバーを作成します。

Names Wizard が起動します。

このウィザードでは、Oracle Names サーバーを作成し、指示に従って構成作業を進めていく過程で、次の項目の入力を求めるプロンプトが表示されます。

- Oracle Names サーバーの一意な名前
- Oracle Names サーバーのプロトコル・アドレス  
TCP/IP を選択する場合、オラクル社では正規に登録されたデフォルトのポート番号である 1575 を Oracle Names サーバーに使用することをお薦めしています。
- データベースに情報を格納するか、または Oracle Names サーバー間で情報をレプリケートするかを選択  
(「Use a region database.」をクリックします)
- データベースのリスナーのプロトコル・アドレス  
TCP/IP を選択する場合、オラクル社では正規に登録されたデフォルトのポート番号である 1521 をリスナーに使用することをお薦めしています。
- データベース・ユーザー ID、パスワード、およびサービス名または SID
- この Oracle Names サーバーがルート管理リージョンにあるかどうかの識別

この Oracle Names サーバーがルート管理リージョンにあると指定した場合、構成作業は完了です。

この Oracle Names サーバーがルート管理リージョンにないと指定した場合は、委任管理リージョンにあるものと見なされます。その場合ウィザードは、ローカル管理リージョンのドメイン名と、ルート管理リージョンにある Oracle Names サーバーのアドレスを求めるプロンプトを表示してから完了します。

Wizard が完了すると、次のメッセージが表示されます。

```
「A Names Server, onames_server, has been created with default
settings.Use the Configure Server section to modify the
default configuration.」
```

4. 「File」->「Save Network Configuration」を選択します。



5. Oracle Names サーバーをさらに追加する場合は、手順 2 ～ 4 を繰り返してリージョンに Oracle Names サーバーを作成します。Oracle Net8 Assistant では、1 台のコンピュータ上に複数の Oracle Names サーバーを作成することをサポートしていません。

Oracle Net8 Assistant では、少なくとも次の項目を設定して `names.ora` ファイルを作成します。

- Oracle Names サーバーの名前 (`NAMES.SERVER_NAME`)
- Oracle Names サーバーのリスニング用プロトコル・アドレス (`NAMES.ADDRESSES`)
- データベース情報 (`NAMES.ADMIN_REGION`)

次に、`names.ora` ファイルを内容に関する注釈付きで示します。

```
# Oracle Names サーバーの名前。名前には、この Oracle Names サーバーのあるドメイン名が
# 含まれている必要があります。
names.server_name=namesvr2.com

# Oracle Names サーバーのリスニング用プロトコル・アドレス。
names.addresses=
  (address=(protocol=tcp) (host=namesrv2-pc) (port=1575))
  [(address=...)]

# データベース・リポジトリ情報。
names.admin_region=
  (region=
    (description=
      (address=(protocol=tcp) (host=sales-server) (port=1521))
      (connect_data=
        (service_name=sales.us.acme.com))
      (userid=system)
      (password=password)
      (name=local_region)
      (refresh=86400)
      (retry=60)
      (expire=600))
    # Oracle Names サーバーが委任管理リージョンにある場合、ルート管理リージョンにおける
    # Oracle Names サーバーのアドレスを識別します。
    names.domain_hints=
      (address=(protocol=tcp) (host=namesrv1-pc) (port=1575))

# このリージョンにより制御されるドメインおよび TTL を指定します。これがルート管理リージョン
# である場合、ルート・ドメインを適切に識別するには、NAME= (NULL) が必要です。
names.domains=
  (domain=
    (name=)
    (min_ttl=86400))
```



**関連項目：**アドレス情報に必要なプロトコル構文については、[付録 B](#) を参照してください。

### 複数のドメインの管理

リージョンに複数のドメインを管理させる場合は、Oracle Net8 Assistant を使用して NAMES.DOMAINS パラメータに追加のドメインを指定します。

1. ナビゲータで、Oracle Names サーバーを展開します。
2. Oracle Names サーバーを選択します。
3. 右ペインのリストから、「Configure Server」を選択します。
4. 「Domains」タブをクリックします。
5. 「Domain Name」フィールドにドメイン名および time-to-live 情報を入力し、「Add」をクリックします。
6. 追加ドメインごとに、手順 5 を繰り返します。
7. 「File」->「Save Network Configuration」を選択します。

次の例では、NAMES.DOMAINS にルート、com、oracle.com および hq.oracle.com ドメインのリストが含まれています。hq.oracle.com を定義するには、hq.oracle.com より前に位置するすべてのドメインを定義しておく必要があります。

```
names.domains=
(domain_list=
  (domain=
    (name=)
    (min_ttl=86400))
  (domain=
    (name=com)
    (min_ttl=86400))
  (domain=
    (name=oracle.com)
    (min_ttl=86400))
  (domain=
    (name=hq.oracle.com)
    (min_ttl=86400)))
```

**作業 3: Oracle Names サーバーの起動** Oracle Net8 Assistant または NAMESCTL 制御ユーティリティを使用して、Oracle Names サーバーを起動します。



Oracle Net8 Assistant による操作	NAMESCTL 制御ユーティリティによる操作
<div>1. ナビゲータで、Oracle Names サーバーを展開します。</div> <div>2. Oracle Names サーバーを選択します。</div> <div>3. 右ペインのリストから、「Manage Server」を選択します。</div> <div>4. 「Control」タブをクリックします。</div> <div>5. 「Start」をクリックします。</div> <div>6. 「Apply」をクリックします。</div> <div>次のメッセージが表示されます。 Server started successfully.  Windows NT では、OracleORACLE_HOMENamesonames_server と呼ばれるサービスが作成されます。</div>	<div>コマンドラインから、次のように入力します。  NAMESCTL  次のエラー・メッセージが表示されても、無視します。  NNL-00024: 警告: SQLNET.ORA に優先ネーム・サーバーがありません。 NNL-00018: 警告: デフォルトのネーム・サーバーにコンタクトできません。</div> <div>NAMESCTL&gt; start</div> <div>NAMESCTL の START コマンドにより Oracle Names サーバーがメモリーにロードされ、実行開始の指示が出されます。Oracle Names サーバーは、起動時に構成とデータをロードします。</div> <div>Windows NT では、OracleORACLE_HOMENamesonames_server と呼ばれるサービスが作成されます。</div>

チェックポイント実行済ファイルのレプリケート

Oracle Names サーバーのデータをチェックポイント実行済ファイルによりレプリケートするには、次の作業を実行します。

- 作業 1: Oracle Names サーバーの作成
- 作業 2: Oracle Names サーバーの検出

**作業 1: Oracle Names サーバーの作成** Oracle Names がインストールされている各コンピュータに Oracle Names サーバーが必要な場合、Oracle Net8 Assistant を使用して Oracle Names サーバーを作成します。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、Oracle Names サーバーを展開します。
3. 「Edit」 -> 「Create」を選択するか、または「+」ボタンをクリックして、新規の Oracle Names サーバーを作成します。  
  
Names Wizard が起動します。



このウィザードでは、Oracle Names サーバーを作成し、指示に従って構成作業を進めていく過程で、次の項目の入力を求めるプロンプトが表示されます。

- Oracle Names サーバーの一意な名前
- Oracle Names サーバーのリスニング用プロトコル・アドレス  
TCP/IP を選択する場合、オラクル社では正規に登録されたデフォルトのポート番号である 1575 を Oracle Names サーバーに使用することをお薦めしています。
- データベースに情報を格納するか、または Oracle Names サーバー間で情報をレプリケートするかを選択  
(「Don't use a region database.」をクリックします)
- この Oracle Names サーバーが、リージョンで最初のネーム・サーバーであるかどうかの識別  
これがリージョンで最初の Oracle Names サーバーでなければ、別の Oracle Names サーバーを検出するか、またはリージョンに存在する別の Oracle Names サーバーのアドレスを指定するように求めるプロンプトが表示されます。
- この Oracle Names サーバーがルート管理リージョンにあるかどうかの識別

この Oracle Names サーバーがルート管理リージョンにあると指定した場合、構成作業は完了です。

この Oracle Names サーバーがルート管理リージョンにないと指定した場合は、委任管理リージョンにあるものと見なされます。その場合ウィザードは、ローカル管理リージョンのドメイン名と、ルート管理リージョンにある Oracle Names サーバーのアドレスを求めるプロンプトを表示してから完了します。

ウィザードが完了すると、次のメッセージが表示されます。

「A Names Server, onames\_server, has been created with default settings. Use the Configure Server section to modify the default configuration.」

4. 「ファイル」メニューから「Save Network Configuration」を選択します。
5. Oracle Names サーバーをさらに追加する場合は、手順 2～4 を繰り返してリージョンに Oracle Names サーバーを作成します。Oracle Net8 Assistant では、1 台のコンピュータ上に複数の Oracle Names サーバーを作成することをサポートしていません。

Oracle Net8 Assistant では、少なくとも次の項目を設定して `names.ora` ファイルを作成します。

- Oracle Names サーバーの名前 (NAMES.SERVER\_NAME)
- Oracle Names のリスニング用プロトコル・アドレス (NAMES.ADDRESSES)



次に、names.ora ファイルを内容に関する注釈付きで示します。

```
# Oracle Names サーバーの名前。名前には、この Oracle Names サーバーのあるドメイン名が
# 含まれている必要があります。
names.server_name=namesvr2.com

# Oracle Names サーバーのリスニング用プロトコル・アドレス。
names.addresses=
  (address=(protocol=tcp) (host=namesrv2-pc) (port=1575))
  [(address=...)]

# Oracle Names サーバーが委任管理リージョンにある場合、ルート管理リージョンにおける
# Oracle Names サーバーのアドレスを識別します。
names.domain_hints=
  (address=(protocol=tcp) (host=namesrv1-pc) (port=1575))

# このリージョンに制御されるドメインおよび TTL を指定します。これがルート管理リージョンで
# ある場合、ルート・ドメインを適切に識別するには、NAME= (NULL) が必要です。
names.domains=
  (domain=
    (name=)
    (min_ttl=86400))
```

**関連項目：**アドレス情報に必要なプロトコル構文については、[付録 B](#) を参照してください。

### 複数のドメインの管理

リージョンに複数のドメインを管理させる場合は、Oracle Net8 Assistant を使用して NAMES.DOMAINS パラメータに追加のドメインを指定します。

1. ナビゲータで、Oracle Names サーバーを展開します。
2. Oracle Names サーバーを選択します。
3. 右ペインのリストから、「Configure Server」を選択します。
4. 「Domains」タブをクリックします。
5. 「Domain Name」フィールドにドメイン名および time-to-live 情報を入力し、「Add」をクリックします。
6. 追加ドメインごとに、手順 5 を繰り返します。
7. 「File」->「Save Network Configuration」を選択します。

次の例では、NAMES.DOMAINS にルート、com、oracle.com および hq.oracle.com ドメインのリストが含まれています。hq.oracle.com を定義するには、hq.oracle.com より前に位置するすべてのドメインを定義しておく必要があります。



```
names.domains=  
  (domain_list=  
    (domain=  
      (name=)  
      (min_ttl=86400))  
    (domain=  
      (name=com)  
      (min_ttl=86400))  
    (domain=  
      (name=oracle.com)  
      (min_ttl=86400))  
    (domain=  
      (name=hq.oracle.com)  
      (min_ttl=86400)))
```

**作業 2: Oracle Names サーバーの検出** すべての Oracle Names サーバーの作成が終わると、同じリージョン内で最初の Oracle Names サーバーを除いて、各 Oracle Names がリージョン内にある他のすべての Oracle Names サーバーを検出するようにします。



Oracle Net8 Assistant による操作	NAMESCTL 制御ユーティリティによる操作
<div>1. 最初の Oracle Names サーバーをリージョンで起動します。</div> <div>a. ナビゲータで、Oracle Names サーバーを展開します。</div> <div>b. Oracle Names サーバーを選択します。</div> <div>c. 右ペインのリストから、「Manage Server」を選択します。</div> <div>d. 「Control」タブをクリックします。</div> <div>e. 「Server Operations」をクリックした後、「Start」をクリックします。</div> <div>f. 「Apply」をクリックして、Oracle Names サーバーを起動します。次のメッセージが表示されます。 「Server started successfully.」</div> <div>2. 2 番目の Oracle Names サーバーから、最初の Oracle Names サーバーを検出します。</div> <div>a. 「Tool」-&gt;「Discover Oracle Names サーバー」を選択します。次のメッセージが表示されます。 Discovered Oracle Names Server in the region. Please exit the tool and start again. Discover コマンドによって、UNIX では \$ORACLE_HOME/network/names に .sdns.ora ファイルが、Windows プラットフォームでは ORACLE_HOME%\network\%names に sdns.ora ファイルが作成されます。このファイルには、Oracle Names サーバーの名前とアドレスが記述されています。Oracle Names サーバーのアドレスを検索するときに、このファイルが読み取られます。 Oracle Names サーバーから応答がない場合、別の Oracle Names サーバーアドレスの入力を求めるダイアログが表示されます。特定の Oracle Names サーバーのネットワーク・アドレスを知っている場合は、それを入力します。</div> <div>3. 手順 1 に従って、2 番目の Oracle Names サーバーを起動します。</div> <div>4. リージョンに追加された各 Oracle Names サーバーについては、手順 2 ～ 3 を繰り返します。</div>	<div>1. 最初の Oracle Names サーバーをリージョンで起動します。</div> <div>NAMESCTL NAMESCTL&gt; start</div> <div>2. 2 番目の Oracle Names サーバーから、最初の Oracle Names サーバーを検出します。</div> <div>NAMESCTL NAMESCTL&gt; reorder_ns</div> <div>REORDER_NS コマンドによって、UNIX では \$ORACLE_HOME/network/names に .sdns.ora ファイルが、Windows プラットフォームでは ORACLE_HOME%\network\%names に sdns.ora ファイルが作成されます。このファイルには、最初の Oracle Names サーバーの名前とアドレスが記述されています。</div> <div>Oracle Names サーバーが見つからず、特定の Oracle Names サーバーのネットワーク・アドレスを知っている場合は、プロンプトに対してそのアドレスを入力します。たとえば、次のように指定します。</div> <div>NAMESCTL&gt; reorder_ns (address=(protocol=tcp) (host=mail-pc) (port=1575))</div> <div>3. 2 番目の Oracle Names サーバーをリージョンで起動します。</div> <div>NAMESCTL&gt; start</div> <div>4. リージョンにおいて最初の Oracle Names サーバーに接続した後、2 番目の Oracle Names サーバーに問い合せてそれ自体が登録されているか確認します。</div> <div>NAMESCTL&gt; query onames_server</div> <div>5. リージョンに追加された各 Oracle Names サーバーについては、手順 2 ～ 4 を繰り返します。</div>

関連項目：4-20 ページの「[検出の理解](#)」を参照してください。



## 作業 4: Oracle Names サーバーを使用するためのクライアントとデータベース・サーバーの構成

クライアントとデータベース・サーバー・マシンを構成して、Oracle Names サーバーを使用するには、次の作業を実行します。

1. 次のメソッドの 1 つを使用して、接続する Oracle Names サーバー s のリストを作成します。
  - 8-23 ページの「[優先 Oracle Names サーバーの構成](#)」の説明に従って、`sqlnet.ora` ファイルに、優先 Oracle Names サーバーの静的リストを作成します。
  - UNIX では `$ORACLE_HOME/network/names` に `..sdns.ora` ファイルを、Windows プラットフォームでは `ORACLE_HOME\network\names` に `sdns.ora` ファイルを動的に作成して Oracle Names サーバーを検出します。このファイルには、Oracle Names サーバーの名前とアドレスが問合せ順に記述されています。クライアント・マシンは、このファイルを読み取って問合せ先の Oracle Names サーバーのアドレスを検索します。次の作業は、Oracle Net8 Assistant または NAMESCTL ユーティリティを使用して実行します。

優先 Oracle Names サーバーは、`..sdns.ora` または `sdns.ora` ファイルで構成される Oracle Names サーバーより優先されます。

**関連項目：**検出の説明は、4-20 ページの「[検出の理解](#)」を参照してください。

`..sdns.ora` ファイルまたは `sdns.ora` ファイルを作成するには次の作業を実行します。

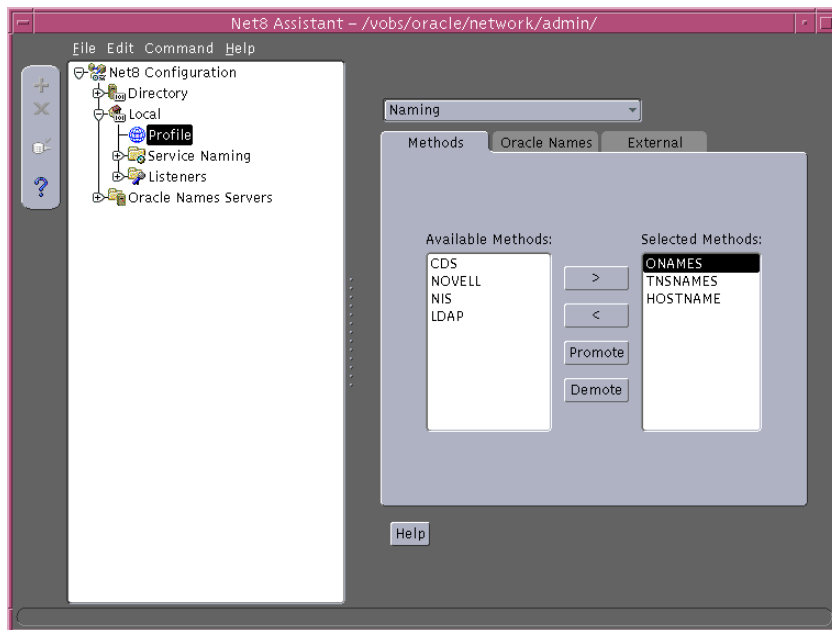


Oracle Net8 Assistant による操作	NAMESCTL 制御ユーティリティによる操作
<p>1. Oracle Net8 Assistant を起動します。</p> <p>- UNIX の場合、\$ORACLE_HOME/bin から netasst を実行します。</p> <p>- Windows NT の場合、「スタート」ボタン -&gt; 「プログラム」 -&gt; 「Oracle - HOME_NAME」 -&gt; 「Network Administration」 -&gt; 「Net8 Assistant」を選択します。</p> <p>2. 「Tool」 -&gt; 「Discover Oracle Names Server」を選択します。</p> <p>次のメッセージが表示されます。</p> <p>Discovered Oracle Names Server in the region. Please exit the tool and start again.</p> <p>Discover コマンドによって、UNIX では \$ORACLE_HOME/network/names に .sdns.ora ファイルが、Windows プラットフォームでは ORACLE_HOME¥network¥names に sdns.ora ファイルが作成されます。このファイルには、Oracle Names サーバーの名前とアドレスが記述されています。Oracle Names サーバーのアドレスを検索するときに、このファイルが読み取られます。</p> <p>Oracle Names サーバーから応答がない場合、別の Oracle Names サーバーアドレスの入力を求めるダイアログが表示されます。特定の Oracle Names サーバーのネットワーク・アドレスを知っている場合は、それを入力します。</p>	<p>他のすべての Oracle Names サーバーを検出します。</p> <p>NAMESCTL NAMESCTL&gt; reorder_ns</p> <p>REORDER_NS コマンドによって、UNIX では \$ORACLE_HOME/network/names に .sdns.ora ファイルが、Windows プラットフォームでは ORACLE_HOME¥network¥names に sdns.ora ファイルが作成されます。このファイルには、Oracle Names サーバーの名前とアドレスが記述されています。Oracle Names サーバーのアドレスを検索するときに、このファイルが読み取られます。</p> <p>Oracle Names サーバーが見つからず、特定の Oracle Names サーバーのネットワーク・アドレスを知っている場合は、プロンプトに対してそのアドレスを入力します。たとえば、次のように指定します。</p> <p>NAMESCTL&gt; reorder_ns (address=(protocol=tcip) (host=mail-pc) (port=1575))</p>

2. Oracle Names をネット・サービス名を解決するための第 1 番目の方法とする場合は、次の手順に従って sqlnet.ora ファイルに NAMES.DIRECTORY\_PATH パラメータを設定します。
- a. ナビゲータで、「Local」 -> 「Profile」を展開します。
  - b. 右ペインのリストから、「Naming」を選択します。
  - c. 「Methods」タブをクリックします。



- d. 「Available Methods」リストから「ONAMES」を選択して、「>」ボタンをクリックします。
- e. 「Selected Methods」リストで「ONAMES」を選択し、「Promote」ボタンを使用して選択した項目をリストの最上位に移動します。



- f. 「File」->「Save Network Configuration」を選択します。

これにより `sqlnet.ora` ファイルに、`NAMES.DIRECTORY_PATH` パラメータの先頭に `ONAMES` を含むエントリが入ります。

```
names.directory_path=(onames, tnsnames, hostname)
```

3. Oracle Names が使用されるクライアントに対して追加機能を構成できます。これらの機能を、次の表にリストし、説明します。



「Net8 Assistant」フィールド	sqlnet.ora パラメータ	説明
「Default Domain」	NAMES.DEFAULT_DOMAIN	クライアントから Oracle Names サーバーへの要求が最も多く発生するドメイン名スペースを指定します。デフォルト・ドメインが設定されると、Oracle Names 要求ですべての未修飾の名前に対してこの名前が自動的に追加されます。Net8 では、デフォルト・ドメインがデフォルトで NULL に設定されています。
「Maximum Wait Each Attempt」	NAMES.INITIAL_RETRY_TIMEOUT	Oracle Names サーバーからの応答を待機する時間を指定します。この時間が経過すると、別の Oracle Names サーバーに要求を再送信します。デフォルトで、Net8 は 15 秒間待機します。
「Attempts Per Names Server」	NAMES.REQUEST_RETRIES	Oracle Names サーバーのリスト全体を試行する回数を指定します。この回数試行しても応答がないと操作は失敗します。デフォルトで Net8 は、操作が失敗になる前に Oracle Names サーバーのリストを 1 回反復します。
「Maximum Open Connections」	NAMES.MAX_OPEN_CONNECTIONS	Oracle Names クライアントが一度にオープンできる接続数を指定します。デフォルト設定で、Net8 では、クライアントを一度に 10 接続までオープンできます。このデフォルト値は、ほとんどの状況で十分な数です。
「Initial Preallocated Requests」	NAMES.MESSAGE_POOL_START_SIZE	これによってクライアントのメッセージ・プール内に初期メッセージ数を事前割当てできます。これらのメッセージは、Oracle Names サーバーに対して将来要求するときに使用できます。デフォルトの設定で、Net8 はプールに 10 メッセージを割り当てます。このデフォルト値は、ほとんどの状況で十分な数です。

これらの機能を追加または構成するには、次の手順に従います。

- a. ナビゲータで、「Local」 -> 「Profile」を展開します。
- b. 右ペインのリストから、「Naming」を選択します。
- c. 「Oracle Names」タブをクリックします。
- d. 前の表に示されているオプションに必要なデータを入力します。
- e. 「File」 -> 「Save Network Configuration」を選択します。



## 作業 5: リスナーの構成

オペレーティング・システムの主プロトコルを使用しない場合などは、データベースのプロトコル・リスニング・アドレスをリスナーに構成します。

**関連項目：**リスナー構成の詳細は、[第 7 章](#)を参照してください。

さらに、listener.ora ファイルに `USE_PLUG_AND_PLAY_listener_name=ON` を設定します。これによりリスナーは、サービス名、SID、グローバル・データベース名などのデータベース情報を Oracle Names サーバーに登録できるようになります。

USE\_PLUG\_AND\_PLAY パラメータを設定するには、次の作業を実行します。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、`$ORACLE_HOME/bin` から `netasst` を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Local」 -> 「Listeners」を展開します。
3. リスナーを選択します。
4. 右ペインのリストから、「General Parameters」を選択します。
5. 「General」タブをクリックします。
6. 「Register Service with Oracle Names」をクリックします。
7. 「File」 -> 「Select Save Network Configuration」を選択します。

## 作業 6: Oracle Names サーバーへのデータベースの登録

6-43 ページの「[Oracle Names メソッドの構成](#)」の表に説明されているように、Oracle Names サーバーにはデータベースの場所、Oracle Names サーバーの場所、ネット・サービス名、別名、グローバル・データベース・リンクおよび Oracle Connection Manager を含む多くの種類の情報が格納されます。データベースはリスナーに情報を登録するため、これらの情報の一部は Oracle Names サーバーに自動的に登録されます。ただし、次に示す情報を格納するには、Oracle Net8 Assistant または NAMESCTL 制御ユーティリティを使用して登録する必要があります。

- [ネット・サービス名](#)
- [グローバル・データベース・リンクの資格証明とデータベース修飾子](#)
- [別名](#)

登録は、1 つの管理リージョンにつき 1 台の Oracle Names サーバーでのみ行われる必要があります。データがリージョン・データベースに格納されている場合は、この Oracle Names



サーバー以外の Oracle Names サーバーによってこの情報が参照され、リージョン・データベースを使用しない場合は、他の Oracle Names に対してこの情報が伝えられます。

これらの情報の登録後、11-11 ページの「[Oracle Net8 Assistant または NAMESCTL ユーティリティによるネットワーク・オブジェクトのテスト](#)」の説明に従って問い合わせる必要があります。

### ネット・サービス名

オラクル社では、各サービスについてネット・サービス名を作成するようにお薦めしています。ネット・サービス名を作成しない場合、接続を試みるユーザーはグローバル・データベース名を渡す必要があります。エンド・ユーザーにこの情報を提供するかわりに、サービスに対してネット・サービス名を作成するようにしてください。

次の作業は、Oracle Net8 Assistant または NAMESCTL 制御ユーティリティを使用して実行します。



## Oracle Net8 Assistant による操作

1. Oracle Net8 Assistant を起動します。  
- UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。  
- Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、Oracle Names サーバーを展開します。
3. Oracle Names サーバーを選択します。
4. 右ペインのリストから、「Manage Data」を選択します。
5. 「Net Service Names」タブをクリックします。
6. 「Add」をクリックします。
7. 「Service Name」フィールドに名前を入力します。ネット・サービス名は、ドメインで修飾できます。ドメインが指定されている場合、ネット・サービス名は自動的にドメインによって修飾されます。

**関連項目：** 8-15 ページの「クライアントのデフォルト・ドメインの構成」

8. リスナーがリスニングするように構成されているプロトコルを選択します。
9. 提供されたフィールドで選択したプロトコルに対して適切なプロトコル・パラメータを入力します。

**関連項目：** アドレス情報に必要なプロトコル構文については、[付録 B](#) を参照してください。

**注意：** 追加のアドレスは、「Address」タブの一番下の「+」をクリックすることによって作成できます。

10. 宛先サービスを入力します。宛先サービスが Oracle リリース 8.1 データベースである場合、「Service Name」フィールドにサービス名を入力します。宛先データベースのリリースが 8.1 より前のものである場合、「Oracle8i Release 8.0 Compatible Identification」をクリックし、その SID を「SID」フィールドに入力します。

**注意：** 宛先サービスに加えてアドバンス CONNECT\_DATA オプションを構成する場合は、「Advanced」をクリックします。

11. 「Execute」をクリックします。
12. 「File」 -> 「Save Network Configuration」を選択します。

## NAMECTL 制御ユーティリティによる操作

ネット・サービス名を作成し、Oracle Names サーバーに登録します。

NAMECTL

```
NAMECTL> register net_service_name -d
(description=[(address_list=(address= ...) (connect_data=(service_name|sid=)))])
```

たとえば、デフォルトの TCP/IP リスニング・アドレスを使用して sales というネット・サービス名を登録するには、次のように入力します。

```
register sales -d
(description=(address=(protocol=tcp) (host=sales-server) (port=1521)) (connect_data=(service_name=sales.us.acme.com)))
```

**関連項目：** アドレス情報に必要なプロトコル構文については、「[付録 B](#)」を、CONNECT\_DATA パラメータについては、C-40 ページの「[CONNECT\\_DATA セクション](#)」を参照してください。



これまで構成方法としてローカル・ネーミングを使用していて、Oracle Names 構成に移行する場合は、tnsnames.ora ファイルで使用されているネット・サービス名を Oracle Names サーバーにアップロードすることができます。ローカル名構成ファイルの情報を Oracle Names サーバーにロードするには、Oracle Net8 Assistant または NAMESCTL 制御ユーティリティを使用します。

Oracle Net8 Assistant による操作	NAMESCTL 制御ユーティリティによる操作
<div>1. Oracle Net8 Assistant を起動します。 - UNIX の場合、\$ORACLE_HOME/bin から netasst を実行します。 - Windows NT の場合、「スタート」ボタン -&gt; 「プログラム」 -&gt; 「Oracle - HOME_NAME」 -&gt; 「Network Administration」 -&gt; 「Net8 Assistant」を選択します。</div> <div>2. ナビゲータで、Oracle Names サーバーを展開します。</div> <div>3. Oracle Names サーバーを選択します。</div> <div>4. 右ペインのリストから、「Manage Data」を選択します。</div> <div>5. 「Net Service Names」タブをクリックします。</div> <div>6. 「Load」をクリックします。</div> <div>7. 「File」フィールドに、現行のマスター・ローカル名構成ファイルへのパスを入力します。</div> <div>8. 「Execute」をクリックします。</div> <div>9. 「File」 -&gt; 「Save Network Configuration」を選択します。</div>	<div>コマンドラインから、次のように入力します。 NAMESCTL NAMESCTL&gt; load_tnsnames tnsnames.ora</div>

**グローバル・データベース・リンクの資格証明とデータベース修飾子**  
オラクル社では、データベースごとに**グローバル・データベース・リンク (global database link)** を1つ用意することをお薦めします。**グローバル・データベース名 (global database name)** と同じであるグローバル・データベース・リンクは、Oracle Names サーバーに自動的に登録されます。

たとえば、グローバル・データベース名が sales.us.acme.com である場合、同じ名前のグローバル・リンクが Oracle Names サーバーに登録されます。表に対して SELECT 文を実行することにより、グローバル・データベース・リンクが正常に機能しているか検証できます。

```
SQL> SELECT * from emp@sales.us.acme.com
```



グローバル・データベース・リンクにユーザー名やパスワードからなる資格証明を指定しない場合には、グローバル・データベース・リンクの使用に必要な追加構成はありません。

グローバル・データベース・リンクのユーザー名およびパスワードを登録するには、Net8 Assistant または NAMESCTL 制御ユーティリティを使用します。NAMESCTL ユーティリティはグローバル・データベース・リンク登録機能に制限があるため、オラクル社では Net8 Assistant の使用をお薦めしています。

Oracle Net8 Assistant による操作	NAMESCTL 制御ユーティリティによる操作
<ol style="list-style-type: none"> <li>Oracle Net8 Assistant を起動します。 - UNIX の場合、\$ORACLE_HOME/bin から netasst を実行します。 - Windows NT の場合、「スタート」ボタン-&gt;「プログラム」-&gt;「Oracle - HOME_NAME」-&gt;「Network Administration」-&gt;「Net8 Assistant」を選択します。</li> <li>ナビゲータで、Oracle Names サーバーを展開します。</li> <li>Oracle Names サーバーを選択します。</li> <li>右ペインのリストから、「Manage Data」を選択します。</li> <li>「Links」タブをクリックします。</li> <li>「Add」を選択します。</li> <li>「DB Link Name」フィールドにグローバル・データベース・リンクを入力します。 <b>注意：</b>初期化パラメータ・ファイルで GLOBAL_NAMES パラメータが TRUE に設定されている場合、グローバル・データベース名を入力する必要があります。</li> <li>「User and Password」フィールドに、このデータベースの有効なユーザー名とパスワードからなる資格証明を入力します。</li> <li>「Execute」をクリックします。</li> <li>リンク修飾子を使用する場合は、6-64 ページの「<a href="#">リンク修飾子の追加</a>」を参照してください。</li> <li>「File」-&gt;「Save Network Configuration」を選択します。</li> </ol>	<p>グローバル・データベース・リンクを作成し、Oracle Names サーバーに登録します。</p> <pre>NAMESCTL NAMESCTL&gt; register dblink -d (description=(address= ...))</pre> <p>たとえば、デフォルトの TCP/IP リスニング・アドレスを使用して sales.us.acme.com というグローバル・データベース・リンクを登録するには、次のように入力します。</p> <pre>register sales.us.acme.com -d (description=(address=(protocol-tcp) (host=sales-pc) (port=1521)))</pre> <p><b>関連項目：</b>アドレス情報に必要なプロトコル構文については、<a href="#">付録 B</a> を参照してください。</p> <p>NAMESCTL ユーティリティでは、ユーザー名とパスワードの資格証明を設定できません。</p>



作成するグローバル・データベース・リンクの数はデータベースごとに 1 個をお勧めしますが、追加できる **リンク修飾子 (link qualifier)** の数に制限はありません。リンク修飾子はグローバル・データベース・リンクに付加され、データベースのユーザー名とパスワードからなる資格証明の代替設定を提供します。たとえば、EMP というリンク修飾子を sales.us.acme.com に付加できます。グローバル・データベース・リンクを使用してリモート・データベースに接続するには、次のように入力します。

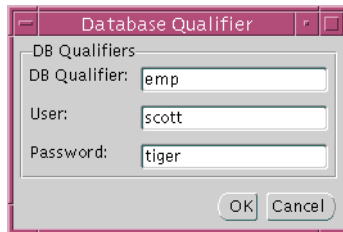
```
SQL> CONNECT @sales.us.acme.com@emp
```

## リンク修飾子の追加

リンク修飾子は、別のスキーマに接続する際に、複数のデータベース・リンクが使用されている主に古い Oracle データベース環境で使用されます。複数のデータベース・リンクを使用しなくても、1 個のデータベース・リンクに対して複数のリンク修飾子を作成できます。複数のリンク修飾子を使用することにより、グローバル・データ・リンクを初期化パラメータ・ファイルの GLOBAL\_NAMES=TRUE 設定に適合させることができます（このパラメータによって、グローバル・データベース・リンクを、グローバル・データベース名に設定できます）。

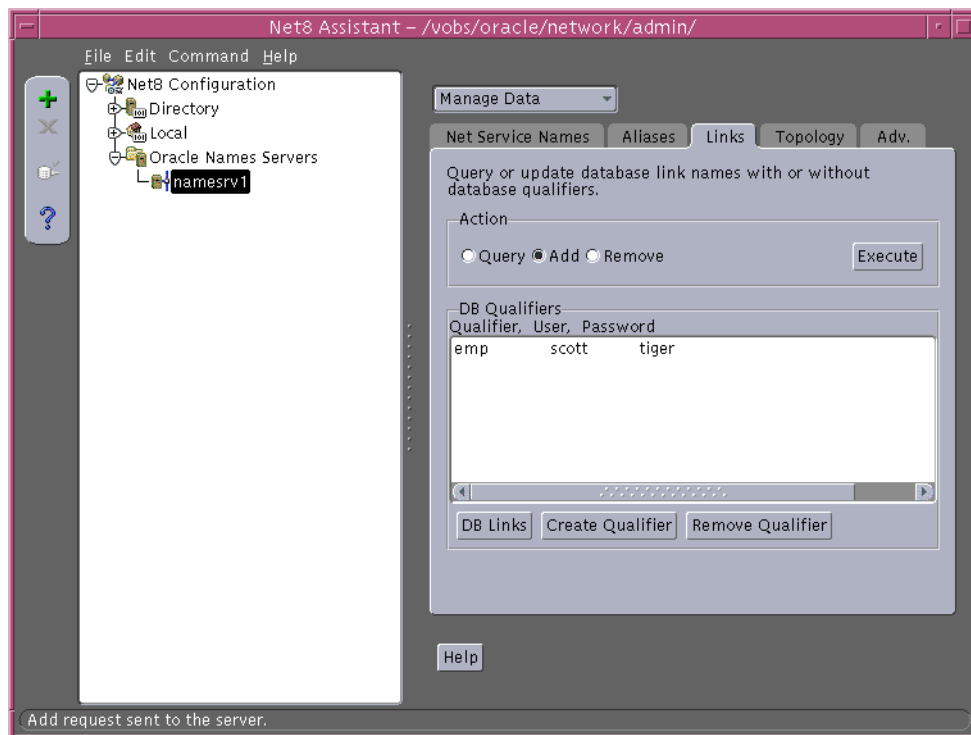
グローバル・データベース・リンクに対してグローバル・リンク修飾子を登録するには、Net8 Assistant を使用します。NAMECTL ユーティリティでは、リンク修飾子を作成できません。

1. 前述の手順 1 ～ 8 を実行します。
2. 「DB Qualifiers」をクリックします。
3. 「Create Qualifier」をクリックします。  
「Database Qualifier」ダイアログ・ボックスが表示されます。



4. リンク修飾子の名前と、有効なユーザー名とパスワードからなる資格証明を入力し、「OK」をクリックします。  
リンク修飾子が「DB Qualifiers」リスト・ボックスに追加されます。





5. 追加するリンク修飾子ごとに手順 3 ～ 4 を繰り返します。
6. 「Execute」をクリックします。
7. 「File」->「Save Network Configuration」を選択します。

グローバル・データベース・リンクが正常に機能しているか検証するには、ユーザーが許可されている表に対して次のような SELECT 文を実行します。

```
sql> SELECT * from table@db_link;
sql> SELECT * from table@db_link@link_qualifier;
```

たとえば、次のように指定します。

```
sql> SELECT * from EMP@sales.us.acme.com;
sql> SELECT * from EMP@sales.us.acme.com@emp;
```

**関連項目：**『Oracle8i 分散システム』



## 別名

別名は、Oracle Names サーバー、ネット・サービス名、グローバル・データベース・リンクなど、Oracle Names サーバーに格納されている任意の項目タイプに対して使用されます。

別名を作成するには、Net8 Assistant を使用して次の作業を実行します。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、`$ORACLE_HOME/bin` から `netasst` を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、Oracle Names サーバーを展開します。
3. Oracle Names サーバーを選択します。
4. 右ペインのリストから、「Manage Data」を選択します。
5. 「Aliases」タブをクリックします。
6. 「Add」を選択します。
7. 「Alias」フィールドに別名を、「Canonical Name」フィールドに実際のオブジェクト名を入力します。
8. 「Execute」をクリックします。
9. 「File」 -> 「Save Network Configuration」を選択します。

## 作業 7: (オプション) 委任管理リージョンへのドメインの委任

委任管理リージョンが構成されている場合、上位レベルの管理リージョンはサブドメインを委任管理リージョンに委任する必要があります。委任しない場合、上位レベルの管理リージョンにある Oracle Names サーバーがすべてのサブドメインに対する権限を持ちます。

いったんドメインが委任されると、カレント・リージョンにある Oracle Names サーバーは、それ以降のすべての処理を、別の Oracle Names サーバーによって管理されるドメインのあるサブリージョンに渡します。

ドメインを委任するには、Oracle Net8 Assistant または NAMESCTL 制御ユーティリティを使用します。



Oracle Net8 Assistant による操作	NAMESCTL 制御ユーティリティによる操作
<p>1. Oracle Net8 Assistant を起動します。</p> <p>- UNIX の場合、\$ORACLE_HOME/bin から netasst を実行します。</p> <p>- Windows NT の場合、「スタート」ボタン -&gt; 「プログラム」 -&gt; 「Oracle - HOME_NAME」 -&gt; 「Network Administration」 -&gt; 「Net8 Assistant」を選択します。</p> <p>2. ナビゲータで、Oracle Names サーバーを展開します。</p> <p>3. 上位レベル・リージョンにある Oracle Names サーバーを選択します。</p> <p>4. 右ペインのリストから、「Manage Data」を選択します。</p> <p>5. 「Topology」タブをクリックします。</p> <p>6. 「Delegate Domain」をクリックします。</p> <p>7. 「Domain Name」フィールドにサブドメインの名前を入力します。</p> <p>8. 「Names Server Name」フィールドに対してドメインにサービスを提供する Oracle Names サーバーと、そのプロトコル・アドレスを入力します。</p> <p><b>関連項目：</b>アドレス情報に必要なプロトコル構文については、<a href="#">付録 B</a> を参照してください。</p> <p>9. 「Execute」をクリックします。</p> <p>10. 「File」 -&gt; 「Save Network Configuration」を選択します。</p>	<p>別名を作成し、Oracle Names サーバーに登録します。</p> <pre>NAMESCTL NAMESCTL&gt; delegate_domain domain_name onames_server [(description=(address_list=) (address= ...) ())]</pre> <p><b>関連項目：</b>アドレス情報に必要なプロトコル構文については、<a href="#">付録 B</a> を参照してください。</p> <p>たとえば、namesrv3.us.acme.com からサービスの提供を受けるドメイン us.acme.com を委任するには、次のように入力します。</p> <pre>delegate_domain us.acme.com namesrv3.us.acme.com (address=(protocol-tcp) (host=namesrv3-pc) (port=1575))</pre>

## 作業 8: データベースへの接続

クライアントは、次の構文を使用してデータベースへ接続できます。

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



## ホスト・ネーミング・メソッドの構成

簡単な接続方法が求められる環境では、ホスト・ネーミングを使用することにより `tnsnames.ora` ファイルでサービス名を検索する必要がなくなります。ただしホスト・ネーミングは、接続プーリング、外部プロシージャ、異種サービスなど、追加の接続情報が必要な、高度の機能が要求される大規模で複雑な環境には不向きです。このような場合は、別のネーミング・メソッドの使用をお勧めします。

ホスト・ネーミング・メソッドは、TCP/IP ネットワーク環境でのみ利用可能です。インスタンスがリスナーに登録されると、データベース名とドメイン名から構成されるサーバーのグローバル・データベース名が登録されます。グローバル・データベース名は、既存の名前解決サービスのホスト名または別名と同じです。クライアントでは、データベースへの接続の際に、接続文字列の中でこの別名が使用されます。

次の場合にクライアントは、別名を使用してサーバーに接続できます。

- インストール済の Net8 Server と Net8 Client ソフトウェアを使用して、Oracle8i データベース・サービスに接続する場合
- クライアントとサーバーが、TCP/IP プロトコルを介して接続する場合
- すべての名前が、**ドメイン名システム (Domain Name System: DNS)** や集中的に管理される TCP/IP `hosts` ファイルのような IP アドレス解決メカニズムを通して解決される場合
- **Oracle Connection Manager** やセキュリティのような高度な機能が不要な場合

ホスト・ネーミング・メソッドを構成するには、次の作業を実行します。

作業 1: グローバル・データベース名を使用したリスナーの構成

作業 2: 第 1 番目のネーミング・メソッドとしての HOSTNAME の構成

作業 3: ホスト名解決の環境の設定

作業 4: データベースへの接続

### 作業 1: グローバル・データベース名を使用したリスナーの構成

グローバル・データベース名は、リスナーに登録する必要があります。登録がどのように行われるかは、データベースのリリースによって異なります。

#### Oracle8i リリース 8.1

データベースがリスナーを検出できる場合は、グローバル・データベース名を含むデータベースに関する情報は自動的にリスナーに登録されます。リスナーは、次の場合に検出されます。

- リスナーが、ポート 1521 の TCP/IP 上で実行されている場合
- LOCAL\_LISTENER パラメータが初期化ファイルに設定されている場合



データベースがリスナーを検出できない場合、次の「[Oracle8i リリース 8.0 またはそれより前のリリース](#)」で説明するように GLOBAL\_DBNAME パラメータを使用して listener.ora ファイルを構成する必要があります。

### Oracle8i リリース 8.0 またはそれより前のリリース

Oracle8i リリース 8.0 データベースまたは Oracle7 データベースの場合、データベース情報は listener.ora ファイルを介してリスナーに登録されます。これらのデータベースでは、次に示すように SID\_LIST\_listener\_name セクションを静的に構成し、そこに GLOBAL\_DBNAME パラメータを含めます。

```
sid_list_listener=(sid_list=
(sid_desc=
(global_dbname=sales.us.acme.com)
(sid_name=db1)
(oracle_home=/u01/app/oracle/8.0.5))
```

**関連項目：**7-10 ページの「[静的サービス情報の構成](#)」

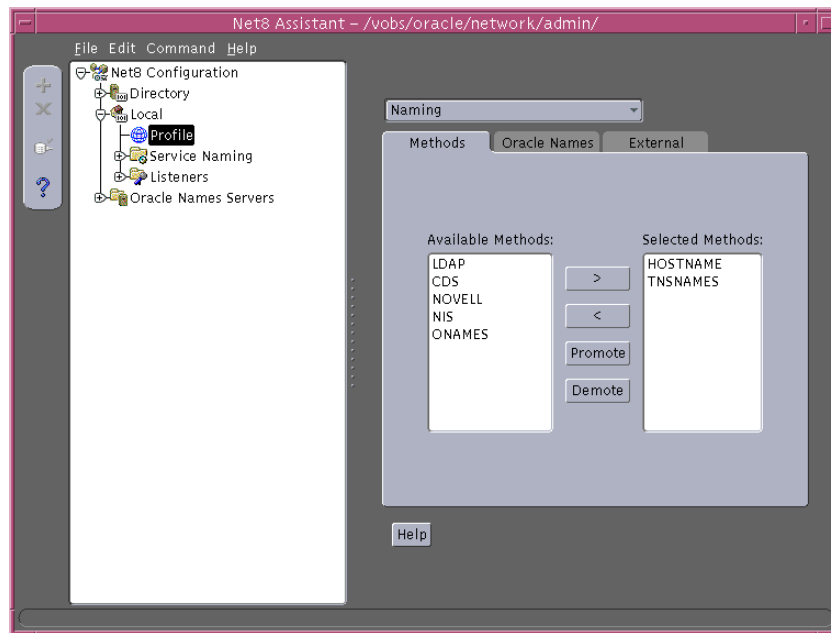
## 作業 2: 第 1 番目のネーミング・メソッドとしての HOSTNAME の構成

sqlnet.ora ファイルの NAMES.DIRECTORY\_PATH パラメータに指定する第 1 番目のネーミング・メソッドとしてホスト・ネーミングを構成します。このパラメータによって、Net8 が接続記述子の名前解決に使用するネーミング・メソッドの順序が指定されます。

ホスト・ネーミングを第 1 番目のネーミング・メソッドに指定するには、次の手順に従います。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Local」 -> 「Profile」を展開します。
3. 右ペインのリストから、「Naming」を選択します。
4. 「Methods」タブをクリックします。
5. 「Available Methods」リストから「HOSTNAME」を選択して、「>」ボタンをクリックします。
6. 「Selected Methods」リストで「HOSTNAME」を選択し、「Promote」ボタンを使用して選択した項目をリストの最上位に移動します。





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

sqlnet.ora ファイルは、先頭に HOSTNAME がリストされる NAMES.DIRECTORY\_PATH パラメータにより更新されます。

```
names.directory_path=(hostname, tnsnames)
```

### 作業 3: ホスト名解決の環境の設定

グローバル・データベース名は、DNS、NIS または集中管理される TCP/IP ホスト・ファイルである /etc/hosts のような IP アドレス解決メカニズムを通して解決する必要があります。

たとえば、あるデータベースの sales.us.acme.com というグローバル・データベース名が sales-server という名前のマシン上に存在する場合、/etc/hosts ファイルのエントリは次のようになります。

#IP address of server	host name	alias
144.25.219.146	sales-server	sales.us.acme.com

グローバル・データベース名のドメイン部分は、ネットワーク・ドメインと一致する必要があります。



## 作業 4: データベースへの接続

クライアントは、別名を使用してデータベースへ接続できます。「作業 3: ホスト名解決の環境の設定」の例を使用して、クライアントでは、次のように接続文字列で sales.us.acme.com を使用できます。

```
CONNECT username/password@sales.us.acme.com
```

クライアントとサーバーが同じ us.acme.com ドメインに所属していれば、クライアントは接続文字列に sales を入力するだけでかまいません。

## 外部ネーミング・メソッドの構成

外部ネーミングとは、Oracle 以外のネーミング・サービスに格納されるネット・サービス名をネットワーク・アドレスに解決する方法です。外部ネーミング・サービスには次のものが含まれます。

- **Network Information Service (NIS)**
- **Novell ディレクトリ・サービス (Novell Directory Services: NDS)**
- **セル・ディレクトリ・サービス (Cell Directory Services: CDS)**

**関連項目：**CDS の構成については、『Oracle8i Advanced Security 管理者ガイド』を参照してください。

## NIS の構成

システム・インフラの一部としてすでにネットワーク情報サービス (NIS) を使用している組織や会社のために、NIS 外部ネーミングを使用して NIS 内に Oracle ネット・サービス名とアドレスを格納するオプションが用意されています。

ユーザーが次のようなコマンドを指定した場合、

```
sqlplus scott/tiger@payroll
```

(payroll は Oracle サービス名) クライアント・プログラム (またはクライアント・プログラムとしての役割を果たすサーバー) を実行するノード上の NIS 外部ネーミングは、ネットワーク内にある NIS サーバーに接続し、ネット・サービス名を NIS サーバーに渡します。NIS サーバーはネット・サービス名を Net8 アドレスに変換し、このアドレスをクライアント・プログラム (またはクライアント・プログラムとしての役割を果たすサーバー) に返します。次に、クライアント・プログラムはこのアドレスを使用して Oracle データベースに接続します。

NIS サーバーとして動作するマシンは、名前要求を処理する ypserv と呼ばれるプログラムを実行します。ypserv では、**マップ (map)** と呼ばれる特殊なファイルに異なるタイプのデータが格納されます。たとえば、パスワードは passwd.byname と呼ばれるマップに格納



されます。Oracle データベース・サービス名は、tnsnames と呼ばれるマップに格納されます。

前の項で紹介したようなコマンドをユーザーが発行すると、NIS 外部ネーミングは RPC コールを使用して ypserv に接続し、Oracle ネット・サービス名の payroll とマップの名前である tnsnames を渡します。ypserv プログラムは、名前 payroll とそれに対応する値（ネット・サービス名のアドレス）を tnsnames マップ内で検索します。このアドレスはクライアントに返され、クライアント・プログラム（またはクライアント・プログラムとしての役割を果たすサーバー）は、このアドレスを使用してデータベース・サーバーに接続します。

### システム要件

NIS 外部ネーミングには、SQL\*Net 2.2 以上が必要です。

### 作業 1: NIS サーバーを構成しての NIS 外部ネーミングのサポート

NIS 外部ネーミングをサポートするようにサーバーを構成する前に、Oracle データベース・ネット・サービス名を変換する必要がある NIS サーバー上で NIS が構築され、動作していることを確認してください。具体的には、NIS のマニュアルを参照してください。

**NIS マップの既存セットへの tnsnames マップの追加** NIS マップの既存セットに tnsnames マップを追加する手順は、次のとおりです。

1. 6-4 ページの「ローカル・ネーミング・メソッドの構成」の説明に従って、tnsnames.ora ファイルを作成します。

---

---

**注意：**tnsnames.ora ファイルのコピーは、なるべく \$ORACLE\_HOME/network/admin ディレクトリに保管してください。このファイルを後でもう一度使用して、ネット・サービス名を NIS マップにロードする必要があります。

---

---

2. tns2nis プログラムを使用して、tnsnames.ora ファイルの内容を tnsnames に変換します。

---

---

**注意：**tns2nis プログラムは、Oracle Universal Installer のメディア上の NIS 外部ネーミングによって提供されます。

---

---

たとえば、引数を 1 つ付けてコマンドラインで tns2nis を実行します。

```
tns2nis tnsnames.ora
```

tns2nis は、カレント・ディレクトリから tnsnames.ora ファイルを読み込みます（tnsnames.ora ファイルがカレント・ディレクトリにない場合は、フル・パス名を使



用してその位置を指定します。例：/etc/native.ora または \$ORACLE\_HOME/network/admin/tnsnames.ora)。

これにより tnsnames マップが、カレント作業ディレクトリに書き込まれます。

3. tnsnames が NIS サーバーにない場合は、tnsnames を NIS サーバーにコピーします。
4. NIS プログラムの makedbm を使用して tnsnames マップをインストールします。詳細は、NIS のマニュアルを参照してください。

---

**注意：**この手順は、NIS の管理担当者が行ってください。

---

makedbm は、tnsnames マップを NIS サーバーが読取り可能な 2 つのファイルにマップします。これらのファイルの位置は、プラットフォームによって異なります。詳細は、ご使用のプラットフォームに対応するマニュアルを参照してください。

たとえば、Sun Solaris 上で tnsnames マップを生成し、インストールするには、root としてコマンドラインから次のように入力します。

```
# makedbm tnsnames /var/yp/'domainname'/tnsnames
```

**tnsnames マップが適切にインストールされたことの確認** 次の形式でコマンドを入力することにより、NIS サーバーをテストしてマップが適切にインストールされていることを確認できます。

```
ypmatch net_service_name tnsnames
```

たとえば、次のように入力できます。

```
ypmatch payroll.com tnsnames
```

例として、次のようにアドレスの長さ（文字数）のついたアドレスが戻されます。

```
99 (description=(address=(protocol=tcp)
(host=garlic) (port=1999)))
(connect_data=(service_name=dirprod)))
```

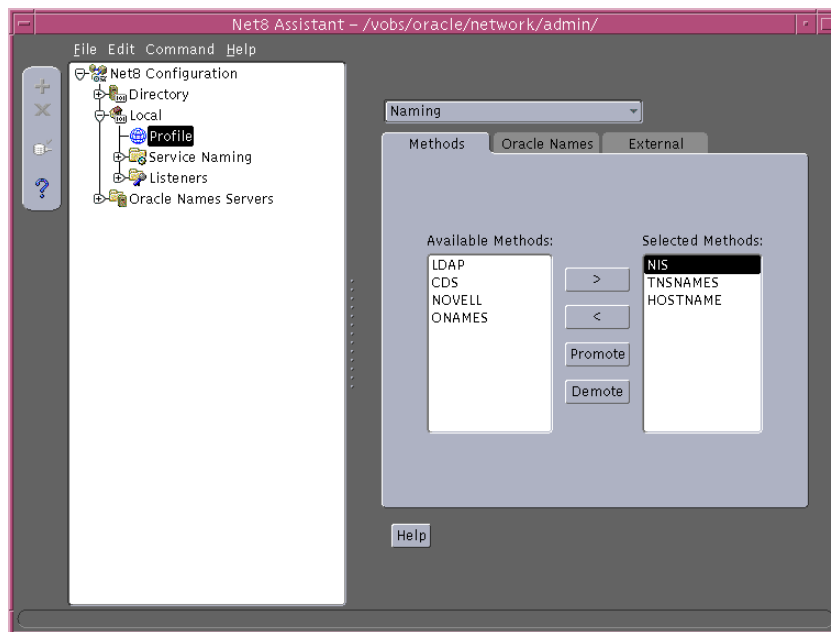
## 作業 2: クライアントの構成

クライアントを構成するには、sqlnet.ora ファイルの NAMES.DIRECTORY\_PATH パラメータに指定する第 1 番目のネーミング・メソッドとして NIS を構成します。このパラメータによって、Net8 が接続記述子の名前解決に使用するネーミング・メソッドの順序が指定されます。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。



- Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」 を選択します。
- 2. ナビゲータで、「Local」 -> 「Profile」を展開します。
- 3. 右ペインのリストから、「Naming」を選択します。
- 4. 「Methods」タブをクリックします。
- 5. 「Available Methods」リストから「NIS」を選択して、「>」ボタンをクリックします。
- 6. 「Selected Methods」リストで「NIS」を選択し、「Promote」ボタンを使用して選択した項目をリストの最上位に移動します。



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

sqlnet.ora ファイルは、先頭に NIS がリストされる NAMES.DIRECTORY\_PATH パラメータにより更新されます。

```
names.directory_path=(nis, hostname, tnsnames)
```



## NDS の構成

NDS 外部ネーミングは、NDS 固有のネーミング表記法を使用して、Novell NDS を利用できるネットワーク上で Oracle データベースに接続できるようにします。NDS 外部ネーミングをクライアントとサーバー上にインストールすると、次のように入力できます。

```
username/password@.OracleInstance.Context
```

NDS 外部ネーミングを使用するネットワーク・ユーザーには、次の利点があります。

- クライアントがデータベースに接続するときに簡単な NDS 名（部分名または完全名）を使用できます。
- ネットワーク・アドレスのメンテナンスを簡素化します。すなわち、1 つの変更が NDS 外部ネーミングを使用するすべてのクライアントに作用します。
- リスナーが通知する必要がないので、ネットワーク通信量が軽減されます。

### クライアントでの動作

NDS 外部ネーミングはクライアント・ワークステーション上に存在し、NDS オブジェクト名をネットワーク・アドレスに変換します。クライアント・コードは、名前が Oracle データベースと一致する NDS オブジェクトについて NDS ツリーから属性を取得します。この名前は、完全名または部分名のいずれでもかまいません。その名前が部分名であれば、その部分名はカレント・ネーミング・コンテキストに基づいて修飾されます。

### NetWare サーバーでの動作

NDS 外部ネーミングのサーバー側には、次の 3 つの局面があります。

- [スキーマの拡張](#)
- [SAP \(Service Advertising Protocol\) の無効化](#)
- [データベース・サービスと NDS へのアドレスの格納](#)

**スキーマの拡張** NetWare 4 および 5 サーバーでの Oracle インストールの最中に、ORACLE:DBInstance と呼ばれるオブジェクト・クラスを含むように NDS スキーマが拡張されます。NDS 外部ネーミングを機能させるには、このクラスに ORACLE:TNSAddress と呼ばれる属性が必要です。クラスが存在しない場合は、そのクラスが作成され、TNSAddress 属性が挿入されます。クラスは存在するが、TNSAddress 属性が存在しない場合は、そのクラスが変更されます。インストール中に、これを実行する NetWare ロードダブル・モジュール (NLM) を ORASCHEM.NLM といいます。

**SAP (Service Advertising Protocol) の無効化** NetWare の SPX プロトコルでは、config.ora ファイル内で ORACLE\_SAP と呼ばれる値が検索されます。このパラメータの値は ON または OFF です。このパラメータを指定しない場合、デフォルトの設定は ON です。これは、SPX ネットワークに関するパフォーマンスに密接な関係があります。ORACLE\_SAP が ON であれば、リスナーは SAP (サービス・アドバタイジング・プロトコル) を使用して、その



アドレスを通知します。ORACLE\_SAP を OFF に設定すると、リスナーは NDS を使用して情報をそのクライアントに送ります。

ネットワークが NDS 使用可能クライアント（すなわち、NetWare 4 以上を使用するクライアント）のみで構成されている場合は、config.ora ファイルを編集して ORACLE\_SAP を OFF に設定すると、ネットワークのパフォーマンスが向上します。

**データベース・サービスと NDS へのアドレスの格納** リスナーが起動されると、リスナーにより、データベース・アドレスが OracleInstance オブジェクトの下の NDS へ格納されます。

---

**注意：**Oracle データベースの NDS オブジェクトをあらかじめ作成しておく必要があります。

---

その時点で、クライアントは NDS データベースからアドレスをアクセスできます。

### システム要件

NDS 外部ネーミングには、SQL\*Net 2.2 以上と Oracle 7.2 以上が必要です。このアダプタは、Novell ライブラリを実行するクライアントと一緒に使用できます。ただし、サーバーでは NetWare 4.1 以上が必要です。

### クライアントの構成

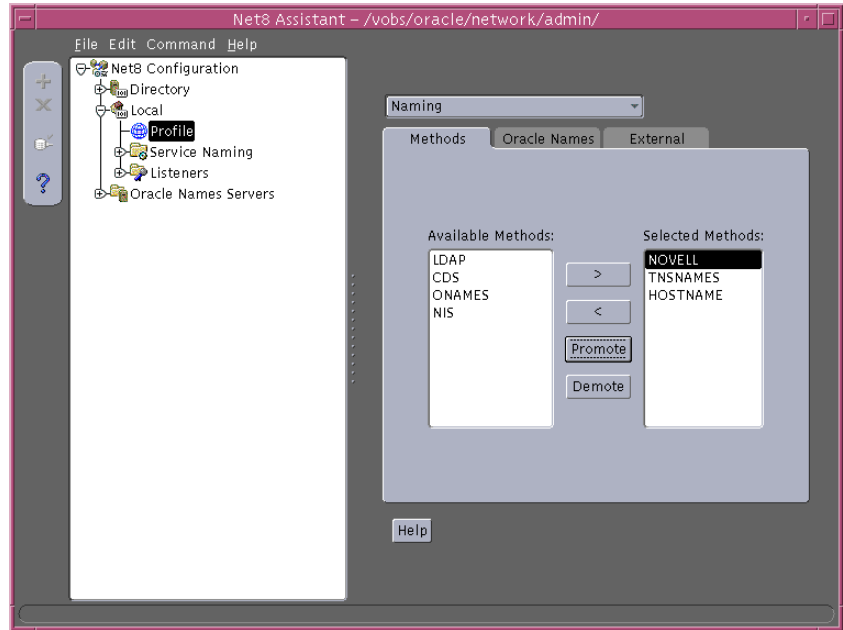
クライアントは、NDS 外部ネーミング・メソッドと NDS ネーミング・コンテキストによって構成される必要があります。

クライアントを構成するには、次の手順に従います。

1. sqlnet.ora ファイルの NAMES.DIRECTORY\_PATH パラメータに指定する第 1 番目のネーミング・メソッドとして NDS を構成します。このパラメータによって、Net8 が名前解決に使用するネーミング・メソッドの順序を指定します。
  - a. Oracle Net8 Assistant を起動します。
    - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
    - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
  - b. ナビゲータで、「Local」 -> 「Profile」を展開します。
  - c. 右ペインのリストから、「Naming」を選択します。
  - d. 「Methods」タブをクリックします。
  - e. 「Available Methods」リストから「Novell」を選択し、「>」ボタンをクリックします。



- f. 「Selected Methods」リストで「Novell」を選択し、「Promote」ボタンを使用して選択した項目をリストの最上位に移動します。



2. sqlnet.ora ファイルの NAMES.NDS.NAME\_CONTEXT パラメータを使用して、タイプなし名またはタイプ付き名を構成します。このパラメータでは、データベース・オブジェクトが存在する NDS ツリー内のネーミング・コンテキストが指定されます。ネーミング・コンテキストを指定するには、次の作業を実行します。
  - a. 「External」タブをクリックします。
  - b. 「Name Context」フィールドにネーミング・コンテキストを入力します。

データベース・オブジェクトの名前が "Payroll.Finance.Oracle" であり、ネーミング・コンテキストが Finance.Oracle の場合、接続文字列では "Payroll" のみを指定する必要があります。"Payroll" が ".Payroll.Finance.Oracle" に修飾されます。これはタイプなし名の例です。

---

**注意：** 先行するドットは、この名前が完全な NDS の名前であることを示します。名前のコンテキスト設定を上書きする場合は、先行するドットを使用して接続文字列内に完全な NDS 名を指定できます。

---



ネーミング・コンテキストが `ou=Finance.o=Oracle` である場合、接続文字列には、`cn=Payroll.ou=Finance.o=Oracle` が含まれる必要があります。これはタイプ付き名の例です。

`sqlnet.ora` ファイルの名前コンテキストによって、`client32net.cfg` ファイルの "name context" エントリを上書きできます。いずれのファイルでも名前コンテキストを指定しないと、その名前は `[root]` にデフォルト設定されます。

`client32net.cfg` ファイル・パラメータの詳細は、Novell クライアントのマニュアルを参照してください。

---

**注意：**名前コンテキストには、先行するドットは含まれません。これが含まれると、NDS エラー・コード -309 (ERR\_EXPECTED\_IDENTIFIER) が発生する場合があります。  
The parameter being parsed is not typed.

---

3. 「File」->「Save Network Configuration」を選択します。

`sqlnet.ora` ファイルでは、次のエントリが更新されます。

```
names.directory_path=(novell, tnsnames, hostname)
names.nds.name_context=Finance.Oracle
```

## サーバーの構成

NetWare サーバーを構成するには、Novell マニュアルの説明に従って Net8 Easy Config アプリケーションを使用します。

## 制約

次に、NDS 外部ネーミングを使用する場合のいくつかの制約を示します。

- NDS データベースに格納されているデータベース・アドレスの長さは、2048 文字を超えることはできません。
- 1 つのデータベース・インスタンスに複数のリスナーを使用することはできません。使用すると、最後に起動するリスナーによってデータベース・オブジェクトに格納されているその他の TNS アドレスが上書きされます。
- SID サポートがサーバー上で使用可能な場合は、データベース・インスタンスに NULL SID を使用できません。インスタンスのうち 1 つでも NULL SID を使用すると、SQL\*Net バージョン 2 または Net8 を使用して接続することができません。
- SID サポートが使用可能でない場合は、リスナーの `SID_LIST` で指定した最後の SID が、使用する SID となります。この場合、SID はユーザーとデータベースに対して透過的です。ユーザーには SID が見えず、データベースは SID を無視します。



---

**注意：**SID サポートは、`config.ora` ファイル内の次のパラメータによって制御されます。

`nw_enable_sid_support=[true|false]`

---







---

## リスナーの構成

この章では、リスナーの構成方法について説明します。内容は次のとおりです。

- 構成の概要
- `listener.ora` ファイル変更後のリスナーの再起動
- リスナー・プロトコル・アドレスの構成
- デフォルト以外のアドレスを使用するリスナーの構成
- 多数の接続要求の処理
- 静的サービス情報の構成
- 事前生成済専用サーバーの構成



## 構成の概要

リスナーは、サーバーに常駐する個別のプロセスです。これは、クライアントの着信接続要求を受け取り、これらの要求のサーバーへの着信を管理します。

リスナーは、1つまたは複数のリスニング・プロトコル・アドレスおよび宛先サービスに関するサービス情報で構成されます。

プロトコル・アドレスは、リスナー構成ファイル `listener.ora` で構成されます。サービス情報は、`listener.ora` ファイルで構成する必要がある場合とない場合があります。

- Oracle リリース 8.1 インスタンスでは、サービス名、インスタンス名、サービス・ハンドラおよびロード情報などがリスナーに自動的に登録されます。**サービス登録 (service registration)** と呼ばれるこの機能では、`listener.ora` ファイルでの構成が必要ありません。

---

**注意：** Oracle8i データベースを管理する際に Oracle Enterprise Manager を使用する場合、`listener.ora` ファイルでのサービス構成が必要です。Oracle Enterprise Manager では、データベースの検出およびリモート管理の実行の際に、`listener.ora` ファイルのサービス情報が使用されます。

---

- Oracle7 または Oracle8 リリース 8.0 データベースおよび Oracle Enterprise Manager 管理ツールを含む他のサービスでは、`listener.ora` ファイルでのサービス構成が必要です。

いったんリスナーを構成すると、リスナー制御ユーティリティ (LSNRCTL) を使用してリスナーを管理できます。

---

**注意：** Oracle8i リリース 8.1.6 データベースでは、リリース 8.1.6 リスナーが必要です。Oracle8i リリース 8.1.6 データベースでは、前のバージョンのリスナーはサポートされていません。ただし、Oracle8i データベースの前のバージョンでリリース 8.1.6 リスナーを使用することはできます。

---

この項では、次の追加構成のトピックについて説明します。

- [デフォルト・リスニング・プロトコル・アドレス](#)
- [インストール中の構成作業](#)
- [Oracle Net8 Assistant による構成](#)

## デフォルト・リスニング・プロトコル・アドレス

リスナーには、LISTENER というデフォルト名があり、次のデフォルト・プロトコル・アドレスをリスニングするよう構成されています。



- TCP/IP プロトコルのポート 1521

```
(address=(protocol=tcp) (host=host_name) (port=1521))
```

- IPC プロトコル

```
(address=(protocol=ipc) (key=PNPKEY))
```

これらのアドレスで構成されたクライアントは、リスナーに接続できます。

## インストール中の構成作業

**Net8 Configuration Assistant** は、ソフトウェアのインストール後、Oracle ユニバーサル・インストーラによって、起動されます。これを使用すると、データベースのプロトコル・アドレスおよびサービス情報を構成できます。

サーバーの「Typical」または「Minimal」インストール後、Net8 Configuration Assistant により、TCP/IP プロトコルを使用するリスナーが LISTENER という名前で自動的に構成されます。「Custom」インストールでは、Net8 Configuration Assistant により、リスナー名およびプロトコル・アドレスを構成するよう求められます。

また、IPC プロトコルを使用する**外部プロシージャ (external procedures)** (PL/SQL コードにより呼び出されるファンクション) のリスニング・アドレスは、インストール・タイプにかかわらず構成されます。

Net8 Configuration Assistant では、Oracle8i データベースおよび外部プロシージャのサービス情報も listener.ora ファイルに構成されます。

---

---

**注意：** サービス情報は Oracle リリース 8.1 の listener.ora ファイルで不要ですが、Net8 Configuration Assistant では、データベースを監視する際に Oracle Enterprise Manager を使用することを仮定します。Oracle Enterprise Manager では、データベース検出の際にこの情報が必要です。

---

---

次に示すのは listener.ora ファイルのサンプルです。LISTENER エントリでは、LISTENER という名前のリスナーに対するリスニング・プロトコル・アドレスが定義され、SID\_LIST\_LISTENER エントリでは、グローバル・データベース名、データベースの Oracle ホームの場所、インスタンスの Oracle システム識別子を含むデータベース・サービスに関する情報が提供されます。

```
listener=
  (description=
    (address_list=
      (address=(protocol=tcp) (host=sale-server) (port=1521))
      (address=(protocol=ipc) (key=extproc))
    )
  )
sid_list_listener=
  (sid_list=
```



```
(sid_desc=
  (global_dbname=sales.us.acme.com)
  (oracle_home=/oracle81)
  (sid_name=sales)))
(sid_desc=
  (sid_name=plsextproc)
  (oracle_home=/oracle81)
  (program=extproc)))
```

## Oracle Net8 Assistant による構成

デフォルトまたはインストールされた構成で十分ではない、またはネットワークにリスナーもしくはデータベースの旧バージョンがある場合、**Net8 Configuration Assistant** を使用して listener.ora 構成を拡張できます。

この項では、次のリスナー構成ファイル項目を取り上げます。

- [listener.ora ファイル変更後のリスナーの再起動](#)
- [リスナー・プロトコル・アドレスの構成](#)
- [多数の接続要求の処理](#)
- [静的サービス情報の構成](#)

### 関連項目：

- 外部プロシージャの構成に関する詳細は、8-39 ページの「[外部プロシージャを使用する場合の Net8 の構成](#)」を参照してください。
- Java オプションと IIOP クライアントの構成に関する詳細は、8-39 ページの「[非 Oracle データベース・サービスへの接続の構成](#)」を参照してください。

## listener.ora ファイル変更後のリスナーの再起動

listener.ora が変更された場合は、常に LSNRCTL ユーティリティを使用してリスナーを一度停止し、再起動する必要があります。

リスナーを起動または停止するには、次のように実行します。

```
LSNRCTL
LSNRCTL> start|stop [listener_name]
```

*listener\_name* は、listener.ora ファイルに定義されたリスナーの名前です。LISTENER という名前のデフォルトのリスナーを使用している場合は、リスナーを指定する必要はありません。



**関連項目：**

- リスナーの起動については、11-2 ページの「[Net8 制御ユーティリティの使用](#)」を参照してください。
- 全 LSNRCTL コマンドの完全なリストについては、A-2 ページの「[リスナー制御ユーティリティ \(LSNRCTL\)](#)」を参照してください。

## リスナー・プロトコル・アドレスの構成

デフォルト以外のプロトコル・リスナー・アドレスを構成するには、次の手順に従います。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン->「プログラム」->「Oracle - HOME\_NAME」->「Network Administration」->「Net8 Assistant」を選択します。
2. ナビゲータで、「Local」->「Listeners」を展開します。
3. リスナーを選択します。

---

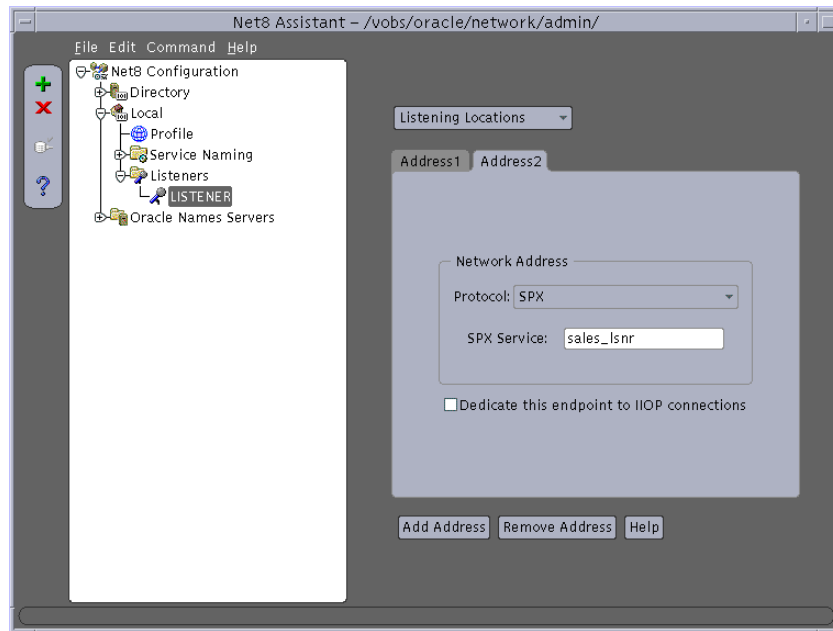
**注意：**listener.ora ファイルは構成済であるが、LSNRCTL 制御ユーティリティを使用してリスナーを一度も起動したことがない場合、リスナーは「Listeners」フォルダで表示されません。その場合は Oracle Net8 Assistant を終了し、11-2 ページの「[Net8 制御ユーティリティの使用](#)」の説明に従ってリスナーを起動した後、再度 Oracle Net8 Assistant を起動してください。

---

リスナーを一度も作成していない場合には、次の作業を実行します。

- a. ツールバーで「+」をクリックするか、「編集」メニューから「作成」を選択します。
  - b. 「Choose Listener Name」ダイアログ・ボックスに一意のリスナー名を入力します。
4. 右ペインのリストから「Listening Locations」を選択します。
  5. 「Add Address」をクリックします。新規のアドレス・タブが表示されます。





6. プロトコルを選択して、そのプロトコルの適切なプロトコル・パラメータ情報を入力します。

**関連項目：**プロトコル・アドレスおよびパラメータについては、[付録 B](#) を参照してください。

TCP/IP 上でリスニングするためにリスナーを構成する場合は、デフォルトのポート 1521 を選択する必要があります。このポートを選択しない場合は、[初期化パラメータ・ファイル \(initialization parameter file\)](#) に LOCAL\_LISTENER パラメータを構成し、なんらかのネーミング・メソッドによってリスナー名を解決する必要があります。

**関連項目：**7-8 ページの「[デフォルト以外のアドレスを使用するリスナーの構成](#)」を参照してください。

使用しているマシンに複数の IP アドレスがあり、リスナーに使用可能なすべての IP アドレスをリスニングさせる場合、TCP/IP または SSL 付き TCP/IP を構成し、「Host」フィールドにマシンのホスト名を入力します。

7. 別のプロトコルを構成するには、ステップ 5 と 6 を繰り返します。
8. 「File」->「Save Network Configuration」を選択します。



listener.ora ファイルは、次のように、TCP/IP および SPX アドレスで静的に構成されます。

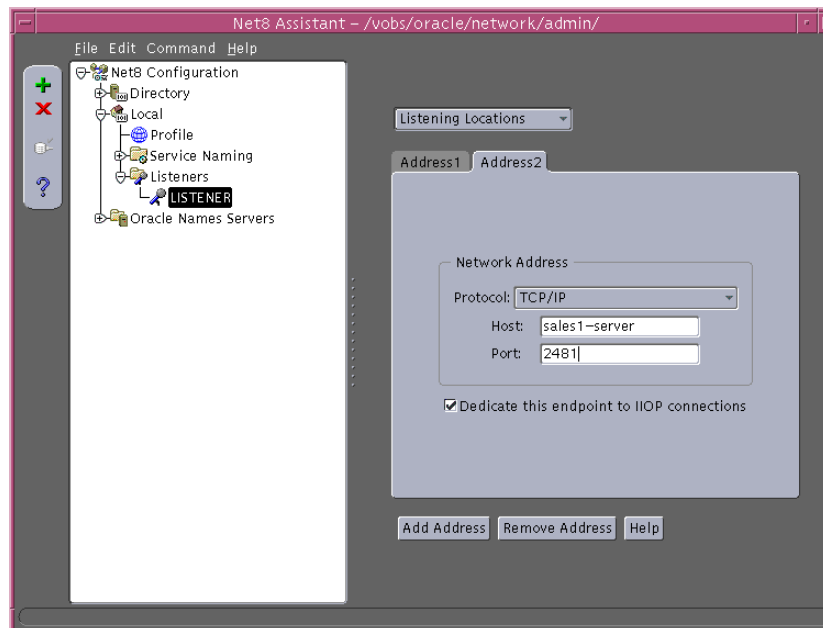
```
listener=
  (address=(protocol= tcp) (host= sales-server) (port= 1521))
  (address=(protocol= spx) (service=orasrvcl))
```

## Oracle8i JServer のためのリスニング・アドレスの構成

データベースの **Oracle8i JServer** の接続では、TCP/IP または SSL 付き TCP/IP プロトコル・アドレスが必要です。

Oracle8i JServer のリスニング・アドレスを構成するには、次の作業が必要です。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Local」 -> 「Listeners」を展開します。
3. リスナーを選択します。
4. 右ペインのリストから「Listening Locations」を選択します。
5. 「Add Address」をクリックします。新規のアドレス・タブが表示されます。





6. 「Protocol」 リストから「TCP/IP」または「TCP/IP with SSL」を選択します。
7. 「Host」 フィールドにデータベースのホスト名を入力します。
8. プロトコルに TCP/IP を選択した場合、「Port」 フィールドに 2481 と入力します。SSL 付き TCP/IP を選択した場合、「Port」 フィールドに 2482 と入力します。
9. 「Dedicate this endpoint to IIOP connections」 をクリックします。
10. 「File」 -> 「Save Network Configuration」 を選択します。

sqlnet.ora ファイルでは、次のエントリが更新されます。

```
listener=
(description_list=
(description=
(address=(protocol=tcp) (host=sales1-server) (port=2481))
(protocol_stack=
(presentation=giop)
(session=raw))))
```

### 関連項目：

- EJB および CORBA アプリケーションへのクライアント接続の構成については、『Oracle8i Enterprise JavaBeans と CORBA 開発者ガイド』を参照してください。
- Java ストアド・プロシージャへのクライアント接続の構成については、『Oracle8i Java ストアド・プロシージャ開発者ガイド』を参照してください。

## デフォルト以外のアドレスを使用するリスナーの構成

デフォルトでは、**PMON プロセス (PMON process)** により、ポート 1521 の TCP/IP デフォルト・ローカル・アドレス上のローカル・リスナーに登録されます。別のリスナーに PMON によって情報を登録する場合、初期化パラメータ・ファイルに LOCAL\_LISTENER パラメータを構成する必要があります。その後、LOCAL\_LISTENER の値は、tnsnames.ora ファイルまたは Oracle Names サーバーのネット・サービス名エントリで解決されます。

- **LOCAL\_LISTENER** パラメータの構成
- **ネーミング・メソッド**の構成

### LOCAL\_LISTENER パラメータの構成

LOCAL\_LISTENER パラメータは、次のように初期化パラメータ・ファイルで構成します。

```
local_listener=listener_name_alias
```



## ネーミング・メソッドの構成

LOCAL\_LISTENER パラメータに指定されたリスナー名の別名は、次のネーミング・メソッドの 1 つを介して解決される必要があります。

- [tnsnames.ora ファイル](#)
- [Oracle Names サーバー](#)

### tnsnames.ora ファイル

ネット・サービス名のエントリは、リスナー・アドレスに対して接続記述子の CONNECT\_DATA なしで作成される必要があります。Oracle Net8 Assistant では、CONNECT\_DATA 情報を使用しない場合は tnsnames.ora ファイルを構成できません。そのため、オラクル社では、tnsnames.ora ファイルを手動で変更することをお勧めします。

たとえば、LOCAL\_LISTENER が LISTENER1 に設定され、LISTENER1 で TCP/IP が使用される場合、tnsnames.ora ファイルのエントリは次のように表示されます。

```
listener1=
  (address=(protocol= tcp)(host= sales-server) (port= 1421))
```

---

**注意：**複数のアドレスがサポートされていますが、接続時フェイルオーバー機能とクライアント・ロード・バランス機能はサポートされていません。Oracle Connection Manager の SOURCE\_ROUTE パラメータはサポートされています。

---

**関連項目：**複数のアドレス構成については、[第 8 章](#)を参照してください。

### Oracle Names サーバー

リスナーの別名に対するエントリを作成するには、Oracle Net8 Assistant を介して Oracle Names サーバーを使用します。

1. Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、Oracle Names サーバーを展開します。
3. Oracle Names サーバーを選択します。
4. 右ペインのリストから、「Manage Data」を選択します。
5. 「Advanced」タブをクリックします。



6. 「Add」をクリックし、「Name」フィールドにリスナーの別名、「Type」フィールドに A.SMD レコード・タイプ、「Value」フィールドに次のようなアドレスを入力します。

```
(address=(protocol=tcp)(host=sales-server)(port=2481))
```

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

## 多数の接続要求の処理

多数の接続要求を処理することが予想される場合は、プロセスに対してキューを指定できます。これにより、リスナーは多数の同時接続要求を動的に処理できます。

リスナーのキューサイズを指定するには、リスニング・アドレスの終端にある QUEUESIZE パラメータに値を入力します。

```
listener=
(address=
(protocol= tcp)
(host= sales-server)
(port= 1521)
(queueSize=20))
```

---

**注意：**現在は、TCP/IP および DECnet 上で動作するリスナーのキューサイズしか設定できません。デフォルトのキューサイズは、使用するシステムによって異なります。Solaris の場合、デフォルトのキューサイズは 5 に設定されています。また、Windows NT 4.0 Workstation のキューサイズは 5 で、Windows NT 4.0 Server のキューサイズは 50 です。

---

## 静的サービス情報の構成

クライアントからの Oracle8 リリース 8.0 や Oracle7 データベースへの接続要求をリスナーが受け入れるには、データベースに関する情報を使用して listener.ora ファイルを構成する必要があります。また、**外部プロシージャ (external procedures)** や **異種サービス (Heterogeneous Services)** などのサービス、および Oracle Enterprise Manager などの一部の管理ツールでは、静的な構成も必要です。

Oracle8 リリース 8.1 では、リスナーによりデータベースおよびインスタンスに関する動的情報が使用されます。この情報は、静的に構成された情報を使用する前にサービス登録を介して受信されます。

listener.ora ファイルは、次のテーブルで説明されるサービス・パラメータで構成されます。



listener.ora ファイルの パラメータ	Net8 Assistant の フィールド	説明
SID_NAME	「SID」	<b>Oracle システム識別子 (Oracle System Identifier: SID)</b> はインスタンスを識別します。SID 値は、初期化パラメータ・ファイルの INSTANCE_NAME パラメータから取得できます。  <b>注意:</b> UNIX および Windows NT では、この設定が必要です。
GLOBAL_DBNAME	「Global Database Name」	<b>グローバル・データベース名 (global database name)</b> は、データベース名およびデータベース・ドメイン名から構成されます。GLOBAL_DBNAME 値は、初期化パラメータ・ファイルの SERVICE_NAMES パラメータまたは DB_NAME および DB_DOMAIN パラメータから取得できます。
ORACLE_HOME	「Oracle Home Directory」	「Oracle Home Directory」フィールドは、指定しているデータベースの Oracle ホームの場所を識別します。  <b>注意:</b> UNIX では、この設定が必要です。

**重要:** Oracle Parallel Server 環境のように、**接続時フェイルオーバー**または**透過的アプリケーション・フェイルオーバー (Transparent Application Failover:TAF)**を使用している場合には、GLOBAL\_DBNAME パラメータを設定しないでください。静的に構成されたグローバル・データベース名により、この機能は使用禁止とされます。

**関連項目:**

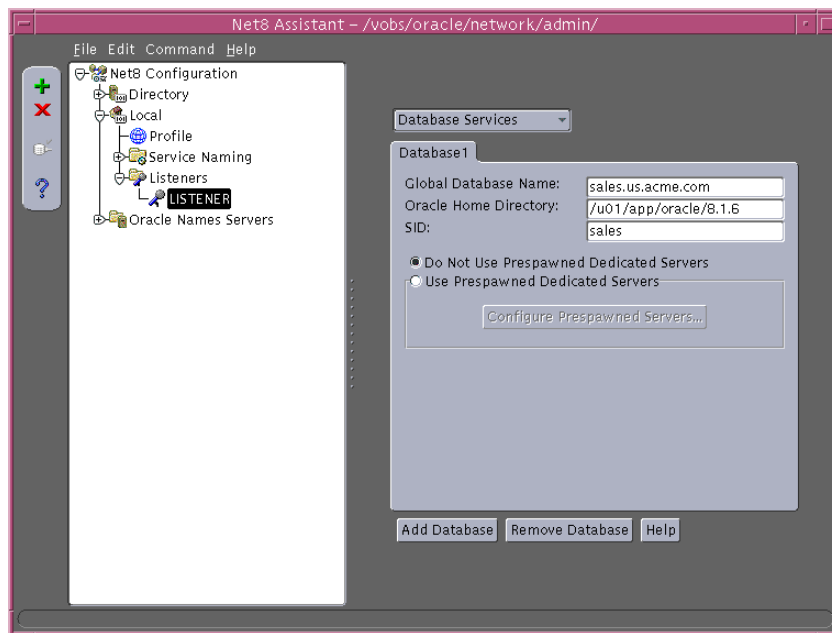
- 外部プロシージャおよび異種サービスのリスナーを静的に構成する方法については、[第 8 章](#)を参照してください。
- Oracle Enterprise Manager については、『Oracle Enterprise Manager 構成ガイド』を参照してください。

リスナーを静的に構成するには、次の手順に従います。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Local」 -> 「Listeners」を展開します。
3. リスナーを選択します。



4. 右ペインのリストから、「Database Services」を選択します。
5. 「Add Database」をクリックします。新規のデータベース・タブが表示されます。



6. グローバル・データベース名、Oracle ホーム・ディレクトリ、インスタンスの Oracle システム識別子 (SID) を適切なフィールドに入力します。
7. 「File」->「Save Network Configuration」を選択します。

sales.us.acme.com と呼ばれるデータベースに対して静的に構成された listener.ora ファイルの例を次に示します。

```
sid_list_listener=
(sid_list=
(sid_desc=
(global_dbname=sales.us.acme.com)
(sid_name=sales)
(oracle_home=/u01/app/oracle/8.1.6)))
```



# 事前生成済専用サーバーの構成

事前生成済専用サーバーとは接続要求を受信する前の起動時にリスナーによって起動される専用サーバーです。所定のマシンでマルチスレッド・サーバー (multi-threaded server: MTS) を使用していない場合、サーバーでの接続確立に要する時間を短縮できます。また、他の接続で使用されたサーバーを切断および再作成せずにサーバー・プロセスを再利用することによって、割当て済のメモリーおよびシステム・リソースを有効に利用します。

**注意：**オラクル社では、パフォーマンスおよびスケーラビリティに関する問題を解決する際に、事前生成済専用サーバーではなく MTS を使用することをお薦めします。

**注意：**事前生成済専用サーバーは、Windows NT では利用できません。

listener.ora ファイルは、次のテーブルで説明される事前生成済専用サーバー・パラメータで構成されます。

listener.ora ファイルのパラメータ	Net8 Assistant のフィールド	説明
PRESPAWN_MAX	「Maximum Prespawned Servers」	リスナーによって事前生成される専用サーバーの最大数を指定します。この数値は、最小でも各プロトコル用のプール・サイズの合計と同じくする必要があります。事前生成済専用サーバーが常に新しい接続に使用できるように、この値を大きな値に設定します。
PROTOCOL	「TCP/IP」、「SPX」、「Named Pipes」、「IPC」	リスナーによって事前生成済専用サーバーの対象となるプロトコルです。
POOL_SIZE	「Number」	リスナーによって指定のプロトコルに関してメンテナンスされる、未使用の、事前生成済専用サーバーの数。0（ゼロ）より大きく、PRESPAWN_MAX 値以下の値を選択します。この値は、任意の時点で予測される平均接続数の近似値にしてください。



listener.ora ファイルの パラメータ	Net8 Assistant の フィールド	説明
TIMEOUT	「Timeout」	非活動状態にある事前生成済専用サーバー・プロセスが次の接続のために待機する時間（分単位）。値には、0（ゼロ）より大きい値を指定します。（値 0 では、非活動状態のシャドウ・プロセスが無限に続くため、マシンのリソースが無駄になります。）短いタイムアウト値を設定してください。タイムアウトは、事前生成済専用サーバー・プロセスが接続と切断を実行した後でのみアクティブ化されます。つまり、初回の接続のために待機している事前生成済専用サーバーでは、タイムアウトは発生しません。

Oracle8i リリース 8.1 では、リスナーによりクライアント要求が、listener.ora ファイルの静的情報ではなくサービス登録（service registration）を介して PMON から受信する情報に基づいて処理されます。特定のサービス名またはインスタンス名の動的情報が使用不能の場合、リスナーでは静的情報が参照されます。このため、事前生成済専用サーバーを使用するには、追加構成が必要です。

一部のクライアントでは、事前生成済専用サーバーを使用し、他のクライアントでは、リスナーに動的に登録されたサービス・ハンドラ（専用サーバー（dedicated server）または MTS ディスパッチャ（dispatcher））を使用するように構成できます。事前生成済専用サーバーを使用するクライアントには、特別な構成は必要ありません。クライアントでは、listener.ora ファイルの GLOBAL\_DBNAME または SID\_NAME パラメータ値に合致する SERVICE\_NAME または SID パラメータ値を含む接続記述子を使用される必要があります。

動的に登録されたサービス・ハンドラが使用されるクライアントでは、listener.ora ファイルで指定された GLOBAL\_DBNAME パラメータ値とは異なる SERVICE\_NAME パラメータ値を含む接続記述子を使用される必要があります。このサービス名は、SERVICE\_NAMES パラメータによって初期化パラメータ・ファイルに構成される必要があります。いったんデータベースが起動されると、サービス名がリスナーに動的に登録され、クライアントはデータベースに接続できます。

リリース 8.1 より古いクライアント環境では、初期化パラメータ・ファイルの INSTANCE\_NAME パラメータに対して類似の変更が必要です。これは listener.ora ファイルで指定された SID\_NAME パラメータ値とは異なる値に設定される必要があります。INSTANCE\_NAME パラメータ値は、SERVICE\_NAMES パラメータ値およびサービス・ハンドラ情報をともしリスナーに動的に登録されます。このように INSTANCE\_NAME パラメータ値が listener.ora ファイルの SID\_NAME パラメータ値とは異なるため、SID を含む（SID\_NAME に合致する）接続記述子を使用するクライアントは、事前生成済専用サーバーにリダイレクトされます。



構成作業については、次の項で説明します。

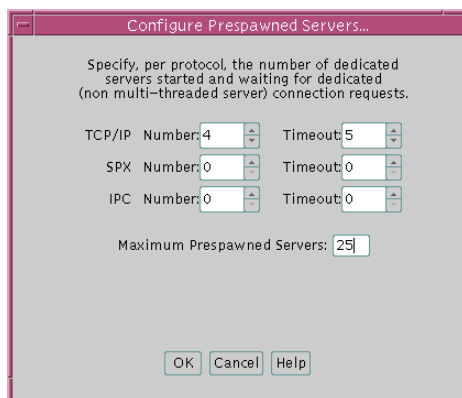
- [事前生成済専用サーバーによる listener.ora ファイルの構成](#)
- [事前生成済専用サーバーを使用しないクライアント構成の変更](#)

## 事前生成済専用サーバーによる listener.ora ファイルの構成

事前生成済専用サーバーを listener.ora ファイルで構成するには、次の作業を実行します。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Local」 -> 「Listeners」を展開します。
3. リスナーを選択します。
4. 右ペインのリストから、「Database Services」を選択します。
5. 「Use Prespawnd Dedicated Servers」を選択し、「Configure Prespawnd Servers」をクリックします。

「Configure Prespawnd Servers」ダイアログ・ボックスが表示されます。



6. 各プロトコルの起動専用サーバー数を指定します。
7. 「File」 -> 「Save Network Configuration」を選択します。



事前生成済専用サーバーで構成された listener.ora ファイルの例を次に示します。

```
sid_list_listener=
(sid_list=
(sid_desc=
(global_dbname=sales.us.acme.com)
(sid_name=sales)
(oracle_home=/u01/app/oracle/product/8.1.6)
(prespawn_max=25)
(prespawn_list=
(prespawn_desc=
(protocol=tcp)
(pool_size=4)
(timeout=5))))))
```

### 事前生成済専用サーバーを使用しないクライアント構成の変更

接続記述子に SERVICE\_NAME を使用するクライアントが存在する場合、初期化パラメータ・ファイルにある SERVICE\_NAMES パラメータの値を別の値と置換し、新しいサービス名に接続します。たとえば、初期化パラメータ・ファイルに service\_names=sales.us.acme.com が含まれる場合、sales.us.acme.com を sales2.us.acme.com と変更します。その後、次で示すように sales2.us.acme.com に接続される接続記述子を作成します。

```
sales=
(description=
(address=(protocol=tcp) (host=sales-server) (port=1521))
(connect_data=
(service_name=sales2.us.acme.com))
```

接続記述子に SID を使用するクライアントが存在する場合、初期化パラメータ・ファイルにある INSTANCE\_NAME パラメータの値を別の値と置換し、新しいインスタンスに接続します。たとえば、初期化パラメータ・ファイルに instance\_name=sales が含まれる場合、sales を sales2 と変更します。その後、次で示すように sales2 に接続される接続記述子を作成します。

```
sales=
(description=
(address=(protocol=tcp) (host=sales-server) (port=1521))
(connect_data=
(sid=sales2))
```



---

## Net8 の拡張機能

この章では、拡張機能を使用可能にする方法を説明します。内容は次のとおりです。

- 拡張ネットワーク・アドレスおよび接続データ情報の構成
- 透過的アプリケーション・フェイルオーバーの構成
- 名前解決のクライアント属性の構成
- プロファイルの詳細情報の構成
- 優先 Oracle Names サーバーの構成
- Oracle Connection Manager 機能の構成
- 非 Oracle データベース・サービスへの接続の構成



## 拡張ネットワーク・アドレスおよび接続データ情報の構成

この項では、拡張接続記述子に関する次のトピックを取り上げます。

- [リスナー・プロトコル・アドレスのリスト作成](#)
- [アドレス・リスト・パラメータの構成](#)
- [拡張接続データ・パラメータの構成](#)

### リスナー・プロトコル・アドレスのリスト作成

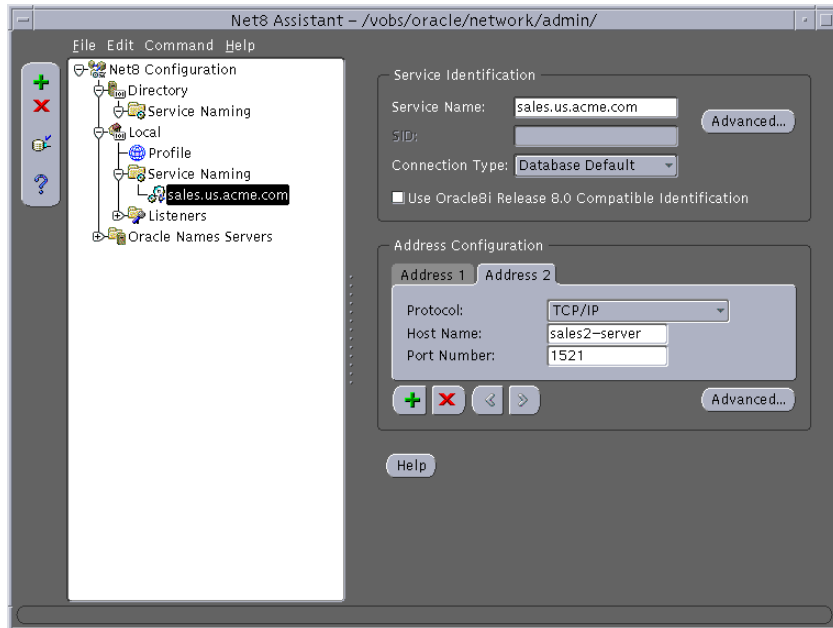
データベース・サービスは、複数のネットワーク・ルートまたはリスナー・アドレスからアクセスできます。次の例では、`sales1-server` または `sales2-server` のいずれかのリスナーを使用して、`sales.us.acme.com` に接続します。

```
sales.us.acme.com=  
(description=  
  (address_list=  
    (address=(protocol=tcp) (host=sales1-server) (port=1521))  
    (address=(protocol=tcp) (host=sales2-server) (port=1521)))  
  (connect_data=  
    (service_name=sales.us.acme.com)))
```

ネットワーク・プロトコル・アドレスを既存のネット・サービス名またはデータベース・サービスに追加するには、次の手順に従います。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、`$ORACLE_HOME/bin` から `netasst` を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Directory」または「Local」 -> 「Service Naming」を展開します。
3. ネット・サービス名またはデータベース・サービスのいずれかを選択します。  
右ペインには、現行の宛先サービスおよびアドレス・リストが表示されます。
4. 「Address Configuration」ボックスの「+」ボタンをクリックします。  
新規のアドレス・タブが表示されます。





5. プロトコルを選択し、必要なアドレス情報を入力します。

**関連項目：** プロトコル・アドレス・パラメータの詳細は、[付録 B](#) を参照してください。

6. 左矢印ボタンと右矢印ボタンを使用して、プロトコル・アドレスをアドレス・リスト内の適切な場所に配置します。複数のアドレス・オプションが構成されていない場合、リスト先頭のアドレスに対して通信が行われます。

**関連項目：** アドレス・リスト・オプションについては、8-3 ページの「[アドレス・リスト・パラメータの構成](#)」を参照してください。

7. 「Apply」をクリックします。

## アドレス・リスト・パラメータの構成

データベース・サービスに複数のリスナー・プロトコル・アドレスからアクセスできる場合は、アドレスの使用順序を指定する必要があります。たとえば、アドレスをランダムに選択することも、連続的に接続を試行することもできます。



1つのネット・サービス名またはデータベース・サービスに複数のプロトコル・アドレスが構成されている場合は、次のオプションを構成できます。

パラメータ	説明
<b>SOURCE_ROUTE</b> (ソース・ルーティング)	ON に設定すると、Net8 は各アドレスを表示順序どおりに使用して、接続先に到達するまで試行を続けます。接続先に到達するまでに複数のアドレス終点を通る必要がある場合は、このパラメータが必要です。この機能は通常、Oracle Connection Manager の機能を使用可能にするために使用します。
<b>FAILOVER</b> (接続時フェイルオーバー)	ON に設定した場合、最初のリスナーへの接続に失敗すると Net8 は別のリスナーにフェイルオーバーします。接続を試行する回数は、リスト内のアドレス数で決まります。OFF に設定すると、接続を試行するアドレスは1つのみです。  複数のアドレス・リスト (ADDRESS_LIST)、接続記述子 (DESCRIPTION) および複数の接続記述子 (DESCRIPTION_LIST) がある場合、接続時フェイルオーバーはデフォルトでオンになっています。  <b>重要:</b> listener.ora ファイルの SID_LIST_listener_name セクションには GLOBAL_DBNAME パラメータを設定しないでください。グローバル・データベース名が静的に構成されると、接続時フェイルオーバーは使用できなくなります。
<b>LOAD_BALANCE</b> (クライアント・ロード・バランス)	ON に設定すると、リスナー・アドレスのリストを順不同に選択して、複数のリスナーに対する負荷のバランスをとります。OFF に設定すると、接続が成功するまでアドレスを順番に試行します。  複数の接続記述子 (DESCRIPTION_LIST) がある場合、クライアント・ロード・バランスはデフォルトで ON になっています。

**注意:** 接続時フェイルオーバーとクライアント・ロード・バランスのパラメータを利用できるのは、リリース 8.1 構成のみです。ソース・ルーティングは、リリース 8.1 以前の構成およびリリース 8.0 構成の両方に利用できます。

クライアント・ロード・バランスや接続時フェイルオーバーをソース・ルーティングと併用することはできません。接続時フェイルオーバーとクライアント・ロード・バランスではリストから1つのアドレスを選択しますが、ソース・ルーティングではリスト内の各アドレスに順に接続します。

これらのパラメータの実装は、使用するネーミング・メソッドによって異なります。接続時フェイルオーバーまたはクライアント・ロード・バランスを構成するツールを次の表に示します。ソース・ルーティングには別の構成が必要となりますが、この項では取り上げません。



**関連項目：** ソース・ルーティングのクライアントの構成の詳細は、8-33 ページの「[Oracle Connection Manager を使用する場合のクライアントの構成](#)」を参照してください。

---

### ネーミング・メソッド 複数のアドレス・オプションを構成するツール

---

**ローカル・ネーミング** Oracle Net8 Assistant を使用して、tnsnames.ora ファイルに格納されるネット・サービス名について接続時フェイルオーバーおよびクライアント・ロード・バランスを行うように構成できます。  
下記の手順を参照してください。

**ディレクトリ・ネーミング** Oracle Net8 Assistant を使用して、ディレクトリに格納されるネット・サービス名およびデータベース・サービスについて接続時フェイルオーバーおよびクライアント・ロード・バランスを行うように構成できます。  
下記の手順を参照してください。

**Oracle Names** 次のツールを使用して、Oracle Names サーバーに格納されるネット・サービス名について接続時フェイルオーバーおよびクライアント・ロード・バランスを行うように構成できます。

- NAMESCTL [REGISTER](#)
- Oracle Net8 Assistant

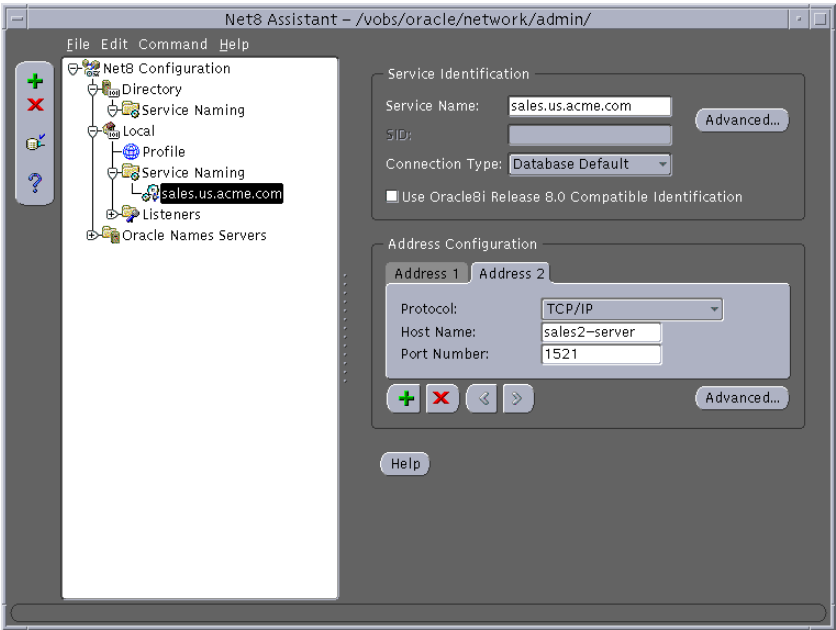
**関連項目：** 6-59 ページの「[作業 6: Oracle Names サーバーへのデータベースの登録](#)」を参照してください。

---

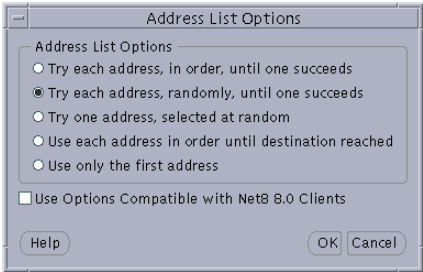
接続時フェイルオーバーまたはクライアント・ロード・バランスを構成するには、次の手順に従います。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Directory」または「Local」 -> 「Service Naming」を展開します。
3. ネット・サービス名またはデータベース・サービスのいずれかを選択します。  
右ペインには、現行の宛先サービスおよびアドレス・リストが表示されます。





4. 「Address Configuration」ボックスの「Advanced」ボタンをクリックします。  
「Address List Options」ダイアログ・ボックスが表示されます。



次の表は、オプションについて説明しています。



Oracle Net8 Assistant のオプション	パラメータ
「Try each address, in order, until one succeeds」	FAILOVER=ON (リリース 8.1 クライアントの場合) SOURCE_ROUTE=OFF (リリース 8.1 以前のクライアントの場合)
「Try each address, randomly, until one succeeds」 <b>注意:</b> このオプションは、「Use Options Compatible with Net8 8.0 Clients」が選択されている場合は、使用できません。	LOAD_BALANCE=ON FAILOVER=ON
「Try one address, selected at random」 <b>注意:</b> このオプションは、「Use Options Compatible with Net8 8.0 Clients」が選択されている場合は、使用できません。	LOAD_BALANCE=ON
「Use each address in order until destination reached」	SOURCE_ROUTE=ON
「Use only the first address」 <b>注意:</b> このオプションは、「Use Options Compatible with Net8 8.0 Clients」が選択されている場合は、使用できません。	LOAD_BALANCE=OFF FAILOVER=OFF SOURCE_ROUTE=OFF

5. アドレス・リスト・オプションの 1 つを選択して「OK」をクリックします。

6. 右ペインで「Apply」をクリックします。

次の例は、「Try each address, randomly, until one succeeds」オプションを使用してクライアント・ロード・バランスを指定した場合の tnsnames.ora ファイルを表します。

```
sales.us.acme.com=
(description=
(address_list=
(load_balance=on)
(address=(protocol=tcp) (host=sales1-server) (port=1521))
(address=(protocol=tcp) (host=sales2-server) (port=1521)))
(connect_data=
(service_name=sales.us.acme.com)))
```

次の例は、「Try each address, in order, until one succeeds」オプションを使用して接続時フェイルオーバーを指定した場合の tnsnames.ora ファイルを表します。

```
sales.us.acme.com=
(description=
```



```
(address_list=
  (load_balance=off)
  (failover=on)
  (address=(protocol=tcp) (host=sales1-server) (port=1521))
  (address=(protocol=tcp) (host=sales2-server) (port=1521)))
(connect_data=(service_name=sales.us.acme.com))
```

拡張接続データ・パラメータの構成

接続記述子の接続データ部分には接続先データベース・サービスを定義します。次の例では、SERVICE\_NAME に sales.us.acme.com というサービスが定義されています。

```
sales.us.acme.com=
  (description=
    (address=(protocol=tcp) (host=sales-server) (port=1521))
    (connect_data=
      (service_name=sales.us.acme.com)))
```

サービス名の他にも、オプションで次の接続データ情報のパラメータを構成できます。

tnsnames.ora ファイル のパラメータ	Oracle Net8 Assistant のフィー ルド/オプション	説明
INSTANCE_NAME	「Instance Name」	アクセスするデータベース・インスタンスを識別します。  インスタンス名は、 <a href="#">初期化パラメータ・ファイル (initialization parameter file)</a> の INSTANCE_NAME パラメータから取得できます。  <b>注意：</b> 「Use Oracle8i Release 8.0 Compatible Identification」が選択されている場合は、使用できません。  <b>関連項目：</b> 6-2 ページの「 <a href="#">接続記述子の理解</a> 」を参照してください。
SDU	「Session Data Unit Size」	ネットワークを介して送信されるデータ・パケットの転送レートを最適化する場合、セッション・データ・ユニット (SDU) サイズを指定すると、パフォーマンスを変更できます。  <b>関連項目：</b> 5-7 ページの「 <a href="#">データ転送の最適化を目的としたセッション・データ・ユニット (SDU) サイズの調整</a> 」を参照してください。
HS	「Use for Heterogeneous Services」	Oracle8i サーバーで非 Oracle システムにアクセスする場合は、このオプションをオンにします。  <b>関連項目：</b> 8-43 ページの「 <a href="#">Oracle 異種サービスを使用する場合の Net8 の構成</a> 」を参照してください。



tnsnames.ora ファイル のパラメータ	Oracle Net8 Assistant のフィー ルド / オプション	説明
<a href="#">RDB_DATABASE</a>	「Oracle Rdb Database」	Oracle Rdb データベースのファイル名を指定します。 <b>関連項目：</b> 8-46 ページの「 <a href="#">Oracle Rdb データベースを使用する場合の Net8 の構成</a> 」を参照してください。
TYPE_OF_SERVICE	「Type of Service」	Oracle Rdb データベースで使用するサービスの種類を指定します。 <b>関連項目：</b> 8-46 ページの「 <a href="#">Oracle Rdb データベースを使用する場合の Net8 の構成</a> 」を参照してください。
GLOBAL_NAME	「Global Database Name」	Oracle Rdb データベースを識別します。 <b>関連項目：</b> 8-46 ページの「 <a href="#">Oracle Rdb データベースを使用する場合の Net8 の構成</a> 」を参照してください。

これらのパラメータの設定方法は、使用するネーミング・メソッドによって異なります。構成するツールを次の表に示します。

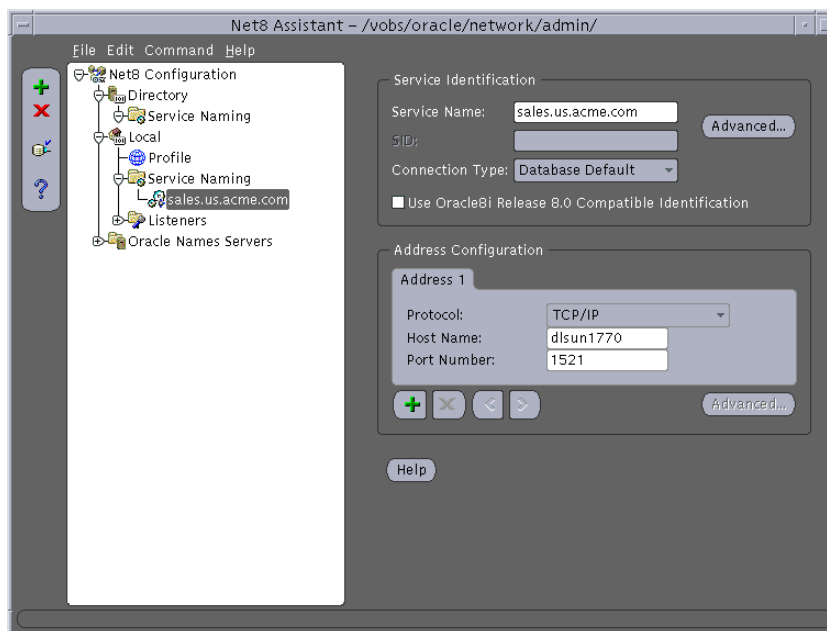
ネーミング・メソッド	複数のアドレス・オプションを構成するツール
ローカル・ネーミング	Oracle Net8 Assistant を使用して、tnsnames.ora ファイルに格納されるネット・サービス名について拡張接続データ情報を構成できます。 下記の手順を参照してください。
ディレクトリ・ネーミング	Oracle Net8 Assistant を使用して、ディレクトリに格納されるネット・サービス名およびデータベース・サービスについて拡張接続データ情報を構成できます。 下記の手順を参照してください。
Oracle Names	次のツールを使用して、Oracle Names サーバーに格納されるネット・サービス名およびデータベース・サービスについて拡張接続データ情報を構成できます。 <ul style="list-style-type: none"> <li>■ NAMESCTL REGISTER</li> <li>■ Oracle Net8 Assistant</li> </ul> <b>関連項目：</b> 6-59 ページの「 <a href="#">作業 6: Oracle Names サーバーへのデータベースの登録</a> 」を参照してください。

ネット・サービス名またはデータベース・サービスの拡張 CONNECT\_DATA パラメータを構成するには、次の手順に従います。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。

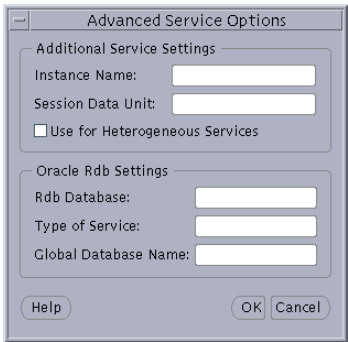


- Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
  - 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Directory」または「Local」 -> 「Service Naming」を展開します。
  3. ネット・サービス名またはデータベース・サービスのいずれかを選択します。  
右ペインには、現行の宛先サービスおよびアドレス・リストが表示されます。



4. 「Service Identification」ボックスの「Advanced」ボタンをクリックします。  
「Advanced Service Options」ダイアログ・ボックスが表示されます。





- 5. 必要に応じてフィールドの入力やオプションの選択を行い、「OK」をクリックします。
- 6. 右ペインで「Apply」をクリックします。

## 透過的アプリケーション・フェイルオーバーの構成

透過的アプリケーション・フェイルオーバー（TAF）を使用すると、一度確立した接続に障害が発生した場合、Net8 は別のリスナーにフェイルオーバーします。これによりユーザーは、最初の接続に障害が発生したことを意識せずに、新しい接続で作業を続行できます。

TAF を使用する場合は、ネット・サービス名の構成において接続記述子の CONNECT\_DATA 部分に FAILOVER\_MODE パラメータを手動で設定する必要があります。

この項では、次のトピックについて説明します。

- [FAILOVER\\_MODE パラメータ](#)
- [TAF 実装](#)
- [検証](#)

### FAILOVER\_MODE パラメータ

FAILOVER\_MODE パラメータは、接続記述子の CONNECT\_DATA 部分に含まれている必要があります。FAILOVER\_MODE には、次のパラメータが含まれています。

tnsnames.ora ファイルの 説明 パラメータ	
BACKUP	バックアップ接続として別のネット・サービス名を指定します。 バックアップは、PRECONNECT を使用して事前に接続を確立する場合に指定します。



**tnsnames.ora ファイルの 説明  
パラメータ**

TYPE (必須)	<p>フェイルオーバーの種類を指定します。デフォルトでは、Oracle Call Interface (OCI) アプリケーションに対して、3 種類の Net8 フェイルオーバー機能を利用できます。</p> <ul style="list-style-type: none"><li>■ SESSION: セッションをフェイルオーバーします。すなわち、ユーザーの接続が失われた場合に、新しいセッションをバックアップへ自動的に作成します。このフェイルオーバーでは、SELECT はリカバリされません。</li><li>■ SELECT: カーソルをオープンしているユーザーが、障害後もカーソルをフェッチできるようにします。ただしこのモードでは、通常の SELECT 操作を行うクライアント側にオーバーヘッドが生じます。</li><li>■ NONE: デフォルトです。フェイルオーバー機能を使用しません。フェイルオーバーが実行されないように、明示的に指定することもできます。</li></ul>
METHOD	<p>プライマリ・ノードからバックアップ・ノードにフェイルオーバーされる速さを決定します。</p> <ul style="list-style-type: none"><li>■ BASIC: フェイルオーバー時に接続を確立します。フェイルオーバー時まで、バックアップ・サーバーはほとんど何もする必要はありません。</li><li>■ PRECONNECT: あらかじめ接続を確立しておきます。フェイルオーバーは高速に行われますが、バックアップ・インスタンスはサポートされた各インスタンスからの接続すべてをサポートする必要があります。</li></ul>
RETRIES	<p>接続を行う回数を指定します。DELAY が指定されている場合、RETRIES のデフォルトは、5 回再試行するように設定されています。</p>
DELAY	<p>次の接続試行まで待機する時間を秒で指定します。RETRIES が指定されると、DELAY はデフォルトで 1 秒に設定されます。</p>

**注意：** Oracle Net8 Assistant では、TAF パラメータはサポートされません。これらのパラメータは、手動で追加する必要があります。

TAF 実装

**重要：** listener.ora の SID\_LIST\_listener\_name セクションには GLOBAL\_DBNAME パラメータを設定しないでください。グローバル・データベース名が静的に構成されると、TAF は使用できなくなります。



TAF は、FAILOVER\_MODE パラメータの設定次第で何通りにでも実装できます。オラクル社では、次の方法をお薦めします。

- 接続時フェイルオーバーおよびクライアント・ロード・バランスを行う TAF の実装
- 接続の再試行
- 接続の事前確立

## 接続時フェイルオーバーおよびクライアント・ロード・バランスを行う TAF の実装

複数のアドレスについて接続時フェイルオーバーおよびクライアント・ロード・バランスを行うように TAF を実装することができます。次の例では、Net8 は sales1-server または sales2-server のいずれかのリスナー・アドレスにランダムに接続します。接続後にインスタンスに障害が発生した場合は、他のノードのリスナーにフェイルオーバーし、実行中の SELECT 文があればその文を保持します。

```
sales.us.acme.com=
  (description=
    (load_balance=on)
    (failover=on)
    (address=
      (protocol=tcp)
      (host=sales1-server)
      (port=1521))
    (address=
      (protocol=tcp)
      (host=sales2-server)
      (port=1521))
    (connect_data=
      (service_name=sales.us.acme.com)
      (failover_mode=
        (type=select)
        (method=basic))))
```

## 接続の再試行

TAF では、RETRIES および DELAY パラメータを使用して、最初の接続試行が失敗した場合に自動的に接続を再試行することができます。次の例では、Net8 は sales1-server のリスナーへの接続を試行します。この接続試行に失敗した場合は、15 秒待機してから再び接続を試行します。接続試行数は最大 20 回です。

```
sales.us.acme.com=
  (description=
    (address=
      (protocol=tcp)
```



```
(host=sales1-server)
(port=1521))
(connect_data=
  (service_name=sales.us.acme.com)
  (failover_mode=
    (type=select)
    (method=basic)
    (retries=20)
    (delay=15))))
```

### 接続の事前確立

バックアップ接続は事前に確立することができます。初期接続およびバックアップ接続は明示的に指定する必要があります。次の例では、Net8 は sales1-server のリスナーに接続します。接続後に sales1-server に障害が発生した場合は、sales2-server にフェイルオーバーし、実行中の SELECT 文があればその文を保持します。

```
sales.acme.com=
  (description=
    (address=
      (protocol=tcp)
      (host=sales1-server)
      (port=1521))
    (connect_data=
      (service_name=sales.us.acme.com)
      (instance_name=sales1)
      (failover_mode=
        (backup=sales2.acme.com)
        (type=select)
        (method=preconnect))))
sales2.acme.com=
  (description=
    (address=
      (protocol=tcp)
      (host=sales2-server)
      (port=1521))
    (connect_data=
      (service_name=sales.us.acme.com)
      (instance_name=sales2)))
```

### 検証

V\$SESSION ビューで FAILOVER\_TYPE、FAILOVER\_METHOD および FAILED\_OVER 列を問い合わせれば、TAF が正しく構成されているかを確認できます。

**関連項目：** V\$SESSION ビューの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。



## 名前解決のクライアント属性の構成

次に示す項では、使用可能なクライアント構成オプションについて説明します。

- クライアントのデフォルト・ドメインの構成
- ネーミング・メソッドの優先順位
- 接続要求のルーティング

### クライアントのデフォルト・ドメインの構成

クライアントが特定ドメインの名前を要求することが多い環境では、NAMES.DEFAULT\_DOMAIN パラメータを使用してクライアントの sqlnet.ora ファイルにデフォルト・ドメインを設定します。

デフォルト・ドメインが設定されると、そのドメインは接続文字列で指定された未修飾の ネット・サービス名に自動的に追加され、tnsnames.ora ファイルまたは Oracle Names サーバーに格納された ネット・サービス名と比較されます。

たとえば、クライアント tnsnames.ora ファイルに sales.us.acme.com という ネット・サービス名がある場合は、接続文字列を次のように入力できます。

```
CONNECT scott/tiger@sales
```

この例では、sales は sales.us.acme.com として検索されます。

CONNECT scott/tiger@sales.us.acme.com のように接続文字列にドメイン拡張子が含まれる場合は、ドメインは追加されません。tnsnames.ora ファイルまたは Oracle Names サーバー内の ネット・サービス名がドメインで修飾されておらず、このパラメータが設定されている場合は、ネット・サービス名にドット (.) を付けて入力します。たとえば、ドメインが us.acme.com に設定され、クライアント tnsnames.ora ファイルに sales という ネット・サービス名がある場合は、次のように接続文字列を入力します。

```
CONNECT scott/tiger@sales.
```

---

---

**注意：** デフォルト・ドメインは、インストール時に Net8 Configuration Assistant によってクライアントのネットワーク・ドメインに設定されます。

---

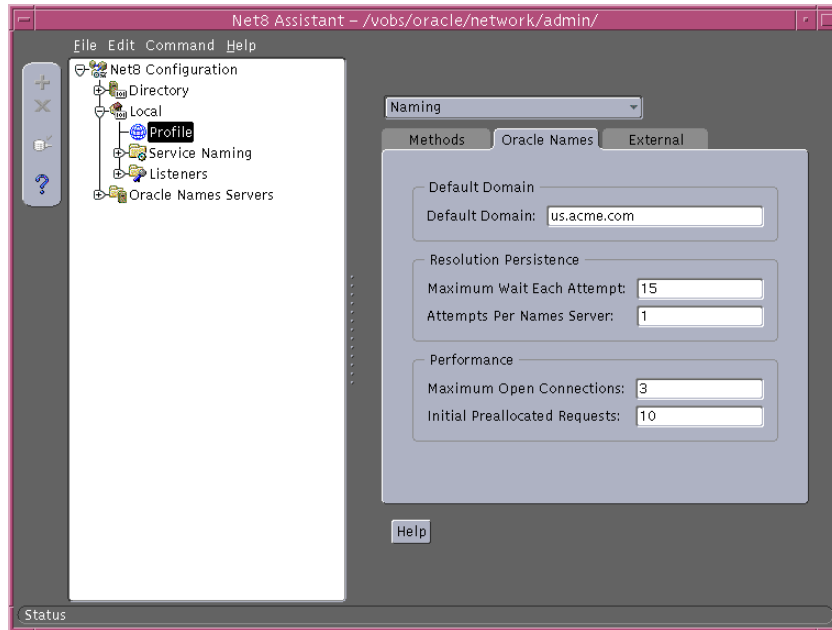
---

デフォルト・ドメインを指定するには、次の手順に従います。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。



2. ナビゲータで、「Local」->「Profile」を展開します。
3. 右ペインのリストから「Naming」を選択します。
4. 「Oracle Names」タブをクリックします。



5. 「Default Domain」フィールドにドメインを入力します。
6. 「File」->「Save Network Configuration」を選択します。

sqlnet.ora ファイルには、次のようなエントリが追加されます。

```
names.default_domain=com
```

## ネーミング・メソッドの優先順位

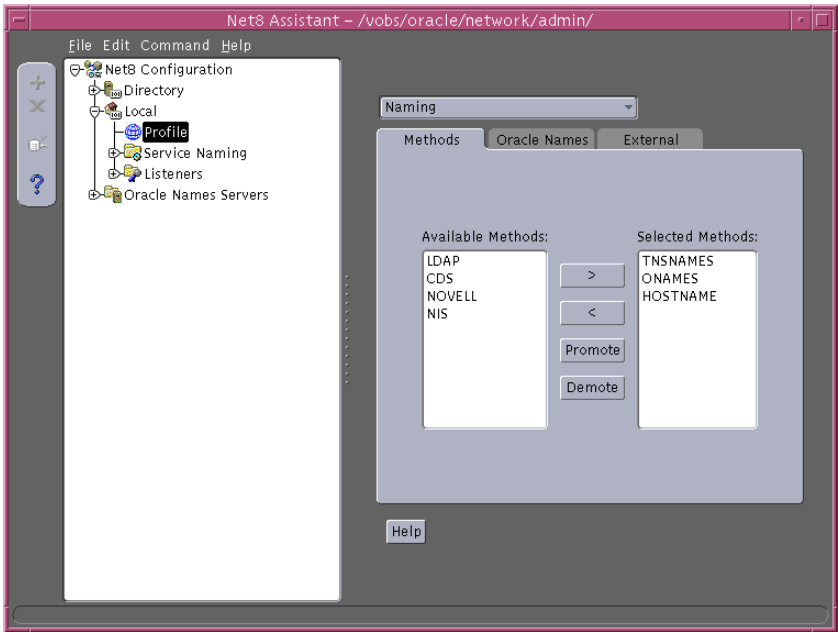
ネーミング・メソッドを構成した後は、第6章に説明されているように、優先順位を設定する必要があります。ネット・サービス名の解決には、まずリストの最初のネーミング・メソッドが使用されます。この方法でネット・サービス名が解決できない場合、リストの2番目の方法が使用されます。

ネーミング・メソッドの順位を指定するには、次の手順に従います。

1. Oracle Net8 Assistant を起動します。



- UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン->「プログラム」->「Oracle - HOME\_NAME」->「Network Administration」->「Net8 Assistant」を選択します。
2. ナビゲータで、「Local」->「Profile」を展開します。
  3. 右ペインのリストから「Naming」を選択します。
  4. 「Methods」タブをクリックします。



使用可能なネーミング・メソッドを次の表に示します。

ネーミング・メソッド	説明
TNSNAMES	ローカル・ネーミング ネット・サービス名は、クライアントに配置された tnsnames.ora ファイルにより解決されます。
LDAP	ディレクトリ・ネーミング ネット・サービス名およびデータベース・サービス名はディレク トリ・サーバーにより解決されます。



ネーミング・メソッド	説明
ONAMES	Oracle Names  ネット・サービス名およびデータベース・サービス名は Oracle Names サーバーにより集中的に解決されます。
HOSTNAME	ホスト・ネーミング  ネット・サービス名は、ホストネーミング・メソッドにより解決されます。ホスト・ネーミングを使用するには、一定の条件を満たす必要があります。  <b>関連項目：</b> 6-68 ページの「 <a href="#">ホスト・ネーミング・メソッドの構成</a> 」を参照してください。
CDS	セル・ディレクトリ・サービス  このネーミング・メソッドは、Oracle Advanced Security とともに使用できます。  <b>関連項目：</b> 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。
NDS	Novell ディレクトリ・サービス (NDS)
NIS	ネットワーク情報サービス (NIS)

- 5. 「Available Methods」リストから目的のネーミング・メソッドを選択します。
- 6. 「Available Methods」リストからネーミング・メソッドを選択して、右矢印ボタンをクリックします。  
  
ネーミング・メソッドが「Selected Methods」リストに移動します。
- 7. Net8 によるネット・サービス名またはデータベース・サービス名の解決を行いたい順にネーミング・メソッドを配置します。「Selected Methods」リスト内でネーミング・メソッドを選択し、「Promote」または「Demote」をクリックして選択項目をリスト内で移動します。
- 8. 「File」->「Save Network Configuration」を選択します。

```
sqlnet.ora ファイルの NAMES.DIRECTORY_PATH パラメータが次のように更新されます。  
  
names.directory_path=(onames, tnsnames)
```

接続要求のルーティング

クライアントと、クライアントとして動作するサーバーは、接続要求をいつでも特定のプロセスに送るように構成できます。次の表に示す sqlnet.ora ファイル・パラメータは、接続要求のルート設定に使用できます。



sqlnet.ora ファイルのパラメータ	Oracle Net8 Assistant のオプション	説明
USE_DEDICATED_SERVER	「Always Use Dedicated Server」	リスナーがこのクライアントから発生するすべてのネットワーク・セッションで専用サーバーを使用するように設定します。自動的に、接続記述子 CONNECT_DATA 部分に (SERVER=DEDICATED) を追加します。これにより、 <b>マルチスレッド・サーバー (multi-threaded server: MTS)</b> が構成されている場合でも、このクライアントからの接続には専用サーバーが使用されます。
USE_CMAN	「Prefer Connection Manager Routing」	可能な限り Oracle Connection Manager を使用するようにクライアントに対して指示します。Oracle Connection Manager のアドレスが不明な場合は、利用できるリスナー・アドレスを介して接続のルートが設定されます。  <b>注意：</b> Oracle Names で Oracle Connection Manager を使用している場合は、クライアントと Oracle Names サーバーの両方で、このオプションを設定する必要があります。

---

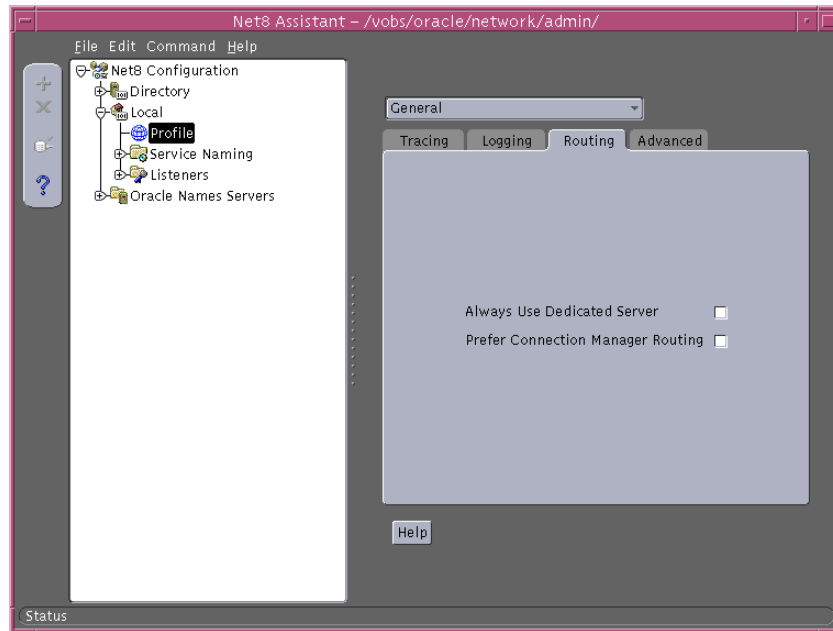
**注意：** Oracle Net8 Assistant には「Use IPC Address for Client」が表示されますが、リリース 8.1 ではこのオプションに対応するパラメータ AUTOMATIC\_IPC が使用禁止になっているため、この機能は使用できません。

---

接続要求のルートを設定するには、次の手順に従います。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Local」 -> 「Profile」を展開します。
3. 右ペインのリストから「General」を選択します。
4. 「Routing」タブをクリックします。





5. 接続要求のルートを設定するいずれかのオプションをクリックします。
6. 「File」->「Save Network Configuration」を選択します。

## プロファイルの詳細情報の構成

sqlnet.ora ファイルには次の拡張パラメータを構成できます。



sqlnet.ora ファイルの パラメータ	Oracle Net8 Assistant の フィールド / オプション	説明
SQLNET.EXPIRE_TIME	「TNS Time Out Value」	<p>このオプションを使用できるのはサーバーのみです。</p> <p>デッド接続タイムアウト値を設定してください。Net8 では、クライアント / サーバーの接続がまだアクティブであるかどうかを確認するために、プローブを定期的に送信します。これは、クライアントの異常終了により、接続が無限にオープンされたままの状態かどうかを確認します。プローブがデッド接続またはすでに使用されていない接続を検出すると、エラーを返し、それによってサーバー・プロセスが終了します。</p> <p>デッド接続検出機能を使用する上での制約は、次のとおりです。</p> <ul style="list-style-type: none"> <li>■ デッド接続検出は、Bequeath 接続では許可されていません。</li> <li>■ プローブ・パケットはごく小さいものですが、ネットワーク・パフォーマンスを低下させるおそれのある余分な通信量を生成します。</li> <li>■ 使用しているオペレーティング・システムによっては、接続プローブ・イベントを他の発生イベントと区別するために、サーバーで追加の処理を行う必要がある場合があります。このために、ネットワーク・パフォーマンスが低下する場合もあります。</li> </ul>
SQLNET.CLIENT_REGISTRATION	「Client Registration ID」	<p>クライアント・マシンの一意な識別子を設定します。この識別子は接続要求とともにリスナーに渡されます。識別子として、最大 128 文字の任意の文字を使用できます。</p>
BEQUEATH_DETACH	「Turn Off UNIX Signal Handling」	<p>クライアント・アプリケーションは bequeath プロトコルを介して内部にサーバー・プロセスを子プロセスとして生成するため、クライアント・アプリケーションは終了したときにその子プロセスをクリーン・アップする必要があります。サーバー・プロセスが接続の責任を完遂すると、それは消滅プロセスになります。シグナル・ハンドラは、これらの消滅プロセスをクリーン・アップする必要があります。このパラメータを設定してシグナル・ハンドラを使用禁止にすることにより、消滅プロセスを UNIX init プロセスに渡すようにクライアント・プロファイルを構成できます。</p>

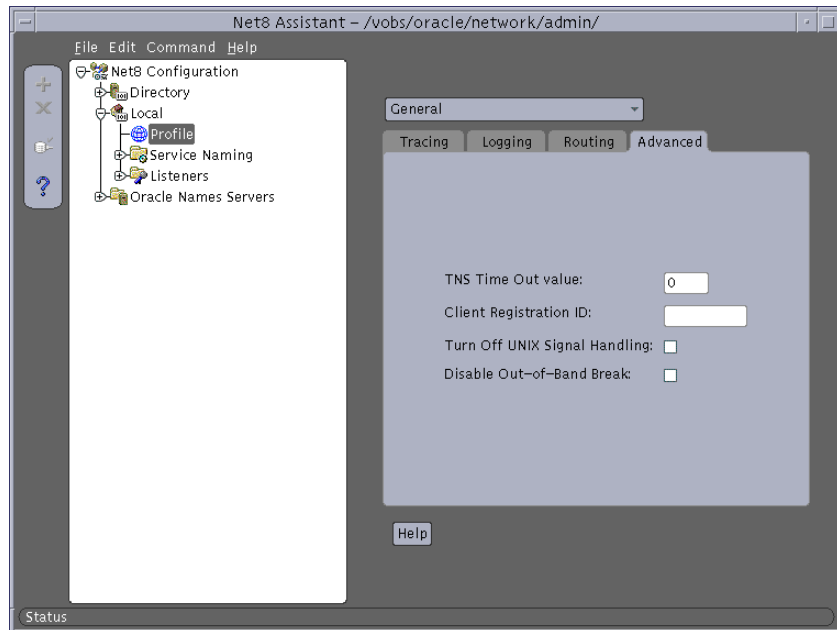


sqlnet.ora ファイルの パラメータ	Oracle Net8 Assistant の フィールド / オプション	説明
DISABLE_OOB	「Disable Out-of-Band Break」	<p>選択解除するか OFF にすると、基底のプロトコルで提供される緊急データを使用してブレイク・メッセージを送受信することができます。</p> <p>選択するか ON にすると、基底のプロトコルで提供される緊急データによるブレイク・メッセージの送受信は使用できなくなります。一度使用可能にすると、この機能は対象クライアントが使用するすべてのプロトコルに適用されます。</p> <p><b>関連項目：</b> 使用プロトコルが緊急データ要求をサポートしているかを確認するには、Oracle オペレーティング・システム・マニュアルを参照してください。この機能をサポートするプロトコルの例として TCP/IP があります。</p>

クライアントの拡張機能を設定するには、次の手順に従います。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Local」 -> 「Profile」を展開します。
3. 右ペインのリストから「General」を選択します。
4. 「Advanced」タブを選択します。





5. 設定するオプションに値を入力します。
6. 「File」->「Save Network Configuration」を選択します。

## 優先 Oracle Names サーバーの構成

Oracle Names をネーミング・メソッドとして使用している場合は、**優先 Oracle Names サーバー (Preferred Oracle Names Server)** の名前とアドレスを指定できます。Net8 は、応答を受信するまで、解決要求をそれぞれの優先 Oracle Names サーバーに発信します。

優先 Oracle Names サーバーは、Oracle Net8 Assistant の「Discover Oracle Names Servers」コマンドまたは NAMESCTL 制御ユーティリティの「**REORDER\_NS**」コマンドのかわりに使用できます。これらのコマンドは、ネットワークの中で使用可能な Oracle Names サーバーのリストを作成し、それらを応答時間の速い順に並べます。これに対して優先 Oracle Names サーバーは、検出プロセスで検索された他の Oracle Names サーバーよりも優先されます。最初に Oracle Names サーバーが検出されると、優先 Oracle Names サーバーを削除しても構いません。

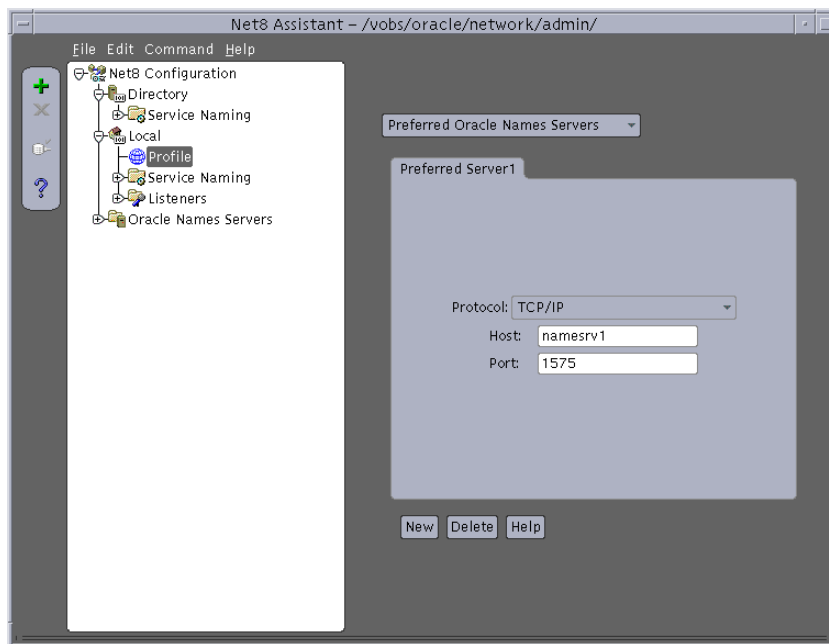


### 関連項目：

- Oracle Names サーバー・リストの作成方法は、4-18 ページの「[Oracle Names のバージョン間の相違](#)」を参照してください。
- Oracle Names サーバー検出に関する詳細は、6-55 ページの「[作業 4: Oracle Names サーバーを使用するためのクライアントとデータベース・サーバーの構成](#)」を参照してください。

優先 Oracle Names サーバーを指定するには、次の手順に従います。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、`$ORACLE_HOME/bin` から `netasst` を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Local」 -> 「Profile」を展開します。
3. 右ペインのリストから「Preferred Oracle Names Servers」を選択します。
4. 「New」をクリックします。





5. プロトコルを選択し、そのアドレスで構成する Oracle Names サーバーに必要なプロトコル・アドレス情報を入力します。

**関連項目：** プロトコル・パラメータ設定の詳細は、[付録 B](#) を参照してください。

6. リストに Oracle Names サーバーを追加するたびにステップ 4 と 5 を繰り返します。
7. 「File」->「Save Network Configuration」を選択します。

sqlnet.ora ファイルの NAMES.PREFERRED\_SERVERS パラメータが次のように更新されます。

```
names.preferred_servers=
  (address=(protocol=tcp) (host=namesrv1) (port=1575))
```

---

---

**注意：** 優先 Oracle Names サーバーは、names.ora ファイルで [NAMES.ADDRESSES](#) パラメータを使用して設定した Oracle Names リスニング・プロトコル・アドレスと一致している必要があります。

---

---

## Oracle Connection Manager 機能の構成

次に示す項では、Oracle Connection Manager 機能の構成方法について説明します。

- [接続集中化機能の使用](#)
- [複数プロトコル・サポートの使用](#)
- [Net8 アクセス制御の使用](#)
- [Oracle Connection Manager を使用する場合のクライアントの構成](#)

### 接続集中化機能の使用

Oracle Connection Manager では、マルチスレッド・サーバー接続先への単一トランスポート接続を通じて複数の論理クライアント・ネットワーク・セッションを多重化または集中化する機能が利用できます。Oracle Connection Manager の接続集中化機能を通じて、この機能が実現されます。

接続集中化を構成するには、次の作業を行います。

[作業 1: Oracle Connection Manager の構成](#)

[作業 2: サーバーの構成](#)

[作業 3: クライアントの構成](#)



作業 1: Oracle Connection Manager の構成

Oracle Connection Manager は、次のデフォルト・リスニング・アドレスでクライアント接続要求を受け入れます。

```
cman=(address=(protocol=tcp) (host=anyhost) (port=1630))
```

また、Oracle Connection Manager は、次のリスニング・アドレスでローカルおよびリモートの管理コマンドをリスニングします。

```
cman_admin=(address=(protocol=tcp) (host=anyhost) (port=1830))
```

デフォルト・アドレスを使用しない場合は、cman.ora ファイルに次のように指定します。

```
cman=(address=( プロトコル・アドレス情報 ))
cman_admin=(address=( プロトコル・アドレス情報 ))
```

cman.ora ファイルは UNIX では \$ORACLE\_HOME/network/admin に、Windows NT では ORACLE\_HOME¥network¥admin に置きます。

関連項目：

- cman.ora ファイルの詳細は、C-65 ページの「[Oracle Connection Manager パラメータ \(cman.ora\)](#)」を参照してください。
- [付録 B](#) (アドレス情報に必要なプロトコル構文に関する詳細) を参照してください。

**注意：** Oracle Net8 Assistant では cman.ora ファイルの構成をサポートしていないので、変更は手動で行う必要があります。

作業 2: サーバーの構成

接続先サーバーが **マルチスレッド・サーバー (multi-threaded server: MTS)** として構成され、多重化オプションがオンになっていることを確認します。MTS および多重化をオンにするには、Oracle Database Configuration Assistant を使用するか、あるいは **初期化パラメータ・ファイル (initialization parameter file)** で MTS\_DISPATCHERS パラメータに PROTOCOL および MUX 属性を設定します。たとえば、次のように指定します。

```
mts_dispatchers="(protocol=tcp) (multiplex=on)"
```

属性	説明
PROTOCOL (PRO または PROT)	ディスパッチャでリスニング終点を生成するために使用されるネットワーク・プロトコル (上の例では TCP)。



属性	説明
MULTIPLEX (MUL または MULT)	<p>接続集中化を使用可能にするために使用されます。</p> <p>1、ON、YES、TRUE または BOTH が指定されると、受信と送信の両方のネットワーク接続で接続集中化が使用可能です。</p> <p>IN が指定されると、クライアントから受信するネットワーク接続で接続集中化が使用可能です。</p> <p>OUT が指定されると、送信するネットワーク接続で接続集中化が使用可能です。</p> <p>0、NO、OFF または FALSE が指定されると、受信と送信の両方のネットワーク接続で接続集中化が使用禁止です。</p>

**関連項目：** MTS の構成方法の詳細は、[第 9 章](#)を参照してください。

### 作業 3: クライアントの構成

クライアント・サポートを実現するには 2 つのアドレスを作成します。このうち最初のアドレスは Oracle Connection Manager、2 番目はリスナーのアドレスです。次の表に示すように、構成はネーミング・メソッドによって異なります。

ネーミング・メソッド	手順
ローカル・ネーミングおよびディレクトリ・ネーミング	<p>アドレス・リストを作成します。リストの最初のアドレスは Oracle Connection Manager、2 番目はリスナーのアドレスです。最初にクライアントが Oracle Connection Manager に接続し、次に Oracle Connection Manager がリスナーに接続するようにするには、SOURCE_ROUTE パラメータを設定します。</p> <p><b>関連項目：</b> Oracle Connection Manager およびリスナーのアドレスの構成方法は、8-33 ページの「<a href="#">Oracle Connection Manager を使用する場合のクライアントの構成</a>」を参照してください。</p>
Oracle Names	<p>Oracle Names サーバーではリスナーおよび Oracle Connection Manager のアドレスは自動的に更新されます。</p> <p>Oracle Names サーバーを使用するクライアントから Oracle Connection Manager に要求を送信するには、各 Oracle Names サーバーが存在するマシンおよびクライアント上の sqlnet.ora ファイルに USE_CMAN=TRUE パラメータを設定します。このパラメータを設定するには、8-18 ページの「<a href="#">接続要求のルーティング</a>」にある手順に従います。</p>



例 8-1 に、通常の tnsnames.ora ファイルと、Oracle Connection Manager を使用するためのエントリ（Oracle Connection Manager のエントリは太字部分）が追加された tnsnames.ora ファイルの比較を示します。

例 8-1 Oracle Connection Manager を使用する場合と使用しない場合の tnsnames.ora ファイル

sales=  
 (description=  
 (address=  
 (protocol=tcp)  
 (host=sales-pc)  
 (port=1521))  
 (connect\_data=  
 (service\_name=sales.com)))

sales=  
 (description=  
 (**source\_route=yes**)  
 (**address=**  
 (**protocol=tcp**)  
 (**host=cman-pc**)  
 (**port=1630**))  
 (address=  
 (protocol=tcp)  
 (host=sales-pc)  
 (port=1521))  
 (connect\_data=  
 (service\_name=sales.com)))

tnsnames.ora ファイルの要素	説明
source_route=yes	Oracle Connection Manager から接続先データベースまでのアドレスのソース・ルートを作成します。
(address= (protocol=tcp) (host=cman-pc) (port=1630))	最初のアドレスは Oracle Connection Manager までのアドレスです。ここから、リスナーを介してデータベース・サービスに接続します。

複数プロトコル・サポートの使用

**注意：** 接続パスで複数の Oracle Connection Manager が使用されている場合は、Oracle Connection Manager を通じて接続するためにクライアントが Oracle Names を使用することはできません。

Oracle Connection Manager には、異なるネットワーク・プロトコルを持つクライアントとサーバーが相互に通信できるようにする複数プロトコルのサポートがあります。Oracle Connection Manager では、Oracle でサポートされるあらゆるプロトコルでリスニングできます。



複数プロトコル・サポートを使用しない場合、SPX を使用するクライアントは、TCP/IP を使用するサーバーに接続できません。Oracle Connection Manager が TCP/IP 用に構成されていれば、クライアントは SPX を使用して Oracle Connection Manager に接続でき、Oracle Connection Manager は TCP/IP を使用するサーバーに接続できます。

複数プロトコル・サポートを構成するには、次の作業を行います。

[作業 1: Oracle Connection Manager の構成](#)

[作業 2: クライアントの構成](#)

## 作業 1: Oracle Connection Manager の構成

Oracle Connection Manager は、次のデフォルト・リスニング・アドレスでクライアント接続要求を受け入れます。

```
cman=(address=(protocol=tcp) (host=anyhost) (port=1630))
```

さらに、次のリスニング・アドレスでローカルおよびリモートの管理コマンドをリスニングします。

```
cman_admin=(address=(protocol=tcp) (host=anyhost) (port=1830))
```

デフォルト・アドレスを使用しない場合は、cman.ora ファイルに次のように指定します。

```
cman=(address=(プロトコル・アドレス情報))  
cman_admin=(address=(プロトコル・アドレス情報))
```

cman.ora ファイルは UNIX では \$ORACLE\_HOME/network/admin に、Windows NT では ORACLE\_HOME¥network¥admin に置きます。

### 関連項目：

- cman.ora ファイルの詳細は、C-65 ページの「[Oracle Connection Manager パラメータ \(cman.ora\)](#)」を参照してください。
- [付録 B](#)（アドレス情報に必要なプロトコル構文に関する詳細）を参照してください。

---

**注意：** Oracle Net8 Assistant では cman.ora ファイルの構成をサポートしていないので、変更は手動で行う必要があります。

---



作業 2: クライアントの構成

クライアント・サポートを実現するには 2 つのアドレスを作成します。このうち最初のアドレスは Oracle Connection Manager、2 番目はリスナーのアドレスです。次の表に示すように、構成はネーミング・メソッドによって異なります。

ネーミング・メソッド	手順
ローカル・ネーミングおよびディレクトリ・ネーミング	<p>アドレス・リストを作成します。リストの最初のアドレスは Oracle Connection Manager、2 番目はリスナーのアドレスです。最初にクライアントが Oracle Connection Manager に接続し、次に Oracle Connection Manager がリスナーに接続するようにするには、SOURCE_ROUTE パラメータを設定します。</p> <p><b>関連項目：</b> Oracle Connection Manager およびリスナーのアドレスの構成方法は、8-33 ページの「<a href="#">Oracle Connection Manager を使用する場合のクライアントの構成</a>」を参照してください。</p>
Oracle Names	<p>Oracle Names サーバーではリスナーおよび Oracle Connection Manager のアドレスは自動的に更新されます。</p> <p>Oracle Names サーバーを使用するクライアントから Oracle Connection Manager に要求を送信するには、各 Oracle Names サーバーが存在するマシンおよびクライアント上の sqlnet.ora ファイルに USE_CMAN=TRUE パラメータを設定します。このパラメータを設定するには、8-18 ページの「<a href="#">接続要求のルーティング</a>」にある手順に従います。</p>

例 8-2 は、SPX を使用するクライアントが Oracle Connection Manager に接続し、Oracle Connection Manager が TCP/IP を使用してデータベース・サーバーに接続することを示しています。

例 8-2 複数プロトコル・サポートを使用するように構成された CMAN パラメータ

```
cman=
  (description=
    (source_route=yes)

    (address=
      (protocol=spx)
      (service=cman))
  (address=
    (protocol=tcp)
    (host=sales-pc)
    (port=1521))
  (connect_data=
    (service_name=sales.com)))
```



tnsnames.ora ファイルの要素	説明
source_route=yes	Oracle Connection Manager から接続先データベースまでのアドレスのソース・ルートを作成します。
(address= (protocol=spx) (service=cman))	最初のアドレスは、SPX を使用するクライアントから Oracle Connection Manager までです。ここから、Oracle Connection Manager は TCP/IP を使用するデータベース・サービスにリスナーを通じて接続します。

## Net8 アクセス制御の使用

Oracle Connection Manager には、TCP/IP 環境で指定されたサーバーへのクライアント・アクセスを制御するときに使用できる機能も含まれています。サーバーへの特定のクライアント・アクセスを可能にするか、または制限するフィルタリング・ルールを設定できます。

この機能を使用するかどうかは、接続集中化および複数プロトコル・サポート機能でフィルタリングを行うかどうかによって依存します。

ネットワーク上にリリース 8.1 のサービスが存在する場合、リリース 8.1 の Oracle Connection Manager がなければこの機能は利用できません。

Net8 アクセス制御を構成するには、次の作業を行います。

作業 1: [Oracle Connection Manager の構成](#)

作業 2: [クライアントの構成](#)

### 作業 1: Oracle Connection Manager の構成

Oracle Connection Manager を構成する手順は、次のとおりです。

1. cman.ora ファイルがない場合は手動で作成します。Oracle Net8 Assistant では、このファイルの構成をサポートしていません。

cman.ora ファイルは UNIX では \$ORACLE\_HOME/network/admin に、Windows NT では ORACLE\_HOME¥network¥admin に置きます。

**関連項目：**

- cman.ora ファイルの詳細は、C-65 ページの「[Oracle Connection Manager パラメータ \(cman.ora\)](#)」を参照してください。
  - アドレス情報に必要なプロトコル構文の詳細は、[付録 B](#) を参照してください。
2. クライアント接続でデフォルトのリスニング・ポート 1630 を使用しない場合、次のエントリを追加します。  
  
cman=(address=(protocol=tcp) (host=anyhost) (port=port))



3. 管理コマンドでデフォルトのリスニング・ポート 1830 を使用しない場合、次のエントリを追加します。

```
cman=(address=(protocol=tcp) (host=anyhost) (port=port) )
```

4. CMAN\_RULES パラメータを追加して次のパラメータを指定します。

```
cman_rules=
(rule_list=
(rule=
(src=shost)
(dst=dhost)
(srv=service)
(act=accept | reject)))
```

パラメータ	説明
SRC	ソースのホスト名または IP アドレス（クライアント）
DST	接続先のホスト名または IP アドレス（サーバー）
SRV	リリース 8.1 データベースのサービス名（ <a href="#">初期化パラメータ・ファイル (initialization parameter file)</a> の SERVICE_NAMES パラメータから取得）、またはリリース 8.1 以前のデータベースの SID 値（ORACLE_SID 環境変数またはレジストリ値から取得）
ACT	上の 3 つのパラメータに基づく受信要求の受け入れまたは拒否

RULE\_LIST 内で、複数の RULE を定義することもできます。最初に適合した RULE のアクション（ACT）が要求に適用されます。CMAN\_RULES が未定義の場合は、すべての接続が許可されます。

次の例は、クライアント・マシン client1-pc のサービス sales.us.acme.com に対する制限、およびクライアント 144.25.23.45 のサービス db1 へのアクセスの許可を示します。

```
cman_rules=
(rule_list=
(rule=(src=client1-pc) (dst=sales-pc) (srv=sales.us.acme.com) (act=reject))
(rule=(src=144.25.23.45) (dst=144.25.187.200) (srv=db1) (act=accept)))
```



## 作業 2: クライアントの構成

クライアント・サポートを実現するには 2 つのアドレスを作成します。このうち最初のアドレスは Oracle Connection Manager、2 番目はリスナーのアドレスです。次の表に示すように、構成はネーミング・メソッドによって異なります。

ネーミング・メソッド	手順
ローカル・ネーミングおよびディレクトリ・ネーミング	<p>アドレス・リストを作成します。リストの最初のアドレスは Oracle Connection Manager、2 番目はリスナーのアドレスです。最初にクライアントが Oracle Connection Manager に接続し、次に Oracle Connection Manager がリスナーに接続するようにするには、SOURCE_ROUTE パラメータを設定します。</p> <p><b>関連項目：</b> Oracle Connection Manager およびリスナーのアドレスの構成方法は、8-33 ページの「<a href="#">Oracle Connection Manager を使用する場合のクライアントの構成</a>」を参照してください。</p>
Oracle Names	<p>Oracle Names サーバーではリスナーおよび Oracle Connection Manager のアドレスは自動的に更新されます。</p> <p>Oracle Names サーバーを使用するクライアントから Oracle Connection Manager に要求を送信するには、各 Oracle Names サーバーが存在するマシンおよびクライアント上の sqlnet.ora ファイルに USE_CMAN=TRUE パラメータを設定します。このパラメータを設定するには、8-18 ページの「<a href="#">接続要求のルーティング</a>」にある手順に従います。</p>

## Oracle Connection Manager を使用する場合のクライアントの構成

**注意：** 接続パスで複数の Oracle Connection Manager が使用されている場合は、Oracle Connection Manager を通じて接続するためにクライアントが Oracle Names を使用することはできません。

Oracle Connection Manager を介してクライアント接続要求をデータベース・サーバーに集中させる場合、クライアントの構成でそのルートを設定する必要があります。これを実現するには、Oracle Connection Manager およびリスナーのプロトコル・アドレスを tnsnames.ora ファイル、ディレクトリまたは Oracle Names サーバーに設定します。次の点に注意してください。

- Oracle Names サーバーを使用する場合、Oracle Names サーバー内のアドレスに Oracle Connection Manager のアドレスを挿入することで、Oracle Connection Manager によりアドレスが自動的に更新されます。Oracle Names サーバーから Oracle Connection



Manager に要求を送信するには、Oracle Names サーバーが存在するマシン上の `sqlnet.ora` ファイルに `USE_CMAN=TRUE` を設定します。このパラメータを設定するには、8-18 ページの「[接続要求のルーティング](#)」にある手順に従います。

- `tnsnames.ora` ファイルまたはディレクトリを使用する場合は、プロトコル・アドレスと `SOURCE_ROUTE` パラメータを設定する必要があります。Oracle Connection Manager 用にローカルまたはディレクトリ・ネーミング・メソッドを構成するには、次の作業を行います。

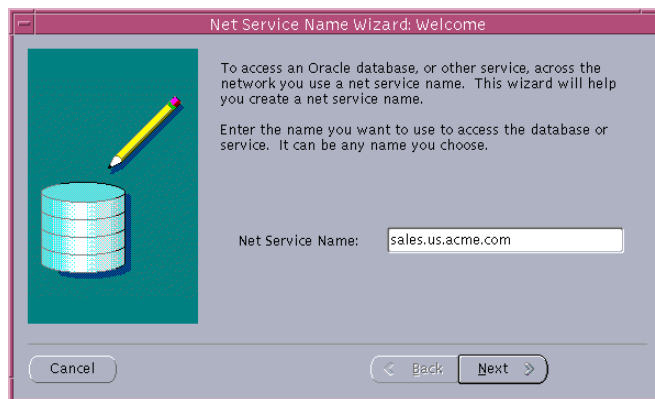
作業 1: Oracle Connection Manager アドレスの構成

作業 2: リスナー・アドレスの構成

## 作業 1: Oracle Connection Manager アドレスの構成

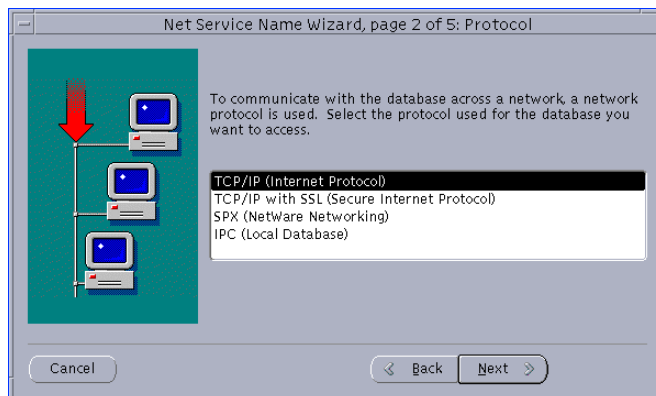
Oracle Connection Manager アドレスを構成するには、次の手順に従います。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、`$ORACLE_HOME/bin` から `netasst` を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Directory」または「Local」を展開します。次に「Service Naming」を選択します。
3. ツールバーの「+」ボタンをクリックするか、「Edit」メニューの「Create」を選択します。  
「Net Service Name Wizard」が起動します。
4. 「Net Service Name」フィールドに名前を入力し、「Next」をクリックします。

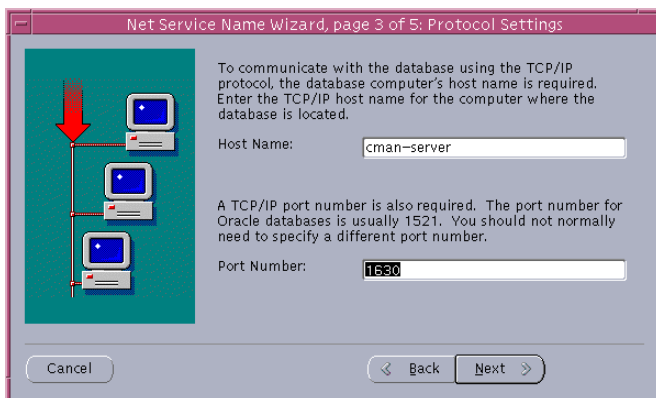




5. Oracle Connection Manager がリスニングに使用するプロトコルを選択し、「Next」をクリックします。デフォルトのプロトコルは TCP/IP です。



6. Oracle Connection Manager のネットワーク・プロトコル情報を入力し、「Next」をクリックします。TCP/IP を使用する場合のデフォルト・ポートは 1630 です。



**関連項目：** プロトコル・パラメータ設定の詳細は、[付録 B](#) を参照してください。

7. リリースを選択し、宛先サービスを入力して「Next」をクリックします。





宛先サービスが Oracle リリース 8.1 の場合は、「(Oracle8i)」をクリックし、「Service Name」フィールドにサービス名を入力します。宛先サービスが Oracle リリース 8.0 またはバージョン 7 のデータベースの場合は、「(Oracle8 or Previous)」をクリックし、「Database SID」フィールドに Oracle システム識別子 (SID) を入力します。

**関連項目：** サービス名文字列の設定に関する詳細は、6-2 ページの「[接続記述子の理解](#)」を参照してください。

8. 「Next」をクリックします。

---

**注意：** この時点では接続のテストはできないので、「Test」はクリックしないでください。

---

9. 「Finish」をクリックして構成を保存し、Net Service Name Wizard を終了します。

新規のネット・サービス名と Oracle Connection Manager アドレスが「Service Naming」フォルダに追加されます。

## 作業 2: リスナー・アドレスの構成

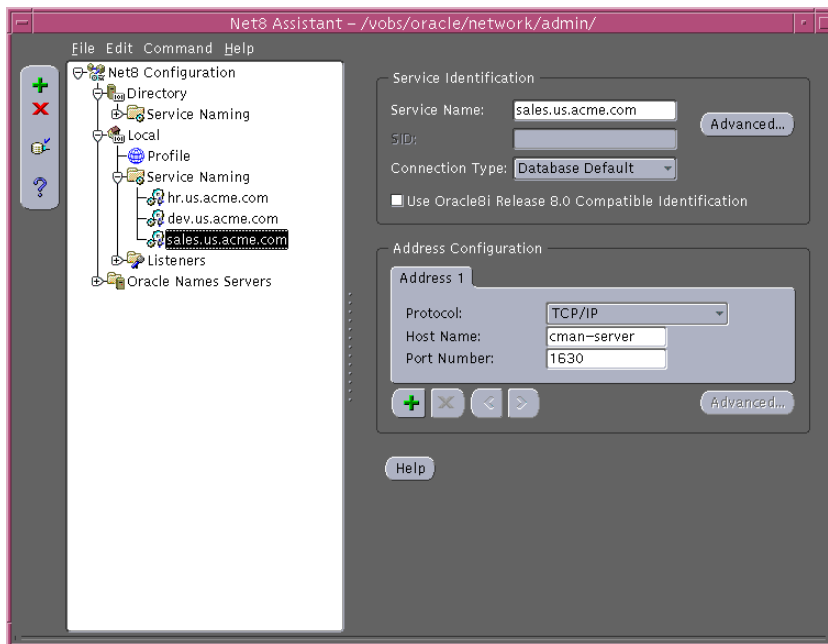
Oracle Connection Manager アドレスを指定ののち、次に、Oracle Connection Manager がサーバーと接続できるようにリスナーのアドレスを作成します。

リスナー・アドレスを構成するには、次の手順に従います。

1. 8-34 ページの「[作業 1: Oracle Connection Manager アドレスの構成](#)」で作成したネット・サービス名をクリックします。

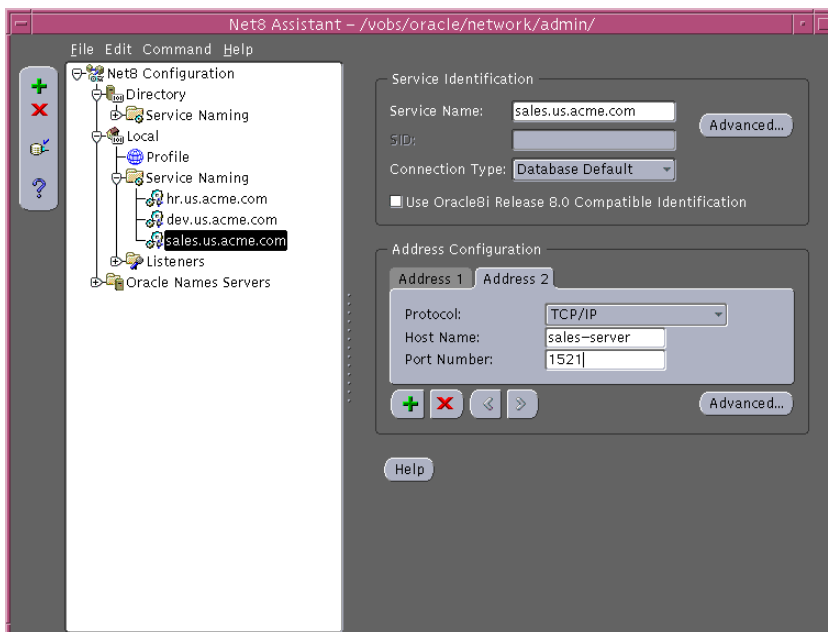


「Address 1」タブには Oracle Connection Manager アドレスが表示されることに注意してください。



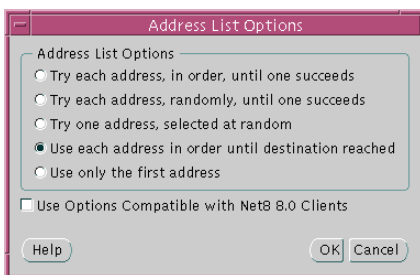
2. 「+」をクリックします。  
新規のアドレス・タブが表示されます。
3. プロトコルを選択し、リスナーのプロトコル情報を入力します。TCP/IP を使用している場合、デフォルト・ポート番号は 1521 です。





**関連項目：** プロトコル・パラメータ設定の詳細は、[付録 B](#) を参照してください。

4. 「Address Configuration」ボックスの「Advanced」ボタンをクリックします。  
「Address List Options」ダイアログ・ボックスが表示されます。



5. 「Use each address in order until destination reached」を選択して、「OK」をクリックします。



このオプションを選択すると、SOURCE\_ROUTE=ON が設定されます。これにより、クライアントは最初のアドレス（Oracle Connection Manager）に接続し、最初のアドレスから 2 番目のアドレス（リスナー）に接続します。

---

---

**注意：** 同一アドレス・リスト内での Oracle Connection Manager ソース・ルーティング機能には、クライアント・ロード・バランスや接続時フェイルオーバーをサポートするオプションはありません。SOURCE\_ROUTE を設定すると、最初のアドレスはクライアントの Oracle Connection Manager への接続で使用され、2 番目のアドレスは Oracle Connection Manager のリスナーへの接続で使用されます。SOURCE\_ROUTE が設定されている場合、クライアント・ロード・バランスと接続時フェイルオーバーは使用禁止です。

複数アドレス・リストのソース・ルーティングの場合は、クライアント・ロード・バランスや接続時フェイルオーバーを行うように手動で構成できます。C-35 ページの「例」を参照してください。

---

---

6. 右ペインで「Apply」をクリックします。

## 非 Oracle データベース・サービスへの接続の構成

次に示す項では、非 Oracle データベース・サービスへの接続の構成方法について説明します。

- [外部プロシージャを使用する場合の Net8 の構成](#)
- [Oracle 異種サービスを使用する場合の Net8 の構成](#)
- [Oracle Rdb データベースを使用する場合の Net8 の構成](#)

### 外部プロシージャを使用する場合の Net8 の構成

外部プロシージャとは、PL/SQL コードからコールできる、第三世代言語（3GL）で記述された関数またはプロシージャです。

外部プロシージャ・コールをリスニングするようにリスナーを構成できます。PL/SQL または SQL アプリケーションが外部プロシージャをコールすると、リスナーは extproc というネットワーク・セッション固有のプロセスを起動します。リスナー・サービスを介して、PL/SQL から extproc に次の情報が渡されます。

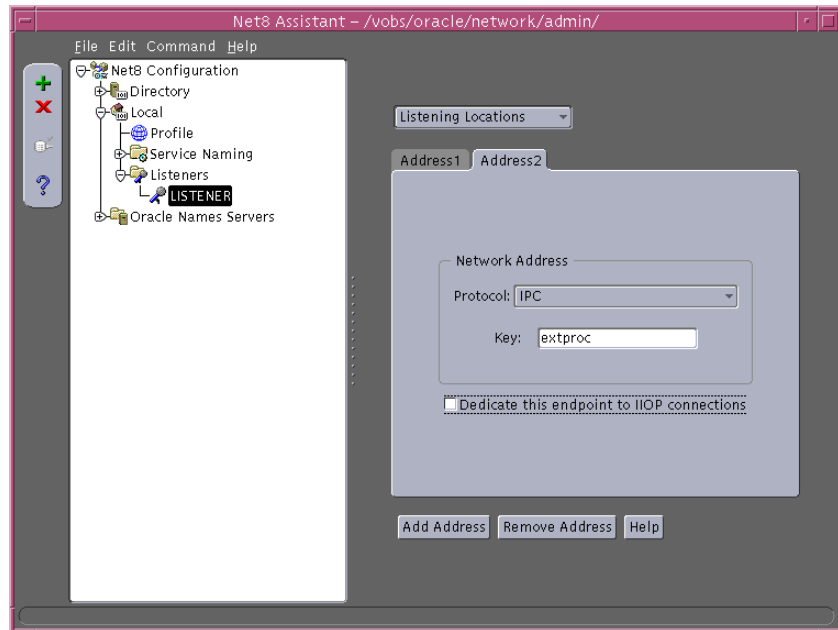
- 共有ライブラリ名
- 外部プロシージャ名
- パラメータ（必要な場合）

次に、extproc により共有ライブラリがロードされ、外部プロシージャがコールされます。



デフォルトでは、サーバー・インストール時に Net8 Configuration Assistant によって外部プロシージャへの接続が構成されます。接続情報が存在しない場合は、listener.ora ファイルと tnsnames.ora ファイルを編集します。これらのファイルは、UNIX では \$ORACLE\_HOME/network/admin、Windows プラットフォームでは ORACLE\_HOME\network\admin にあります。編集方法は次のとおりです。

1. listener.ora ファイルに TCP/IP または IPC のリスニング・アドレスを構成します。次に示す手順は、IPC アドレスの作成方法を説明しています。
  - a. Oracle Net8 Assistant を起動します。
    - UNIX の場合は、\$ORACLE\_HOME/bin から netasst を実行します。
    - Windows NT の場合は、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
  - b. ナビゲータで、「Local」->「Listeners」を展開します。
  - c. リスナーを選択します。
  - d. 右ペインのリストから「Listening Locations」を選択します。
  - e. 「Add Address」をクリックします。新規のアドレス・タブが表示されます。





- f. 「Protocol」リストから「IPC」を選択し、「KEY」に値を入力します。オラクル社では、KEY 値として `extproc` をお勧めします。

---

**注意：** 使用しているマシンに Oracle ホームやリスナーが複数ある場合は、各リスナーに一意的な KEY を指定してください。オラクル社では、第 1 リスナーは `extproc1`、第 2 リスナーは `extproc2` とすることをお勧めします。

---

**関連項目：** リスナー・プロトコル・アドレスの構成に関する詳細は、7-5 ページの「[リスナー・プロトコル・アドレスの構成](#)」を参照してください。

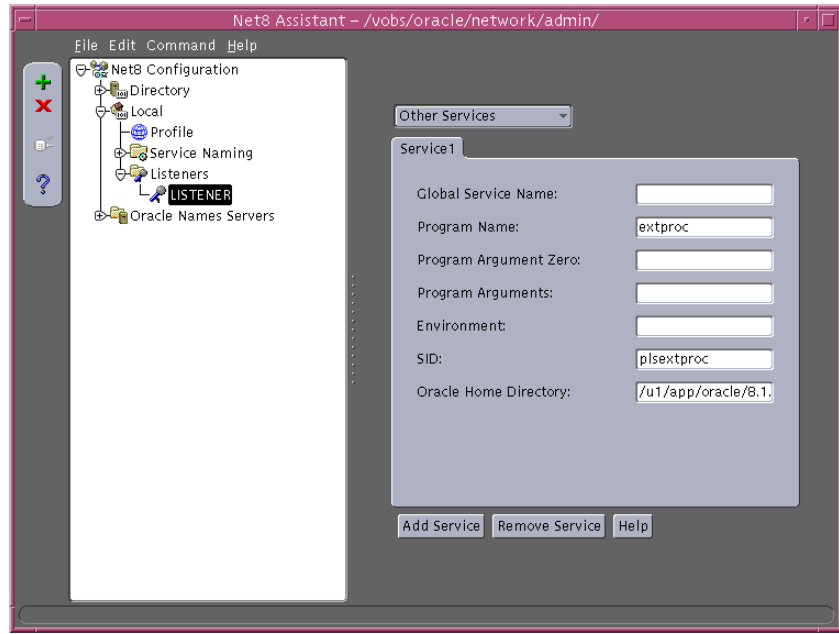
2. `extproc` に関するサービス情報として、次のパラメータを `listener.ora` ファイルに追加します。

listener.ora ファイルのパラメータ	Oracle Net8 Assistant のフィールド	説明
SID_NAME	「SID」	非 Oracle データベースの Oracle システム識別子 (SID) を指定します。
ORACLE_HOME	Oracle ホーム・ディレクトリ	<code>extproc</code> プログラムの Oracle ホームの場所を指定します。
PROGRAM	「Program Name」	<code>extproc</code> 実行可能プログラムの名前を指定します。

SID\_NAME、ORACLE\_HOME および PROGRAM パラメータを構成するには、次の手順に従います。

- a. 右ペインのリストから「Other Services」を選択します。
- b. 「Add Service」をクリックします。新規の「Service」タブが表示されます。





- c. 「Program Name」フィールドに extproc、「SID」フィールドに SID (plsextproc など)、「Oracle Home」フィールドに extproc 実行可能プログラムがある Oracle ホームをそれぞれ入力します。
- d. 「File」->「Save Network Configuration」を選択します。

listener.ora ファイルの外部プロシージャは次のように更新されます。

```
listener=
  (address=
    (protocol=ipc)
    (key=extproc))
sid_list_listener
  (sid_list=
    (sid_desc=
      (sid_name=plsextproc)
      (oracle_home=/u1/app/oracle/8.1.6)
      (program=extproc)))
```

- 3. listener.ora ファイルに構成された情報と同じネット・サービス名をサーバーの tnsnames.ora ファイルに作成します。
- 4. ステップ 2 で示した listener.ora ファイルに基づいて、tnsnames.ora ファイルに次のエントリを構成します。



```
extproc_connection_data=
(description=
(address=(protocol=ipc) (key=extproc))
(connect_data=
(sid=plsextproc)))
```

**関連項目：** ネット・サービス名の作成方法の詳細は、6-4 ページの「[ローカル・ネーミング・メソッドの構成](#)」を参照してください。

## Oracle 異種サービスを使用する場合の Net8 の構成

異種サービスは、Oracle サーバー内に統合されたコンポーネントで、Oracle サーバーから非 Oracle システムにアクセスする汎用的なテクノロジーを提供するものです。異種サービスを使用すると、次のことが可能になります。

- Oracle SQL を使用して、Oracle サーバーに存在するデータと同じように、非 Oracle システムに格納されているデータに透過的にアクセスすること
- Oracle プロシージャ・コールを使用して、Oracle 分散環境から非 Oracle システム、サービスまたはアプリケーション・プログラミング・インタフェース（API）に透過的にアクセスすること

異種サービスは Oracle サーバーで汎用的なテクノロジーを提供しますが、特定の非 Oracle システムにアクセスするには異種サービス・エージェントが必要です。

非 Oracle システムへの接続を開始する際、Oracle サーバーはゲートウェイ上のリスナーを介してエージェント・プロセスを起動します。Oracle サーバーがエージェントに接続できるようにするには、次の手順を実行します。

1. ゲートウェイ上のリスナーが、Oracle サーバーから受信した要求をリスニングし、異種サービス・エージェントを起動するように構成します。このために、listener.ora ファイルに次のパラメータを構成します。

listener.ora ファイルのパラメータ	Oracle Net8 Assistant のフィールド	説明
SID_NAME	「SID」	Oracle システム識別子（SID）を指定します。
ORACLE_HOME	「Oracle Home」	実行可能エージェントの Oracle ホームの場所を指定します。
PROGRAM	「Program Name」	実行可能エージェントの名前を指定します。

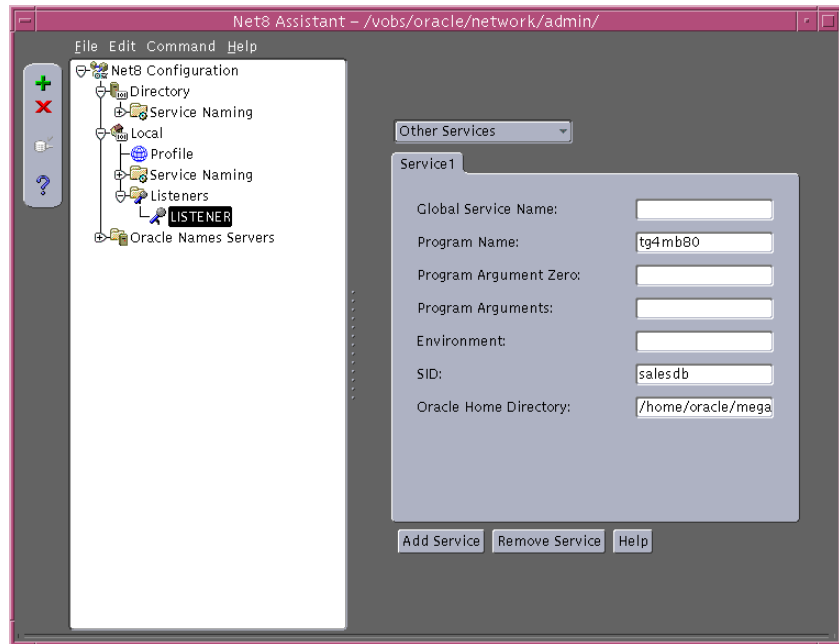
SID、ORACLE\_HOME および PROGRAM パラメータを構成するには、次の手順に従います。

- a. Oracle Net8 Assistant を起動します。
  - UNIX の場合は、\$ORACLE\_HOME/bin から netasst を実行します。



- Windows NT の場合は、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。

- b. ナビゲータで、「Local」 -> 「Listeners」を展開します。
- c. リスナーを選択します。
- d. 右ペインのリストから「Other Services」を選択します。
- e. 「Add Service」をクリックします。新規の「Service」タブが表示されます。



- f. ゲートウェイの作成を実行するプログラム名を「Program Names」フィールドに、実行可能エージェントが存在する Oracle ホームを「Oracle Home」フィールドに、非 Oracle システムの SID またはサービス名を「SID」フィールドにそれぞれ入力します。
- g. 「File」 -> 「Save Network Configuration」を選択します。

listener.ora ファイルの異種サービスに関する情報は次のように更新されます。

```
sid_list_listener=  
(sid_list=  
  (sid_desc=  
    (sid_name=salesdb)
```



```
(oracle_home=/home/oracle/megabase/8.1.6)
(program=tg4mb80))
```

**関連項目：** 『Oracle8i 分散システム』を参照してください。

2. Oracle データベースが存在するマシンで、ゲートウェイ上のリスナーに接続するネット・サービス名を設定します。接続で異種サービスを利用できるように、接続記述子にも HS=OK 句を設定する必要があります。

- a. Oracle Net8 Assistant を起動します。

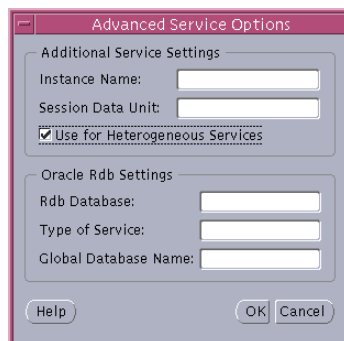
-UNIX の場合は、\$ORACLE\_HOME/bin から netasst を実行します。

-Windows NT の場合は、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。

- b. Oracle サーバーから非 Oracle システムへの接続に使用するネット・サービス名を作成します。

**関連項目：** 6-6 ページの「[作業 1: ネット・サービス名の構成](#)」(ローカル・ネーミングのネット・サービス名作成方法)を参照してください。

- c. ナビゲータで、「Local」 -> 「Service Naming」を展開します。
- d. ネット・サービス名を選択します。
- e. 「Service Identification」グループ・ボックスの「Advanced」をクリックします。  
「Advanced Service Options」ダイアログ・ボックスが表示されます。



- f. 「Use for Heterogeneous Services」をクリックし、「OK」をクリックします。
- g. 「File」 -> 「Save Network Configuration」を選択します。



tnsnames.ora ファイルでは、異種サービス用に構成された新規ネット・サービス名が次のように更新されます。

```
megabase6_sales=
(description=
  (address=(protocol=tcp) (host=dlsun206) (port=1521))
(connect_data=
  (service_name=sales6)
  (hs=ok)))
```

Oracle Rdb データベースを使用する場合の Net8 の構成

Oracle Rdb は、Digital の 64 ビット・プラットフォームで使用するデータベースです。Net8 を使用する場合、Oracle Rdb サーバーはクライアントにとって Oracle8i データベースと同じように見えます。Oracle Rdb には独自のリスナーがあるので、クライアントは Oracle7 との通信と同じように Oracle Rdb と通信できます。

Oracle Rdb への接続を開始するには、次の表に示すパラメータを使用して、Oracle Rdb データベースに接続するネット・サービス名を設定します。

tnsnames.ora ファイルの パラメータ	Oracle Net8 Assistant のフィールド	説明
RDB_DATABASE	「Rdb Database」	Oracle Rdb データベースのファイル名を指定 します。このパラメータは CONNECT_DATA セクションに記述します。
TYPE_OF_SERVICE	「Type of Service」	Oracle Rdb データベースで使用するサービスの 種類を指定します。このサービスは Rdb イン タフェース・ツールで使用されます。この機能 は、アプリケーションが Oracle Rdb と Oracle データベースの両方をサポートし、アプリケー ションを任意に選択する（負荷のバランスをと る）場合にのみ使用します。このパラメータは DESCRIPTION セクションに記述します。
GLOBAL_NAME	「Global Database Name」	Oracle Rdb データベースを識別します。省略 可能な機能です。このパラメータは CONNECT_DATA セクションに記述します。

**関連項目：** Oracle Rdb のマニュアルを参照してください。

Oracle Rdb データベースを使用するようにクライアントを構成するには、Oracle Net8 Assistant を使用します。

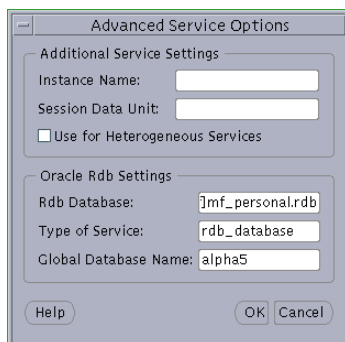
- 1. Oracle Net8 Assistant を起動します。



- UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. Oracle8i サーバーから非 Oracle システムへの接続に使用するネット・サービス名を作成します。

**関連項目：** 6-6 ページの「[作業 1: ネット・サービス名の構成](#)」（ローカル・ネーミングのネット・サービス名作成方法）を参照してください。

3. ナビゲータで、「Directory」または「Local」 -> 「Service Naming」を展開します。
4. ネット・サービス名を選択します。
5. 「Service Identification」グループ・ボックスの「Advanced」をクリックします。  
「Advanced Service Options」ダイアログ・ボックスが表示されます。



6. 「Rdb Database」フィールドに Oracle Rdb データベースのファイル名を入力します。
7. オプションの「Global Database Name」フィールドにグローバル・データベース名を入力し、必要の場合は「Type of Service」フィールドにサービスの種類を指定して、「OK」をクリックします。
8. 「File」 -> 「Save Network Configuration」を選択します。

tnsnames.ora ファイルでは、Oracle Rdb データベースのために構成された新規ネット・サービス名が次のように更新されます。

```
alpha5=
  (description=
    (address=...)
    (connect_data=
      (service_name=generic)
      (rdb_database=[.mf]mf_personnel.rdb)
```



```
(global_name=alpha5)))
```

次の例では、TYPE\_OF\_SERVICE を使用して、Oracle Rdb データベース・サービスや Oracle データベース・サービスのロード・バランスをとるように構成されています。

```
alpha5=
(description_list=
(description=
(address=...)
(connect_data=
(service_name=generic)
(rbd_database=[.mf]mf_personnel.rdb)
(global_name=alpha5)))
(description=
(address=...)
(connect_data=
(service_name=sales.com))
(type_of_service=oracle8_database))
```

**関連項目：** Oracle Rdb のマニュアルを参照してください。



---

## マルチスレッド・サーバーの構成

この章では、**マルチスレッド・サーバー (multi-threaded server: MTS)** の構成方法について説明します。内容は次のとおりです。

- 概要
- **MTS\_DISPATCHERS** パラメータを使用した MTS の構成
- 接続プーリングの使用
- リソースの割当て
- クライアントでの MTS の使用
- クライアントでの MTS の変更

### 関連項目：

MTS 構成の詳細は、『Oracle8i 管理者ガイド』を参照してください。

MTS パラメータのチューニング方法の詳細は、『Oracle8i パフォーマンスのための設計およびチューニング』を参照してください。



# 概要

専用サーバー・アーキテクチャを使用する注文入力システムを考えます。顧客の注文を店員がデータベースに入力します。トランザクションの大部分の間、店員は電話で顧客と話し、店員のユーザー処理専用のサーバー・プロセスはアイドル状態です。サーバー・プロセスはトランザクションの大部分の間は不要です。サーバー・プロセスがシステム・リソースを保持しているので、注文を入力している店員以外がシステムを使用しても速く動きません。

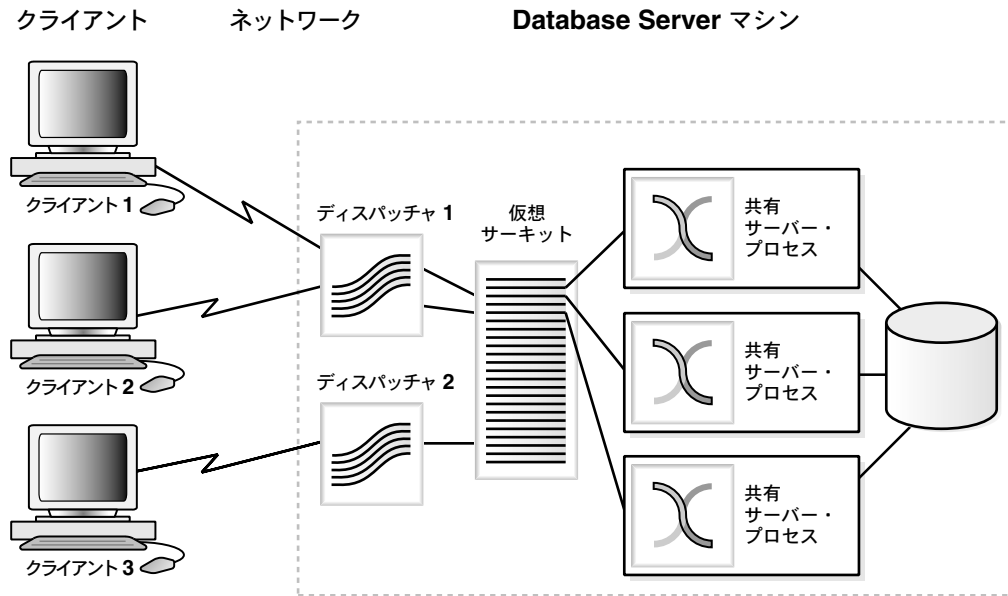
MTS アーキテクチャでは、接続のたびに専用サーバー・プロセスを使用する必要はありません（[図 9-1](#) を参照してください）。ディスパッチャは、受信した複数のネットワーク・セッション要求を共有サーバー・プロセス・プールに送信します。共有サーバー・プロセス・プールの中でアイドル状態にある共有サーバー・プロセスは、共通キューから要求を取り出します。つまり、少数の共有サーバーで多数の専用サーバーと同じ量の処理を実行できます。また、各ユーザーに必要なメモリー量も比較的小さいので、より少ないメモリーとプロセス管理で、より多くのユーザーをサポートできます。

MTS の利点は、システムのオーバーヘッドを減らし、使用するリソース量を削減しながら、サポートするユーザーの数を増加させられることです。

**関連項目：** 2-6 ページの「[マルチスレッド・サーバー・モデル](#)」を参照してください。



図 9-1 MTS アーキテクチャ



**関連項目：** 2-6 ページの「マルチスレッド・サーバー・モデル」を参照してください。

## MTS\_DISPATCHERS パラメータを使用した MTS の構成

MTS 構成を使用可能にするには、[初期化パラメータ・ファイル \(initialization parameter file\)](#) に MTS\_DISPATCHERS パラメータを設定します。

---

**注意：** Oracle Database Configuration Assistant を使用してこのパラメータを構成します。

---

このパラメータを設定した後、MTS 構成を使用可能にするためにインスタンスを再起動します。

MTS\_DISPATCHERS は次の書式で設定します。

```
mts_dispatchers="(attribute=value)"
```



MTS を使用可能にするには、[ADDRESS](#) ([ADD](#) または [ADDR](#))、[DESCRIPTION](#) ([DES](#) または [DESC](#))、[PROTOCOL](#) ([PRO](#) または [PROT](#)) のいずれかの属性を設定する必要があります。

属性	説明
<a href="#">ADDRESS</a> ( <a href="#">ADD</a> または <a href="#">ADDR</a> )	ディスパッチャのリスニング対象の終点となるネットワーク・プロトコル・アドレスを指定します。
<a href="#">DESCRIPTION</a> ( <a href="#">DES</a> または <a href="#">DESC</a> )	ディスパッチャのリスニング対象の終点となるネットワーク設定を指定します。この指定にはネットワーク・プロトコル・アドレスも含まれます。構文は次のとおりです。  (description=(address=...))
<a href="#">PROTOCOL</a> ( <a href="#">PRO</a> または <a href="#">PROT</a> )	ディスパッチャがリスニング終点を生成するときに使用するネットワーク・プロトコルを指定します。たとえば、次のように指定します。  (protocol=tcp)  <b>関連項目：</b> プロトコル・アドレス構文の詳細は、 <a href="#">付録 B</a> を参照してください。

属性 [CONNECTIONS](#) ([CON](#) または [CONN](#))、[DISPATCHERS](#) ([DIS](#) または [DISP](#))、[LISTENER](#) ([LIS](#) または [LIST](#))、[MULTIPLEX](#) ([MUL](#) または [MULT](#))、[POOL](#)、[SERVICE](#) ([SER](#) または [SERV](#))、[SESSIONS](#) ([SES](#) または [SESS](#)) および [TICKS](#) はオプションです。

属性	説明
<a href="#">CONNECTIONS</a> ( <a href="#">CON</a> または <a href="#">CONN</a> )	各ディスパッチャで許容されるネットワーク接続の最大数を指定します。  デフォルト値は、オペレーティング・システム固有です。たとえば、Sun Solaris および Windows NT のデフォルト値は 1024 です。
<a href="#">DISPATCHERS</a> ( <a href="#">DIS</a> または <a href="#">DISP</a> )	最初に起動するディスパッチャ数を指定します。デフォルトは 1 です。



属性	説明
LISTENER (LIS または LIST)	<p><b>PMON プロセス (PMON process)</b> によってディスパッチャ情報が登録されるリスナーの別名を指定します。別名には、ネーミング・メソッドで解決される名前を設定してください。</p> <p>この属性は、次の場合にのみ指定する必要があります。</p> <ul style="list-style-type: none"> <li>■ リスナーが非デフォルト・アドレス (TCP/IP、ポート 1521 以外) を使用するローカル・リスナーです。</li> <li>■ リスナーが他のノードに存在します。</li> </ul> <p>デフォルト以外のローカル・リスナーは、LOCAL_LISTENER パラメータで指定することもできます。</p> <p><b>重要:</b> リスナー別名は、サーバーの tnsnames.ora ファイルや Oracle Names サーバーなどのネーミング・メソッドで解決される名前にしてください。</p> <p>たとえば、リスナー別名が listener_sales で、ポート 1521 に 2 つのリスニング終点があり、選択されたネーミング・メソッドがローカル・ネーミング・メソッドの場合、tnsnames.ora ファイルのエントリは次のようになります。</p> <pre>listener_sales= (description=   (address=(protocol=tcp) (host=sales1-server) (port= 1521))   (address=(protocol=tcp) (host=sales2-server) (port= 1521)))</pre>
MULTIPLEX (MUL または MULT)	<p>Oracle Connection Manager の <b>接続集中化 (connection concentration)</b> 機能を使用可能にします。</p> <p>1、ON、YES、TRUE または BOTH が指定されると、受信と送信の両方のネットワーク接続で接続集中化が使用可能です。</p> <p>IN が指定されると、クライアントから受信するネットワーク接続で接続集中化が使用可能です。</p> <p>OUT が指定されると、送信するネットワーク接続で接続集中化が使用可能です。</p> <p>0、NO、OFF または FALSE が指定されると、受信と送信の両方のネットワーク接続で接続集中化が使用禁止です。</p> <p><b>関連項目:</b> 8-25 ページの「<b>接続集中化機能の使用</b>」を参照してください。</p>



属性	説明
POOL	<p><b>接続プーリング (connection pooling)</b> を使用可能にします。</p> <p>数値が指定されると、受信と送信の両方のネットワーク接続に対して接続プーリングが使用可能になります。指定された数値は、受信および送信ネットワーク接続のタイムアウト・ティック数です。</p> <p>ON、YES、TRUE または BOTH が指定されると、受信と送信の両方のネットワーク接続に対して接続プーリングが使用可能になります。受信および送信ネットワーク接続のタイムアウトにはデフォルトの 10 ティックが使用されます。</p> <p>IN が指定されると、接続プーリングは受信ネットワーク接続に対して使用可能になります。受信ネットワーク接続のタイムアウトにはデフォルトの 10 ティックが使用されます。(IN=20) のように、IN にタイムアウト・ティック値を割り当てることもできます。指定したタイムアウトの数値が 0 または 1 の場合は、デフォルト値の 10 ティックが使用されます。</p> <p>OUT が指定されると、接続プーリングは送信ネットワーク接続に対して使用可能になります。送信ネットワーク接続のタイムアウトにはデフォルトの 10 ティックが使用されます。(OUT=20) のように、OUT にタイムアウト・ティック値を割り当てることもできます。指定したタイムアウトの数値が 0 または 1 の場合は、デフォルト値の 10 ティックが使用されます。</p> <p>NO、OFF または FALSE が指定されると、受信と送信の両方のネットワーク接続に対して接続プーリングは使用禁止になります。</p> <p><b>関連項目：</b> 9-8 ページの「<a href="#">接続プーリングの使用</a>」を参照してください。</p>
SERVICE (SER または SERV)	ディスパッチャがリスナーに登録するサービス名を指定します。値が指定されない場合は、SERVICE_NAMES パラメータで指定されたサービス名が使用されます。
SESSIONS (SES または SESS)	<p>各ディスパッチャで許容されるネットワーク・セッションの最大数。</p> <p>デフォルト値は、オペレーティング・システム固有です。ほとんどのオペレーティング・システムのデフォルトは 16K です。</p>
TICKS	<p>ネットワーク・ティックの秒数。設定された値に POOL タイムアウト値を乗算すれば、接続プーリング・タイムアウト時間が得られます。</p> <p>デフォルトは 15 秒です。高速ネットワークの場合の推奨ティック・サイズは 1 秒です。低速ネットワークの場合の推奨ティック・サイズは 3 ～ 4 秒です。</p>



## ディスパッチャの初期数の設定

インスタンス起動時に起動されるディスパッチャの数は、DISPATCHERS 属性で制御されます。

---

**注意：** 共有サーバーとは異なり、ディスパッチャ数は動的に変更しません。ディスパッチャ数は、ALTER SYSTEM コマンドを使用して明示的に変更する必要があります。この方法では、MTS\_MAX\_DISPATCHERS パラメータで指定された上限の範囲内でディスパッチャ数を変更できます。これらのパラメータの詳細は、『Oracle8i パフォーマンスのための設計およびチューニング』を参照してください。

---

各インスタンスに適切なディスパッチャ数は、データベースに期待するパフォーマンス、プロセス（オペレーティング・システムに依存）ごとの接続数に関するホストのオペレーティング・システムの制限、およびネットワーク・プロトコルごとに必要な接続数によって異なります。

### ディスパッチャの初期数の計算

オペレーティング・システムでプロセスごとに可能な接続数がわかれば、次の式を使用して、インスタンスの起動時に作成するディスパッチャの初期数をネットワーク・プロトコルごとに計算します。

$$\begin{array}{l} \text{number} \\ \text{of} \\ \text{dispatchers} \end{array} = \text{CEIL} \left( \frac{\text{maximum number of concurrent sessions}}{\text{connections per dispatcher}} \right)$$

CEIL は、数値を次に高位の整数にまるめる処理を表します。

例として、次のようなシステムを考えます。

- TCP/IP 経由では同時に 4000 ユーザーが接続し、プロセス当たり 1,000 の接続がサポートされます。
- SPX 経由では同時に 3000 ユーザーが接続し、プロセス当たり 1,000 の接続がサポートされます。

この場合、TCP/IP の DISPATCHERS 属性には 4 以上のディスパッチャ、SPX の DISPATCHERS 属性には 3 以上のディスパッチャを設定する必要があります。

```
mts_dispatchers="(protocol=tcp)(dispatchers=4)(connections=1000)"
mts_dispatchers="(protocol=spx)(dispatchers=3)(connections=1000)"
```

パフォーマンスによっては、ディスパッチャ数を調整する必要があります。



### ディスパッチャ・アドレスの設定

**例 1** ディスパッチャに使用する IP アドレスは、次のように設定します。

```
mts_dispatchers=" (address=(protocol=tcp) (host=144.25.16.201)
(dispatchers=2) "
```

これにより、HOST=144.25.16.201 でリスニングするディスパッチャが 2 つ起動します。Net8 では、ディスパッチャの TCP/IP ポートは動的に選択されます。

**例 2** ディスパッチャの正確な位置を設定するには、次のように PORT を追加します。

```
mts_dispatchers=" (address=(protocol=tcp)
(host=144.25.16.201) (port=5000)) (dispatchers=1) "
mts_dispatchers=" (address=(protocol=tcp)
(host=144.25.16.201) (port=5001)) (dispatchers=1) "
```

---

---

**注意：** 初期化ファイルには複数の MTS\_DISPATCHERS を指定できませんが、1 箇所にとまとめて記述する必要があります。

---

---

## 接続プーリングの使用

接続プーリングは、ディスパッチャへの物理ネットワーク接続数を削減することのできるリソース利用機能です。これは、クライアント・プロセス間で接続を共有またはプールすることにより実現されます。

接続プーリングを構成するには、初期化パラメータ・ファイルの MTS\_DISPATCHERS パラメータに POOL 属性および次に示すオプション属性を設定します。

- CONNECTIONS (CON または CONN)
- SESSIONS (SES または SESS)
- TICKS

9-7 ページの「**ディスパッチャの初期数の計算**」の例で、接続プーリングを使用すればディスパッチャ当たりの接続数を 1,000、TCP/IP のセッション数を 4,000、SPX のセッション数を 3,000 にすることができます。これにより、プロトコル当たりの構成が 1 ディスパッチャに削減されます。設定は次のとおりです。

```
mts_dispatchers=" (protocol=tcp) (dispatchers=1) (pool=on) (tick=1)
(connections=1000) (sessions=4000) "
mts_dispatchers=" (protocol=spx) (dispatchers=1) (pool=on) (tick=1) (connections=2000) "
```



## リソースの割当て

Oracle8i データベースは複数のサービス名で表すことができます。このため、MTS ディスパッチャ・プールを特定のサービスを要求するクライアントに排他的に割り当てることができます。これによりミッション・クリティカル要求に対してはより多くのリソースが割り当てられ、実質的にその要求の優先順位が高くなります。

たとえば、次の初期化パラメータ・ファイルの例では2つのディスパッチャがあります。最初のディスパッチャは、sales.us.acme.com を要求するクライアントの要求を処理します。もう一方のディスパッチャは、admsales.us.acme.com を要求するクライアントの要求のみ処理します。

```
service_names=sales.us.acme.com
instance_name=sales
mts_dispatchers="(protocol=tcp) "
mts_dispatchers="(protocol=tcp) (service=admsales.us.acme.com) "
```

## クライアントでの MTS の使用

MTS が構成され、ディスパッチャが未登録の状態ではクライアント接続要求が来た場合、その要求は専用サーバーで処理できます (listener.ora ファイルに構成します)。特定のクライアントが常にディスパッチャを使用するようにする場合は、接続記述子の接続データ部分に (SERVER=SHARED) を構成します。たとえば、次のように指定します。

```
sales=
(description=
  (address=(protocol=tcp) (host=sales-server) (port= 1521))
  (connect_data=
    (service_name=sales.us.acme.com)
    (server=shared)))
```

ディスパッチャが使用不可能な場合、クライアント接続要求は拒否されます。

**関連項目：** SERVER パラメータの設定方法は、8-8 ページの「[拡張接続データ・パラメータの構成](#)」を参照してください。

## クライアントでの MTS の変更

データベースが MTS として構成されていて、特定のクライアントに専用サーバーが必要な場合、そのクライアントについて専用サーバーを使用するように構成することができます。それには次の方法があります。

- ネット・サービス名の接続記述子で、CONNECT\_DATA セクションに (SERVER=DEDICATED) を入れます。たとえば、次のように指定します。

```
sales=
```



```
(description=
  (address=(protocol=tcp) (host=sales-server) (port= 1521))
  (connect_data=
    (service_name=sales.us.acme.com)
    (server=dedicated)))
```

- クライアント・プロファイル (sqlnet.ora ファイル) に USE\_DEDICATED\_SERVER=ON と指定します。これにより、クライアントが使用する接続記述子の CONNECT\_DATA セクションに (SERVER=DEDICATED) が追加されます。たとえば、次のように指定します。

```
use_dedicated_server=on
```

---

---

**注意：** USE\_DEDICATED\_SERVER が ON に設定されると、既存の (SERVER=*value*) エントリは (SERVER=DEDICATED) に上書きされます。

---

---

### 関連項目：

- SERVER パラメータの設定方法は、8-8 ページの「[拡張接続データ・パラメータの構成](#)」を参照してください。
- USE\_DEDICATED\_SERVER パラメータの設定方法は、8-18 ページの「[接続要求のルーティング](#)」を参照してください。



---

## プログラマのための Net8 拡張機能

Net8 には、Net8 OPEN と呼ばれるアプリケーション・プログラム・インタフェース (API) があり、このインタフェースでプログラマはデータベース・アプリケーションとデータベース以外のアプリケーションの両方を開発できます。また Net8 には、UNIX クライアント・プログラミングにおいてシグナル・ハンドラとアラーム・プログラミング、bequeath プロトコル・サポートおよび子プロセスの終了など、プログラマにとっていくつかの利点があります。内容は次のとおりです。

- [Net8 OPEN](#)
- [UNIX クライアント・プログラミング](#)



# Net8 OPEN

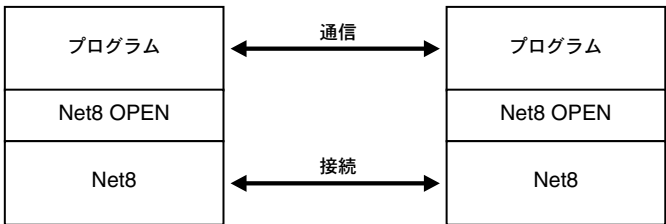
Net8 には、Net8 OPEN と呼ばれるアプリケーション・プログラム・インタフェース（API）があり、このインタフェースでプログラマは次のことを行うことができます。

- 環境内ですでに採用されている Net8 ネットワークを使用するデータベース・アプリケーションとデータベース以外のアプリケーションの両方の開発
- ネットワーク・インタフェースへのコールを変更することなく 1 つのマシン上で開発されたアプリケーションの他のマシンでの実行

Net8 OPEN は、すべての業界標準ネットワーク・プロトコルに対して単一の共通インタフェースをアプリケーションに提供します。

Net8 OPEN と他の製品との関係を [図 10-1](#) に示します。

図 10-1 Net8 OPEN



Net8 OPEN を使用して、次に示す様々な問題を解決できます。

- 3 層接続（クライアント / エージェント / サーバー）  
アプリケーションを使用してエージェントと通信します。ここで エージェントは、たとえば Oracle または Oracle 以外のデータ・ソースへ同時接続するアプリケーション・サーバーです。
- 分散アプリケーション  
データベースまたは他のミドルウェアを必要とせずに、既存の Oracle ネットワーク上で実行できる分散アプリケーションを構築します。
- 機能拡張されたクライアント  
非 SQL 情報を SQL アプリケーションと統合します。たとえば、プロセス制御アプリケーションによるセンサーなどの非 SQL アプリケーションとの交信が可能です。



## Net8 OPEN API ファンクション・コール

リモート・プロシージャ・コール・インタフェースとは異なり、Net8 OPEN はデータを送受信する基本的なアプリケーションを開発するためのバイト・ストリーム指向の API を備えています。Net8 OPEN で開発されたアプリケーションでは、ネットワークを介して送信された値は、受信側で正しく解釈されることを保証する必要があります。

Net8 OPEN API は 5 つのファンクション・コールから構成されます。

- [TNSOPEN](#)
- [TNSCLOSE](#)
- [TNSSEND](#)
- [TNSRECV](#)
- [TNSCONTROL](#)

---

### TNSOPEN

---

説明:	Net8 OPEN API の接続ハンドルを初期化します。このファンクションは、ユーザーが行う最初の Net8 OPEN コールにしてください。tnsopen() では接続は確立されないことに注意してください。接続は、必要に応じて送信または受信コールによって確立されます。
構文:	<pre>int tnsopen(handlep, name) void **handlep; const char *name;</pre> <p>クライアント・プログラム（接続を初期化します）を記述する場合、name には tnsnames.ora ファイルと同じ形式のネット・サービス名を指定します。</p> <p>サーバー・プログラムを記述する場合、name は NULL にします。これによりサーバー・プログラムは、最初の tnsrecv() コールでデータを受信すると、自動的に接続をピックアップします。</p>
パラメータ:	handlep (IN/OUT): Net8 接続ハンドルを受信するためのアドレス name (IN): ネット・サービス名
前提条件:	handlep パラメータは、NULL にしないでください。
戻り値:	正常終了の場合、ゼロ値が戻ります。エラーが発生した場合、正の Net8 API エラー番号が戻されます。

---



TNSCLOSE	
説明:	接続を停止します。ユーザーは、接続をクローズし、ハンドルを正常に解放するためにこのファンクションをコールする必要があります。
構文:	<pre>int tnsclose(handlep) void **handlep;</pre>
パラメータ:	handlep (IN/OUT): Net8 接続ハンドルへのポインタのアドレス
前提条件:	handlep パラメータは、NULL にしないでください。
戻り値:	正常終了の場合、ゼロ値が戻り、*handlep に NULL が設定されます。エラーが発生した場合、正の Net8 API エラー番号が戻されます。

TNSSEND	
説明:	<p>Net8 接続ハンドルにデータを送信します。</p> <p>クライアント側で初めて tnssend() をコールすると、ハンドルにデータが送信される前に接続が確立されます。クライアント・アプリケーションは、サーバーとの接続を確立するために、tnsopen() に続くコールとして tnssend() をコールする必要があります。クライアント・アプリケーションが tnsrecv() を最初にコールしたり、サーバー・プログラムが tnssend() を最初にコールすることは正しくありません。</p> <p>したがって、tnssend() コールにより接続確立に関するエラーが戻る場合もあります。たとえば、不正な TNS アドレスを指定した場合、最初に発生するのは tnsopen() のエラーではなく、tnssend() コールのエラーです。</p>
構文:	<pre>int tnssend(handle, data, length) void *handle; const void *data; size_t *length;</pre>
パラメータ:	<p>handle (IN/OUT) - tnsopen() によって戻される Net8 接続ハンドルへのポインタ</p> <p>data (IN): 送信されるデータへのポインタ</p> <p>length (IN/OUT): 送信予定のデータの長さ (バイト単位) と実際に書き込まれたバイト数に対するポインタ</p>
前提条件:	パラメータは、NULL にしないでください。



---

**TNSSEND**

---

戻り値: 正常終了の場合、ゼロ値と、実際に書き込まれたバイト数が `length` パラメータによって指示された値として戻ります。エラーが発生した場合、正の Net8 API エラー番号が戻されます。

---

---

**TNSRCV**

---

説明: Net8 接続ハンドルからデータを受信します。

サーバー側での `tnsrecv()` に対する最初のコールで、ハンドルからデータを受信する前に接続が確立されます。サーバーは、クライアントからの接続を受諾するために、`tnsopen()` に続く最初のコールとして `tnsrecv()` をコールする必要があります。

着信する接続は Net8 リスナー (`tnslsr`) によって受諾され、必要に応じてサーバー・プログラムのコピーが自動的に起動され、それに新しい接続が渡されます。このような処理を行うには、ネットワーク・リスナーにサーバー・プログラムの場所を設定する必要があります。

構文: 

```
int tnsrecv(handle, data, length)
void *handle;
void *data;
size_t *length;
```

パラメータ: `handle` (IN/OUT): `tnsopen()` によって戻される Net8 接続ハンドルへのポインタ

`data` (IN/OUT): データを受信するバッファへのポインタ

`length` (IN/OUT): データを受信する予定のバッファの長さを実際に受信したバイト数に対するポインタ

前提条件: すべてのパラメータは、NULL にしないでください。

戻り値: 正常終了の場合、ゼロ値が戻り、実際に受信したバイト数が `length` パラメータによって指示された値として戻ります。エラーが発生した場合、正の Net8 API エラー番号が戻されます。

---

---

**TNSCONTROL**

---

説明: 接続をブロック・モードまたは非ブロック・モードに設定します。

構文: 

```
int tnscontrol(handle, cmd)
void *handle;
int *cmd;
```



TNSCONTROL

パラメータ:	<p>handle (IN): tnsopen() によって戻される Net8 接続ハンドルへのポインタ</p> <p>cmd (IN): 接続に適用されるオプション。現在サポートされている値を次に示します。</p> <ul style="list-style-type: none"><li>■ TNSAPINONBLOCKING 接続を非ブロック・モードに設定します。</li><li>■ TNSAPIBLOCKING 接続をブロック・モードに設定します。</li></ul>
前提条件:	handle は NULL ではなく、cmd はサポートされているコマンドにしてください。
戻り値:	オプションが問題なく設定された場合はゼロ値が戻ります。

Net8 OPEN アプリケーション・プログラム・インタフェースのインストール先

アプリケーション・プログラム・インタフェースは、標準の Net8 インストレーションの一部として用意されています。そのインタフェースを使用するには、次のものを用意する必要があります。

- TNSAPI.H
  - API インタフェースとエラーが記述されているヘッダー・ファイルです。このファイルは、UNIX では \$ORACLE\_HOME/network/public、Windows NT では ORACLE\_HOME¥network¥tnsapi¥include にあります。
  - Net8 OPEN ライブラリ
- 他の Oracle ネットワーク・ライブラリと同じ場所に置かれます。名前には TNSAPI が含まれます。なお、ライブラリの名前は、プラットフォームごとに異なります。UNIX の場合は \$ORACLE\_HOME/network/lib ディレクトリにあり、名前は LIBTNSAPI.A です。Windows プラットフォームの場合は ORACLE\_HOME/network/tnsapi/lib にあり、ファイル名は TNSAPI.DLL および TNSAPI.LIB です。
- サンプル Make ファイル
- ネットワーク・ディレクトリ内にプラットフォーム用として、このファイルが用意されています。これらのファイルを使用して、アプリケーションを構築するための適切なリンク行を判断できます。



## 独自のアプリケーションの開発

Net8 OPEN ファンクションを参照するモジュールには、次に示すように、TNSAPI.H が含まれる必要があります。

```
#include <tnsapi.h>
```

Make ファイル（またはその他の作成コマンドの場所）には、TNSAPI.H を検出できるようにインクルード・パスを正しく設定します。インストールで用意されるサンプル Make ファイルを参照してください。

## Net8 OPEN アプリケーションを使用するためのシステムの構築

Net8 OPEN アプリケーションを認識するように Net8 を構成するには、次の手順に従います。

1. サーバー・プログラムの場所を listener.ora ファイルに追加します。これによりリスナーは、接続要求を受信したときにこのサーバーを起動します。

これを行うには、Oracle データベースと同じようなサービスのシステム識別子 (SID) 名を選択します。データベースと同じ SID は選択しないでください。

たとえば、"chat" プログラムを構成する場合は、SID "chatsid" です。プログラムを Oracle サーバーの実行可能プログラムと同じディレクトリに配置します。このディレクトリは通常、UNIX では \$ORACLE\_HOME/bin、Windows NT では ORACLE\_HOME¥bin です。

次のように、リスナー構成ファイルに次のエントリを設定します。

```
sid_list_listener =
  (sid_list =
    (sid_desc =
      (sid_name = chatsid)           /*SID 名*/
      (oracle_home = /usr/oracle)   /*$ORACLE_HOME の bin ディレクトリ*/
      (program = chatsvr)))         /*サーバー・プログラム名*/
```

リスナーが新しいサービスを認識するように、再起動する必要があります。

2. アプリケーション・サーバーのアドレスを tnsnames.ora ファイルに追加します。

たとえば、リスナーが次のアドレスでリスニングしているとします。

```
(description=(address=(protocol=tcp) (host=unixhost) (port=1521)))
```

先ほど作成したサービスを chat という名前にするとします。この場合、tnsnames.ora ファイルに次のエントリを追加します。

```
chat=
(description=
```



```
(address=(protocol=tcp) (host=unixhost) (port=1521))
(connect_data=(service_name=chat)))
```

3. Oracle サーバーの実行可能プログラムと同じディレクトリ内に、サービスの実行可能プログラムを配置します。UNIX プラットフォームの場合、実行可能ファイルは listener.ora ファイルで指定した \$ORACLE\_HOME/bin ディレクトリに置きます。この例では、プログラム chatsvr の場所は /usr/oracle/bin/chatsvr となります。  
オペレーティング・システムによっては、プログラムの実行許可があるかどうかを確認する必要があります。

## サンプル・プログラム

Net8 OPEN には、2つのサンプル・アプリケーションが用意されています。

- finger

これは、サーバーに接続し、ログインしているユーザーの情報を取り出すユーティリティです。このユーティリティには、分散アプリケーションの構築に関する基本的な手順を示す1対のプログラムが含まれています。このクライアント・プログラムは、Solaris と Windows NT の両方で実行されます。サーバーは UNIX 固有です。

- tftp

このサンプル・クライアント・プログラムとサンプル・サーバー・プログラムは、UNIX にインプリメントされており、tftp プロトコルを使用する単純なファイル転送を支援します。

## Net8 OPEN API エラー

この項では、前述のファンクション・コールに失敗したときに戻される可能性があるエラー番号の一覧を示します。送信または受信コールが行われた時点で接続が確立されていなかった場合は、これらのコールから接続関連エラーが戻る可能性があることに注意してください。

```
20002 - SDFAIL_TNSAPIE - The underlying "send" command failed in tnssend().
20003 - RECVFAIL_TNSAPIE - The underlying "receive" command failed in tnsrecv().
20004 - INVSROP_TNSAPIE - Operation is invalid as the server.
20005 - INVCLIO_TNSAPIE - Operation is invalid as the client.
20006 - HDLUNINI_TNSAPIE - The connection should be initialized by calling
tnsopen().
20007 - INHFAIL_TNSAPIE - Server failed in inheriting the connection from the
listener.
20008 - ACPTFAIL_TNSAPIE - Server failed in accepting the connection request from
the client.
20009 - NULHDL_TNSAPIE - A null handle was passed into the call, which is not
allowed.
20010 - INVOP_TNSAPIE - An invalid operation called was passed into the call.
```



```
20011 - MALFAIL_TNSAPIE - A malloc failed in TNS API call.
20012 - NLINIFAIL_TNSAPIE - Failed in NL initialization.
20013 - NMTOOLONG_TNSAPIE - Service name is too long.
20014 - CONFFAIL_TNSAPIE - Client connect request failed.
20015 - LSNFAIL_TNSAPIE - Server failed to listen for connect request.
20016 - ANSFALL_TNSAPIE - Server failed to answer connect request.
20017 - NMRESFAIL_TNSAPIE - Failed to resolve service name.
20018 - WOULDBLOCK_TNSAPIE - Operation would block.
20019 - CTLFAIL_TNSAPIE - Control call failed.
20020 - TNSAPIE_ERROR - TNS error occurred.
20021 - INVCTL_TNSAPIE - Invalid operation request in control call.
```

## UNIX クライアント・プログラミング

UNIX のイベント・プログラミングでは、UNIX シグナルを使用する必要があります。イベントが発生すると、シグナルによってプロセスに通知されます。プロセスは、生成された特定のシグナルに関連するコードを実行します。UNIX では、特定のシグナル・コールのための 1 つのプロセスが 1 つ以上のシグナル・ハンドラまたはアラームを設定することは許されていません。プロセスが SIGCHLD（子プロセスのステータス変更に関するシグナル）のようなシグナルに対して次のシグナル・ハンドラまたはアラーム要求を設定した場合、UNIX は SIGCHLD に対する直前の要求を無効化し、それを失います。

作成したアプリケーションからこのような要求が出された場合、シグナル処理またはアラームのためにシステムが特定の要求を消失し、応答がなくなる可能性があります。要求されたシグナルによっては、シグナル・ハンドラ上の問題のためにシステムは消滅したプロセスを正常にクリーン・アップしない可能性があります。

Net8 には、Oracle がこれらの要求を使用するとともにシグナル・ハンドラとアラームを使用できるようにする 2 つの解決策が用意されています。

- [シグナル・ハンドラとアラームのプログラミング](#)
- [Bequeath](#)

## シグナル・ハンドラとアラームのプログラミング

Net8 には、各シグナルに対するすべてのシグナル・ハンドラ要求またはアラーム要求のテーブルを維持するオペレーティング・システム依存（OSD）コールがあります。シグナル・ハンドラまたはアラームを使用するすべてのプログラムは、この Oracle OSD コールを使用する必要があります。OSD コールには、特定のコールに対して複数のシグナル・ハンドラまたはアラームを設定することが許されない UNIX 環境のプログラマのための解決策が用意されています。シグナル・ハンドラまたはアラームを使用するすべてのプログラムは、Oracle OSD コールを使用する必要があります。ただし、これは、現在内部でしか使用できません。

それまでは、データベース接続を行う前に、クライアントのすべてのシグナル・ハンドラを設定すると、OSD コールはシグナルに対して設定された最後のシグナル・ハンドラを記憶



し、それをシグナル・ハンドラ・テーブルに追加します。なお、これを行うと、シグナル・ハンドラを使用禁止にできなくなります。

## Oracle OSD シグナル処理規則

すべての SIGCHLD コールに対してデーブル駆動型共有 OSD シグナル・ハンドラを使用する場合は、次の規則に従う必要があります。

- 子プロセス ID を認識して、正しいプロセスをクリーン・アップできるようにします。
- 適切な子プロセス ID に対して、wait() ではなく、waitpid() コールを使用します。
- waitpid() コールを非ブロック化に設定します。

## Bequeath

この項の対象者は、子プロセスのステータスの変化を SIGCHLD コールによって把握する部分に UNIX シグナル・ハンドラを使用し、アプリケーションのネットワーク部分に Net8 を使用する UNIX アプリケーション・プログラマです。

クライアント・アプリケーションは、同じマシン上の Oracle データベースと通信するように命令されると、bequeath プロトコルを使用して接続を確立します。bequeath プロトコルは、クライアントがリスナーを使用せずにデータベースと接続できるようにします。

bequeath プロトコルは、クライアント・アプリケーションごとにサーバー・プロセスを内部的に起動します。ある意味で、bequeath プロトコルはリモート・リスナーが接続に対して行う同じ操作をローカルで実行します。

## 子プロセスの終了

クライアント・アプリケーションは bequeath プロトコルを介して内部にサーバー・プロセスを子プロセスとして生成するため、クライアント・アプリケーションは子プロセスが終了したときにそのプロセスをクリーン・アップする必要があります。サーバー・プロセスが接続の責任を完遂すると、それは消滅プロセスになります。シグナル・ハンドラは、これらの消滅プロセスをクリーン・アップする必要があります。別の方法として、シグナル・ハンドラを使用禁止にすることにより、このプロセスを UNIX init プロセスに渡すようにクライアントの sqlnet.ora ファイルを構成できます。

BEQUEATH\_DETACH パラメータを sqlnet.ora ファイルに構成します。

```
bequeath_detach=yes
```

このパラメータを選択すると、すべての子プロセスは UNIX init プロセス (pid = 1) に渡されます。init プロセスは、消滅した子プロセスを自動的にチェックして終了します。

bequeath は、シグナル・ハンドラの使用を自動的に選択して子プロセスのステータスの変化を捕捉します。作成したアプリケーションでシグナル・ハンドラを使用していない場合、このデフォルトによる影響はありません。



# 第 III 部

---

## Net8 テストおよびトラブルシューティング

第 III 部では、Net8 接続の確立方法および問題を識別し診断する方法について説明します。  
第 III 部には、次の章があります。

- 第 11 章「接続の確立とネットワークのテスト」
- 第 12 章「トラブルシューティング」







---

## 接続の確立とネットワークのテスト

ネットワークの構成が終了すると、接続を確立して各コンポーネントをテストし、ネットワークが正しく機能しているかどうかを確認する必要があります。Net8 には、Oracle Names サーバー、リスナー、Oracle Connection Manager を容易に起動し、テストおよび制御できる各種のツールがあります。

この章では、制御ユーティリティを使用し、接続を確立してネットワーク・コンポーネントをテストする手順を説明します。内容は次のとおりです。

- データベースへの接続
- ネットワークのテスト



## データベースへの接続

データベースに接続するには、ネットワーク・コンポーネントを起動し、ネット・サービス名を指定した接続文字列を入力する必要があります。たとえば次のように入力します。

```
CONNECT username/password@connect_identifier
```

この項の内容は、次のとおりです。

- [Net8 制御ユーティリティの使用](#)
- [接続文字列の入力](#)
- [接続識別子の構文特性](#)
- [接続の開始](#)

## Net8 制御ユーティリティの使用

Net8 には、各ネットワーク・コンポーネントを容易に起動、テスト、制御できる次のユーティリティがあります。

- Oracle Names 制御ユーティリティ (NAMESCTL)
- リスナー制御ユーティリティ (LSNRCTL)
- Connection Manager 制御ユーティリティ (CMCTL)

**関連項目：** [付録 A「制御ユーティリティ」](#) を参照してください。

### Oracle Names 制御ユーティリティ (NAMESCTL) の使用

Oracle Names 制御ユーティリティ (NAMESCTL) は、Oracle Names サーバーを起動および制御するためにオペレーティング・システム・プロンプトから実行するツールです。

Oracle Names 制御ユーティリティの一般的な書式は、次のとおりです。

```
NAMESCTL command
```

プログラム・プロンプトで、NAMESCTL コマンドを発行することもできます。コマンドラインで NAMESCTL を入力すると、プログラムがオープンします。次に、プログラム・プロンプトから必要なコマンドを入力します。たとえば、次のコマンドは、Oracle Names サーバーを起動します。

```
NAMESCTL> start
```

### リスナー制御ユーティリティ (LSNRCTL) の使用

リスナー制御ユーティリティ (LSNRCTL) は、リスナーを起動および制御するためにオペレーティング・システム・プロンプトから実行するツールです。リスナー制御ユーティリティの一般的な書式は、次のとおりです。



```
LSNRCTL command [listener_name] [args]
```

プログラム・プロンプトからリスナー制御コマンドを発行することもできます。コマンドラインに LSNRCTL と入力すると、プログラムがオープンします。次に、プログラム・プロンプトから必要なコマンドを入力します。たとえば、次のコマンドは、接続が開始されてからリスナーが有効な接続要求を待つ時間（秒単位）を決定します。

```
LSNRCTL> set connect_timeout 20
```

## Connection Manager 制御ユーティリティ（CMCTL）の使用

Connection Manager 制御ユーティリティ（CMCTL）は、Oracle Connection Manager を起動し、制御するためにオペレーティング・システム・プロンプトから実行するツールです。Connection Manager 制御ユーティリティの一般的な書式は、次のとおりです。

```
CMCTL command
```

プログラム・プロンプトで、CMCTL コマンドを発行することもできます。コマンドラインに CMCTL と入力すると、プログラムがオープンします。次に、プログラム・プロンプトから必要なコマンドを入力します。たとえば、次のコマンドは Oracle Connection Manager を起動します。

```
CMCTL> start
```

## Net8 コンポーネント起動の概要

クライアント・ワークステーションとその他のサーバーは、Oracle サーバーにログインするときにネット・サービス名を使用してリスナーに接続します。

すべてのネットワーク・コンポーネントをインストールし、構成した後は、それを起動してネットワークを機能させる必要があります。起動に必要な手順は、次のとおりです。

作業 1: Oracle Names サーバーの起動

作業 2: Oracle Names クライアント・キャッシュの開始

作業 3: リスナーの起動

作業 4: データベースの起動

作業 5: Oracle Connection Manager の起動

---

---

**注意：** Oracle Names サーバーがデータベースを使用してネットワーク情報を格納する場合は、最初にデータベースを起動し、次にリスナーを起動する必要があります。

---

---



作業 1: Oracle Names サーバーの起動

Oracle Names サーバー・ソフトウェアをインストールし構成しているマシンで、Oracle Net8 Assistant か NAMESCTL 制御ユーティリティを使用して、Oracle Names サーバーを起動します。

NAMESCTL 制御ユーティリティによる操作	Oracle Net8 Assistant による操作
コマンドラインから、次のように入力します。  NAMESCTL  NAMESCTL> start  NAMESCTL の STARTUP コマンドにより Oracle Names サーバーがメモリーにロードされ、実行開始の指示が出されます。Oracle Names サーバーは、起動時に構成とデータをロードします。	<ol style="list-style-type: none"><li>1. Net8 Assistant を起動します。  -UNIX の場合、\$ORACLE_HOME/bin で netasst を実行します。  -Windows NT の場合、「スタート」ボタン -&gt; 「プログラム」 -&gt; 「Oracle - HOME_NAME」 -&gt; 「Network Administration」 -&gt; 「Net8 Assistant」を選択します。</li><li>2. ナビゲータで、「Oracle Names Server」フォルダを展開します。</li><li>3. Oracle Names サーバーを選択します。</li><li>4. 右ペインのリスト・ボックスから「Manage Server」を選択します。</li><li>5. 「Control」タブをクリックします。</li><li>6. 「Server Operations」フィールドの「Start」ラジオ・ボタンをクリックします。</li><li>7. 「適用」をクリックします。</li></ol>

Windows NT の場合、Oracle Names サーバーはコントロール パネルから起動することもできます。

1. 「コントロール パネル」ウィンドウの「サービス」アイコンをダブルクリックします。
2. 「OracleHOME\_NAMENamesonames\_server」サービスを選択します。
3. 「開始」をクリックしてサービスを開始します。
4. 「サービス」ウィンドウの「閉じる」をクリックします。

作業 2: Oracle Names クライアント・キャッシュの開始

クライアントが Oracle Names サーバー内のアドレス情報を何度も検索しなくてすむように、クライアントに情報のキャッシュを作成します。この情報は ckpcch.ora ファイルに格納されます。このファイルのデフォルトの場所は、UNIX では \$ORACLE\_HOME/network/names、Windows プラットフォームでは ORACLE\_HOME¥network¥names です。

クライアント・キャッシュを作成するには、次の手順に従います。



1. 6-55 ページの「作業 4: Oracle Names サーバーを使用するためのクライアントとデータベース・サーバーの構成」の説明に従って、Oracle Names サーバーの検出が実行されたことを確認します。検出中に収集された情報は、クライアントのキャッシュで使用されます。
2. クライアント上にクライアント・キャッシュを作成して起動します。

```
NAMECTL
NAMECTL> start_client_cache
```

Windows NT の場合、クライアント・キャッシュはコントロール パネルから起動することもできます。

1. 「コントロール パネル」ウィンドウの「サービス」アイコンをダブルクリックします。
2. 「OracleHOME\_NAMEClientCache」サービスを選択します。
3. 「開始」をクリックしてサービスを開始します。
4. 「サービス」ウィンドウの「閉じる」をクリックします。

### 作業 3: リスナーの起動

Net8 がサーバー上の接続を受け付ける場合、サーバーで LSNRCTL 制御ユーティリティを使用して、リスナーを起動する必要があります。

1. コマンドラインから、次のように入力します。

```
LSNRCTL
LSNRCTL> status [listener_name]
```

*listener\_name* は、*listener.ora* ファイルに定義されたリスナー名を表します。  
LISTENER という名前のデフォルトのリスナーを使用している場合は、リスナーを識別する必要はありません。

STATUS コマンドでリスナーが実行されていることが示された場合は手順 2 に進みます。リスナーが実行されていない場合、手順 3 に進みます。

リスナーが実行されていない場合でも、リスナーを停止して再起動することをお勧めします。リスナーを停止するには、次のように入力します。

```
LSNRCTL> set password password
LSNRCTL> stop [listener_name]
```

SET PASSWORD が必要となるのは、*listener.ora* ファイルにパスワードが設定されている場合のみです。デフォルトのパスワードは、ORACLE です。

2. リスナーを起動します。次のように入力します。

```
LSNRCTL> start [listener_name]
```



LSNRCTL は、リスナーが正常に起動されたことを示すステータス・メッセージを表示します。そのリスナーで期待されるすべてのサービスがステータス・メッセージ内のサービス・サマリーの中に記述されていることを確認します。

3. LSNRCTL ユーティリティを終了します。次のように入力します。

```
LSNRCTL> exit
```

Windows NT の場合、リスナーはコントロール パネルから起動することもできます。

1. 「コントロール パネル」ウィンドウの「サービス」アイコンをダブルクリックします。
2. 「OracleHOME\_NAME\_TNSListener」サービス（デフォルト・リスナー名 LISTENER を使用している場合のサービス名）、または「OracleHOME\_NAME\_TNSListener $lsnr$ 」（ $lsnr$  は非デフォルト・リスナー名）を選択します。
3. 「開始」をクリックしてサービスを開始します。
4. 「サービス」ウィンドウの「閉じる」をクリックします。

### 作業 4: データベースの起動

SQL\*Plus などのツールを使用して、データベースを起動します。

1. 次のように入力して、データベースに接続せずに SQL\*Plus を起動します。

```
sqlplus /nolog
```

2. SYSDBA として Oracle に接続します。

```
SQL> connect username/password as sysdba
```

3. STARTUP コマンドを入力するときに、データベース名とパラメータ・ファイルのフル・パスを指定します。

```
SQL> startup database_name pfile=file
```

PFILE オプションを指定しない場合は、標準の初期化ファイルの場所が使用されます。この場所は、UNIX プラットフォームでは \$ORACLE\_BASE/admin/db\_name/pfile/sid、Windows NT では ORACLE\_BASE¥admin¥db\_name¥pfile¥sid になります。データベース名を指定しない場合は、初期化ファイルに指定された DB\_NAME パラメータの値が使用されます。

**関連項目：** データベースの起動に関する詳細は、『Oracle8i 管理者ガイド』を参照してください。



## 作業 5: Oracle Connection Manager の起動

Oracle Connection Manager をインストールし構成している場合、CMCTL 制御ユーティリティを使用して Oracle Connection Manager を起動します。

1. コマンドラインから、次のように入力します。

```
CMCTL  
CMCTL> start cman
```

CMCTL は、Oracle Connection Manager が正常に起動されたことを示すステータス・メッセージを表示します。

2. CMCTL ユーティリティを終了します。次のように入力します。

```
CMCTL> exit
```

Windows NT の場合、Connection Manager はコントロール パネルからも起動できます。

1. 「コントロール パネル」ウィンドウの「サービス」アイコンをダブルクリックします。
2. Oracle Names を使用している場合、使用可能な Oracle Names サーバーについての情報を取得するために「OracleHOME\_NAMECMAdmin」を選択し、「開始」をクリックします。Oracle Names を使用していない場合、このサービスは起動しないでください。

サービスが起動します。

3. Oracle Connection Manager を起動するために「OracleHOME\_NAMECMan」を選択し、「開始」をクリックします。

サービスが起動します。

4. 「サービス」ウィンドウの「閉じる」をクリックします。

## 接続文字列の入力

ネットワーク・コンポーネントを起動すると、11-2 ページの「データベースへの接続」に従ってネットワーク経由の接続を確立できます。接続の確立方法は、第 6 章「ネーミング・メソッドの構成」で構成したネーミング・メソッドと、接続に使用するツールによって異なります。

接続文字列は基本的に次のような形式になります。

```
CONNECT username/password@connect_identifier
```

## 接続識別子の構文特性

接続識別子に空白が含まれる場合は、引用符 (') または二重引用符 (") で囲む必要があります。次の例では、空白を含む接続識別子が引用符で囲まれています。



```
CONNECT scott/tiger@'(description=(address=(protocol=tcp)(host=sales-server)
(port=1521))(connect_data=(service_name=sales.us.acme.com)))'
CONNECT scott/tiger@'cn=sales, cn=OracleContext, dc=us, dc=acme, dc=com'
```

接続識別子の中で二重引用符 (") が使用されている場合は、引用符 (') で囲みます。たとえば、次のように指定します。

```
CONNECT scott/tiger@'sales@Good"Fast"Food.com'
```

接続識別子の中で引用符 (') が使用されている場合は、二重引用符 (") で囲みます。たとえば、次のように指定します。

```
CONNECT scott/tiger@"cn=sales, cn=OracleContext, ou=Mary's Dept, o=acme"
```

## 接続の開始

Oracle Server への接続を開始する方法は多数あります。一般的に使用される方法を次に示す項で説明します。

- [クライアントをテストするためのオペレーティング・システムからの接続](#)
- [クライアントをテストするためのツール・ログオン画面からの接続](#)
- [クライアントをテストするための 3GL からの接続](#)
- [Tools の特殊コマンドによる接続](#)

具体的な使用方法是、それぞれの方法によって多少の違いがあります。ここでは、前述した一般的な方法について簡単に説明します。特定ツールで使用される方法については、そのツールのユーザーズ・ガイドを参照してください。

### クライアントをテストするためのオペレーティング・システムからの接続

コマンドラインからアプリケーションをデータベース・サーバーに接続させるための一般的な書式を次に示します。

```
tool username/password@net_service_name
```

```
sqlplus system/password@sales
```

ログオン時にパスワードを表示しないようにするには、コマンドラインでパスワード・パラメータを入力しないでおきます。たとえば、次のように指定します。

```
sqlplus system@sales
```

パスワードの入力を求めるメッセージが表示されますが、画面上にパスワードは表示されません。



ほとんどの Oracle Tools は、オペレーティング・システムのコマンドラインを使用して接続できますが、他の方法でも接続できます。

## クライアントをテストするためのツール・ログオン画面からの接続

一部のツールには、ログオンの代替形式としてログオン画面が用意されています。ユーザーがデータベース・サーバーにログオンするには、ツール・ログオン画面のユーザー名フィールドにユーザー名とネット・サービス名 (`username@net_service_name`) の両方を指定し、パスワード・フィールドに通常どおりにパスワードを入力します。

## クライアントをテストするための 3GL からの接続

3GL で作成されたアプリケーションでは、プログラムは次の構文でサーバーへの接続を確立する必要があります。

```
exec sql connect :username identified by :password
```

この接続要求における `:username` および `:password` は 3GL 変数です。これらの変数は、静的あるいはユーザーからの入力によってプログラム内部に設定されます。データベース・サーバーに接続するとき、`:username` 変数の値は次のような書式になります。

```
username@net_service_name
```

`:password` 変数には、接続先データベース・アカウントのパスワードが入ります。

## Tools の特殊コマンドによる接続

一部の Oracle Tools には、そのツールが起動された後、ツールから離れずに代替ユーザー名を使用してデータベースに接続できるコマンドがあります。SQL\*Plus では、次の構文で CONNECT コマンドを使用できます。

```
SQL> CONNECT username/password@net_service_name
```

たとえば、次のように指定します。

```
SQL> CONNECT scott/tiger@serverx
```

これは、オペレーティング・システムのプロンプトではなくツールのプロンプトに応答して入力するという点以外は、オペレーティング・システムのコマンドラインを使用する方法と同じです。

他の Oracle Tools では、この説明とは異なる独自の機能やインタフェースに固有の方法が使用されます。たとえば、Oracle CDE Tools では、ログオン・ボタンとユーザー名およびパスワード、リモート・データベース ID フィールドがあるポップアップ・ウィンドウが使用されます。



## ネットワークのテスト

ネットワークをテストする望ましい順序は、次のとおりです。

1. 各 Oracle Names サーバー（ネットワーク・レイアウトに含まれている場合）を起動し、テストします。
2. リスナーを個別に起動し、テストします。
3. 各 Oracle Connection Manager（ネットワーク・レイアウトに含まれている場合）を起動し、テストします。
4. Server でループバック・テストを行います。
5. クライアントで接続できるかどうかテストします。

この項の内容は、次のとおりです。

- [Oracle Names サーバーのテスト](#)
- [Oracle Net8 Assistant または NAMESCTL ユーティリティによるネットワーク・オブジェクトのテスト](#)
- [リスナーのテスト](#)
- [Oracle Connection Manager のテスト](#)
- [サーバーでの構成のテスト](#)
- [クライアントからのネットワーク接続のテスト](#)

## Oracle Names サーバーのテスト

Oracle Names サーバーをテストするには、NAMESCTL PING コマンドを使用します。例として、us.acme ドメインの Oracle Names サーバー labrador を PING でテストする 2 つの方法を示します。

NAMESCTL プロンプトから、次のように入力します。

```
NAMESCTL> ping labrador.us.acme
```

同じ PING コマンドで複数の Oracle Names サーバーをテストすることもできます。たとえば、次のように指定します。

```
NAMESCTL> ping huey.uk.acme duey.uk.acme louie.uk.acme
```

Oracle Names サーバーに接触するためにかかる時間が出力され、肯定応答が戻ります。PING が失敗した場合は、Oracle Names サーバーが起動していることを確認するか、構成した Oracle Names サーバー・アドレスを再確認します。



## Oracle Net8 Assistant または NAMESCTL ユーティリティによるネットワーク・オブジェクトのテスト

Oracle Names サーバーに格納された情報は、QUERY コマンドで登録を確認するように問い合わせることができます。

QUERY コマンドを使用するには、Oracle Net8 Assistant か NAMESCTL 制御ユーティリティのどちらかを使用します。



Oracle Net8 Assistant による操作	NAMESCTL 制御ユーティリティによる操作
<p>1. Oracle Net8 Assistant を起動します。</p> <p>- UNIX の場合、\$ORACLE_HOME/bin で netasst を実行します。</p> <p>- Windows NT の場合、「スタート」ボタン-&gt;「プログラム」-&gt;「Oracle - HOME_NAME」-&gt;「Network Administration」-&gt;「Net8 Assistant」を選択します。</p> <p>2. 「Oracle Names サーバー」フォルダをダブルクリックします。</p> <p>3. Oracle Names サーバーを選択します。</p> <p>4. ドロップダウン・リスト・ボックスから「Manage Data」を選択します。</p> <p>5. 「Advanced」タブをクリックします。</p> <p>6. 「Query」を選択します。</p> <p>7. 「Name」フィールドに、問い合わせるオブジェクトの名前を入力します。</p> <p>8. 必要に応じて、「Name」フィールドにオブジェクトのタイプを入力します。</p> <p>- A.SMD: ネット・サービス名のデータベース・ネットワーク定義など、ネットワーク・アドレス</p> <p>- CNAME.SMD: 別名</p> <p>- DL.RDBMS.OMD: グローバル・データベース・リンク</p> <p>- DLCR.RDBMS.OMD: リンク修飾子</p> <p>- NS.SMD: Oracle Names サーバーのアドレス。Oracle Names サーバー間で通信する際に使用するシステム・データ。</p> <p>- V1ADD.NPO.OMD: SQL*Net バージョン 1 の接続文字列</p> <p>9. 「Execute」を選択します。</p> <p>10. 「File」メニューから「Save Network Configuration」を選択します。</p> <p>11. 「File」メニューから「Exit」を選択し、Net8 Assistant アプリケーションを終了します。</p>	<p>Oracle Names サーバーが管理領域に配置されている場合、Oracle Names サーバーに別名を作成して登録します。</p> <p>全データを表示するには、次のように入力します。</p> <p>NAMESCTL</p> <p>NAMESCTL&gt; query name *</p> <p>取り出す情報のタイプも指定できます。一般的なオブジェクトのタイプは次のとおりです。</p> <p>- A.SMD: ネット・サービス名のデータベース・ネットワーク定義など、ネットワーク・アドレス</p> <p>- CNAME.SMD: 別名</p> <p>- DL.RDBMS.OMD: グローバル・データベース・リンク</p> <p>- DLCR.RDBMS.OMD: リンク修飾子</p> <p>- NS.SMD: Oracle Names サーバーのアドレス。Oracle Names サーバー間で通信する際に使用するシステム・データ。</p> <p>- V1ADD.NPO.OMD: SQL*Net バージョン 1 の接続文字列</p> <p>ネット・サービス名のデータベース・アドレスを表示するには、次のように入力します。</p> <p>NAMESCTL</p> <p>NAMESCTL&gt; query name a.smd</p> <p>次の例は、ネット・サービス名 sales への問合せです。</p> <p>NAMESCTL</p> <p>NAMESCTL&gt; query sales a.smd</p> <p>QUERY コマンドは、トランザクションに要した時間とネットワーク・オブジェクトに関する情報を戻します。</p>



## リスナーのテスト

リスナーをテストするには、そのリスナーによって制御されるアクティブ・データベースに対してクライアントから接続を開始します。テストの手順は 11-13 ページの「[サーバーでの構成のテスト](#)」で説明します。

## Oracle Connection Manager のテスト

Oracle Connection Manager をテストするには、ソース・ルート・アドレスが作成されたアクティブなデータベースに対して、クライアントから接続を開始します。

## サーバーでの構成のテスト

ネットワークを構成し終えると、サーバーで[ループバック・テスト \(loopback test\)](#) を実行して、構成をテストします。

ループバック・テストでは、Net8 を使用してサーバーからプロセス間通信 (IPC) を利用せずにサーバー自体に戻ります。ループバックが正常に終了すると、Net8 がサーバー側で機能していることが確認できます。

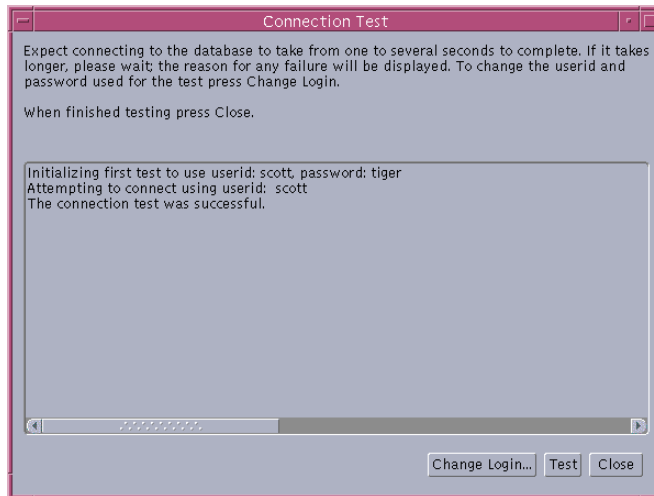
ループバック・テストを実行するには、Net8 Assistant を使用します。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Directory」または「Local」->「Service Naming」を展開します。
3. ネット・サービス名またはデータベース・サービスを選択します。
4. 「Command」->「Test Net Service」を選択します。

このテストでは、データベースとリスナーが実行されている必要があります。リスナーが実行されていない場合は、11-2 ページの「[Net8 制御ユーティリティの使用](#)」を参照してコンポーネントを起動してください。

テストが正常に終了すると、「Connect Test」ダイアログ・ボックスに次のメッセージが表示されます。





テストが正常終了しなかった場合は、次の処置を行ってください。

- データベースとリスナーが実行されていることを確認し、「Test」をクリックします。
- 「Change Login」をクリックして接続に使用するユーザー名とパスワードを変更し、「Test」をクリックします。

5. 「Close」をクリックして「Connect Test」ダイアログ・ボックスを閉じます。

## クライアントからのネットワーク接続のテスト

ネットワーク内でいくつかの異なるクライアントをテストするには、11-7 ページの「[接続文字列の入力](#)」にある手順に従って、それらの各クライアントからサーバーに接続を開始します。

また、Net8 にはネットワーク接続を容易に評価するために、次のツールがあります。

- [TNSPING ユーティリティ](#)
- [TRCROUTE ユーティリティ](#)
- [Oracle Net8 Assistant](#)

### TNSPING ユーティリティ

TNSPING は、Net8 ネットワーク上のサービス（Oracle データベースまたは Oracle Names サーバー、その他の Oracle サービスなど）に正常に到達できるかどうかを判断するためのユーティリティです。



TNSPING を使用してクライアントからサーバーに（または、あるサーバーから別のサーバーに）正常に接続できた場合は、Net8 サービスに到達するために必要な往復時間（ミリ秒単位）の概算が表示されます。

接続に失敗した場合は、発生したエラーを記述するメッセージが表示されます。これによって、データベース接続に対するオーバーヘッドなしで発生中のネットワーク・エラーを参照できます。

**TNSPING の使用** 次のような手順で、TNSPING ユーティリティを呼び出します。

```
tnsping net_service_name [count]
```

---

---

**注意：** プラットフォームによってインタフェースが異なる場合がありますが、プログラムは同じ引数を受け入れられます。TNSPING を実行すると、正しいインタフェース要件が表示されます。

---

---

- *net service name:* tnsnames.ora ファイル、または NIS や DCE の CDS などのネーム・サービスに存在する名前にしてください。
- *count* (オプション) : プログラムがサーバーへの到達を試行する回数を指定します。

指定されたサービス名がデータベース名の場合、TNSPING は、該当するリスナーへの接触を試行します。データベース自体が実行されているかどうかは、事実上判別されません。データベースへの接続を行うには、SQL\*Plus を使用します。

この後、TNSPING の例をいくつか示します。

**データベースへの到達** spotdb というネット・サービス名を使用してデータベースに接続するには、次のように入力します。

```
tnsping spotdb
```

これによって、次のメッセージが出力されます。

```
TNS Ping Utility for SunOS:
Copyright (c) Oracle Corporation 1998. All rights reserved.
Attempting to contact
(ADDRESS=(PROTOCOL=TCP) (HOST=spot) (PORT=1521))
OK (50msec)
```

TNSPING を 10 回試行して、stprd データベースに接続できるかどうかを判断するには、次のコマンドを使用します。

```
tnsping stprd 10
```



このコマンドによって、次のメッセージが出力されます。

```
TNS Ping Utility for SunOS:
Copyright (c) Oracle Corporation 1998. All rights reserved.
Attempting to contact (ADDRESS=(PROTOCOL=TCP) (HOST=spot) (PORT=1521))
OK (290 msec)
OK (100 msec)
OK (70 msec)
OK (70 msec)
OK (60 msec)
OK (70 msec)
OK (70 msec)
OK (80 msec)
OK (180 msec)
OK (340 msec)
```

**無効なネット・サービス名** 無効なネット・サービス名への接続を試行する TNSPING の例を次に示します。

```
tnsping bad_db
```

この試行では、次のメッセージが出力されます。

```
TNS Ping Utility for SunOS:
Copyright (c) Oracle Corporation 1998. All rights reserved.
TNS-03505: Failed to resolve name
```

**リスナーのない有効なネット・サービス名** 次に、名前は有効であるが、アドレスに変換するためのリスナーが配置されていない（たとえば、リスナーが起動されていない）場合の TNSPING の使用例を示します。

```
tnsping testing
```

次のメッセージが戻ります。

```
TNS Ping Utility for SunOS:
Copyright (c) Oracle Corporation 1998. All rights reserved.
Attempting to contact (ADDRESS=(PROTOCOL=tcp) (HOST=spot) (PORT=1521))
TNS-12541: TNS:no listener
```

**Oracle Names サーバーへの到達** Oracle Names サーバーに到達できるかどうかをチェックするには、次のように Net8 アドレスを使用してコマンドを入力します。

```
tnsping (address=(protocol=tcp) (host=fido) (port=1575))
```

次のようなメッセージがユーザーに戻されます。



```
TNS Ping Utility for SunOS:  
Copyright (c) Oracle Corporation 1998. All rights reserved.  
Attempting to contact (ADDRESS=(PROTOCOL=TCP) (HOST=fido) (PORT=1575))  
OK (70 msec)
```

## TRCROUTE ユーティリティ

管理者は、Trace Route ユーティリティ (TRCROUTE) を使用して、クライアントからサーバーへのパスまたはルートを知ることができます。TRCROUTE は、問題に遭遇すると、1つのエラーではなくエラー・スタックをクライアントに戻します。これらの追加エラー・メッセージによって、トラブルシューティングが簡単になります。

TRCROUTE は、特殊な種類の接続パケットとして送信される点で TNSPING とは異なります。接続先に向かって移動するとき、TRCROUTE 接続パケットは、経由するすべてのノードの TNS アドレスを収集します。エラーが発生した場合、TRCROUTE は、エラーの発生場所を示すエラー情報を収集します。Trace Route Utility は、収集した情報をクライアント画面に表示します。TRCROUTE の出力は、ファイルにリダイレクトして印刷することもできます。

**要件** Trace Route は、Net8 と SQL\*Net バージョン 2.3 以降でのみ動作します。クライアントからサーバーへのルートに沿ったすべてのノードは、SQL\*Net バージョン 2.3 以降を使用する必要があります。パス上に 2.3 より前のノードがある場合は、次のエラーが表示されます。

TNS-03603: SQL\*Net の 2.3 以前のバージョンのノードに直面しました。

TRCROUTE は、パス上のどのノードに問題があるかを示します。

**パフォーマンスへの影響** Trace Route Utility は、最小限のリソースを使用します。これは、特殊な接続パケットの接続データから情報を収集します。標準的な接続パケットには作用しません。

TRCROUTE は、サーバーに作用しません。TRCROUTE 接続パケットの受信と処理はリスナーが行います。これは、拒絶パケットの中に情報を入れることでクライアントに情報を戻します。サーバーは、新しいプロセスを起動したり、ダミー接続を処理する必要はありません。

**TRCROUTE の使用** TRCROUTE を呼び出すには、コマンドラインから次のように入力します。

```
trcroute net_service_name
```

次に、トレース・ルートの出力例を 2 つ示します。

**例 11-1** は、Trace Route パケットがクライアントからリスナーに正常に送信されたことを示します。



### 例 11-1 正常な Trace Route

```
%trcroute tcp_direct
Trace Route Utility for Solaris:
Copyright (c) Oracle Corporation 1998. All rights reserved.

Route of TRCROUTE:-----
Node: Client          Time and address of entry into node:
-----
01-DEC-98 13:26:36 ADDRESS= PROTOCOL=TCP Host=shining-sun Port=1581
Node: Server          Time and address of entry into node:
-----
01-DEC-98 13:27:20 ADDRESS= PROTOCOL=TCP Host=setting-sun Port=1521
```

例 11-2 は、リスナーが起動していないため、Trace Route パケットがそのリスナーへの到達に失敗したことを示します。

### 例 11-2 エラーの発生した Trace Route

```
% trcroute tcp_direct
Trace Route Utility for SVR4:
Copyright (c) Oracle Corporation 1998. All rights reserved.

Route of TRCROUTE:-----
Node: Client          Time and address of entry into node:
-----
01-DEC-98 11:12:34 ADDRESS= PROTOCOL=TCP Host=shining-sun Port=1581
TNS-12224: TNS: リスナーがありません。
TNS-12541: TNS: リスナーがありません。
TNS-12560: TNS: プロトコル・アダプタ・エラー
TNS-03601: ルート情報の収集に失敗しました。
```

## Oracle Net8 Assistant

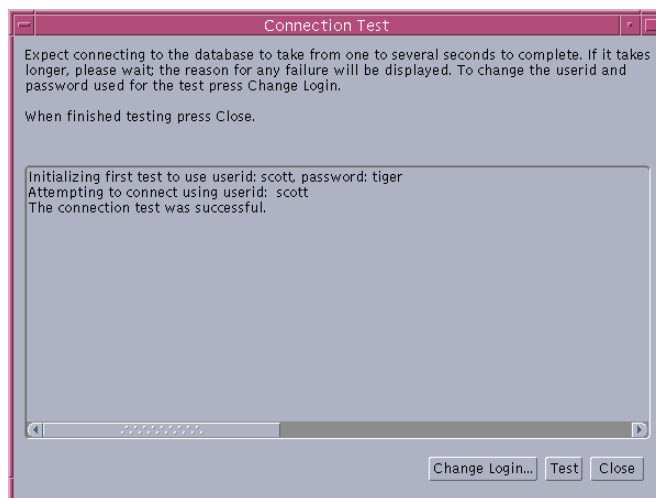
クライアント・マシンの接続性を検証するには、Net8 Assistant を使用します。

1. Oracle Net8 Assistant を起動します。
  - UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ナビゲータで、「Directory」または「Local」 -> 「Service Naming」を展開します。
3. ネット・サービス名またはデータベース・サービスを選択します。
4. 「Command」 -> 「Test Net Service」を選択します。



このテストでは、データベースとリスナーが実行されている必要があります。リスナーが実行されていない場合は、11-2 ページの「[Net8 制御ユーティリティの使用](#)」を参照してコンポーネントを起動してください。

テストが正常に終了すると、「Connect Test」ダイアログ・ボックスに次のメッセージが表示されます。



テストが正常終了しなかった場合は、次の処置を行ってください。

- データベースとリスナーが実行されていることを確認し、「Test」をクリックします。
  - 「Change Login」をクリックして接続に使用するユーザー名とパスワードを変更し、「Test」をクリックします。
5. 「Close」をクリックして「Connect Test」ダイアログ・ボックスを閉じます。







---

## トラブルシューティング

Net8 には、ログ・ファイルとトレース・ファイルを使用してネットワーク上の問題点を理解して解決する方法があります。これらのファイルは、エラーが発生すると、ネットワーク・コンポーネント間の対話を追跡し記録します。この情報を評価すると、非常に複雑なネットワーク上の問題点でも容易に診断してトラブルシューティングできます。

この章では、一般的なネットワーク・エラーとそれらのエラーを解決する手順について説明します。また、さらに複雑なネットワーク上の問題点を診断してトラブルシューティングするためにエラー情報をロギングしてトレースする方法についても説明します。内容は次のとおりです。

- [Net8 の診断](#)
- [一般的なエラー・メッセージの解決](#)
- [フィールドからのトラブルシューティングのヒント](#)
- [ログ・ファイルとトレース・ファイルを使用したネットワーク上の問題点のトラブルシューティング](#)
- [エラー情報のロギング](#)
- [エラー情報のトレース](#)
- [オラクル社カスタマ・サポート・センターへの連絡](#)



## Net8 の診断

Net8 のインストールと構成を終了し、基本的な peer-to-peer（単一プロトコル・ネットワーク）接続を確立しようとして ORA ERROR が戻された場合は、この項を参考にして問題の原因を診断してください。

基底にあるエラーが Net8 によりエラー番号やエラー・メッセージで報告されますが、これらは実際の問題を指しているとは限りません。この項は、Net8 の正常に機能する部分と機能していない部分を切り分けるのに役立ちます。また、エラーが次のどの項目に属するかを判断するのに役立ちます。

- Oracle ソフトウェア
- オペレーティング・システム・レイヤー
- その他のネットワーク・レイヤー

ほとんどの場合、各種ネットワーク・レイヤーを順にテストすることで問題を発見できます。

## サーバーの診断

チェック項目は次のとおりです。（この項の指示に従うにはサーバー管理者の助けが必要になることもあります）。

- 他のシステム（ワークステーション / サーバー）は Net8 を使用してサーバーに接続できます。
- サーバー、データベースまたはリスナーの構成は、しばらくの間変更されていません。

上の質問に 1 つでも当てはまる場合は、この項をスキップして 12-3 ページの「[クライアントの診断](#)」に進んでください。

不明な場合、またはいずれにも当てはまらない場合、このまま続けてください。

サーバー上の Net8 を診断するには次の作業を行います。

- [作業 1: データベースの実行状態の確認](#)
- [作業 2: ループバック・テストの実行](#)

### 作業 1: データベースの実行状態の確認

データベースが稼働していることを確認するには、次の手順を実行します。

有効なユーザー名とパスワードを使用してデータベースにログインし、接続します。たとえば、次のように指定します。

```
sqlplus system/manager
```

データベースに接続していることを確認するメッセージが表示されます。次のエラーが表示される場合は、データベース管理者に連絡してください。



- 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。
- ORA-01034: ORACLE は使用できません。

## 作業 2: ループバック・テストの実行

サーバーからデータベースへの**ループバック・テスト (loopback test)** を実行するには、次の手順に従います。

1. listener.ora、tnsnames.ora および sqlnet.ora の各ファイルが、UNIX では \$ORACLE\_HOME/network/admin、Windows NT では ORACLE\_HOME¥network¥admin にあることを確認します。  
構成ファイルの検索順序は次のとおりです。
  - a. アプリケーションが実行される現行の作業ディレクトリ
  - b. TNS\_ADMIN 環境変数  
Windows NT で TNS\_ADMIN が環境変数として定義されていない場合は、レジストリに設定されている可能性があります。
  - c. TNS\_ADMIN が定義されていない場合、UNIX では \$ORACLE\_HOME/network/admin、Windows プラットフォームでは ORACLE\_HOME¥network¥admin
2. ループバック・テストを実行するには、11-13 ページの「サーバーでの構成のテスト」にある手順に従います。
  - ループバック・テストでエラーが発生する場合、次の手順に進みます。
  - ループバック・テストが正常に終了する場合は、「クライアントの診断」に進みます。
3. 発生したエラーに関する詳細情報は、**データベースの問題および解決**を行う Web サイト <http://support.oracle.com> を確認するか、オラクル社カスタマ・サポート・センターに連絡してください。

## クライアントの診断

この時点で、次の最低 1 項目が確認されているので、Net8 サーバー側のリスナーは正常に機能していることになります。

- ネットワークが機能していることを示すサーバーでのループバック・テストが正常終了した
- 他のマシンは Net8 を使用して同じデータベースに接続している
- このマシンに変更を加える（新たな製品のインストールやネットワーク構成の変更など）前は、このワークステーションからの接続が機能していた

クライアントの診断を行うには、次の手順に従います。



- 1. サーバーにインストールされているのと同じ Oracle プロトコルがインストールされていることを確認します。

UNIX の場合、adapters プログラムを実行することで、このことを確認できます。  
\$ORACLE\_HOME/bin で adapters を実行します。

次のような出力が表示されます。

Installed Net8 Transport Protocols are:

IPC  
TCP/IP  
BEQueath  
SSL  
RAW

- 2. ...  
基底にあるネットワーク・トランスポートの基本的な接続性を確認します。Net8 テクノロジーを使用して正常に接続できるかどうかは、基底のネットワークに依存します。

プロトコル	確認方法
TCP/IP	ワークステーションから、リスナーやデータベースが配置されているサーバーに対して、ファイル転送やターミナル・エミュレーション・ユーティリティ（FTP、TELNET および PING）を使用します。
SPX	<ul style="list-style-type: none"><li>■ データベースを実行しているマシンに Netware ログインを実行する</li><li>■ ネットワーク上のドライブをマップできること、または Print Servers や File Servers などのその他の Novell サービスを使用できることを確認する</li><li>■ SPX ネットワーク上の Novell Server や Novell File Server から DISPLAY SERVERS コマンドを発行して、リスナー・サービスがブロードキャストを実行していることを確認する</li></ul>
Named Pipes	<ul style="list-style-type: none"><li>■ マイクロソフト・ネットワーク上のその他のコンピュータやサーバーを確認する</li><li>■ マイクロソフト・ネットワーク内のドライブを共有できることを確認する</li></ul>

- 3. Net8 クライアントと適切なプロトコルの両方が存在することを確認するために、すべての Net8 ソフトウェアがインストールされていることを確認します。
- 4. クライアント・マシンで、tnsnames.ora および sqlnet.ora ファイルが UNIX では \$ORACLE\_HOME/network/admin、Windows プラットフォームでは ORACLE\_HOME¥network¥admin にあることを確認します。



sqlnet.ora および tnsnames.ora の検索順序は次のとおりです。

- a. アプリケーションが実行される現行の作業ディレクトリ
- b. TNS\_ADMIN 環境変数  
Windows NT で TNS\_ADMIN 環境変数が変数として定義されていない場合は、レジストリに設定されている可能性があります。
- c. TNS\_ADMIN が定義されていない場合、UNIX では \$ORACLE\_HOME/network/admin、Windows プラットフォームでは ORACLE\_HOME¥network¥admin

選択した Oracle データベースに Net8 を使用して接続している他の稼働中のクライアント・マシンがある場合は、既存のファイルのバックアップをとって、稼働中のマシンの tnsnames.ora ファイルと sqlnet.ora ファイルを稼働していないクライアント・ワークステーションにコピーします。これにより、ファイルにエラーが含まれている可能性が排除されます。

5. 11-14 ページの「[クライアントからのネットワーク接続のテスト](#)」に従って、Net8 レイヤーをテストします。

---

**注意：** TNSPING は使用しないでください。TNSPING は TCP/IP PING ユーティリティと同様に機能し、ソケットの作成やオープンは行われず、リスナーとも接続しません。これで確認できるのは、リスナーがサーバー側に存在することのみです。

---

6. それでも接続がうまくいかない場合、次の作業を行います。
  - 12-12 ページの項「[ログ・ファイルとトレース・ファイルを使用したネットワーク上の問題点のトラブルシューティング](#)」に従って、トレースを使用します。
  - データベースの問題および解決を行う Web サイト  
<http://support.oracle.com> で、発生したエラーに関する診断情報を確認します。
  - オラクル社カスタマ・サポート・センターに連絡します。

## 一般的なエラー・メッセージの解決

ネットワーク通信が複雑であるため、様々な理由で各種ソースからネットワーク・エラーが発生する場合があります。エラーが発生した場合、SQL\*Plus や SQL\*Forms などのアプリケーション（Net8 からのネットワーク・サービスに依存しています）には、通常、エラー・メッセージが発生します。

次のリストは、最も一般的なネットワーク・エラー・メッセージです。



- ORA-12154: TNS: サービス名を解決できませんでした。
- ORA-12198: TNS: 接続先へのパスが見つかりませんでした。
- ORA-12203: TNS: 接続先に接続できません。
- ORA-12224: TNS: リスナーがありません。
- ORA-12533: TNS: ADDRESS パラメータが不正です。
- ORA-12545: 接続に失敗しました。ターゲット・ホストまたはオブジェクトが存在しません。
- ORA-12560: TNS: プロトコル・アダプタ・エラー
- ORA-03113: 通信チャネルでファイルの終わりが検出されました。
- ORA-03121: インタフェース・ドライバが接続されていないため関数は実行されません。

次の表は、各ネットワーク・エラーと、それらのエラーをトラブルシューティングする手順を示したものです。

エラー番号：メッセージ	説明 / トラブルシューティングの手順
ORA-12154: TNS: サービス名を解決できませんでした。	<p><b>原因：</b>Net8 が <code>tnsnames.ora</code> 構成ファイルに指定されたネット・サービス名を検出できません。</p> <p><b>処置：</b></p> <ol style="list-style-type: none"><li>1. <code>tnsnames.ora</code> ファイルが存在することを確認します。</li></ol> <p><b>関連項目：</b>構成ファイルの場所に関する情報は、12-3 ページの「<a href="#">クライアントの診断</a>」の手順 4（12-4 ページ）を参照してください。</p> <ol style="list-style-type: none"><li>2. <code>tnsnames.ora</code> ファイルのコピーが存在しないことを確認します。</li><li>3. 接続文字列で指定したネット・サービス名が <code>tnsnames.ora</code> ファイルの接続記述子にマップされていることを確認します。</li><li>4. <code>sqlnet.ora</code> ファイルのコピーが存在しないことを確認します。</li><li>5. ドメイン名を使用している場合は、<code>sqlnet.ora</code> ファイルに <code>NAMES.DEFAULT_DOMAIN</code> パラメータがあることを確認します。このパラメータが存在しない場合は、接続文字列にドメイン名を指定する必要があります。 ドメイン名を使用しておらず、このパラメータが存在する場合は、このパラメータを削除するか、またはコメントにして使用禁止にします。</li><li>6. ログイン・ダイアログ・ボックスから接続する場合は、接続するネット・サービス名の前に <code>@</code> 記号を付けていないことを確認します。</li><li>7. クライアントのトレースをオンにして、操作を再度実行します。</li></ol>



エラー番号：メッセージ	説明 / トラブルシューティングの手順
ORA-12198: TNS: 接続先へのパスが見つかりませんでした。	<b>原因：</b> クライアントが目的のデータベースを見つけられません。 <b>処置：</b>
ORA-12203: TNS: 接続先に接続できません。	<ol style="list-style-type: none"><li>1. 到達するネット・サービス名を正しく入力したかどうかを確認します。</li><li>2. tnsnames.ora ファイルの接続記述子に指定したネット・サービス名 ADDRESSS パラメータが正しいかどうかを確認します。</li><li>3. tnsnames.ora ファイルが正しいディレクトリに格納されていることを確認します。 <b>関連項目：</b> 構成ファイルの場所に関する情報は、12-3 ページの「<a href="#">クライアントの診断</a>」の手順 4 (12-4 ページ) を参照してください。</li><li>4. リモート・ノード上のリスナーが起動し、動作していることを確認します。次のように入力します。 <pre>lsnrctl lsnrctl&gt; status listener_name</pre><i>listener_name</i> は、listener.ora ファイルに定義されたリスナー名を表します。LISTENER という名前のデフォルトのリスナーを使用している場合は、リスナーを識別する必要はありません。 リスナーが実行中でないことが出力により示される場合、次のコマンドでリスナーを起動します。 <pre>lsnrctl&gt; start listener_name</pre></li><li>5. ログイン・ボックスから接続する場合は、接続するネット・サービス名の前に @ 記号を付けていないことを確認します。</li></ol>



エラー番号：メッセージ	説明 / トラブルシューティングの手順
ORA-12203: TNS: 接続先に接続できません。	<p>ORA-12203 エラーは総合的なエラーで、2 次的なエラーを含んでいる場合があります。このため、2 次的な ORA メッセージがあるかを最新の sqlnet.log ファイルで確認します。このログ・ファイルを解析して 2 次的なエラーがないことが判明した場合、次のいずれかが問題の原因である可能性があります。</p> <p><b>原因：</b> 選択したネットワーク・プロトコルには適切でない Oracle プロトコルがインストールされています。プロトコル・サポート・ドライバが欠落している場合、通常は sqlnet.log またはクライアント・トレース・ファイルに次のエラーが作成されます。</p> <ul style="list-style-type: none"><li>■ ORA-12203</li><li>■ ORA-12538</li><li>■ ORA-00508</li></ul> <p><b>処置：</b> 適切な Oracle プロトコルがインストールされていることを確認します。UNIX の場合、adapters プログラムを実行することで、このことを確認できます。 \$ORACLE_HOME/bin で adapters を実行します。</p> <p>次のように出力されます。</p> <p>Installed Net8 Transport Protocols are:</p> <pre>IPC TCP/IP BEQueath SSL RAW ...</pre>
ORA-12203 続き	<p><b>原因：</b> 接続文字列に無効なネット・サービス名が指定されています。</p> <p><b>処置：</b> 接続文字列に指定されたネット・サービス名が tnsnames.ora ファイルに存在すること、およびそのネット・サービス名の ADDRESS 情報が有効であることを確認します。チェック項目は次のとおりです。</p> <ul style="list-style-type: none"><li>■ HOST 名や SERVICE 名は正しいこと</li><li>■ 指定した PORT は正しいこと</li></ul>



エラー番号：メッセージ	説明 / トラブルシューティングの手順
ORA-12203 続き	<p><b>原因：</b> Net8 は tnsnames.ora ファイルに指定された接続記述子を検索できません。</p> <p><b>処置：</b> データベースが稼働していることを確認してから、次の作業を行います。</p> <ol style="list-style-type: none"><li>リスナーが実行中であることを確認します。次のように入力します。 <pre>lsnrctl lsnrctl&gt; status listener_name</pre><p><i>listener_name</i> は、listener.ora ファイルに定義されたリスナー名を表します。LISTENER という名前のデフォルトのリスナーを使用している場合は、リスナーを識別する必要はありません。</p><p>リスナーが実行中でないことが出力により示される場合、次のコマンドでリスナーを起動します。</p><pre>lsnrctl&gt; start listener_name</pre></li><li>tnsnames.ora ファイルが正しい場所にあることを確認します。構成ファイルの場所に関する情報は、12-3 ページの「<a href="#">クライアントの診断</a>」の手順 4（12-4 ページ）を参照してください。</li></ol>
ORA-12203 続き	<p><b>原因：</b> 接続先システムのリスナーがリスニングしていません。</p> <p><b>処置：</b> リモート・システムのリスナーが稼動中であることを確認します。次のように入力します。 <pre>lsnrctl lsnrctl&gt; status listener_name</pre><p><i>listener_name</i> は、listener.ora ファイルに定義されたリスナー名を表します。LISTENER という名前のデフォルトのリスナーを使用している場合は、リスナーを識別する必要はありません。</p><p>リスナーが実行中でないことが出力により示される場合、次のコマンドでリスナーを起動します。</p><pre>lsnrctl&gt; start listener_name</pre></p>
ORA-12203 続き	<p><b>原因：</b> 基底のネットワーク・トランスポートに問題があります。</p> <p><b>処置：</b> 使用するネットワーク・プロトコルに供給されているユーティリティで、プロトコル自体が機能していることを確認します。たとえば、TCP/IP の場合、リモート・システムに対して PING を実行します。</p>
ORA-12203 続き	<p><b>原因：</b> tnsnames.ora ファイルが適切なディレクトリに配置されていません。</p> <p><b>処置：</b> tnsnames.ora ファイルが正しい場所にあることを確認します。</p> <p><b>関連項目：</b> 構成ファイルの場所に関する情報は、12-3 ページの「<a href="#">クライアントの診断</a>」の手順 4（12-4 ページ）を参照してください。</p>



エラー番号：メッセージ	説明 / トラブルシューティングの手順
ORA-12203 続き	<p><b>原因：</b> TCP/IP アドレスの (HOST=<i>server_name</i>) パラメータまたは SPX の (SERVICE=<i>tns_application</i>) パラメータがクライアント・マシンとサーバー・マシンとで一致していません。</p> <p><b>処置：</b> これらのパラメータの値がサーバーとクライアントで同じであることを確認します。</p> <p>TCP/IP セットアップの場合は、サーバーの listener.ora ファイルの HOST パラメータとクライアントの tnsnames.ora ファイルの HOST パラメータが、同じ名前か、各システムで同一の IP アドレスに変換される名前を示していることを確認します。これは特に、サーバーが複数の IP アドレスを持ち、それぞれがサーバー上の各種ネットワーク・インタフェースに割り当てられている場合に重要です。</p> <p>SPX セットアップの場合、名前がサーバー・マシンとクライアント・マシンで同じである必要があります。</p>
ORA-12224: TNS: リスナーがありません。	<p><b>原因：</b> リスナーが実行されていないため、接続要求を完了できません。</p> <p><b>処置：</b></p> <ol style="list-style-type: none"><li>1. 指定した接続先アドレスが、リスナーで使用されているアドレスの 1 つと一致していることを確認します。</li><li>2. このエラーが、バージョンの互換性に起因する問題でないかどうかを確認します。</li></ol>
ORA-12533: TNS: ADDRESS パラメータが不正です。	<p><b>原因：</b> tnsnames.ora ファイルにおいて、指定された接続記述子の ADDRESS セクションにあるプロトコル固有パラメータが正しくありません。</p> <p><b>処置：</b> 正しいプロトコル構文の詳細は、B-2 ページの「<a href="#">プロトコルのパラメータ</a>」を参照してください。</p>
ORA-12545: 接続に失敗しました。ターゲット・ホストまたはオブジェクトが存在しません。	<p><b>原因：</b> リモート・ノード上のリスナーに接続できません。</p> <p><b>処置：</b></p> <ol style="list-style-type: none"><li>1. tnsnames.ora ファイルおよび listener.ora ファイル内の ADDRESS が正しいかどうかを確認します。</li><li>2. リモート・ノード上のリスナーが起動されたことを確認します。次のように入力します。<pre>lsnrctl lsnrctl&gt; status listener_name</pre><p><i>listener_name</i> は、listener.ora ファイルに定義されたリスナー名を表します。LISTENER という名前のデフォルトのリスナーを使用している場合は、リスナーを識別する必要はありません。</p><p>リスナーが実行中でないことが出力により示される場合、次のコマンドでリスナーを起動します。</p><pre>lsnrctl&gt; start listener_name</pre></li></ol>



エラー番号：メッセージ	説明 / トラブルシューティングの手順
ORA-12560: TNS: プロトコル・アダプタ・エラー	<p><b>原因：</b> リスナーが、ユーザーをデータベース・サーバーに接続するプロセスの起動に失敗した。</p> <p><b>処置：</b></p> <ol style="list-style-type: none"> <li>1. トレースをオンにして、操作を再度実行します。</li> <li>2. 問題を診断するために、トレース・ファイルの内容を検証します。</li> </ol> <p><b>関連項目：</b> 12-28 ページの「<a href="#">エラー情報のトレース</a>」を参照してください。</p>
ORA-03113: 通信チャネルでファイルの終わりが検出されました。	<p><b>原因：</b> 通信チャネル上で、予測しなかったファイルの終わりが処理された。これは、通信リンクが少なくとも一時的にダウンした可能性があることを示す場合があります。また、サーバーがダウンしていることを示す場合もあります。</p> <p><b>処置：</b> サーバーのログとトレースをチェックしてください。</p>
ORA-03121: インタフェース・ドライバが接続されていないため関数は実行されません。	<p><b>原因：</b> 接続文字列で使用される SQL*Net バージョン 1 の接頭辞に誤りがあります。</p> <p><b>処置：</b> 接続文字列では次の接頭辞は使用しないでください。</p> <ul style="list-style-type: none"> <li>■ T:</li> <li>■ X:</li> <li>■ P:</li> </ul> <p><b>原因：</b> ローカル Oracle データベースがインストールされていないクライアント・マシンでユーザー名とパスワードのみが指定された。</p> <p><b>処置：</b> 接続文字列を指定します。</p>

## フィールドからのトラブルシューティングのヒント

ネットワークの問題の診断に困ったときは、次のヒントが参考になることがあります。

- **構成中はサーバー・マシン名ではなくノードやネットワーク・アドレスを使用する** - これにより内部的な検索の問題が除去され、接続がやや高速になります。

TCP/IP - TCP/IP アドレスのホスト名ではなく、198.32.3.5 のようなインターネット・アドレスを使用します。たとえば、tnsnames.ora ファイルの (HOST=server\_name) 行を (HOST=198.32.3.5) のようにインターネット・アドレスに変更します。

- **SPX 接続の問題を考慮する** - リモート・リスナーでの接続を要求するワークステーションは、まず NetWare IPX ネットワークでの SPX サービスのロケーションを認識する必要があります。

クライアント・ワークステーションは、SPX サービスの検索要求を発行します。サービスが見つからない場合、ワークステーションにエラーが戻されます。

- **ループバック・テストを実行する** - 11-13 ページの「[サーバーでの構成のテスト](#)」に従って、サーバーでループバック・テストを実行します。テストが正常に終了する場合は、



FTP を使用して tnsnames.ora ファイルと sqlnet.ora ファイルをクライアントに転送します。

- **作業マシンとサーバーの間をチェックする** - ワイド・エリア・ネットワーク（WAN）の場合、すべての中間システムが正常に機能していることを確認します。すべてのマシンが正常である場合、タイミングに問題がある可能性があります。
- **タイミングに問題がないかどうかを確認する** - タイミングの問題は、クライアント・ログ・ファイルの ORA-12203、ORA-12535 または ORA-12547 のエラーに関係します。

この問題を解決するには、名前ではなく正確なアドレスを使用して接続のスピード・アップを図り、listener.ora ファイルの [CONNECT\\_TIMEOUT\\_listener\\_name](#) パラメータの値を増やします。このパラメータのデフォルト値は 10 秒です。

- **どの Oracle アプリケーションでエラーが発生しているかを判断する** - SQL\*Plus は機能していても、CASE ツールが機能していない可能性があります。データ量に問題があることが判明した場合、基本的な接続を使用してサイズの大きい（5MB）ファイルを転送してみます。

## トラブルシューティング時のチェックリスト

問題を診断する際に、次の項目をチェックしてください。

- **すべてのマシンに問題があるのか、1 台のマシンのみか**

同じソフトウェア（Oracle およびサードパーティ製品）をインストールしていても、機能するマシンと機能しないマシンがある場合、距離が近ければ、ネットワーク・ケーブルを取り替えて問題が進展するかどうかを確認します。状況に変化があった場合、問題はクライアント / サーバー接続に関するもので、PC に限定的なものではないことを示しています。

- **クライアントとサーバー間のリンクの種類（X.25、ISDN、トークン・リングまたは専用線など）は何か**

Sniffers や LAN Analyzers は、断続的な接続エラーやタイムアウトおよび再送信パケットの検出を行うのに便利です。また、会話のどちら側が応答を待っているのかを知ることができます。

- **Oracle のアプリケーションは機能しても、サードパーティ・アプリケーションにエラーが発生するか**

## ログ・ファイルとトレース・ファイルを使用したネットワーク上の問題点のトラブルシューティング

Net8 は、問題が発生するたびに、問題点の原因の内容について詳細を提供します。この情報は、ログ・ファイルとトレース・ファイル内に作成、格納されます。エラー情報のロギングとトレースのプロセスにより、ネットワーク上の問題点を容易に診断し解決できます。



サーバーとリスナーの場合、ログ・ファイルのデフォルトの場所は、UNIX では \$ORACLE\_HOME/network/log、Windows プラットフォームでは ORACLE\_HOME¥network¥log、トレース・ファイルのデフォルトの場所は、UNIX では \$ORACLE\_HOME/network/trace、Windows プラットフォームでは ORACLE\_HOME¥network¥trace です。クライアントの場合、ログ・ファイルとトレース・ファイルのデフォルトの場所は、現行の作業ディレクトリです。

## エラー情報のロギング

Oracle ネットワーク製品で発生するすべてのエラーは、ネットワーク管理者やデータベース管理者が評価できるように、ログ・ファイルに追加記録されます。ログ・ファイルは、画面に表示されるエラー・メッセージで障害を識別できない場合に、管理者用の追加情報として使用できます。エラー・スタックとしてのログ・ファイルは、各種レイヤーでのソフトウェアの状態を示します。

すべてのエラーが記録されることを保証するためにはクライアントまたはネーム・サーバーに対するロギングを使用禁止にすることはできません。さらに、ログ・ファイルを置換したり消去できるのは管理者のみです。リスナーのログ・ファイルには、すべてのクライアント接続要求に関する監査証跡と大半のリスナー制御コマンドが含まれています。

この項の内容は、次のとおりです。

- [エラー・スタック](#)
- [ログ・ファイル名](#)
- [ログ・パラメータの理解と設定](#)
- [制御ユーティリティ実行時のログ設定](#)
- [ログ・ファイルの使用](#)
- [リスナーのログ監査証跡情報](#)
- [Oracle Connection Manager ログの理解](#)

## エラー・スタック

ログ・ファイルには、エラー・スタックに含まれている情報があります。エラー・スタックとは、ネットワーク・エラーの結果として Oracle 通信スタック内に各レイヤーごとに作成される情報です。

図 12-1 は、エラー・スタックに表示される可能性のある Oracle ネットワーク製品間の関係を示したものです。



図 12-1 ネットワーク製品とエラー・スタック・コンポーネント

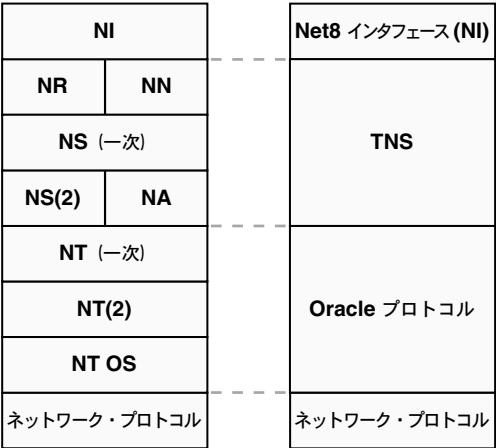


図 12-1 のレイヤーは次のとおりです。

- NI Net8 インタフェース・レイヤー
- NR ネットワーク・ルーティング
- NN ネットワーク・ネーミング (Oracle Names)
- NS ネットワーク・セッション (一次および二次層)
- NA ネットワーク認証 (NA) とネットワーク暗号化 (NAE) など、固有のサービス
- NT ネットワーク・トランスポート (一次、二次およびオペレーティング・システム層)

ネットワークによって、これらすべてのコンポーネントが含まれている場合と、含まれていない場合があります。

エラー例

たとえば、クライアント・アプリケーションのユーザーが Net8 と TCP/IP を使用してデータベース・サーバーとの接続を確立しようとする場合は、次のように入力します。

```
sqlplus scott/tiger@hrserver.com
```

SQL\*Plus バナーが画面に表示され、次のエラーが表示されます。



ORA-12203: TNS: 接続先に接続できません。

このメッセージは、データベースに連絡できなかったために、サーバーへの接続に失敗したことを示します。アプリケーションでは1行のエラー・メッセージしか表示されませんが、ネットワーク層によってログ・ファイルの中に情報量の多いエラー・スタックが記録されます。クライアント側では、sqlnet.log と呼ばれるログ・ファイルに次のようなORA-12203 エラーに対応するエラー・スタックが含まれています。

### 例 12-1 一般的なエラー・スタック

```
*****
Fatal OSN connect error 12203, connecting to:
  (DESCRIPTION= (CONNECT_DATA= (SID=trace) (CID= (PROGRAM=
    (HOST=lala) (USER=sviavant))) (ADDRESS_LIST= (ADDRESS=
    (PROTOCOL=ipc) (KEY=trace)) (ADDRESS= (PROTOCOL=tcp)
    (HOST=lala) (PORT=1521))))
VERSION INFORMATION:
TNS for SunOS:
Oracle Bequeath NT Protocol Adapter for SunOS:
Unix Domain Socket IPC NT Protocol Adaptor for SunOS:
TCP/IP NT Protocol Adapter for SunOS:
  Tracing to file: /home/sviavant/trace_admin.trc
  Tns error struct:
    nr err code: 12203
    TNS-12203: TNS: 接続先に接続できません。
    ns main err code: 12541
    TNS-12541: TNS: リスナーがありません。
    ns secondary err code: 12560
    nt main err code: 511
    TNS-00511: リスナーがありません。
    nt secondary err code: 61
    nt OS err code: 0
```

## ログ・ファイル名

各 Net8 コンポーネントは、独自のログ・ファイルを作成します。次の表は、デフォルトのファイル名とそれに含まれている情報の説明を示したものです。



ログ・ファイル	エラー情報の内容
sqlnet.log	クライアントまたはサーバー、あるいはその両方
listener.log	リスナー
names.log	Oracle Names サーバー
cman_pid.log (UNIX の場合)	Oracle Connection Manager CMGW ゲートウェイ・プロセス
cmanpid.log (Windows NT の場合)	
cmadm_pid.log (UNIX の場合)	Oracle Connection Manager CMADMIN 管理プロセス
cmadmpid.log (Windows NT の場合)	

ログ・パラメータの理解と設定

ロギングされる情報の種類と量、ファイルが格納されている位置など、ロギングを制御するパラメータは、次のように各ネットワーク・コンポーネントの構成ファイル内に設定されます。

ネットワーク・コンポーネント	パラメータを設定する構成ファイル
クライアント	sqlnet.ora
サーバー	sqlnet.ora
リスナー	listener.ora
Oracle Names サーバー	names.ora
Oracle Connection Manager プロセス	cman.ora

この項の内容は、次のとおりです。

- [sqlnet.ora](#) パラメータ
- [listener.ora](#) ファイルのパラメータ
- [names.ora](#) ファイルのパラメータ
- [cman.ora](#) ファイルのパラメータ
- [構成ファイルのログ・パラメータ設定](#)

**関連項目：** [付録 C「構成パラメータ」](#) を参照してください。



## sqlnet.ora パラメータ

sqlnet.ora ファイルに設定できるパラメータは次のとおりです。

sqlnet.ora ファイルのパラメータ	Oracle Net8 Assistant のフィールド	説明
LOG_DIRECTORY_CLIENT	「Client Information: Log Directory」	クライアント・ログ・ファイルの作成先となるディレクトリを設定します。デフォルトのクライアント・ディレクトリは、現行の作業ディレクトリです。
LOG_FILE_CLIENT	「Client Information: Log File」	クライアントのトレース・ファイルの名前を設定します。デフォルトのログ・ファイル名は sqlnet.log です。
LOG_DIRECTORY_SERVER	「Server Information: Log Directory」	サーバー・ログ・ファイルの作成先となるディレクトリを設定します。デフォルトのサーバー・ディレクトリは、UNIX では \$ORACLE_HOME/network/log、Windows NT では ORACLE_HOME¥network¥log です。
LOG_FILE_SERVER	該当せず	サーバーのログ・ファイルの名前を設定します。デフォルトのログ・ファイル名は sqlnet.log です。

## listener.ora ファイルのパラメータ

listener.ora ファイルに設定できるログ・パラメータは次のとおりです。

listener.ora ファイルのパラメータ	Oracle Net8 Assistant のフィールド	説明
LOG_DIRECTORY_listener_name LOG_FILE_listener_name	「Log File」	リスナー・イベントで自動的に生成されるファイルの作成先となるディレクトリとログ・ファイルを設定します。デフォルトのディレクトリは、UNIX では \$ORACLE_HOME/network/log、Windows NT では ORACLE_HOME¥network¥log です。デフォルトのファイル名は listener.log です。

## names.ora ファイルのパラメータ

names.ora ファイルに設定できるログ・パラメータは次のとおりです。



names.ora ファイルのパラメータ	Oracle Net8 Assistant のオプション	説明
NAMES.LOG_DIRECTORY	「Log Directory」	ログ・ファイルの作成先となるディレクトリを設定します。デフォルトのディレクトリは、UNIX では \$ORACLE_HOME/network/log、Windows NT では ORACLE_HOME¥network¥log です。
NAMES.LOG_FILE	「Log File」	クライアントのトレース・ファイルの名前を設定します。デフォルトのログ・ファイル名は names.log です。

**cman.ora ファイルのパラメータ**

cman.ora ファイルに設定できるログ・パラメータは次のとおりです。

cman.ora ファイルのパラメータ	説明
LOG_LEVEL	ロギングのレベルを設定します。次の 5 つのレベルがサポートされています。 <ul style="list-style-type: none"><li>■ レベル 0 - ロギングなし</li><li>■ レベル 1 - 基本的なレポートの作成</li><li>■ レベル 2 - RULE_LIST 一致検索レポートの作成</li><li>■ レベル 3 - リレー・ブロッキング・レポートの作成</li><li>■ レベル 4 - リレー I/O カウント・レポートの作成</li></ul> CMGW ゲートウェイ・プロセスにより作成されるログ・ファイル名は、UNIX では cman_pid.log、Windows NT では cmanpid.log です。CMADMIN 管理プロセスにより作成されるログ・ファイル名は、UNIX では cmadm_pid.log、Windows NT では cmadmpid.log です。

**構成ファイルのログ・パラメータ設定**

sqlnet.ora ファイル、listener.ora ファイルおよび names.ora ファイルのロギング・パラメータは Oracle Net8 Assistant で設定できます。cman.ora ファイルのロギング・パラメータは手動で設定する必要があります。

**関連項目：** C-65 ページの「[Oracle Connection Manager パラメータ \(cman.ora\)](#)」を参照してください。

ロギング・パラメータを設定するには、次の手順に従います。

1. Oracle Net8 Assistant を起動します。



- UNIX の場合、\$ORACLE\_HOME/bin で netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2. ログ・パラメータを指定します。

ログ・ファイル	ロギング・パラメータの設定方法
sqlnet.log	<ol style="list-style-type: none"><li>1. ナビゲータで、「Local」 -&gt; 「Profile」を展開します。</li><li>2. 右ペインのリストから「General」を選択します。</li><li>3. 「Logging」タブをクリックします。</li><li>4. 設定を指定します。</li></ol>
listener.log	<ol style="list-style-type: none"><li>1. ナビゲータで、「Local」 -&gt; 「Listeners」を展開します。</li><li>2. リスナーを選択します。</li><li>3. 右ペインのリストから「General」を選択します。</li><li>4. 「Logging and Tracing」タブをクリックします。</li><li>5. 設定を指定します。</li></ol>
names.log	<ol style="list-style-type: none"><li>1. ナビゲータで、「Oracle Names サーバース」を展開します。</li><li>2. Oracle Names サーバースを選択します。</li><li>3. 右ペインのリストから「Configure Server」を選択します。</li><li>4. 「Adv.」タブをクリックします。</li><li>5. ログ・ディレクトリとファイル名を指定します。</li></ol>

3. 「File」 -> 「Save Network Configuration」を選択します。
- Net8 Assistant アプリケーションが終了します。

制御ユーティリティ実行時のログ設定

ロギングは、制御ユーティリティの実行中に設定できます。制御ユーティリティを使用してロギングを設定しても、\*.ora ファイルにパラメータは設定されません。設定は、制御ユーティリティの現在のセッションでのみ有効です。

- リスナーの場合は、LSNRCTL 制御ユーティリティから [SET LOG\\_FILE](#) や [SET LOG\\_DIRECTORY](#) コマンドを実行します。実行方法は A-2 ページの「リスナー制御ユーティリティ (LSNRCTL)」を参照してください。
- Oracle Connection Manager の場合は、CMCTL 制御ユーティリティから [SET LOG\\_LEVEL](#) を実行します。実行方法は A-77 ページの「Oracle Connection Manager 制御ユーティリティ (CMCTL)」を参照してください。



- Oracle Names サーバーの場合は、NAMESCTL 制御ユーティリティから [SET LOG\\_FILE\\_NAME](#) コマンドを実行します。実行方法は、A-25 ページの「[Oracle Names 制御ユーティリティ \(NAMESCTL\)](#)」を参照してください。あるいは次のように Oracle Net8 Assistant から設定します。
  - a. Oracle Net8 Assistant を起動します。
    - UNIX の場合、`$ORACLE_HOME/bin` で `netasst` を実行します。
    - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
  - b. ナビゲータで、「Oracle Names Servers」フォルダを展開します。
  - c. Oracle Names サーバーを選択します。
  - d. 右ペインのリストから「Manage Server」を選択します。
  - e. 「Logging」タブをクリックします。
  - f. ログ・ディレクトリとファイル名を指定します。
  - g. 「File」 -> 「Save Network Configuration」を選択します。

## ログ・ファイルの使用

ログ・ファイルを使用してネットワーク・エラーを診断するには、次の手順に従います。

1. アプリケーションから受信した最新のエラー番号でログ・ファイルを探します。ほとんどの場合、これがログ・ファイルの最後のエントリです。
2. ファイルの一番下から順にたどり、エラー・レポートで 0 以外の最初のエントリを探します。通常は、これが実際の原因です。
3. そのエラーが希望の情報を示していない場合は、正しいエラー情報が見つかるまでスタック内の次のエラーを見直します。
4. それでもエラーの原因が明らかにならない場合は、トレースをオンにし、エラー・メッセージが発生した文を再度実行します。

## リスナー・ログ・ファイルの理解

この項では、リスナー・ログ・ファイルに記録される情報について説明します。内容は次のとおりです。

- [リスナーのログ監査証跡情報](#)
- [サービス登録イベント情報](#)
- [Direct Hand-Off 情報](#)



## リスナーのログ監査証跡情報

リスナー・ログ・ファイルには、ネットワーク利用度の統計を収集および分析できる監査証跡情報と、次の内容を示す情報があります。

- クライアントの接続要求
- LSNRCTL 制御ユーティリティが発行した start、stop、status、reload、service の各コマンド

**リスナーのログ監査証跡の書式** 監査証跡のテキストは次のフィールドで構成されます。

*Timestamp \* Connect Data \* [Protocol Info] \* Event \* [SID | Service] \* Return Code*

監査証跡のプロパティは次のとおりです。

- 各フィールドはアスタリスク (\*) で区切られます。
- プロトコル・アドレス情報および SID またはサービス名情報は、接続が試行された場合のみ記録されます。
- 成功した接続またはコマンドには、コード 0 が戻ります。
- 失敗した場合は、エラー・メッセージにマップされるコードが生成されます。

**関連項目：** 一般的な Net8 エラーの詳細は、12-5 ページの「[一般的なエラー・メッセージの解決](#)」、すべてのエラー・メッセージを網羅したリストは、『Oracle8i エラー・メッセージ』を参照してください。

STOP 要求に関する一般的なログ・ファイル出力を[例 12-2](#)に示します。

### 例 12-2 正常な Reload 要求に対するリスナー・ログ・イベント

```
14-SEP-1999 00:29:54 *
(CONNECT_
DATA= (CID= (PROGRAM=) (HOST=dlsun1013) (USER=jdoe)) (COMMAND=stop) (ARGUMENTS=64) (SERVICE
=LISTENER) (VERSION=135290880))
* reload * 0
```

接続要求に関する一般的なログ・ファイル出力を[例 12-3](#)に示します。

### 例 12-3 正常な接続要求に対するリスナー・ログ・イベント

```
10-AUG-1999 15:28:58 *
(CONNECT_DATA= (service_name=sales.us.acme.com) (CID= (PROGRAM=) (HOST=dlsun1013)
(USER=jdoe)))
* (ADDRESS= (PROTOCOL=tcp) (HOST=144.25.185.246) (PORT=41349)) * establish
* sales.us.acme.com * 0
```



**監査証拠情報の使用** 監査証拠情報を表に保存した後、それをレポート形式に書式設定することで、傾向とユーザー・アクティビティを検証できます。データを表にインポートするには、SQL\*Loaderなどのインポート・ユーティリティを使用します。

サービス登録イベント情報

リスナーではサービス登録イベントが記録されます。サービス登録を行うと、インスタンスのバックグラウンド・プロセス PMON からリスナーに次の情報が供給されます。

- データベースの各稼動インスタンスのサービス名
- データベースのインスタンス名
- 使用可能なサービス・ハンドラ（ディスパッチャまたは専用サーバー）
- ディスパッチャ、インスタンスおよびノードのロード情報

listener.log ファイルに記録されるサービス登録関連イベントは次のとおりです。

イベント	説明
SERVICE_REGISTER	リスナーがインスタンスの登録情報を受信した。
SERVICE_UPDATE	リスナーが特定インスタンスについて更新された登録情報（ディスパッチャやインスタンスのロード情報など）を受信した。
SERVICE_DIED	リスナーから PMON への接続が切断された。インスタンスの登録情報はすべて廃棄されます。PMON によって再度登録されるまで、クライアントはそのインスタンスには接続できません。

**サービス登録情報の書式** サービス登録イベントは次のフィールドで構成されます。

Timestamp \* Event \* Instance Name \* Return Code

サービス登録フィールドのプロパティは次のとおりです。

- 各フィールドはアスタリスク（\*）で区切られます。
- 通常は、1つのインスタンスのイベントが複数回連続して記録されます。
- 正常に登録されると、コード 0 が戻ります。これは、クライアントがインスタンスに接続できたことを示します。
- 失敗した場合は、エラー・メッセージに対応するコードが生成されます。

**関連項目：** 一般的な Net8 エラーの詳細は、12-5 ページの「[一般的なエラー・メッセージの解決](#)」、すべてのエラー・メッセージを網羅したリストは、『Oracle8i エラー・メッセージ』を参照してください。



例 12-4 は、サービス登録イベントが記録されたログ・ファイルを示します。リスナーは正常な SERVICE\_REGISTER イベントの後はクライアント要求を受信できますが、SERVICE\_DIED イベントの後はクライアント要求を受信できない点に注意してください。

#### 例 12-4 サービス登録イベントが記録されたリスナー・ログ

```
-----
10-AUG-1999 15:28:43 * service_register * sales * 0
10-AUG-1999 15:28:43 * service_register * sales * 0
10-AUG-1999 15:28:58 *
(CONNECT_DATA=(service_name=sales.us.acme.com) (CID=(PROGRAM=) (HOST=dlsun1013)
(USER=jdoe)))
* (ADDRESS=(PROTOCOL=tcp) (HOST=144.25.185.246) (PORT=41349)) * establish
* sales.us.acme.com * 0
10-AUG-1999 15:38:44 * service_update * sales * 0
10-AUG-1999 15:38:44 * service_update * sales * 0
10-AUG-1999 15:48:45 * service_update * sales * 0
10-AUG-1999 15:48:45 * service_update * sales * 0
10-AUG-1999 15:50:57 *
(CONNECT_DATA=(service_name=sales.us.acme.com) (CID=(PROGRAM=) (HOST=dlsun1013) (U
SER=jdoe)))
* (ADDRESS=(PROTOCOL=tcp) (HOST=144.25.185.246) (PORT=41365)) * establish
* sales.us.acme.com * 0
10-AUG-1999 15:51:26 * service_died * sales * 12537
10-AUG-1999 15:51:26 * service_died * sales * 12537
10-AUG-1999 15:52:06 *
(CONNECT_DATA=(service_name=sales.us.acme.com) (CID=(PROGRAM=) (HOST=dlsun1013) (U
SER=jdoe)))
* (ADDRESS=(PROTOCOL=tcp) (HOST=144.25.185.246) (PORT=41406)) * establish
* sales.us.acme.com * 12514
TNS-12514: TNS:listener could not resolve SERVICE_NAME given in connect
descriptor
-----
```

### Direct Hand-Off 情報

リスナーでは、MTS ディスパッチャへの Direct Hand-Off イベントが記録されます。このイベントは次のフィールドで構成されます。

*Timestamp \* Presentation \* Handoff \* Error Code*

Direct Hand-Off フィールドのプロパティは次のとおりです。

- 各フィールドはアスタリスク (\*) で区切られます。
- 成功した接続またはコマンドには、コード 0 が戻ります。
- 失敗した場合は、エラー・メッセージに対応するコードが生成されます。



**関連項目：** 一般的な Net8 エラーの詳細は、12-5 ページの「[一般的なエラー・メッセージの解決](#)」、すべてのエラー・メッセージを網羅したリストは、『Oracle8i エラー・メッセージ』を参照してください。

ログ・ファイルの Direct Hand-Off イベントの例を[例 12-5](#)に示します。

### 例 12-5 Direct Hand-Off のリスナー・ログ・イベント

```
21-MAY-1999 10:54:55 * oracle.aurora.net.SALESHttp2 * handoff * 0
```

## Oracle Connection Manager ログの理解

Oracle Connection Manager の CMGW ゲートウェイ・プロセスにより作成されるログ・ファイル名は、UNIX では `cman_pid.log`、Windows NT では `cmanpid.log` です。CMADMIN 管理プロセスにより作成されるログ・ファイル名は、`cmadm_pid.log` です。このログ・ファイルの場所は、UNIX では `$ORACLE_HOME/network/log`、Windows NT では `ORACLE_HOME\network\log` です。

### 例 12-6 cman\_pid.log

```
(TIMESTAMP=30-OCT-98 18:03:10) (EVENT=10) (VERSION=8.1.6.0.0)
(TIMESTAMP=30-OCT-98 18:03:10) (EVENT=36) (rule_list=
(rule=(src=spcstn) (dst=x) (srv=x) (act=accept)))
(TIMESTAMP=30-OCT-98 18:03:10) (EVENT=32) (PARAMETER_LIST=(MAXIMUM_RELAYS=1024) (RELAY_
STATISTICS=no) (AUTHENTICATION_LEVEL=0) (LOG_LEVEL=1) (SHOW_TNS_INFO=no) (ANSWER_
TIMEOUT=0) (MAXIMUM_CONNECT_DATA=1024) (USE_ASYNC_CALL=yes) (TRACING=no) (TRACE_
DIRECTORY=default) (MAX_FREELIST_BUFFERS=0))
(TIMESTAMP=30-OCT-98 18:03:10) (EVENT=34) (ADDRESS_LIST=
(ADDRESS=(PROTOCOL=tcp) (HOST=) (PORT=1610) (QUEUESIZE=32)))
(TIMESTAMP=30-OCT-98 18:03:12) (EVENT=38) (COMMAND=2)
(TIMESTAMP=30-OCT-98
18:03:27) (EVENT=26) (RLYNO=0) (SRC=(ADDRESS=(PROTOCOL=tcp) (HOST=spcstn.us.oracle.com) (
PORT=34758))) (DST=(ADDRESS=(PROTOCOL=tcp) (HOST=144.25.187.89) (PORT=1581)))
(TIMESTAMP=30-OCT-98 18:03:43) (EVENT=28) (RLYNO=0) (SINCE=30-OCT-98
18:03:27) (STATISTICS=(IN=(BYTES=0) (PACKETS=0) (DCDS=0) (OUT=(BYTES=0) (PACKETS=0) (DCDS=
0))))
```

### 例 12-7 cmadm\_pid.log

```
(TIMESTAMP=30-OCT-98 18:03:09) (EVENT=Sent Admin Status to UI)
(TIMESTAMP=30-OCT-98 18:03:10) (EVENT=CMan Registration)
```



## cman\_pid.log のイベント・コード

UNIX の `cman_pid.log`、および Windows NT の `cmanpid.log` では、イベントはイベント・コード番号でレポートされます。レポートされるイベント・コードは、`cman.ora` ファイルの `LOG_LEVEL` パラメータ、または CMCTL の [SET LOG\\_LEVEL](#) コマンドで設定されたログ・レベルによって異なります。この項では、各イベント・コードの意味を説明します。

### ログ・レベル1のイベント

コード	説明
10	ゲートウェイの起動中
12	ゲートウェイのシャットダウン中
14	TNS アドレスでリスニングを実行中
18	応答エラー <b>関連項目:</b> 12-26 ページの「 <a href="#">イベント・コード 18 の原因</a> 」を参照してください。
20	受信コールを拒否 <b>関連項目:</b> 12-26 ページの「 <a href="#">イベント・コード 20 の原因</a> 」を参照してください。
26	リレーは現在オープンしている
28	リレーが現在クローズしている
30	統計情報レポート
32	<parameter_list>
34	<address_list>
36	<rule_list>
38	CMCTL コマンド
40	ゲートウェイがビジーのため、CMCTL コマンドを拒否
42	デッド接続を検出
44	リレーのタイムアウト
11	<address_list> の引数に誤りがある
13	<parameter_list> の引数に誤りがある
15	<rule_list> の引数に誤りがある
23	CMCTL レコードの不良
25	コマンドライン引数が長すぎる



コード	説明
27	メモリーの割当てエラー
29	TNS エラー
31	CMCTL 要求の処理中に TNS エラーが発生

**イベント・コード 18 の原因** 応答エラーには次の原因が考えられます。

コード	説明
1	タイムアウト
2	接続データ・バッファの容量不足
3	TNS による拒否
4	TNS パケットのチェックサム・エラー

**イベント・コード 20 の原因** 受信コールが拒否されるのは次の場合です。

コード	説明
1	ゲートウェイのシャットダウン中
1	ゲートウェイはオフライン
3	受信コールに接続データが含まれていない
4	受信コールの接続データに誤りがある
5	すべてのリレーが使用中
6	リレー・バッファを取得できない
7	TNS の致命的エラー
8	ASO サービスが使用不可
9	規則でのフィルタリングによる拒否
10	送信コール・エラー
11	Net8/TNS による拒否
12	リスナーが動作していない
13	リスナーに到達できていない



コード	説明
14	ホスト名検索エラー
15	プロトコル・アダプタ（およびプロトコル・スタック）がロードされていない
16	SOURCE_ROUTE が設定されていない
17	規則または接続文字列データ不良のため拒否

#### ログ・レベル 2 のイベント

コード	説明
102	受信コールへの応答
104	送信コールの作成
105	受信コールの受入れ
106	規則合致レポート

#### ログ・レベル 3 のイベント

コード	説明
202	コールがブロックされます（非同期 TNS サポート されていません）
204	リレーのブロック <b>関連項目：</b> <a href="#">「イベント・コード 204 の原因」</a> を参照してください。
206	バッファに残余データが含まれている

#### イベント・コード 204 の原因

コード	説明
1	書き込み側の準備を待機中
2	書き込み側の未処理分消去を待機中
3	受信時に WOULDBLOCK エラー発生
4	送信時に WOULDBLOCK または PARTIAL エラー発生



コード	説明
5	WOULDBLOCK または PARTIAL 送信エラー再発生

ログ・レベル 4 のイベント

コード	説明
302	このバイト数を読み取る
304	このバイト数を書き込む
306	再試行時にこのバイト数を書き込む

# エラー情報のトレース

トレース機能は、実行されたネットワーク・イベントが記述される一連の詳細文を生成します。操作をトレースすると、ログ・ファイルにある情報より多くの Net8 コンポーネントの内部操作に関する情報が得られます。この情報は、ファイル出力されエラーの原因となったイベントを確認できます。

**注意：** トレース機能は大量のディスク容量を消費するため、システムのパフォーマンスに大きな影響を与える可能性があります。したがって、必要ときにのみトレースを行うようにしてください。

この項の内容は、次のとおりです。

- [トレース・ファイルの名前](#)
- [トレース・パラメータの理解と設定](#)
- [制御ユーティリティ実行時のトレース設定](#)
- [Net8 トレースの評価](#)
- [Trace Assistant を使用したトレース・ファイルの検査](#)

## トレース・ファイルの名前

各 Net8 コンポーネントは、独自のトレース・ファイルを作成します。次の表は、デフォルトのファイル名とそれに含まれている情報の説明を示したものです。



トレース・ファイル	エラー情報の内容
sqlnet.trc	クライアント
svr_pid.trc	サーバー
listener.trc	リスナー
names.trc	Oracle Names サーバー
cman_pid.trc (UNIX の場合)	Oracle Connection Manager CMGW ゲートウェイ・プロセス
cmanpid.trc (Windows NT の場合)	
cmadm_pid.trc (UNIX の場合)	Oracle Connection Manager CMADMIN 管理プロセス
cmadmpid.trc (Windows NT の場合)	

## トレース・パラメータの理解と設定

ロギングされる情報の種類と量、ファイルが格納されている位置など、トレース機能を制御するパラメータは、次のように各ネットワーク・コンポーネントの構成ファイル内に設定されます。

トレース対象のネットワーク・コンポーネント	パラメータを設定する構成ファイル
クライアント	sqlnet.ora
サーバー	sqlnet.ora
リスナー	listener.ora
Oracle Names サーバー	names.ora
Oracle Connection Manager プロセス	cman.ora

この項の内容は、次のとおりです。

- [sqlnet.ora](#) ファイルのパラメータ
- [listener.ora](#) ファイルのパラメータ
- [names.ora](#) ファイルのパラメータ
- [cman.ora](#) ファイルのパラメータ
- [構成ファイルのトレース・パラメータ設定](#)



**関連項目：** これらのパラメータの詳細は、[付録 C](#) を参照してください。

**sqlnet.ora ファイルのパラメータ**

sqlnet.ora ファイルに設定できるパラメータは次のとおりです。

sqlnet.ora ファイルのパラメータ	Oracle Net8 Assistant のフィールド	説明
TRACE_LEVEL_CLIENT	「Client Information: Trace Level」	<p>トレース機能によりクライアントで記録される詳細レベルを示す。トレース・レベルは、0 から 16 までの数値（0 はトレースなし、16 は最大限のトレースを表す）か、OFF、ADMIN、USER または SUPPORT の値で指定します。</p> <ul style="list-style-type: none"><li>■ OFF（0 と同じ）はトレースを実行しません。</li><li>■ USER（4 と同じ）はユーザーによるエラー条件を特定するためのトレースを実行します。</li><li>■ ADMIN（6 と同じ）はインストレーション固有の問題を特定するためのトレースを実行します。</li><li>■ SUPPORT（16 と同じ）はサポートに対してトラブルシューティングのためのトレース情報を提供します。</li></ul>
TRACE_DIRECTORY_CLIENT	「Client Information: Trace Directory」	<p>クライアント・トレース・ファイルの作成先となるディレクトリを設定します。デフォルトのクライアント・ディレクトリは、UNIX では \$ORACLE_HOME/network/trace、Windows NT では ORACLE_HOME¥network¥trace です。</p>
TRACE_FILE_CLIENT	「Client Information: Trace File」	<p>クライアントのトレース・ファイルの名前を設定します。デフォルトのトレース・ファイル名は sqlnet.trc です。</p>



sqlnet.ora ファイルのパラメータ	Oracle Net8 Assistant のフィールド	説明
TRACE_UNIQUE_CLIENT	「Client Information: Unique Trace File Name」	<p>クライアント・トレース・ファイルごとに一意のトレース・ファイル名を作成するかどうかを決定します。値を ON に設定すると、複数のトレース・ファイルが共存できるように、各トレース・ファイルの名前にプロセス識別子が付加されます。値を OFF に設定すると、クライアントで新しいトレース・ファイルが作成されるときに、既存のファイルが上書きされます。</p> <p>このパラメータで作成される一意のトレース・ファイル名は、sqlnetpid.trc です。</p>
TRACE_LEVEL_SERVER	「Server Information: Trace Level」	<p>トレース機能によりサーバーで記録される詳細レベルを示す。トレース・レベルは、0 から 16 までの数値（0 はトレースなし、16 は最大限のトレースを表す）か、OFF、ADMIN、USER または SUPPORT の値で指定します。</p> <ul style="list-style-type: none"> <li>■ OFF（0 と同じ）はトレースを実行しません。</li> <li>■ USER（4 と同じ）はユーザーによるエラー条件を特定するためのトレースを実行します。</li> <li>■ ADMIN（6 と同じ）はインストレーション固有の問題を特定するためのトレースを実行します。</li> <li>■ SUPPORT（16 と同じ）はサポートに対してトラブルシューティングのためのトレース情報を提供します。</li> </ul>
TRACE_DIRECTORY_SERVER	「Server Information: Trace Directory」	<p>サーバー・トレース・ファイルの作成先となるディレクトリを設定します。デフォルトのサーバー・ディレクトリは、UNIX では \$ORACLE_HOME/network/trace、Windows NT では ORACLE_HOME\network\trace です。</p>
TRACE_FILE_SERVER	「Server Information: Trace File」	<p>クライアントのトレース・ファイルの名前を設定します。デフォルトのトレース・ファイル名は svr_pid.trc です。</p>

sqlnet.ora には、次に示す TNSPING ユーティリティのオプション・トレース・パラメータを手動で追加することもできます。TNSPING では、Net8 ネットワーク上のサービス（データベース、Oracle Names サーバーまたはその他の TNS サービスなど）が正常に到達できるかどうかを判定します。



sqlnet.ora ファイルのパラメータ	説明
TNSPING.TRACE_LEVEL	トレース機能により TNSPING ユーティリティで記録される詳細レベルを示す。トレース・レベルは、0 から 16 までの数値（0 はトレースなし、16 は最大限のトレースを表す）か、OFF、ADMIN、USER または SUPPORT の値で指定します。 <ul style="list-style-type: none"><li>OFF（0 と同じ）はトレースを実行しません。</li><li>USER（4 と同じ）はユーザーによるエラー条件を特定するためのトレースを実行します。</li><li>ADMIN（6 と同じ）はインストレーション固有の問題を特定するためのトレースを実行します。</li><li>SUPPORT（16 と同じ）はサポートに対してトラブルシューティングのためのトレース情報を提供します。</li></ul>
TNSPING.TRACE_DIRECTORY	TNSPING トレース・ファイルの作成先となるディレクトリを設定します。デフォルトのディレクトリは、UNIX では \$ORACLE_HOME/network/trace、Windows NT では ORACLE_HOME¥network¥trace です。

listener.ora ファイルのパラメータ

listener.ora ファイルに設定できるトレース・パラメータは次のとおりです。

listener.ora ファイルのパラメータ	Oracle Net8 Assistant のフィールド	説明
TRACE_LEVEL_listener_name	「Trace Level」	トレース機能によりリスナーで記録される詳細レベルを示す。トレース・レベルは、0 から 16 までの数値（0 はトレースなし、16 は最大限のトレースを表す）か、OFF、ADMIN、USER または SUPPORT の値で指定します。 <ul style="list-style-type: none"><li>OFF（0 と同じ）はトレースを実行しません。</li><li>USER（4 と同じ）はユーザーによるエラー条件を特定するためのトレースを実行します。</li><li>ADMIN（6 と同じ）はインストレーション固有の問題を特定するためのトレースを実行します。</li><li>SUPPORT（16 と同じ）はサポートに対してトラブルシューティングのためのトレース情報を提供します。</li></ul>



listener.ora ファイルのパラメータ	Oracle Net8 Assistant のフィールド	説明
TRACE_DIRECTORY_ listener_name	「Trace File」	トレース・ファイルの作成先となるディレクトリとファイルを設定します。デフォルトのディレクトリは、UNIX では \$ORACLE_HOME/network/trace、Windows NT では ORACLE_HOME¥network¥trace です。デフォルトのファイル名は listener.trc です。
TRACE_FILE_listener_name		

names.ora ファイルのパラメータ

names.ora ファイルに設定できるトレース・パラメータは次のとおりです。

names.ora ファイルのパラメータ	Oracle Net8 Assistant のオプション	説明
NAMES.TRACE_DIRECTORY	「Trace Directory」	トレース・ファイルの作成先となるディレクトリを設定します。デフォルトのディレクトリは、UNIX では \$ORACLE_HOME/network/trace、Windows NT では ORACLE_HOME¥network¥trace です。
NAMES.TRACE_FILE	「Trace File」	トレース・ファイルの名前を設定します。デフォルトのトレース・ファイル名は names.trc です。
NAMES.TRACE_LEVEL	該当せず	<p>トレース機能により Oracle Names サーバーで記録される詳細レベルを示す。トレース・レベルは、0 から 16 までの数値（0 はトレースなし、16 は最大限のトレースを表す）か、OFF、ADMIN、USER または SUPPORT の値で指定します。</p> <ul style="list-style-type: none"><li>■ OFF（0 と同じ）はトレースを実行しません。</li><li>■ USER（4 と同じ）はユーザーによるエラー条件を特定するためのトレースを実行します。</li><li>■ ADMIN（6 と同じ）はインストレーション固有の問題を特定するためのトレースを実行します。</li><li>■ SUPPORT（16 と同じ）はサポートに対してトラブルシューティングのためのトレース情報を提供します。</li></ul>



names.ora ファイルのパラメータ	Oracle Net8 Assistant のオプション	説明
NAMES.TRACE_UNIQUE	「Make Trace File Unique」	値を ON に設定すると、複数のトレース・ファイルが共存できるように、トレース・セッションごとに一意なファイル名が作成されます。生成されるトレース・ファイルの名前には、プロセス識別子が付加されます。  このパラメータで作成される一意のトレース・ファイル名は、namespid.trc です。

**cman.ora ファイルのパラメータ**

cman.ora ファイルに設定できるトレース・パラメータは次のとおりです。

cman.ora ファイルのパラメータ	説明
TRACING	Oracle Connection Manager でのトレースを使用可能にするかどうかを決定します。  YES を設定すると、Oracle Connection Manager でのトレースを使用可能にします。CMGW ゲートウェイ・プロセスにより作成されるトレース・ファイル名は、UNIX では cman_pid.trc、Windows NT では cmanpid.trc です。 CMADMIN 管理プロセスにより作成されるトレース・ファイル名は、UNIX では cmadm_pid.trc、Windows NT では cmadmpid.trc です。
TRACE_DIRECTORY	トレース・ファイルの作成先となるディレクトリを設定します。  デフォルトのディレクトリは、UNIX では \$ORACLE_HOME/network/trace、Windows NT では ORACLE_HOME¥network¥trace です。

**構成ファイルのトレース・パラメータ設定**

sqlnet.ora、listener.ora および names.ora のトレース・パラメータは Oracle Net8 Assistant で設定できます。cman.ora のトレース・パラメータは手動で設定する必要があります。

**関連項目：** C-65 ページの「[Oracle Connection Manager パラメータ \(cman.ora\)](#)」を参照してください。

ロギング・パラメータを設定するには、次の手順に従います。

- 1. Oracle Net8 Assistant を起動します。



- UNIX の場合、\$ORACLE\_HOME/bin から netasst を実行します。
  - Windows NT の場合、「スタート」ボタン -> 「プログラム」 -> 「Oracle - HOME\_NAME」 -> 「Network Administration」 -> 「Net8 Assistant」を選択します。
2.   トレース・パラメータを指定します。

トレース・ファイル	トレース・パラメータの設定方法
sqlnet.trc( クライアント用 )	1.   ナビゲータで、「Local」 -> 「Profile」を展開します。
svr_pid.trc( サーバー用 )	2.   右ペインのリストから「General」を選択します。
	3.   「Tracing」タブをクリックします。
	4.   設定を指定します。
listener.trc	1.   ナビゲータで、「Local」 -> 「Listeners」を展開します。
	2.   リスナーを選択します。
	3.   右ペインのリストから「General」を選択します。
	4.   「Logging and Tracing」タブをクリックします。
	5.   設定を指定します。
names.trc	1.   ナビゲータで、「Oracle Names Servers」フォルダを展開します。
	2.   Oracle Names サーバーを選択します。
	3.   右ペインのリストから「Configure Server」を選択します。
	4.   「Adv.」タブをクリックします。
	5.   トレース・ディレクトリとファイル名を指定します。

3.   「File」 -> 「Save Network Configuration」を選択します。

制御ユーティリティ実行時のトレース設定

トレースは、制御ユーティリティの実行中に設定できます。制御ユーティリティを使用してトレースを設定しても、\*.ora ファイルにパラメータは設定されません。設定は、制御ユーティリティの現在のセッションでのみ有効です。

- リスナーの場合、LSNRCTL 制御ユーティリティから SET TRC\_FILE や SET TRC\_DIRECTORY、SET TRC\_LEVEL コマンドを実行します。実行方法は A-2 ページの「[リスナー制御ユーティリティ \(LSNRCTL\)](#)」を参照してください。
- Oracle Names サーバーの場合は、NAMESCTL 制御ユーティリティから SET TRACE\_FILE\_NAME や SET TRACE\_LEVEL コマンドを実行します。実行方法は A-25 ページの



「[Oracle Names 制御ユーティリティ \(NAMESCTL\)](#)」を参照してください。あるいは次のように Oracle Net8 Assistant から設定します。

1. Oracle Net8 Assistant を起動します。  
-UNIX の場合、\$ORACLE\_HOME/bin で netasst を実行します。  
-Windows NT の場合、「スタート」ボタン->「プログラム」->「Oracle - HOME\_NAME」->「Network Administration」->「Net8 Assistant」を選択します。
2. ナビゲータで、「Oracle Names サーバ s」を展開します。
3. Oracle Names サーバを選択します。
4. 右ペインのリストから「Manage Server」を選択します。
5. 「Logging」タブをクリックします。
6. トレース・レベル、ディレクトリおよびファイル名を指定します。
7. 「File」->「Save Network Configuration」を選択します。

---

---

**注意：** Oracle Connection Manager の場合、トレースの設定は Connection Manager 構成ファイルからのみ行うことができます。詳細は C-65 ページの「[Oracle Connection Manager パラメータ \(cman.ora\)](#)」を参照してください。

---

---

## Net8 トレースの評価

手動で、または Trace Assistant Tool を使用して、トレース・ファイルを評価すると、次のことが理解でき、ネットワークの診断やトラブルシューティングが容易に行うことができます。

- ネットワーク・ノード間のパケットの流れ
- Net8 のどのコンポーネントが障害に陥っているか
- 関連するエラー・コード

### ネットワーク・ノード間のデータ・パケットの流れの理解

Net8 は、データ・パケットを送受信することにより、その機能を実行します。SUPPORT のトレース・レベルを指定すると、トレース・ファイル内の Net8 パケットの実際の内容を参照できます。送受信したパケット・タイプの順序により、接続を確立した方法を確認できます。

**データ・パケット形式の理解** トレース・ファイルの各行は、後ろにメッセージが続くプロシージャで始まります。各プロシージャの後には、実際のデータを示す 16 進データの行があ



ります。パケットの内部をフローする実際のデータは、16 進データの右側に表示される場合があります。

次の表は、Net8 パケット・キーワードの一覧と、それらのキーワードが表すパケット・タイプを示したものです。

キーワード	パケット・タイプ
NSPTCN	Connect
NSPTAC	Accept
NSPTRF	Refuse
NSPTRS	Resend
NSPDA	データ
NSPCNL	Control
NSPTMK	Marker

**注意：** Oracle ネットワーク製品または EBCDIC データを通じて暗号化を使用している場合、このデータは表示されません。

たとえば、次の行はネットワークを通じて NSPTCN パケットを送る nscon と呼ばれるプロセスを記述しています。

```
nscon: sending NSPTCN packet
```

各パケットには、パケット・タイプを示すキーワードがあります。すべてのパケット・タイプは、接頭辞 NSP から始まります。特定のパケット情報についてトレース・ファイルを検討する場合、このことを覚えておくと便利です。

例 12-8 は、一般的なパケット情報を示したものです。

#### 例 12-8 パケット情報

```
nscon: entry
nscon: doing connect handshake...
nscon: sending NSPTCN packet
npsend: entry
npsend: plen=187, type=1
npsend: 187 bytes to transport
npsend:packet dump
```



```
nspsend:00 BB 00 00 01 00 00 00 |.....|
nspsend:01 33 01 2C 0C 01 08 00 |.3.,....|
nspsend:7F FF 7F 08 00 00 00 01 |.....|
nspsend:00 99 00 22 00 00 08 00 |..."....|
nspsend:01 01 28 44 45 53 43 52 |..(DESCR|
nspsend:49 50 54 49 4F 4E 3D 28 |IPTION=(|
nspsend:43 4F 4E 4E 45 43 54 5F |CONNECT_|
nspsend:44 41 54 41 3D 28 53 49 |DATA=(SI|
nspsend:44 3D 61 70 33 34 37 64 |D=ap347d|
nspsend:62 31 29 28 43 49 44 3D |b1)(CID=|
nspsend:28 50 52 4F 47 52 41 4D |(PROGRAM|
nspsend:3D 29 28 48 4F 53 54 3D |=)(HOST=|
nspsend:61 70 32 30 37 73 75 6E |ap207sun|
nspsend:29 28 55 53 45 52 3D 6D |)(USER=m|
nspsend:77 61 72 72 65 6E 29 29 |warren))|
nspsend:29 28 41 44 44 52 45 53 |)(ADDRESS|
nspsend:53 5F 4C 49 53 54 3D 28 |S_LIST=(|
nspsend:41 44 44 52 45 53 53 3D |ADDRESS=|
nspsend:28 50 52 4F 54 4F 43 4F |(PROTOCO|
nspsend:4C 3D 74 63 70 29 28 48 |L=tcp)(H|
nspsend:4F 53 54 3D 61 70 33 34 |OST=ap34|
nspsend:37 73 75 6E 29 28 50 4F |7sun)(PO|
nspsend:52 54 3D 31 35 32 31 29 |RT=1521)|
nspsend:29 29 29 00 00 00 00 00 |))).....|
nspsend: normal exit
nscon: exit (0)
```

### 関連エラー出力の理解

Net8 で接続に問題が生じるたびに、<ERROR> または <FATAL> の接頭辞を持つトレース・ファイルにエラー・コードが記録されます。[例 12-9](#) は、一般的なトレース・ファイルのエラー出力を示したものです。

#### 例 12-9 トレース・ファイルのエラー出力

```
nspsend: entry
nspsend: plen=244, type=6
ntpwr: entry
ntpwr: exit
-<ERROR>- nspsend: transport write error
nspsend: error exit
nserror: entry
-<ERROR>- nserror: nsres: id=0, op=65, ns=12541, ns2=12560; nt[0]=511,
nt[1]=61,nt[2]=0
-<ERROR>- nsopen: unable to open transport
nricdt: Call failed...
nricdt: exit
```



```

-<ERROR>- osnqper: error from nricall
-<ERROR>- osnqper: nr err code: 12203
-<ERROR>- osnqper: ns main err code: 12541
-<ERROR>- osnqper: ns (2) err code: 12560
-<ERROR>- osnqper: nt main err code: 511
-<ERROR>- osnqper: nt (2) err code: 61
-<ERROR>- osnqper: nt OS err code: 0
osnqme: entry
osnqme: reporting nr (1) error: (12203) as rdbms err (12203)
osnqme: exit
-<ERROR>- onstns: Couldn't connect, returning 12203
nricall: Exiting NRICALL with following termination result -1
nricall: exit
osnqme: entry
osnqme: reporting nr (1) error: (12203) as rdbms err (12203)
osnqme: exit
-<ERROR>- onstns: Couldn't connect, returning 12203
-<ERROR>- osnqper: error from nricall

```

エラー・コードを評価する最も能率的な方法は、記録されている最新の NS エラー・コードを評価することです。これは、セッション・レイヤーが接続を制御しているためです。最も重要なエラー・メッセージは、ファイルの最後部にあるメッセージです。それらは最新のエラーであり、接続上の問題の原因を示しています。

特定のリターン・コードについての詳細は、Oracle UNIX エラー・ツール `oerr` を使用してください。コマンドライン・プロンプトで次のように入力し、`oerr` ツールを使用して Net8 リターン・コードに関するさらに詳細な情報を検索してください。

```
oerr tns error_number
```

## Trace Assistant を使用したトレース・ファイルの検査

Net8 には、既存のトレース・ファイル・テキスト行をさらに読みやすいパラグラフに変換して、トレース・ファイルの情報を理解しやすくする Trace Assistant というツールがあります。Trace Assistant は、レベル 16 (SUPPORT) の SQL\*Net または Net8 トレース・ファイルにのみ対応します。

Trace Assistant を実行するには、コマンドライン・プロンプトで次のように入力します。

```
trcasst [options] filename
```

オプションの詳細は次のとおりです。



オプション	説明
-o	接続性と Two Task Common (TTC) 情報を表示します。-o に続いて、次のオプションを使用できます。 <ul style="list-style-type: none"><li>■ c (サマリー接続情報)</li><li>■ d (詳細接続情報)</li><li>■ u (サマリー TTC 情報)</li><li>■ t (詳細 TTC 情報)</li><li>■ q (サマリー TTC 情報を拡張する SQL コマンドを表示します)</li></ul>
-p	オラクル社の内部使用オプション。
-s	統計情報を表示します。
-e	エラー情報の表示を可能にします。-e に続いて、次のエラー・レベルを指定できます。 <ul style="list-style-type: none"><li>■ 0 またはなし (nserror 機能からダンプされた NS エラー番号を変換し、他のエラーすべてをリストします)</li><li>■ 1 (nserror 機能から NS エラー変換のみを表示します)</li><li>■ 2 (エラー番号を変換しないで表示します)</li></ul>
オプションが指定されない場合のデフォルトは、-odt -e -s (エラー・レベル 0 で、詳細接続情報および詳細 Two-Task Common 情報、統計情報を表示) です。	

例 12-10 は、Trace Assistant がどのようにトレース・ファイル情報をさらに読みやすい形式に変換するかを示したものです。



## 例 12-10 Trace Assistant の一般的な変換結果

トレース・ファイル	オプション -e0 または -e1 を指定した Trace Assistant による変換結果
nsc2addr: normal exit nsopen: entry nsmal: 404 bytes at 0x10d5a48 nsopen: opening transport... -<ERROR>- ntus2err: sd=13, op=1, resnt[0]=511, resnt[1]=2, resnt[2]=0 -<ERROR>- nserror: nsres: id=0, op=65, ns=12541, ns2=12560; nt[0]=511, nt[1]=2, nt[2]=0 -<ERROR>- nsopen: unable to open transport	Error found. Error Stack follows: id: 00000 Operation code: 00065 NS Error 1: 12541 NS Error 2: 12560 NT Generic Error: 00511 Protocol Error: 00146 OS Error: 00000  NS & NT Errors Translation 12541, 00000, "TNS:no listener" // *Cause: The connection request could not be completed because the listener // is not running. // *Action: Ensure that the supplied destination address matches one of // the addresses used by the listener - compare the tnsnames.ora entry with // the appropriate listener.ora file (or tnsnav.ora if the connection is to // go by way of an Interchange). Start the listener on the remote machine. / 12560, 00000, "TNS:protocol adapter error" // *Cause: A generic protocol adapter error occurred. // *Action: Check addresses used for proper protocol specification. Before // reporting this error, look at the error stack and check for lower level // transport errors. For further details, turn on tracing and re-execute the // operation. Turn off tracing when the operation is complete. / 00511, 00000, "No listener" // *Cause: The connect request could not be completed because no application // is listening on the address specified, or the application is unable to // service the connect request in a sufficiently timely manner. // *Action: Ensure that the supplied destination address matches one of // the addresses used by the listener - compare the tnsnames.ora entry with // appropriate listener.ora file. // Start the listener on the remote machine.

ただし、トレース・ファイル内に nserror 機能からログに記録されなかった他のエラーが存在する場合もあります。



## Net8 パケットでネットワークを横断する情報の理解

Trace Assistant では、Net8 と **Two-Task Common (TTC)** 通信レイヤーの両方からのデータ・パケットを表示できます。Trace Assistant には、これらのパケットを表示するための2つのオプションがあります。

- サマリー接続性 (-oc オプションを使用)
- 詳細接続性 (-od オプションを使用)

### Net8 パケットの例

次の例は、様々なトランザクションで各種パケットが Net8 レイヤーとの間で送受信される時、Trace Assistant がどのようにそれらのパケットを表示するかを示したものです。

- bequeath 接続
- リダイレクト接続
- データ・パケット

送信パケットまたは受信パケットには接頭辞 ---> Send *nnn* bytes または <--- Received *nnn* bytes があることに注意してください。この接頭辞は、このノードが *nnn* バイトからなる特定タイプのパケットを送信または受信していることを表します。これらの接頭辞によって、ノードがクライアントまたはサーバーのどちらであるかが判断できます。接続要求は常にクライアントによって送信され、サーバー（またはリスナー）によって受信されます。

### 例 12-11 bequeath 接続で送られたサマリー・データ・パケット

---

#### trcasst -oc <filename> を使用する場合

---

この例には2つのパケットが示されています。最初のパケットは、クライアントからリスナーに送信される接続パケットです。2番目のパケットは、サーバーから戻される受諾パケットです。

```
---> Send      192 bytes - Connect packet
      Connect data length: 142
      (description=(address=(protocol=tcp) (host=dlsun) (port=1521)) (connect_
data=(sid=db1) (cid=(program=) (host=dlsun) (user=use1))))
<--- received 24 bytes - accept packet
      Accept data length: 0
```

---



---

**例 12-12 bequeath 接続で送られた詳細データ・パケット**

---

**trcasst -od <filename> を使用する場合**

---

この例は、接続交渉中に接続データとともに送信されたすべての詳細を示しています。

```
---> Send      50 bytes - Connect packet
Current NS version number is: 309.
Lowest NS version number can accommodate is: 300.
Global options for the connection:
    can receive attention
    no attention processing
    Don't care
    Maximum SDU size: 2048
    Maximum TDU size: 5120
    NT protocol characteristics:
        Test for more data
        Spawner is running
        Hang on to Listener connection
        Full duplex I/O
        Urgent data support
        Generate SIGURG signal
        Handoff connection to another
    Line turnaround value: 0
    Connect data length: 234
    Connect data offset: 50
    Connect data maximum size: 2048
        Native Services wanted
        Native Services wanted
Cross facility item 1: 0
    Cross facility item 2: 0
    Connection id: 0x0000000000000000
    Packet data is in the following data packet
---> Send      244 bytes - Data packet
(DESCRIPTION=(ADDRESS=(PROTOCOL=beq) (PROGRAM=/private/oracle/bin/oracle) (ARGV0=oracle) (A
RGS=' (DESCRIPTION=(LOCAL=YES) (ADDRESS=(PROTOCOL=beq))) (DETACH=NO)) (CONNECT_
DATA=(CID=(PROGRAM=) (HOST=dlsun) (USER=use1))))))
<--- Received 24 bytes - Accept packet
    Accepted NS version number is: 307.
Global options for the connection:
    no attention processing
    Don't care
    Accepted maximum SDU size: 2048
    Accepted maximum TDU size: 4096
    Connect data length: 0
        Native Services wanted
        Native Services wanted
```

---



例 12-13 リダイレクト接続で送られたサマリー・データ・パケット

trcasst -oc <filename> を使用する場合

```
---> Send      187 bytes - Connect packet
      Connect data length: 153
(DESCRIPTION= (CONNECT_
DATA= (SID=ap347db1) (CID= (PROGRAM=) (HOST=apsun) (USER=use2))) (ADDRESS_
LIST= (ADDRESS= (PROTOCOL=tcp) (HOST=apsun) (PORT=1521))))
<--- Received 8 bytes - Resend packet
---> Send      187 bytes - Connect packet
      Connect data length: 153
(DESCRIPTION= (CONNECT_DATA= (SID=apdb1) (CID= (PROGRAM=) (HOST=apsun) (USER=use2))) (ADDRESS_
LIST= (ADDRESS= (PROTOCOL=tcp) (HOST=apsun) (PORT=1521))))
<--- Received 24 bytes - Accept packet
      Accept data length: 0
```

例 12-14 データ・パケット

trcasst -oc <filename> または -od <filename> を使用する場合

接続が確立された後、他のノードに送信されるデータが Two-Task Common 層から Net8 に与えられます。サマリー・ビューと詳細ビューはともに、同じサマリー情報を生成します。

```
---> Send      30 bytes - Data packet
<--- Received 201 bytes - Data packet
---> Send     439 bytes - Data packet
<--- Received 400 bytes - Data packet
```

Two Task Common パケットの例

TTC は、データベースに送信されるカーソルのオープン、行の選択、行の更新などの要求を処理します。すべての要求は、サーバーによって応答されます。ログオンを要求した場合は、その要求を完了させたデータベースから応答が戻ります。例 12-15 から 12-47 ページの例 12-17 までは、可能な情報のタイプを示したものです。

TTC のサマリー情報は、各行に 1 つのパケットでなく 2 つのパケットを表示するという点で、他の表示とは異なります。このように表示されるのは、TTC が動作するときの要求 / 応答のペアリング・プロセスを反映するためです。

例 12-15 Two-Task Common のサマリー情報

trcasst -ou <filename> を使用する場合

(O3LOGA)	1st half of challenge-response logon	80	78
(O3LOGON)	2nd half of challenge-response logon	97	59
(OOPEN)	# 1	21	16



例 12-15 Two-Task Common のサマリー情報

trcasst -ou <filename> を使用する場合			
(OPARSEX)	# 1	245	59
(OCLOSE)	# 1	17	11
(OVERSION)		29	16
(OOPEN)	# 2	21	16
(OALL7)	# 2 Parse Can Defn=2 Exec Fetch "SELECT A.V	268	100
(OOPEN)	# 3	21	16
(OALL7)	# 3 Parse Exec=1 "SELECT USER FROM SYS.DUAL	152	70
(OALL7)	# 3 Defn=1 Fetc	117	88
(OCLOSE)	# 3	17	11

各行では、最初に表示される項目が実際に作成される要求です。2 番目の項目は、それがトランザクションに関連する場合はカーソル番号です。3 番目の項目は、フラグ・リストまたは応答中の SQL コマンドです。フラグは、要求に次の特性があることを示します。

!PL/SQL = PL/SQL 要求ではない

COM = コミット

IOV = I/O ベクター取得

DEFN = 定義

EXEC = 実行

FETCH = フェッチ

CAN = 取消し

DESCSEL = select の詳解

DESCBND = バインドの詳解

BND = バインド

PARSE = 解析

EXACT = 強制

右端には、送受信されたバイト数が表示されます。



3 行目の OOPEN は、出力がどのように要求 / 応答ペアを表示するかを示す最初の例です。OOPEN は後ろに #1 を付けて表示されますが、それはクライアントからカーソルのオープン要求が送信され、サーバーがオープンしたカーソルの番号である 1 で応答していることを示します。要求 / 応答のペアは 1 行に配置されるため、このオプションと他の接続オプションを組み合わせ使用しないでください。

### 例 12-16 Two-Task Common のサマリー情報

---

#### trcasst -ot <filename> を使用する場合

---

```
start of user function (TTIFUN)
  1st half of challenge-response logon (O3LOGA)
    Username: applsys
    Terminal: ttyp5
    Machine: ap207sun
    System User: mwarren
    Process: 24459
    Program: aiap45@ap207sun (TNS interface)
return opi parameter (TTIRPA)
  OPI parameter: 3309B1A977A62A3C
start of user function (TTIFUN)
  2nd half of challenge-response logon (O3LOGON)
    Username: applsys
    Terminal: ttyp5
    Machine: ap207sun
    System User: mwarren
    Process: 24459
    Program: aiap45@ap207sun (TNS interface)
ORACLE function complete (TTIOER)
start of user function (TTIFUN)
  Open a cursor
return opi parameter (TTIRPA)
  Cursor #: 1
start of user function (TTIFUN)
  Parse and Execute (OPARSEX) Cursor # 1
alter session set nls_language= 'AMERICAN' nls_territory= 'AMERICA' nls_currency=
'$' nls_iso_currency= 'AMERICA' nls_numeric_characters= '.,' nls_date_format=
'DD-MON-YY' nls_date_language= 'AMERICAN' nls_sort= 'BINARY'
ORACLE function complete (TTIOER)
start of user function (TTIFUN)
  Close cursor (OCLOSE) Cursor # 1
V6 Oracle func complete (TTISTA)
  Succeeded
```

---



例 12-17 サマリー Two-Task の最上部にある詳細 SQL 情報

trcasst -ouq <filename> を使用する			
(O3LOGA)	1st half of challenge-response logon	180	78
(O3LOGON)	2nd half of challenge-response logon	197	59
(OOPEN)	# 1	21	16
(OPARSEX)	# 1 alter session set nls_language= 'AMERICAN' nls_territory= 'AMERICA' nls_currency= '\$' nls_iso_currency= 'AMERICA' nls_numeric_characters= '.,' nls_date_format= 'DD-MON-YY' nls_date_language= 'AMERICAN' nls_sort= 'BINARY'	245	59
(OCLOSE)	# 1	17	11
(O71SESOPN)	(get session ID)	47	18
(OOPEN)	# 1	21	16
(OVERSION)	Oracle7 Server Release 8.1.3.0.0 - Production Release with the distributed and parallel query optionsPL/SQL Release 2.1.4.0.0 - Production	29	157
(O71SESOPN)	(get session ID)	47	18

詳細 Two-Task オプションに自動的に与えられる詳細 SQL 情報を表示するには、サマリー Two-Task コマンドに q を追加します。

適切な統計に収集されたデータの分析

収集される統計の種類は、ネットワーク・パートナー間で送受信されたコール（TTC）、パケットおよびバイトの数に関するものです。次の例は、代表的なトレース・ファイルの統計を示したものです。

例 12-18 一般的なトレース・ファイルの統計

trcasst -s <filename> を使用する			
=====			
Trace File Statistics:			
-----			
Net8:			
Total Calls:	466 sent,	491 received,	423 upi
Total Bytes:	119214 sent,	86614 received	
Average Bytes:	255 sent,	176 received	
Maximum Bytes:	2048 sent,	2048 received	
GRAND TOTAL PACKETS	sent: 466	received: 491	



トレース・ファイルの例

次の例は、デコードされた完全なトレース・ファイルを示したものです。この例は、Oracle クライアント・ツールの SQL\*Plus で行った次の要求に対して作成されました。

```
connect scott/tiger@june
```

画面にはメッセージ「ORA-12154: TNS: サービス名を解決できませんでした。」が表示されました。

例 12-19 トレース・ファイルの例

説明	トレース・ファイルの情報
Trace Configuration Information セクション内のトレース・ファイルのトレース・レベルと位置に注意してください。	-- TRACE CONFIGURATION INFORMATION FOLLOWS --- New trace stream is "C:\ORANT\%network%\trace\%net8.trc" New trace level is 16 --- TRACE CONFIGURATION INFORMATION ENDS ---
Oracle Names コンポーネントはネット・サービス名 june.com を検出できません。クライアントがネット・サービス名 june に拡張子 com を追加している点に注意してください。	nnfotran: tnsname.ora entry for name "june.com" not found nnftqnm: Error querying june.com of attribute A.SMD errcode 406 nnfgrwsp: Query unsuccessful, skipping to next adapter
クライアントは、ネット・サービス名アドレスを解決するために Oracle Names サーバー oranamesrvr0 にアクセスしようとしています。	nnfgrwsp: Switching to ONAMES adapter nnfgrwsp: Original name: june nnfgrwsp: Qualified name: june.com nngsget_get_stream: looking for "(DESCRIPTION=(CONNECT_DATA=(RPC=ON)) (ADDRESS=(PROTOCOL=tcp) (HOST=oranamesrvr0) (PORT=1575)))" nngsget_get_stream: cache miss, opening new stream nngsnad_new_stream_addr: "(DESCRIPTION=(CONNECT_DATA=(RPC=ON)) (ADDRESS=(PROTOCOL=tcp) (HOST=oranamesrvr0) (PORT=1575)))" nngsget_get_stream: no caller address will be sent to callee



## 例 12-19 トレース・ファイルの例

説明	トレース・ファイルの情報
ネットワーク・ルーティング (NR) は、 orenamesrvr0 へのルーティングを行います。	<pre> nricall: entry nric2a: entry nric2a: Getting local community information nriglp: entry nriglp: Looking for local addresses setup by nrigla nriglp: No addresses in the preferred address list nriglp: exit nric2a: TNSNAV.ORA is not present. No local communities entry. nrigla: entry nrigla: Getting local address information nrigla: Simple address... nrigla: No community component so just use straight address nrigla: exit nridst: entry nridst: Resolving address to use to call destination or next hop nridst: Found destination address nridst: Local address nridst: Local destination community found nridst: exit nric2a: This is a local community access nric2a: exit nricall: Got routable address information  nricall: Making call with following address information: (DESCRIPTION=(CONNECT_ DATA=(RPC=ON) ) (ADDRESS=(PROTOCOL=tcp) (HOST=orenamesrvr0) (PORT =1575))) nricdt: entry nricdt: Calling with outgoing connect data (DESCRIPTION=(CONNECT_ DATA=(RPC=ON) ) (ADDRESS=(PROTOCOL=tcp) (HOST=orenamesrvr0) (PORT =1575))) </pre>
ネットワーク・セッション (NS) は、orenamesrvr0 へのセッションを設定します。	<pre> nscall: entry nscall: connecting... nsc2addr: entry nsc2addr: (DESCRIPTION=(CONNECT_ DATA=(RPC=ON) ) (ADDRESS=(PROTOCOL=tcp) (HOST=orenamesrvr0) (PORT =1575))) </pre>



例 12-19 トレース・ファイルの例

説明	トレース・ファイルの情報
ネットワーク・トランスポート（NT）は、トランスポート・セッションを設定します。	nttbnd2addr: entry nttbnd2addr: port resolved to 1575 nttbnd2addr: looking up IP addr for host: oranamesrvr0 nttbnd2addr: exitnsopen: entry nsmal: entry nsmal: 330 bytes at 0x30d76e74 nsmal: normal exit nsopen: opening transport... nttcon: entry nttcon: toc = 1 nttcnp: entry nttcnp: creating a socket. nttcnp: exit nttcni: entry nttcni: trying to connect to socket 1. ntt2err: entry
ネットワーク・トランスポート（NT）は、Oracle Names サーバーが稼動していないのでエラー「no listener」を戻します。	-<ERROR>- ntt2err: soc 1 error - operation=1, ntresnt[0]=511, ntresnt[1]=61 ntresnt[2]=0 ntt2err: exit nttcni: exit nttcon: exit nserror: entry
エラーは、次のレイヤー（NS）に伝えられます。	-<ERROR>- nserror: nsres: id=0, op=65, ns=12541, ns2=12560; nt[0]=511, nt[1]=61,nt[2]=0 -<ERROR>- nsopen: unable to open transport nsmfr: entry nsmfr: 330 bytes at 0x30d76e74 nsmfr: normal exit nsopen: error exit nscall: error exit nricdt: Call failed... nricfg: entry nricfg: exit nricdt: Call made to destination nricdt: exit  nricall: Failed to copy originating community name value binding nricall: Exiting NRICALL with following termination result -1 nricall: exit nngsfad_free_stream_addr: "(DESCRIPTION=(CONNECT_DATA=(RPC=ON)) (ADDRESS=(PROTOCOL=tcp) (HOST=oranamesrvr0) (PORT=1575)))" -<ERROR>- nngsget_get_stream: open failure, error stack follows



例 12-19 トレース・ファイルの例

説明	トレース・ファイルの情報
エラーは、次のレイヤー (TNS) に伝えられます。	TNS-12224: TNS: リスナーがありません。 TNS-12541: TNS: リスナーがありません。 TNS-12560: TNS: プロトコル・アダプタ・エラー TNS-00511: リスナーがありません。 nnfgrwsp: Query unsuccessful, skipping to next adapter
Oracle Names サーバーが使用できないので、アドレスは Oracle Names サーバー上で見つかりません。	nnfun2a: address for name "june" not found nngsfad_free_stream_addr: "(DESCRIPTION = (CONNECT_DATA=(RPC=ON)) (ADDRESS=(PROTOCOL=tcp) (HOST=oramesrvr0) (PORT=1575)))" nngtdei_deinit_msg: free message pool block nngtfms_free_msg: message ID -10429 nngtfms_free_msg: message was a request nngtfms_free_msg: message free, type 100 nngtfoa_free_objarr: free message object array nngtfmt_free_msg_type: type-specific message free, type 100 nngtfoa_free_objarr: free message object array nngtfms_free_msg: message ID 0 nngtfms_free_msg: message was a response nngtfms_free_msg: message free, type 0 nngsdei_deinit_streams: deinit nngscls_close_stream: UID 11 not established, ignored nngscls_close_stream: UID 0 not established, ignored osnqrm: Return code from nnfsn2a is 409
エラーがユーザーに戻されます。	--<ERROR>- onstns: Couldn't connect, returning 12154 onstns: exit osnqtg: Count in the OSN global area is now 0 rigbd: entry nrigbd: exit osnqtg: Count in the NL global area is now 0

トレース・ファイル例のサマリー

このトレース・ファイルは、エラー「ORA-12154: TNS: サービス名を解決できませんでした。」が発生したときに、Net8 で何が起こったかについてのサマリーを示します。この例では、クライアントはネット・サービス名 june への接続に失敗しました。これは、sqlnet.ora ファイルに NAMES.DEFAULT\_DOMAIN = COM パラメータが設定されているためです。このパラメータは、ネット・サービス名 june など、要求されたすべてのネット・サービス名に拡張子 com を追加します。ところがこのネット・サービス名は、クライアントの tnsnames.ora ファイルにも Oracle Names サーバーにも定義されていません。この問題を解決するには、次のいずれかを行います。

- sqlnet.ora ファイルを編集して、NAMES.DEFAULT\_DOMAIN 構成パラメータを削除します。



- 要求する接続先を `june` ではなく `june.com` とします。

## オラクル社カスタマ・サポート・センターへの連絡

この章を読んでも問題が解決できない場合、またはオラクル社カスタマ・サポート・センターに連絡してエラーを報告するように指示された場合は、次の情報を手元にご用意ください。

- アプリケーションが稼働しているハードウェアとオペレーティング・システムのリリース番号
- 現在の問題に関係するすべての Oracle ネットワーク製品の 5 桁のリリース番号
- 使用しているサードパーティ・ベンダーとバージョン
- 複数のエラー・コードやメッセージが表示された場合は、正確なコード番号およびメッセージ・テキスト、それらの表示順序
- クライアントとサーバー間に存在するリンクの種類
- 機能する部分の説明
- 正確なエラー・メッセージ（ある場合）
- トレースを実行した場合は Net8 トレース、実行していない場合はログ・ファイル



# 第 IV 部

---

## リファレンス

第IV部は、このガイドの参照資料です。次の付録があります。

- [付録 A「制御ユーティリティ」](#)
- [付録 B「プロトコル・アドレス」](#)
- [付録 C「構成パラメータ」](#)
- [付録 D「Net8 の LDAP スキーマ」](#)







---

## 制御ユーティリティ

Net8 には、リスナー、Oracle Names および Oracle Connection Manager を制御するユーティリティがあります。この付録では、前提条件、パスワード、構文または引数の規則、使用上の注意、それらを適用するときに役に立つ例など、各ユーティリティで使用するコマンドを示します。

この付録では、次の 3 つの制御ユーティリティについて説明します。

- リスナー制御ユーティリティ (LSNRCTL)
- Oracle Names 制御ユーティリティ (NAMESCTL)
- Oracle Connection Manager 制御ユーティリティ (CMCTL)



## 動作モード

制御ユーティリティは、ランタイム・セッションの間動作します。ユーティリティは、次の3つのモードで実行できます。

- インタプリタ・モード — 制御ユーティリティがコマンドラインからロードされます。ロードされると、次のプロンプトが表示されます。

```
LSNRCTL>
```

- コマンドライン・モード — 完全なコマンドをプログラムに対するパラメータとして指定して制御ユーティリティを実行すると、ほとんどのコマンドをオペレーティング・システムのコマンドラインから実行することもできます。この場合は、制御ユーティリティがコマンドをロードおよび実行してから、オペレーティング・システム・プロンプトに戻ります。たとえば、次のコマンドを使用します。

```
NAMESCTL start  
NAMESCTL status sales.com
```

- バッチ・コマンド・モード — 複数のコマンドを標準テキスト・ファイル内で組み合わせて、それらを一連のコマンドとして実行できます。コマンドをバッチ・モードで実行する場合は、次のフォーマットを使用します。

```
control_utility @file_name
```

- REM または # を使用して、バッチ・スクリプト内でコメントを示すことができます。他のすべての行はコマンドと見なされます。バッチ実行中は、コマンドの確認は不要です。

## リスナー制御ユーティリティ (LSNRCTL)

リスナー制御ユーティリティ (LSNRCTL) を使用して、リスナーを管理することができます。LSNRCTL の一般構文は、次のとおりです。

```
LSNRCTL command [listener_name]
```

*listener\_name* は、*listener.ora* ファイルで定義されているリスナーの名前です。LISTENER という名前のデフォルト・リスナーを使用している場合は、リスナーを識別する必要はありません。

LSNRCTL は、TNS\_ADMIN 環境変数によって定義されたディレクトリ内で *listener.ora* ファイルを検索します。または UNIX の場合は \$ORACLE\_HOME/network/admin、Windows NT の場合は ORACLE\_HOME¥network¥admin のディレクトリが検索されます。LSNRCTL は、LISTENER という名前のリスナーを検索します。LISTENER という名前のリスナーのエントリが見つからない場合、特に指定されていない限り、LSNRCTL はコマンドの適用対象をデフォルトのリスナーとみなします。つまり、ポート 1521 の TCP/IP、およびキー PNPKEY の IPC をリスニングしている LISTENER という名前のリスナーです。



LSNRCTL には、次に示すタイプのコマンドが含まれています。

- 操作コマンド (START、STOP など)
- 修飾子コマンド (SET *command* など)
- 情報コマンド (STATUS、SHOW *command* など)
- コマンド・ユーティリティ操作コマンド (EXIT、QUIT、HELP など)

これらのコマンドを使用すると、1 つまたは複数のリスナーで基本的な管理機能を実行できます。また、リスナーのパラメータ設定も表示および変更できます。

## SET および SHOW コマンド

SET コマンドを使用して、リスナーのパラメータ値や、LSNRCTL 制御ユーティリティ・セッション中の LSNRCTL 環境を変更できます。その後、[SAVE\\_CONFIG](#) コマンドを使用して、これらの値や設定を保存できます。SHOW コマンドは、現行の構成の設定値を表示するために使用します。

## セキュリティ

PASSWORDS\_*listener\_name* パラメータが *listener.ora* ファイル内で設定されている場合や、新規の暗号化パスワードを作成するために [CHANGE\\_PASSWORD](#) コマンドが使用された場合は、LSNRCTL 制御ユーティリティはリスナーの停止などの重要な操作に対して、[SET\\_PASSWORD](#) コマンドを必要とします。

## LSNRCTL コマンド

次のコマンドが、LSNRCTL で使用できます。

---

### CHANGE\_PASSWORD

目的:	<p><i>listener.ora</i> ファイル内の PASSWORDS_<i>listener_name</i> パラメータで設定された、リスナーの暗号化パスワードを動的に変更できます。</p> <p>このコマンドは、<i>listener.ora</i> ファイル内の PASSWORDS_<i>listener_name</i> パラメータによってすでに確定された非暗号化パスワードは変更しません。このコマンドは、新しいパスワードの確定、または、<i>listener.ora</i> ファイル内で暗号化されているパスワードの変更のみを行います。</p>
前提条件:	なし



---

### CHANGE\_PASSWORD

パスワード:	必要 パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文:	オペレーティング・システムからは、次のように入力します。 <code>LSNRCTL change_password [listener_name]</code> LSNRCTL ユーティリティからは、次のように入力します。 <code>LSNRCTL&gt; change_password [listener_name]</code>
引数:	リスナー名 (デフォルト名 LISTENER を使用しない場合)
使用上の注意:	古いパスワードと新しいパスワードの入力を促すプロンプトが表示されます。次に、新しいパスワードを再入力すると、古いパスワードが変更されます。操作中は、古いパスワードも新しいパスワードも表示されません。
例:	<pre>LSNRCTL&gt; change_password Old password: New password: Reenter new password: Connecting to (ADDRESS=(PROTOCOL=ipc) (KEY=iris)) Password changed for LISTENER The command completed successfully</pre>

---

---

### DBSNMP\_START

目的:	同じノードで稼働している Oracle データベースの SNMP サブエージェントを起動します。
前提条件:	コマンドはローカルに実行する必要があります。
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 <code>LSNRCTL dbsnmp_start</code> LSNRCTL ユーティリティからは、次のように入力します。 <code>LSNRCTL&gt; dbsnmp_start</code>
引数:	なし
使用上の注意:	なし
例:	<code>LSNRCTL&gt; dbsnmp_start</code>

---



---

**DBSNMP\_STATUS**

目的:	Oracle データベースの SNMP サブエージェントが稼働しているかどうかを確認します。
前提条件:	なし
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 <code>LSNRCTL dbsnmp_status</code> LSNRCTL ユーティリティからは、次のように入力します。 <code>LSNRCTL&gt; dbsnmp_status</code>
引数:	なし
使用上の注意:	Oracle データベースが存在するノード上で実行する必要があります。
例:	<code>LSNRCTL&gt; dbsnmp_status</code>

---

---

**DBSNMP\_STOP**

目的:	同じノードで稼働している Oracle データベースの SNMP サブエージェントを停止します。
前提条件:	ローカルに実行する必要があります。
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 <code>LSNRCTL dbsnmp_stop</code> LSNRCTL ユーティリティからは、次のように入力します。 <code>LSNRCTL&gt; dbsnmp_stop</code>
引数:	なし
使用上の注意:	なし
例:	<code>LSNRCTL&gt; dbsnmp_stop</code>

---

---

**EXIT**

目的:	LSNRCTL を終了して、オペレーティング・システム・プロンプトに戻ります。
-----	---



EXIT	
前提条件:	なし
パスワード:	不要
構文:	LSNRCTL ユーティリティからは、次のように入力します。  LSNRCTL> exit
引数:	なし
使用上の注意:	このコマンドは <b>QUIT</b> コマンドと同じです。
例:	LSNRCTL> exit

HELP	
目的:	使用可能な LSNRCTL コマンドの一覧を表示します。HELP コマンドのうちの 1 つに応答して、LSNRCTL はコマンドの使用方法についてのヘルプを表示します。
前提条件:	なし
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。  LSNRCTL help LSNRCTL help [command]  LSNRCTL ユーティリティからは、次のように入力します。  LSNRCTL help LSNRCTL> help [command]
引数:	HELP コマンド。次の出力例にコマンドが示されています。



---

**HELP**

例:                   LSNRCTL> help  
                      The following operations are available  
                      An asterisk (\*) denotes a modifier or extended command:  
                      change\_password  
                      dbsnmp\_start  
                      dbsnmp\_stop  
                      dbsnmp\_status  
                      exit  
                      quit  
                      reload  
                      services  
                      set\*  
                      show\*  
                      spawn  
                      start  
                      status  
                      stop  
                      trace  
                      version

---

---

**QUIT**

目的:                LSNRCTL を終了して、オペレーティング・システム・プロンプトに戻ります。

前提条件:           なし

パスワード:          不要

構文:                LSNRCTL ユーティリティからは、次のように入力します。  
                      LSNRCTL> quit

引数:                なし

使用上の注意:        このコマンドは **EXIT** コマンドと同じです。

例:                   LSNRCTL> quit

---

---

**RELOAD**

目的:                リスナー・アドレス以外のすべてをシャットダウンし、  
                      listener.ora ファイルを再度読み込みます。このコマンドを使用  
                      すると、リスナーを実際に停止せずに、サービスを追加、変更できま  
                      す。



---

### RELOAD

前提条件:	このコマンドは有効なノードには作用しません。この場合は、リスナーを停止して再起動する必要があります。
パスワード:	必要 パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文:	オペレーティング・システムからは、次のように入力します。 <code>LSNRCTL reload [listener_name]</code> LSNRCTL ユーティリティからは、次のように入力します。 <code>LSNRCTL&gt; reload [listener_name]</code>
引数:	リスナー名 (デフォルト名 LISTENER を使用しない場合)
例:	<code>LSNRCTL&gt; reload</code>

---

---

### SAVE\_CONFIG

目的:	リスナーのトレース・レベル、トレース・ファイル、トレース・ディレクトリ、ロギング、および接続タイムアウトなどの現行の構成状態を、listener.ora ファイルと比較します。変更はすべて、形式、コメントおよび大文字と小文字をできるだけ維持したまま、listener.ora に保存されます。listener.ora ファイルの変更前に、listener.bak という名前でバックアップ・ファイルが作成されます。
パスワード:	必要 パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文:	オペレーティング・システムからは、次のように入力します。 <code>LSNRCTL save_config [listener_name]</code> LSNRCTL ユーティリティからは、次のように入力します。 <code>LSNRCTL&gt; save_config [listener_name]</code>
引数:	リスナー名 (デフォルト名 LISTENER を使用しない場合)
使用上の注意:	管理者がオンラインで変更した構成箇所をリスナー構成ファイルに保存する場合は、このコマンドを使用します。このコマンドは特に、変更された暗号化パスワードの保存に役立ちます。
例:	<code>LSNRCTL&gt; save_config listener</code>

---



---

**SERVICES**

目的:	リスナーがクライアントからの接続要求を転送する宛先のサービス・ハンドラ（ディスパッチャ、専用サーバーおよび事前生成済専用サーバー）についての詳細情報を表示します。この情報には、確立された接続数や拒否された接続数などが含まれます。
前提条件:	なし
パスワード:	必要 パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文:	オペレーティング・システムからは、次のように入力します。 <code>LSNRCTL services [listener_name]</code> LSNRCTL ユーティリティからは、次のように入力します。 <code>LSNRCTL&gt; services [listener_name]</code>
引数:	リスナー名（デフォルト名 LISTENER を使用しない場合）
使用上の注意:	<b>SET DISPLAYMODE</b> コマンドは、出力の書式や詳細レベルを変更します。

---



---

## SERVICES

例：

下の例は、NORMAL モードでの SERVICES の出力です。この出力が示す内容は、次のとおりです。

- ORCL という名前のインスタンスは、s1 と s2 の2つのサービスに属し、合計で3つのサービス・ハンドラを持っています。
- 下の出力の Relevant handlers: 1 によって指定されているように、サービス s1 は、1つのディスパッチャによってのみ処理されます（また、このディスパッチャは、サービス s1 を要求するクライアントのみを処理します）。
- 下の出力の Relevant handlers: 2 によって指定されているように、サービス s2 は、1つのディスパッチャおよび1つの専用サーバーによって処理されます。

```
LSNRCTL> set displaymode normal
LSNRCTL> services
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=net) (QUEUE_SIZE=32)))
Services Summary...
Service "s1"                has 1 instances.
  Instance "ORCL"
    Status: READY   Total handlers: 3   Relevant handlers: 1
    D000 established:0 refused:0 current:0 max:1022
state:ready

  (ADDRESS=(PROTOCOL=tcp) (HOST=dlsun1013.us.oracle.com) (PORT=1582)
  )
    Session: NS
Service "s2"                has 1 instances.
  Instance "ORCL"
    Status: READY   Total handlers: 3   Relevant handlers: 2
    D0001 established:0 refused:0 current:0 max:1022
state:ready

  (ADDRESS=(PROTOCOL=tcp) (HOST=dlsun1013.us.oracle.com) (PORT=53101
  ))
    Session: NS
    DEDICATED established:0 refused:0 current:0 max:1
state:ready
    Session: NS
The command completed successfully
```

---



---

**SET**

目的:	現行の LSNRCTL セッション用に設定できる使用可能な構成コマンドのリストを表示します。
前提条件:	なし
パスワード:	不要
構文:	<p>オペレーティング・システムからは、次のように入力します。</p> <pre>LSNRCTL set LSNRCTL set [command]</pre> <p>LSNRCTL ユーティリティからは、次のように入力します。</p> <pre>LSNRCTL&gt; set LSNRCTL set [command]</pre>
引数:	SET コマンド。次の出力例にコマンドが示されています。
例:	<pre>LSNRCTL&gt; set The following operations are available with set. An asterick (*) denotes a modifier or extended command. connect_timeout current_listener displaymode log_file log_directory log_status password raw_mode save_config_on_stop startup_waittime trc_file trc_directory trc_level use_plugandplay</pre>

---

**SET CONNECT\_TIMEOUT**

目的:	接続が開始されてからリスナーが有効な接続要求を待つ時間を秒で指定します。
前提条件:	なし
パスワード:	不要



---

### SET CONNECT\_TIMEOUT

構文: オペレーティング・システムからは、次のように入力します。

```
LSNRCTL set connect_timeout {time}
```

LSNRCTL ユーティリティからは、次のように入力します。

```
LSNRCTL> set connect_timeout {time}
```

引数: 秒数

使用上の注意: なし

例:

```
LSNRCTL> set connect_timeout 20
Connecting to (ADDRESS=(PROTOCOL=ipc) (KEY=iris))
LISTENER parameter "connect_timeout" set to 20
The command completed successfully
```

---

---

### SET CURRENT\_LISTENER

目的: パラメータを設定または表示するリスナーの名前を設定できます。

パスワード: 不要

構文: LSNRCTL ユーティリティからは、次のように入力します。

```
LSNRCTL> set current_listener [listener_name]
```

引数: リスナー名 (デフォルト名 LISTENER を使用しない場合)

使用上の注意: [SET CURRENT\\_LISTENER](#) が設定されると、LSNRCTL コマンドは設定したリスナーに対して実行されます。したがって、リスナーの名前を指定する必要はなくなります。

例:

```
LSNRCTL> set current_listener [listener_name]
```

---

---

### SET DISPLAYMODE

目的: LSNRCTL [SERVICES](#) および LSNRCTL [STATUS](#) コマンドについて表示する書式、詳細レベルを変更します。

パスワード: 不要

構文: LSNRCTL ユーティリティからは、次のように入力します。

```
LSNRCTL> set displaymode [argument]
```



---

**SET DISPLAYMODE**

- 引数: COMPAT (デフォルト) - 旧バージョンのリスナーと互換性のある出力を表示します。
- NORMAL - 書式化された記述的な出力形式で表示します。オラクル社では、この方法をお薦めします。
- VERBOSE - 書式化された記述的な出力形式で、リスナーから受信したすべてのデータを表示します。
- RAW - リスナーから受信したすべてのデータを書式化せずに表示します。この出力は、オラクル社カスタマ・サポート・センターから指示があった場合にのみ使用してください。
- 例: LSNRCTL> set displaymode normal
- 

---

**SET LOG\_DIRECTORY**

- 目的: リスナー・ログ・ファイルが書き込まれるディレクトリを設定します。デフォルトでは、ログ・ファイルは、UNIX の場合 \$ORACLE\_HOME/network/log、Windows NT の場合 ORACLE\_HOME/network/log に書き込まれます。
- 前提条件: なし
- パスワード: 必要
- パスワードが設定されている場合、このコマンドの前に **SET PASSWORD** コマンドを実行する必要があります。
- 構文: オペレーティング・システムからは、次のように入力します。
- ```
LSNRCTL set log_directory {directory}
```
- LSNRCTL ユーティリティからは、次のように入力します。
- ```
LSNRCTL> set log_directory {directory}
```
- 引数: ディレクトリ・パス
- 例: LSNRCTL> set log\_directory /usr/oracle/admin  
Connecting to (ADDRESS=(PROTOCOL=ipc) (KEY=iris)) LISTENER  
parameter "log\_directory" set to /usr/oracle/admin  
The command completed successfully
-



---

### SET LOG\_FILE

目的:	リスナー・ログ・ファイルの名前を設定します。デフォルトのログ・ファイル名は <code>listener.log</code> です。
前提条件:	なし
パスワード:	必要 パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文:	オペレーティング・システムからは、次のように入力します。 <code>LSNRCTL set log_file {file_name}</code> LSNRCTL ユーティリティからは、次のように入力します。 <code>LSNRCTL&gt; set log_file {file_name}</code>
引数:	ファイル名
例:	<code>LSNRCTL&gt; set log_file list.log</code> Connecting to (ADDRESS=(PROTOCOL=ipc)(KEY=iris))LISTENER parameter "log_file" set to list.log The command completed successfully

---

---

### SET LOG\_STATUS

目的:	リスナーのロギングをオンまたはオフにします。
前提条件:	なし
パスワード:	必要 パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文:	オペレーティング・システムからは、次のように入力します。 <code>LSNRCTL set log_status [argument]</code> LSNRCTL ユーティリティからは、次のように入力します。 <code>LSNRCTL&gt; set log_status [argument]</code>
引数:	ON   OFF (デフォルト)
例:	<code>LSNRCTL&gt; set log_status on</code>

---



---

**SET PASSWORD**

目的:	<p><a href="#">SAVE_CONFIG</a> や <a href="#">STOP</a> などの特権的な LSNRCTL ユーティリティ・コマンドを実行するためのパスワードを設定します。</p> <p>入力するパスワードは、<code>listener.ora</code> ファイルの <a href="#">PASSWORDS_listener_name</a> パラメータまたは LSNRCTL ユーティリティの <a href="#">CHANGE_PASSWORD</a> コマンドの 1 つのセットと一致している必要があります。</p>
構文:	<p>オペレーティング・システムからは、次のように入力します。</p> <pre>LSNRCTL set password</pre> <p>LSNRCTL ユーティリティからは、次のように入力します。</p> <pre>LSNRCTL&gt; set password</pre> <p>非暗号化パスワードの構文は次のとおりです。</p> <pre>set password password</pre>
引数:	パスワード
使用上の注意:	<p>シェルの起動時またはセッション中のいつでも、このコマンドを入力できます（リスナーを停止する前に、SET PASSWORD コマンドを入力する必要があります）。</p> <p>パスワードをより安全に入力するには、対話モードを使用します。</p>
例:	<pre>LSNRCTL&gt; set password enter listener password: password</pre>

---

---

**SET SAVE\_CONFIG\_ON\_STOP**

目的:	<p>LSNRCTL セッションの終了後、LSNRCTL <a href="#">SET</a> コマンドによって行われた変更を <code>listener.ora</code> ファイルに保存するかどうかを指定します。</p> <p>LSNRCTL セッションが終了する直前にすべての値が保存されます。形式、コメント、大文字と小文字は、可能な限り維持されます。すべてのパラメータをすぐに保存する場合は、<a href="#">SAVE_CONFIG</a> コマンドを使用します。</p> <p>変更はすべて、形式、コメントおよび大文字と小文字をできるだけ維持したまま、<code>listener.ora</code> ファイルに保存されます。<code>listener.ora</code> ファイルの変更前に、<code>listener.bak</code> という名前でバックアップ・ファイルが作成されます。</p>
パスワード:	不要

---



---

### SET SAVE\_CONFIG\_ON\_STOP

構文: オペレーティング・システムからは、次のように入力します。

```
LSNRCTL set save_config_on_stop [argument]
```

LSNRCTL ユーティリティからは、次のように入力します。

```
LSNRCTL> set save_config_on_stop [argument]
```

引数: ON | OFF (デフォルト)

例: LSNRCTL> set save\_config\_on\_stop on

---

---

### SET STARTUP\_WAITTIME

目的: リスナーが START コマンドに応答するまで休止する時間を指定します。

前提条件: なし

パスワード: 必要

パスワードが設定されている場合、このコマンドの前に **SET PASSWORD** コマンドを実行する必要があります。

構文: オペレーティング・システムからは、次のように入力します。

```
LSNRCTL set startup_waittime [time]
```

LSNRCTL ユーティリティからは、次のように入力します。

```
LSNRCTL> set startup_waittime [time]
```

引数: 秒数

例: LSNRCTL> set startup\_waittime 10  
Connecting to (ADDRESS=(PROTOCOL=ipc) (KEY=iris))LISTENER  
parameter "startup\_waittime" set to 10  
The command completed successfully

---

---

### SET TRC\_DIRECTORY

目的: リスナー・トレース・ファイルが書き込まれるディレクトリを設定します。デフォルトでは、トレース・ファイルは、UNIX の場合 \$ORACLE\_HOME/network/trace、Windows NT の場合 ORACLE\_HOME/network/trace に書き込まれます。

前提条件: なし

パスワード: 不要



---

**SET TRC\_DIRECTORY**

構文: オペレーティング・システムからは、次のように入力します。  
LSNRCTL set trc\_directory {directory}  
LSNRCTL ユーティリティからは、次のように入力します。  
LSNRCTL> set trc\_directory {directory}

引数: ディレクトリ・パス

例: LSNRCTL> set trc\_directory /usr/oracle/admin  
Connecting to (ADDRESS=(PROTOCOL=ipc) (KEY=iris))  
LISTENER parameter "trc\_directory" set to /usr/oracle/admin  
The command completed successfully

---

---

**SET TRC\_FILE**

目的: リスナー・トレース・ファイルの名前を設定します。デフォルトのトレース・ファイル名は listener.trc です。

前提条件: なし

パスワード: 不要

構文: オペレーティング・システムからは、次のように入力します。  
LSNRCTL set trc\_file {file\_name}  
LSNRCTL ユーティリティからは、次のように入力します。  
LSNRCTL> set trc\_file {file\_name}

引数: ファイル名

例: LSNRCTL> set trc\_file list.trc  
Connecting to (ADDRESS=(PROTOCOL=ipc) (KEY=iris))  
LISTENER parameter "trc\_file" set to list.trc  
The command completed successfully

---

---

**SET TRC\_LEVEL**

目的: リスナーのトレース・レベルを設定します。

前提条件: なし

パスワード: 必要

パスワードが設定されている場合、このコマンドの前に **SET PASSWORD** コマンドを実行する必要があります。



---

### SET TRC\_LEVEL

構文:	オペレーティング・システムからは、次のように入力します。 <code>LSNRCTL set trc_level {level}</code> LSNRCTL ユーティリティからは、次のように入力します。 <code>LSNRCTL&gt; set trc_level {level}</code>
引数:	トレース・レベル: <ul style="list-style-type: none"><li>■ OFF - トレース出力しない</li><li>■ USER - ユーザー・トレース情報</li><li>■ ADMIN - 管理トレース情報</li><li>■ SUPPORT - オラクル社カスタマ・サポート・センター用のトレース情報</li></ul>
使用上の注意:	このコマンドは TRACE と同様に機能します。
例:	<code>LSNRCTL&gt; set trc_level admin</code>

---

---

### SET USE\_PLUGANDPLAY

目的:	リスナーに、データベース情報を Oracle Names サーバーに登録するよう指示します。
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 <code>LSNRCTL set use_plugandplay [argument]</code> LSNRCTL ユーティリティからは、次のように入力します。 <code>LSNRCTL&gt; set use_plugandplay [argument]</code>
引数:	ON   OFF (デフォルト)
例:	<code>LSNRCTL&gt; set use_plugandplay on</code>

---



---

**SHOW**

目的:	<p>現行の LSNRCTL セッションについて表示できる、使用可能なコマンドのリストを表示します。LSNRCTL は SHOW コマンドのうちの 1 つに対して、そのパラメータに対するリスナーの現行の設定を表示します。</p> <p>SET PASSWORD を除くすべての SET コマンドには、同等の SHOW コマンドがあります。</p>
前提条件:	なし
パスワード:	<p>SHOW SAVE_CONFIG_ON_STOP および SHOW USE_PLUGANDPLAY については必要。</p> <p>パスワードが設定されている場合、これらのコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。</p>
構文:	<p>オペレーティング・システムからは、次のように入力します。</p> <pre>LSNRCTL show LSNRCTL show [listener_name] [command]</pre> <p>LSNRCTL ユーティリティからは、次のように入力します。</p> <pre>LSNRCTL&gt; show LSNRCTL&gt; show [listener_name] [command]</pre>
引数:	<ul style="list-style-type: none"><li>■ リスナー名 (デフォルト名 LISTENER を使用しない場合)</li><li>■ SHOW コマンド。次の出力例にコマンドが示されています。</li></ul>
例:	<pre>LSNRCTL&gt; show The following properties are available with SHOW: An asterisk (*) denotes a modifier or extended command: connect_timeout current_listener displaymode log_file log_directory log_status password raw_mode save_config_on_stop startup_waittime trc_file trc_directory trc_level use_plugandplay</pre>

---



---

### SPAWN

目的:	リスナーが稼働しているマシンに格納されていて、 <code>listener.ora</code> ファイルに別名でリストされているプログラムを起動します。
前提条件:	なし
パスワード:	必要 パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文:	オペレーティング・システムからは、次のように入力します。 <code>LSNRCTL spawn [listener_name] alias (arguments=arg1,arg2,)</code> LSNRCTL ユーティリティからは、次のように入力します。 <code>LSNRCTL&gt; spawn [listener_name] alias (arguments=arg1,arg2,)</code>
引数:	<ul style="list-style-type: none"><li>■ リスナー名 (デフォルト名 LISTENER を使用しない場合)</li><li>■ <code>listener.ora</code> ファイルにリストされているプログラムの別名。</li><li>■ 作成するプログラムに送られる引数</li></ul>
例:	<code>LSNRCTL&gt; spawn nstest_alias (ARGUMENTS='')</code>

---

---

### START

目的:	指定したリスナーを起動します。
前提条件:	リスナーが稼働していないようにしてください。
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 <code>LSNRCTL start [listener_name]</code> LSNRCTL ユーティリティからは、次のように入力します。 <code>LSNRCTL&gt; start [listener_name]</code>
引数:	リスナー名 (デフォルト名 LISTENER を使用しない場合)

---



**START**

使用上の注意:

listener.ora ファイル内で LISTENER 以外の名前で構成されているリスナーを起動するには、そのリスナーの名前を指定します。

たとえば、リスナー名が TCP\_LSNR の場合は、次のように入力します。

```
LSNRCTL start tcp_lsnr
```

または、LSNRCTL ユーティリティ・プロンプトから、次のように入力します。

```
LSNRCTL> start tcp_lsnr
```

例:

```
LSNRCTL> start
```

```
Starting /private/dsteiner/sales/bin/tnslsnr: please wait...
```

```
TNSLSNR for Solaris: Version 8.1.6.0.0
```

```
System parameter file is
```

```
/private/dsteiner/sales/network/admin/listener.ora
```

```
Log messages written to
```

```
/private/dsteiner/sales/network/log/listener.log
```

```
Trace information written to
```

```
/private/dsteiner/sales/network/trace/listener.trc
```

```
Listening on:
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=dlsun1770) (PORT=1521)))
```

```
(PROTOCOL_STACK=(PRESENTATION=TTCP) (SESSION=NS)))
```

```
Connecting to
```

```
(ADDRESS=(PROTOCOL=TCP) (HOST=dlsun1770) (PORT=1521))
```

```
STATUS of the LISTENER
```

```
-----
```

```
Alias LISTENER
```

```
Version TNSLSNR for Solaris: Version 8.1.6.0.0
```

```
Start Date 08-JUL-99 15:50:10
```

```
Uptime 0 days 0 hr. 0 min. 0 sec
```

```
Trace Level user
```

```
Security OFF
```

```
SNMP OFF
```

```
Listener Parameter File
```

```
/private/dsteiner/sales/network/admin/listener.ora
```

```
Listener Log File
```

```
/private/dsteiner/sales/network/log/listener.log
```

```
Listener Trace File
```

```
/private/dsteiner/sales/network/trace/listener.trc
```

```
Services Summary...
```

```
PLSExtProc has 1 service handler(s)
```

```
The command completed successfully
```



---

## STATUS

目的:	リスナーについての次の基本的なステータス情報を表示します。 <ul style="list-style-type: none"><li>■ リスナーのバージョン</li><li>■ 開始時刻と終了時刻</li><li>■ トレース・レベル</li><li>■ ログギング・オプションとトレース・オプションの現行の設定</li><li>■ 使用している listener.ora</li><li>■ listener.ora ファイルでパスワードが暗号化されているか</li><li>■ ネットワーク・リスナーが、SNMP ベースのネットワーク管理システムからの問合せに応答できるかどうか</li><li>■ このリスナーに登録されているデータベース・サービスのリスト</li><li>■ このリスナーがリスニングしているアドレス</li></ul>
前提条件:	なし
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 <code>LSNRCTL status [listener_name]</code> LSNRCTL ユーティリティからは、次のように入力します。 <code>LSNRCTL&gt; status [listener_name]</code>
引数:	リスナー名 (デフォルト名 LISTENER を使用しない場合)
使用上の注意:	<a href="#">SET DISPLAYMODE</a> コマンドは、出力の書式や詳細レベルを変更します。

---



---

**STATUS**

例：

下の例は、NORMAL モードでの STATUS の出力です。出力には、次の内容が含まれます。

- リスニングのエンドポイントのサマリー
- LSNRCTL SERVICES の出力を簡略化したサービス・サマリー

```
LSNRCTL> set displaymode normal
LSNRCTL> status
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=net) (QUEUESIZE=32)))
STATUS of the LISTENER
-----
Alias                               LISTENER
Version                             TNSLSNR for Solaris: Version
8.1.6.0.0 -
Development
Start Date                           08-SEP-1999 10:52:32
Uptime                               0 days 0 hr. 0 min. 54 sec
Trace Level                           support
Security                             OFF
SNMP                                 OFF
Listener Parameter File              /listener.ora
Listener Log File                    /vobs/oracle/network/log/listener.log
Listener Trace File                  /vobs/oracle/network/trace/listener.trc
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=net)))

  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=dlsun1013.us.oracle.c
om) (PORT=1521)))

  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=dlsun1013.us.oracle.c
om) (PORT=5522
)) (PRESENTATION=GIOP))

  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=dlsun1013.us.oracle.c
om) (PORT=5523
)) (PRESENTATION=oracle.aurora.net.Echo))

  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=dlsun1013.us.oracle.c
om) (PORT=8080
)) (PRESENTATION=oracle.aurora.net.BabyHttp2))
Services Summary...
Service "s1"                has 1 instances.
  Instance "ORCL"
    Status: READY  Total handlers: 3  Relevant handlers: 1
Service "s2"                has 1 instances.
  Instance "ORCL"
    Status: READY  Total handlers: 3  Relevant handlers: 2
The command completed successfully
```

---



---

### STOP

目的:	指定されたリスナーを停止します。
前提条件:	リスナーが稼働している必要があります。
パスワード:	必要 パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文:	オペレーティング・システムからは、次のように入力します。 <code>LSNRCTL stop [listener_name]</code> LSNRCTL ユーティリティからは、次のように入力します。 <code>LSNRCTL&gt; stop [listener_name]</code>
引数:	リスナー名 (デフォルト名 LISTENER を使用しない場合)
使用上の注意:	リスナーを停止するときは注意してください。プラットフォームとプロトコルによっては、リスナーを停止すると、現在稼働している Net8 接続がすべてシャットダウンされます。接続が継続される場合もありますが、実行中のプロセスをクローズするまでは、リスナーを再起動できません。リスナーの停止前に、すべてのネットワーク・ユーザーに警告メッセージを送信することをお勧めします。
例:	<code>LSNRCTL&gt; stop</code>

---

### TRACE

目的:	リスナーのトレース機能をオンにします。
パスワード:	必要 パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文:	オペレーティング・システムからは、次のように入力します。 <code>LSNRCTL trace level [listener_name]</code> LSNRCTL ユーティリティからは、次のように入力します。 <code>LSNRCTL&gt; trace level [listener_name]</code>

---



---

**TRACE**

引数:	トレース・レベル: <ul style="list-style-type: none"><li>■ OFF - トレース出力しない</li><li>■ USER - ユーザー・トレース情報</li><li>■ ADMIN - 管理トレース情報</li><li>■ SUPPORT - オラクル社カスタマ・サポート・センター用のトレース情報</li></ul> リスナー名（デフォルト名 LISTENER を使用しない場合）
使用上の注意:	このコマンドは SET TRC_LEVEL と同様に機能します。
例:	LSNRCTL> trace admin listener

---

---

**VERSION**

目的:	現行の LSNRCTL のバージョンを表示します。
前提条件:	なし
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 LSNRCTL version [ <i>listener_name</i> ] LSNRCTL ユーティリティからは、次のように入力します。 LSNRCTL> version [ <i>listener_name</i> ]
引数:	リスナー名（デフォルト名 LISTENER を使用しない場合）
例:	LSNRCTL> version listener1

---

## Oracle Names 制御ユーティリティ (NAMESCTL)

Oracle Names 制御ユーティリティ (NAMESCTL) は、Oracle Names サーバーを操作、制御するためにオペレーティング・システム・プロンプトから実行するツールです。NAMESCTL の一般構文は、次のとおりです。

NAMESCTL *command*

このユーティリティには、次に示すタイプのコマンドが含まれています。

- 操作コマンド (START、STOP、RESTART など)
- 修飾子コマンド (SET *property* など)



- 情報コマンド (STATUS、SHOW *property*、PING など)
- コマンド・ユーティリティ操作コマンド (EXIT、QUIT、HELP など)

このコマンドを使用して 1 つ以上の Oracle Names サーバーで、基本的な管理機能を実行できます。また、パラメータ設定の表示および変更もできます。

## SET および SHOW コマンド

SET コマンドを使用して、Oracle Names サーバーのパラメータ値や、NAMESCTL 制御ユーティリティ・セッション中の NAMESCTL 環境を変更できます。その後、[SET SAVE\\_CONFIG\\_ON\\_STOP](#) コマンドまたは [SET SAVE\\_CONFIG\\_INTERVAL](#) コマンドを使用して、これらの値や設定を保存できます。SHOW コマンドは、現行の構成の設定値を表示するために使用します。

## セキュリティ

names.ora ファイルに [NAMES.PASSWORD](#) パラメータが設定されている場合、NAMESCTL 制御ユーティリティは Oracle Names サーバーの停止などの重要な操作に対して、[SET PASSWORD](#) コマンドを必要とします。

[NAMESCTL.SERVER.PASSWORD](#) パラメータが NAMESCTL を実行しているノードの sqlnet.ora ファイルに設定されている場合は、保護された操作を実行するたびに SET PASSWORD コマンドを要求されることはありません。

Oracle Names サーバーのパスワードをクライアントの sqlnet.ora ファイル内で明示的に指定することでセキュリティの問題が懸念される場合は、[NAMESCTL.SERVER.PASSWORD](#) パラメータを指定せずに必ず次のコマンドを使用してください。

```
NAMESCTL> set password
```

パスワードがネットワーク上に送信される場合、sqlnet.ora ファイルで [NAMESCTL.INTERNAL\\_ENCRYPT\\_PASSWORD](#) が FALSE に設定されていない限り、パスワードは names.ora ファイルでの設定にかかわらず暗号化されます。このパラメータが FALSE に設定されている場合は、パスワードは暗号化されません。

## NAMESCTL の分散処理

Oracle Names 制御ユーティリティは、ネットワーク内の他の Oracle Names サーバーに対しても動作します。1 人の管理者がリージョン内のすべての Oracle Names サーバーを管理する場合や、特定の Oracle Names サーバーの可用性をチェックする場合は、NAMESCTL ユーティリティのこの機能が非常に役立ちます。

大半のコマンドは、コマンドを実行する対象の Oracle Names サーバーを指定するために、最後の引数に Oracle Names サーバーの名前が入ります。サーバーの名前を指定しないと、SET で設定してある Oracle Names サーバーにコマンドが適用されます。たとえば、次のように指定します。



```
show system_queries dolphin.acme
```

このコマンドにより、Oracle Names サーバーの `dolphin.acme` に対するシステム問合せ、およびそれらの問合せが次にいつ行われるかが表示されます。個々の Oracle Names サーバーに対して一連のコマンドを実行する前に、次のように入力してください。

```
NAMESCTL> set server onames_server
```

## NAMESCTL の確認モード

一部の NAMESCTL コマンドは、実行する前に確認が必要です。コマンドを発行すると、次のプロンプトが表示されます。

```
confirm: [yes or no]
```

コマンドを実行するときは「yes」と入力し、コマンドを取り消すときは「no」と入力します。

`sqlnet.ora` ファイルのパラメータ `NAMESCTL.NOCONFIRM` を TRUE に設定すると、確認モードをオフにできます。

## NAMESCTL コマンド

次のコマンドは、Oracle Names 制御ユーティリティ (NAMESCTL) で使用できます。

---

### DELEGATE\_DOMAIN

目的: 現行の管理リージョンのサブリージョンの先頭としてドメインを定義します。

前提条件: なし

パスワード: 不要

構文: オペレーティング・システム・プロンプトからは、次のように入力します。

```
NAMESCTL delegate_domain {domain}{names_server}  
{(address=...)}
```

NAMESCTL ユーティリティからは、次のように入力します。

```
delegate_domain {domain}{names_server}{(address=...)}
```

---



---

**DELEGATE\_DOMAIN**

引数:	<p><code>{domain}</code> - ドメイン名を指定します。</p> <p><code>{onames_server}</code> - Oracle Names サーバーの名前を指定します。</p> <p><code>{{address=...}}</code> - Oracle Names サーバーのプロトコル・アドレスを指定します。</p> <p><b>関連項目:</b> プロトコル・アドレスの定義に関する詳細は、<a href="#">付録 B</a> を参照してください。</p>
使用上の注意:	<p>このコマンドを使用して、名前空間を動的に細分化することができます。</p> <p>ドメインがリージョンから委任されていない場合は、そのリージョン内の Oracle Names サーバーがすべてのサブドメインに対する権限を持ちます。ドメインを委任するには、まず新しいリージョンを作成する必要があります。</p> <p>いったんドメインが委任されると、カレント・リージョンにある Oracle Names サーバーはそれ以降のすべての処理を、別の Oracle Names サーバーによって管理されるドメインのあるサブリージョンに渡します。</p>
例:	<pre>NAMESCTL&gt; delegate_domain webwidgets.acme.com ns1.webwidgets.acme.com (address=(protocol=tcp) (host=fred.webwidgets.acme.com) (port=1575))</pre>

---

**DOMAIN\_HINT**

目的:	<p>リモート・リージョンからデータを要求するためのドメイン・ヒントを指定できます。ドメイン・ヒントには、リモート・ドメインの名前、およびそのドメイン内の Oracle Names サーバーのアドレスが少なくとも 1 つ含まれます。ドメイン・ヒントにより、Oracle Names サーバーは要求を特定のアドレスに送信でき、ネットワーク通信量を軽減することができます。</p> <p>ドメイン・ヒントがない場合、Oracle Names サーバーは既知のすべてのリモート Oracle Names サーバーに要求を転送し、次に、リモート Oracle Names サーバーがそのリージョン内のルート Oracle Names サーバーに要求を転送します。その後、ルート Oracle Names サーバーは、要求が参照しているドメインの情報を持つ Oracle Names サーバーに要求を転送します。</p>
前提条件:	なし
パスワード:	不要



---

**DOMAIN\_HINT**

構文:	オペレーティング・システムからは、次のように入力します。 <pre>NAMESCTL domain_hint {domain} {names_server} {(address=...)}</pre> NAMESCTL ユーティリティからは、次のように入力します。 <pre>domain_hint {domain} {names_server} {(address=...)}</pre>
引数:	<pre>{domain}</pre> - ドメイン名を指定します。  <pre>{onames_server}</pre> - Oracle Names サーバーの名前を指定します。  <pre>{(address=...)}</pre> - Oracle Names サーバーのプロトコル・アドレスを指定します。  <b>関連項目:</b> プロトコル・アドレスの定義に関する詳細は、 <a href="#">付録 B</a> を参照してください。
使用上の注意:	ルート・リージョンでないリージョンに対しては、このコマンドを使用して少なくともルート・リージョンを定義しないと、他のリージョン内のオブジェクトを探すことができません。ローカル Oracle Names サーバーが他の特定のリージョンに直接アクセスできるように、追加のヒントを提供することができます。
例:	<pre>NAMESCTL&gt; domain_hint acme.com ns0.acme.com (address=(protocol=tcp) (host=top.acme.com) (port=1575))</pre>

---

---

**DUMP\_LDAP**

目的:	ドメインまたはリージョン内のすべての正当データを問い合わせたり、そのデータを LDAP 準拠のディレクトリ・サービスまたは後でディレクトリにロードできるように <a href="#">LDAP Data Interchange Format (LDIF)</a> ファイルにエクスポートします。
前提条件:	ディレクトリには、すでに Oracle コンテキストおよび Oracle スキーマがある必要があります。
パスワード:	ディレクトリに対してパスワードが必要な場合があります。

---



---

## DUMP\_LDAP

構文:

**LDIF ファイルにデータをエクスポートする場合**

```
NAMESCTL> dump_ldap [source] [destination]  
[options] {-f filename}
```

**ディレクトリにデータをエクスポートする場合**

```
NAMESCTL> dump_ldap [source] [destination]  
[options] {-h host} {-p port} {-D user_dn} {-w  
password}
```

引数:

[source]= {domain} [-R]

{domain} - 移行するドメインのオブジェクトを指定します。デフォルトのドメインはルートです。クライアントが接続しているサーバーがそのドメインの正当なものでない場合、そのドメインに対して権威のあるサーバーに処理が転送されます。

[-R] - 指定されたドメインの認可されたサブドメインもすべて移行することを指定します。

---



---

**DUMP\_LDAP**

[*destination*] - DIT 内の、オブジェクトが格納されるディレクトリの DN を指定します。

DN は次のいずれかの方法で指定できます。

- すべてのツリー・レベルを含む完全な DN。たとえば、`(dn:cn=OracleContext,dc=com,dc=acme)` は、`acme.com` の下の `cn=OracleContext` にオブジェクトをエクスポートすることを指定します。
- RDN にワイルドカードを使用することにより、ツリー階層の目的の深さに対応するテンプレート。たとえば、`(dn:o,ou,dc)` と指定すると、**Organization** (o)、**OrganizationUnit** (ou)、ドメイン・コンポーネント (dc) を含む DIT 構造にマップします。

**注意:** ディレクトリの DIT に `cn=OracleContext` RDN が含まれている場合は、RDN テンプレートを使用するときに `-x` オプションを使用してください。これにより、データベース・サービスとネット・サービス名が `cn=OracleContext` の下にマップされます。

ソース指定で `-R` (再帰) 変数を指定すると、宛先の DN をディレクトリの DIT にマップする方法が決定されます。

- ソースに `-R` を指定しない場合は、指定されたソース・ドメインが、宛先の DN に指定された DIT に直接マップされます。
- `-R` 変数を指定した場合は、指定されたドメインで始まるすべてのサブドメインのデータがエクスポートされます。すべてのサブドメインは、指定された宛先の RDN にマップされます。宛先の DN の RDN にワイルドカードが使用された場合 (値なしで指定されている場合) は、各サブドメインはそのサブドメインと同じ名前の RDN タイプにマップされます。

**注意:** DIT で各サブドメインに `cn=OracleContext` RDN が含まれている場合は、`-x` オプションを使用してください。これにより、データベース・サービスとネット・サービスの名前は、各サブドメインで `cn=OracleContext` の下にマップされます。

**注意:** DN が指定されていない場合は、ドメインは DIT 内のドメイン・コンポーネント (dc) にマップされます。

---



---

**DUMP\_LDAP**

[*options*] - データのエクスポート方法を指定する引数を指定します。

-a - すべてのオブジェクトを DIT に追加することを指定します。オブジェクトがすでに DIT に存在する場合、エラーが発生します。このオプションが指定されていない場合、既存のオブジェクトは修正されますが、追加されません。

-c - エラー時にもエクスポートを続行することを指定します。

-n - 実際にはエクスポートを実行しません。テスト・ランを実行するために、この変数を使用します。

-q - 結果非表示モードで操作を実行します。

-x - 宛先の DN に cn=OracleContext を追加します。

au

{*f filename*} - 移行したいオブジェクトを LDIF ファイルにダンプすることを指定します。後で、LDIF ファイルからディレクトリにロードできます。デフォルトのファイル名は、onames.ldif です。

次の変数は、ディレクトリ・サーバーの位置を指定します。

[-h *host*] - ディレクトリ・サーバーのホスト名を指定します。

[-p *port*] - ディレクトリがリスニングのために構成されているポート番号を指定します。デフォルトの TCP/IP ポート番号は 389 です。

次の変数は、認証資格証明を指定します。

[-D *user\_dn*] - DN の形式で追加および変更権限が与えられているディレクトリ管理者を指定します。たとえば、  
c=us,o=acme,ou=admin,cn=scott は、Scott という名前の管理者用の DN です。

[-w *password*] - ディレクトリ管理者のパスワードを指定します。

使用上の注意：

**関連項目：** 6-31 ページの「[Oracle Names サーバーからのネットワーク・オブジェクトのエクスポート](#)」を参照してください。

例：

**LDIF ファイルにデータをエクスポートする場合**

```
NAMESCTL> dump_ldap sj.us.scszi.com - R
(dn:cn=OracleContext,ou=sj,dc=us,dc=sczi,dc=com) -f
test.ldif
```

**直接ディレクトリにデータをエクスポートする場合**

```
NAMESCTL> dump_ldap sj.us.scszi.com - R
(dn:cn=OracleContext,ou=sj,dc=us,dc=sczi,dc=com) -a -h
dlsun1598 -p 389 -D cn=orcladmin -w welcome
```

---



---

**DUMP\_TNSNAMES**

目的:	ローカル・リージョンで定義されているすべてのアドレスの名前とアドレスを、tnsnames.ora ファイルに書き出します。
前提条件:	なし
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 <code>NAMESCTL dump_tnsnames</code> NAMESCTL ユーティリティからは、次のように入力します。 <code>dump_tnsnames</code>
引数:	なし
使用上の注意:	ローカル・リージョンで定義されているすべてのアドレスの名前とアドレスを、tnsnames.ora に書き出します（アドレス型レコード (A.SMD) すべて）。  このコマンドは、tnsnames.ora ファイルを作成するか、一部を書き換えます。既存の tnsnames.ora にある、Oracle Names サーバーで定義されていないエントリは、そのまま残ります。Oracle Names サーバーにも定義されている tnsnames.ora ファイルのエントリは、すべて上書きされます。Oracle Names サーバーで定義されていて、tnsnames.ora で定義されていないエントリは追加されます。
例:	<code>NAMESCTL&gt; dump_tnsnames</code>

---

---

**EXIT**

目的:	NAMESCTL ユーティリティを終了します。
前提条件:	NAMESCTL ユーティリティがロードされている必要があります。
パスワード:	不要
構文:	NAMESCTL ユーティリティからは、次のように入力します。EXIT
引数:	なし
使用上の注意:	EXIT コマンドは、Oracle Names サーバーには適用されません。 NAMESCTL ユーティリティのみに適用されます。 このコマンドは <b>QUIT</b> コマンドと同じです。



EXIT

例：  
namsctl> exit  
NAMESCTL finished.

FLUSH

目的：ローカル・キャッシュ・チェックポイント・ファイルからすべてのリモート・リージョン情報を消去するように Oracle Names サーバーに指示します。ローカル・キャッシュ・チェックポイント・ファイルのデフォルトのファイル名は ckpcch.ora です。

前提条件：複数のリージョンで構成されている環境のみに適用されます（データを集中管理する環境には、未権限データがありません）。

パスワード：必要  
パスワードが設定されている場合、このコマンドの前に **SET PASSWORD** コマンドを実行する必要があります。

構文：オペレーティング・システム・プロンプトからは、次のように入力します。  
NAMESCTL flush [onames\_server] [onames\_server]\*  
NAMESCTL ユーティリティからは、次のように入力します。  
flush [onames\_server] [onames\_server]\*

引数：Oracle Names サーバー名を空白で区切って指定します。引数を指定しない場合は、現行の Oracle Names サーバーのキャッシュからのみ外部名が削除されます。

使用上の注意：FLUSH は、キャッシュに格納されている外部データすべてを消去します。通常は、次の場合に外部データをキャッシュから削除します。

- ネットワーク内で大量のデータを変更して、通常の TTL エイジング・メカニズムの所要時間が長すぎる場合。
- キャッシュに格納されている外部データの名前を変換するときに、識別不能なエラーが発生した場合。キャッシュからすべての外部データを削除すると、次にそのデータが要求されたときに、再び検索されます。

名前は現行の Oracle Names サーバーから削除されます。現行の Oracle Names サーバーは、デフォルトの優先 Oracle Names サーバー、または SET SERVER コマンドで設定した Oracle Names サーバーです。

例：  
NAMESCTL> flush  
Confirm [yes or no]: yes



---

**FLUSH\_NAME**

目的:	ローカル・キャッシュ・チェックポイント・ファイルから特定のリージョンの情報を消去するように Oracle Names サーバーに指示します。ローカル・キャッシュ・チェックポイント・ファイルのデフォルトのファイル名は、ckpccch.ora です。
前提条件:	このパラメータは、複数のリージョンで構成されている環境のみに適用されます。(データを集中管理する環境には、未権限データがありません)。
パスワード:	必要  パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文:	オペレーティング・システム・プロンプトからは、次のように入力します。  <code>NAMESCTL flush_name {domain}</code>  NAMESCTL ユーティリティからは、次のように入力します。  <code>flush_name {domain}</code>
引数:	シングル・ドメイン名
使用上の注意:	<p>FLUSH_NAME は、Oracle Names サーバーのリージョンの外部からキャッシュに書き込まれたデータ（つまり、未権限データ）のみを削除します。通常は、ソース・データが変更された可能性を示唆するように名前が異常に動作する場合に、外部データを削除します。</p> <p>名前は現行の Oracle Names サーバーから削除されます。現行の Oracle Names サーバーは、デフォルトの優先 Oracle Names サーバー、または SET SERVER コマンドで設定した Oracle Names サーバーです。</p>
例:	<code>NAMESCTL&gt; flush_name mountain.acme.com</code>

---

**HELP**

目的:	使用可能な NAMESCTL コマンドの一覧を表示します。HELP コマンドのうちの 1 つに応答して、NAMESCTL はコマンドの使用方法についてのヘルプを表示します。
前提条件:	なし
パスワード:	不要



---

## HELP

構文: オペレーティング・システムからは、次のように入力します。

```
NAMESCTL help
NAMESCTL help [command]
```

NAMESCTL ユーティリティからは、次のように入力します。

```
NAMESCTL help
NAMESCTL> help [command]
```

引数: Help コマンド。次の出力例にコマンドが示されています。

例:

```
NAMESCTL> help
The following operations are available:
An asterisk (*) denotes a modifier or extended command:
exit
flush
flush_name
log_stats
ping
query
quitreload
repeat*
reset_stats
restart
set*
show
shutdown
start
startup
status
stop
version
```

---

---

## LIST\_DELEGATED

目的: ルート・リージョンまたは指定したドメインの、委任されたすべてのドメインをリストします。

前提条件: なし

パスワード: 不要



---

**LIST\_DELEGATED**

構文:	オペレーティング・システムからは、次のように入力します。 NAMESCTL list_delegated [domain] NAMESCTL ユーティリティからは、次のように入力します。 NAMESCTL> list_delegated [domain]
引数:	ドメイン名
使用上の注意:	このコマンドは、Oracle Names から LDAP 準拠のディレクトリにエクスポートする前に、現行のドメイン構造を理解するのに役立ちます。 <b>関連項目:</b> 6-31 ページの「 <a href="#">Oracle Names サーバーからのネットワーク・オブジェクトのエクスポート</a> 」を参照してください。
例:	NAMESCTL> list_delegated europe.acme.com asia.acme.com africa.acme.com

---

---

**LIST\_DOMAINS**

目的:	ルート・リージョンのすべてのドメイン、または指定されたドメインのサブドメインをリストします。
前提条件:	なし
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 NAMESCTL _list_domain [domain] NAMESCTL ユーティリティからは、次のように入力します。 NAMESCTL _list_domain [domain]
引数:	ドメイン名
使用上の注意:	このコマンドは、Oracle Names から LDAP 準拠のディレクトリにエクスポートする前に、現行のドメイン構造を理解するのに役立ちます。 <b>関連項目:</b> 6-31 ページの「 <a href="#">Oracle Names サーバーからのネットワーク・オブジェクトのエクスポート</a> 」を参照してください。

---



## LIST\_DOMAINS

```
例: NAMESCTL> list_domains
com
sczi.com
us.sczi.com
sj.us.sczi.com
```

## LIST\_OBJECTS

目的: ルート・リージョンまたは指定したドメインのすべてのネットワーク・オブジェクトをリストします。

前提条件: なし

パスワード: 不要

構文: オペレーティング・システムからは、次のように入力します。

NAMECTL list objects [-R] [*domain*]

NAMESCTL ユーティリティからは、次のように入力します。

NAMECTL> list objects [-R] [*domain*]

引数: [R] - 指定したドメインの権限のあるすべてのサブドメインをリストすることを指定します。

[*domain*] - ドメイン名を指定します。

使用上の注意: このコマンドは、Oracle Names から LDAP 準拠のディレクトリにエクスポートする前に、現行のドメイン構造を理解するのに役立ちます。

**関連項目:** 6-31 ページの「Oracle Names サーバーからのネットワーク・オブジェクトのエクスポート」を参照してください

```
例: NAMESCTL> list_objects
partsdb.widgets.acme.com
toolsdb.widgets.acme.com
partsdb.components.widgets.acme.com
sparepartsdb.gadgets.widgets.acme.com
```

## LOAD\_TNSNAMES

目的: 1 つまたは複数の `tnsnames.ora` ファイルで定義されているすべてのネット・サービス名とアドレスを、Oracle Names サーバーにロードします。



---

**LOAD\_TNSNAMES**

前提条件: なし

パスワード: 不要

構文: オペレーティング・システム・プロンプトからは、次のように入力します。

```
NAMECTL load_tnsnames {tnsnames.ora} [tnsnames.ora]
```

NAMECTL ユーティリティからは、次のように入力します。

```
load_tnsnames {tnsnames.ora} [tnsnames.ora]
```

引数: 1 つまたは複数の tnsnames.ora ファイル

使用上の注意: 通常、このコマンドは tnsnames.ora ファイルを使用した後、サイトが Oracle Names の使用を開始するときに 1 度実行します。各リージョンでこのコマンドを 1 度実行してください。この操作中に定義された名前は永続的に定義され、1 つの Oracle Names サーバーからリージョン内のすべてのサーバーに伝達されます。

例:

```
NAMECTL>load_tnsnames network/admin/tnsnames.ora
Name: koala.lab.npd.us.oracle.com
Response status: normal, successful completion
Name: devdd.rdbms.us.oracle.com
Response status: normal, successful completion
Name: envyd.lab.npd.us.oracle.com
Response status: normal, successful completion
Name: stealth.npd.us.oracle.com
Response status: normal, successful completion
Name: null.us.oracle.com
Response status: normal, successful completion
Name: slime.lab.npd.us.oracle.com
Response status: normal, successful completion
Name: felix.hp.us.oracle.com
Response status: normal, successful completion
Name: dtnet1.dec.oracle.com
Response status: normal, successful completion
Name: devds.rdbms.us.oracle.com
Response status: normal, successful completion
```

---



---

**LOG\_STATS**

目的: 現行の統計を、その Oracle Names サーバーの構成済ログ・ファイルに記録します。デフォルトのログ・ファイル名は、names.log です。

前提条件: なし



---

**LOG\_STATS**

パスワード :	必要  パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文 :	オペレーティング・システム・プロンプトからは、次のように入力します。  NAMESCTL log_stats [onames_server] [onames_server]*  NAMESCTL ユーティリティからは、次のように入力します。  log_stats [onames_server] [onames_server]*
引数 :	Oracle Names サーバー名を空白で区切って指定します。引数を指定しないと、現行の Oracle Names サーバーの統計のみがリセットされます。
使用上の注意 :	<b>STATUS</b> コマンドの結果または他の動作から、データをログに記録する必要があると判断した場合は、統計を記録します。LOG_STATS は、現行のログ統計の間隔には影響しません。
例 :	NAMESCTL> log_stats Statistics counters logged.

---

---

**PASSWORD**

目的 :	<b>STOP</b> 、 <b>RESTART</b> 、 <b>RELOAD</b> などの特権的な NAMESCTL ユーティリティ・コマンドを実行するための暗号化パスワードを設定します。
前提条件 :	NAMESCTL ユーティリティがロードされている必要があります。
パスワード :	該当せず
構文 :	NAMESCTL ユーティリティからは、次のように入力します。  password [password]
引数 :	names.ora ファイル内の <b>NAMES.PASSWORD</b> パラメータで暗号化されている値と一致するテキスト文字列。

---



---

**PASSWORD**

使用上の注意:	<p>このコマンドは、names.ora ファイル内の <b>NAMES.PASSWORD</b> パラメータによってすでに確定されたパスワードは変更しません。NAMESCTL 変数のみが設定されます。この変数の値がコマンド要求時に NAMESCTL から Oracle Names サーバーに送信され、Oracle Names サーバー上で構成されている値と比較されます。これらの値が一致している場合は、パスワードを必要とする操作が許可されます。</p> <p>特権操作、つまり Oracle Names サーバーの機能を変更する操作のみが、このコマンドの影響を受けます。SHOW や STATUS などは特権操作と見なされないので、パスワードは不要です。</p> <p>PASSWORD コマンドの引数としてパスワードを指定します。引数を指定しないと、パスワードの入力を促すプロンプトが表示されます。入力したパスワードは画面に表示されません。</p> <p>パスワードがネットワークで送信されるときは、パスワードがどのように設定されていても、必ず暗号化されます。</p>
例:	<pre>NAMESCTL&gt; password open_sesame NAMESCTL&gt; password Enter name server password:</pre>

---

---

**PING**

目的:	現行の Oracle Names サーバーまたは指定した Oracle Names サーバーに接続し、要求 / 応答時間を表示します。
前提条件:	なし
パスワード:	不要
構文:	<p>オペレーティング・システム・プロンプトからは、次のように入力します。</p> <pre>NAMESCTL ping [onames_server] [onames_server]*</pre> <p>NAMESCTL ユーティリティからは、次のように入力します。</p> <pre>ping [onames_server] [onames_server]*</pre>
引数:	Oracle Names サーバー名を空白で区切って指定します。引数を指定しないと、現行の Oracle Names サーバーのみが ping されます。
使用上の注意:	ping は、Oracle Names サーバーが機能しているかどうかを確認してから、NAMESCTL ユーザーの場所から Oracle Names サーバーまでの通常の応答時間を表示します。
例:	<pre>NAMESCTL&gt; ping nserver.com Round trip time is 0.04 seconds</pre>

---



---

**QUERY**

目的:	Oracle Names サーバーに格納されているネットワーク・オブジェクトの内容を取り出します。
前提条件:	なし
パスワード:	不要
構文:	<p>オペレーティング・システム・プロンプトからは、次のように入力します。</p> <p>NAMESCTL query object_name [オブジェクト型] [修飾子]</p> <p>NAMESCTL ユーティリティからは、次のように入力します。</p> <p>query object_name [オブジェクト型] [修飾子]</p> <p><b>一般的なオブジェクト型:</b></p> <ul style="list-style-type: none"><li>■ A.SMD - データベース・サービス定義またはネット・サービス名にあるネットワーク・アドレス。</li><li>■ CNAME.SMD - 別名 (標準名 (canonical name) という場合もあります)。</li><li>■ DL.RDBMS.OMD - グローバル・データベース・リンク。</li><li>■ DLCR.RDBMS.OMD - グローバル・データベース・リンク修飾子。</li><li>■ NS.SMD - Oracle Names サーバー・アドレス。Oracle Names サーバー間で通信する際に使用するシステム・データ。</li><li>■ V1ADD.NPO.OMD - SQL*Net バージョン 1 の接続文字列。</li></ul> <p><b>有効な修飾子:</b></p> <ul style="list-style-type: none"><li>■ AUTHORITY - データがローカル・キャッシュに格納されていても、問合せをデータのソース (データがローカルとみなされる管理リージョン内) で変換します。データがソースで変更された疑いがある場合に、この修飾子を使用します。</li><li>■ NOFORWARD - データを問い合わせますが、要求は送りません。データがローカル・データでないときに NOFORWARD を指定すると、問合せが変換されません。</li><li>■ TRACE - 応答へのパスのトレースを実現します。要求の転送先 Oracle Names サーバーを調べる場合に役立ちます。</li></ul>
引数:	ネットワーク・オブジェクト名は必須



---

**QUERY**

使用上の注意：

QUERY を使用して、定義済のデータを見つけられるかどうか、およびその内容が正しいかどうかをテストすることができます。

QUERY コマンドは、常に、デフォルトまたは SET SERVER コマンドで指定した、現行の Oracle Names サーバー上で動作します。

名前のみをパラメータに指定して QUERY コマンドを使用する場合、Oracle Names サーバーはその名前を持つデータの個数と操作の実行時間を知らせてきます。

QUERY コマンドの引数として名前と型を指定すると、特定の名前が検索されて、ユーザーに戻されます。

QUERY コマンドでは、複数の引数を指定できます。たとえば、次のように指定します。

```
query sales.com a.smd authority trace
```

例：

```
NAMESCTL> query bones.dem.medicine a.smd
Total response time:0.04 seconds
Response status:normal, successful completion
Authoritative answer:yes
Number of answers:1
Canonical name:bones.dem.medicine
TTL: 1 day
Alias translations:
    from:bones.dem.medicine
    to: bones.dem.medicine
Answers:
    data type is "a.smd"
    Syntax is ADDR:... (DESCRIPTION= (ADDRESS=
(PROTOCOL=TCP) (Host=bones-pc)
(Port=1521)) (CONNECT_DATA=(SERVICE_NAME=bones.dem.medicine)))
```

---

---

**QUIT**

目的： NAMESCTL ユーティリティを終了します。

前提条件： NAMESCTL ユーティリティがロードされている必要があります。

パスワード： 不要

構文： NAMESCTL ユーティリティからは、次のように入力します。

```
QUIT
```

引数： なし



QUIT

使用上の注意 : QUIT コマンドは、Oracle Names サーバーには適用されず、NAMESCTL ユーティリティのみに適用されます。

QUIT コマンドは EXIT コマンドと同様に機能します。

例 : NAMESCTL> quit  
nl-00851: NAMESCTL が終了しました。

REGISTER

目的 : Oracle Names サーバーにネットワーク・オブジェクトを登録します。

パスワード : 不要

構文 : NAMESCTL ユーティリティからは、次のように入力します。

```
register object_name [-t サービスの種類]
[-d [(description=) (address=( プロトコル・アドレス情報)) ()]] [-h
onames_server]
[-l listener_name]
```

**サービスの種類 :**

- ORACLE\_DATABASE
- ORACLE\_LISTENER
- ORACLE\_NAMESERVER

**関連項目 :** プロトコル・アドレスとパラメータの詳細は、付録 B を参照してください。

引数 : オブジェクト名は必須です。サービスおよびアドレス・データ、ホストを指定しなくても、登録プロセスを実行できます。ただし、登録したオブジェクトを使用するには、これらのデータが必要です。つまり、アドレスを指定しないで登録したオブジェクト名は使用できません。

使用上の注意 : REGISTER は、サービスおよびサービスのタイプ、ホスト名、アドレスを手動で登録するためのコマンドです。サービスのタイプとデータを有効な文字列で指定できますが、一般的な登録では、サービスの種類として database または listener を指定し、データとして TNS アドレスを指定します。

オブジェクトの登録は、リージョン内の他のすべての予約済 Oracle Names サーバーに伝えられます。

例 : NAMESCTL> register parts -t oracle\_database -d (DESCRIPTION= (ADDRESS= (PROTOCOL=TCP) (HOST=nineva) (PORT=1575)) (CONNECT\_ DATA=(SERVICE\_NAME=db3)))



---

**RELOAD**

目的:	Oracle Names サーバーの管理リージョン内でデータが変更されたかどうかをただちにチェックします。変更されたデータがある場合、Oracle Names サーバーはすべてのデータベース・サービス名、ネット・サービス名、グローバル・データベース・リンク、および別名を再ロードします。
前提条件:	なし
パスワード:	必要  パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文:	オペレーティング・システム・プロンプトからは、次のように入力します。  <code>NAMESCTL reload [onames_server] [onames_server]*</code>  NAMESCTL ユーティリティからは、次のように入力します。  <code>reload [onames_server] [onames_server]*</code>
引数:	Oracle Names サーバー名を空白で区切って指定します。引数を指定しないと、現行の Oracle Names サーバーのみが再ロードされます。
使用上の注意:	すべての Oracle Names サーバーは、names.ora ファイルの NAMES.ADMIN_REGION 構成パラメータで指定されているデータベースから、データを直接ロードします。  複数のリージョンで構成されている環境では、RELOAD は現行の管理リージョンのデータのみに適用されます。キャッシュ内の外部データはすべて変更されません。
例:	<code>NAMESCTL&gt; reload</code> <code>Server reloaded.</code>

---

---

**REORDER\_NS**

目的:	ローカル Oracle Names サーバーとそのアドレスがリストされたファイルを作成します。
前提条件:	なし
パスワード:	不要



REORDER\_NS

構文:	<p>オペレーティング・システム・プロンプトからは、次のように入力します。</p> <p>NAMESCTL reorder_ns [(description=) (address=(プロトコル・アドレス情報)) []]</p> <p>NAMESCTL ユーティリティからは、次のように入力します。</p> <p>reorder_ns [(description=) (address=(プロトコル・アドレス情報)) []]</p> <p><b>関連項目:</b> プロトコル・アドレスとパラメータの詳細は、<a href="#">付録 B</a> を参照してください。</p>
引数:	<p>オプションの Oracle Names サーバー・アドレスが、接続するための最初のサーバーとして使用されます。</p>
使用上の注意:	<p>このコマンドは、クライアントが名前を調べるために Oracle Names サーバーに接続できるように、Oracle Names サーバーの名前とアドレスが定義されたファイルを生成します。</p> <p>REORDER_NS コマンドは、次のタスクを実行します。</p> <ol style="list-style-type: none"><li>1. Oracle Names サーバーを検索します。sqlnet.ora で優先 Oracle Names サーバー・パラメータを探します。続いて、予約済のサーバーを呼び出すか、ポート 1575 の TCP/IP を使用してローカル Oracle Names サーバーを検索します。</li><li>2. ローカル・リージョン内のすべての Oracle Names サーバーに対して問合せを送ります。</li><li>3. これらのサーバーのそれぞれに PING を送ります。</li><li>4. サーバーのリストを応答時間の短い順にソートします。</li><li>5. Oracle Names サーバー・リストにソートした名前とアドレスのリストを書き込みます。</li></ol>
例:	<p>NAMESCTL&gt; reorder_ns (address=(protocol=tcp) (host=nineva) (port=1575))</p>

REPEAT

目的:	<p><a href="#">QUERY</a>、<a href="#">REGISTER</a>、<a href="#">TIMED_QUERY</a> または <a href="#">UNREGISTER</a> を複数回実行して、平均戻り速度を計算します。</p>
前提条件:	<p>なし</p>
パスワード:	<p>不要</p>



---

**REPEAT**

構文: オペレーティング・システム・プロンプトからは、次のように入力します。

NAMESCTL repeat *number* query *type*

NAMESCTL ユーティリティからは、次のように入力します。

repeat *number* query *type*

引数: *number* - 整数

*type* - オブジェクト・タイプ。[QUERY](#) コマンドでの指定と同様。

使用上の注意: REPEAT コマンドは、要求の平均応答時間を知るのに役立ちます。

実行回数として、あまり大きな数を指定しないでください。実行を繰り返している間、NAMESCTL ユーティリティは他の作業を実行できません。

例:

```
NAMESCTL> repeat 10 query manatee a.smd
Number of requests: 10
Average response time: 0.01 seconds
Minimum response time: 0.01 seconds
Maximum response time: 0.04 seconds
Total response time: 0.14 seconds
Response status: normal, successful completion
Authoritative answer: yes
Number of answers: 1
TTL: 1 day
Answers:
    data type is "a.smd"
        Syntax is ADDR: (DESCRIPTION= (ADDRESS=
(PROTOCOL=TCP) (Host=salmon)
(Port=1522)) (CONNECT_DATA= (SID=otter)))
```

---

---

**RESET\_STATS**

目的: Oracle Names サーバーの統計を Oracle Names サーバーの起動時の元の値にリセットします。

前提条件: なし

パスワード: 必要

パスワードが設定されている場合、このコマンドの前に [SET PASSWORD](#) コマンドを実行する必要があります。



---

**RESET\_STATS**

構文:	オペレーティング・システム・プロンプトからは、次のように入力します。  NAMESCTL reset_stats [onames_server] [onames_server] *  NAMESCTL ユーティリティからは、次のように入力します。  reset_stats [onames_server] [onames_server] *
引数:	Oracle Names サーバー名を空白で区切って指定します。引数を指定しないと、現行の Oracle Names サーバーの統計のみがリセットされます。
使用上の注意:	統計がただちにリセットされることを除けば、RESET_STATS は、RESET_STATS_INTERVAL (NAMES.ORA パラメータ) が終了するのを待つと同じことです。
例:	NAMESCTL> reset_stats Confirm [yes or no]: yes Server statistics reset.

---

---

**RESTART**

目的:	Oracle Names サーバーを起動時の元の状態にリセットします。
前提条件:	なし
パスワード:	必要  パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文:	オペレーティング・システム・プロンプトからは、次のように入力します。  NAMESCTL restart [onames_server] [onames_server] *  NAMESCTL ユーティリティからは、次のように入力します。  restart [onames_server] [onames_server] *
引数:	Oracle Names サーバー名を空白で区切って指定します。引数を指定しないと、現行の Oracle Names サーバーのみが再起動されます。
使用上の注意:	Oracle Names サーバーがすでに稼働しているという点を除いて、RESTART は STARTUP と同じです。  データが再ロードされ、統計がリセットされ、すべての外部データが削除されます。有効な外部キャッシュ・データ（つまり、TTL が 0 より大きいデータ）が、チェックポイント・ファイルから取り出されます（TTL の値は、0 より大きく設定する必要があります）。

---



---

**RESTART**

例 :                   NAMECTL> restart  
                      Confirm [yes or no]: yes  
                      Server restarted.

---

---

**SET**

目的 :                現行の NAMECTL セッションに対して設定できる使用可能な構成コマンドのリストを表示します。

前提条件 :           なし

パスワード :         不要

構文 :                オペレーティング・システムからは、次のように入力します。

```
NAMECTL set  
NAMECTL set [command]
```

NAMECTL ユーティリティからは、次のように入力します。

```
NAMECTL> set  
NAMECTL set [command]
```

引数 :                SET コマンド。次の出力例にコマンドが示されています。

例 :                   NAMECTL> set  
                      The following operations are available after set  
                      An asterisk (\*) denotes a modifier or extended command:  
                      cache\_checkpoint\_interval  
                      default\_domain  
                      forwarding\_available  
                      log\_file\_name  
                      log\_stats\_interval  
                      NAMECTL\_trace\_level  
                      password  
                      requests\_enabled  
                      reset\_stats\_interval  
                      save\_config\_interval  
                      save\_config\_on\_stop  
                      server  
                      trace\_file\_name  
                      trace\_level

---



---

### SET CACHE\_CHECKPOINT\_INTERVAL

目的:	リモート・リージョンについて収集したすべての情報を、ローカル・キャッシュ・ファイルに保存する頻度を設定します。デフォルトのローカル・キャッシュ・ファイルは、ckpcch.ora です。
パスワード:	不要
構文:	<p>オペレーティング・システムからは、次のように入力します。</p> <pre>NAMESCTL set cache_checkpoint_interval time</pre> <p>NAMESCTL ユーティリティからは、次のように入力します。</p> <pre>set cache_checkpoint_interval time</pre>
引数:	<p>time は秒単位です。</p> <p>たとえば、CACHE_CHECKPOINT_INTERVAL を 36 時間に設定する場合は、次のように入力します。</p> <pre>set cache_checkpoint_interval 129600</pre>
使用上の注意:	<p>最小値: 10 秒</p> <p>最大値: 259200 (3 日)</p> <p>デフォルト: 0 (キャッシュ頻度なし)</p>
例:	<pre>NAMESCTL&gt; set cache_checkpoint_interval 10</pre>

---

### SET DEFAULT\_DOMAIN

目的:	NAMESCTL クライアントが最も頻繁に名前解決要求を検索するドメインを設定したり、sqlnet.ora ファイルの <a href="#">NAMES.DEFAULT_DOMAIN</a> で指定されたドメインを変更します。
前提条件:	NAMESCTL ユーティリティがロードされている必要があります。
パスワード:	不要
構文:	<p>NAMESCTL ユーティリティからは、次のように入力します。</p> <pre>set default_domain domain_name</pre>
引数:	ドメイン名



---

**SET DEFAULT\_DOMAIN**

使用上の注意:           デフォルト・ドメインを設定すると、すべての未修飾のネット・サービス名やサービス名にデフォルト・ドメインが付加されます。たとえば、デフォルト・ドメインを us.acme に設定している場合、グローバル名 sales.us.acme は、次のようにして問い合わせることができます。

```
NAMESCTL> query sales
```

例:                       NAMESCTL> set default\_domain us.acme.com  
Default domain is now "US.ACME.COM"

---

---

**SET FORWARDING\_AVAILABLE**

目的:                   リモート Oracle Names サーバーへのクライアント名前要求の転送機能をオンまたはオフにします。

前提条件:              Oracle Names サーバーが稼働している必要があります。

パスワード:            必要

パスワードが設定されている場合、このコマンドの前に **SET PASSWORD** コマンドを実行する必要があります。

構文:                   オペレーティング・システムからは、次のように入力します。

```
NAMESCTL set forwarding_available on|off|yes|no
```

NAMESCTL ユーティリティからは、次のように入力します。

```
set forwarding_available off
```

引数:                   ON | OFF | YES | NO

デフォルト値:           OFF

---



---

**SET FORWARDING\_AVAILABLE**

使用上の注意：                      この設定が適用されるのは、ローカル・クライアントを持たず、リモート Oracle Names サーバーからの要求のみを処理する Oracle Names サーバーのみです。通常は、クライアントまたはサービスをもたないルート・リージョン内の Oracle Names サーバーのみに適用されます。このような Oracle Names サーバーがリージョン間の要求処理でボトルネックの原因となる場合は、転送機能をオフにすると、その Oracle Names サーバーの作業負荷を半減できます。転送機能がオフに設定された Oracle Names サーバーは、要求を転送して応答を戻すかわりに、要求に応答できる Oracle Names サーバーのアドレスを要求側に知らせます。この場合、作業負荷の全体量は減少せずに、転送機能のない Oracle Names サーバーから要求側の Oracle Names サーバーに作業負荷が移動するのみです。

**警告：** SET FORWARDING\_AVAILABLE をオフに設定すると、その Oracle Names サーバーに直接依存しているすべてのクライアントはリモート名を変換できません。クライアントは、Oracle Names サーバーのように要求をリダイレクトできません。NAMES.PREFERRED\_SERVERS 構成パラメータに他の Oracle Names サーバーがリストされていても、クライアントの要求は失敗します。

例：                                      NAMESCTL> set forwarding\_available off  
Request processing is now disabled.

---

---

**SET LOG\_FILE\_NAME**

目的：                                      Oracle Names サーバーのログ・ファイル名を設定します。デフォルトのログ・ファイル名は、names.log です。

前提条件：                                なし

パスワード：                              必要  
パスワードが設定されている場合、このコマンドの前に **SET PASSWORD** コマンドを実行する必要があります。

構文：                                      オペレーティング・システムからは、次のように入力します。  
NAMESCTL set log\_file\_name file\_name  
NAMESCTL ユーティリティからは、次のように入力します。  
set log\_file\_name file\_name

引数：                                      ログ・ファイルのファイル名

使用上の注意：                            LOG\_FILE\_NAME は、すべてのロギング・メッセージの宛先を変更します。



---

**SET LOG\_FILE\_NAME**

例: `NAMESTL> set log_file_name namesvr1`

---

---

**SET LOG\_STATS\_INTERVAL**

目的: 統計をログ・ファイルに記録する頻度を変更します。デフォルトのログ・ファイル名は、`names.log` です。

前提条件: なし

パスワード: 必要

パスワードが設定されている場合、このコマンドの前に **SET PASSWORD** コマンドを実行する必要があります。

構文: オペレーティング・システムからは、次のように入力します。

```
NAMESTL set log_stats_interval time
```

NAMESTL ユーティリティからは、次のように入力します。

```
NAMESTL> set log_stats_interval time
```

引数: `time` は、秒数、または `[<n> DAY[S]] [<hh>:<mi>:<ss>]` で指定します。  
たとえば、`LOG_STATS_INTERVAL` を 36 時間に設定する場合は、次の両方の設定が可能です。

```
set log_stats_interval 129600
```

```
set log_stats_interval 1 day 12:00:00
```

日数と時間:分:秒を組み合わせで指定するか、日数を指定しないで時間のみを指定できます。

制限: 最小値: 10 秒

最大値: 制限なし

特殊値: 0 (リセットされないことを意味します)

デフォルト値: 0 (記録しません)

使用上の注意: `LOG_STATS_INTERVAL` の初期値は、Oracle Names サーバーのロード時に、`sqlnet.ora` ファイルの `NAMES.LOG_STATS_INTERVAL` パラメータに構成されている値に基づいて設定されます。デフォルト値は 0 (記録しません) です。`SET LOG_STATS_INTERVAL` は、サーバーの稼働中にこの値をオーバーライドするためのコマンドです。

例: `NAMESTL> set log_stats_interval 7200`  
`Statistic counter logging interval is now 2 hours`

---



SET NAMESCTL_TRACE_LEVEL	
目的:	NAMESCTL ユーティリティのトレース・レベルを指定します。
前提条件:	なし
パスワード:	必要 パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文:	オペレーティング・システムからは、次のように入力します。 NAMESCTL set NAMESCTL_trace_level level NAMESCTL ユーティリティからは、次のように入力します。 NAMESCTL> set NAMESCTL_trace_level level
引数:	OFF - トレースを実行しません。 USER - ユーザーによるエラー条件を判別するためのトレースを実行します。 ADMIN - インストール固有の問題を判別するためのトレースを実行します。 SUPPORT - サポートに必要なレベルのトレースを実行します。
使用上の注意:	トレースは、NAMESCTL ユーティリティの処理で発生する予測不能な障害または識別不能な障害を診断するのに役立ちます。トレースは、通常の NAMESCTL 処理で行われる一連のイベントをオペレーティング・システム・ファイルに書き込んで、管理者が検査できるようにします。 トレース出力は、OFF（なし）、USER（基本的情報）または ADMIN という3段階のレベルで指定します。 引数を指定しないと、クライアントの sqlnet.ora ファイル内の値にリセットされます。デフォルトの設定は、OFF です。
例:	NAMESCTL> set NAMESCTL_trace_level admin Controller's local trace level changed from 0 to 4

SET PASSWORD	
目的:	<b>STOP</b> 、 <b>RESTART</b> 、 <b>RELOAD</b> などの特権的な NAMESCTL ユーティリティ・コマンドを実行するためのパスワードを設定します。 入力するパスワードは、names.ora ファイル内の <b>NAMES.PASSWORD</b> パラメータの1つのセットと一致している必要があります。



---

**SET PASSWORD**

前提条件:	NAMESCTL ユーティリティがロードされている必要があります。
パスワード:	該当せず
構文:	<p>NAMESCTL ユーティリティからは、次のように入力します。</p> <pre>NAMESCTL&gt; set password [password]</pre>
引数:	現行の Oracle Names サーバーのパラメータ <b>NAMES.PASSWORD</b> に格納されている値と一致するテキスト文字列
使用上の注意:	<p>SET PASSWORD では、Oracle Names サーバーのパスワードは変更されません。この変数の値が、NAMESCTL コマンドで Oracle Names サーバーに送信され、Oracle Names サーバー上で構成されている値と比較されます。これらの値が一致している場合は、パスワードを必要とする操作が許可されます。</p> <p>特権操作、つまり Oracle Names サーバーの機能を変更する操作のみが、このコマンドの影響を受けます。SHOW や STATUS などは特権操作と見なされないので、パスワードは不要です。</p> <p>SET PASSWORD コマンドの引数としてパスワードを指定します。引数を指定しないと、パスワードの入力を促すプロンプトが表示されません。入力したパスワードは画面に表示されません。</p> <p>パスワードがネットワークで送信されるときは、パスワードがどのように設定されていても、必ず暗号化されます。</p>
例:	<pre>NAMESCTL&gt; set password open_sesame NAMESCTL&gt; set password enter name server password:</pre>

---

**SET REQUESTS\_ENABLED**

目的:	現行の Oracle Names サーバーが要求に応答するかどうかを決定します。
前提条件:	なし
パスワード:	必要
構文:	<p>パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。</p> <p>オペレーティング・システムからは、次のように入力します。</p> <pre>NAMESCTL set requests_enabled on off</pre> <p>NAMESCTL ユーティリティからは、次のように入力します。</p> <pre>NAMESCTL&gt; set requests_enabled on off</pre>



SET REQUESTS\_ENABLED

引数: ON | OFF

使用上の注意: このプロパティを OFF に設定すると、名前解決を要求するすべてのクライアントが拒否されます。Oracle Names サーバーが予期せず動いているときは、この設定を使用してネーム・サーバーを診断できます。

例: NAMESCTL> set requests\_enabled off  
Confirm [yes or no]: yes  
General request processing is now disabled

SET RESET\_STATS\_INTERVAL

目的: 統計がゼロまたはカレント・サーバーの初期値にリセットされる間隔を変更します。

前提条件: なし

パスワード: 必要  
パスワードが設定されている場合、このコマンドの前に **SET PASSWORD** コマンドを実行する必要があります。

構文: オペレーティング・システムからは、次のように入力します。  
NAMESCTL set reset\_stats\_interval time  
NAMESCTL ユーティリティからは、次のように入力します。  
NAMESCTL> set reset\_stats\_interval time

引数: time は、秒数または [n DAY[S]] [hh:mi:ss] の形式で指定します。  
たとえば、RESET\_STATS\_INTERVAL を 72 時間に増やすには、次の両方の設定が可能です。  
set reset\_stats\_interval 259200  
set reset\_stats\_interval 3 days

制限: 最小値: 10 秒  
最大値: 制限なし  
デフォルト値: 0 (リセットしません)

使用上の注意: RESET\_STATS\_INTERVAL の初期値は、Oracle Names サーバーをロードするときの NAMES.RESET\_STATS\_INTERVAL パラメータに基づいて設定されます。SET RESET\_STATS\_INTERVAL は、Oracle Names サーバーの稼働中にこの値を上書きするためのコマンドです。

例: NAMESCTL> set reset\_stats\_interval 1 day  
Statistic counter reset interval is now 24 hours



---

## SET SAVE\_CONFIG\_INTERVAL

目的:	NAMESCTL SET コマンドで行った変更が、一定間隔で <code>names.ora</code> に保存されます。
パスワード:	必要  パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文:	オペレーティング・システムからは、次のように入力します。 <code>NAMESCTL set save_config_interval time</code> NAMESCTL ユーティリティからは、次のように入力します。 <code>set save_config_interval time</code>
引数:	<code>time</code> は秒単位です。
例:	<code>NAMESCTL&gt; set save_config_interval 10</code>

---

---

## SET SAVE\_CONFIG\_ON\_STOP

目的:	NAMESCTL セッション終了後、NAMESCTL SET コマンドによって行われた変更を <code>names.ora</code> ファイルに保存するかどうかを指定します。  NAMESCTL セッションが終了する直前にすべての値が保存されます。形式、コメント、大文字と小文字は、可能な限り維持されます。
パスワード:	必要  パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文:	オペレーティング・システムからは、次のように入力します。 <code>NAMESCTL set save_config_on_stop on off</code> NAMESCTL ユーティリティからは、次のように入力します。 <code>NAMESCTL&gt; set save_config_on_stop on off</code>
引数:	ON   OFF
例:	<code>NAMESCTL&gt; set save_config_on_stop on</code>

---

---

## SET SERVER

目的:	Oracle Names サーバーの名前を設定します。
-----	-----------------------------



SET SERVER

前提条件:	NAMESCTL ユーティリティがロードされている必要があります。
パスワード:	不要
構文:	<p>NAMESCTL ユーティリティからは、次のように入力します。</p> <pre>NAMESCTL&gt; set server [onames_server] [(description=] (address=( プロトコル・アドレス情報 )) []]</pre> <p><b>関連項目:</b> プロトコル・アドレスとパラメータの詳細は、<a href="#">付録 B</a> を参照してください。</p>
引数:	<p>有効な Oracle Names サーバーまたは有効な Oracle Names サーバー・アドレス</p> <p>引数を指定しない場合、sqlnet.ora ファイルの <a href="#">NAMES.PREFERRED_SERVERS</a> パラメータによって設定されている値を使用します。</p>
使用上の注意:	<p>SET SERVER を使用すると、NAMESCTL ユーティリティの実行中に、Oracle Names サーバーを切り替えることができます。修飾子として指定できるのは、現行の Oracle Names サーバーのメモリーに定義されている名前、または任意の Oracle Names サーバーの TNS アドレスです。</p> <p>指定した Oracle Names サーバー名は、通常の名前検索で変換されません。現行の Oracle Names サーバーがアドレスを知っているか取り出せる場合のみ、別の Oracle Names ネーム・サーバーを設定できます。現行の Oracle Names サーバーが設定されていない場合は、アドレスを入力しないとこのコマンドを実行できません。</p>
例:	<pre>NAMESCTL&gt; set server server1.us.acme</pre>

SET TRACE\_FILE\_NAME

目的:	Oracle Names サーバーのトレース・ファイル名を設定します。デフォルトのトレース・ファイル名は names.trc です。
前提条件:	なし
パスワード:	不要
構文:	<p>オペレーティング・システムからは、次のように入力します。</p> <pre>NAMESCTL set trace_file_name file_name</pre> <p>NAMESCTL ユーティリティからは、次のように入力します。</p> <pre>NAMESCTL&gt; set trace_file_name file_name</pre>
引数:	ファイル名



---

**SET TRACE\_FILE\_NAME**

例: NAMESCTL> set trace\_file\_name namesvr1

---

---

**SET TRACE\_LEVEL**

目的: Oracle Names サーバーのトレースのレベルを設定します。

前提条件: なし

パスワード: 必要

パスワードが設定されている場合、このコマンドの前に **SET PASSWORD** コマンドを実行する必要があります。

構文: オペレーティング・システムからは、次のように入力します。

```
NAMESCTL set trace_level level
```

NAMESCTL ユーティリティからは、次のように入力します。

```
NAMESCTL> set trace_level level
```

引数: トレース・レベル:

- OFF - トレース出力しない
- USER - ユーザー・トレース情報
- ADMIN - 管理トレース情報
- SUPPORT - オラクル社カスタマ・サポート・センター用のトレース情報

使用上の注意: トレースは、現行の Oracle Names サーバーの処理で発生する予測不能な障害、または識別不能な障害を診断するのに役立ちます。トレースは、通常の Oracle Names サーバー処理で行われる一連のイベントをオペレーティング・システム・ファイルに書き込んで、管理者が検査できるようにします。

TRACE\_LEVEL を設定すると、トレースがただちに開始されます。トレース・レベルを OFF にリセットするまで、すべての操作がトレースされます。

トレース・ファイルは非常に大きくなります。問題を診断した後は、トレース・レベルを必ずオフにしてください。

例: NAMESCTL> set trace\_level admin  
Trace level is now 6.

---



---

**SHOW**

目的: 現在の NAMESCTL セッションについて表示できる、使用可能なコマンドのリストを表示します。NAMESCTL は SHOW コマンドのうちの 1 つに回答して、そのパラメータの現在の設定を表示します。

SET PASSWORD を除くすべての SET コマンドには、同等の SHOW コマンドがあります。

前提条件: なし

構文: オペレーティング・システムからは、次のように入力します。

```
NAMESCTL show
NAMESCTL show [command]
```

NAMESCTL ユーティリティからは、次のように入力します。

```
NAMESCTL> show
NAMESCTL> show [command]
```

引数: NAMESCTL> show

The following operations are available after show

アスタリスク (\*) は、修飾子コマンドまたは拡張コマンドを示します。

```
cache_checkpoint_interval
default_domain
forwarding_available
log_file_name
log_stats_interval
NAMESCTL_trace_level
requests_enabled
reset_stats_interval
save_config_interval
save_config_on_stop
server
status
system_queries
trace_file_name
trace_level
version
```

例: CMCTL> show trace\_level

---



---

**SHOW CACHE\_CHECKPOINT\_INTERVAL**

目的:	Oracle Names サーバーのキャッシュがキャッシュ・チェックポイント・ファイルに書き込まれる頻度を表示します。デフォルトのキャッシュ・チェックポイント・ファイルは、ckpcch.ora です。
前提条件:	なし
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 <code>NAMECTL show cache_checkpoint_interval</code> NAMECTL ユーティリティからは、次のように入力します。 <code>NAMECTL&gt; show cache_checkpoint_interval</code>
引数:	なし
使用上の注意:	CACHE_CHECKPOINT_INTERVAL の初期値は、names.ora ファイルの <a href="#">NAMES.CACHE_CHECKPOINT_INTERVAL</a> の値で設定されます。デフォルト値は 0 で、キャッシュ・チェックポイントは機能しません。キャッシュ・チェックポイント・ファイルに書き込まれるデータには、ネット・サービスの名前とアドレス、およびクライアントにかわって Oracle Names サーバーが外部リジョンに問合せを転送して取得した Oracle Names サーバー・アドレスなどがあります。
例:	<code>NAMECTL&gt; show cache_checkpoint_interval</code> Cache checkpoint interval is currently 8 minutes 20 seconds

---

---

**SHOW DEFAULT\_DOMAIN**

目的:	<a href="#">SET DEFAULT_DOMAIN</a> または sqlnet.ora ファイルの <a href="#">NAMES.DEFAULT_DOMAIN</a> によって設定されたドメインを表示します。
前提条件:	なし
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 <code>NAMECTL show default_domain</code> NAMECTL ユーティリティからは、次のように入力します。 <code>NAMECTL&gt; show default_domain</code>
引数:	なし



---

## SHOW DEFAULT\_DOMAIN

使用上の注意: デフォルト・ドメインを設定すると、すべての未修飾のネット・サービス名やサービス名にデフォルト・ドメインが付加されます。たとえば、デフォルト・ドメインを `us.acme` に設定している場合、グローバル名 `sales.us.acme` は、次のようにして問い合わせることができます。

```
NAMESCTL> query sales
```

例: 

```
NAMESCTL> show default_domain  
Current default domain is "com"
```

---

---

## SHOW FORWARDING\_AVAILABLE

目的: Oracle Names サーバーがリモート Oracle Names サーバーにクライアント要求を転送するかどうかを表示します。

前提条件: なし

パスワード: 不要

構文: オペレーティング・システムからは、次のように入力します。

```
NAMESCTL show forwarding_available
```

NAMESCTL ユーティリティからは、次のように入力します。

```
NAMESCTL> show forwarding_available
```

引数: Oracle Names サーバー名を空白で区切って指定します。名前を指定しないと、現行のネーム・サーバーの設定が表示されます。

使用上の注意: デフォルトでは、すべての Oracle Names サーバーがリモート Oracle Names サーバーに要求を転送します。転送が使用禁止になっている場合、リモート Oracle Names サーバーへの要求は、要求された名前に対して権限を持つリージョン内の Oracle Names サーバーにリダイレクトされます。

転送を使用禁止にすると、特定の Oracle Names サーバーのロードを軽減できますが、リモート Oracle Names サーバーにクライアントを送信することができなくなります。

転送機能をオンまたはオフにするには、[SET FORWARDING\\_AVAILABLE](#) コマンドを使用します。

例: 

```
NAMESCTL> show forwarding_available  
Request forwarding is currently enabled
```

---



---

**SHOW LOG\_FILE\_NAME**

目的:	Oracle Names サーバーがロギング情報を書き込むファイルの名前を表示します。
前提条件:	なし
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 <code>NAMESCTL show log_file_name</code> NAMESCTL ユーティリティからは、次のように入力します。 <code>NAMESCTL&gt; show log_file_name</code>
引数:	なし
使用上の注意:	ログ・ファイル名の初期値は、names.ora ファイルの <b>NAMES.LOG_FILE</b> の値で設定されます。デフォルトのファイル名は names.log です。
例:	<code>NAMESCTL&gt; show log_file_name</code> Log file name is currently /private/ora23/network/names.log

---

---

**SHOW LOG\_STATS\_INTERVAL**

目的:	統計がログ・ファイルに記録される頻度を表示します。
前提条件:	なし
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 <code>NAMESCTL show [onames_server] [onames_server]* log_stats_interval</code> NAMESCTL ユーティリティからは、次のように入力します。 <code>NAMESCTL&gt; show [onames_server] [onames_server]* log_stats_interval</code>
引数:	Oracle Names サーバー名を空白で区切って指定します。名前を指定しないと、現行のネーム・サーバーの設定が表示されます。
使用上の注意:	LOG_STATS_INTERVAL の初期値は、names.ora ファイルの NAMES.LOG_STATS_INTERVAL の値に設定されます。デフォルト値は 0 で、ロギングは実行されません。
例:	<code>NAMESCTL&gt; show log_stats_interval</code> Statistic counter logging is currently disabled

---



SHOW NAMESCTL_TRACE_LEVEL	
目的:	NAMESCTL ユーティリティのトレース・レベルを表示します。
前提条件:	なし
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 NAMESCTL show NAMESCTL_trace_level NAMESCTL ユーティリティからは、次のように入力します。 NAMESCTL> show NAMESCTL_trace_level
引数:	なし
使用上の注意:	トレースは、NAMESCTL ユーティリティの処理で発生する予測不能な障害または識別不能な障害を診断するのに役立ちます。トレースは、通常の NAMESCTL 処理で行われる一連のイベントをオペレーティング・システム・ファイルに書き込んで、管理者が検査できるようにします。  トレース出力は、OFF（なし）、USER（基本的情報）または ADMIN（最大量の情報）という3段階のレベルで指定します。
例:	NAMESCTL> show NAMESCTL_trace_level Controller's trace level is currently 0

SHOW REQUESTS_ENABLED	
目的:	Oracle Names サーバーが要求に応答しているかどうかを表示します。
前提条件:	なし
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 NAMESCTL show [onames_server] [onames_server]* requests_enabled NAMESCTL ユーティリティからは、次のように入力します。 NAMESCTL> show [onames_server] [onames_server]* requests_enabled
引数:	Oracle Names サーバー名を空白で区切って指定します。名前を指定しないと、現行のネーム・サーバーの設定が表示されます。
使用上の注意:	要求が使用禁止の場合、Oracle Names サーバーへのすべての要求が拒否されます。



---

**SHOW REQUESTS\_ENABLED**

例: `NAMECTL> show requests_enabled`  
`General request processing is currently enabled`

---

---

**SHOW RESETS\_STATS\_INTERVAL**

目的: 統計をログ・ファイルにダンプする間隔を表示します。

前提条件: なし

パスワード: 不要

構文: オペレーティング・システムからは、次のように入力します。

`NAMECTL show reset_stats_interval`

NAMECTL ユーティリティからは、次のように入力します。

`NAMECTL> show reset_stats_interval`

使用上の注意: RESET\_STATS\_INTERVAL の初期値は、NAMES.RESET\_STATS\_INTERVAL の値に設定されます。デフォルト値は 0 (リセットしません) です。この場合は、Oracle Names サーバーが稼働している全時間の統計が蓄積されます。たとえば、統計が毎日リセットされる場合、統計はサーバーが動作していた総時間ではなく当日の合計を表します。

例: `NAMECTL> show reset_stats_interval`  
`Statistic counter reset interval is currently 5 minutes`

---

---

**SHOW SAVE\_CONFIG\_INTERVAL**

目的: names.ora ファイルに保存するために、SET コマンドでスケジュールされている間隔を秒数で表示します。

前提条件: このコマンドは、SET SAVE\_CONFIG\_INTERVAL コマンドが設定された場合のみ、間隔を表示します。

パスワード: 不要

構文: オペレーティング・システムからは、次のように入力します。

`NAMECTL show save_config_interval`

NAMECTL ユーティリティからは、次のように入力します。

`show save_config_interval`

例: `NAMECTL> set save_config_interval 10`

---



SHOW SAVE_CONFIG_ON_STOP	
目的:	names.ora ファイルに変更を保存するために、NAMESCTL SET コマンドがスケジュールされているかどうかを示します。
前提条件:	なし
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 NAMESCTL show save_config_on_stop NAMESCTL ユーティリティからは、次のように入力します。 NAMESCTL> show save_config_on_stop
引数:	なし
例:	NAMESCTL> show save_config_on_stop

SHOW SERVER	
目的:	現行の Oracle Names サーバー名を表示します。
前提条件:	なし
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 NAMESCTL show server NAMESCTL ユーティリティからは、次のように入力します。 NAMESCTL> show server
引数:	なし
使用上の注意:	SHOW SERVER は、コマンドが適用される現行の Oracle Names サーバーを表示します。
例:	NAMESCTL> show server currently managing name server "NameServer.us.oracle.com" Version banner is "Oracle Names for SunOS: Version 8.1.6.0.0"

SHOW STATUS	
目的:	Oracle Names サーバーの一般的なステータス情報を表示します。
前提条件:	なし
パスワード:	不要



---

**SHOW STATUS**

構文	オペレーティング・システムからは、次のように入力します。 NAMECTL show status [onames_server] [onames_server]* NAMECTL ユーティリティからは、次のように入力します。 NAMECTL> show status [onames_server] [onames_server]*
引数:	Oracle Names サーバー名を空白で区切って指定します。名前を指定しないと、現行のネーム・サーバーの設定が表示されます。
使用上の注意:	Oracle Names サーバーの現在の状態を表示します。 このコマンドは <b>STATUS</b> コマンドと同じです。
例:	NAMECTL> show status Version Banner is "Oracle Names for SunOS: Version 8.1.6.0.0" Server has been running for:1 day 2 hours 3 minutes 35.16 seconds....

---

---

**SHOW SYSTEM\_QUERIES**

目的:	すべてのシステム問合せの次のオカレンスを表示します。
前提条件:	このコマンドは分散構成のみに適用されます。1つの管理リージョンのみで構成されている環境では、システム問合せが実行されません。
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 NAMECTL show system_queries NAMECTL ユーティリティからは、次のように入力します。 NAMECTL> show system_queries
引数:	なし
使用上の注意:	システム問合せは、Oracle Names サーバー間の情報を最新の状態に保つために、ときどき実行されます。  システム問合せとしてリストされているアクティビティを変更できるアクションは特にありません。システム問合せを表示することによって、管理者はシステムの変更がいつ発生するかを理解でき、 <b>RESTART</b> を実行するかどうかを判断できます。



SHOW SYSTEM\_QUERIES

例： NAMESCTL> show system\_queries  
System query index number:1  
Query ID:49824  
Query next issued in:2 hours 55 min 3.84 seconds  
Query state:2  
Name:""  
Desired data type:ns.smd

SHOW TRACE\_FILE\_NAME

目的： 現在の Oracle Names サーバーの、トレース・ファイル名とパスを表示します。

前提条件： なし

パスワード： 不要

構文： オペレーティング・システムからは、次のように入力します。  
NAMESCTL show trace\_file\_name  
NAMESCTL ユーティリティからは、次のように入力します。  
NAMESCTL> show trace\_file\_name

引数： なし

使用上の注意： トレース・ファイル名は、初期値として names.ora ファイル内の [NAMES.TRACE\\_FILE](#) の値が設定されます。デフォルト値は、names.trc です。このファイルは有効なファイル名を持ち、Oracle Names サーバーがデータを書き込めるファイルである必要があります。

例： NAMESCTL> show trace\_file\_name  
Trace file name is currently  
/private/ora23/network/names.trc

SHOW TRACE\_LEVEL

目的： 現在の Oracle Names サーバーをトレースする、トレース・レベルを表示します。

前提条件： なし

パスワード： 不要



---

**SHOW TRACE\_LEVEL**

構文:	オペレーティング・システムからは、次のように入力します。 NAMESCTL show trace_level NAMESCTL ユーティリティからは、次のように入力します。 NAMESCTL> show trace_level
引数:	なし
使用上の注意:	トレースは、現行の Oracle Names サーバーの処理で発生する予測不能な障害、または識別不能な障害を診断するのに役立ちます。トレースは、通常の Oracle Names サーバー処理で行われる一連のイベントをオペレーティング・システム・ファイルに書き込んで、管理者が検査できるようにします。  トレース出力は、OFF (なし)、USER (基本的情報) または ADMIN (最大量の情報) という3段階のレベルで指定します。
例:	NAMESCTL> show trace_level Trace level is currently 0

---

---

**SHOW VERSION**

目的:	Oracle Names サーバーのカレント・バージョンと名前を表示します。
前提条件:	なし
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 NAMESCTL show version [onames_server] [onames_server]* NAMESCTL ユーティリティからは、次のように入力します。 NAMESCTL> show version [onames_server] [onames_server]*
引数:	Oracle Names サーバー名を空白で区切って指定します。名前を指定しないと、現行のネーム・サーバーの設定が表示されます。
使用上の注意:	このコマンドで、サーバーの名前とバージョンを確認します。 NAMESCTL をサーバーに接続するたびに、このコマンドが使用可能になります。
例:	NAMESCTL> show version Currently managing Oracle Names Server "NameServer.com" Version banner is "Oracle Names for SunOS: Version 8.1.6.0.0"

---



SHUTDOWN	
目的:	1 つ以上の Oracle Names サーバーを停止します。
前提条件:	Oracle Names サーバーが起動している必要があります。
パスワード:	必要  パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文:	オペレーティング・システムからは、次のように入力します。 <code>NAMESCTL shutdown [onames_server] [onames_server]*</code> NAMESCTL ユーティリティからは、次のように入力します。 <code>NAMESCTL&gt; shutdown [onames_server] [onames_server]*</code>
引数:	Oracle Names サーバー名を空白で区切って指定します。引数を指定しないと、現行の Oracle Names サーバーのみが停止します。
使用上の注意:	SHUTDOWN は、現行の Oracle Names サーバーを停止して、メモリからプログラムを削除します。Oracle Names サーバーを停止する必要があるのは、アップグレードやマシン保守などの運用上の理由がある場合のみです。Oracle Names サーバーを停止および起動するときは、ネットワーク内の任意の場所から実行できる <b>RESTART</b> コマンドを使用するようお勧めします。SHUTDOWN と START を個々に実行する場合は、Oracle Names サーバー・マシン上で実行する必要があります。  このコマンドは <b>STOP</b> コマンドと同じです。
例:	<code>NAMESCTL&gt; shutdown</code> <code>Confirm [yes or no] yes</code> <code>Server shut down.</code>
START	
目的:	Oracle Names サービス・プログラムをロードして、システム・データとローカル管理リージョン・データのロードを開始します。
前提条件:	Oracle Names サーバーが停止している必要があります。
パスワード:	不要



---

**START**

構文: オペレーティング・システムからは、次のように入力します。

```
NAMESCTL start
```

NAMESCTL ユーティリティからは、次のように入力します。

```
NAMESCTL> start
```

引数: なし

使用上の注意: START は、Oracle Names サーバーを初めてメモリーにロードするときに使用するコマンドです。Oracle Names サーバーは起動時に、操作パラメータ設定のための構成ファイルを読み込み、管理リージョンのデータをすべてロードします。

Oracle Names サーバー起動時のセキュリティは、Oracle Names がインストールされているオペレーティング・システムによって保証されます。Oracle Names サーバーはローカル・セッションから起動する必要があるため、ネットワーク・セキュリティが損なわれることはありません。

このコマンドは **STARTUP** コマンドと同じです。

---



START

```
例: NAMESCTL> start
Starting "/private/dsteiner/sales/bin/names"...server
successfully started

Currently managing name server "namesrv1.us.oracle.com"
Version banner is "Oracle Names for Solaris: Version 8.1.6.0.0"

Server name: namesrv1.us.oracle.com
Server has been running for: 0.16 seconds
Request processing enabled: yes
Request forwarding enabled: yes
Requests received: 0
Requests forwarded: 0
Foreign data items cached: 0
Region data next checked for reload in: not set
Region data reload check failures: 0
Cache next checkpointed in: not set
Cache checkpoint interval: not set
Cache checkpoint file name:
/private/dsteiner/sales/network/names/ckpcch.ora
Statistic counters next reset in: not set
Statistic counter reset interval: not set
Statistic counters next logged in: not set
Statistic counter logging interval: not set
Trace level: 0
Trace file name:
/private/dsteiner/sales/network/trace/names_10841.trc
Log file name:
/private/dsteiner/sales/network/log/names.log
System parameter file name:
/private/dsteiner/sales/network/admin/names.ora
Command-line parameter file name: ""
Administrative region name: ""
Administrative region description: ""
ApplTable Index: 0
Contact: ""
Operational Status: 0
Save Config on Stop: yes
```

START\_CLIENT\_CACHE

目的: クライアント・キャッシュ・デーモン・プロセスを起動します。



---

**START\_CLIENT\_CACHE**

前提条件:	クライアント・キャッシュ・デーモン・プロセスが停止している必要があります。 クライアント・キャッシュ・デーモン・プロセスを実行する前に、Oracle Names サーバー・リストが必要です。
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 <code>NAMECTL start_client_cache</code> NAMECTL ユーティリティからは、次のように入力します。 <code>NAMECTL&gt; start_client_cache</code>
引数:	なし
使用上の注意:	起動すると、クライアント・キャッシュ・デーモン・プロセスは、Oracle Names サーバーから受信したすべての情報を格納し、検索を高速に行うことができますようにします。
例:	<code>NAMECTL&gt; start_client_cache</code>

---

---

**STARTUP**

目的:	Oracle Names サービス・プログラムをロードして、システム・データとローカル管理リージョン・データのロードを開始します。
前提条件:	Oracle Names サーバーが停止している必要があります。
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 <code>NAMECTL startup</code> NAMECTL ユーティリティからは、次のように入力します。 <code>NAMECTL&gt; startup</code>
引数:	なし
使用上の注意:	このコマンドは <b>START</b> コマンドと同じです。
例:	START の例を参照してください。

---



STATUS	
目的:	1 つ以上の Oracle Names サーバーの統計と内部設定の多くを表示します。
前提条件:	Oracle Names サーバーが起動している必要があります。
パスワード:	不要
構文:	オペレーティング・システムからは、次のように入力します。 NAMESCTL status [onames_server] [onames_server]* NAMESCTL ユーティリティからは、次のように入力します。 NAMESCTL> status [onames_server] [onames_server]*
引数:	Oracle Names サーバー名を空白で区切って指定します。引数を指定しないと、現行の Oracle Names サーバーの状態のみが表示されます。
使用上の注意:	STATUS は、Oracle Names サーバー稼働中のアクティビティ、および一時点における Oracle Names サーバーの状態を表示します。
例:	NAMESCTL> status Version banner is "Oracle Names for SunOS: 8.1.6.0.0" Server name: NSERVER.com Server has been running for: 1 day 20 hours .....
STOP	
目的:	1 つ以上の Oracle Names サーバーを停止します。
前提条件:	Oracle Names サーバーが起動している必要があります。
パスワード:	必要 パスワードが設定されている場合、このコマンドの前に <b>SET PASSWORD</b> コマンドを実行する必要があります。
構文:	オペレーティング・システムからは、次のように入力します。 NAMESCTL stop [onames_server] [onames_server]* NAMESCTL ユーティリティからは、次のように入力します。 NAMESCTL> stop [onames_server] [onames_server]*
引数:	Oracle Names サーバー名を空白で区切って指定します。引数を指定しないと、現行の Oracle Names サーバーのみが停止します。



---

## STOP

使用上の注意: STOP は、現行の Oracle Names サーバーを停止して、メモリーからプログラムを削除します。Oracle Names サーバーを停止する必要があるのは、アップグレードやマシン保守などの運用上の理由がある場合のみです。Oracle Names サーバーを停止および起動するときは、ネットワーク内の任意の場所から実行できる **RESTART** コマンドを使用するようお勧めします。**STOP** と **START** を個々に実行する場合は、Oracle Names サーバー・マシン上で実行する必要があります。

このコマンドは **SHUTDOWN** コマンドと同じです。

例: 

```
NAMESCTL> stop
Confirm [yes or no]: yes
Server shut down
```

---

---

## TIMED\_QUERY

目的: Oracle Names サーバーのキャッシュにあるすべてのデータを表示します。

パスワード: 不要

構文: オペレーティング・システムからは、次のように入力します。

```
NAMESCTL timed_query
```

NAMESCTL ユーティリティからは、次のように入力します。

```
NAMESCTL> timed_query [time]
```

引数: 秒数

使用上の注意: **TIMED\_QUERY** は、引数 *time* で表される時刻以降のデータをすべて返します。*time* 引数で、所定の時刻以降に登録されたすべてのオブジェクトが戻されます。*time* 引数を使用するには、起動後に登録されたすべての情報を最初の **TIMED\_QUERY** でダンプします。このダンプが終了すると、最新ダンプ情報の終了時点を示す最終タイムスタンプが表示されます。その時点以降に記録されたすべてのデータを表示するには、*time* 引数で最終タイムスタンプ番号を指定します。

---

---

## UNREGISTER

目的: ネットワーク・オブジェクトを Oracle Names サーバーから削除します。



---

UNREGISTER

前提条件:	なし
パスワード:	不要
構文:	<p>オペレーティング・システムからは、次のように入力します。</p> <pre>NAMESCTL unregister object_name [-d [(description=] (address=(protocol_address_information)) [])] [-h hostname] [-l listener_name]</pre> <p>NAMESCTL ユーティリティからは、次のように入力します。</p> <pre>NAMESCTL&gt; unregister object_name [-d [(description=] (address=(protocol_address_information)) [])] [-l listener_name]</pre> <p><b>関連項目:</b> プロトコル・アドレスとパラメータの詳細は、<a href="#">付録 B</a> を参照してください。</p>
引数:	オブジェクト名 (必須) および登録されているアドレスまたはリスナー名、ホスト名
使用上の注意:	登録されているサービスを手動で登録解除するためのコマンドです。そのオブジェクトの定義が、リージョン内の Oracle Names サーバーから削除されます。オブジェクトが、アドレスまたはリスナー名、ホスト名で登録されている場合は、アドレスまたはリスナー名、ホスト名をコマンドラインで指定しないと、オブジェクトを登録解除できません。
例:	<pre>NAMESCTL&gt; unregister parts -t oracle_database -d (description= (address= (protocol=tcp) (host=nineva) (port=1575)) (connect_data=(service_name=db3)))</pre>

---

VERSION

目的:	Oracle Names サーバーのカレント・バージョンと名前を表示します。
前提条件:	なし
パスワード:	不要
構文:	<p>オペレーティング・システムからは、次のように入力します。</p> <pre>NAMESCTL version</pre> <p>NAMESCTL ユーティリティからは、次のように入力します。</p> <pre>NAMESCTL&gt; version</pre>



---

**VERSION**

引数:	Oracle Names サーバー名を空白で区切って指定します。名前を指定しないと、現行のネーム・サーバーの設定が表示されます。
使用上の注意:	このコマンドで、サーバーの名前とバージョンを確認します。NAMECTL をサーバーに接続するたびに、このコマンドが使用可能になります。
例:	<pre>NAMECTL&gt; version Currently managing Oracle Names Server "NameServer.com" Version banner is "Oracle Names for SunOS: Version 8.1.6.0.0"</pre>

---

## Oracle Connection Manager 制御ユーティリティ (CMCTL)

Oracle Connection Manager 制御ユーティリティ (CMCTL) は、Oracle Connection Manager を起動し、制御するためにオペレーティング・システム・プロンプトから実行するツールです。CMCTL の一般構文は、次のとおりです。

`CMCTL command [process_type]`

`process_type` はコマンドを実行するプロセスの名前です。これらを次に示します。

- `cman` - [CMGW \(Oracle Connection Manager ゲートウェイ・プロセス\)](#) および [CMADMIN \(Oracle Connection Manager 管理プロセス\)](#) の両方 (オラクル社では、`cman` をお勧めします)
- `cm` - CMGW プロセス

---

**注意:** `cm` は、リソースを予約するためのみに使用してください。CMGW プロセスは、CMADMIN プロセスなしで Oracle Connection Manager の基本機能をすべて実行します。

CMCTL では、CMADMIN プロセス用に `adm` プロセス・タイプもサポートされます。このプロセス・タイプは使用しないでください。

---

たとえば、ゲートウェイ・プロセスと管理プロセスの両方を起動する場合は、次のように入力します。

オペレーティング・システムからは、次のように入力します。

```
CMCTL start cman
```

CMCTL プログラムからは、次のように入力します。

```
CMCTL> start cman
```



CMCTL ユーティリティには、次のタイプのコマンドがあります。

- 操作コマンド (START、STOP、SHUTDOWN など)
- 修飾子コマンド (SET コマンドなど)
- 情報コマンド (STATUS、STATS、SHOW コマンドなど)
- コマンド・ユーティリティ操作コマンド (EXIT、QUIT、HELP など)

## SET および SHOW コマンド

SET コマンドを使用して、Oracle Connection Manager のパラメータ値や、CMCTL 制御ユーティリティ・セッション中の CMCTL 環境を変更できます。これらの設定は、現行の CMCTL セッションに対してのみ有効です。パラメータ設定を cman.ora ファイルに保存することはできません。

SHOW コマンドは、現行の構成の設定値を表示するために使用します。

## Shutdown コマンド

Oracle Connection Manager をシャットダウンするときは、**STOP** や **STOPNOW** コマンドよりも **SHUTDOWN** コマンドを使用することをお勧めします。SHUTDOWN には、STOP および STOPNOW のすべての機能が含まれています。

## CMCTL コマンド

次のコマンドは、CMCTL で使用できます。

ACCEPT_CONNECTIONS	
目的:	Oracle Connection Manager への新しい接続を使用可能または使用禁止にします。
前提条件:	Oracle Connection Manager が動作している必要があります。
構文:	オペレーティング・システムからは、次のように入力します。  CMCTL accept_connections [argument]  CMCTL ユーティリティからは、次のように入力します。  CMCTL> accept_connections [argument]
引数:	ON (デフォルト) - Oracle Connection Manager への新しい接続を可能にします。  OFF - Oracle Connection Manager への新しい接続を拒否します。
使用上の注意:	OFF に設定されても、既存の接続には影響しません。







HELP	
目的:	使用可能な CMCTL コマンドの一覧を表示します。HELP コマンドのうちの 1 つに応答して、CMCTL はコマンドの使用方法についてのヘルプを表示します。
前提条件:	なし
構文:	オペレーティング・システムからは、次のように入力します。  CMCTL help  CMCTL help [command]  CMCTL ユーティリティからは、次のように入力します。  CMCTL> help CMCTL> help [command]
引数:	HELP コマンド。次の出力例にコマンドが示されています。
例:	CMCTL> help The following operations are available An asterisk (*) denotes a modifier or extended command: accept_connections close_relay exit quit set* show* shutdown start stats status stop stopnow version

QUIT	
目的:	CMCTL を終了し、オペレーティング・システム・プロンプトに戻ります。
前提条件:	なし
構文:	CMCTL ユーティリティからは、次のように入力します。  CMCTL> quit
使用上の注意:	このコマンドは EXIT コマンドと同じです。



---

## SET

目的:	現行の CMCTL セッションに対して設定できる使用可能な構成コマンドのリストを表示します。
前提条件:	なし
構文:	<p>オペレーティング・システムからは、次のように入力します。</p> <pre>CMCTL set CMCTL set [command]</pre> <p>CMCTL ユーティリティからは、次のように入力します。</p> <pre>CMCTL set CMCTL&gt; set [command]</pre>
引数:	SET コマンド。次の出力例にコマンドが示されています。
例:	<pre>CMCTL&gt; set The following operations are available An asterisk (*) denotes a modifier or extended command: authentication_level displaymode log_level relay_statistics</pre>

---

---

## SET AUTHENTICATION\_LEVEL

目的:	Oracle Connection Manager のセキュリティのレベルを設定します。
前提条件:	Oracle Connection Manager が動作している必要があります。
構文:	<p>オペレーティング・システムからは、次のように入力します。</p> <pre>CMCTL set authentication_level level</pre> <p>CMCTL ユーティリティからは、次のように入力します。</p> <pre>CMCTL&gt; set authentication_level level</pre>
引数:	<p>認証レベル:</p> <p>0 (デフォルト) - クライアント接続に対する認証は不要です。</p> <p>1 - クライアント認証を行うための Secure Network Service (SNS) を実行していない接続を拒否します。</p> <p><b>注意:</b> SNS は Oracle Advanced Security に組み込まれています。</p>



## SET AUTHENTICATION\_LEVEL

```
例:          CMCTL> authentication_level 0
          Profile of the CMAN
          -----
          AUTHENTICATION LEVEL    = 0
```

## SET DISPLAYMODE

目的: CMCTL [START](#)、CMCTL [STATS](#)、CMCTL [STATUS](#)、CMCTL [STOP](#)、CMCTL [VERSION](#) コマンドの出力を変更します。

前提条件: なし

構文: オペレーティング・システムからは、次のように入力します。

```
CMCTL set displaymode [argument]
```

CMCTL ユーティリティからは、次のように入力します。

```
CMCTL> set displaymode [argument]
```

引数: COMPAT (デフォルト) - 旧バージョンの Oracle Connection Manager と互換性のある出力を表示します。

VERBOSE (推奨値) - 書式化された詳細な出力を表示します。

```
例:          CMCTL> set displaymode compat
            Current display mode is COMPATible
```

## SET LOG\_LEVEL

目的: Oracle Connection Manager のログ・レベルを設定します。

前提条件: Oracle Connection Manager が動作している必要があります。

構文:

```
CMCTL set log_level [level]
```

CMCTL ユーティリティからは、次のように入力します。

```
CMCTL> set log level [level]
```



---

**SET LOG\_LEVEL**

- 引数: 0 ～ 4:
- レベル 0 (デフォルト) - ロギングなし
  - レベル 1 - 基本的なレポートの作成
  - レベル 2 - RULE\_LIST 一致検索レポートの作成
  - レベル 3 - リレー・ブロック・レポートの作成
  - レベル 4 - リレー I/O カウント・レポートの作成

例: CMCTL> set log\_level 0

Profile of the CMAN

-----  
LOG\_LEVEL = 0

---



---

**SET RELAY\_STATISTICS**

目的: Oracle Connection Manager の接続の I/O に関する統計収集機能をオンまたはオフにします。

リレーが終了し、このコマンドが ON に設定されている場合、UNIX では cman\_pid.log ファイルの LOG\_RECORD に、Windows NT では cmanpid.log ファイルの LOG\_RECORD に次の統計が記録されます。

- IN バイト数
- OUT バイト数
- IN パケット数
- OUT パケット数

前提条件: Oracle Connection Manager が動作している必要があります。

構文: オペレーティング・システムからは、次のように入力します。

CMCTL set relay\_statistics [argument]

CMCTL ユーティリティからは、次のように入力します。

CMCTL> set relay\_statistics [argument]

引数: ON  
OFF (デフォルト)



SET RELAY\_STATISTICS

例： CMCTL> set relay\_statistics off

Profile of the CMAN

-----

RELAY\_STATISTICS = no

SHOW

目的： 現行の CMCTL セッションについて表示できる、使用可能なコマンドのリストを表示します。CMCTL は SHOW コマンドのうちの 1 つに回答して、そのパラメータの現行の設定を表示します。

前提条件： なし

構文： オペレーティング・システムからは、次のように入力します。

CMCTL show

CMCTL show [command]

CMCTL ユーティリティからは、次のように入力します。

CMCTL> show

CMCTL> show [command]

引数： SHOW コマンド。次の出力例にコマンドが示されています。

例： CMCTL> show

The following operations are available

An asterisk (\*) denotes a modifier or extended command:

address

ALL

displaymode

profile

relay

rules

SHOW ADDRESS

目的： Oracle Connection Manager のリスニング用プロトコル・アドレスのリストを表示します。

前提条件： なし



---

**SHOW ADDRESS**

構文: オペレーティング・システムからは、次のように入力します。  
CMCTL show address  
CMCTL ユーティリティからは、次のように入力します。  
CMCTL> show address

引数: なし

使用上の注意: アドレスは、cman.ora ファイルの **CMAN** パラメータで設定されます。

例: CMCTL> show address

```
Address List
-----
(ADDRESS=(PROTOCOL=tcp) (HOST=dlsun1594) (PORT=1630) (QUEUESIZE=32)
)
(ADDRESS=(PROTOCOL=tcp) (HOST=dlsun1594) (PORT=1631) (QUEUESIZE=32)
)
```

---

---

**SHOW ALL**

目的: **SHOW ADDRESS**、**SHOW PROFILE** および **SHOW RULES** コマンドの出力を表示します。

前提条件: なし

構文: オペレーティング・システムからは、次のように入力します。  
CMCTL show all  
CMCTL ユーティリティからは、次のように入力します。  
CMCTL> show all

引数: なし



## SHOW ALL

例 :

```

CMCTL> show all
Address List
-----
(AADDRESS=(PROTOCOL=tcp) (HOST=dlsun1594) (PORT=1630) (QUEUE SIZE=32)
)
(AADDRESS=(PROTOCOL=tcp) (HOST=dlsun1594) (PORT=1631) (QUEUE SIZE=32)
)

Profile of the CMAN
-----
MAXIMUM_RELAYS           = 2048
RELAY_STATISTICS         = yes
AUTHENTICATION_LEVEL     = 0
LOG_LEVEL                = 4
ANSWER_TIMEOUT           = 0
MAXIMUM_CONNECT_DATA     = 1024
USE_ASYNC_CALL           = yes
TRACING                  = yes
TRACE_DIRECTORY          = default
MAX_FREELIST_BUFFERS     = 0
REMOTE_ADMIN             = no

Rule List
-----
(rule=(src=144.25.185.60) (dst=x) (srv=x) (act=accept))
(rule=(src=sguan-pc) (dst=x) (srv=x) (act=accept))

```

---

## SHOW DISPLAYMODE

目的 :

CMCTL [START](#)、CMCTL [STATS](#)、CMCTL [STATUS](#)、CMCTL [STOP](#)、CMCTL [VERSION](#) コマンド用に使用されている現行の表示モードを示します。

COMPAT (デフォルト) - 旧バージョンの Oracle Connection Manager と互換性のある出力を表示します。

VERBOSE - 書式化された詳細な出力を表示します。

構文 :

オペレーティング・システムからは、次のように入力します。

```
CMCTL show displaymode
```

CMCTL ユーティリティからは、次のように入力します。

```
CMCTL> show displaymode
```

例 :

```
CMCTL> show displaymode
Current display mode is VERBOSE
```

---



---

## SHOW PROFILE

目的: Oracle Connection Managerの現行のパラメータ設定をリストします。  
表示される情報は、`cman.ora` ファイルの **CMAN\_PROFILE** パラメータ、および CMCTL **SET** コマンドによって行われたすべての変更に基づくものです。

構文: オペレーティング・システムからは、次のように入力します。

```
CMCTL show profile
```

CMCTL ユーティリティからは、次のように入力します。

```
CMCTL> show profile
```

例: 

```
CMCTL> show profile
Profile of the CMAN
```

```
-----
MAXIMUM_RELAYS           = 2048
RELAY_STATISTICS         = yes
AUTHENTICATION_LEVEL     = 0
LOG_LEVEL                = 4
SHOW_TNS_INFO            = yes
ANSWER_TIMEOUT           = 0
MAXIMUM_CONNECT_DATA     = 1024
USE_ASYNC_CALL           = yes
TRACING                  = yes
TRACE_DIRECTORY          = default
MAX_FREELIST_BUFFERS     = 0
REMOTE_ADMIN             = no
```

---

---

## SHOW RELAY

目的: Oracle Connection Manager の指定されたりレー（接続）またはすべてのアクティブなリレーの現在のステータスを表示します。

前提条件: Oracle Connection Manager が動作している必要があります。

構文: オペレーティング・システムからは、次のように入力します。

```
CMCTL show relay {argument}
```

CMCTL ユーティリティからは、次のように入力します。

```
CMCTL> show relay {argument}
```

引数: *relay\_num* - 特定のりレーについてのステータス情報を表示します。

**ACTIVE** - アクティブなリレーのリストを表示します。

---



SHOW RELAY

使用上の注意：                      アクティブなリレーを表示する場合は、リレー番号しか表示されません。特定のリレー番号を表示する場合は、次の情報が表示されます。

- リレー番号
- ソース・アドレス（クライアント側エンドポイント）
- 接続先アドレス（サーバー側エンドポイント）
- IN バイト数
- IN パケット数
- IN DCD プローブ数
- OUT バイト数
- OUT パケット数
- OUT DCD プローブ数

```
例：
CMCTL> show relay active

Active Relays
-----
0000 0001
CMCTL> show relay 0

Relay Information
-----
Relay number           0
Start-up time          22-JUL-1999 19:47:17
Src
  (ADDRESS= (PROTOCOL=tcp) (HOST=144.25.185.60) (PORT=35279))
Dest
  (ADDRESS= (PROTOCOL=tcp) (HOST=144.25.185.60) (PORT=1521))
Number of IN bytes      438
Number of IN packets    7
Number of IN DCD probes 0
Number of OUT bytes     364
Number of OUT packets   7
Number of OUT DCD probes 0
```

SHOW RULES

目的：                                  Oracle Connection Manager によって使用されている現行の Net8 アクセス・ルールをリスト表示します。



---

## SHOW RULES

前提条件:	Oracle Connection Manager が動作している必要があります。
構文:	<p>オペレーティング・システムからは、次のように入力します。</p> <pre>CMCTL show rules</pre> <p>CMCTL ユーティリティからは、次のように入力します。</p> <pre>CMCTL&gt; show rules</pre>
引数:	なし
使用上の注意:	ルールは、cman.ora ファイル内の <b>CMAN_RULES</b> パラメータによって設定されます。
例:	<pre>CMCTL&gt; show rules</pre> <p>Rule List</p> <p>-----</p> <pre>(rule=(src=144.25.185.60) (dst=x) (srv=x) (act=accept)) (rule=(src=sguan-pc) (dst=x) (srv=x) (act=accept))</pre>

---

## SHUTDOWN

目的:	Oracle Connection Manager プロセスをシャットダウンします。
前提条件:	なし
構文:	<p>オペレーティング・システムからは、次のように入力します。</p> <pre>CMCTL shutdown [argument] [cman]</pre> <p>CMCTL ユーティリティからは、次のように入力します。</p> <pre>CMCTL&gt; shutdown [argument] [cman]</pre>
引数:	<p>NORMAL (デフォルト) - 新しい接続は許可されず、すべての既存の接続が終了した後に Oracle Connection Manager が終了します。</p> <p>ABORT - オープンしている接続をすべて終了させ、Oracle Connection Manager をただちにシャットダウンします。</p> <p>cman - CMGW および CMADMIN の両方のプロセスを終了します。NORMAL の場合も ABORT の場合も、この引数が使用されます。したがって、これは明示的に指定する必要はありません。</p>



---

**SHUTDOWN**

例 :

```
CMCTL> shutdown

The command completed successfully

CMCTL> shutdown abort

The command completed successfully
```

---

---

**START**

目的 : Oracle Connection Manager を起動します。

前提条件 : 同じリスニング・アドレスを使用している Oracle Connection Manager が動作していない。

構文 : オペレーティング・システムからは、次のように入力します。

```
CMCTL start [process_type]
```

CMCTL ユーティリティからは、次のように入力します。

```
CMCTL> start [process_type]
```

引数 : cman (デフォルト) - CMGW および CMADMIN の両方のプロセスを起動します。

cm - CMGW プロセスを起動します。

使用上の注意 : [SET DISPLAYMODE](#) コマンドは、出力の書式や詳細レベルを変更します。

---



---

**START**

例:

COMPAT モード:

```
CMCTL> start cman
ADMIN Status:
  (STATUS= (VERSION=8.1.6.0.0) (STARTED=22-JUL-1999
19:34:15) (STATE=RUNNING))
CMAN Status:
  (STATUS= (VERSION=8.1.6.0.0) (STARTED=22-JUL-1999
19:34:17) (STATE=running))
```

VERBOSE モード:

```
CMCTL> start cman
Starting /vobs/oracle/bin/cmadmin: please wait...
STATUS of the cman_admin
-----
ADMIN Version           8.1.6.0.0
Start-up time           22-JUL-1999 19:40:00
Current state           RUNNING

Starting /vobs/oracle/bin/cmfw: please wait...
STATUS of the cman
-----
CMAN Version            8.1.6.0.0
Start-up time           22-JUL-1999 19:40:02
Current state           running
```

---

---

**STATS**

目的: Oracle Connection Manager の統計情報を表示します。

前提条件: Oracle Connection Manager が動作している必要があります。

構文: オペレーティング・システムからは、次のように入力します。

```
CMCTL stats [process_type]
```

CMCTL ユーティリティからは、次のように入力します。

```
CMCTL> stats [process_type]
```

引数:

cman (デフォルト) - CMGW および CMADMIN プロセスの統計を表示します。

cm - CMGW プロセスの統計を表示します。

---



STATS

- 使用上の注意 :
- 次の統計が表示されます。
- TOTAL\_RELAYS (COMPAT) | 処理された接続の合計数 (VERBOSE)  
Oracle Connection Manager が、起動後に確立した接続の総数
  - ACTIVE\_RELAYS (COMPAT) | 現在アクティブなリレーの数 (VERBOSE)  
現在アクティブな接続の数
  - MOST\_RELAYS (COMPAT) | ピーク時のアクティブなリレー数 (VERBOSE)  
Oracle Connection Manager の起動後に、同時に確立された接続の最大数
  - OUT\_OF\_RELAY (COMPAT) | max\_relays を超えたために拒否された接続の合計数 (VERBOSE)  
Oracle Connection Manager の起動後に、リレーの範囲外により拒否された接続要求の総数
  - TOTAL\_REFUSED (COMPAT) | 拒否された接続の合計数 (VERBOSE)  
Oracle Connection Manager の起動後に、拒否された接続要求の総数

使用上の注意 :

SET DISPLAYMODE コマンドは、出力の書式や詳細レベルを変更します。

例 :

COMPAT モード :

CMCTL> stats  
CMAN Status:  
(STATISTICS=(TOTAL\_RELAYS=0) (ACTIVE\_RELAYS=0) (MOST\_RELAYS=0) (OUT\_OF\_RELAY=0) (TOTAL\_REFUSED=0))

VERBOSE モード :

CMCTL> stats  
STATISTICS of CMAN  
-----  
Total number of connections handled 0  
Number of currently active relays 0  
Peak active relays 0  
Total refusals due to max\_relays exceeded 0  
Total number of connections refused 0



---

**STATUS**

目的: バージョン、開始時刻、現在の統計などの基本的な統計情報を表示します。

前提条件: なし

構文: オペレーティング・システムからは、次のように入力します。

```
CMCTL> status [process_type]
```

CMCTL ユーティリティからは、次のように入力します。

```
CMCTL> status [process_type]
```

引数: cman (デフォルト) - CMGW と CMADMIN の両方のプロセスのステータス情報を表示します。

cm - CMGW プロセスのステータス情報を表示します。

使用上の注意: [SET DISPLAYMODE](#) コマンドは、出力の書式や詳細レベルを変更します。

例: COMPAT モード:

```
CMCTL> status
```

```
CMAN Status:
```

```
(STATUS=(VERSION=8.1.6.0.0) (STARTED=22-JUL-1999  
19:34:17) (STATE=running))
```

```
ADMIN Status:
```

```
(STATUS=(VERSION=8.1.6.0.0) (STARTED=22-JUL-1999  
19:34:15) (STATE=RUNNING))
```

VERBOSE モード:

```
CMCTL> status
```

```
STATUS of the cman
```

```
-----  
CMAN Version           8.1.6.0.0  
Start-up time          22-JUL-1999 19:40:02  
Current state          running
```

```
STATUS of the cman_admin
```

```
-----  
ADMIN Version          8.1.6.0.0  
Start-up time          22-JUL-1999 19:40:00  
Current state          RUNNING
```

---



STOP	
目的:	オープンしている接続がある場合は停止の確認を要求してから、Oracle Connection Manager をただちにシャットダウンします。
前提条件:	Oracle Connection Manager が動作している必要があります。
構文:	オペレーティング・システムからは、次のように入力します。 CMCTL stop [process_type] CMCTL ユーティリティからは、次のように入力します。 CMCTL> stop [process_type]
引数:	cman (デフォルト) - CMGW および CMADMIN の両方のプロセスを停止します。 cm - CMGW および CMADMIN の両方のプロセスを停止します。
使用上の注意:	このコマンドよりも <b>SHUTDOWN</b> コマンドを使用することをお勧めします。 接続がアクティブな状態のままで停止コマンドを発行すると、停止の確認を求められます。

STOPNOW	
目的:	オープンしている接続を警告なしですべて終了し、Oracle Connection Manager をただちに終了させます。
前提条件:	Oracle Connection Manager が動作している必要があります。
構文:	オペレーティング・システムからは、次のように入力します。 CMCTL stopnow CMCTL ユーティリティからは、次のように入力します。 CMCTL> stopnow
引数:	cman (デフォルト) - CMGW および CMADMIN の両方のプロセスを停止します。 cm - CMGW および CMADMIN の両方のプロセスを停止します。
使用上の注意:	このコマンドよりも、 <b>SHUTDOWN</b> コマンドを使用することをお勧めします。
例:	CMCTL> stopnow



---

**VERSION**

目的: CMCTL ユーティリティのカレント・バージョンと名前を表示します。

前提条件: なし

構文: オペレーティング・システムからは、次のように入力します。

CMCTL version

CMCTL ユーティリティからは、次のように入力します。

CMCTL> version

使用上の注意: **SET DISPLAYMODE** コマンドは、出力の書式を変更します。

例: COMPAT モード:

CMCTL> version

CMCTL Version 8.1.6.0.0

VERBOSE モード:

CMCTL> version

CMCTL Version 8.1.6.0.0

CMAN Version 8.1.6.0.0

---







---

## プロトコル・アドレス

ネットワーク・オブジェクトは、プロトコル・アドレスによって識別されます。接続が作成されると、要求の受信者（リスナー、Oracle Names サーバー、Oracle Connection Manager など）とクライアントは同じプロトコルを使用して構成されます。

クライアントは、このアドレスを使用して特定のネットワーク・オブジェクトがある場所に接続要求を送信します。一方、受信者はこのアドレス上でリスニングし、クライアントの情報が自分の持っているアドレス情報に一致するかどうかに基づいて接続を許可します。

この付録の内容は次のとおりです。

- [ADDRESS と ADDRESS\\_LIST](#)
- [プロトコルのパラメータ](#)
- [お薦めするポート番号](#)
- [ポート番号の制限](#)



# ADDRESS と ADDRESS\_LIST

プロトコル・アドレスは、ADDRESS および ADDRESS\_LIST の 2 つの要素で構成されます。

ADDRESS	
目的:	プロトコル・アドレスを定義します。このパラメータは、ADDRESS_LIST または DESCRIPTION の中に埋め込むことができます。DESCRIPTION は、通常、tnsnames.ora または listener.ora ファイルでのみ使用されます。
	<b>関連項目:</b> B-2 ページの「 <a href="#">プロトコルのパラメータ</a> 」に説明されているように、各プロトコルにはアドレスに必要な特定のパラメータがあります。
例:	(address= (protocol=tcp) (host=sales-pc) (port=1521))

ADDRESS_LIST	
目的:	共通の特性を共有するアドレスのリストを定義します。
例:	(address_list= (load_balance=on) (address= (protocol=tcp) (host=sales-pc) (port=1521)) (address= (protocol=tcp) (host=hr-pc) (port=1521))) (address_list= (address= (protocol=tcp) (host=finance-pc) (port=1521)))

## プロトコルのパラメータ

リスナー、Oracle Names サーバーおよび Oracle Connection Manager は、プロトコル・アドレスによって識別されます。次の表は、サポートしている Oracle プロトコルで使用されるパラメータについて説明しています。



プロトコル	パラメータ	説明
すべて	PROTOCOL	TNS ベースのアプリケーションが存在するネットワークのタイプを示す。
Bequeath	PROGRAM	Oracle8i の実行可能ファイルを識別します。
Bequeath	ARGV0	サービス名を識別します。
Bequeath	ARGS	接続のソース（ローカル・クライアント）を識別します。
IPC	KEY	サーバーの識別方法を示す。オラクル社では、サービス名の使用をお薦めしています。
Named Pipes	SERVER	使用している Oracle8i サーバー・コンピュータの名前を示す。
Named Pipes	PIPE	サーバーへの接続に使用するパイプ名を示す (Named Pipes でサーバーに指定したものと同一 PIPE キーワード)。これには任意の名前を付けることができます。
LU6.2	LU_NAME	Oracle8i サーバーを識別します。完全に修飾された名前にする必要があります。
LU6.2	LLU または LOCAL_LU	ローカルの LU 別名を識別します。このパラメータは、LLU_NAME と一緒に使用できません。
LU6.2	LLU_NAME または LOCAL_LU_NAME	ローカルの LU 名を示す。完全に修飾された名前にする必要があります。このパラメータは、LLU と一緒に使用できません。
LU6.2	MODE または MDN	LU6.2 ネットワークのログ・モード・エントリを識別します。この値は、通常 ORAPLU62 です。
LU6.2	PLU または PARTNER_LU_NAME	Oracle8i サーバーを識別します。完全に修飾された名前にする必要があります。このパラメータは、LLU と一緒に使用できません。
LU6.2	PLU_LA または PARTNER_LU_LOCAL_ALIAS	Oracle8i server のパートナ LU 別名を識別します。このパラメータは、PLU と一緒に使用できません。
LU6.2	TP_NAME または TPN	ホスト・マシンのトランザクション・プログラム名を識別します。このパラメータは、MVS ホストへの接続では必要ありません。
SPX	SERVICE	ネットワーク上にある TNS ベースのアプリケーションの名前を定義します。



プロトコル	パラメータ	説明
TCP/IP と SSL 付き TCP/IP	HOST	マシンのホスト名を識別します。
TCP/IP と SSL 付き TCP/IP	PORT	マシンの IP アドレスを識別します。

## お薦めするポート番号

オラクル社は、次のポート番号の使用をお薦めします。

ポート	説明
1521	クライアントからリスナーへの接続用のデフォルトのリスニング・ポート。将来のリリースでは、このポート番号が公式に登録されたポート番号である TCP/IP の 2483 と SSL 付き TCP/IP の 2484 に変更される可能性があります。
2481	クライアントから Oracle8i JServer オプションへの TCP/IP 接続でお薦めする、公式に登録されたリスニング・ポート
2482	クライアントから Oracle8i JServer への SSL 付き TCP/IP 接続でお薦めする、公式に登録されたリスニング・ポート
2484	クライアントからリスナーへの SSL 付き TCP/IP 接続でお薦めする、公式に登録されたリスニング・ポート
1575	クライアントから Oracle Names サーバーへの TCP/IP 接続または SSL 付き TCP/IP 接続用の、デフォルトの公式に登録されたリスニング・ポート
1630	クライアントから Oracle Connection Manager への接続用の、デフォルトの、公式に登録されたリスニング・ポート
1830	Oracle Connection Manager への管理コマンド用の、デフォルトの、公式に登録されたリスニング・ポート

## ポート番号の制限

オラクル社では、1 から 65535 までのポート番号を許可しています。ただし、1024 未満のポート番号は、多くのオペレーティング・システムで、特権的な処理で使用するために予約されています。

リスナーが 1024 未満のポート番号を使用するように構成されている場合、LSNRCTL ユーティリティを使用してリスナーを起動することができません。LSNRCTL は、それ自身がルートとして実行している場合でも、ルート（UNIX システム上で）としてリスナーを起動できません。ただし、常にルートとしてリスナーの実行可能ファイル（tnslsnr）を実行で



きます。このファイルを実行するには、オペレーティング・システム・プロンプトで、コマンド `tnslsnr listener_name` を入力します。リスナーの起動後は、LSNRCTL ユーティリティが（リスナーの停止も含め）リスナーを管理します。







---

## 構成パラメータ

この付録には、Net8 構成パラメータの全リストが記載されています。内容は次のとおりです。

- 構成ファイルの構文規則
- プロファイル・パラメータ (sqlnet.ora)
- ローカル・ネーミング・パラメータ (tnsnames.ora)
- リスナー・パラメータ (listener.ora)
- Oracle Names パラメータ (names.ora)
- Oracle Connection Manager パラメータ (cman.ora)
- プロトコル固有のパラメータ (protocol.ora)
- ディレクトリ・サーバー・アクセス・パラメータ (ldap.ora)
- リリース 8.1 での構成ファイルの変更点



## 構成ファイルの構文規則

Net8 ネットワークの構成ファイルは、キーワードと値が対となるパラメータの集まりです。キーワードと値の対をカッコで囲みます。

```
parameter=(keyword=value)
```

一部のキーワードは、他のキーワードと値の対をそのキーワードの値とします。

```
(keyword=  
  (keyword=value)  
  (keyword=value))
```

たとえば、ローカル名構成ファイル (tnsnames.ora) のアドレス部に、次の行を含めることができます。

```
(address=  
  (protocol=tcp)  
  (host=max)  
  (port=1521))
```

どのキーワードが、他のキーワードと値の対の親または所有者であるかをインデントで反映するように構成ファイルを設定します。インデントの使用は義務付けられていませんが、このフォーマットによって読みやすく理解しやすいファイルになります。

このようにファイルをインデントしない場合でも、行を折り返すときは空白を1つ以上入れないと、折り返し行が新しいパラメータとして解釈されてしまいます。次のレイアウトは受け入れられます。

```
(address=(protocol=tcp)  
  (host=max) (port=1521))
```

次のレイアウトは受け入れられません。

```
(address=(protocol=tcp)  
(host=max) (port=1521))
```

## 構成ファイルの構文規則の詳細

次の規則が構成ファイルの構文に適用されます。

- キーワードと値の対を1つ以上含むパラメータで始まる構成ファイルのキーワードは、一番左の列にしてください。キーワードの前に1つ以上の空白がある場合は、前の行の継続行として解釈されます。
- すべての文字は、ネットワーク・キャラクタ・セットに属している必要があります（次の項を参照してください）。



- キーワードでは大 / 小文字は区別されません。値は、オペレーティング・システムおよびプロトコルによって、大 / 小文字を区別する場合と区別しない場合があります。
- キーワードと値の対の中で、"=" の前後の空白はオプションです。
- キーワードの階層がある場合、キーワードの後ろには必ず他のキーワードが続きます。階層のどのレベルでも、キーワードを任意の順序で入力できます。たとえば、次の構文はどちらも有効です。

```
(address=
  (protocol=tcp)
  (host=martha)
  (port=1521))
```

```
(address=
  (protocol=tcp)
  (port=1521)
  (host=martha))
```

- キーワードに空白を含めることはできません。値に空白を含めるときは、二重引用符 (") または単一引用符 (') で囲む必要があります。
- 接続記述子の最大長は 4KB です。
- コメントを入力するときは、行の先頭でシャープ記号 (#) を使用します。シャープ記号の後から行の終わりまでが、コメントとみなされます。
- キーワードと値の対で等号の前または後に 1 つの語または連結語を入力する場合は、カッコは不要です。

## ネットワーク・キャラクタ・セット

キーワードと値の対で使えるネットワーク・キャラクタ・セットには、次の文字があります。接続記述子は 1 バイト文字にしてください。

A-Z、a-z

0-9

()<>/\

,.:;'"=-\_

\$+\*#&!%?@

このキャラクタ・セットの中では、次の記号が予約済です。

()=\ "'#



キーワードまたは値を引用符で囲まない場合は、予約記号をキーワードまたは値の一部としてではなくデリミタとして使用します。単一引用符または二重引用符を使用して、予約記号を含む値を囲むことができます。引用符で囲んだ値の中に引用符を含めるには、別のタイプの引用符を使用します。バックスラッシュ (\) をエスケープ文字として使用します。

予約記号の使用例として、アドレス内の数値 DECnet オブジェクトがあります。OBJECT を ABC や #123 などの名前にすることができます。これらは次の形式で入力します。

(OBJECT=ABC)

または

(OBJECT=#123)

符号 "#" は予約されている記号なので、文字の前にバックスラッシュを付ける必要があります。

次の文字は、接続記述子の中では使用できますが、キーワードまたは値の中では使用できません。

<space> <tab> <CR> <newline>

## キャラクタ・セット

リスナー名、ネット・サービス名および Oracle Names サーバーは、次のキャラクタ・セットに制限されています。

[a...z] [A...Z] [0...9] \_

最初の文字は、アルファベットの文字にしてください。一般的に、最大 64 文字が受け付けられます。データベース・サービス名はデータベース管理者が定義したグローバル・データベース名と一致する必要があります。グローバル・データベース名は、データベース名（最初は 8 文字に制限されています）とデータベース・ドメインで構成されています。サービス名とグローバル・データベース名は、大 / 小文字を区別しません。

## プロファイル・パラメータ (sqlnet.ora)

プロファイルでは、次のパラメータを使用できます。

sqlnet.ora ファイルは、UNIX プラットフォームの \$ORACLE\_HOME/network/admin、Windows プラットフォームの ORACLE\_HOME¥network¥admin にあるか、あるいは TNS\_ADMIN 環境変数またはレジストリ値によって指定されたディレクトリにあります。

---

### BEQUEATH\_DETACH

目的: UNIX システムでシグナル処理のオンとオフを切り換えます。

デフォルト: NO で、シグナル処理をオンにします。



---

**BEQUEATH\_DETACH**

値 :

- YES - シグナル処理をオフにします
- NO - シグナル処理をオンにします

例 : bequeath\_detach=yes

---

---

**DAEMON.TRACE\_DIRECTORY**

目的 : Oracle Enterprise Manager のデーモン・トレース・ファイルのディレクトリを指定します。

デフォルト : UNIX プラットフォームでは \$ORACLE\_HOME/network/trace、Windows プラットフォームでは ORACLE\_HOME\network\trace

例 : daemon.trace\_directory=/oracle/traces

---

---

**DAEMON.TRACE\_LEVEL**

目的 : Oracle Enterprise Manager デーモンのトレースをオフにするか、特定のレベルでオンにします。

デフォルト : OFF

値 :

- OFF - トレース出力しません
- USER - ユーザー・トレース情報
- ADMIN - 管理トレース情報
- SUPPORT - オラクル社カスタマ・サポート・センター用のトレース情報

例 : daemon.trace\_level=user

---

---

**DAEMON.TRACE\_MASK**

目的 : Oracle Enterprise Manager デーモンのトレース・エントリのみをトレース・ファイルに記録することを指定します。

例 : daemon.trace\_mask= (106)

---



DISABLE_OOB	
目的:	オフにすると、基底プロトコルによって指定された緊急データを使用して "ブレーク" メッセージを Net8 で送受信できます。  オンにすると、基底プロトコルによって指定された緊急データを使用している "ブレーク" メッセージを Net8 で送受信できなくなります。これは、1 度使用可能にすると、このクライアントが使用するすべてのプロトコルに適用されます。  <b>関連項目:</b> 使用しているプロトコルが緊急データ依頼をサポートするようにするかどうかを知るには、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。TCP/IP は、この機能をサポートするプロトコルの一例です。
デフォルト:	OFF
例:	disable_oob=ON

LOG_DIRECTORY_CLIENT	
目的:	クライアント・ログ・ファイルの作成先となるディレクトリを指定します。
デフォルト:	実行可能ファイルの起動元のカレント・ディレクトリ。
例:	log_directory_client=/oracle/network/log

LOG_DIRECTORY_SERVER	
目的:	サーバー・ログ・ファイルの作成先となるディレクトリを指定します。
デフォルト:	実行可能ファイルの起動元のカレント・ディレクトリ。
例:	log_directory_server=/oracle/network/log



---

**LOG\_FILE\_CLIENT**

目的: クライアントのログ・ファイルの名前を指定します。

デフォルト: sqlnet.log

例: log\_file\_client=client

---

---

**LOG\_FILE\_SERVER**

目的: サーバーのログ・ファイルの名前を指定します。

デフォルト: sqlnet.log

例: log\_file\_server=svr.log

---

---

**NAMES.DCE.PREFIX**

目的: 名前参照の検索に使用する DCE セル名（接頭辞）を指定します。

デフォルト: /./:subsys/oracle/names

例: names.dce.prefix=/./:subsys/oracle/names

---

---

**NAMES.DEFAULT\_DOMAIN**

目的: クライアントが最も頻繁に名前解決要求を検索するドメインを設定します。このパラメータを設定すると、デフォルトのドメイン名が未修飾のネット・サービス名またはサービス名に自動的に付けられます。

たとえば、デフォルトのドメインを us.acme.com に設定した場合、接続文字列 CONNECT scott/tiger@sales は sales.us.acme.com であるとして検索されます。接続文字列にドメイン拡張子（CONNECT scott/tiger@sales.acme.com など）が含まれている場合は、ドメインは追加されません。

デフォルト: NULL

例: names.default\_domain=com

---



NAMES.DIRECTORY_PATH	
目的:	クライアントの名前解決参照に使用するネーミング・メソッドの順番を指定します。
デフォルト:	TNSNAMES, ONAMES, HOSTNAME
値:	<ul style="list-style-type: none"><li>■ TNSNAMES</li><li>■ ONAMES</li><li>■ HOSTNAME</li><li>■ DCE</li><li>■ LDAP</li><li>■ NIS</li><li>■ NOVELL</li></ul>
例:	names.directory_path=(tnsnames, onames)

NAMES.INITIAL_RETRY_TIMEOUT	
目的:	クライアントが優先サーバー・リスト内の次のサーバーに対する要求を送り直すまでに、Oracle Names サーバーからの応答を待つ時間を決定します。
デフォルト:	15
最小値:	1
最大値:	600
例:	names.initial_retry_timeout=20

NAMES.MAX_OPEN_CONNECTIONS	
目的:	Oracle Names クライアントが一度にオープン可能な接続数を決定します。
デフォルト:	10
最小値:	3
最大値:	64
例:	names.max_open_connections=3



---

**NAMES.MESSAGE\_POOL\_START\_SIZE**

目的:	転送されるメッセージ要求に使用するメッセージの初期数を、クライアントのメッセージ・プールに対して指定します。
デフォルト:	10
最小値:	3
最大値:	256
例:	<code>names.message_pool_start_size=10</code>

---

---

**NAMES.NIS.META\_MAP**

目的:	<b>NIS</b> 属性を NIS マップ名にマップするときに使用する <b>マップ (map)</b> ファイルを指定します。
デフォルト:	<code>sqlnet.maps</code>
例:	<code>names.nis.meta_map=sqlnet.maps</code>

---

---

**NAMES.NDS.NAME.CONTEXT**

目的:	データベース・オブジェクトが置かれている <b>NDS</b> ツリーのネーミング・コンテキストを指定します。
デフォルト:	なし
例:	<code>cn=Payroll.o=Oracle</code>

---

---

**NAMES.PREFERRED\_SERVERS**

目的:	クライアント名の要求に使用する Oracle Names サーバーの名前、アドレスおよび順序を指定します。
デフォルト:	なし
例:	<code>names.preferred_servers= (address_list=     (address=(protocol=ipc) (key=n23))     (address=(protocol=tcp) (host=nineva) (key=1575))     (address=(protocol=tcp) (host=cicada) (key=1575)))</code>

---



NAMES.REQUEST_RETRIES	
目的:	操作が失敗になる前に、優先 Oracle Name サーバー・リスト内の各 Oracle Name サーバーをクライアントが試行する回数を指定します。
デフォルト:	1
最小値:	1
最大値:	5
例:	names.request_retries=5

NAMESCTL.INTERNAL_ENCRYPT_PASSWORD	
目的:	TRUE に設定すると、NAMESCTL はパスワードを暗号化して Oracle Names サーバーに送ります。  FALSE に設定すると、NAMESCTL はパスワードを暗号化せずに Oracle Names サーバーに送ります。これによって、非暗号化されたパスワードを <a href="#">NAMES.PASSWORD</a> パラメータを使用する names.ora ファイルに設定できます。
デフォルト:	TRUE
値:	TRUE   FALSE
例:	namesctl.internal_encrypt_password=true

NAMESCTL.INTERNAL_USE	
目的	TRUE に設定すると、文書化されていない内部コマンドを NAMESCTL で使用できます。内部コマンドを識別できるように、すべての内部コマンドの先頭にアンダースコアが付いています。



---

**NAMESCTL.NO\_INITIAL\_SERVER**

目的:	TRUE に設定すると、NAMESCTL はデフォルトの Oracle Names サーバーに接続できないときにエラー・メッセージを表示しません。
デフォルト:	FALSE
値:	TRUE   FALSE
例:	<code>namesctl.no_initial_server=true</code>

---

---

**NAMESCTL.NOCONFIRM**

目的:	NAMESCTL ユーティリティの実行中に確認をとまう重要なコマンド (STOP、RELOAD、RESTART) をプロンプトするかどうかを指定します。
デフォルト:	OFF
値:	ON   OFF
例:	<code>namesctl.noconfirm=on</code>

---

---

**NAMESCTL.SERVER\_PASSWORD**

目的:	<code>names.ora</code> ファイルの <a href="#">NAMES.PASSWORD</a> パラメータで設定した構成済のパスワードに一致する値を指定します。これにより、NAMESCTL ユーティリティで STOP、RESTART、RELOAD などの保護オプションを実行するたびに、 <a href="#">SET PASSWORD</a> コマンドでパスワードを入力する必要がなくなります。
例:	<code>namesctl.server_password=secret</code>

---

---

**NAMESCTL.TRACE\_LEVEL**

目的:	NAMESCTL プログラムのトレース・レベルを指定します。
デフォルト:	OFF
値:	OFF、USER、ADMIN、SUPPORT
例:	<code>namesctl.trace_level=admin</code>

---



NAMESCTL.TRACE_FILE	
目的:	NAMESCTL のトレース出力を書き込むファイルを指定します。
デフォルト:	namesctl_pid.trc
例:	namesctl.trace_file=nmsctl

NAMESCTL.TRACE_DIRECTORY	
目的:	NAMESCTL ユーティリティのトレース出力を書き込むディレクトリを指定します。
デフォルト:	UNIX では \$ORACLE_HOME/network/trace、Windows NT では ORACLE_HOME¥network¥trace
例:	namesctl.trace_directory=/oracle/trace

NAMESCTL.TRACE_UNIQUE	
目的:	作成したトレース・ファイルの名前にプロセス識別子を付加して、各トレース・ファイルが共存できるようにするかどうかを指定します。
デフォルト:	ON
値:	ON   OFF
例:	namesctl.trace_unique=on

OSS.SOURCE.MY_WALLET	
目的:	Wallet の場所を指定します。Wallet は、SSL で処理される証明書、キーおよびトラスト・ポイントです。  <b>関連項目:</b> 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。
デフォルト:	なし
例:	oss.source.my_wallet= (source= (method=file) (method_data= (directory=/home/smalladi/oss)))



---

**SQLNET.AUTHENTICATION\_SERVICES**

目的:	1 つ以上の認証サービスを使用可能にします。認証がインストールされている場合は、このパラメータを NONE または認証方式の 1 つに設定することをお勧めします。
デフォルト:	なし
値:	<b>Net8 で使用可能な認証:</b> NONE - 使用できる認証方式はありません。有効なユーザー名とパスワードをデータベースへのアクセスに使用できます。 ALL - すべての認証方式を使用可能にします BEQ - <a href="#">Bequeath プロトコル</a> を使用します NDS - <a href="#">NDS 認証</a> を使用します NTS - <a href="#">Windows NT のシステム固有な認証 (Windows NT native authentication)</a> を使用します <b>Oracle Advanced Security で使用可能な認証:</b> KERBEROS5 - Kerberos 認証を使用します SECURID - SecurID 認証を使用します CYBERSAFE - Cybersafe 認証を使用します IDENTIX - Identix 認証を使用します RADIUS - RADIUS 認証を使用します DCEGSSAPI - DCE GSSAPI 認証を使用します <b>関連項目:</b> 『Oracle8i Advanced Security 管理者ガイド』を参照してください。 例: <code>sqlnet.authentication_services=(beq, kerberos5, cybersafe)</code>

---

---

**SQLNET.AUTHENTICATION\_KERBEROS5\_SERVICE**

目的:	Kerberos サービス・チケットを得るために使用するサービス名を定義します。 <b>関連項目:</b> 『Oracle8i Advanced Security 管理者ガイド』を参照してください。
デフォルト:	なし
例:	<code>sqlnet.authentication_kerberos5_service= oracle</code>

---



---

### SQLNET.AUTHENTICATION\_GSSAPI\_SERVICE

目的: CyberSAFE サービスのプリンシパルを定義します。

**関連項目:** 『Oracle8i Advanced Security 管理者ガイド』を参照してください。

---

---

### SQLNET.CLIENT\_REGISTRATION

目的: クライアント・マシンの一意な識別子を設定します。この識別子は接続要求とともにリスナーに渡され、監査証跡に含められます。識別子として、最大 128 文字の任意の英数字を使用できます。

デフォルト: なし

例: `sqlnet.client_registration=1432`

---

---

### SQLNET.CRYPTO\_CHECKSUM\_CLIENT

目的: クライアントのチェックサム動作を指定します。

**関連項目:** 『Oracle8i Advanced Security 管理者ガイド』を参照してください。

デフォルト: ACCEPTED

値:

- ACCEPTED - もう一方が希望する場合にセキュリティ・サーバーを有効にします。
- REJECTED - もう一方が希望していても、セキュリティ・サーバーを有効にしません。
- REQUESTED - もう一方が許可する場合にセキュリティ・サーバーを有効にします。
- REQUIRED - セキュリティ・サーバーを有効にしないと接続が実行されません。

例: `sqlnet.crypto_checksum_client=accepted`

---



---

**SQLNET.CRYPTO\_CHECKSUM\_SERVER**

目的:	サーバーのチェックサム動作を指定します。 <b>関連項目:</b> 『Oracle8i Advanced Security 管理者ガイド』を参照してください。
デフォルト:	ACCEPTED
値:	<ul style="list-style-type: none"><li>■ ACCEPTED - もう一方が希望する場合にセキュリティ・サーバーを有効にします。</li><li>■ REJECTED - もう一方が希望していても、セキュリティ・サーバーを有効にしません。</li><li>■ REQUESTED - もう一方が許可する場合にセキュリティ・サーバーを有効にします。</li><li>■ REQUIRED - セキュリティ・サーバーを有効にしないと接続が実行されません。</li></ul>
例:	sqlnet.crypto_checksum_server=accepted

---

---

**SQLNET.CRYPTO\_CHECKSUM\_TYPE\_CLIENT**

目的:	このクライアントが使用できる暗号チェックサム・アルゴリズムのリストを指定します。 <b>関連項目:</b> 『Oracle8i Advanced Security 管理者ガイド』を参照してください。
デフォルト:	MD5
値:	MD5 - RSA データ・セキュリティの MD5 アルゴリズム
例:	sqlnet.crypto_checksum_types_client=(md5)

---

---

**SQLNET.CRYPTO\_CHECKSUM\_TYPE\_SERVER**

目的:	このサーバーが使用できる暗号チェックサム・アルゴリズムのリストを指定します。 <b>関連項目:</b> 『Oracle8i Advanced Security 管理者ガイド』を参照してください。
デフォルト:	MD5
値:	MD5 - RSA データ・セキュリティの MD5 アルゴリズム
例:	sqlnet.crypto_checksum_types_server=(md5)

---



---

### SQLNET.CRYPTO\_SEED

目的: 暗号キーを生成するときに使用する文字を指定します。文字がランダムなほど、キーの安全性が向上します。10 ～ 70 文字のランダムな文字列を指定します。暗号化またはチェックサムをオンにするときは、このパラメータを指定する必要があります。暗号化は、クライアントに [SQLNET.ENCRYPTION\\_CLIENT](#) パラメータが指定され、サーバーに [SQLNET.ENCRYPTION\\_SERVER](#) パラメータが指定された場合に有効になります。チェックサムは、クライアントに [SQLNET.CRYPTO\\_CHECKSUM\\_CLIENT](#) パラメータが指定され、サーバーに [SQLNET.CRYPTO\\_CHECKSUM\\_SERVER](#) パラメータが指定された場合に有効になります。

**関連項目:** 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。

デフォルト: qwertyuiopasdfghjkl;zxcvbnm,.s1

例: sqlnet.crypto\_seed="qwertyuiopasdfghjkl;zxcvbnm,.s1"

---

---

### SQLNET.ENCRYPTION\_CLIENT

目的: クライアントの暗号化を有効にします。

**関連項目:** 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。

デフォルト: ACCEPTED

値:

- ACCEPTED - もう一方が希望する場合にセキュリティ・サーバーをオンにします。
- REJECTED - もう一方が希望していても、セキュリティ・サーバーをオンにしません。
- REQUESTED - もう一方が許可する場合にセキュリティ・サーバーをオンにします。
- REQUIRED - セキュリティ・サーバーをオンにしないと接続が実行されません。

例: sqlnet.encryption\_client=accepted

---

---

### SQLNET.ENCRYPTION\_SERVER

目的: クライアントの暗号化を有効にします。

**関連項目:** 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。

---



---

**SQLNET.ENCRYPTION\_SERVER**

デフォルト: ACCEPTED

値:

- ACCEPTED - もう一方が希望する場合にセキュリティ・サーバーを有効にします。
- REJECTED - もう一方が希望していても、セキュリティ・サーバーを有効にしません。
- REQUESTED - もう一方が許可する場合にセキュリティ・サーバーを有効にします。
- REQUIRED - セキュリティ・サーバーを有効にしないと接続が実行されません。

例: sqlnet.encryption\_server=accepted

---

---

**SQLNET.ENCRYPTION\_TYPES\_CLIENT**

目的: このクライアントが使用可能な暗号化アルゴリズムのリストを指定します。

**関連項目:** 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。

デフォルト: アルゴリズムが指定されていないときは、すべてのアルゴリズムが使用可能です。

値: 次の値を 1 つ以上指定できます。

RC4\_40 - RSA RC4 (40 ビットのキー・サイズ)

RC4\_56 - RSA RC4 (56 ビットのキー・サイズ)

RC4\_128 - RSA RC4 (128 ビットのキー・サイズ)

DES - 標準 DES (56 ビットのキー・サイズ)

DES40 - DES40 (40 ビットのキー・サイズ)

例: sqlnet.encryption\_types\_client=(rc4\_40)

---

---

**SQLNET.ENCRYPTION\_TYPES\_SERVER**

目的: このクライアントがサーバーとして機能するときに使用可能な暗号化アルゴリズムのリストを指定します。

**関連項目:** 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。

---



---

### SQLNET.ENCRYPTION\_TYPES\_SERVER

デフォルト:	アルゴリズムが指定されていないときは、すべてのアルゴリズムが使用可能です。
値:	RC4_40 - RSA RC4 (40 ビットのキー・サイズ) RC4_56 - RSA RC4 (56 ビットのキー・サイズ) RC4_128 - RSA RC4 (128 ビットのキー・サイズ) DES - 標準 DES (56 ビットのキー・サイズ) DES40 - DES40 (40 ビットのキー・サイズ)
例:	sqlnet.encryption_types_server=(rc4_40, des, ...)

---

---

### SQLNET.EXPIRE\_TIME

目的:	セッションが有効であるかどうかを確認するためのプローブを送る時間間隔を決定します。 <b>関連項目:</b> 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。
デフォルト:	なし
最小値:	0 分
推奨値:	10 分
例:	sqlnet.expire_time=10

---

---

### SQLNET.IDENTIX\_FINGERPRINT\_DATABASE

目的:	認証指紋データベースのサービス名または別名を指定します。 <b>関連項目:</b> 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。
デフォルト:	なし
例:	sqlnet.identix_fingerprint_database=fingrdb

---

---

### SQLNET.IDENTIX\_FINGERPRINT\_DATABASE\_USER

目的:	指紋データベースに予約済ユーザー名を指定します。 <b>関連項目:</b> 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。
-----	--

---



---

**SQLNET.IDENTIX\_FINGERPRINT\_DATABASE\_USER**

デフォルト: なし

例: sqlnet.identix\_fingerprint\_database\_user=manager

---

---

**SQLNET.IDENTIX\_FINGERPRINT\_DATABASE\_PASSWORD**

目的: 指紋データベースに予約済パスワードを指定します。

**関連項目:** 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。

デフォルト: なし

例: sqlnet.identix\_fingerprint\_database\_password=password

---

---

**SQLNET.IDENTIX\_FINGERPRINT\_METHOD**

目的: 指紋データベースのメソッド名を指定します。メソッド名はORACLE にしてください。

**関連項目:** 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。

デフォルト: なし

例: sqlnet.identix\_fingerprint\_method=oracle

---

---

**SQLNET.KERBEROS5\_CC\_NAME**

目的: Kerberos 資格証明キャッシュ・ファイルの完全パス名を指定します。

**関連項目:** 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。

デフォルト: UNIX では /usr/tmp/krbcache、Windows では  
c:¥tmp¥krbcache

例: sqlnet.kerberos5\_cc\_name= /usr/tmp/krbcache

---



---

### SQLNET.KERBEROS5\_CLOCKSKEW

目的:	Kerberos 資格証明が無効になるまでの秒数を指定します。 <b>関連項目:</b> 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。
デフォルト:	300
例:	sqlnet.kerberos5_clockskew=1200

---

---

### SQLNET.KERBEROS5\_CONF

目的:	Kerberos 構成ファイルの完全パス名を指定します。この構成ファイルは、デフォルト KDC の領域が入っていて、領域を KDC ホストにマップします。 <b>関連項目:</b> 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。
デフォルト:	UNIX では /krb5/krb.conf、Windows では c:\krb5\krb.conf
例:	sqlnet.kerberos5_conf=/krb5/krb.conf

---

---

### SQLNET.KERBEROS5\_KEYTAB

目的:	Kerberos プリンシパル / シークレット・キー・マッピング・ファイルの完全パス名を指定します。このファイルを使用して、キーを抽出し、着信認証情報を解釈します。 <b>関連項目:</b> 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。
デフォルト:	UNIX では /etc/v5srvtab、Windows では c:\krb5\v5srvtab
例:	sqlnet.kerberos5_keytab=/etc/v5srvtab

---

---

### SQLNET.KERBEROS5\_REALMS

目的:	Kerberos 領域変換ファイルの完全パス名を指定します。このファイルを使用して、ホスト名またはドメイン名から領域にマップします。 <b>関連項目:</b> 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。
-----	--



---

**SQLNET.KERBEROS5\_REALMS**

デフォルト: UNIX では /krb5/krb.realms、Windows では  
c:\krb5\krb.realms

例: sqlnet.kerberos5\_realms= /krb5/krb.realms

---

---

**SQLNET.RADIUS\_ALTERNATE**

目的: プライマリ・サーバーを利用できないときに使用する、代替用の RADIUS サーバーを指定します。サーバーの IP アドレスまたはホスト名を指定できます。

**関連項目:** 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。

デフォルト: なし

例: sqlnet.radius\_altermate=radius2

---

---

**SQLNET.RADIUS\_ALTERNATE\_PORT**

目的: 代替用の RADIUS サーバーのリスニング・ポートを指定します。

**関連項目:** 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。

デフォルト: 1645

例: sqlnet.radius\_altermate\_port= 1667

---

---

**SQLNET.RADIUS\_ALTERNATE\_RETRIES**

目的: 代替 RADIUS サーバーに Oracle Server がメッセージを再送する回数を指定します。

**関連項目:** 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。

デフォルト: 3

例: sqlnet.radius\_altermate\_retries=4

---



---

### SQLNET.RADIUS\_AUTHENTICATION

目的:	プライマリ RADIUS サーバーの場所を、ホスト名または IP アドレスで指定します。 <b>関連項目:</b> 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。
デフォルト:	ローカル・ホスト
例:	<code>sqlnet.radius_authenetication= officeacct</code>

---

---

### SQLNET.RADIUS\_AUTHENTICATION\_INTERFACE

目的:	ユーザーとの対話に使用するユーザー・インタフェースを含むクラスを指定します。 <b>関連項目:</b> 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。
デフォルト:	DefaultRadiusInterface
例:	<code>sqlnet.radius_authenetication_interface=</code>

---

---

### SQLNET.RADIUS\_AUTHENTICATION\_PORT

目的:	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。 <b>関連項目:</b> 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。
デフォルト:	1645
例:	<code>sqlnet.radius_authenetication_port= 1667</code>

---

---

### SQLNET.RADIUS\_AUTHENTICATION\_RETRIES

目的:	プライマリ RADIUS サーバーに Oracle Server がメッセージを再送する回数を指定します。 <b>関連項目:</b> 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。
デフォルト:	3
例:	<code>sqlnet.radius_authenetication_retries=4</code>

---



---

**SQLNET.RADIUS\_AUTHENTICATION\_TIMEOUT**

目的:	プライマリ RADIUS サーバーからの応答を Oracle Server が待機する秒数を指定します。 <b>関連項目:</b> 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。
デフォルト:	5 秒
例:	sqlnet.radius_authenetication_timeout= 10

---

---

**SQLNET.RADIUS\_CHALLENGE\_RESPONSE**

目的:	チャレンジ・レスポンスのオンとオフを切り替えます。
デフォルト:	OFF
値:	ON   OFF
例:	sqlnet.radius_challenge_response=ON

---

---

**SQLNET.RADIUS\_SECRET**

目的:	RADIUS シークレット・キーの場所を指定します。 <b>関連項目:</b> 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。
デフォルト:	UNIX では \$ORACLE_HOME/network/security/radius.key、 Windows NT では ORACLE_ HOME¥network¥security¥radius.key
例:	sqlnet.radius_secret= oracle/bin/admin/radiuskey

---

---

**SQLNET.RADIUS\_SEND\_ACCOUNTING**

目的:	アカウンティングを ON または OFF にします。有効にした場合、パケットはリスニング・ポート +1 のアクティブな RADIUS サーバーに送信されます。デフォルトのポートは 1646 です。 <b>関連項目:</b> 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。
デフォルト:	OFF
値:	ON   OFF



SQLNET.RADIUS\_SEND\_ACCOUNTING

例: sqlnet.radius\_send\_accounting=ON

SSL\_CLIENT\_AUTHENTICATION

目的: サーバーに加えて、クライアントを SSL を使用して認証するかどうかを指定します。  
**関連項目:** 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。

デフォルト: TRUE

値: TRUE | FALSE

SSL\_CIPHER\_SUITES

目的: 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。  
**関連項目:** 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。

デフォルト: なし

値: Cipher Suites の詳細は、『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。

例: ssl\_cipher\_suite=(ssl\_rsa\_with\_rc4\_138\_md5)

SSL\_VERSION

目的: SSL 接続のバージョンを合せます。  
クライアントとサーバーは、互換性のあるバージョンを使用する必要があります。  
**関連項目:** 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。

デフォルト: UNDETERMINED

値: UNDETERMINED | 2.0 | 3.0

例: ssl\_version=2.0



---

**TNSPING.TRACE\_DIRECTORY**

目的: TNSPING ユーティリティ・トレース・ファイルの作成先となるディレクトリを指定します。

デフォルト: \$ORACLE\_HOME/network/trace

例: tnsping.trace\_directory=/oracle/traces

---

---

**TNSPING.TRACE\_LEVEL**

目的: 指定のレベルでの TNSPING ユーティリティのトレースのオン / オフを切り替えます。

デフォルト: OFF

値:

- OFF - トレース出力しません
- USER - ユーザーによるエラー状況のトレース情報
- ADMIN - 管理トレース情報
- SUPPORT - オラクル社カスタマ・サポート・センター用のトレース情報

例: tnsping.trace\_level=admin

---

---

**TRACE\_DIRECTORY\_CLIENT**

目的: クライアント・トレース・ファイルの作成先となるディレクトリを指定します。

デフォルト: \$ORACLE\_HOME/network/trace

例: trace\_directory\_client=/oracle/traces

---

---

**TRACE\_DIRECTORY\_SERVER**

目的: サーバー・トレース・ファイルの作成先となるディレクトリを指定します。

デフォルト: \$ORACLE\_HOME/network/trace

例: trace\_directory\_server=/oracle/traces

---



TRACE_FILE_CLIENT	
目的:	クライアントのトレース・ファイル名を指定します。
デフォルト:	sqlnet.trc
例:	trace_file_client=clientsqlnet.trc

TRACE_FILE_SERVER	
目的:	サーバーのトレース・ファイル名を指定します。
デフォルト:	svr_pid.trc
例:	trace_file_server=svrsqlnet.trc

TRACE_FILELEN_CLIENT	
目的:	クライアント・トレース・ファイルのサイズをキロバイト (KB) で指定します。サイズに到達すると、トレース情報が次のファイルに書き込まれます。ファイル数は、 <a href="#">TRACE_FILENO_listener_name</a> パラメータで指定します。
例:	trace_filelen_client=100

TRACE_FILELEN_SERVER	
目的:	サーバー・トレース・ファイルのサイズをキロバイト (KB) で指定します。サイズに到達すると、トレース情報が次のファイルに書き込まれます。ファイル数は、 <a href="#">TRACE_FILENO_listener_name</a> パラメータで指定します。
例:	trace_filelen_server=100



---

**TRACE\_FILENO\_CLIENT**

目的:	クライアントのトレースを行うトレース・ファイル数を指定します。このパラメータを <b>TRACE_FILELEN_listener_name</b> パラメータといっしょに設定すると、トレース・ファイルは循環方式で使用されます。最初のファイルから書き込まれ、次に 2 番目のファイルという順になります。最後のファイルが書き込まれると、最初のファイルから再使用されます。  トレース・ファイル名は、その順序番号で区別されます。たとえば、svr_pid.trc というトレース・ファイルが使用され、このパラメータが 3 に設定されている場合は、トレース・ファイルに svr1_pid.trc, svr2_pid.trc および svr3_pid.trc という名前が付けられます。  また、トレース・ファイル内のトレース・イベントの前に、ファイルの順序番号が書かれます。
デフォルト:	なし
例:	trace_fileno_server=3

---

---

**TRACE\_FILENO\_SERVER**

目的:	サーバーのトレースを行うトレース・ファイル数を指定します。このパラメータを <b>TRACE_FILELEN_listener_name</b> パラメータといっしょに設定すると、トレース・ファイルは循環方式で使用されます。最初のファイルから書き込まれ、次に 2 番目のファイルという順になります。最後のファイルが書き込まれると、最初のファイルから再使用されます。  トレース・ファイル名は、その順序番号で区別されます。たとえば、svr_pid.trc というデフォルトのトレース・ファイルが使用され、このパラメータが 3 に設定されている場合は、トレース・ファイルに svr1_pid.trc, svr2_pid.trc および svr3_pid.trc という名前が付けられます。  また、トレース・ファイル内のトレース・イベントの前に、ファイルの順序番号が書かれます。
デフォルト:	なし
例:	trace_fileno_server=3

---

---

**TRACE\_LEVEL\_CLIENT**

目的:	指定のレベルでのクライアントのトレースのオンとオフを切り替えます。
-----	-----------------------------------



---

**TRACE\_LEVEL\_CLIENT**

デフォルト:                   OFF

値:                   

- OFF - トレース出力しません
- USER - ユーザー・トレース情報
- ADMIN - 管理トレース情報
- SUPPORT - オラクル社カスタマ・サポート・センター用トレース情報

例:                   trace\_level\_client=user

---

---

**TRACE\_LEVEL\_SERVER**

目的:                   指定のレベルでのサーバーのトレースのオンとオフを切り替えます。

デフォルト:                   OFF

値:                   

- OFF - トレース出力しません
- USER - ユーザー・トレース情報
- ADMIN - 管理トレース情報
- SUPPORT - オラクル社カスタマ・サポート・センター用のトレース情報

例:                   trace\_level\_server=admin

---

---

**TRACE\_TIMESTAMP\_CLIENT**

目的:                   タイムスタンプを *dd-month-yyyy hh:mm:ss* の形式で、  
sqlnet.trc というデフォルト名のクライアント・トレース・  
ファイルのトレース・イベントに追加します。

デフォルト:                   OFF

値:                   ON または TRUE | OFF または FALSE

例:                   trace\_timestamp\_server=true

---

---

**TRACE\_TIMESTAMP\_SERVER**

目的:                   タイムスタンプを *dd-month-yy hh:mm:ss* の形式で、  
svr\_pid.trc というデフォルト名のサーバー・トレース・ファイルに追加しま  
す。



---

**TRACE\_TIMESTAMP\_SERVER**

デフォルト:	OFF
値:	ON または TRUE   OFF または FALSE
例:	trace_timestamp_server=true

---

---

**TRACE\_UNIQUE\_CLIENT**

目的:	クライアント・トレース・ファイルごとに一意のトレース・ファイルを作成するかどうかを決定します。値を ON に設定すると、sqlnetpid.trc という名前の複数のトレース・ファイルが共存できるように、各トレース・ファイルの名前にプロセス識別子が付加されます。値を OFF に設定すると、クライアントで新しいトレース・ファイルが作成されるときに、既存のファイルが上書きされます。
-----	--

デフォルト:	ON
例:	trace_unique_client=on

---



---

**USE\_CMAN**

目的:

TRUE に設定した場合は、クライアントに対して、最初のアドレスが Oracle Connection Manager の位置、2 番目のアドレスがリスナーの位置になっているアドレス・リストが含まれている接続記述子を使用するように指定されます。この方法で、クライアントに Oracle Connection Manager へのルートが設定されます。

次の例には 2 つのアドレス・リストが記載されています。最初のアドレス・リストによってクライアントに Oracle Connection Manager へのルートが設定され、2 番目のアドレス・リストによってクライアントからリスナーへのルートが設定されます。

```
sales=
  (description=
    (load_balance=on)
    (failover=on)
    (address_list=
      (source_route=yes)
      (address=(protocol=tcp) (host=host1) (port=1630))
      (address=(protocol=tcp) (host=host2) (port=1521)))
    (address_list=
      (address=(protocol=tcp) (host=host3) (port=1521)))
    (connect_data=(service_name=sales.us.acme.com)))
```

USE\_CMAN=TRUE を使用しない場合、クライアントにはアドレス・リストの中から 1 つがランダムに選択されます。ADDRESS\_LIST の選択に失敗した場合は、別のアドレス・リストにフェイルオーバーします。USE\_CMAN=TRUE を使用する場合は、常にクライアントに最初のアドレス・リストが使用されます。

Oracle Connection Manager のアドレスが不明な場合は、利用できるリスナー・アドレスを介して接続のルートが設定されます。

**注意:** Oracle Names で Oracle Connection Manager を使用している場合は、クライアントと Oracle Names サーバーの両方で、このオプションを設定する必要があります。

デフォルト:

FALSE

値:

TRUE | FALSE

例:

use\_cman=true

---



**USE\_DEDICATED\_SERVER**

目的:	ON に設定した場合は、(SERVER=DEDICATED) が接続記述子の接続データに自動的に追加されます。この方法では、MTS が構成されている場合にも、このクライアントによる接続に専用サーバーが使用されます。  クライアントが使用する接続記述子の CONNECT_DATA セクションに (SERVER=DEDICATED) を追加します。このパラメータは、 <b>SERVER</b> パラメータの現行の値を上書きします。
デフォルト:	OFF
値:	<ul style="list-style-type: none"> <li>■ ON - 専用サーバー・プロセスを生成します</li> <li>■ OFF - 既存のサーバー・プロセスに渡します</li> </ul>
例:	use_dedicated_server=on

## ローカル・ネーミング・パラメータ (tnsnames.ora)

ローカル・ネーミング・メソッドを使用すると、ネット・サービス名が tnsnames.ora ファイルに追加されます。tnsnames.ora ファイルは、UNIX プラットフォームの \$ORACLE\_HOME/network/admin、Windows プラットフォームの ORACLE\_HOME¥network¥admin に置かれているか、あるいは、TNS\_ADMIN 環境変数またはレジストリ値によって指定されたディレクトリに置かれています。

ネット・サービス名は、接続記述子に含まれているデータベース・ネットワーク・アドレスにマップされた別名です。接続記述子には、プロトコル・アドレスによるリスナーの場所や、接続するデータベースのサービス名が含まれます。他のサーバーのクライアントになるクライアントとサーバーは、このネット・サービス名を使用してアプリケーションに接続します。

## 一般構文

tnsnames.ora ファイルの基本的な構文を、例 C-1 に示します。DESCRIPTION には接続記述子、ADDRESS にはリスナー・アドレス、そして CONNECT\_DATA にはデータベース・サービスの識別情報が含まれます。

**例 C-1 tnsnames.ora の一般構文**

```
net_service_name=
(
  description=
    (
      address= (protocol_address_information))
    (connect_data=
      (service_name=service_name)))
)
```



## 複数の記述

tnsnames.ora ファイルには、1つ以上の接続記述子を持つネット・サービス名を登録できます。各接続記述子には、1つ以上のリスナー・アドレスを登録できます。例 C-2 に、複数のアドレスを持っている接続記述子を 2 つ示します。DESCRIPTION\_LIST は、接続記述子のリストを定義します。

---

---

**注意：** Oracle Net8 Assistant は、ネット・サービス名の複数の接続記述子はサポートしません。

---

---

### 例 C-2 tnsnames.ora の複数の記述子

```
net_service_name=
(description_list=
  (description=
    (address= (protocol_address_information))
    (address= (protocol_address_information))
    (address= (protocol_address_information))
    (connect_data=
      (service_name=service_name))))
  (description=
    (address= (protocol_address_information))
    (address= (protocol_address_information))
    (address= (protocol_address_information))
    (connect_data=
      (service_name=service_name)))))
```

## 複数のアドレス・リスト

tnsnames.ora は、それぞれに異なる複数のアドレス・リストを持っている接続記述子もサポートします。例 C-3 は、2つのアドレス・リストを示したものです。最初のアドレス・リストでは、クライアント・ロード・バランスを指定し、接続時フェイルオーバーを使用しません。これらの設定は、ADDRESS\_LIST の ADDRESS に指定されたリスナーにしか適用されません。次のアドレス・リストでは、接続時フェイルオーバーを指定し、クライアント・ロード・バランスを使用しません。これらの設定は、ADDRESS\_LIST の ADDRESS に指定されたリスナーにしか適用されません。クライアントはまず、1 番目または 2 番目のアドレスをランダムに選び、次に 3 番目と 4 番目のアドレスを順に試します。

---

---

**注意：** Oracle Net8 Assistant は、接続記述子に 1 つのアドレス・リストしかサポートしていません。

---

---

### 例 C-3 tnsnames.ora の複数のアドレス・リスト

```
net_service_name=
```



```
(description=
  (address_list=
    (load_balance=on)
    (failover=off)
    (address= (protocol_address_information))
    (address= (protocol_address_information)))
  (address_list=
    (load_balance=off)
    (failover=on)
    (address= (protocol_address_information))
    (address= (protocol_address_information)))
  (connect_data=
    (service_name=service_name)))
```

---

**注意：**リリース 8.1 以前のように、アドレス・リストが1つしかない場合は、リストを ADDRESS\_LIST に埋め込む必要はありません。

---

## クライアント・ロード・バランスと DESCRIPTION\_LIST の変更された機能

リリース 8.1 以前は、DESCRIPTION\_LIST が複数のリスナーのクライアント・ロード・バランスに使用されていました。クライアント・ロード・バランスは、現在でも DESCRIPTION\_LIST ではデフォルトで有効ですが、ADDRESS\_LIST に明示的に指定することができ、一連の ADDRESS や DESCRIPTION に関連付けることもできます。例 C-4 は、リリース 8.1 以前とリリース 8.1 の tnsnames.ora を、Oracle Parallel Server 環境のクライアント・ロード・バランスを使用して比較したものです。



**例 C-4 リリース 8.1 とそれ以前のリリースのクライアント・ロード・バランス****リリース 8.1 以前のクライアント・ロード・バランス**

次の例には、op、op1 および op2 のネット・サービス名を持っているリリース 8.0 の tnsnames.ora ファイルを示します。

op では、2 つのインスタンスのいずれかをクライアントがランダムに選択できます。op1 と op2 では、クライアントは特定の Oracle Parallel Server インスタンスに接続できます。

```
op=
  (description_list=
    (description=
      (address=
        (protocol=tcp)
        (host=opsnt1)
        (port=1521))
      (connect_data=(sid=op1)))

    (description=
      (address=
        (protocol=tcp)
        (host=opsnt2)
        (port=1521))
      (connect_data=(sid=op2))))

op1=
  (description=
    (address=
      (protocol=tcp)
      (host=opsnt1)
      (port=1521))
    (connect_data=(sid=op1)))

op2=
  (description=
    (address=
      (protocol=tcp)
      (host=opsnt1)
      (port=1521))
    (connect_data=(sid=op2)))
```

**リリース 8.1 でのクライアント・ロード・バランス**

次の例には、op、op1 および op2 のネット・サービス名を持っているリリース 8.1 の tnsnames.ora ファイルを示します。

op では、クライアントは、データベース・サービス op.com にランダムに接続でき、複数の DESCRIPTION のかわりに複数の ADDRESS を使用して 2 つのインスタンスのいずれかを選択できます。op1 および op2 では、クライアントは特定の Oracle Parallel Server インスタンスに接続できます。

```
op=
  (description=
    (load_balance=on)
    (address=
      (protocol=tcp)
      (host=opsnt1)
      (port=1521))
    (address=
      (protocol=tcp)
      (host=opsnt2)
      (port=1521))
    (connect_data=
      (service_name=op.com)))

op1=
  (description=
    (address=
      (protocol=tcp)
      (host=opsnt1)
      (port=1521))
    (connect_data=
      (service_name=op)
      (instance_name=op1)))

op2=
  (description=
    (address=
      (protocol=tcp)
      (host=opsnt2)
      (port=1521))
    (connect_data=
      (service_name=op)
      (instance_name=op2)))
```

---



## 例

クライアントが Oracle Connection Manager を通してサービスに接続する場合、最初のアドレス（またはアドレスのセット）は Oracle Connection Manager に接続するためのクライアント用のアドレスで、最後のアドレス（またはアドレスのセット）は、リスナーを使用してサービスに接続する Oracle Connection Manager 用のアドレスとなります。

Oracle Connection Manager 用の SOURCE\_ROUTE パラメータを YES に設定すると、クライアントは接続する際にリストにあるすべてのアドレスを通ります。

例 C-5 は、複数の Oracle Connection Manager アドレスを使用する場合の例です。

### 例 C-5 tnsnames.ora に指定する複数の Oracle Connection Manager アドレス

```
sample1=
  (description=
    (source_route=yes)
    (address=(protocol=tcp) (host=host1) (port=1630))      # hop 1
    (address_list=
      (failover=on)
      (load_balance=off)                                   # hop 2
      (address=(protocol=tcp) (host=host2a) (port=1630))
      (address=(protocol=tcp) (host=host2b) (port=1630)))
    (address=(protocol=tcp) (host=host3) (port=1521))      # hop 3
    (connect_data=(service_name=sales.us.acme.com)))
```

例 C-5 では次のようになります。

1. クライアントは、最初の Oracle Connection Manager のアドレスに接続するように指示されます。

```
(address=(protocol=tcp) (host=host1) (port=1630))
```

2. 最初の Oracle Connection Manager は、Oracle Connection Manager アドレスの 1 つを使用して、他の Oracle Connection Manager に接続します。1 番目のアドレスに失敗した場合、2 番目のアドレスで接続を試みます。この動作は、次の部分で指示されます。

```
(address_list=
  (failover=on)
  (load_balance=off)
  (address=(protocol=tcp) (host=host2a) (port=1630))
  (address=(protocol=tcp) (host=host2b) (port=1630)))
```

3. 続いて Oracle Connection Manager は、次のリスナー・アドレスを使用してデータベース・サービスに接続します。

```
(address=(protocol=tcp) (host=host3) (port=1521))
```



例 C-6 は、2 つの Oracle Connection Manager と 2 つのリスナー・アドレス間のクライアント・ロード・バランスの例を示しています。

#### 例 C-6 tnsnames.ora におけるクライアント・ロード・バランス

```
sample2=
(description=
  (load_balance=on)
  (failover=on)
  (address_list=
    (source_route=yes)
    (address=(protocol=tcp) (host=host1) (port=1630))
    (address=(protocol=tcp) (host=host2) (port=1521)))
  (address_list=
    (source_route=yes)
    (address=(protocol=tcp) (host=host3) (port=1630))
    (address=(protocol=tcp) (host=host4) (port=1521)))
  (connect_data=(service_name=sales.us.acme.com)))
```

上の例では次の動作が行われます。

1. クライアントは、ADDRESS\_LIST をランダムに選択し、選択した ADDRESS\_LIST で失敗した場合、他にフェイルオーバーするように指示されます。この動作は、LOAD\_BALANCE および FAILOVER パラメータを ON にすることで指示されます。
2. ADDRESS\_LIST が選択されると、クライアントはまず ADDRESS\_LIST に指定されたポート 1630 を使用する Oracle Connection Manager のアドレスを使用して、Oracle Connection Manager に接続します。
3. 続いて Oracle Connection Manager は、ADDRESS\_LIST に指定されたリスナー・アドレスを使用して、データベース・サービスに接続します。

## DESCRIPTIONS および DESCRIPTION\_LIST

---

### DESCRIPTION

目的: 接続記述子の開始を表します。データベース・リスニング・アドレスの定義、接続するデータベースのサービス名が指定されます。このパラメータは、DESCRIPTION\_LIST の下に埋め込むことができます。

例: 

```
net_service_name=
(description=
  (address=...)
  (connect_data=(service_name=sales.us.acme.com)))
```

---



---

## DESCRIPTION\_LIST

目的: 特定のネット・サービス名の接続記述子のリストを定義します。

例:

```
net_service_name=
(description_list=
  (description=
    (address=...)
    (connect_data=(service_name=sales.acme.com)))
  (description=
    (address=...)
    (connect_data=(service_name=sales2.us.acme.com))))
```

---

## ADDRESS と ADDRESS\_LIST

---

### ADDRESS

目的: リスナー・プロトコル・アドレスを定義します。このパラメータは、ADDRESS\_LIST または DESCRIPTION の下に埋め込むことができます。

**関連項目:** 各プロトコルで使用する適正なパラメータの説明は、[付録 B](#) を参照してください。

例:

```
net_service_name=
(description=
  (address=(protocol=tcp) (host=sales-svr) (port=1521))
  (connect_data=(service_name=sales.us.acme.com)))
```

---

---

### ADDRESS\_LIST

目的: アドレスのリストを定義します。アドレス・リストしか存在しない場合、ADDRESS\_LIST は必要ありません。このパラメータは、DESCRIPTION または DESCRIPTION\_LIST の下に埋め込むことができます。

例:

```
net_service_name=
(description=
  (address_list=
    (address=(protocol=tcp) (host=sales1-svr) (port=1521))
    (address=(protocol=tcp) (host=sales2-svr) (port=1521)))
  (address_list=
    (failover=on)
    (address=(protocol=tcp) (host=sales3-svr) (port=1521))
    (address=(protocol=tcp) (host=sales4-svr) (port=1521))
    (connect_data=(service_name=sales.us.acme.com)))
```

---



## リストのオプション・パラメータ

---

### FAILOVER

目的: ON に設定した場合は、最初のリスナーへの接続に失敗した後で、Net8 は別のリスナーにフェイルオーバーします。OFF に設定すると、接続を試行するアドレスは1つのみです。

このパラメータは、DESCRIPTION\_LIST、DESCRIPTION または ADDRESS\_LIST の下に埋め込むことができます。

**重要:** GLOBAL\_DBNAME パラメータは、listener.ora の SID\_LIST\_listener\_name セクションには設定しないでください。静的に構成されたグローバル・データベース名によって接続時フェイルオーバーが使用できなくなります。

デフォルト: DESCRIPTION\_LIST、DESCRIPTION、ADDRESS\_LIST で ON

値: ON | OFF | YES | NO | TRUE | FALSE

例:

```
net_service_name=
(description=
  (failover=on)
  (address=(protocol=tcp) (host=sales1-svr) (port=1521))
  (address=(protocol=tcp) (host=sales2-svr) (port=1521)))
(connect_data=(service_name=sales.us.acme.com))
```

---

---

### LOAD\_BALANCE

目的: ON に設定した場合、アドレスのリストを順不同に選択して、各種リスナーまたは Oracle Connection Manager のアドレスにおけるロード・バランスをとるように Net8 を設定します。OFF に設定した場合、Net8 は、接続が成功するまでアドレスを順番に試行するように設定されます。

このパラメータは、DESCRIPTION\_LIST、DESCRIPTION または ADDRESS\_LIST の下に埋め込むことができます。

デフォルト: DESCRIPTION\_LIST で ON

値: ON | OFF | YES | NO | TRUE | FALSE

例:

```
net_service_name=
(description=
  (load_balance=on)
  (address=(protocol=tcp) (host=sales1-svr) (port=1521))
  (address=(protocol=tcp) (host=sales2-svr) (port=1521)))
(connect_data=(service_name=sales.us.acme.com))
```

---



---

**SDU**

目的:	Net8 に、ネットワークを介して送信されるデータ・パケットの転送レートを、指定したセッション・データ・ユニット (SDU) サイズで最適化するように指示します。このパラメータは、DESCRIPTION の下に埋め込む必要があります。
デフォルト:	2048
値:	SDU サイズは、通常の伝送フレーム・サイズの倍数に設定する必要があります。イーサネットの通常のフレーム・サイズは 1500 であるため、イーサネット・プロトコルでの最も効果的な SDU サイズは 1500 の倍数です。
例:	<pre>net_service_name= (description=   (sdu=3000)   (address=...)   (address=...)   (connect_data=     (server_name=sales.com))</pre>

---

---

**SOURCE\_ROUTE**

目的:	<p>このパラメータを ON または YES に設定すると、Net8 は接続先に到達するまで、各アドレスを順に使用します。</p> <p>このパラメータは Oracle Connection Manager で必要です。Oracle Connection Manager では、クライアントから Oracle Connection Manager への初期の接続が必要です。また、Oracle Connection Manager からリスナーへの第 2 の接続も必要です。</p> <p>このパラメータは、DESCRIPTION_LIST、DESCRIPTION または ADDRESS_LIST の下に埋め込むことができます。</p>
デフォルト:	OFF
値:	YES   NO   ON   OFF
例:	<pre>net_service_name= (description=   (source_route=on)   (address=(protocol=tcp) (host=cman-pc) (port=1630))   (address=(protocol=tcp) (host=sales1-svr) (port=1521)))   (connect_data=(service_name=sales.us.acme.com))</pre>

---



---

## TYPE\_OF\_SERVICE

目的:

Oracle Rdb データベースで使用するサービスの種類を指定します。このサービスは RDB インタフェース・ツールで使用されます。この機能は、アプリケーションが Oracle Rdb と Oracle データベースの両方をサポートし、アプリケーションを任意に選択する（ロード・バランスをとる）場合にのみ使用します。このパラメータは、DESCRIPTION の下に埋め込む必要があります。

例:

```
net_service_name=
(description_list=
(description=
(address=...))
(connect_data=
(service_name=generic)
(rdb_database=[.mf]mf_personal.rdb)
(global_name=alpha5))
(type_of_service=rdb_database))
(description=
(address=(protocol=tcp) (host=sales1-svr) (port=1521))
(connect_data=
(service_name=sales.us.acme.com))
(type_of_service=oracle8_database)))
```

---

## CONNECT\_DATA セクション

---

### CONNECT\_DATA

目的:

接続するデータベースを定義します。このパラメータは、DESCRIPTION の下に埋め込む必要があります。

例:

```
net_service_name=
(description=
(address=(protocol=tcp) (host=sales1-svr) (port=1521))
(address=(protocol=tcp) (host=sales2-svr) (port=1521))
(connect_data=(service_name=sales.us.acme.com)))
```

---

CONNECT\_DATA には次のパラメータが許可されています。

- [FAILOVER\\_MODE](#)
- [GLOBAL\\_NAME](#)
- [HS](#)
- [INSTANCE\\_NAME](#)
- [RDB\\_DATABASE](#)



- [SERVER](#)
- [SERVICE\\_NAME](#)
- [SID](#)

---

**FAILOVER\_MODE**

目的:

Net8 に、実行時に最初リスナーで失敗した場合に、別のリスナーにフェイルオーバーするように指示します。構成に応じて、進行中のセッションや SELECT 文は自動的にフェイルオーバーされます。このパラメータは、CONNECT\_DATA の下に埋め込む必要があります。

このタイプのフェイルオーバーは、[透過的アプリケーション・フェイルオーバー \(Transparent Application Failover:TAF\)](#) と呼ばれています。[接続時フェイルオーバー FAILOVER](#) パラメータと混同しないようにしてください。

**関連項目:**

- 概念的な情報は、2-19 ページの「[可用性を高めるための透過的アプリケーション・フェイルオーバー](#)」を参照してください。
  - 詳細な完全構成情報は、8-11 ページの「[透過的アプリケーション・フェイルオーバーの構成](#)」を参照してください。
-



---

## FAILOVER\_MODE

### サブパラメータ

FAILOVER\_MODE は、次のサブパラメータをサポートしています。

- **BACKUP** - サービス名によりフェイルオーバー・ノードを指定します。個別のネット・サービス名をフェイルオーバー・ノードに作成する必要があります。
- **TYPE - (必須)** フェイルオーバーの種類を指定します。デフォルトでは、Oracle Call Interface (OCI) アプリケーションに対して、3 種類の Net8 フェイルオーバー機能を利用できます。

**SESSION:** セッションをフェイルオーバーします。すなわち、ユーザーの接続が失われた場合に、新しいセッションを自動的に作成して、ユーザーをバックアップします。このフェイルオーバーでは、SELECT はリカバリされません。

**SELECT:** カーソルをオープンしているユーザーが、障害後もカーソルをフェッチできるようにします。ただしこのモードでは、通常の実行操作を行うクライアント側にオーバーヘッドが生じます。

**NONE:** デフォルトです。フェイルオーバー機能を使用しません。フェイルオーバーが実行されないように、明示的に指定することもできます。

- **METHOD:** プライマリ・ノードからバックアップ・ノードにフェイルオーバーされる速さを決定します。

**BASIC:** フェイルオーバー時に接続を確立します。フェイルオーバー時まで、バックアップ・サーバーはほとんど何もする必要はありません。

**PRECONNECT:** あらかじめ接続を確立しておきます。フェイルオーバーは高速に行われますが、バックアップ・インスタンスはサポートされた各インスタンスからの接続すべてをサポートする必要があります。

- **RETRIES:** 接続を試みる回数を指定します。DELAY が指定されている場合、RETRIES のデフォルトは、5 回再試行するように設定されています。
- **DELAY:** 次に接続を試みるまでの待機時間を秒数で指定します。RETRIES が指定されている場合の DELAY のデフォルト設定は 1 秒です。

例：

実装例は、8-12 ページの「**TAF 実装**」を参照してください。

---



---

**GLOBAL\_NAME**

目的: Oracle Rdb データベースを識別します。このパラメータは、CONNECT\_DATA の下に埋め込む必要があります。

例:

```
net_service_name=
  (description=
    (address=...)
    (address=...)
    (connect_data=
      (service_name=generic)
      (rdb_database= [.mf]mf_personal.rdb)
      (global_name= alpha5)))
```

---

---

**HS**

目的: Net8 に、非 Oracle システムに接続するように指示します。このパラメータは、CONNECT\_DATA の下に埋め込む必要があります。

デフォルト: なし

値: OK

例:

```
net_service_name=
  (description=
    (address=...)
    (address=...)
    (connect_data=
      (sid=sales6)
      (hs=ok)))
```

---

---

**INSTANCE\_NAME**

目的: アクセスするデータベース・インスタンスを識別します。インスタンス名は、初期化パラメータ・ファイルの INSTANCE\_NAME パラメータから取得できます。

このパラメータは、CONNECT\_DATA の下に埋め込む必要があります。

**関連項目:** インスタンス名文字列の設定についての情報は、6-2 ページの「[接続記述子の理解](#)」を参照してください。

---



---

## INSTANCE\_NAME

例:

```
net_service_name=
(description=
(address=...)
(address=...)
(connect_data=
(service_name=sales.com)
(instance_name=sales1)))
```

---

---

## RDB\_DATABASE

目的: Oracle Rdb データベースのファイル名を指定します。このパラメータは、CONNECT\_DATA の下に埋め込む必要があります。

例:

```
net_service_name=
(description=
(address=...)
(address=...)
(connect_data=
(service_name=sales.com)
(rdb_database= [.mf]mf_personal.rdb)))
```

---

---

## SERVER

目的: 特定タイプのサービス・ハンドラにクライアントを接続するようにリスナーを設定します。このパラメータは、CONNECT\_DATA の下に埋め込む必要があります。

値: DEDICATED | SHARED

DEDICATED は、このクライアントを専用サーバー・プロセスに接続します。

SHARED は、このクライアントを共有サーバーに接続します。

**注意:** このクライアントを共有サーバーに接続するには、MTS が構成されている必要があります。

**注意:** sqlnet.ora ファイルの [USE\\_DEDICATED\\_SERVER](#) は、このパラメータを上書きします。

例:

```
net_service_name=
(description=
(address=...)
(address=...)
(connect_data=
(server_name=sales.com)
(server=dedicated)))
```

---



---

## SERVICE\_NAME

目的: アクセスするリリース 8.1 サービスを識別します。このパラメータは、CONNECT\_DATA の下に埋め込む必要があります。

SERVICE\_NAME は、一般にグローバル・データベース名に設定されます。グローバル・データベース名は、データベース名 (DB\_NAME) とドメイン名 (DB\_DOMAIN) から構成され、インストール中またはデータベースの作成中に入力します。

### 関連項目:

- SERVICE\_NAME の使用の詳細は、2-27 ページの「[SID ではなくサービス名によるデータベースの識別](#)」を参照してください。
- インスタンス名を設定する方法は、6-2 ページの「[接続記述子の理解](#)」を参照してください。

例:

```
net_service_name=
(description=
(address=...)
(address=...)
(connect_data=
(service_name=sales.com)))
```

---

---

## SID

目的: Oracle8 または Oracle7 のデータベースのデータベース・インスタンスを識別します。データベースが Oracle8i の場合は、SID ではなく [SERVICE\\_NAME](#) を使用してください。

**関連項目:** 2-27 ページの「[SID ではなくサービス名によるデータベースの識別](#)」を参照してください。

このパラメータは、CONNECT\_DATA の下に埋め込む必要があります。

例:

```
net_service_name=
(description=
(address=...)
(address=...)
(connect_data=
(sid=sales)))
```

---

# リスナー・パラメータ (listener.ora)

ネットワーク・リスナー構成ファイル (listener.ora) では、次の項目を使用できます。

- [リスナー・アドレス・セクション](#)



- [SID\\_LIST\\_listener\\_name 静的サービス・セクション](#)
- [制御パラメータ](#)

listener.ora ファイルは、UNIX の \$ORACLE\_HOME/network/admin、Windows NT の ORACLE\_HOME¥network¥admin にあるか、あるいは TNS\_ADMIN 環境変数またはレジストリ値によって指定されたディレクトリにあります。

## リスナー・アドレス・セクション

listener.ora ファイルのリスナー・アドレス・セクションは、リスナー・プロトコル・アドレスを定義します。

---

### DESCRIPTION

目的:	リスナー・プロトコル・アドレスを定義します。 <b>関連項目:</b> 構成に関する情報については、8-39 ページの「 <a href="#">非 Oracle データベース・サービスへの接続の構成</a> 」を参照してください。
例:	<pre>listener_name= (description=   (address=(protocol=ipc) (key=extproc0))   (address=(protocol=tcp) (host=sales-pc) (port=1521)))</pre>

---

---

### ADDRESS

目的:	リスナー・プロトコル・アドレスを定義します。このパラメータは、ADDRESS_LIST または DESCRIPTION の下に埋め込むことができます。 <b>関連項目:</b> 各プロトコルで使用する適正なパラメータの説明は、 <a href="#">付録 B</a> を参照してください。
例:	<pre>listener_name= (description=   (address=(protocol=ipc) (key=extproc0))   (address=(protocol=tcp) (host=sales-pc) (port=1521)))</pre>

---

---

### ADDRESS\_LIST

目的:	リスナー・プロトコル・アドレスのリストを定義します。アドレスのリストが 1 つのみの場合、このパラメータを使用する必要はありません。このパラメータは、DESCRIPTION の下に埋め込むことができます。
-----	--



---

**ADDRESS\_LIST**

```

例：
listener_name=
(description=
  (address_list=
    (address=(protocol=ipc) (key=extproc0))
    (address=(protocol=tcp) (host=sales-pc) (port=1521))))

```

---

**SID\_LIST\_listener\_name 静的サービス・セクション**

listener.ora の SID\_LIST セクションを使用して、サービス情報をリスナーによって静的に構成できます。

Oracle8 または Oracle7 のデータベース、外部プロシージャおよび Oracle Enterprise Manager を含む一部の管理ツールとの下位互換性を維持するには、SID\_LIST が必要です。

Oracle8i データベース情報は、インスタンスの起動時にリスナーを使用して動的に登録されます。したがって、Oracle8i データベースを監視するために Oracle Enterprise Manager を使用する場合以外は、この情報は必要ありません。

```

sid_list_listener_name=
(sid_list=
  (sid_desc=
    (global_dbname=global_database_name)
    (sid_name=sid)
    (oracle_home=oracle_home)
    (prespawn_max=99)
    (prespawn_list=
      (prespawn_desc=
        (protocol=tcp)
        (pool_size=10)
        (timeout=2))
      (prespawn_desc=...)))
    (sid_desc=...))

```

SID\_LIST には、次のパラメータが含まれます。

---

**GLOBAL\_DBNAME**

目的： データベースのグローバル・データベース名を識別します。この名前には、データベース名およびデータベース・ドメインが登録されています。GLOBAL\_DBNAME 値は、初期化パラメータ・ファイルの SERVICE\_NAMES パラメータから取得できます。

このパラメータは、SID\_DESC の下に埋め込む必要があり、SERVICE\_NAMES パラメータの値と一致させてください。



---

## GLOBAL\_DBNAME

例:

```
sid_list_listener_name=
  (sid_list=
    (sid_desc=
      (global_dbname=oracle.com)
      (sid_name=orcl)
      (oracle_home=/usr/oracle)))
```

---

---

## ORACLE\_HOME

目的: サービスの Oracle ホームの場所を識別します。このパラメータは、SID\_DESC の下に埋め込む必要があります。

例:

```
sid_list_listener_name=
  (sid_list=
    (sid_desc=
      (sid_name=extproc)
      (oracle_home=/usr/oracle)
      (program=extproc)))
```

---

---

## PROGRAM

目的: サービスの実行可能プログラム名を識別します。

例:

```
sid_list_listener_name=
  (sid_list=
    (sid_desc=
      (sid_name=extproc)
      (oracle_home=oracle)
      (program=extproc)))
```

---

---

## SID\_NAME

目的: インスタンスの Oracle システム識別子 (SID) を識別します。SID 値は、初期化パラメータ・ファイルの INSTANCE\_NAME パラメータから取得できます。

このパラメータは、SID\_DESC の下に埋め込む必要があります。

例:

```
sid_list_listener_name=
  (sid_list=
    (sid_desc=
      (global_dbname=oracle.com)
      (sid_name=orcl)
      (oracle_home=/usr/oracle)))
```

---



---

## SID\_LIST

目的: SID の説明リストを識別します。

例: 

```
sid_list_listener_name=  
  (sid_list=  
    (sid_desc=...)  
    (sid_desc=...))
```

---

---

## SID\_DESC

目的: 指定のデータベース・インスタンスに関するサービス情報を提供します。

このパラメータは、SID\_LIST の下に埋め込むことができます。

例: 

```
sid_list_listener_name=  
  (sid_list=  
    (sid_desc=...)  
    (sid_desc=...))
```

---

---

## PRESPAWN\_MAX

目的: リスナーによって作成される事前生成済専用サーバー・プロセスの最大数を指定します。この数値は、最小でも各プロトコル用のプール・サイズの合計と同じくする必要があります。事前に生成される専用サーバー・プロセスが常に新しい接続に使用できるように、この値を大きな値に設定します。

このパラメータは、SID\_LIST の下に埋め込む必要があります。

**注意:** 事前に生成される専用サーバーは、Windows NT 上では構成できません。

例: 

```
sid_list_listener_name=  
  (sid_list=  
    (sid_desc=  
      (global_dbname=oracle.com)  
      (sid_name=sid)  
      (oracle_home=/usr/oracle)  
      (prespawn_max=99)))
```

---

---

## PRESPAWN\_LIST

目的: 事前生成済専用サーバー・プロセスのプロトコル説明のリストを指定します。このパラメータは、SID\_DESC の下に埋め込む必要があります。



---

## PRESPAWN\_LIST

例 :

```
sid_list_listener_name=
(sid_list=
(sid_desc=
(global_dbname=oracle.com)
(sid_name=sid)
(oracle_home=/usr/oracle)
(prespawn_max=99)
(prespawn_list=
(prespawn_desc=
(protocol=tcp)
(pool_size=10)
(timeout=2))
(prespawn_desc=...)))
```

---

---

## PRESPAWN\_DESC

目的 :

リスナーによって事前に生成される専用サーバー・プロセスのプロトコル、および事前に生成される専用サーバーの特性を定義します。プロトコルにはそれぞれ固有の PRESPAWN\_DESC 説明が必要です。このパラメータは、PRESPAWN\_LIST の下に埋め込む必要があります。

**関連項目 :** PROTOCOL、POOL\_SIZE および TIMEOUT の説明は、7-13 ページの「[事前生成済専用サーバーの構成](#)」を参照してください。

例 :

```
sid_list_listener_name=
(sid_list=
(sid_desc=
(global_dbname=oracle.com)
(sid_name=sid)
(oracle_home=/usr/oracle)
(prespawn_max=99)
(prespawn_list=
(prespawn_desc=
(protocol=tcp)
(pool_size=10)
(timeout=2))
(prespawn_desc=...)))
```

---



## 制御パラメータ

次のパラメータはリスナーの動作を制御します。

---

### **CONNECT\_TIMEOUT\_*listener\_name***

目的: 接続の開始後にリスナーが有効な接続要求を待つ時間を秒数で指定します。

デフォルト: 10 秒

例: connect\_timeout\_listener=12

---

---

### **LOG\_DIRECTORY\_*listener\_name***

目的: リスナー・ログ・ファイルの作成先となるディレクトリを指定します。

デフォルト: UNIX では \$ORACLE\_HOME/network/log、Windows NT では ORACLE\_HOME¥network¥log

例: log\_directory\_listener=/oracle/network/admin/log

---

---

### **LOG\_FILE\_*listener\_name***

目的: リスナーのログ・ファイルの名前を指定します。

デフォルト: listener.log

例: log\_file\_listener=list.log

---

---

### **LOGGING\_*listener\_name***

目的: ログイングをオンまたはオフにします。

デフォルト: ON

値: ON | OFF

例: logging\_listener=on

---



---

### OSS.SOURCE.MY\_WALLET

目的:	Wallet の場所を指定します。Wallet は、安全に接続できるようにする SSL で処理される証明書、キーおよびトラスト・ポイントです。 <b>関連項目:</b> 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。
デフォルト:	なし
例:	<pre>oss.source.my_wallet=   (source=     (method=file)     (method_data=       (directory=/home/smalladi/oss)))</pre>

---

---

### PASSWORDS\_listener\_name

目的:	リスナーに非暗号化パスワードを設定し、 <a href="#">SAVE_CONFIG</a> および <a href="#">STOP</a> など、LSNRCTL ユーティリティから使用される特定の特権操作が安全に行われるようにします。1 つ以上のパスワードを設定できます。非暗号化パスワードは、LSNRCTL ユーティリティの <a href="#">CHANGE_PASSWORD</a> コマンドで設定できます。
デフォルト:	oracle
例:	<pre>passwords_listener=(oracle8)</pre>

---

---

### SAVE\_CONFIG\_ON\_STOP\_listener\_name

目的:	TRUE に設定すると、LSNCRCTL <a href="#">SET</a> コマンドを使用して変更したパラメータはすべて、以前の listener.ora ファイルの設定を置き換えます。
デフォルト:	FALSE
例:	<pre>save_config_on_stop_listener=true</pre>

---

---

### SSL\_CLIENT\_AUTHENTICATION

目的:	SSL を使用してクライアントを認証するかどうかを指定します。
デフォルト:	TRUE
値:	TRUE   FALSE

---



---

**SSL\_CLIENT\_AUTHENTICATION**

使用上の注意: データベースがクライアントを認証します。したがって、この値は FALSE に設定する必要があります。このパラメータを TRUE に設定すると、リスナーがクライアントを認証しようとしませんが、その結果障害が発生する可能性があります。

**関連項目:** 『Oracle8i Advanced Secutiry 管理者ガイド』を参照してください。

---

---

**STARTUP\_WAIT\_TIME\_listener\_name**

目的: ネットワーク・リスナーが、最初の LSNRCTL [STATUS](#) コマンドに応答するまでの待ち時間を秒数で設定します。

デフォルト: 0 秒

例: startup\_wait\_time\_listener=5

---

---

**TRACE\_DIRECTORY\_listener\_name**

目的: リスナー・トレース・ファイルの作成先となるディレクトリを指定します。

デフォルト: UNIX では \$ORACLE\_HOME/network/trace、Windows NT では `ORACLE_HOME¥network¥trace`

例: trace\_directory\_listener=/oracle/network/admin/trace

---

---

**TRACE\_FILE\_listener\_name**

目的: リスナーのトレース・ファイルの名前を指定します。

デフォルト: listener.trc

例: trace\_file\_listener=list.trc

---

---

**TRACE\_FILELEN\_listener\_name**

目的: リスナー・トレース・ファイルのサイズをキロバイト (KB) で指定します。サイズに到達すると、トレース情報が次のファイルに書き込まれます。ファイル数は、[TRACE\\_FILELEN\\_listener\\_name](#) パラメータで指定します。



---

### **TRACE\_FILELEN\_*listener\_name***

デフォルト: 無制限

例: `trace_filelen_listener=100`

---

---

### **TRACE\_FILENO\_*listener\_name***

目的: リスナーのトレースを行うトレース・ファイル数を指定します。このパラメータを `TRACE_FILENO_listener_name` パラメータと一っしょに設定すると、トレース・ファイルは循環方式で使用されます。最初のファイルから書き込まれ、次に 2 番目のファイルという順になります。最後のファイルが書き込まれると、最初のファイルから再使用されます。

トレース・ファイル名は、その順序番号で区別されます。たとえば、`listener.trc` のデフォルト・ファイルを使用し、このパラメータを 3 に設定すると、トレース・ファイルの名前は `listener1.trc`、`listener2.trc` および `listener3.trc` になります。

また、トレース・ファイル内のトレース・イベントの前にファイルの順序番号が書かれます。

デフォルト: 1

例: `trace_fileno_listener=3`

---

---

### **TRACE\_LEVEL\_*listener\_name***

目的: 指定したレベルでトレースをオン / オフに切り替えます。

デフォルト: OFF

値:

- OFF - トレース出力しません
- USER - ユーザー・トレース情報
- ADMIN - 管理トレース情報
- SUPPORT - オラクル社カスタマ・サポート・センター用のトレース情報

例: `trace_level_listener=admin`

---



---

**TRACE\_TIMESTAMP\_listener\_name**

目的:	タイムスタンプを <i>dd-month-yyyy hh:mm:ss</i> の形式で、リスナー・トレース・ファイルのトレース・イベントに追加します。
デフォルト:	OFF
値:	ON または TRUE   OFF または FALSE
例:	<code>trace_timestamp_listener=true</code>

---

---

**USE\_PLUG\_AND\_PLAY\_listener\_name**

目的:	リスナーに、データベース情報を Oracle Names サーバーに登録するよう指示します。
デフォルト:	OFF
値:	ON   OFF
例:	<code>use_plug_and_play=on</code>

---

## Oracle Names パラメータ (names.ora)

次のパラメータは、Oracle Names 構成ファイル (names.ora) で使用できます。names.ora は、UNIX では \$ORACLE\_HOME/network/admin、Windows NT では ORACLE\_HOME¥network¥admin にあります。

---

**NAMES.ADDRESSES**

目的:	Oracle Names サーバーが受信するプロトコル・アドレスのリストを作成します。有効な ADDRESS または ADDRESS_LIST を指定します。
-----	---

**関連項目:**

- アドレス構文の説明は、B-2 ページの「[ADDRESS と ADDRESS\\_LIST](#)」を参照してください。
- 各プロトコルで使用する適正なパラメータの説明は、B-2 ページの「[プロトコルのパラメータ](#)」を参照してください。

デフォルト:	<code>names.addresses= (address=(protocol=tcp) (host=oramesrvr0) (port=1575))</code>
--------	--

---



---

### NAMES.ADMIN\_REGION

目的:	管理リージョンのデータ・ソースを記述します。このパラメータを設定すると、データベースが情報のリポジトリとして定義されます。このパラメータを設定しなければ、Oracle Names サーバー間でデータのレプリケーションがキャッシュされます。
デフォルト:	NULL
例:	<pre>names.admin_region=   (region=     (description=       (address=(protocol=tcp)         (host=nineva) (port=1575)))     (connect_data=       (service_name=sales.us.acme.com)))     (userid=system)     (password=manager)     (refresh=172800)     (retry=2700)     (expire=8700)     (version=34619392))</pre>

---

---

### NAMES.AUTHORITY\_REQUIRED

目的:	システム問合せに権限応答が必要かどうかを指定します。
デフォルト:	FALSE
例:	<pre>names.authority_required=true</pre>

---

---

### NAMES.AUTO\_REFRESH\_EXPIRE

目的:	NAMES.DOMAIN.HINTS パラメータを使用して取得された別のリージョンのデータベース・サーバー・アドレスを、Oracle Names サーバーがキャッシュに入れておく時間を秒数で指定します。指定した時間が経過すると、Oracle Names サーバーは別のリージョンのデータベース・サーバーに問合せを発行してアドレスを更新します。
デフォルト:	600 秒
指定可能な値:	60 ~ 1209600 秒
例:	<pre>names.auto_refresh_expire=1200000</pre>

---



---

**NAMES.AUTO\_REFRESH\_RETRY**

目的:	ドメイン・ヒント・リスト上の Oracle Names サーバーを Oracle Names サーバーが再試行する間隔を秒数で指定します。
デフォルト:	180
最小値:	60
最大値:	3600
例:	<code>names.auto_refresh_retry=180</code>

---

---

**NAMES.CACHE\_CHECKPOINT\_FILE**

目的:	Oracle Names サーバーがチェックポイント・ファイルを書き込むファイルの名前とパスを指定します。
デフォルト:	UNIX では <code>\$ORACLE_HOME/network/names/ckpcch.ora</code> 、 Windows NT では <code>ORACLE_HOME\network\names\ckpcch.ora</code>
例:	<code>names.cache_checkpoint_ file=c:\oracle\network\names\cachecck.ora</code>

---

---

**NAMES.CACHE\_CHECKPOINT\_INTERVAL**

目的:	Oracle Names サーバーが、格納済データのチェックポイントをチェックポイント・ファイルに書き込む間隔を秒数で指定します。各 Oracle Names サーバーは、キャッシュに格納したデータを定期的にファイルに書き込んで、起動時の障害に備えることができます。
デフォルト:	0 (キャッシュ頻度なし)
最小値:	10 秒
最大値:	259200 秒 (3 日)
例:	<code>names.cache_checkpoint_interval=24</code>

---

---

**NAMES.CONFIG\_CHECKPOINT\_FILE**

目的:	Oracle Names サーバーの構成設定のチェックポイントに使用するファイルの名前とパスを指定します。
-----	---

---



---

### NAMES.CONFIG\_CHECKPOINT\_FILE

デフォルト: UNIX では \$ORACLE\_HOME/network/names/ckpcfg.or、  
Windows NT では ORACLE\_  
HOME¥network¥names¥ckpcfg.ora

例: names.config\_checkpoint\_  
file=c:¥oracle¥network¥names¥configck.ora

---

---

### NAMES.CONNECT\_TIMEOUT

目的: Oracle Names サーバーが、クライアントからの接続が完了するの  
を待つ時間を秒数で制限します。

デフォルト: 3 秒

最小値: 1 秒

最大値: 600 秒

例: names.connect\_timeout=8

---

---

### NAMES.DEFAULT\_FORWARDERS

目的: 問合せを転送するときに使用する、その他の Oracle Names サー  
バーのアドレス・リストを指定します。

例: names.default\_forwarders= (forwarder\_list=  
(forwarder=(name=rootserv1.com) (address=(protocol=tcp)  
(port=4200) (host=roothost))))

---

---

### NAMES.DEFAULT\_FORWARDERS\_ONLY

目的: TRUE に設定した場合は、デフォルトの送り先としてリストされ  
ている Oracle Names サーバーのみを対象に、Oracle Names サー  
バーが [NAMES.DEFAULT\\_FORWARDERS](#) パラメータを使用し  
て問合せを送ります。

デフォルト: FALSE

使用上の注意: FALSE に設定すると、デフォルトの送り先としてリストされてい  
る Oracle Names サーバーは、キャッシュ内で見つかる Oracle  
Names サーバーより前にコールされます。

---



---

**NAMES.DOMAIN\_HINTS**

目的: 1 つ以上のリモート・リージョンのサーバー名およびアドレス、ドメインをすべてリストします。Oracle Names サーバーが、その他のリージョンにある Oracle Names サーバーを認識できるようになります。この中には、ルート・リージョンに存在しない Oracle Names サーバーも含めてすべての Oracle Names サーバーのルート・リージョンが含まれます。最適化で必要になれば、その他のリージョンを指定できます。

例: 

```
names.domain_hints=
(hint_desc=(hint_list=
(hint=(name=rootserv1.com)
(address=(protocol=tcp) (host=nineva) (port=4200))))
```

---

---

**NAMES.DOMAINS**

目的: サーバーのローカル・リージョン内のドメインのリスト、およびこれらのドメイン内のデータに対するデフォルト生存時間 (TTL) を指定します。

例: 

```
names.domains=
(domain_list=
(domain=(name=) (min_ttl=86400))
(domain=(name=com) (min_ttl=86400)))
```

---

---

**NAMES.FORWARDING\_AVAILABLE**

目的: ON に設定すると、Oracle Names サーバーはクライアント要求をリモートの Oracle Names サーバーに送ります。OFF に設定すると、ローカル・ドメインの外部のネットワークにアクセスできないクライアントが名前を解決できなくなります。

デフォルト: ON

値: ON | OFF

例: 

```
names.forwarding_available=off
```

---



NAMES.FORWARDING_DESIRED	
目的:	TRUE に設定した場合、Oracle Names サーバーからクライアントにリモートの Oracle Names サーバーアドレスの場所の情報が提供されます。この方法で、クライアントは該当する Oracle Names サーバーにリダイレクトされます。FALSE に設定した場合は、Oracle Names サーバーがクライアントに代わってリモートの Oracle Names サーバーに接続します。
デフォルト:	TRUE
値:	TRUE   FALSE
例:	names.forwarding_desired=true

NAMES.KEEP_DB_OPEN	
目的:	リージョン・データベースへの TNS 接続を、操作間でオープンしたままにするかどうかを指定します。このパラメータを FALSE に設定すると、ロード、再ロードまたは再ロードチェックが終わるたびに接続がクローズされます。
デフォルト:	TRUE
値:	TRUE   FALSE
例:	names.keep_db_open=FALSE

NAMES.LOG_DIRECTORY	
目的:	Oracle Names サーバー操作イベントを書き込むログ・ファイルの作成先となるディレクトリを指定します。
デフォルト:	UNIX では \$ORACLE_HOME/network/log、Windows NT では ORACLE_HOME¥network¥log
例:	names.log_directory=c:¥oracle¥network¥names

NAMES.LOG_FILE	
目的:	Oracle Names サーバー操作イベントを書き込む出力ファイルの名前を指定します。ファイルの拡張子は常に .log です。このパラメータの拡張子は入力しないでください。
デフォルト:	NAMES



---

**NAMES.LOG\_FILE**

例: `names.log_file=onames`

---

---

**NAMES.LOG\_STATS\_INTERVAL**

目的: ログ・ファイルに完全な統計をダンプする間隔を秒数で指定します。

デフォルト: 0 (0=OFF)

最小値: 10 秒

最大値: なし

例: `names.log_stats_interval=12`

---

---

**NAMES.LOG\_UNIQUE**

目的: このパラメータを TRUE に設定すると、ログ・ファイル名が一意的な名前になるので、既存のログ・ファイルは上書きされません。

デフォルト: FALSE

値: TRUE | FALSE

例: `names.log_unique=true`

---

---

**NAMES.MAX\_OPEN\_CONNECTIONS**

目的: Oracle Names サーバーが一度にオープンできる接続数を指定します。この値には、リスニング用の 1 接続、クライアント用の 5 接続、ローカル管理リージョンで定義されている各リモート・ドメイン用の 1 接続の合計値または 10 のどちらか大きい方が生成されます。

計算で求めた値を、ほとんどのインストレーションに適用できます。

デフォルト: 入力データに基づいて計算されます。

最小値: 2

最大値: 64

例: `names.max_open_connections=52`

---



NAMES.MAX_REFORWARDS	
目的:	サーバーが操作を転送する最大回数を指定します。
デフォルト:	2
最小値:	1
最大値:	15
例:	names.max_reforwards=2

NAMES.MESSAGE_POOL_START_SIZE	
目的:	転送されるメッセージの着信または発信に使用するメッセージの初期数を、サーバーのメッセージ・プールに対して指定します。
デフォルト:	10
最小値:	3
最大値:	256
例:	names.message_pool_start_size=10

NAMES.NO_MODIFY_REQUESTS	
目的:	TRUE に設定すると、サーバーはリージョン内のデータを変更する操作を拒否します。
デフォルト:	FALSE
値:	TRUE   FALSE
例:	names.no_modify_requests=true

NAMES.NO_REGION_DATABASE	
目的:	TRUE に設定すると、サーバーはリージョン・データベースを検索しません。
デフォルト:	FALSE
例:	names.no_region_database=true



---

**NAMES.PASSWORD**

目的:	Oracle Names サーバーに暗号化されたパスワードを設定し、 NAMESCTL ユーティリティから使用される <b>STOP</b> 、 <b>RESTART</b> および <b>RELOAD</b> などの特定の特権操作が安全に実行されるようにします。  このパラメータを Oracle Net8 Assistant で設定すると、パスワードが暗号化されます。クリアテキスト・パスワードは手動で設定できます。パスワードがクリアテキストの場合は、 <code>sqlnet.ora</code> ファイルに設定されている <b>NAMESCTL.INTERNAL_ENCRYPT_PASSWORD</b> パラメータを <b>FALSE</b> に設定します。
デフォルト:	なし
例:	<code>names.password=625926683431aa55</code>

---

---

**NAMES.REGION\_CHECKPOINT\_FILE**

目的:	リージョンのデータ（ローカル・リージョン内の Oracle Names サーバーのドメイン・アドレス、データベース・アドレスなど）にチェックポイントを使用するためのファイルの名前とパスを指定します。
デフォルト:	UNIX では <code>\$ORACLE_HOME/network/names/ckpreg.ora</code> 、 Windows NT では <code>ORACLE_HOME\network\names\ckpreg.ora</code>
例:	<code>names.region_checkpoint_ file=c:\oracle\network\names\regionck.ora</code>

---

---

**NAMES.RESET\_STATS\_INTERVAL**

目的:	Oracle Names サーバーが収集した統計を蓄積する時間を秒数で指定します。頻度が指定されると、統計はゼロにリセットされます。デフォルト値の 0 は、統計がリセットされないことを表します。
デフォルト:	0（リセットしない）
最小値:	10 秒
最大値:	なし
例:	<code>names.reset_stats_interval=15</code>

---



---

### NAMES.SAVE\_CONFIG\_ON\_STOP

目的:	TRUE に設定した場合、Oracle Names サーバーは実行時の構成設定を <code>names.ora</code> ファイルに保存します。NAMESCTL SET 操作で変更されたパラメータはすべて、変更前の <code>names.ora</code> 設定を上書きします。
デフォルト:	FALSE
例:	<code>names.save_config_on_stop=FALSE</code>

---

---

### NAMES.SERVER\_NAME

目的:	各 Oracle Names サーバーを名前で一意に識別します。特定の Oracle Names サーバーを参照するときは、必ずこの名前を使用します。
デフォルト:	<code>ONAMES_onames_server</code>
例:	<code>names.server_name=namesrv1.us.oracle.com</code>

---

---

### NAMES.TRACE\_DIRECTORY

目的:	Oracle Names サーバートレース・セッションのトレース・ファイルを書き込むディレクトリの名前を指定します。
デフォルト:	UNIX では <code>\$ORACLE_HOME/network/trace</code> 、Windows NT では <code>ORACLE_HOME¥network¥trace</code>
例:	<code>names.trace_directory=/oracle/network/admin/trace</code>

---

---

### NAMES.TRACE\_FILE

目的:	Oracle Names サーバートレース・セッションの出力ファイルの名前を指定します。このファイルの拡張子は常に <code>.trc</code> です。
デフォルト:	NAMES
例:	<code>names.trace_file=onames</code>

---

---

### NAMES.TRACE\_FUNC

目的:	関数名トレースを有効にします。
デフォルト:	FALSE

---



---

**NAMES.TRACE\_FUNC**

例: `names.trace_func=false`

---

---

**NAMES.TRACE\_LEVEL**

目的: Oracle Names サーバーのトレース・レベルを指定します。

デフォルト: OFF

値:

- OFF - トレース出力しません
- USER - ユーザー・トレース情報
- ADMIN - 管理トレース情報
- SUPPORT - オラクル社カスタマ・サポート・センター用のトレース情報

例: `names.trace_level=admin`

---

---

**NAMES.TRACE\_UNIQUE**

目的: 各トレース・ファイルに一意な名前を割り当て、複数のトレース・ファイルが存在できるかどうかを指定します。この値を ON に設定すると、プロセス識別子が各トレース・ファイルの名前に追加されます。

たとえば、デフォルトのトレース・ファイル名である `names.trc` が使用される場合は、トレース・ファイル名 `namespid.trc` が作成されます。

デフォルト: ON

値: ON | OFF

例: `names.trace_unique=on`

---

## Oracle Connection Manager パラメータ (cman.ora)

Oracle Connection Manager の構成ファイル `cman.ora` では、次のパラメータが使用可能です。`cman.ora` ファイルは、UNIX の `$ORACLE_HOME/network/admin`、Windows NT の `ORACLE_HOME¥network¥admin` に置かれているか、あるいは、`TNS_ADMIN` 環境変数またはレジストリ値によって指定されたディレクトリに置かれています。



---

## CMAN

目的: Oracle Connection Manager ゲートウェイ・プロセス CMGW のリスニング・アドレスを指定します。

デフォルト: `cman=(address=(protocol=tcp) (host=local_host) (port=1630))`

構文: `cman=`  
`([address_list= ]`  
`(address= ...)`  
`[(address= ...)])`

---

---

## CMAN\_ADMIN

目的: Oracle Connection Manager 管理プロセス **CMADMIN** ([Oracle Connection Manager 管理プロセス](#)) のリスニング・アドレスを指定します。

デフォルト: `cman_admin=(address=(protocol=tcp) (host=anyhost) (port=1830))`

構文: `cman_admin=`  
`([address_list= ]`  
`(address= ...)`  
`[(address= ...)])`

---

---

## CMAN\_PROFILE

目的: Oracle Connection Manager に関連するパラメータを設定します。

---



---

**CMAN\_PROFILE**

デフォルト:

- ANSWER\_TIMEOUT=0
  - AUTHENTICATION\_LEVEL=0
  - LOG\_LEVEL=0
  - MAX\_FREELIST\_BUFFERS=0
  - MAXIMUM\_CONNECT\_DATA=1024
  - MAXIMUM\_RELAYS=128
  - RELAY\_STATISTICS=NO
  - REMOTE\_ADMIN=NO
  - SHOW\_TNS\_INFO=NO
  - UNIX では TRACE\_DIRECTORY=\$ORACLE\_HOME/network/trace、Windows NT では TRACE\_DIRECTORY=ORACLE\_HOME¥network¥trace
  - TRACE\_FILELEN=unlimited
  - TRACE\_FILENO=1
  - TRACE\_TIMESTAMP=NO
  - TRACING=NO
  - USE\_ASYNC\_CALL=YES
-



---

## CMAN\_PROFILE

値 :

- ANSWER\_TIMEOUT=[0 ~ *n*]
- AUTHENTICATION\_LEVEL=[0 | 1]
- LOG\_LEVEL=[0 ~ 4]
- MAXIMUM\_CONNECT\_DATA=[257 ~ 4096]
- MAX\_FREELIST\_BUFFERS=[0 ~ 10240]
- MAXIMUM\_RELAYS=[1 ~ 2048]
- RELAY\_STATISTICS=[YES | TRUE | ON | 1 | NO | FALSE | OFF | 0]
- REMOTE\_ADMIN=[YES | TRUE | ON | 1 | NO | FALSE | OFF | 0]
- SHOW\_TNS\_INFO=[YES | TRUE | ON | 1 | NO | FALSE | OFF | 0]
- TRACE\_DIRECTORY= ディレクトリ
- TRACE\_FILELEN=KB で表されたファイル・サイズ
- TRACE\_FILENO= 数値
- TRACE\_TIMESTAMP=[YES | TRUE | ON | 1 | NO | FALSE | OFF | 0]
- TRACING=[YES | TRUE | ON | 1 | NO | FALSE | OFF | 0]
- USE\_ASYNC\_CALL=[YES | TRUE | ON | 1 | NO | FALSE | OFF | 0]

例 :

```
cman_profile=
(parameter_list=
  (maximum_relays=512)
  (log_level=1)
  (relay_statistics=yes)
  (remote_admin=yes)
  (show_tns_info=yes)
  (use_async_call=yes)
  (authentication_level=0)
  (tracing=yes)
  (trace_timestamp=yes)
  (trace_filelen=100)
  (trace_fileno=2))
```

---



CMAN_PROFILE 属性	説明
ANSWER_TIMEOUT	<p>Oracle Connection Manager が、着信接続要求に関連するプロトコル・ハンドシェークをタイムアウトする秒数を決定します。</p> <p>設定範囲は 0 ～ <math>n</math> です。</p>
AUTHENTICATION_LEVEL	<p>Oracle Connection Manager が施行できる次のセキュリティのレベルを設定します。</p> <ul style="list-style-type: none"> <li>1 の場合は、安全性の高いネットワーク・サービス (SNS) を使用しない接続要求を拒否するように Oracle Connection Manager を設定します。SNS は、Oracle Advanced Security の一部です。</li> <li>0 (デフォルト) の場合、クライアントとサーバーの間の SNS をチェックしないように Oracle Connection Manager を設定します。</li> </ul>
LOG_LEVEL	<p>Oracle Connection Manager で実行されるロギングのレベルを決定します。</p> <ul style="list-style-type: none"> <li>デフォルトは 0 で、ロギングを実行しません。</li> <li>範囲は 0 ～ 4 です。</li> </ul> <p>レベル 1 - 基本的なレポートの作成            レベル 2 - RULE_LIST 一致検索レポートの作成            レベル 3 - リレー・ブロッキング・レポートの作成            レベル 4 - リレー I/O カウント・レポートの作成</p> <p>CMGW ゲートウェイ・プロセスはログ・ファイル <code>cman_pid.log</code> を作成し、CMADMIN 管理プロセスは、ログ・ファイル <code>cmadm_pid.log</code> を作成します。この 2 つのログ・ファイルは、UNIX では <code>\$ORACLE_HOME/network/log</code>、Windows NT では <code>ORACLE_HOME¥network¥log</code> に配置されます。</p>
MAX_FREELIST_BUFFERS	<p>リレーがクローズした後に、TNS がオペレーティング・システムに返さずに将来の再使用のためにフリーリストに維持するバッファの最大値を決定します。</p> <p>設定範囲は 0 ～ 10240 です。</p>
MAXIMUM_CONNECT_DATA	<p>着信接続要求の接続データ文字列の長さを制限します。</p> <p>設定範囲は 257 ～ 4096 です。</p>
MAXIMUM_RELAYS	<p>Oracle Connection Manager がサポートする、同時接続の最大数を決定します。</p> <p>設定範囲は 1 ～ 2048 です。</p>



CMAN_PROFILE 属性	説明
RELAY_STATISTICS	<p>I/O 統計を記録するかどうかを決定します。</p> <p>YES の場合は、リレー I/O アクティビティに関する次の統計を維持するように Oracle Connection Manager を設定します。</p> <ul style="list-style-type: none"><li>■ IN バイト数</li><li>■ OUT バイト数</li><li>■ IN パケット数</li><li>■ OUT パケット数</li></ul> <p>この情報は、cman_pid.log ファイルに格納されます。</p> <p>NO の場合、I/O 統計は記録されません。</p>
REMOTE_ADMIN	<p>Oracle Connection Manager へのリモート・アクセスを許可するかどうかを決定します。</p> <p>YES の場合、リモート CMCTL セッションから Oracle Connection Manager へのアクセスが許可されます。</p> <p>NO の場合、ローカル Oracle Connection Manager へのアクセスのみが許可されます。その結果、リモート CMCTL セッションを実行しているユーザーは Oracle Connection Manager にアクセスできなくなります。</p>
SHOW_TNS_INFO	<p>TNS 情報を記録するかどうかを決定します。</p> <p>YES の場合、TNS 情報を cman_pid.log ファイルに格納するように Oracle Connection Manager を設定します。</p> <p>NO の場合、TNS イベントはログ・ファイルに挿入されません。</p>
TRACING	<p>Oracle Connection Manager でトレーシングを使用できるようにするかどうかを決定します。</p> <p>YES を設定すると、Oracle Connection Manager でのトレースを使用可能にします。CMGW プロセスはトレース・ファイル cman_pid.trc を作成し、CMADMIN 管理プロセスはトレース・ファイル cmadm_pid.trc を作成します。</p>
TRACE_DIRECTORY	<p>トレース・ディレクトリを指定します。<a href="#">TRACE_FILELEN</a> パラメータまたは <a href="#">TRACE_TIMESTAMP</a> パラメータの使用を計画している場合は、このパラメータをデフォルト以外の場所に設定してください。</p>



CMAN_PROFILE 属性	説明
TRACE_FILELEN	<p>トレース・ファイルのサイズをキロバイト (KB) で指定します。サイズに到達すると、トレース情報が次のファイルに書き込まれます。ファイル数は、<a href="#">TRACE_FILENO</a> パラメータで指定します。</p> <p><b>注意：</b>このパラメータには、<a href="#">TRACE_DIRECTORY</a> パラメータによって指定されているデフォルト以外のトレース・ディレクトリが必要です。</p>
TRACE_FILENO	<p>トレースを行うトレース・ファイル数を指定します。このパラメータを <a href="#">TRACE_FILELEN</a> パラメータと一っしょに設定すると、トレース・ファイルは循環方式で使用されます。最初のファイルから書き込まれ、次に 2 番目のファイルという順になります。最後のファイルが書き込まれると、最初のファイルから再使用されます。</p> <p>トレース・ファイル名は、その順序番号で区別されます。たとえば、このパラメータを 3 に設定した場合、CMGW ゲートウェイ・トレース・ファイルの名前は cman1_pid.trc、cman2_pid.trc および cman3_pid.trc になり、CMADMIN 管理トレース・ファイルの名前は cmadm1_pid.trc、cmadm2_pid.trc および cmadm3_pid.trc になります。</p> <p>また、トレース・ファイル内のトレース・イベントの前に、ファイルの順序番号が書かれます。</p>
TRACE_TIMESTAMP	<p>タイムスタンプを <code>dd-month-yyyy hh:mm:ss</code> の形式で、トレース・ファイルのトレース・イベントに追加します。</p> <p><b>注意：</b>このパラメータには、<a href="#">TRACE_DIRECTORY</a> パラメータによって指定されているデフォルト以外のトレース・ディレクトリが必要です。</p>
USE_ASYNC_CALL	<p>Oracle Connection Manager が Net8 接続の確立フェーズの応答またはコールを実行しているときに、非同期ファンクションを使用するかどうかを決定します。</p> <p>YES の場合、すべての非同期ファンクションを使用するように Oracle Connection が設定されます。</p> <p>NO の場合、非同期ファンクションを使用しないように Oracle Connection が設定されます。</p> <p><b>注意：</b>Oracle Connection Manager はバンド外ブレイク (out-of-bound break) をサポートするので、サーバーにバンド外ブレイクを送ります。</p>



---

## CMAN\_RULES

目的:	<p>Oracle Connection Manager のネットワーク・アクセス制御にフィルタ規則を設定します。</p> <p>規則が指定されない場合はすべての接続が受け入れられます。この機能は、接続集中化および複数プロトコル・サポート機能でフィルタを使用するかどうかに従って使用します。</p>
構文:	<pre>cman_rules= (rule_list= (rule=   (src=host)   (dst=host)   (srv= service_name sid)   (act=accept reject)) [(rule= ...)])</pre>
値:	<ul style="list-style-type: none"><li>■ SRC - セッション要求のソース・ホスト名または IP アドレス（ドット表記法）</li><li>■ DST - 接続先サーバーのホスト名または IP アドレス（ドット表記法）</li><li>■ SRV - データベース・サーバー SID</li><li>■ ACT - 前の特性を持つ着信要求の受入れまたは拒否</li></ul>
使用上の注意:	<p>ホスト名のワイルド・カード文字は、'x' です。IP アドレス (d.d.d.d) の場合は、この d にワイルド・カード 'x' を付けることができます。</p> <p>クライアントの接続文字列に SID と SERVICE_NAME の両方が含まれている場合、クライアントがアクセスを行うには、要求された名前の両方がそれぞれ規則で許可される必要があります。</p>
例:	<pre>cman_rules= (rule_list= (rule=   (src=client1-pc)   (dst=sales-pc)   (srv= sales.acme.com)   (act=reject)) (rule=   (src=src=144.25.23.45)   (dst=144.25.187.200)   (srv= db1)   (act=accept)))</pre> <hr/>



## プロトコル固有のパラメータ (protocol.ora)

一部のプロトコルには、プラットフォーム固有のパラメータが存在します。これらは、`protocol.ora` ファイルに格納されます。`protocol.ora` ファイルは、UNIX プラットフォームの `$ORACLE_HOME/network/admin`、Windows NT プラットフォームの `ORACLE_HOME¥network¥admin` に置かれているか、あるいは `TNS_ADMIN` 環境変数またはレジストリ値によって指定されたディレクトリに置かれています。

一部の共通パラメータのリストを次に示します。使用しているオペレーティング・システムとプラットフォームの全パラメータ・リストは、使用しているオペレーティング・システム対応のマニュアルを参照してください。

---

### ***protocol.EXCLUDED\_NODES***

目的:	データベースへのアクセスを制限するクライアントを指定します。
構文:	<code>protocol.excluded_nodes= (hostname   ip_address, hostname   ip_address,...)</code>
例:	<code>tcp.excluded_nodes= (hr.com, 144.25.5.25)</code>

---



---

### ***protocol.INVITED\_NODES***

目的:	データベースへのアクセスを許可するクライアントを指定します。 <i>protocol.EXCLUDED_NODES</i> パラメータとこのリストの両方がある場合は、このリストが優先されます。
構文:	<code>protocol.invited_nodes= (hostname   ip_address, hostname   ip_address,...)</code>
例:	<code>tcp.invited_nodes= (sales.com, 144.185.5.73)</code>

---



---

### ***protocol.VALIDNODE\_CHECKING***

目的:	<i>protocol.INVITED_NODES</i> と <i>protocol.EXCLUDED_NODES</i> をチェックして、アクセスを許可または拒否するクライアントを決定します。
デフォルト:	NO
値:	YES   NO
例:	<code>tcp.validnode_checking=yes</code>

---



TCP.NODELAY	
目的:	TCP/IP プロトコル・スタック内でのバッファ・フラッシングで遅延なしを指定します。
デフォルト:	NO
値:	YES   NO
例:	tcp.nodelay=yes

## ディレクトリ・サーバー・アクセス・パラメータ (ldap.ora)

次のパラメータは、クライアントおよびサーバー上の Net8 Configuration Assistant によって、ldap.ora ファイルに設定されています。

これらのパラメータまたはその設定は変更しないでください。ldap.ora ファイルは、UNIX プラットフォームの \$ORACLE\_HOME/network/admin、Windows NT プラットフォームの ORACLE\_HOME¥network¥admin に置かれているか、あるいは TNS\_ADMIN 環境変数またはレジストリ値によって指定されたディレクトリに置かれています。

DIRECTORY_SERVERS	
目的:	主 LDAP ディレクトリ・サービスと代替 LDAP ディレクトリ・サービスのリストをそのホスト、ポート番号といっしょに登録します。
値:	host:port[:sslport]
例:	directory_servers=dlsun1778:389, raffles:400:636

DIRECTORY_SERVER_TYPE	
目的:	使用されている LDAP ディレクトリ・サーバーのタイプを登録します。
値:	OID   AD   NDS OID - Oracle インターネット・ディレクトリ AD - Microsoft のアクティブ・ディレクトリ NDS - Novell ディレクトリ・サービス
例:	directory_type=oid

DEFAULT_ADMIN_CONTEXT	
目的:	管理コンテキスト (administrative context) を指定します。



DEFAULT_ADMIN_CONTEXT	
値 :	有効 DN
例 :	networkobj_context="o=OracleSoftware,c=US"

## リリース 8.1 での構成ファイルの変更点

この項では、次のトピックについて説明します。

- リリース 8.1 で廃止されたパラメータ
- 構成ファイルで使用する構文の変更点
- tnsnames.ora の変更点
- cman.ora の変更点

## リリース 8.1 で廃止されたパラメータ

次の表は、リリース 8.1 で廃止されたパラメータについて説明しています。

パラメータ	該当するファイル	説明
AUTOMATIC_IPC	sqlnet.ora	セッションに強制的に IPC アドレスを使用させるために使用したパラメータです。パフォーマンス上の問題から、このパラメータは破棄されました。AUTOMATIC_IPC のかわりに、IPC アドレスを構成する必要があります。
NAMES.USE_PLUG_AND_PLAY	names.ora	このパラメータは、動的検出オプションを有効または無効にするために使用されていました。Oracle Names サーバーを検出する別の方法が用意されています。 <b>関連項目：</b> 6-43 ページの「 <a href="#">Oracle Names メソッドの構成</a> 」を参照してください。
NAMES.DOMAIN_CHECKPOINT_FILE	names.ora	チェックポイント・ドメイン・データ（リージョン内に存在するすべてのデータベース・アドレスとその他のデータ）に使用されていたファイルの名前。 この情報に関するチェックポイント・ファイルを作成するには、NAMES.REGION_CHECKPOINT_FILE パラメータを使用する必要があります。
NAMES.TOPOLOGY_CHECKPOINT_FILE	names.ora	チェックポイント・トポロジー・データ（ローカル・リージョン内に存在するサーバーのドメイン・アドレス）に使用されていたファイルの名前。 この情報に関するチェックポイント・ファイルを作成するには、NAMES.REGION_CHECKPOINT_FILE パラメータを使用する必要があります。



構成ファイルで使用される構文の変更点

次の表は、Net8 のリリース 8.0 と 8.1 の構文における相違点を説明しています。

影響のある要素	Net8 リリース 8.0	Net8 リリース 8.1
tnsnames.ora ファイル内で、システム識別子 (SID) ではなく、サービス名とインスタンス名を指定して接続	<p>次に示すように、データベースの SID は tnsnames.ora の CONNECT_DATA セクションで指定する必要があります。</p> <pre>service_name=   (description=     (address_list=       (address=...)       (address=...))   (connect_data=(sid=sales)))</pre>	<p>データベースに複数のインスタンスにまたがるサービスを含めることができるようになったため、SID のかわりにサービス名を指定するか、またはオプションでインスタンス名を指定します。</p> <p>次のエントリによって、クライアントは sales.com というサービスに接続できるようになります。</p> <pre>net_service_name=   (description=     (address=...)     (address=...)   (connect_data=     (service_name=sales.com)))</pre> <p>次のエントリによって、クライアントは op.com サービスの一部である op1 インスタンスに接続できるようになります。</p> <pre>net_service_name=   (description=     (address=...)     (address=...)   (connect_data=     (service_name=op.com)     (instance_name=op1)))</pre> <p>SERVICE_NAME には、グローバル・データベース名、データベース名とドメイン名で構成された名前の値が存在し、INSTANCE_NAME には、インストール中またはデータベースの作成中に入力された SID の値があります。</p> <p><b>関連項目：</b></p> <ul style="list-style-type: none"><li>■ 2-27 ページの「<a href="#">SID ではなくサービス名によるデータベースの識別</a>」を参照してください。</li><li>■ 6-2 ページの「<a href="#">接続記述子の理解</a>」を参照してください。</li></ul>



影響のある要素	Net8 リリース 8.0	Net8 リリース 8.1
tnsnames.ora ファイルの DESCRIPTION_LIST と ADDRESS_LIST を使用したクライアント・ロード・バランス	<p>複数の接続記述子からネット・サービス名へのマッピングと、複数リスナーに対するクライアント・ロード・バランスでは、DESCRIPTION_LIST が使用されていました。</p> <p>クライアント・ロード・バランスをとるには、ユーザーが各リスナーに対して接続記述子を個別に定義する必要がありました。</p>	<p>このリリースでも、クライアント・ロード・バランスは DESCRIPTION_LIST に対してデフォルトでオンに設定されていますが、新たに用意された LOAD_BALANCE パラメータを 1 つの ADDRESS_LIST に対して使用するか、または ADDRESS セットやセットの DESCRIPTION に LOAD_BALANCE パラメータを関連付けて明示的に指定することもできます。次のエントリによって、クライアント・ロード・バランスはアドレス・リストに関連付けられ、すべてのリスナーの ADDRESS に適用されます。</p> <pre>net_service_name= (description=   (load_balance=on)   (address=...)   (address=...)   (connect_data=     (service_name=sales.com)))</pre> <p>クライアント・ロード・バランスはどのバージョンのクライアントでも機能しますが、Oracle8i リリース 8.1 サーバーが必要です。</p> <p><b>関連項目：</b></p> <ul style="list-style-type: none"><li>■ 8-3 ページの「<a href="#">アドレス・リスト・パラメータの構成</a>」を参照してください。</li><li>■ C-33 ページの「<a href="#">クライアント・ロード・バランスと DESCRIPTION_LIST の変更された機能</a>」を参照してください。</li></ul>



影響のある要素	Net8 リリース 8.0	Net8 リリース 8.1
tnsnames.ora ファイルの ADDRESS_LIST による接続時フェイルオーバー	接続時フェイルオーバーは ADDRESS_LIST に含意されており、最初のリスナーでクライアント接続が失敗すると別のリスナーにフェイルオーバーします。Net8 と SQL*Net は、接続に成功するまですべてのリスナー・アドレスを試行します。	<p>フェイルオーバーは、ADDRESS_LIST、DESCRIPTION_LIST および DESCRIPTION セットに対してデフォルトでサポートされています。また、フェイルオーバーは、新たに用意された FAILOVER パラメータを ADDRESS セットに対して使用することにより明示的に指定することもできます。次のエントリによって、接続時フェイルオーバーはアドレス・リストに関連付けられ、すべてのリスナーの ADDRESS に適用されます。</p> <pre>net_service_name= (description=   (failover=on)   (address=...)   (address=...)   (connect_data=     (service_name=sales.com)))</pre> <p>接続時フェイルオーバーはどのバージョンのクライアントでも機能しますが、Oracle8i リリース 8.1 サーバーが必要です。</p> <p><b>関連項目：</b></p> <ul style="list-style-type: none"><li>■ 8-3 ページの「<a href="#">アドレス・リスト・パラメータの構成</a>」を参照してください。</li><li>■ C-31 ページの「<a href="#">ローカル・ネーミング・パラメータ (tnsnames.ora)</a>」を参照してください。</li></ul>



影響のある要素	Net8 リリース 8.0	Net8 リリース 8.1
tnsnames.ora ファイルに SOURCE_ROUTE=ON を設定し、Oracle Connection Manager を媒介とするルーティング接続	<p>Oracle Connection Manager を介して宛先サービスまでのアドレスのソース・ルートを作成する SOURCE_ROUTE パラメータは、ADDRESS_LIST の外側に記述する必要がありました。</p> <pre>service_name= (description=   (source_route=on) (address_list=   (address=...)   (address=...)) (connect_data=   (sid=sales)))</pre>	<p>SOURCE_ROUTE を ADDRESS_LIST の外側に記述する必要はなくなりました。かわりに、次に示すように ADDRESS リストに関連付けて記述します。</p> <pre>net_service_name= (description=   (source_route=on)   (address=...)   (address=...)   (connect_data=     (service_name=sales.com)))</pre> <p><b>関連項目：</b></p> <ul style="list-style-type: none"><li>■ 8-33 ページの「<a href="#">Oracle Connection Manager を使用する場合のクライアントの構成</a>」を参照してください。</li><li>■ C-31 ページの「<a href="#">ローカル・ネーミング・パラメータ (tnsnames.ora)</a>」を参照してください。</li></ul>
tnsnames.ora ファイルと listener.ora ファイルの ADDRESS_LIST 内にあるネストされたアドレス	<p>アドレス・リストが1つしかない場合でも、次に示すように ADDRESS_LIST を使用する必要がありました。</p> <pre>service_name= (description=   (address_list=     (address=...)     (address=...)) (connect_data=(sid=sales)))</pre>	<p>アドレス・リストが1つしかない場合、次に示すように Net8 のアドレス・リストを ADDRESS_LIST に埋め込む必要はありません。</p> <pre>net_service_name= (description=   (address=...)   (address=...)   (connect_data=     (service_name=sales.com)))</pre> <p>ネストされたアドレスはどのバージョンのクライアントでも機能しますが、Oracle8i データベースが必要です。</p> <p><b>関連項目：</b></p> <ul style="list-style-type: none"><li>■ 6-4 ページの「<a href="#">ローカル・ネーミング・メソッドの構成</a>」を参照してください。</li><li>■ C-31 ページの「<a href="#">ローカル・ネーミング・パラメータ (tnsnames.ora)</a>」を参照してください。</li></ul>



影響のある要素	Net8 リリース 8.0	Net8 リリース 8.1
listener.ora ファイルの SID_LIST_listener_name 情報	<p>listener.ora では、リスナーからサービスの提供を受けているデータベースの SID の定義が必要でした。データベースの SID は、listener.ora ファイルの SID_LIST セクションに指定する必要がありました。たとえば、次のように指定していました。</p> <pre>sid_list_listener=(sid_list=(sid_desc=(global_dbname=sales.com)(sid_name=db1)(oracle_home=/usr/bin/oracle)))</pre>	<p>データベースのインスタンスとマルチスレッド・サーバー・ディスパッチャは自動的にリスナーに登録されますので、次の場合を除いてデータベース情報を明示的に定義する必要はありません。</p> <ul style="list-style-type: none"><li>■ Oracle8 または Oracle7 のデータベースに接続する場合</li><li>■ この情報を必要とする Oracle Enterprise Manager のような管理ツールを使用する場合</li></ul> <p><b>関連項目：</b> 7-10 ページの「<a href="#">静的サービス情報の構成</a>」を参照してください。</p>
cman.ora ファイルの CMADMIN プロセスのアドレス	<p>Oracle Connection Manager ゲートウェイ・プロセスの CMGW のみがアドレスを必要としていました。管理プロセスの CMADMIN は、内部 IPC アドレスを使用します。</p> <pre>cman=(address=(protocol=tcp) (host=cman-pc) (port=1630))</pre>	<p>CMADMIN と CMGW の両方にプロトコル・アドレスが必要になりました。</p> <p>この新しいアドレスは、次に示すように CMAN_ADMIN パラメータを使用して記述します。</p> <pre>cman=(address=(protocol=tcp) (host=cman-pc) (port=1630)) cman_admin=(address=(protocol=tcp) (host=cman-pc) (port=1830))</pre> <p><b>関連項目：</b></p> <ul style="list-style-type: none"><li>■ 概念的な情報は、4-5 ページの「<a href="#">Oracle Connection Manager プロセス</a>」を参照してください。</li><li>■ 構成に関する情報については、8-25 ページの「<a href="#">接続集中化機能の使用</a>」を参照してください。</li></ul>



## tnsnames.ora の変更点

次の表は、Net8 リリース 8.1 で tnsnames.ora ファイルに追加された新しいパラメータを説明しています。

パラメータ	説明
FAILOVER	<p>ON に設定すると、最初のリスナーへの接続に失敗すると Net8 は別のリスナーにフェイルオーバーします。このパラメータによって、次に示すようにアドレスの試行回数が決まります。</p> <pre>net_service_name= (description=   (failover=on)   (address=...)   (address=...)   (connect_data=     (service_name=service_name)))</pre> <p>OFF に設定すると、Net8 はアドレスを 1 回試行します。</p> <p>デフォルトでは、このパラメータは ADDRESS_LIST、DESCRIPTION_LIST および DESCRIPTION セットに対して ON になっています。</p> <p><b>注意:</b> アクティブなセッションでのフェイルオーバーは、透過的アプリケーション・フェイルオーバー・パラメータを使用して構成できます。詳細は、8-11 ページの「<a href="#">透過的アプリケーション・フェイルオーバーの構成</a>」を参照してください。</p> <p><b>関連項目:</b></p> <ul style="list-style-type: none"><li>■ 8-3 ページの「<a href="#">アドレス・リスト・パラメータの構成</a>」を参照してください。</li><li>■ C-31 ページの「<a href="#">ローカル・ネーミング・パラメータ (tnsnames.ora)</a>」を参照してください。</li></ul>



パラメータ	説明
INSTANCE_NAME	<p>SERVICE_NAME パラメータの他に、INSTANCE_NAME パラメータを使用してアクセス先のデータベース・インスタンスを識別することもできます。</p> <p>INSTANCE_NAME は、接続先の特定インスタンスを識別することが重要な意味を持つ Oracle Parallel Server データベースのみに必要です。次の例に、op1 と op2 の複数のインスタンスにまたがったサービス op.us.acme.com の説明を示します。エンド・ユーザーは、どちらのインスタンスからでも op.us.acme.com サービスに接続できます。</p> <pre>net_service_name=   (description=     (address=...)     (connect_data=       (service_name=op.us.acme.com)       (instance_name=op1))) net_service_name=   (description=     (address=...)     (connect_data=       (service_name=op.us.acme.com)       (instance_name=op2)))</pre> <p>INSTANCE_NAME は、一般にインストール中またはデータベースの作成中に入力する SID です。</p> <p><b>関連項目：</b> インスタンス名文字列の詳細は、6-2 ページの「<a href="#">接続記述子の理解</a>」を参照してください。</p>

---



パラメータ	説明
LOAD_BALANCE	<p>ON に設定すると、リスナー・アドレスのリストを順不同に選択して、複数のリスナーに対するロード・バランスをとります。次に例を示します。</p> <pre>net_service_name= (description=   (load_balance=on)   (address=...)   (address=...)   (connect_data=     (service_name=sales.us.acme.com)))</pre> <p>OFF に設定すると、接続が成功するまでアドレスを順番に試行します。</p> <p>このパラメータは、デフォルトで DESCRIPTION_LIST に対して ON になっています。</p> <p><b>関連項目：</b></p> <ul style="list-style-type: none"><li>■ 8-3 ページの「<a href="#">アドレス・リスト・パラメータの構成</a>」を参照してください。</li><li>■ C-31 ページの「<a href="#">ローカル・ネーミング・パラメータ (tnsnames.ora)</a>」を参照してください。</li></ul>
SERVICE_NAME	<p>接続する Oracle8i データベースを識別します。次に例を示します。</p> <pre>net_service_name= (description=   (address=...)   (address=...)   (connect_data=     (service_name=sales.com)))</pre> <p>SERVICE_NAME は、一般にグローバル・データベース名です。グローバル・データベース名は、データベース名とドメイン名から構成され、インストール中またはデータベースの作成中に入力します。</p> <p><b>関連項目：</b> サービス名文字列の詳細は、6-2 ページの「<a href="#">接続記述子の理解</a>」を参照してください。</p>

## cman.ora の変更点

次の表は、Net8 リリース 8.1 で cman.ora ファイルに追加された新しいパラメータを説明しています。



パラメータ	説明
CMAN_ADMIN	<p>次に示すように、管理プロセスである CMADMIN プロセスのアドレスを識別します。</p> <pre>cman_admin=   (address=     (protocol=tcp)     (host= cman-pc)     (port=1650))</pre> <p>Oracle Connection Manager 制御ユーティリティの CMCTL が実行されているときに、CMADMIN プロトコル・アドレスを探してそのコマンドを実行します。</p> <p>デフォルトの TCP/IP アドレスであるポート 1830 を使用している場合、CMAN_ADMIN を明示的に指定する必要はありません。</p> <p><b>関連項目：</b></p> <ul style="list-style-type: none"><li>■ 8-25 ページの「<a href="#">接続集中化機能の使用</a>」を参照してください。</li><li>■ A-77 ページの「<a href="#">Oracle Connection Manager 制御ユーティリティ (CMCTL)</a>」を参照してください。</li></ul>
REMOTE_ADMIN	<p>Oracle Connection Manager へのリモート・アクセスを許可するかどうかを決定します。</p> <pre>cman_profile=   (parameter_list=     (remote_admin=yes))</pre> <p><b>関連項目：</b> A-77 ページの「<a href="#">Oracle Connection Manager 制御ユーティリティ (CMCTL)</a>」を参照してください。</p>



---

## Net8 の LDAP スキーマ

この付録では、Net8 オブジェクトのディレクトリに定義されている **Oracle スキーマ** のオブジェクト・クラスと属性について説明します。今後の機能に予定されているかまたは別の Oracle 製品に使用されているオブジェクト・クラスおよび属性については言及しません。

この付録は次の項で構成されています。

- 構造化オブジェクト・クラス
- 属性



## 構造化オブジェクト・クラス

Oracle スキーマは、Net8 の検索用に次の構造化オブジェクト・クラスをサポートしています。

- [orclDBServer](#)
- [orclNetService](#)
- [orclNetDescription](#)
- [orclNetDescriptionList](#)
- [orclNetAddress](#)
- [orclNetAddressList](#)

### orclDBServer

#### 説明

データベース・サービスのエントリの属性を定義します。

#### 属性

[orclNetDescName](#)

[orclVersion](#)

### orclNetService

#### 説明

ネット・サービス名エントリの属性を定義します。

#### 属性

[orclNetDescName](#)

[orclVersion](#)

### orclNetDescription

#### 説明

リスナー・プロトコル・アドレスおよびサーバーへの接続情報が含まれている接続記述子を指定します。



**属性**

- [orclNetAddrList](#)
- [orclNetInstanceName](#)
- [orclNetConnParamList](#)
- [orclNetFailover](#)
- [orclNetLoadBalance](#)
- [orclNetProtocolStack](#)
- [orclNetSdu](#)
- [orclNetServiceName](#)
- [orclNetSourceRoute](#)
- [orclSid](#)
- [orclVersion](#)

**orclNetDescriptionList****説明**

接続記述子のリストを指定します。

**属性**

- [orclNetDescList](#)
- [orclVersion](#)

**orclNetAddress****説明**

リスナー・プロトコル・アドレスを指定します。

**属性**

- [orclNetAddressString](#)
- [orclNetProtocol](#)
- [orclVersion](#)



orclNetAddressList

説明

アドレスのリストを指定します。

属性

- [orclNetAddrList](#)
- [orclNetFailover](#)
- [orclNetLoadBalance](#)
- [orclNetSourceRoute](#)
- [orclVersion](#)

属性

次の表に、オブジェクト・クラスに使用される属性のリストを示します。このリストは変更される場合があります。

属性	説明
orclNetAddrList	1 つ以上のリスナー・プロトコル・アドレスを識別します。
orclNetAddressString	リスナー・プロトコル・アドレスを定義します。
orclNetConnParamList	将来の接続データ・パラメータのプレース・ホルダー
orclNetDescList	1 つ以上の接続記述子を識別します。
orclNetDescName	接続記述子または接続記述子のリストを識別します。
orclNetFailover	接続時フェイルオーバーをアドレス・リストに対してオンにします。
orclNetInstanceName	アクセスするインスタンス名を指定します。
orclNetLoadBalance	クライアント・ロード・バランスをアドレス・リストに対してオンにします。
orclNetProtocol	orclAddressString 属性で使用されたプロトコルを識別します。
orclNetProtocolStack	Oracle8i Jserver への接続に関するプレゼンテーション・レイヤーとセッション・レイヤーの情報を識別します。
orclNetSdu	セッション・データ・ユニット (SDU) のサイズを指定します。
orclNetServiceName	Oracle8i データベース・サービス名を CONNECT_DATA 部分に指定します。
orclNetSourceRoute	目的に到達するまで各アドレスを順に使用するように Net8 を設定します。



属性	説明
orclSid	Oracle8 または Oracle7 の Oracle システム識別子 (SID) を接続記述子の CONNECT_DATA 部分に指定します。
orclVersion	エントリの作成に使用されたソフトウェアのバージョンを指定します。







## A

### ACL

「[アクセス制御リスト \(Access Control List: ACL\)](#)」を参照してください。

### API

「[Net8 Open](#)」を参照してください。

### ASCII キャラクタ・セット (ASCII character set)

American Standard Code for Information Interchange キャラクタ・セットの略。デジタル・データを使用して英数字情報を表現するための規則。IBM および IBM 互換コンピュータを除く大半のコンピュータで使用している照合順番。

## B

### Bequeath プロトコル

リスナーとサーバーが同じノード上に存在する場合、リスナーは、接続要求の受信時に専用サーバーを作成または起動できる。専用サーバーは1つのネットワーク・セッションのみにコミットされ、そのネットワーク・セッションを実行している間存在する。リスナーが専用サーバー・プロセスを作成し、そのプロセスにネットワーク・セッションの制御を渡す、または bequeath するときに発生するイベントの順序。

## C

### CDS

Cell ディレクトリ・サービス。「[セル・ディレクトリ・サービス \(Cell Directory Services: CDS\)](#)」を参照してください。



## cman.ora ファイル

着信要求のプロトコル・アドレス、および管理コマンド Oracle Connection Manager のパラメータ、**Net8 アクセス制御 (Net8 access control)** の規則を指定する構成ファイル。

## connect\_data

宛先データベースの**サービス名 (service name)** または **Oracle システム識別子 (Oracle System Identifier: SID)** を定義する**接続記述子 (connect descriptor)** の一部。次の例では、SERVICE\_NAME に sales.us.acme.com というサービスが定義されている。

```
(description=
  (address= (protocol=tcp) (host=sales-pc) (port=1521)
  (connect_data=
    (service_name=sales.us.acme.com)))
```

# D

## DNS

ドメイン名システム。「**ドメイン名システム (Domain Name System: DNS)**」を参照してください。

# G

## General Inter-Orb Protocol (GIOP)

Java オプション接続で使用し、メッセージ機能を提供する**プレゼンテーション・レイヤー (presentation layer)** ・タイプの 1 つ。

# I

## Inter-Orb Protocol (IIOP)

**Oracle8i JServer** へ接続するための TCP/IP または SSL 付き TCP/IP 上での **General Inter-Orb Protocol (GIOP)** の実装。

## IIOP クライアント (IIOP Clients)

**General Inter-Orb Protocol (GIOP)** を使用して Java オプションへアクセスするクライアント。次のようなクライアントを指す。

- Enterprise JavaBean (EJB)
- CORBA Servers
- Java ストアド・プロシージャ



## IP アドレス (IP address)

ネットワーク上のノードを識別するために使用する。ネットワーク上の各マシンには、ネットワーク ID と一意なホスト ID から構成される一意な IP アドレスが割り当てられる。このアドレスは、通常、各オクテットの 10 進値をピリオドで区切った値によって、小数点付き 10 進数表記で表現される (例: 144.45.9.22)。

## IPC

「[プロセス間通信 \(Interprocess Communication\)](#)」を参照してください。

# J

## Java Database Connectivity (JDBC) ドライバ (Java DataBase Connectivity Drivers)

Java プログラマに Oracle データベースへのアクセスを提供する JDBC ドライバ。

# L

## LDAP

Lightweight Directory Access Protocol。業界標準の[ディレクトリ・サーバー \(directory server\)](#)をサポートしている設計規則の枠組み。

## LDAP Data Interchange Format (LDIF)

LDAP コマンドライン・ユーティリティの入力ファイルの形式設定に関する一連の規格。

## ldap.ora ファイル

Net8 Configuration Assistant によって作成されるファイル。次のディレクトリ・アクセス情報が格納されている。

- ディレクトリのタイプ
- ディレクトリの場所
- クライアントまたはサーバーが、データベース・サービスへ接続するための接続識別子を検索または構成するために使用するデフォルトの管理コンテキスト。

ldap.ora ファイルは、UNIX プラットフォームの `$ORACLE_HOME/network/admin` および Windows プラットフォームの `ORACLE_HOME¥network¥admin` に置かれている。

## listener.ora ファイル

リスナーの構成ファイルで、次の内容を識別する。

- リスナーの名前
- 接続要求を受付けるプロトコル・アドレス。



- リスニング対象サービス。

listener.ora ファイルは、通常、UNIX プラットフォームの \$ORACLE\_HOME/network/admin および Windows NT の ORACLE\_HOME¥network¥admin に置かれている。

Oracle リリース 8.1 のデータベースでは、**サービス登録 (service registration)** により、データベースの識別を必要としない。ただし、Oracle Enterprise Manager の使用を計画する場合、Oracle リリース 8.1 データベースには静的サービス構成が必要である。

## LU6.2 プロトコル (LU6.2 protocol)

Logical Unit Type 6.20PC 上の Oracle アプリケーションを Oracle データベースと通信できるようにするプロトコル。この通信は、System Network Architecture (SNA) ネットワークを介して、Advanced Program-to-Program Communication (APPC) アーキテクチャをサポートするホスト・システム上の Oracle データベースとの間で行われる。

# M

## Microsoft の Active Directory

Windows 2000 Server に収録されている LDAP 準拠ディレクトリ・サービスの 1 つ。ネットワーク上のオブジェクトに関する情報を格納し、ユーザーおよびネットワーク管理者がこの情報を利用できるようにする。Active Directory により、ユーザーは、ネットワーク上のリソースにシングル・ログオン・プロセスでアクセスできるようになる。

Active Directory は、クライアントがアクセス可能なサービス情報を格納するディレクトリ・ネーミング・メソッドとして構成できる。

## MTS

「**マルチスレッド・サーバー (multi-threaded server: MTS)**」を参照してください。

# N

## names.ora ファイル

**Oracle Names サーバー**のパラメータ設定が格納されている構成ファイル。

## Named Pipes プロトコル

クライアント / サーバー (分散アプリケーション) 間でプロセス間通信を提供する高水準のインタフェースの 1 つ。Named Pipes を使用するネットワーク上でクライアント / サーバー対話が可能。



## NDS 認証

**認証方式 (authentication method)** の 1 つ。これにより、単一の NDS ディレクトリ・ツリーの下にある複数サーバーおよび複数データベース・ネットワークに対して、クライアントのシングル・ログイン・アクセスが可能になる。

## NDS

「**Novell ディレクトリ・サービス (Novell Directory Services: NDS)**」を参照してください。

## Net8

ネットワーク上でクライアント / サーバー通信とサーバー / サーバー通信の両方を可能にする Oracle のリモート・データ・アクセス・ソフトウェア。Net8 は分散処理と分散データベース機能をサポートする。Net8 は多数の通信プロトコルで動作し、それらを相互に接続する。Net8 は SQL\*Net バージョン 2 との後方互換性を有する。

## Net8 アクセス制御 (Net8 access control)

指定したサーバーに対するアクセスを、クライアントに許可または禁止するためのルールを設定する Oracle Connection Manager の機能。ファイアウォール・サポートとも呼ばれる。

## Net8 Configuration Assistant

インストール後に次の基本ネットワーク・コンポーネントを構成するためのツール。

- リスナー名およびプロトコル・アドレス
- クライアントが**接続識別子 (connect identifier)**の解決に使用するネーミング・メソッド
- tnsnames.ora ファイル内のネット・サービス名
- ディレクトリ・サーバー・アクセス

## Net8 Open

Net8 へのアプリケーション・プログラム・インタフェース (API)。これによって、プログラマは開発環境にすでに展開されている Net8 ネットワークを使用しながら、データベース・アプリケーションとデータベース以外のアプリケーションの両方を開発できる。Net8 Open は、すべての業界標準ネットワーク・プロトコルに対して単一の共通インタフェースをアプリケーションに提供する。

## Network Information Service (NIS)

Sun Microsystems 社の Yellow Pages (yp) クライアント / サーバー・プロトコル。ユーザ名やホスト名などのシステム構成データをネットワーク上のコンピュータ間で配布するためのプロトコルである。

## NI

ネットワーク・インタフェース。



## NIS

ネットワーク情報サービス「[Network Information Service \(NIS\)](#)」を参照してください。

## NN

ネットワーク・ネーミング (Oracle Names)。

## Novell ディレクトリ・サービス (Novell Directory Services: NDS)

ネットワーク・リソースに関する情報を格納する分散コンピューティング基盤の1つ。NDSにより、ユーザーの物理的位置や必要なリソースの位置に関係なく、容易なネットワーク・アクセスが実現できる。

NDSを使用すると、ネットワーク・ファイル・システム内の任意のネットワーク・リソースを表すオブジェクトによって、ネットワーク管理を簡易化できる。

## NPI

「[ネットワーク・プログラム・インタフェース \(Network Program Interface: NPI\)](#)」を参照してください。

## NR

ネットワーク・ルーティング。

## NS

ネットワーク・セッション。「[ネットワーク・セッション \(Network Session: NS\)](#)」を参照してください。

## NT

ネットワーク・トランスポート。「[トランスポート \(transport\)](#)」を参照してください。

# O

## OCI

Oracle コール・インタフェース。「[Oracle コール・インタフェース \(Oracle Call Interface : OCI\)](#)」を参照してください。

## OPI

「[Oracle プログラム・インタフェース \(Oracle Program Interface: OPI\)](#)」を参照してください。

## Open Systems Interconnection (OSI)

異機種間コンピュータ・ネットワーク・アーキテクチャの国際標準の枠組みとして、ISOによって開発されたネットワーク・アーキテクチャ・モデル。



OSI アーキテクチャは、次の7レイヤーに分割される。最下位から順に、1 物理レイヤー、2 データ・リンク・レイヤー、3 ネットワーク・レイヤー、4 トランスポート・レイヤー、5 セッション・レイヤー、6 プレゼンテーション・レイヤー、7 アプリケーション・レイヤーとなる。各レイヤーは、その直下のレイヤーを使用し、上位レイヤーにサービスを提供する。

### **Oracle Advanced Security**

企業のネットワークを保護して、インターネットに安全に拡張するためのセキュリティ機能の統合パッケージを提供する製品。Oracle Advanced Security は、ネットワーク暗号化と認証ソリューション、シングル・サインオン・サービスおよびセキュリティ・プロトコルの単一の統合源を提供する。業界標準を統合することにより、ネットワークに対する他に例のないセキュリティが得られる。

### **Oracle Connection Manager**

クライアント接続要求をその次のホップに送るか、または直接データベース・サーバーに送る際の経路となるルーター。Connection Manager を通じて接続要求を送るクライアントは、Connection Manager で構成される接続集中化または Net8 アクセス制御、複数プロトコル・サポートの機能を利用できる。

### **Oracle Connection Manager 制御ユーティリティ (Oracle Connection Manager Control Utility: CMCTL)**

Net8 に含まれているユーティリティ。Oracle Connection Manager の起動、停止および状態の取得などの様々な機能を制御する。

### **Oracle Internet Directory**

Oracle リリース 8.1 データベース上のアプリケーションとしてインプリメントされているディレクトリ・サービス。分散ユーザーおよびネットワーク・リソースについての情報を検索できる。これは、インターネット標準のオープン・ディレクトリ・アクセス・プロトコルである Lightweight Directory Access Protocol (LDAP) バージョン 3 に、Oracle8i Server の持つパフォーマンス、スケーラビリティ、堅牢性および可用性の高さを兼ね備えたものである。

### **Oracle Database Configuration Assistant**

データベースの作成、削除、変更を可能にするツール。

### **Oracle Names**

Oracle Names サーバーの 1 システムで構成されるディレクトリ・サービスで、ネットワーク上の各 Net8 サービスに対して名前からアドレスへの変換処理を提供する。

### **Oracle Names サーバー**

クライアントが長いアドレスではなく単純な名前で接続を要求できるように、Oracle Names を使用してサービスのネットワーク・アドレスを単純な名前と一緒に格納するサーバー。



## Oracle Names 制御ユーティリティ (Oracle Names Control Utility: NAMESCTL)

Oracle Names に収録されているユーティリティの 1 つ。Oracle Names サーバーの起動、停止、状態の取得など、Oracle Names サーバーの様々な機能を制御する。

## Oracle Net8 Assistant

グラフィカル・ユーザー・インタフェース・ツールの 1 つ。構成機能にコンポーネント制御機能を組み合わせることによって、Net8 の構成および管理のための統合環境を提供する。クライアントとサーバーのどちらでも使用できる。

Oracle Net8 Assistant を使用すると、次のネットワーク・コンポーネントを構成できる。

- ネーミング: **接続識別子 (connect identifier)** を定義し、それを **接続記述子 (connect descriptor)** にマップしてネットワーク位置およびサービスの識別情報を識別する。  
Oracle Net8 Assistant では、ローカルの tnsnames.ora ファイル、集中化された LDAP 準拠ディレクトリ・サービスまたは Oracle Names サーバーでの接続記述子の構成がサポートされる。
- ネーミング・メソッド: 接続識別子を接続記述子に解決する様々な方法を構成する。
- リスナー: リスナーの作成および構成を行って、クライアント接続を受け取る。

## Oracle Rdb

Digital 社の 64 ビット・プラットフォーム用データベース。Oracle Rdb には独自のリスナーがあるため、クライアントは Oracle7 と通信するのと同じように Oracle Rdb と通信できる。

## ORACLE\_HOME

ディレクトリベースのオペレーティング・システムで、Oracle ディレクトリ階層内の最高位ディレクトリに付けられた代替名。

## Oracle8i JServer

Java ストアド・プロシージャ、JDBC、SQLJ、Common Object Request Broker Architecture (CORBA)、Oracle8i データベースの Enterprise JavaBeans (EJB) をサポートする。

## Oracle コール・インタフェース (Oracle Call Interface : OCI)

アプリケーション・プログラミング・インタフェース (API) の 1 つ。これにより、第三世代言語に固有の手順やファンクション・コールを使用して、Oracle データベース・サーバーにアクセスし、SQL 文実行のすべての段階を制御するアプリケーションを作成できる。OCI では、C、C++、COBOL、FORTRAN など、多数の第三世代言語のデータ型、呼出し規則、構文、意味をサポートしている。

## Oracle コンテキスト

すべての Oracle ソフトウェア関連情報がその下位に保存される cn=OracleContext の **RDN**。**ディレクトリ・ネーミング (directory naming)** および **エンタープライズ・ユーザー (enterprise user)** セキュリティのエントリなどがある。



1つのディレクトリには、1つ以上の Oracle コンテキストが存在できる。Oracle コンテキストは、**ディレクトリ・ネーミング・コンテキスト (directory naming context)** と関連付けられている場合がある。

### Oracle システム識別子 (Oracle System Identifier: SID)

リリース 8.1 以前の Oracle データベースが動作しているときに、その特定のインスタンスを識別する名前。どのようなデータベースの場合も、データベースを参照しているインスタンスが少なくとも 1 つは存在する。

8.1 より古いリリースでは、SID はデータベースの識別に使用されている。SID は、**tnsnames.ora ファイル**における接続記述子の部分および **listener.ora ファイル**におけるリスナー定義の部分に含まれていた。

### Oracle スキーマ (Oracle schema: LDAP に関連する場合)

**ディレクトリ・サーバー (directory server)** に格納できる内容を決める一連の規則。Oracle には、Net8 エントリなど、Oracle エントリの多くのタイプに適用する独自のスキーマが存在する。Net8 エントリの Oracle スキーマには、そのエントリに登録される場合がある属性がある。

### Oracle プログラム・インタフェース (Oracle Program Interface: OPI)

**OCI** によって送信された各メッセージに対する応答を受け持つネットワークング・レイヤー。たとえば、OCI が 25 行のデータのフェッチを要求すると、OPI はフェッチした 25 行のデータを OCI に戻す。

### Oracle プロトコル (Oracle Protocol)

ネットワーク上でのデータ転送方法を定義する一連の規則。TCP/IP や SPX など、いくつかの業界標準トランスポート・プロトコルがある。

### OSI

Open System Interconnection。「**Open Systems Interconnection (OSI)**」を参照してください。

## P

### PMON プロセス (PMON process)

ユーザー・プロセスが異常終了したときにプロセス・リカバリを実行するプロセス・モニター・データベース・プロセス。PMON の役割は、キャッシュのクリーン・アップおよびプロセスが使用していたリソースの解放である。PMON は、ディスパッチャやサーバー・プロセスが異常終了すると、そのチェックおよび再起動も実行する。PMON は、**サービス登録 (service registration)** の一部として、リスナーにインスタンス情報を登録する。



## R

### RDBMS

リレーショナル・データベース管理システム。

### RDN

「[相対識別名 \(Relative Distinguished Name: RDN\)](#)」を参照してください。

### RPC

リモート・プロシージャ・コール。

## S

### Secure Sockets Layer (SSL)

ネットワーク接続のセキュリティ確保のために Netscape Communications Corporation によって設計された業界標準プロトコル。SSL では、公開鍵の構造基盤 (PKI) によって認証、暗号化およびデータ整合性が提供される。

### SID

Oracle システム識別子。「[Oracle システム識別子 \(Oracle System Identifier: SID\)](#)」を参照してください。

### SID\_LIST\_listener\_name

リスナーでサービスされるデータベースの [Oracle システム識別子 \(Oracle System Identifier: SID\)](#) を定義する、[listener.ora ファイル](#)のセクション。現在は、リリース 8.1 インスタンス情報が自動的にリスナーに登録されるため、このセクションはバージョン 7.x およびバージョン 8.0 の Oracle データベースのみで有効。静的な構成は、外部プロシージャや異機種間サービスなどのその他のサービス、および Oracle Enterprise Manager などの一部の管理ツールでも必要である。

### SPX プロトコル

Sequenced Packet Exchange プロトコル。多くの主要ネットワーク管理システム（特に Novell Advanced NetWare）に適合し、パフォーマンスの高いことで知られるプロトコル。

### SQL\*Net

Net8 の前身。Oracle サーバーと協力して、Oracle RDBMS または Oracle Tools (SQL\*Forms など) を走らせる複数のコンピュータが、ネットワークを介してデータを交換できるようにする Oracle 製品。SQL\*Net は分散処理と分散データベース機能をサポートする。SQL\*Net は多くの通信プロトコルで動作し、それらを相互に接続する。



## sqlnet.ora ファイル

クライアントまたはサーバーの構成ファイルで、次の内容を指定する。

- 非修飾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
- 名前の解決時にクライアントが使用するネーミング・メソッドの順序
- 使用する機能のロギングおよびトレース
- 接続のルート
- 優先 Oracle Names サーバー
- 外部ネーミング・パラメータ
- Oracle Advanced Security パラメータ

sqlnet.ora ファイルは、通常、UNIX プラットフォームの \$ORACLE\_HOME/network/admin および Windows プラットフォームの ORACLE\_HOME¥network¥admin に置かれている。

## SSL

Secure Sockets Layer。「[Secure Sockets Layer \(SSL\)](#)」を参照してください。

### SSL プロトコル付き TCP/IP (TCP/IP with SSL protocol)

Secure Sockets Layer 付き TCP/IP。クライアント上の Oracle アプリケーションに対して、リモートの Oracle データベースとの通信を、TCP/IP および [SSL](#) を介して可能にするプロトコル (TCP/IP および SSL を使用したネットワーク通信をサポートするホスト・システム上で Oracle データベースが動作している場合)。

### System Global Area (SGA)

データを格納し、Oracle [インスタンス \(instance\)](#) の情報を制御する共有メモリー構造のグループ。

## T

### TCP/IP プロトコル (TCP/IP protocol)

Transmission Control Protocol/Internet Protocol。事実上標準のイーサネット・プロトコル。ネットワークを介したクライアント / サーバー対話に使用する。

### Thin JDBC ドライバ (Thin JDBC Driver)

Thin JDBC ドライバは、Oracle の Type 4 ドライバで、Java アプレットおよび Java アプリケーションの開発者用に設計されている。JDBC ドライバは、Java ソケットを通して Oracle データベース・サーバーに直接接続を確立する。データベースへのアクセスは、[Net8 と Two-Task Common \(TTC\)](#) の実装に支援される。



## TNS

「[Transparent Network Substrate \(TNS\)](#)」を参照してください。

### tnsnames.ora ファイル

[接続記述子 \(connect descriptor\)](#) にマップされた [ネット・サービス名 \(net service name\)](#) が登録されている構成ファイル。このファイルの使用先は、[ローカル・ネーミング・メソッド](#) である。tnsnames.ora ファイルは、通常、UNIX プラットフォームの \$ORACLE\_HOME/network/admin および Windows プラットフォームの ORACLE\_HOME¥network¥admin に置かれている。

### Transparent Network Substrate (TNS)

Net8、Oracle Connection Manager および Oracle Names に組み込まれた、標準のネットワーク・トランスポート・プロトコルで機能する基本的なテクノロジー。

## TTC

Two-Task Common。「[Two-Task Common \(TTC\)](#)」を参照してください。

### Two-Task Common (TTC)

TTC は、クライアントとサーバーの異なるキャラクタ・セットや形式に対して、キャラクタ・セットとデータ型の変換を提供するために、標準的な Net8 接続で使用される [プレゼンテーション・レイヤー \(presentation layer\)](#)。

## U

### UPI

ユーザー・プログラム・インタフェース。

## W

### Windows NT のシステム固有な認証 (Windows NT native authentication)

[認証方式 \(authentication method\)](#) の 1 つ。これにより、Windows NT サーバーおよびそのサーバー上で動作しているデータベースに対して、クライアントのシングル・ログイン・アクセスが可能になる。

## あ

### アクセス制御 (access control)

「[Net8 アクセス制御 \(Net8 access control\)](#)」を参照してください。



### 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指示のグループ。この指示は、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）について、特定のデータに対するアクセス・レベルを許可する。

### アドレス (address)

リスナー・アドレス、Oracle Connection Manager、Oracle Names サーバーなどのネットワーク・オブジェクトを、一意に識別するためのネットワーク上の場所。アドレスには特定のフォーマットがあり、一意である必要がある。

### 異種サービス (Heterogeneous Services)

Oracle サーバーから非 Oracle サーバーをアクセスするための基本的な技術を提供する、統合されたコンポーネント。異機種間サービスを使用すると、次のことが可能となる。

- Oracle SQL を使用して、Oracle サーバーに存在するデータと同じように、Oracle 以外のシステムに格納されているデータに透過的にアクセスすること
- Oracle プロシージャ・コールを使用して、Oracle 分散環境から Oracle 以外のシステム、サービスまたはアプリケーション・プログラミング・インタフェース (API) に透過的にアクセスすること

### 位置の透過性 (location transparency)

データ表の置かれている位置が不明の場合でも、アプリケーションがそのデータ表にアクセスできるようにする分散データベースの特性。すべてのデータ・テーブルは1台のデータベース内にあるものと見なされ、システムは表名に基づいてデータの実際の位置を判別する。ユーザーは1つの文で複数のノード上のデータを参照できる。システムは、必要に応じて、SQL 文（の一部）を自動的かつ透過的にリモート・ノードに経路指定して実行する。ユーザーまたはアプリケーションに影響を与えないで、データをノード間で移動できる。

### 委任管理 (delegated administration)

**ルート管理リージョン (root administrative region)** の下にある1つ以上の管理リージョンにネットワーク管理を委任する Net8 ネットワーク。分散管理または非集中管理とも呼ばれる。「**集中管理 (central administration)**」と対比。

### 委任管理リージョン (delegated administrative region)

階層的に**ルート管理リージョン (root administrative region)** の下にあるリージョン。ルート管理リージョン以外のすべてのリージョン。

### インスタンス (instance)

データベースを参照している稼働中の Oracle8i ソフトウェアのインスタンス。データベースをデータベース・サーバー（コンピュータの種類には無関係）上で起動すると、Oracle は、**System Global Area (SGA)** と呼ばれるメモリー領域を割り当てて、1つ以上の Oracle プロセスを起動する。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ーおよびプロセスは、関連のデータベースのデータを効果的に管理して、デー



データベース・ユーザーにサービスを提供する。パラレル・サーバー・データベースでは、任意のインスタンスに接続できる。

### **エラー・メッセージ (error message)**

プログラムまたはコマンドの実行を妨げる問題または状態を知らせるコンピュータ・プログラムのメッセージ。

### **エンタープライズ・ユーザー (enterprise user)**

各エンタープライズ・ユーザーは、エンタープライズ全体にわたる一意な ID を持つ。エンタープライズ・ユーザーは、スキーマを介して個々のデータベースに接続する。エンタープライズ・ユーザーには、データベースに対する各自のアクセス権限を決定するエンタープライズ・ロールが割り当てられる。

### **エンタープライズ・ロール (enterprise role)**

エンタープライズ・ロールは、通常のデータベース・ロールと類似している。ただし、認可の範囲が複数のデータベースに及ぶ点異なる。エンタープライズ・ロールは、特定のデータベースでの権限を定義するロールのカテゴリである。エンタープライズ・ロールは、特定のデータベースのデータベース管理者によって作成される。エンタープライズ・ロールの許可または取消しは、1 人以上のエンタープライズ・ユーザーに対して可能である。これらのロールの許可または取消しに関する情報も、ディレクトリに格納される。データベース管理者は、グローバル・ロールを作成できるだけでなく、データベース・スキーマに対してロールまたは権限を許可することもできる。スキーマの共有者は、全員、エンタープライズ・ロール以外に、これらのローカル・ロールおよびローカル権限を取得する。

各エンタープライズ・ユーザーは、エンタープライズ全体にわたる一意な ID を持つ。エンタープライズ・ユーザーは、スキーマを介して個々のデータベースに接続する。エンタープライズ・ユーザーには、データベースに対する各自のアクセス権限を決定するエンタープライズ・ロールが割り当てられる。

### **エントリ (entry)**

ディレクトリの構成要素。ここには、ディレクトリ・ユーザーが参照するためのオブジェクトの情報が格納されている。

### **オブジェクト・クラス (object class)**

名前の付いた属性グループ。エントリに属性を割り当てるには、これらの属性を保持するオブジェクト・クラスをそのエントリに割り当てる。

同じオブジェクト・クラスにあるすべてのオブジェクトは、同じ属性を共有する。



# か

## 階層ネーミング・モデル (hierarchical naming model)

階層的に関連付けられた複数のドメインに名前を分割する Oracle Names の基盤構造。Oracle Names の場合、階層ネーミング・モデルは、集中管理または委任管理で使用できる。

## 外部ドメイン (foreign domains)

所定の管理リージョン内で管理されないドメイン。外部ドメインとリージョンの関係は相対的なもので、絶対的な意味の外部ドメインは存在しない。ネットワーク管理者は、一般的にネーム・サーバーのキャッシュ・パフォーマンスを最適化するために特定のリージョンに対する外部ドメインを定義する。

## 外部ネーミング (external naming)

NIS や NDS などの、サポートされているサードパーティー製のネーミング・サービスを使用する、**ネット・サービス名 (net service name)** 変換。

## 外部プロシージャ (external procedures)

PL/SQL コードから呼び出せる、第三世代言語 (3GL) で記述されたファンクションまたはプロシージャ。外部プロシージャとして、C のみがサポートされている。

## 仮想サーキット (virtual circuit)

**ディスパッチャ (dispatcher)** が使用する共有メモリーの 1 つで、クライアント・データベースの接続要求および応答を目的とする。ディスパッチャは、要求が到着すると、仮想サーキットを共通キューに置く。アイドル状態の共有サーバーは、共通キューから仮想サーキットを取り出して要求を処理し、共通キューから別の仮想サーキットを取り出そうとする前に、取り出した仮想サーキットを放棄する。

## 管理コンテキスト (administrative context)

**Oracle コンテキスト** がその下に存在するディレクトリ・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 (directory naming context)** の場合がある。クライアントは、ディレクトリ・アクセス構成時に、ディレクトリ構成ファイル (ldap.ora) 内の管理コンテキストで構成される。管理コンテキストによって、クライアントによる予想アクセス対象エントリの所属先ディレクトリにある Oracle コンテキストの位置が指定される。

## 管理リージョン (administrative region)

Net8 ネットワーク・コンポーネントを管理するための構造エンティティ。各管理リージョンには次の要素が含まれる。

- 1 つ以上のドメイン
- 1 つ以上の Oracle Names サーバー
- 1 つ以上のデータベースおよびリスナー



### キーワード値ペア (keyword-value pair)

接続記述子および多くの構成ファイルで標準的な情報単位として使用する、キーワードと値の組合せ。キーワードと値の対はネストすることができる。つまり、キーワードは別のキーワードと値の対を値として持つことができる。

### キャッシュ (cache)

前にアクセスされたデータが再び要求されたときに、その要求を素早く処理できるようにデータを格納しておくメモリー。

### 均一ネーミング・モデル (flat naming model)

1つのドメインのみが存在する Oracle Names 基盤構造。このドメイン内では、すべての名前が一意である必要がある。

### クライアント (client)

別のアプリケーションまたはコンピュータのサービス、データ、処理を要求するユーザー、ソフトウェア・アプリケーションまたはコンピュータ。2タスク環境では、クライアントはユーザー・プロセスとなる。ネットワーク環境では、クライアントはローカル・ユーザー・プロセスで、サーバーはローカルまたはリモートとなる。

### クライアント/サーバー・アーキテクチャ (client-server architecture)

2台のCPU間で処理を分割するソフトウェア・アーキテクチャ。1台のCPUは、トランザクションでサービスを要求して受け取るクライアントとして機能し、もう1台はトランザクションでサービスを提供するサーバーとして機能する。

### クライアント・プロファイル (client profile)

クライアントのプロパティ。[ネーミング・メソッド \(naming method\)](#) の優先順位、クライアントとサーバーの [ロギング \(logging\)](#) および [トレース \(tracing\)](#)、名前の要求元となるドメイン、[Oracle Names](#) および [Oracle Advanced Security](#) に対するその他のクライアント・オプションなどがある。

### クライアント・ロード・バランス (client load balancing)

ロード・バランス。複数のリスナーが1つのデータベースをサービスする場合、クライアントは接続要求を送るリスナーをランダムに選択できる。このランダム化によって、すべてのリスナーが着信接続要求の処理負担を共有できる。

### グローバル・データベース名 (global database name)

あるデータベースを他のデータベースから一意に識別するためのデータベースの完全な名前。グローバル・データベース名の形式は、`database_name.database_domain`。たとえば、`sales.us.acme.com`。

データベース名の `sales` は、単純な名前を表し、データベース・ドメインの `.us.acme.com` は、データベースが配置されているデータベース・ドメインを表し、グ



グローバル・データベース名を一意に識別する。可能であれば、データベース・ドメインにネットワーク・ドメインを反映させることをお勧めする。

グローバル・データベース名は、データベースのデフォルト・サービス名で、**初期化パラメータ・ファイル (initialization parameter file)** ファイルの SERVICE\_NAMES パラメータで指定される。

### グローバル・データベース・リンク (global database link)

ネットワーク内の各データベースを他のすべてのデータベースとリンクするデータベース・リンク。このリンクによって、ネットワーク内のどのデータベースを使用するユーザーも、SQL 文またはオブジェクト定義でグローバル・オブジェクト名を指定することができる。**Oracle Names サーバー**には、グローバル・データベース・リンクを格納できる。

### 構成ファイル

ネットワークのコンポーネントを識別および特徴付けるためのファイル。ネットワーク・コンポーネントをネーミングし、コンポーネント間の関係を識別する。

## さ

### サービス登録 (service registration)

**PMON プロセス (PMON process)** がリスナーに自動的に情報を登録する機能。この情報はリスナーに登録されるため、listener.ora ファイルをこの静的情報で構成する必要はない。

サービス登録では、次の情報がリスナーに登録される。

サービス登録によってリスナーに登録される情報を、次に示す。

- データベースの各稼動インスタンスのサービス名
- データベースのインスタンス名
- 各インスタンスで使用可能なサービス・ハンドラ（ディスパッチャおよび専用サーバー）

これにより、リスナーは、クライアントの要求を適切に送ることができる。

- ディスパッチャ、インスタンスおよびノードのロード情報

このロード情報により、リスナーは、クライアント接続の要求を最適に処理できるディスパッチャを判別できる。すべてのディスパッチャがブロックされている場合、リスナーは、この接続に対する専用サーバーを生成できる。

### サービス・ハンドラ (service handler)

サービス・ハンドラは、マルチスレッド・サーバー・**ディスパッチャ (dispatcher)**、**専用サーバー (dedicated server)**、**事前生成済専用サーバー (prespawned dedicated server)** のいずれかとなる。



### サービス名 (service name)

データベースの論理表現。データベースをクライアントに提供する方法。データベースは複数のサービスとして提供できる。また、サービスは複数のデータベース・インスタンスとしてインプリメントできる。サービス名は、インストール中またはデータベースの作成中に入力された**グローバル・データベース名 (global database name)**、すなわちデータベース名とドメイン名からなる名前を表す文字列となる。グローバル・データベース名が不明な場合は、初期化パラメータ・ファイルにある SERVICE\_NAMES パラメータの値からそのデータベース名を確認できる。

サービス名は、**接続記述子 (connect descriptor)** の **connect\_data** 部分に組み込まれている。

### サービス・レプリケーション (service replication)

ネットワーク上でディレクトリ・システムを完全にレプリケートするプロセス。新しいサービスは1つのネーム・サーバーのみに登録する必要がある。新しく登録されたサービスは、サービス・レプリケーション・プロセスによって、ネットワーク上のすべてのアクティブ・ネーム・サーバーに自動的にレプリケートされる。

### 識別名 (Distinguished Name: DN)

**ディレクトリ・サーバー (directory server)** 内のエントリの名前。DNによって、LDAPディレクトリ階層内でのエントリの位置が指定される。この方法は、ディレクトリ・パスによってファイルの位置が正確に指定されるのと類似している。

### システムまたはトポロジー・データ (system or topology data)

Oracle Names サーバーが通常の機能を制御したり、他の Oracle Names サーバーと通信する際に使用するデータ。交換およびルート・リージョンの Oracle Names サーバー、委任リージョンの Oracle Names サーバーなどを含む。

### 事前生成済専用サーバー (prespawnd dedicated server)

事前生成済専用サーバー・プロセスは、着信接続要求前に Net8 リスナーで事前に開始される。あるマシンで**マルチスレッド・サーバー (multi-threaded server: MTS)**を使用していないか、サポートしていない場合は、事前生成済専用サーバー・プロセスによってサーバー上での接続確立時間を短縮できる。また、他の接続のサーバー・プロセスを再利用することによって、割当て済のメモリー・システム・リソースを有効に利用する。

### 集中管理 (central administration)

ネットワーク管理が、ネットワーク全体に対して1つの**管理リージョン (administrative region)**で構成されている Oracle Names ネットワーク。集中管理では、すべてのネーム・サーバーがネットワーク内の他のネーム・サーバーおよびすべてのサービスを認識する。「**委任管理 (delegated administration)**」と対比。

### 初期化パラメータ・ファイル (initialization parameter file)

データベースとインスタンスを初期化する情報を含むファイル。



### セッション・レイヤー (session layer)

セッション・レイヤーは、**プレゼンテーション・レイヤー (presentation layer)** のエンティティに必要なサービスを提供する。エンティティでは、対話を構成および同期することができ、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 (Network Session: NS)** がある。

### セル・ディレクトリ・サービス (Cell Directory Services: CDS)

外部ネーミング方式の 1 つ。ユーザーは各種の Oracle ツールおよびアプリケーションから存在を意識しないで使用でき、これによって分散コンピューティング環境 (DCE) での Oracle8i データベースにアクセスできる。

### 接続 (connection)

ネットワーク上の 2 つのプロセス間で行われる対話。接続は、宛先 (サーバー) との接続を要求する起動側 (クライアント) によって開始される。

### 接続記述子 (connect descriptor)

ネットワーク接続の宛先を示す特殊なフォーマットの記述子。接続記述子には、宛先サービスおよびネットワーク・ルートの情報が登録されている。

宛先サービスを表示するには、Oracle リリース 8.1 のデータベースに対応するその**サービス名 (service name)** または Oracle リリース 8.0 またはバージョン 7 バージョンのデータベースに対応するその **Oracle システム識別子 (Oracle System Identifier: SID)** を使用する。ネットワーク・ルートを利用すると、最低でも、ネットワーク・アドレスによってリスナーの位置が明確になる。

### 接続先 (destination)

接続の終点であるクライアント。接続の起動側は、データまたはサービスを接続先に要求する。

### 接続識別子 (connect identifier)

**接続記述子 (connect descriptor)** に解決される **ネット・サービス名 (net service name)** または **サービス名 (service name)**。ユーザーは、ユーザー名およびパスワードを、接続識別子とともに、接続先サービスの接続文字列に渡すことによって接続要求を開始する。例を次に示す。

```
CONNECT username/password@connect_identifier
```

### 接続集中化 (connection concentration)

Oracle Connection Manager の機能で、サーバーのリソースを節約するために、クライアントから送られる複数の接続要求を統合して、サーバーと単一接続を確立する。



### 接続時フェイルオーバー

最初のリスナーが応答しない場合、クライアントの接続要求は他のリスナーに転送される。リスナーは接続の前にインスタンスが起動されているかどうかを確認するため、接続時フェイルオーバーは**サービス登録 (service registration)** によって有効になる。

### 接続ロード・バランス化 (connection load balancing)

同じサービスに対する様々なインスタンスやディスパッチャ間のアクティブな接続を、均等に分散する。これにより、リスナーはディスパッチャごとの接続数およびインスタンスが起動しているノードのロード量に基づいて、ルーティングを決定できる。

### 接続プーリング (connection pooling)

マルチスレッド・サーバーに対する物理ネットワーク接続数を最大にすることのできる、リソース利用およびユーザー拡張機能。

### 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 (net service name)** など、ユーザーが接続するサービスに渡す情報。例を次に示す。

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

### 接続要求 (connection request)

起動側が接続の開始を要求する際にリスナーに送信する通知。

### 専用サーバー (dedicated server)

各ユーザー・プロセスに対して専用のサーバー・プロセスを必要とするサーバー。クライアントごとに1つのサーバー・プロセスが存在する。Net8 は、既存のサーバー・プロセスのアドレスをクライアントに送り返す。続いて、クライアントは指定されたサーバー・アドレスに接続要求を再送する。「**マルチスレッド・サーバー (multi-threaded server: MTS)**」と対比。

### 相対識別名 (Relative Distinguished Name: RDN)

ローカルで、最も細分化されたレベルのエントリ名。この名前には、この名前以外に、このエントリを一意にアドレス指定する機能を持つ修飾エントリ名は存在しない。  
cn=sales,dc=us,dc=acme,dc=com という例では、cn=sales が RDN である。

### 属性 (attribute)

エントリのいくつかの側面を説明する情報の一つ。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 (object class)** に所属する。さらに、各属性にはタイプと値がある。タイプは属性の情報の種類を説明するものであり、値には実際のデータが格納されている。



# た

## 多重化 (multiplexing)

オペレーティング・システムのリソースを節約するために、複数のセッションを結合して1つのトランスポート接続で転送すること。「[接続集中化 \(connection concentration\)](#)」も参照してください。

## データ・パケット (data packet)

「[パケット \(packet\)](#)」を参照してください。

## データベース管理者 (database administrator: DBA)

(1) Oracle Server 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、この2つは同じ人物を表す。多くのサイトでは複数の DBA が配置される。

## データベース・リンク

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、オプションでユーザー名とパスワードを識別するために、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。定義されると、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。

「[グローバル・データベース・リンク \(global database link\)](#)」、「[プライベート・データベース・リンク \(private database link\)](#)」および「[パブリック・データベース・リンク \(public database link\)](#)」も参照してください。

## ディスパッチャ (dispatcher)

専用サーバー・プロセスをクライアントごとに必要とせずに、多数のクライアントが同じサーバーに接続できるようにするプロセス。ディスパッチャは、複数の着信ネットワーク・セッション要求を処理して、共有サーバー・プロセスに送る。「[マルチスレッド・サーバー \(multi-threaded server: MTS\)](#)」も参照してください。

## ディレクトリ・サーバー (directory server)

Lightweight Directory Access Protocol (LDAP) でアクセスする LDAP 準拠のディレクトリ・サーバー。ディレクトリは、データベース・ネットワーク・コンポーネント、ユーザー・ポリシーおよび企業ポリシーの優先順位、ユーザー認証およびセキュリティ情報の集



中の格納および検索を実現できるため、クライアント側およびサーバー側に局在化しているファイルを置き換えることができる。

### ディレクトリ情報ツリー (Directory Information Tree: DIT)

木の形に似た階層構造で、対象のエントリの**識別名 (Distinguished Name: DN)** の「**ディレクトリ・サーバー (directory server)**」に格納されている。

### ディレクトリ・ネーミング (directory naming)

中央ディレクトリ・サーバーに格納されているデータベース・サービスまたはネット・サービス名を接続記述子に解決する**ネーミング・メソッド (naming method)**。

ディレクトリによって、データベース・サービスおよびネット・サービス名の集中管理が実現でき、サービスの追加または再配置に伴う作業負荷が軽減できる。ネット・サービス名は別名サービスに構成可能だが、ディレクトリはネット・サービス名を使用せず直接データベース・サービスを参照できる。構成をさらに簡易化するため、データベース・サービスは、インストール時にエントリとして自動的にディレクトリに追加される。

### 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・サーバー内で重要となるサブツリー。これは、通常、いくつかの組織サブツリーの最上位に位置する。ディレクトリの中には、固定のコンテキストのみが可能なものがある。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可能である。

### デフォルト・ドメイン (default domain)

大半のクライアント要求が行われる**ドメイン (domain)**。クライアントが常駐しているドメイン、またはクライアントがネットワーク・サービスを頻繁に要求するドメイン。デフォルト・ドメインは、未修飾ネットワーク名の要求に追加すべきドメインを決定するクライアント構成パラメータでもある。名前の要求に文字 "." が入っていない場合は、未修飾の要求。

### 透過的アプリケーション・フェイルオーバー (Transparent Application Failover: TAF)

Oracle Parallel Server や Oracle Fail Safe など、可用性の高い環境を目的としたランタイム・フェイルオーバー。これは、アプリケーションとサービス間接続のフェイルオーバーおよび再確立を参照する。これを利用すると、クライアント・アプリケーションは、接続が失敗した場合データベースに自動的に再接続できる。また、オプションで進行中の SELECT 文を再開できる。この再接続は、Oracle Call Interface (OCI) ライブラリの内部から自動的に行われる。

### ドメイン (domain)

**ドメイン名システム (Domain Name System: DNS)** 名前空間内の任意のツリーまたはサブツリー。ドメインが最もよく使用されるのは、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参照するときである。



## ドメイン名システム (Domain Name System: DNS)

ドメイン名システム (DNS) は、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 (domain)** の階層に編成されている。DNS は、ユーザー・フレンドリな名前でもコンピュータを検索するために 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 (IP address)** に解決する。

## ドメスティック・ドメイン (domestic domains)

所定の管理リージョン内で管理されるドメイン群。ドメスティック・ドメインとリージョンの関係は相対的なもので、絶対的な意味のドメスティック・ドメインは存在しない。ローカル・ドメインとも呼ばれる。

## トランスポート (transport)

データ・フロー制御およびエラー・リカバリ・メソッドによってエンド・エンド間の信頼性を維持するネットワーキング・レイヤー。Net8 では、**Oracle プロトコル (Oracle Protocol)** をトランスポート・レイヤーに使用する。

## トレース (tracing)

操作に関する詳細情報を出力ファイルに書き込む機能。トレース機能によって、操作が実行されるときイベントを示す詳細な文が生成される。管理者はトレース機能を使用して、異常な状態を診断する。通常、トレース機能はオンになっていない。

「**ロギング (logging)**」も参照してください。

な

## 認証方式 (authentication method)

分散環境におけるユーザー、クライアントおよびサーバーの認証に高い信頼性を与えるセキュリティ方式。ネットワーク認証方式を利用すると、ユーザーはシングル・サインオンの恩恵を受けることもできる。次に示す認証方式のサポートは、**Oracle Advanced Security** のインストールの有無に左右される。

- CyberSafe
- SecurID
- RADIUS
- Identix
- Kerberos
- **SSL**
- **Windows NT のシステム固有な認証 (Windows NT native authentication)**



## ネーミング・メソッド (naming method)

データベース・サービスに対する接続試行時に、クライアント・アプリケーションが**接続識別子 (connect identifier)** を **接続記述子 (connect descriptor)** に解決するために使用する方法。Net8 によって提供されるネーミング・メソッドは次の 5 種類。

- ローカル・ネーミング
- ディレクトリ・ネーミング (directory naming)
- ホスト・ネーミング (host naming)
- Oracle Names
- 外部ネーミング (external naming)

## ネーミング・モデル (naming model)

名前を割り当てることができる一連のドメインと構造。

均一ネーミング・モデルでは単一のドメイン。

階層ネーミング・モデルでは、最高位のドメインがルート・ドメインで、その他のドメインはすべて階層的に関連付けられる。

## ネット・サービス名 (net service name)

**接続記述子 (connect descriptor)** に解決されるサービスの単純な名前。ユーザーは、ユーザー名およびパスワードを、ネット・サービス名とともに、接続先サービスの接続文字列に渡すことによって接続要求を開始する。例を次に示す。

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

ネット・サービス名は、必要に応じて、次のような様々な場所に格納できる。

- 各クライアントのローカル構成ファイル `tnsnames.ora`
- ディレクトリ・サーバー
- Oracle Names サーバー
- 外部ネーミング・サービス。たとえば、**NDS**、**NIS** または **CDS**

## ネットワーク (network)

ハードウェアとソフトウェアによってリンクされ、データまたは周辺装置（あるいは、その両方）を共有できる複数のコンピュータで構成されているグループ。

## ネットワーク・インタフェース (Network Interface: NI)

Oracle のクライアント、サーバーまたは外部プロセスが Net8 の機能にアクセスするための汎用インタフェースを提供するネットワーク・レイヤー。接続のブレイク要求とリセット要求を処理する。



### ネットワーク・オブジェクト (network object)

ネットワーク上でアドレスを直接指定できるサービス。リスナーや Oracle Names サーバーなど。

### ネットワーク管理者 (network administrator)

ネットワーク・コンポーネントのインストール、構成、テストといったネットワーク管理作業を実行する人。ネットワーク管理者は、構成ファイル、接続記述子、サービス名、別名、パブリック・データベース・リンク、グローバル・データベース・リンクなどを管理する。

### ネットワーク・キャラクタ・セット (network character set)

Oracle の定義では、キーワードと値の対（つまり、接続記述子と構成ファイル）で値として使用できる文字のセット。文字セットには、大 / 小文字の英数字およびいくつかの特殊文字が含まれる。

### ネットワーク・サービス (network service)

Oracle Application Network では、サービス利用者に対していくつかのタスクが実行される。たとえば、ネーム・サーバーは名前解決サービスをクライアントに提供する。

### ネットワーク・セッション (Network Session: NS)

クライアント・アプリケーションとサーバー間で接続を確立および維持するために、標準的な Net8 接続で使用される[セッション・レイヤー \(session layer\)](#)。

### ネットワーク・プログラム・インタフェース (Network Program Interface: NPI)

サーバー / サーバー間対話用インタフェースの 1 つ。[OCI](#) がクライアントに対して行うすべての機能を実行して、調整サーバーが追加サーバーに SQL 要求を作成できるようにする。

### ネットワーク・プロトコル (network protocol)

「[Oracle プロトコル \(Oracle Protocol\)](#)」を参照してください。

### ネットワーク・リスナー (network listener)

「[リスナー \(listener\)](#)」を参照してください。

### ノード (node)

ネットワークの一部を構成するコンピュータまたは端末。

は

### パケット (packet)

接続またはデータ転送が要求されるたびに、ネットワーク上で送信される情報のブロック。パケットに収容される情報は、接続、受取り、リダイレクト、データなど、パケットの種類によって異なる。パケット情報はトラブルシューティングに役立つことがある。



## パスワード (password)

データ・セキュリティに使用し、所有者のみが認識できる文字列 (単語または句)。オペレーティング・システムまたはソフトウェア・アプリケーション (Oracle データベースなど) と接続する場合に、オペレーティング・システム・ログイン ID または Oracle ユーザー名、アカウント名などと一緒にパスワードを入力する。ユーザー名または ID は公開されているが、秘密のパスワードによって、ユーザー名の所有者のみがその名前を使用したり、そのデータにアクセスすることができる。

## パブリック・データベース・リンク (public database link)

DBA が作成するすべてのユーザーがアクセスできるローカル・データベース上のデータベース・リンク。

「[データベース・リンク](#)」、「[グローバル・データベース・リンク \(global database link\)](#)」および「[プライベート・データベース・リンク \(private database link\)](#)」も参照してください。

## パラメータ (parameter)

プログラム、コマンドまたは関数に渡す情報。ファイル指定、キーワード、定数値などがある。

## 非集中管理 (decentral administration)

「[委任管理 \(delegated administration\)](#)」を参照してください。

## ファイアウォール・サポート (firewall support)

「[Net8 アクセス制御 \(Net8 access control\)](#)」を参照してください。

## フェイルオーバー (failover)

「[接続時フェイルオーバー](#)」を参照してください。

## 複数プロトコル・サポート (multi-protocol support)

異なるネットワーク・プロトコルを持つクライアントとサーバーが相互に通信できるようにする、Oracle Connection Manager の機能。この機能は、Oracle Multi-Protocol Interchange として SQL\*Net バージョン 2 ですでに実現している機能の置換え。

## プライベート・データベース・リンク (private database link)

あるユーザーが専用に使用するデータベース・リンク。

「[データベース・リンク](#)」、「[グローバル・データベース・リンク \(global database link\)](#)」および「[パブリック・データベース・リンク \(public database link\)](#)」も参照してください。



## プレゼンテーション・レイヤー (presentation layer)

通信中にアプリケーション・レイヤー・エンティティが通信または参照する情報の表現を管理する。セッション・レイヤーの例は、[Two-Task Common \(TTC\)](#) と [General Inter-Orb Protocol \(GIOP\)](#)。

## プロセス間通信 (Interprocess Communication)

リスナーと同じノードに存在するクライアント・アプリケーションが、データベースと通信するために使用するプロトコルの1つ。

## プロトコル・アドレス (protocol address)

ネットワーク・オブジェクトのネットワーク・アドレスを識別するアドレス。

接続が確立されると、要求の受信者 ([リスナー \(listener\)](#)、[Oracle Names サーバー](#)、[Oracle Connection Manager](#) など) およびクライアントは同じプロトコル・アドレスで構成される。クライアントは、このアドレスを使用して接続要求を特定のネットワーク・オブジェクト位置に送信し、レスピエントはこのアドレスで要求のリスニングを行う。クライアントおよび接続レスピエントにとって、同じプロトコルをインストールするのと同様に、同じアドレスを構成することが重要である。

## プロトコル・スタック (protocol stack)

特定の [プレゼンテーション・レイヤー \(presentation layer\)](#) と [セッション・レイヤー \(session layer\)](#) の組合せを示す。

## プロファイル (profile)

クライアントまたはサーバー上での Net8 機能の使用可能化および構成の優先順位を指定するパラメータの集合。プロファイルの格納および実装は、`sqlnet.ora` ファイルを介して行う。

## 分散管理 (distributed administration)

「[委任管理 \(delegated administration\)](#)」を参照してください。

## 分散処理 (distributed processing)

フロントエンド処理とバックエンド処理を別々のコンピュータで行うこと。Net8 は、リモート・データベースに透過的に接続することにより分散処理をサポートする。

## 別名 (alias)

既存のネットワーク・オブジェクトに対する代替名。別名は最初のネットワーク・オブジェクトと同じ名前に変換される。Oracle Names サーバーには、定義済のネット・サービス名、データベース・サーバーまたはデータベース・リンクの別名が格納される。

## ホスト・ネーミング (host naming)

[ネット・サービス名 \(net service name\)](#) 解決により、TCP/IP 環境内のユーザーは既存の名前解決サービスを通して、ネット・サービス名を変換できる。この名前解決サービスには、



## ドメイン名システム (Domain Name System: DNS)、Network Information Service (NIS)

または単に中央でメンテナンスされる /etc/hosts ファイル・セットの場合がある。ホスト・ネーミングでは、サーバーのホスト名またはホスト名の別名を使用するのみで、ユーザーは Oracle サーバーに接続できる。この機能を利用するために、クライアントを構成する必要はない。この方法は、単純な TCP/IP 環境にお薦めする。

## ま

### マップ (map)

**Network Information Service (NIS)** ypserv プログラムが名前要求を処理するために使用するファイル。

### マルチスレッド・サーバー (multi-threaded server: MTS)

より多くのユーザーをサポートするために、多数のユーザー・プロセスがごくわずかなサーバー・プロセスを共有できるように構成されたサーバー。MTS の構成により、多数のユーザー・プロセスが**ディスパッチャ (dispatcher)**に接続する。ディスパッチャは、複数の着信ネットワーク・セッション要求を共通キューに送る。複数のサーバー・プロセスの共有プールの中で、あるアイドル状態の共有サーバー・プロセスが共通キューから要求を取り出す。これは、サーバー・プロセスの小規模プールが大量のクライアントを処理できることを意味する。「**専用サーバー (dedicated server)**」と対比。

## や

### ユーザー名 (user name)

ユーザーを Oracle サーバーおよび他のユーザーが識別するための名前。すべてのユーザー名にパスワードが関連付けられており、Oracle データベースに接続するときはユーザー名とパスワードを両方入力する必要がある。

### 優先 Oracle Names サーバー (Preferred Oracle Names Server)

名前を解決するときにクライアントが優先的に使用する Oracle Names サーバー。通常は、クライアントと物理的に最も近い Oracle Names サーバー、または最も安価なネットワーク・リンクで利用できる Oracle Names サーバー。

### 予約済 Oracle Names サーバー (well-known Oracle Names サーバー)

1 つ以上の Oracle Names サーバーのアドレスが、Oracle Names サーバーとそのクライアントの両方にハードコード化される。予約済アドレスを持つ Oracle Names サーバーにはそのアドレスでアクセスできるため、クライアントに構成ファイルでサーバーを検索するように指示する必要がない。



**リージョン (region)**

「[管理リージョン \(administrative region\)](#)」を参照してください。

**リージョン・データベース (region database)**

Oracle Names 情報を格納する Oracle データベースの表。

**リスナー (listener)**

着信クライアント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、サーバー上の個別のプロセス。

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）はネットワーク・セッションをサーバーに要求し、リスナーが実際の要求を受け取る。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。

**リスナー制御ユーティリティ (Listener Control Utility: LSNRCTL)**

Net8 に含まれているユーティリティ。リスナーの起動および停止、状態の取得などの様々な機能を制御する。

**リンク修飾子 (link qualifier)**

グローバル・データベース・リンクに付加される修飾子で、データベースのユーザー名とパスワードからなる資格証明の代替設定を提供する。たとえば、emp というリンク修飾子を sales.us.acme.com のグローバル・データベース・リンクに付加できる。

**ルート管理リージョン (root administrative region)**

分散環境内で最高位の管理リージョン。ルート管理リージョンはルート・ドメインを含む。

**ルート・ドメイン (root domain)**

階層ネーミング・モデル内で最高位のドメイン。

**ループバック・テスト (loopback test)**

サーバーからサーバー自体に戻す接続。ループバックが正常に終了すると、Net8 がサーバー側で機能していることが確認できる。

**ローカル・ネーミング**

**ネット・サービス名 (net service name)** 変換。各クライアントの [tnsnames.ora ファイル](#) に構成および格納されている情報を使用して、ネットワーク・アドレスを検索する。ローカル・ネーミングは、あまり変更されることのない少数のサービスからなる単純な分散ネットワークに最適な方法である。



### ロード・バランシング (load balancing)

クライアント接続を複数のリスナー、ディスパッチャ、インスタンスおよびノードに均等に分散し、1つのコンポーネントに負荷がかかりすぎないようにする機能。

Net8 は、**クライアント・ロード・バランス (client load balancing)** および **接続時ロード・バランス化 (connection load balancing)** をサポートしている。

### ロギング (logging)

エラーおよびサービス・アクティビティ、統計をログ・ファイルに書き込む機能。ログ・ファイルは、画面に表示されるエラー・メッセージで障害を識別できない場合に、管理者用の追加情報として使用できる。エラー・スタックとしてのログ・ファイルは、各種レイヤーでのソフトウェアの状態を示す。

「**トレース (tracing)**」も参照してください。



# 索引

## 記号

" 記号

予約済、構成ファイル, C-3

# 記号

予約済、構成ファイル, C-3

' 記号

予約済、構成ファイル, C-3

() 記号

予約済、構成ファイル, C-3

= 記号

予約済、構成ファイル, C-3

/ (スラッシュ) 記号

予約済、構成ファイル, C-3

## 数字

1024 ポート, B-5

1521 ポート, 6-47, 7-6, B-4

使用していない場合, 7-8

1575 ポート, 4-20, 6-51, B-4, C-55

1630 ポート, 4-5, 8-26, 8-29, 8-31, B-4

1646 ポート, C-23

1830 ポート, 8-26, 8-29, B-4, C-80

2481 ポート, B-4

2482 ポート, B-4

2483 ポート, B-4

## A

ACCEPT\_CONNECTIONS コマンド, A-78

ACTIVE\_RELAYS, A-92

ACT パラメータ, 8-32, C-72

ADDRESS\_LIST パラメータ, B-2, C-37, C-46

ADDRESS (ADD または ADDR) 属性, 9-4

ADDRESS パラメータ, B-2, C-37, C-46

Advanced Program-to-Program Communication (APPC), 4-3

ANSWER\_TIMEOUT パラメータ, C-69

APPC アーキテクチャ, 4-3

ARGS パラメータ, B-3

ARGV0 パラメータ, B-3

A.SMD レコード, 7-10, 11-12, A-42

「Attempts Per Names Server」フィールド, 6-58

AUTHENTICATION\_LEVEL パラメータ, C-69

AUTOMATIC\_IPC パラメータ, 8-19, C-75

## B

BACKUP パラメータ, 8-11, C-42

bequeath, 2-17

イベント・プログラミング, 10-10

説明, 4-4

BEQUEATH\_DETACH パラメータ, 8-21, 10-10, C-4

bequeathed セッション, 2-17

Bequeath プロトコル

ARGS パラメータ, B-3

ARGV0 パラメータ, B-3

PROGRAM パラメータ, B-3

## C

CDS 外部ネーミング, 2-33, 4-21, 6-71, 8-18

Cell ディレクトリ・サービス外部ネーミング, 2-33, 4-21, 8-18

構成, 6-71

CHANGE\_PASSWORD コマンド, A-3

ckcfg.ora ファイル, 6-44

ckpcch.ora ファイル, 6-45, 11-4

ckpreg.ora ファイル, 6-45



- 「Client Registration ID」フィールド, 8-21
- client32net.cfg ファイル, 6-78
- CLOSE\_RELAY コマンド, A-79
- cmadm\_pid.log ファイル, 12-16, 12-24
- cmadm\_pid.trc ファイル, 12-29
- CMADMIN プロセス, 4-5
- CMAN\_ADMIN パラメータ, C-66, C-80, C-84
- cmn\_pid.log ファイル, 12-16, 12-24
- cmn\_pid.trc ファイル, 12-29
- CMAN\_PROFILE パラメータ, C-66
- CMAN\_RULES パラメータ, 8-32, C-72
- cmn.ora ファイル
  - 構成パラメータのリファレンス, C-65, C-72
  - 作成, 8-26, 8-29, 8-31
  - パラメータ
    - ACT, C-72
    - ACT パラメータ, 8-32
    - ANSWER\_TIMEOUT, C-69
    - AUTHENTICATION\_LEVEL, C-69
    - CMAN, C-66
    - CMAN\_ADMIN, C-66, C-80, C-84
    - CMAN\_PROFILE, C-66
    - CMAN\_RULES, C-72
    - DST, C-72
    - DST パラメータ, 8-32
    - LOG\_LEVEL, C-69
    - MAX\_FREELIST\_BUFFERS, C-69
    - MAXIMUM\_CONNECT\_DATA, C-69
    - MAXIMUM\_RELAYS, C-69
    - RELAY\_STATISTICS, C-70
    - REMOTE\_ADMIN, C-70, C-84
    - SHOW\_TNS\_INFO, C-70
    - SRC, 8-32, C-72
    - SRV, 8-32, C-72
    - TRACE\_DIRECTORY, C-70
    - TRACE\_FILELEN, C-71
    - TRACE\_TIMESTAMP, C-71
    - TRACING, C-70
    - USE\_ASYNC\_CALL, C-71
- リリース 8.1、変更, C-83, C-84
- CMAN パラメータ, C-66
- CMCTL STATS コマンド
  - ACTIVE\_RELAYS, A-92
  - MOST\_RELAYS, A-92
  - OUT\_OF\_RELAY, A-92
  - TOTAL\_REFUSED, A-92
  - TOTAL\_RELAYS, A-92

- CMCTL ユーティリティ
  - SET コマンド, A-78
  - 機能と構文書式, A-77
  - コマンド
    - ACCEPT\_CONNECTIONS, A-78
    - CLOSE\_RELAY, A-79
    - EXIT, A-79
    - HELP, A-80
    - QUIT, A-80
    - SET, A-81
    - SET AUTHENTICATION\_LEVEL, A-81
    - SET DISPLAYMODE, A-82
    - SET LOG\_LEVEL, A-82
    - SET RELAY\_STATISTICS, A-83
    - SHOW, A-78, A-84
    - SHOW ADDRESS, A-84
    - SHOW ALL, A-85
    - SHOW DISPLAYMODE, A-86
    - SHOW PROFILE, A-87
    - SHOW RELAY, A-87
    - SHOW RULES, A-88
    - SHUTDOWN, A-78, A-89
    - START, 11-7, A-90
    - STATS, A-91
    - STATUS, A-93
    - STOP, A-94
    - STOPNOW, A-94
    - VERSION, A-95
  - コマンドのリファレンス, A-78 ~ A-95
  - シャットダウン、推奨方法, A-78
  - 使用, 11-3
  - 説明, 4-25
  - ネットワークのテスト, 11-3
- CMGW プロセス, 4-5
- CNAME.SMD レコード, 11-12, A-42
- cn 属性, 2-36
- 「Configure Prespawned Servers」ダイアログ・ボックス, 7-15
- CONNECT\_DATA パラメータ, C-40
- CONNECT\_TIMEOUT\_listener\_name パラメータ, C-51
- CONNECTIONS (CON または CONN) 属性, 9-4
- CyberSAFE 認証方式
  - 「Oracle Advanced Security」を参照
- c 属性, 2-36



## D

---

DAEMON.TRACE\_DIRECTORY パラメータ, C-5  
DAEMON.TRACE\_LEVEL パラメータ, C-5  
DAEMON.TRACE\_MASK パラメータ, C-5  
DBAccess Interface, 3-12  
DBSNMP\_START コマンド, A-4  
DBSNMP\_STATUS コマンド, A-5  
DBSNMP\_STOP コマンド, A-5  
DCE インテグレーション、説明, 4-23  
dc 属性, 2-36  
「Default Domain」 フィールド, 6-58  
DEFAULT\_ADMIN\_CONTEXT パラメータ, C-74  
DELAY パラメータ, 8-12, C-42  
DELEGATE\_DOMAIN コマンド, 6-67, A-27  
DESCRIPTION\_LIST パラメータ, C-37  
DESCRIPTION パラメータ, 6-5, C-36, C-46  
Direct Hand-Off, xvi  
    イベントの listener.log 格納, 12-23  
    説明, 2-8  
「Directory Server Migration Wizard」, 6-29  
DIRECTORY\_SERVER\_TYPE パラメータ, C-74  
DIRECTORY\_SERVERS パラメータ, C-74  
DISABLE\_OOB パラメータ, 8-22, C-6  
DISPATCHERS (DIS または DISP) 属性, 9-4  
DLCR.RDBMS.OMD レコード, 11-12, A-42  
DL.RDBMS.OMD レコード, 11-12, A-42  
DNS、説明、用語集 -23  
Domain Name Service、説明、用語集 -23  
DOMAIN\_HINT コマンド, A-28  
DST パラメータ, 8-32, C-72  
DUMP\_LDAP コマンド, A-29  
DUMP\_TNSNAMES コマンド, A-33

## E

---

EXIT コマンド  
    CMCTL, A-79  
    LSNRCTL, A-5  
    NAMESCTL, A-33  
extproc プログラム, 8-39

## F

---

FAILOVER\_MODE パラメータ, 8-11, C-41  
FAILOVER パラメータ, 2-18, 2-26, 8-4, C-38, C-81  
finger ユーティリティ, 10-8

FLUSH\_NAME コマンド, A-35  
FLUSH コマンド, A-34

## G

---

General Inter-Orb Protocol  
    説明, 3-13  
    プレゼンテーション, 3-14  
GIOP  
    説明, 3-13  
    プレゼンテーション, 3-14  
GLOBAL\_DBNAME パラメータ, 7-11, C-47  
GLOBAL\_NAME パラメータ, 8-9, 8-46, C-43

## H

---

HELP コマンド  
    CMCTL, A-80  
    LSNRCTL, A-6  
    NAMESCTL, A-35  
hosts ファイル, 4-19  
HOST パラメータ, B-4  
HS パラメータ, 8-8, C-43

## I

---

Identix 認証方式  
    「Oracle Advanced Security」を参照  
    「Initial Preallocated Requests」フィールド, 6-58  
    「Instance Name」フィールド, 8-8  
INSTANCE\_NAME パラメータ, 2-28, 6-2, 8-8,  
    C-43, C-82  
IPC プロトコル  
    KEY パラメータ, B-3  
IPC、アドレスのパラメータ, B-3

## J

---

Java Database Connectivity (JDBC) ドライバ (Java  
    DataBase Connectivity Drivers)  
    「JDBC」を参照  
Java DBAccess Interface レイヤー、説明, 3-12  
Java NS、説明, 3-12  
Java Sockets、説明, 3-12  
JavaNet、説明, 3-12  
JavaTTC、説明, 3-12



## JDBC

- Java DBAccess Interface, 3-12
- Java NS, 3-12
- Java Sockets, 3-12
- JavaNet, 3-12
- OCI, 3-10
- TCP/IP プロトコル, 3-12
- Thin, 3-10
- ディレクトリ・ネーミング, 3-13
- ドライバ, 3-9

## K

---

Kerberos 認証方式

「Oracle Advanced Security」を参照

KEY パラメータ, B-3

## L

---

ldapmodify ツール, 6-27

ldap.ora ファイル

- DIRECTORY\_TYPE パラメータ, C-74
- DEFAULT\_ADMIN\_CONTEXT パラメータ, C-74
- DIRECTORY\_SERVERS パラメータ, C-74
- 作成, 2-39, 6-14

LDAP スキーマ

- オブジェクト・クラス, D-1, D-5
- 属性, D-1, D-5

LIST\_DELEGATED コマンド, A-36

LIST\_DOMAINS コマンド, A-37

LIST\_OBJECTS コマンド, A-38

LISTENER (LIS または LIST) 属性, 9-5

listener.log ファイル, 12-15

listener.ora ファイル

- パラメータ
- ORACLE\_HOME, 7-11

listener.ora ファイル

- パラメータ
- SID\_NAME, 7-11
- 構成パラメータのリファレンス, C-45 ~ C-55
- 説明, 1-5
- パラメータ
- ADDRESS, C-46
- ADDRESS\_LIST, C-46
- CONNECT\_TIMEOUT\_listener\_name, C-51
- DESCRIPTION, C-46
- GLOBAL\_DBNAME, 7-11, C-47

LOG\_DIRECTORY\_listener\_name, C-51

LOG\_FILE\_listener\_name, C-51

LOGGING\_listener\_name, C-51

ORACLE\_HOME, 8-41, 8-43, C-48

OSS.MY.WALLET, C-52

PASSWORDS\_listener\_name, C-52

POOL\_SIZE, 7-13

PRESPAWN\_DESC, C-50

PRESPAWN\_LIST, C-49

PRESPAWN\_MAX, 7-13, C-49

PROGRAM, 8-41, 8-43, C-48

PROTOCOL, 7-13

SAVE\_CONFIG\_ON\_STOP\_listener\_name, C-52

SID\_DESC, C-49

SID\_LIST\_listener\_name, C-47, C-49

SID\_NAME, 8-41, 8-43, C-48

SQLNET.CLIENT\_AUTHENTICATION, C-52

STARTUP\_WAITTIME\_listener\_name, C-53

TIMEOUT, 7-14

TRACE\_DIRECTORY\_listener\_name, C-53

TRACE\_FILE\_listener\_name, C-53

TRACE\_FILEN\_listener\_name, C-53

TRACE\_FILENO\_listener\_name, C-54

TRACE\_LEVEL\_listener\_name, C-54

TRACE\_TIMESTAMP\_listener\_name, C-55

USE\_PLUG\_AND\_PLAY\_listener\_name, C-55

リリース 8.1、変更, C-80

SID\_LIST\_listener\_name は不要, C-80

listener.trc ファイル, 12-28

LLU\_NAME パラメータ, B-3

LLU パラメータ, B-3

LOAD\_BALANCE パラメータ, 8-4, C-38, C-77, C-83

LOAD\_TNSNAMES コマンド, 6-62, A-38

LOCAL\_LISTENER パラメータ, 7-6, 7-8, 7-9, 9-5

LOCAL\_LU\_NAME パラメータ, B-3

LOCAL\_LU パラメータ, B-3

LOG\_DIRECTORY\_CLIENT パラメータ, 12-17, C-6

LOG\_DIRECTORY\_listener\_name パラメータ, 12-17, C-51

LOG\_DIRECTORY\_SERVER パラメータ, 12-17, C-6

LOG\_FILE\_CLIENT パラメータ, 12-17, C-7

LOG\_FILE\_listener\_name パラメータ, 12-17, C-51

LOG\_FILE\_SERVER パラメータ, 12-17, C-7

LOG\_LEVEL パラメータ, 12-18, C-69

LOG\_STATS コマンド, A-39

LOGGING\_listener\_name パラメータ, C-51



LSNRCTL ユーティリティ, A-3  
SET コマンド, A-3  
SHOW コマンド, A-3  
機能と構文書式, A-2  
コマンド  
    CHANGE\_PASSWORD, A-3  
    DBSNMP\_START, A-4  
    DBSNMP\_STATUS, A-5  
    DBSNMP\_STOP, A-5  
    EXIT, A-5  
    HELP, A-6  
    QUIT, A-7  
    RELOAD, A-7  
    SAVE\_CONFIG, A-8  
    SERVICES, A-9  
    SET, A-11  
    SET CONNECT\_TIMEOUT, A-11  
    SET CURRENT\_LISTENER, A-12  
    SET DISPLAYMODE, A-12  
    SET LOG\_DIRECTORY, A-13  
    SET LOG\_FILE, A-14  
    SET LOG\_STATUS, A-14  
    SET PASSWORD, 11-5, A-15  
    SET STARTUP\_WAITTIME, A-16  
    SET TRC\_DIRECTORY, A-16  
    SET TRC\_FILE, A-17  
    SET TRC\_LEVEL, A-17  
    SET USE\_PLUGANDPLAY, A-18  
    SHOW, A-19  
    SHOW CONNECT\_TIMEOUT, A-19  
    SHOW CURRENT\_LISTENER, A-19  
    SHOW DISPLAYMODE, A-19  
    SHOW LOG\_DIRECTORY, A-19  
    SHOW LOG\_FILE, A-19  
    SHOW LOG\_STATUS, A-19  
    SHOW PASSWORD, A-19  
    SHOW RAW\_MODE, A-19  
    SHOW SAVE\_CONFIG\_ON\_STOP, A-19  
    SHOW STARTUP\_WAITTIME, A-19  
    SHOW TRC\_DIRECTORY, A-19  
    SHOW TRC\_FILE, A-19  
    SHOW TRC\_LEVEL, A-19  
    SHOW USE\_PLUGANDPLAY, A-19  
    SPAWN, A-20  
    START, 11-6, A-20  
    STATUS, 11-5, A-22  
    STOP, 11-5, A-24

TRACE, A-24  
VERSION, A-25  
コマンドのリファレンス, A-3 ~ A-25  
説明, 4-24  
ネットワークのテスト, 11-2  
リスナーの起動, 7-4  
リスナーの停止, 7-4  
LU\_NAME パラメータ, B-3  
LU6.2 パラメータ  
    LU\_NAME, B-3  
    MDN パラメータ, B-3  
LU6.2 プロトコル, B-2  
    LLU\_NAME, B-3  
    LLU パラメータ, B-3  
    LOCAL\_LU\_NAME, B-3  
    LOCAL\_LU パラメータ, B-3  
    MODE パラメータ, B-3  
    PARTNER\_LU\_LOCAL\_ALIAS パラメータ, B-3  
    PARTNER\_LU\_NAME パラメータ, B-3  
    PLU\_LA パラメータ, B-3  
    PLU パラメータ, B-3  
    TP\_NAME プロトコル, B-3  
    TPN パラメータ, B-3  
    アドレスのパラメータ, B-3  
    サポートするトポロジー, 4-4  
    サポートするネットワーク, 4-4  
    説明, 4-3

## M

MAX\_FREELIST\_BUFFERS パラメータ, C-69  
「Maximum Open Connections」フィールド, 6-58  
「Maximum Prespawnd Servers」フィールド, 7-13  
「Maximum Wait Each Attempt」フィールド, 6-58  
MAXIMUM\_CONNECT\_DATA パラメータ, C-69  
MAXIMUM\_RELAYS パラメータ, C-69  
MDN パラメータ, B-3  
METHOD パラメータ, 8-12, C-42  
Microsoft の Active Directory, 2-35  
MODE パラメータ, B-3  
MOST\_RELAYS, A-92  
MTS  
    「マルチスレッド・サーバー」を参照  
MTS\_DISPATCHERS パラメータ, 8-26, 9-3, 9-8  
    ADDRESS 属性, 9-4  
    CONNECTIONS 属性, 9-4  
    DISPATCHERS 属性, 9-4



LISTENER 属性, 9-5  
MULTIPLEX 属性, 8-27, 9-5  
POOL 属性, 9-6  
PROTOCOL 属性, 8-26, 9-4  
SERVICE 属性, 9-6  
SESSIONS 属性, 9-6  
TICKS 属性, 9-6  
初期設定, 9-7

MULTIPLEX (MUL または MULT) 属性, 8-27, 9-5

## N

---

### Named Pipes プロトコル

PIPE パラメータ, B-3  
SERVER パラメータ, B-3  
アドレスのパラメータ, B-3  
説明, 4-3

Names Wizard, 6-47, 6-50

NAMES.ADDRESSES パラメータ, 6-48, 6-51, C-55  
NAMES.ADMIN\_REGION パラメータ, 6-48, C-56  
NAMES.AUTHORITY\_REQUIRED パラメータ, C-56  
NAMES.AUTO\_REFRESH\_EXPIRE パラメータ, C-56  
NAMES.AUTO\_REFRESH\_RETRY パラメータ, C-57  
NAMES.CACHE\_CHECKPOINT\_FILE パラメータ,  
C-57

NAMES.CACHE\_CHECKPOINT\_INTERVAL パラ  
メータ, C-57

NAMES.CONFIG\_CHECKPOINT\_FILE パラメータ,  
C-57

NAMES.CONNECT\_TIMEOUT パラメータ, C-58  
NAMESCTL.INTERNAL\_ENCRYPT\_PASSWORD パ  
ラメータ, C-10

NAMESCTL.INTERNAL\_USE パラメータ, C-10  
NAMESCTL.NO\_INITIAL\_SERVER パラメータ, C-11  
NAMESCTL.NOCONFIRM パラメータ, C-11  
NAMESCTL.SERVER\_PASSWORD パラメータ, C-11  
NAMESCTL.TRACE\_DIRECTORY パラメータ, C-12  
NAMESCTL.TRACE\_FILE パラメータ, C-12  
NAMESCTL.TRACE\_LEVEL パラメータ, C-11  
NAMESCTL.TRACE\_UNIQUE パラメータ, C-12  
NAMESCTL ユーティリティ, A-26

Oracle Names サーバーの起動, 11-4  
Oracle Names サーバーの検出, 6-54, 6-56  
tnsnames.ora ファイルの Oracle Names への  
ロード, 6-62  
オブジェクトの問合せ, 11-12  
確認モード, A-27

機能と構文書式, A-25  
グローバル・データベース・リンク, 6-63  
コマンド

DELEGATE\_DOMAIN, A-27  
DOMAIN\_HINT, A-28  
DUMP\_LDAP, A-29  
DUMP\_TNSNAMES, A-33  
EXIT, A-33  
FLUSH, A-34  
FLUSH\_NAME, A-35  
HELP, A-35  
LIST\_DELEGATED, A-36  
LIST\_DOMAINS, A-37  
LIST\_OBJECTS, A-38  
LOAD\_TNSNAMES, A-38  
LOG\_STATS, A-39  
PASSWORD, A-40  
PING, A-41  
QUERY, 11-11, A-42  
QUIT, A-43  
REGISTER, A-44  
RELOAD, A-45  
REORDER\_NS, A-45  
REPEAT, A-46  
RESET\_STATS, A-47  
RESTART, A-48  
SET, A-26, A-49  
SET CACHE\_CHECKPOINT\_INTERVAL, A-50  
SET DEFAULT\_DOMAIN, A-50  
SET FORWARDING\_AVAILABLE, A-51  
SET LOG\_FILE\_NAME, A-52  
SET LOG\_STATS\_INTERVAL, A-53  
SET NAMESCTL\_TRACE\_LEVEL, A-54  
SET PASSWORD, A-54  
SET REQUESTS\_ENABLED, A-55  
SET RESET\_STATS\_INTERVAL, A-56  
SET SAVE\_CONFIG\_INTERVAL, A-57  
SET SAVE\_CONFIG\_ON\_STOP, A-57  
SET SERVER, A-57  
SET TRACE\_FILE\_NAME, A-58  
SET TRACE\_LEVEL, A-59  
SHOW, A-26, A-60  
SHOW CACHE CHECKPOINT\_INTERVAL,  
A-61  
SHOW DEFAULT\_DOMAIN, A-61  
SHOW FORWARDING\_AVAILABLE, A-62  
SHOW LOG\_FILE\_NAME, A-63



SHOW LOG\_STATS\_INTERVAL, A-63  
 SHOW NAMESCTL\_TRACE\_LEVEL, A-64  
 SHOW REQUEST\_ENABLED, A-64  
 SHOW RESETS\_STATS\_INTERVAL, A-65  
 SHOW SAVE\_CONFIG\_INTERVAL, A-65  
 SHOW SAVE\_CONFIG\_ON\_STOP, A-66  
 SHOW SERVER, A-66  
 SHOW STATS, A-66  
 SHOW SYSTEM\_QUERIES, A-67  
 SHOW TRACE\_FILE\_NAME, A-68  
 SHOW TRACE\_LEVEL, A-68  
 SHOW VERSION, A-69  
 SHUTDOWN, A-70  
 START, A-70  
 START\_CLIENT\_CACHE, 11-5, A-72  
 STARTUP, A-73  
 STATUS, A-74  
 STOP, A-74  
 TIMED\_QUERY, A-75  
 UNREGISTER, A-75  
 VERSION, A-76  
 コマンドのリファレンス, A-27 ~ A-77  
 説明, 4-25  
 動作モード, A-2  
 ドメインの委任, 6-67  
 ネット・サービス名, 6-61  
 ネットワーク・オブジェクトのテスト, 11-11  
 ネットワークのテスト, 11-2  
 分散処理, A-26  
 NAMES.DCE.PREFIX パラメータ, C-7  
 NAMES.DEFAULT\_DOMAIN パラメータ, 6-6, 6-58, 8-16  
 NAMES.DEFAULT\_FORWARDERS\_ONLY パラメータ, C-58  
 NAMES.DEFAULT\_FORWARDERS パラメータ, C-58  
 NAMES.DEFAULT.DOMAIN パラメータ, C-7  
 NAMES.DIRECTORY\_PATH パラメータ, 8-18, C-8  
   CDS, 8-18  
   HOSTNAME, 8-18  
   LDAP, 8-17  
   NDS, 8-18  
   NIS, 8-18  
   ONAMES, 8-18  
   TNSNAMES, 8-17  
 NAMES.DOMAIN\_CHECKPOINT\_FILE パラメータ, C-75  
 NAMES.DOMAIN\_HINTS パラメータ, C-59  
 NAMES.DOMAINS パラメータ, 6-49, 6-52, C-59  
 NAMES.FORWARDING\_AVAILABLE パラメータ, C-59  
 NAMES.FORWARDING\_DESIRED パラメータ, C-60  
 namesini.sql スクリプト, 6-46  
 NAMES.INITIAL\_RETRY\_TIMEOUT パラメータ, 6-58, C-8  
 NAMES.KEEP\_DB\_OPEN パラメータ, C-60  
 NAMES.LOG\_DIRECTORY パラメータ, 12-18, C-60  
 NAMES.LOG\_FILE パラメータ, 12-18, C-60  
 NAMES.LOG\_STATS\_INTERVAL パラメータ, C-61  
 NAMES.LOG\_UNIQUE パラメータ, C-61  
 names.log ファイル, 12-15  
 NAMES.MAX\_OPEN\_CONNECTIONS パラメータ, 6-58, C-8, C-61  
 NAMES.MAX\_REFORWARDS パラメータ, C-62  
 NAMES.MESSAGE\_POOL\_START\_SIZE パラメータ, 6-58  
 NAMES.MESSAGE\_POOL\_START\_SIZE パラメータ, C-9, C-62  
 NAMES.NDS.NAME\_CONTEXT パラメータ, 6-77  
 NAMES.NDS.NAME.CONTEXT パラメータ, C-9  
 NAMES.NIS.META\_MAP パラメータ, C-9  
 NAMES.NO\_MODIFY\_RESPONSE パラメータ, C-62  
 NAMES.NO\_REGION\_DATABASE パラメータ, C-62  
 names.ora ファイル  
   構成パラメータのリファレンス, C-55, C-65  
   廃止されたパラメータ, C-75  
   パラメータ  
     NAMES.ADDRESSES, C-55  
     NAMES.ADMIN\_REGION, C-56  
     NAMES.AUTHORITY\_REQUIRED, C-56  
     NAMES.AUTO\_REFRESH\_EXPIRE, C-56  
     NAMES.AUTO\_REFRESH\_RETRY, C-57  
     NAMES.CACHE\_CHECKPOINT\_FILE, C-57  
     NAMES.CACHE\_CHECKPOINT\_INTERVAL, C-57  
     NAMES.CONFIG\_CHECKPOINT\_FILE, C-57  
     NAMES.CONNECT\_TIMEOUT, C-58  
     NAMES.DEFAULT\_FORWARDERS, C-58  
     NAMES.DEFAULT\_FORWARDERS\_ONLY, C-58  
     NAMES.DOMAIN\_CHECKPOINT\_FILE, C-75  
     NAMES.DOMAIN\_HINTS, C-59  
     NAMES.DOMAINS, C-59  
     NAMES.FORWARDING\_AVAILABLE, C-59  
     NAMES.FORWARDING\_DESIRED, C-60



- NAMES.KEEP\_DB\_OPEN, C-60
- NAMES.LOG\_DIRECTORY, C-60
- NAMES.LOG\_FILE, C-60
- NAMES.LOG\_STATS\_INTERVAL, C-61
- NAMES.LOG\_UNIQUE, C-61
- NAMES.MAX\_OPEN\_CONNECTIONS, C-61
- NAMES.MAX\_REFORWARDS, C-62
- NAMES.MESSAGE\_POOL\_START\_SIZE, C-62
- NAMES.NO\_MODIFY\_REQUESTS, C-62
- NAMES.NO\_REGION\_DATABASE, C-62
- NAMES.PASSWORD, C-63
- NAMES.REGION\_CHECKPOINT\_FILE, C-63
- NAMES.RESET\_STATS\_INTERVAL, C-63
- NAMES.SAVE\_CONFIG\_ON\_STOP, C-64
- NAMES.SERVER\_NAME, C-64
- NAMES.TOPOLOGY\_CHECKPOINT\_FILE, C-75
- NAMES.TRACE\_DIRECTORY, C-64
- NAMES.TRACE\_FILE, C-64
- NAMES.TRACE\_FUNC, C-64
- NAMES.TRACE\_LEVEL, C-65
- NAMES.TRACE\_UNIQUE, C-65
- NAMES.USE\_PLUG\_AND\_PLAY, C-75
- NAMES.PASSWORDS パラメータ, C-63
- namespid.trc ファイル, 12-34
- NAMES.PREFERRED\_SERVERS パラメータ, 8-24, 8-25, C-9
- NAMES.REGION\_CHECKPOINT\_FILE パラメータ, C-63
- NAMES.REQUEST\_RETRIES パラメータ, 6-58, C-10
- NAMES.RESET\_STATS\_INTERVAL パラメータ, C-63
- NAMES.SAVE\_CONFIG\_ON\_STOP パラメータ, C-64
- NAMES.SERVER\_NAMES パラメータ, C-64
- NAMES.SERVER\_NAME パラメータ, 6-48, 6-51
- NAMES.TOPOLOGY\_CHECKPOINT\_FILE パラメータ, C-75
- NAMES.TRACE\_DIRECTORY パラメータ, 12-33, C-64
- NAMES.TRACE\_FILE パラメータ, 12-33, C-64
- NAMES.TRACE\_FUNC パラメータ, C-64
- NAMES.TRACE\_LEVEL パラメータ, 12-33, C-65
- NAMES.TRACE\_UNIQUE パラメータ, 12-34, C-65
- names.trc ファイル, 12-28
- namesupg.sql スクリプト, 6-46
- NAMES.USE\_PLUG\_AND\_PLAY パラメータ, C-75

- NDS 外部ネーミング, 2-33, 6-71, 8-18
  - 接続, 6-75
  - 接続の確立, 6-75
- Net Service Name Wizard, 6-6, 6-20
- NET\_CONFIG\_METHOD パラメータ, C-74
- Net8
  - JDBC ドライバ, 3-9
    - 概念の説明, 1-2
    - 概要, 1-2
    - 主要機能, 2-6
    - 制御ユーティリティ, 11-2
    - 接続操作, 2-2
    - 操作, 2-6
    - 例外操作, 2-6
- Net8 Client、説明, 4-2
- Net8 Configuration Assistant
  - OracleDBCreators グループ, 2-39, 2-46, 6-19
  - OracleNetAdmins グループ, 2-45, 6-19
  - OracleSecurityAdmins グループ, 2-46, 6-19
  - 起動, 4-23
  - クライアント
    - 管理コンテキストの選択, 6-16
    - ディレクトリ・アクセス, 6-18
    - ローカル・ネーミング・メソッド, 6-9
  - サーバー
    - Oracle コンテキストの作成, 2-38, 6-19
    - Oracle スキーマの作成, 6-19
    - 管理コンテキストの選択, 6-16
    - ディレクトリ・アクセス, 6-19
  - 説明, 4-23
  - ネット・サービス名, 6-9
  - リスナーの構成, 7-3
    - ローカル・ネーミング・メソッド, 6-9
- Net8 Firewall Proxy, 4-7
- Net8 OPEN
  - API のインストール先, 10-6
  - API ファンクション・コール, 10-3
  - C 言語との互換性, 10-3
  - アプリケーションを使用するためのシステム構成, 10-7
  - エラー・メッセージ, 10-8
  - 構成要件, 10-6
  - サンプル・アプリケーション, 10-8
  - 非 SQL 情報との統合, 10-2
  - 分散アプリケーション用, 10-2
  - ライブラリ, 10-6



Net8 OPEN による独自アプリケーションの開発, 10-7  
Net8 Server、説明, 4-2  
Net8 アクセス制御, 4-6, 8-31  
NIS 外部ネーミング, 2-32, 4-21, 6-71, 8-18  
    接続, 6-71  
    接続の確立, 6-71  
NIS マップ, 6-72  
NI レイヤー, 3-6  
NNL-00018 エラー・メッセージ, 6-50  
NNL-00024 エラー・メッセージ, 6-50  
Novell ディレクトリ・サービス外部ネーミング, 2-33,  
    2-35, 8-18  
    構成, 6-71  
    接続, 6-75  
    接続の確立, 6-75  
    説明, 4-21  
NPI, 3-8  
NS.SMD レコード, 11-12, A-42  
NS レイヤー, 3-7

## O

---

OCI、説明, 3-5  
ONRS\_CONFIG テーブル, 6-44  
Open System Interconnection (OSI) モデル  
    クライアント・アプリケーション・レイヤー, 3-3  
    セッション・レイヤー, 3-4  
    説明, 3-2  
    トランスポート・レイヤー, 3-4  
    物理, 3-4  
    プレゼンテーション・レイヤー, 3-3  
    リンク・レイヤー, 3-4  
OPI、説明, 3-8  
ORA-00508 エラー・メッセージ, 12-8  
ORA-1017 エラー・メッセージ, 12-3  
ORA-1034 エラー・メッセージ, 12-3  
ORA-12203 エラー・メッセージ, 12-8, 12-12  
    サンプルのエラー・スタック, 12-15  
ORA-12533 エラー・メッセージ, 12-10  
ORA-12535 エラー・メッセージ, 12-12  
ORA-12538 エラー・メッセージ, 12-8  
ORA-12547 エラー・メッセージ, 12-12  
ORA-12560 エラー・メッセージ, 12-11  
ORA-3121 エラー・メッセージ, 12-11  
Oracle Advanced Security  
    CyberSAFE 認証方式, 4-22  
DCE インテグレーション, 4-23  
DES\_40 暗号化, 4-22  
Identix 認証方式, 4-22  
Kerberos 認証方式, 4-23  
MD5 チェックサム, 4-22  
RADIUS 認証方式, 4-23  
RC4\_128 暗号化, 4-22  
RC4\_40 暗号化, 4-22  
RC4\_56 暗号化, 4-22  
SecurID 認証方式, 4-23  
SSL 認証方式, 4-23  
    機能, 4-22  
    シングル・サインオン, 4-22  
    説明, 4-22  
    ネットワーク・セキュリティ, 4-22  
Oracle Connection Manager  
    CMADMIN プロセス, 4-5  
    CMGW プロセス, 4-5  
    Net8 Firewall Proxy, 4-7  
    Net8 アクセス制御, 4-6  
    SOURCE\_ROUTE パラメータ, C-39  
    アクセス制御, 4-6  
    起動, 11-7  
    構成  
        Net8 アクセス制御, 8-31  
        Oracle Connection Manager のプロトコル・アド  
        レス, 8-34  
        アクセス制御, 8-31  
        接続集中化, 8-25  
        複数プロトコル・サポート, 8-28  
        プロトコル・アドレス, B-1  
        リスナーのプロトコル・アドレス, 8-36  
    構成パラメータのリファレンス, C-65 ~ C-72  
    接続集中化, 4-6  
    接続のルーティング, 1-3, 8-18  
    テスト, 11-13  
    トレース・ファイル, 12-34  
        構成, 12-34  
    複数プロトコル環境での役割, 4-8  
    複数プロトコル・サポート, 4-7  
    プロセス, 4-5  
    マルチスレッド・サーバー, 4-6, 8-25  
    ログ・ファイル, 12-18  
        構成, 12-18  
        理解, 12-24



Oracle Connection Manager 制御ユーティリティ  
(Oracle Connection Manager Control Utility:  
CMCTL)  
「CMCTL ユーティリティ」を参照, 11-7  
Oracle Database Configuration Assistant  
ディレクトリ・ネーミング, 2-38, 6-19  
Oracle Enterprise Manager  
listener.ora 内の `SID_LIST_listener_name` パラ  
メータ, C-80  
listener.ora の静的サービス情報, 2-18, 5-7  
`SID_LIST_listener_name` listener.ora のパラメータ,  
7-10  
接続時フェイルオーバー, 2-18, 5-7  
Oracle Internet Directory, 2-35  
Oracle Names, 2-31  
namesini.sql スクリプト, 6-46  
namesupg.sql スクリプト, 6-46  
Oracle Names サーバーの検出, 6-53, 6-55  
Oracle Names サーバーの作成, 6-44  
.sdns.ora ファイル, 4-20  
アーキテクチャ, 4-8  
委任  
管理リージョン, 4-18  
インストール, 6-44  
オブジェクトの問合せ, 11-12  
階層ネーミング・モデル, 4-14  
格納  
Oracle Connection の情報, 4-13  
その他の Oracle Names サーバーの名前およびア  
ドレス, 4-12  
データベース・グローバル・データベース名お  
よびアドレス, 4-12  
データベース・リンク, 4-13  
ネット・サービス名, 4-12  
別名, 4-13  
管理リージョン, 4-10  
起動  
Oracle Net8 Assistant から, 11-12  
コマンドラインから, 11-12  
コントロール パネルから, 11-7  
クライアント・キャッシュ, 11-4  
検出, 4-20  
構成, 6-43 ~ 6-67  
クライアントとサーバー, 6-55  
リージョン・データベース, 6-46  
コマンドラインからの起動, 11-12  
情報のリポジトリとしてのデータベース, 4-11

シングル・ドメイン・ネーミング・モード, 4-14  
推奨環境, 5-3  
接続, 6-67  
接続の確立, 2-31, 6-67  
説明, 2-31, 4-9  
ディレクトリ・サーバーへのオブジェクトのエク  
ポート, 6-31 ~ 6-42, A-29  
デフォルト・ネットワーク・ドメイン, 4-15  
動的検出オプション, 4-19  
登録  
オブジェクト, 6-59  
グローバル・データベース・リンク, 6-62  
グローバル・データベース・リンク修飾子, 6-62  
ネット・サービス名, 6-60  
ドメイン, 4-11  
複数の管理, 6-49, 6-52  
トレース・ファイル, 12-33  
ネットワーク・コンポーネントのネーミング, 4-14  
バージョン 1, 4-18  
バージョン 2, 4-19  
バージョン 8, 4-19  
バージョン間の相違, 4-18  
複数のドメイン, 4-15  
複数のリージョン, 4-16  
リージョン, 4-10  
リージョン・データベース, 4-11  
リスナーの構成, 6-59  
利点および不利な点, 5-3  
ルート管理リージョン, 4-16  
レプリケートされたデータ, 6-50  
レプリケート・データ, 4-11  
ログ・ファイル, 12-17  
Oracle Names サーバー  
ckcfg.ora ファイル, 6-44  
ckpcch.ora ファイル, 6-45  
ckpreg.ora ファイル, 6-45  
委任  
ドメイン, 6-66  
委任管理リージョン, 4-16  
格納  
Oracle Connection の情報, 4-13  
その他の Oracle Names サーバーの名前およびア  
ドレス, 4-12  
データベース・グローバル・データベース名お  
よびアドレス, 4-12  
データベース・リンク, 4-13



- ネット・サービス名, 4-12
- 別名, 4-13
- 起動, 6-49, 11-4
  - Oracle Net8 Assistant から, 11-12
  - クライアント・キャッシュ, 11-4
  - コマンドラインから, 11-12
  - コントロール パネルから, 11-7
- クライアント・キャッシュ, 11-4
- 検出, 6-53
- 構成
  - 複数のドメイン, 6-49, 6-52
  - プロトコル・アドレス, B-1
- 構成パラメータのリファレンス, C-55 ~ C-65
- 作成, 6-44
  - リージョン・データベース, 6-46
  - レプリケートされたデータ, 6-50
- 説明, 4-9
- テスト, 11-10
  - TNSPING ユーティリティ, 11-16
- デフォルトの構成, 6-45
- 登録
  - オブジェクト, 6-59
  - グローバル・データベース・リンク, 6-62
  - グローバル・データベース・リンク修飾子, 6-62, 6-64
  - ネット・サービス名, 6-60
  - トレース・ファイル, 12-33
  - に格納されるデータ, 4-12
  - 優先, 4-18, 8-23
  - 予約済, 4-19
  - ルート管理リージョン, 4-16
  - レプリケートされたデータ, 6-50
  - ログ・ファイル, 12-17
  - 構成, 12-18
- Oracle Names サーバーの検出
  - 説明, 4-20
  - プロシージャ, 6-53
- Oracle Names 制御ユーティリティ (NAMESCTL)
  - 「NAMESCTL ユーティリティ」を参照
- Oracle Names でのデータのレプリケーション, 6-50
- Oracle Names でのレプリケート・データ, 4-11
- Oracle Net8 Assistant, 8-46
  - 「Default Domain」フィールド, 6-58
  - 「Directory Server Migration Wizard」, 6-29
  - 「Instance Name」フィールド, 8-8
  - Names Wizard, 6-47, 6-50
  - Net Service Name Wizard, 6-6, 6-20

- Oracle Names
  - 「Attempts Per Names Server」フィールド, 6-58
  - 「Default Domain」フィールド, 6-58
  - 「Initial Preallocated Requests」フィールド, 6-58
  - 「Maximum Open Connections」フィールド, 6-58
  - 「Maximum Wait Each Attempt」フィールド, 6-58
  - Oracle Names サーバーの起動, 11-12
  - グローバル・データベース・リンク, 6-63
  - グローバル・データベース・リンク修飾子, 6-64
  - 検出, 6-56
  - デフォルト以外のリスナー, 7-9
  - ネット・サービス名, 6-61
  - ネットワーク・オブジェクトの問合せ, 11-11
  - 別名, 6-66
  - 優先 Oracle Names サーバー, 8-23
  - リージョン・データベース, 6-46
  - リスナーからのデータベース情報の登録, 6-59
- Oracle Names サーバーの起動, 6-54, 11-4
- Oracle Names サーバーの検出, 6-56
- Oracle Net8 Assistant の「Session Data Unit (SDU)」フィールド, 8-8
- 「Oracle Rdb Database」フィールド, 8-9
- Oracle Rdb データベース, 8-46
- 「Rdb Database」フィールド, 8-46
- 「Timeout」フィールド, 7-14
- 「Type of Service」フィールド, 8-9
- 「Use for Heterogeneous Services」オプション, 8-8, 8-45
- アドレスの追加, 8-2
- 異種サービス接続, 8-43
- 外部プロシージャ接続, 8-39
- 管理リージョンへのドメインの委任, 6-67
- 起動, 4-24
- クライアント
  - Oracle Connection Manager, 8-34
  - クライアント・ロード・バランス, 8-3
  - 接続時フェイルオーバー, 8-3
  - デフォルト・ネットワーク・ドメイン, 8-15
  - ローカル・ネーミング・メソッド, 6-6
- 接続要求のルーティング, 8-18
- 説明, 4-24
- ディレクトリ・ネーミング, 2-38
- テスト
  - クライアントの構成, 11-18
  - サーバーの構成, 11-13



トレース・ファイル, 12-34  
 ネーミング・メソッドの指定, 8-16  
 ネット・サービス名, 6-6  
 複数のアドレス・オプション, 8-3  
 プロファイル  
   「Always Use Dedicated Server」オプション, 8-19  
   「Client Registration ID」フィールド, 8-21  
   「Disable Out-of-Band Break」オプション, 8-22  
   「Prefer Connection Manager Routing」オプション, 8-19  
   「TNS Time Out Value」オプション, 8-21  
   「Turn Off UNIX Signal Handling」オプション, 8-21  
   詳細オプション, 8-20  
   接続のルーティング, 8-19  
   ネーミング・メソッド指定, 8-16  
   優先 Oracle Names サーバー, 8-23  
 リスナー  
   「Configure Prespawnd Servers」ダイアログ・ボックス, 7-15  
   「Database Services」タブ, 7-12  
   「Global Database Name」フィールド, 7-11, 8-9, 8-46  
   「Maximum Prespawnd Servers」フィールド, 7-13  
   「Oracle Home Directory」フィールド, 8-41, 8-43  
   Oracle Names でのデフォルト以外の, 7-9  
   Oracle Names とともにデータベース情報を登録, 6-59  
   「Program Name」フィールド, 8-41, 8-43  
   「Register Service with Oracle Names」オプション, 6-59  
   「SID」フィールド, 7-11  
   構成, 7-4  
   事前生成済専用サーバー, 7-15  
   静的サービス情報, 7-10  
   プロトコル・アドレス, 7-5  
   リスナーの「Oracle Home Directory」フィールド, 7-11  
   リスナーの「SID」フィールド, 8-41, 8-43  
   ローカル・ネーミング・メソッド, 6-6  
   ログ・ファイル, 12-18  
 Oracle Net8 Assistant の「Always Use Dedicated Server」オプション, 8-19  
 Oracle Net8 Assistant の「Database Services」タブ, 7-12  
 Oracle Net8 Assistant の「Disable Out-of-Band Break」オプション, 8-22  
 Oracle Net8 Assistant の「Global Database Name」フィールド, 7-11, 8-9, 8-46  
 Oracle Net8 Assistant の「Oracle Home Directory」フィールド, 7-11, 8-41, 8-43  
 Oracle Net8 Assistant の「Prefer Connection Manager Routing」オプション, 8-19  
 Oracle Net8 Assistant の「Program Name」フィールド, 8-41, 8-43  
 Oracle Net8 Assistant の「Session Data Unit (SDU) Size」フィールド, 8-8  
 Oracle Net8 Assistant の「SID」フィールド, 7-11, 8-41, 8-43  
 Oracle Net8 Assistant の「Turn Off UNIX Signal Handling」オプション, 8-21  
 Oracle Net8 Assistant の「Type of Service」フィールド, 8-9, 8-46  
 Oracle Net8 Assistant の「Use for Heterogeneous Services」オプション, 8-8, 8-45  
 Oracle Parallel Server  
   FAILOVER\_MODE パラメータ, 8-11, C-41  
   FAILOVER パラメータ, 8-4, C-38  
   INSTANCE\_NAME パラメータ, C-44, C-82  
   LOAD\_BALANCE パラメータ, C-38, C-83  
   接続時フェイルオーバー, 2-18, 8-4, C-38  
   透過的アプリケーション・フェイルオーバー, 8-11  
   「Oracle Rdb Database」フィールド, 8-9  
 Oracle Rdb データベース  
   接続の構成, 8-46  
   説明, 8-46  
 Oracle コンテキスト  
   作成, 6-15, 6-19  
   説明, 2-36  
 ORACLE\_HOME、定義, xviii  
 ORACLE\_HOME パラメータ, 7-11, 8-41, 8-43, C-48  
 Oracle8i JServer  
   接続, 3-14  
   リスナーの構成, 7-7  
 Oracle8i JServer の接続, 7-7  
 OracleDBCreators グループ, 2-39, 2-46, 6-19  
 OracleNetAdmins グループ, 2-39, 2-45, 6-19, 6-20, 6-27  
 OracleSecurityAdmins グループ, 2-46, 6-19



Oracle コール・インタフェース (OCI) レイヤー、  
説明, 3-5  
Oracle システム識別子、リスナーの構成, 7-11  
Oracle スキーマ  
オブジェクト・クラス, D-1 ~ D-5  
作成, 6-19  
説明, 2-46  
属性, D-1, D-5  
Oracle プログラム・インタフェース (OPI)、説明, 3-8  
Oracle プロトコル  
Bequeath, 4-4, B-2  
IPC, B-2  
LU6.2, 4-3, B-2  
Named Pipes, 4-3, B-2  
SPX, 4-3, B-3  
SSL 付き TCP/IP, 4-2, B-4  
TCP/IP, 4-2, B-4  
アドレスの構成, B-2  
説明, 3-7, 4-2  
Oracle ホーム・ディレクトリ  
リスナーの構成, 7-11  
Oracle ホーム・ディレクトリ、定義, xviii  
orclDBServer オブジェクト・クラス, 2-47, D-2  
orclDescList 属性, D-4  
orclDescName 属性, D-4  
orclLoadBalance 属性, D-4  
orclNetAddressList オブジェクト・クラス, 2-47, D-4  
orclNetAddress オブジェクト・クラス, 2-47, D-3  
orclNetAddrList 属性, D-4  
orclNetAddrString 属性, D-4  
orclNetConnParamList 属性, D-4  
orclNetDescriptionList オブジェクト・クラス, 2-47,  
D-3  
orclNetDescription オブジェクト・クラス, 2-47, D-2  
orclNetFailover 属性, D-4  
orclNetInstanceName 属性, D-4  
orclNetProtocolStack 属性, D-4  
orclNetSdu 属性, D-4  
orclNetServiceName 属性, D-4  
orclNetService オブジェクト・クラス, 2-47, D-2  
orclNetSourceRoute 属性, D-4  
orclProtocol 属性, D-4  
orclSid 属性, D-5  
orclVersion 属性, D-5  
OSI モデル  
クライアント・アプリケーション・レイヤー, 3-3  
セッション・レイヤー, 3-4

トランスポート・レイヤー, 3-4  
ネットワーク・レイヤー, 3-4  
物理, 3-4  
プレゼンテーション・レイヤー, 3-3  
OSS.MY.WALLET パラメータ, C-12, C-52  
OUT\_OF\_RELAY, A-92  
ou 属性, 2-36  
o 属性, 2-36

## P

---

PARTNER\_LU\_LOCAL\_ALIAS パラメータ, B-3  
PARTNER\_LU\_NAME パラメータ, B-3  
PASSWORDS\_listener\_name パラメータ, C-52  
PASSWORD コマンド, A-40  
PING ユーティリティ, 11-10, A-41  
Oracle Names サーバーのテスト, 11-10  
PIPE パラメータ, B-3  
PLU\_LA パラメータ, B-3  
PLU パラメータ, B-3  
PMON プロセス, 2-19  
POOL\_SIZE パラメータ, 2-15, 7-13  
POOL 属性, 9-6  
PORT パラメータ, B-4  
PRESPAWN\_DESC パラメータ, C-50  
PRESPAWN\_LIST パラメータ, C-49  
PRESPAWN\_MAX パラメータ, 2-16, 7-13, C-49  
PROGRAM パラメータ, 8-41, 8-43, B-3, C-48  
protocol.EXCLUDED\_NODES パラメータ, C-73  
protocol.INVITED\_NODES パラメータ, C-73  
protocol.ora ファイル, C-74  
protocol.EXCLUDED\_NODES パラメータ, C-73  
protocol.INVITED\_NODES パラメータ, C-73  
protocol.VALIDNODE\_CHECKING パラメータ,  
C-73  
TCP.NODELAY パラメータ, C-73  
構成パラメータのリファレンス, C-73  
データベースへのアクセスの制御, C-73  
PROTOCOL (PRO または PROT) 属性, 8-26, 9-4  
protocol.VALIDNODE\_CHECKING パラメータ, C-73  
PROTOCOL パラメータ, 7-13, B-3

## Q

---

QUERY コマンド, 6-54, 11-11, 11-12  
A.SMD レコード, 7-10, 11-12, A-42  
CNAME.SMD, 11-12, A-42



DLCR.TDBMS.OMD レコード, 11-12, A-42  
DL.RDBMS.OMD, 11-12, A-42  
NAMESCTL、リファレンス, A-42  
NS.SMD, 11-12, A-42  
VIADD.NPO.OMD, 11-12, A-42  
QUEUE\_SIZE パラメータ, 7-10  
リスナー・キューサイズの調整, 5-5, 7-10  
QUIT コマンド  
CMCTL, A-80  
LSNRCTL, A-7  
NAMESCTL, A-43

## R

---

RADIUS 認証方式  
「Oracle Advanced Security」を参照  
「Rdb Database」フィールド, 8-46  
RDB\_DATABASE パラメータ, 8-46, C-44  
「Register Service with Oracle Names」オプション,  
6-59  
REGISTER コマンド, 6-61, 6-63, A-44  
RELAY\_STATISTICS パラメータ, C-70  
RELOAD コマンド  
LSNRCTL, A-7  
NAMESCTL, A-45  
REMOTE\_ADMIN パラメータ, C-70, C-84  
REORDER\_NS コマンド, 6-54, 6-56, 8-23, A-45  
REPEAT コマンド, A-46  
RESET\_STATS コマンド, A-47  
RESTART コマンド, A-48  
RETRIES パラメータ, 8-12, C-42

## S

---

SAVE\_CONFIG\_ON\_STOP\_listener\_name パラメータ,  
C-52  
SAVE\_CONFIG コマンド, A-8  
.sdns.ora ファイル, 4-20  
SDU  
「セッション・データ・ユニット」を参照  
SDU パラメータ, 8-8, C-39  
SecurID 認証方式  
「Oracle Advanced Security」を参照  
SERVER パラメータ, 2-8, B-3, C-44  
SERVICE\_DIED イベント, 12-22  
SERVICE\_NAMES パラメータ, 2-28, 6-3, 9-6  
SERVICE\_NAME パラメータ, 6-2, C-45, C-83

SERVICE\_REGISTER イベント, 12-22  
SERVICE\_UPDATE イベント, 12-22  
SERVICE (SER または SERV) 属性, 9-6  
SERVICES コマンド, 2-8, 2-13, A-9  
SERVICE パラメータ, B-3  
SESSIONS (SES または SESS) 属性, 9-6  
SET AUTHENTICATION\_LEVEL コマンド, A-81  
SET CACHE\_CHECKPOINT\_INTERVAL コマンド,  
A-50  
SET CONNECT\_TIMEOUT コマンド, A-11  
SET CURRENT\_LISTENER コマンド, A-12  
SET DEFAULT\_DOMAIN コマンド, A-50  
SET DISPLAYMODE コマンド  
CMCTL, A-82  
LSNRCTL, A-12  
SET FORWARDING\_AVAILABLE コマンド, A-51  
SET LOG\_DIRECTORY コマンド, A-13  
SET LOG\_FILE\_NAME コマンド, A-52  
SET LOG\_FILE コマンド, A-14  
SET LOG\_LEVEL コマンド, A-82  
SET LOG\_STATS\_INTERVAL コマンド, A-53  
SET LOG\_STATUS コマンド, A-14  
SET NAMESCTL\_TRACE\_LEVEL コマンド, A-54  
SET PASSWORD コマンド  
LSNRCTL, 11-5, A-15  
NAMESCTL, A-54  
SET RELAY\_STATISTICS コマンド, A-83  
SET REQUESTS\_ENABLED コマンド, A-55  
SET RESET\_STATS\_INTERVAL コマンド, A-56  
SET SAVE\_CONFIG\_INTERVAL コマンド, A-57  
SET SAVE\_CONFIG\_ON\_STOP コマンド, A-15  
LSNRCTL, A-15  
NAMESCTL, A-57  
SET SERVER コマンド, A-57  
SET STARTUP\_WAITTIME コマンド, A-16  
SET TRACE\_FILE\_NAME コマンド, A-58  
SET TRACE\_LEVEL コマンド, A-59  
SET TRC\_DIRECTORY コマンド, A-16  
SET TRC\_FILE コマンド, A-17  
SET TRC\_LEVEL コマンド, A-17  
SET USE\_PLUGANDPLAY コマンド, A-18  
SET コマンド  
CMCTL, A-81  
LSNRCTL, A-11  
NAMESCTL, A-49  
SHOW ADDRESS コマンド, A-84  
SHOW ALL コマンド, A-85



SHOW CACHE\_CHECKPOINT\_INTERVAL コマンド,  
     A-61  
 SHOW CONNECT\_TIMEOUT コマンド, A-19  
 SHOW CURRENT\_LISTENER コマンド, A-19  
 SHOW DEFAULT\_DOMAIN コマンド, A-61  
 SHOW DISPLAYMODE コマンド  
     CMCTL, A-86  
     LSNRCTL, A-19  
 SHOW FORWARDING\_AVAILABLE コマンド, A-62  
 SHOW LOG\_DIRECTORY コマンド, A-19  
 SHOW LOG\_FILE\_NAME コマンド, A-63  
 SHOW LOG\_FILE コマンド, A-19  
 SHOW LOG\_STATS\_INTERVAL コマンド, A-63  
 SHOW LOG\_STATUS コマンド, A-19  
 SHOW NAMESCTL\_TRACE\_LEVEL コマンド, A-64  
 SHOW PASSWORD コマンド, A-19  
 SHOW PROFILE コマンド, A-87  
 SHOW RAW\_MODE コマンド, A-19  
 SHOW RELAY コマンド, A-87  
 SHOW REQUESTS\_ENABLED コマンド, A-64  
 SHOW RESETS\_STATS\_INTERVAL コマンド, A-65  
 SHOW RULES コマンド, A-88  
 SHOW SAVE\_CONFIG\_INTERVAL コマンド, A-65  
 SHOW SAVE\_CONFIG\_ON\_STOP コマンド, A-19,  
     A-66  
 SHOW SERVER コマンド, A-66  
 SHOW STARTUP\_WAITTIME コマンド, A-19  
 SHOW STATUS コマンド, A-66  
 SHOW SYSTEM\_QUERIES コマンド, A-67  
 SHOW TRACE\_FILE\_NAME コマンド, A-68  
 SHOW TRACE\_LEVEL コマンド, A-68  
 SHOW TRC\_DIRECTORY コマンド, A-19  
 SHOW TRC\_FILE コマンド, A-19  
 SHOW TRC\_LEVEL コマンド, A-19  
 SHOW USE\_PLUGANDPLAY コマンド, A-19  
 SHOW VERSION コマンド, A-69  
 SHOW\_TNS\_INFO パラメータ, C-70  
 SHOW コマンド  
     CMCTL, A-84  
     LSNRCTL, A-19  
     NAMESCTL, A-60  
 SHUTDOWN コマンド  
     CMCTL, A-89  
     NAMESCTL, A-70  
 SID、リスナーの構成, 7-11  
 SID\_DESC パラメータ, C-49

SID\_LIST\_listener\_name パラメータ  
     Oracle Enterprise Manager  
         listener.ora 要件, 7-10  
 SID\_LIST\_listener\_name パラメータ, C-49  
     Oracle Enterprise Manager  
         listener.ora 要件, C-47  
     異種サービス, 8-43  
     外部プロシージャ, 8-41  
 SID\_NAME パラメータ, 7-11, 8-41, 8-43, C-48  
 SID パラメータ, C-45  
 SNA, 4-4  
 SOURCE\_ROUTE パラメータ, 8-4, 8-7, 8-28, 8-30,  
     8-34, C-39  
 SPAWN コマンド, A-20  
 SPX プロトコル  
     SERVICE パラメータ, B-3  
     アドレスのパラメータ, B-3  
     説明, 4-3  
 SQLNET\_CRYPTO\_CHECKSUM\_TYPE\_CLIENT パラ  
     メータ, C-15  
 SQLNET\_CRYPTO\_CHECKSUM\_TYPE\_SERVER パラ  
     メータ, C-15  
 SQLNET.ALTERNATE\_PORT パラメータ, C-21  
 SQLNET.AUTHENTICATION\_GSSAPI\_SERVICE パ  
     ラメータ, C-14  
 SQLNET.AUTHENTICATION\_KERBEROS5\_  
     SERVICE パラメータ, C-13  
 SQLNET.AUTHENTICATION\_SERVICES パラメータ,  
     C-13  
 SQLNET.CLIENT\_AUTHENTICATION パラメータ,  
     C-24, C-52  
 SQLNET.CLIENT\_REGISTRATION パラメータ, 8-21,  
     C-14  
 SQLNET.CRYPTO\_CHECKSUM\_CLIENT パラメータ,  
     C-14  
 SQLNET.CRYPTO\_CHECKSUM\_SERVER パラメータ,  
     C-15  
 SQLNET.CRYPTO\_SEED パラメータ, C-16  
 SQLNET.ENCRYPTION\_CLIENT パラメータ, C-16  
 SQLNET.ENCRYPTION\_SERVER パラメータ, C-16  
 SQLNET.ENCRYPTION\_TYPES\_CLIENT パラメータ,  
     C-17  
 SQLNET.ENCRYPTION\_TYPES\_SERVER パラメータ,  
     C-17  
 SQLNET.EXPIRE\_TIME パラメータ, 8-21, C-18  
 SQLNET.IDENTIX\_FINGERPRINT\_DATABASE\_  
     PASSWORD パラメータ, C-19



SQLNET.IDENTIX\_FINGERPRINT\_DATABASE\_  
     USER パラメータ, C-18  
 SQLNET.IDENTIX\_FINGERPRINT\_DATABASE パラ  
     メータ, C-18  
 SQLNET.IDENTIX\_FINGERPRINT\_METHOD パラ  
     メータ, C-19  
 SQLNET.KERBEROS5\_CC\_NAME パラメータ, C-19  
 SQLNET.KERBEROS5\_CLOCKSKEW パラメータ,  
     C-20  
 SQLNET.KERBEROS5\_CONF パラメータ, C-20  
 SQLNET.KERBEROS5\_KEYTAB パラメータ, C-20  
 SQLNET.KERBEROS5\_REALMS パラメータ, C-20  
 sqlnet.log ファイル, 12-15  
     サンプルのクライアント側エラー・スタック,  
         12-15  
 sqlnet.ora ファイル  
     構成パラメータのリファレンス, C-4, C-31  
     パラメータ  
         AUTOMATIC\_IPC, C-75  
         BEQUEATH\_DETACH, 10-10, C-4  
         DAEMON.TRACE\_DIRECTORY, C-5  
         DAEMON.TRACE\_LEVEL, C-5  
         DAEMON.TRACE\_MASK, C-5  
         DISABLE\_OOB, C-6  
         LOG\_DIRECTORY\_CLEINT, C-6  
         LOG\_DIRECTORY\_SERVER, C-6  
         LOG\_FILE\_CLIENT, C-7  
         LOG\_FILE\_SERVER, C-7  
         NAMES\_DIRECTORY\_PATH, C-8  
         NAMESCTL\_INTERNAL\_ENCRYPT\_  
             PASSWORD, C-10  
         NAMESCTL.INTERNAL\_USE, C-10  
         NAMESCTL.NO\_INITIAL\_SERVER, C-11  
         NAMESCTL.NOCONFIRM, C-11  
         NAMESCTL.SERVER\_PASSWORD, C-11  
         NAMESCTL.TRACE\_DIRECTORY, C-12  
         NAMESCTL.TRACE\_FILE, C-12  
         NAMESCTL.TRACE\_LEVEL, C-11  
         NAMESCTL.TRACE\_UNIQUE, C-12  
         NAMES.DCE\_PREFIX, C-7  
         NAMES.DEFAULT\_DOMAIN, 8-15, C-7  
         NAMES.DIRECTORY\_PATH, 8-16  
         NAMES.INITIAL\_RETRY\_TIMEOUT, C-8  
         NAMES.MAX\_OPEN\_CONNECTIONS, C-8  
         NAMES.MESSAGE\_POOL\_START\_SIZE, C-9  
         NAMES.NDS.NAME.CONTEXT, C-9  
         NAMES.NIS.META\_MAP, C-9  
         NAMES.PREFERRED\_SERVERS, 8-24, C-9  
         NAMES.REQUEST\_RETRIES, C-10  
         OSS.MY.WALLET, C-12  
         SQLNET\_CCRYPTO\_CHECKSUM\_TYPE\_  
             CLIENT, C-15  
         SQLNET\_CCRYPTO\_CHECKSUM\_TYPE\_  
             SERVER, C-15  
         SQLNET.ALTERNATE\_PORT, C-21  
         SQLNET.AUTHENTICATION\_GSSAPI\_  
             SERVICE, C-14  
         SQLNET.AUTHENTICATION\_KERBEROS5\_  
             SERVICE, C-13  
         SQLNET.AUTHENTICATION\_SERVICES,  
             C-13  
         SQLNET.CLIENT\_AUTHENTICATION, C-24  
         SQLNET.CLIENT\_REGISTRATION, C-14  
         SQLNET.CRYPTO\_CHECKSUM\_CLIENT, C-14  
         SQLNET.CRYPTO\_CHECKSUM\_SERVER,  
             C-15  
         SQLNET.CRYPTO\_SEED, C-16  
         SQLNET.ENCRYPTION\_SERVER, C-16  
         SQLNET.ENCRYPTION\_TYPES\_CLIENT, C-17  
         SQLNET.ENCRYPTION\_TYPES\_SERVER, C-17  
         SQLNET.ENCYRPTION\_CLIENT, C-16  
         SQLNET.EXPIRE\_TIME, C-18  
         SQLNET.IDENTIX\_FINGERPRINT\_  
             DATABASE, C-18  
         SQLNET.IDENTIX\_FINGERPRINT\_  
             DATABASE\_PASSWORD, C-19  
         SQLNET.IDENTIX\_FINGERPRINT\_  
             DATABASE\_USER, C-18  
         SQLNET.IDENTIX\_FINGERPRINT\_METHOD,  
             C-19  
         SQLNET.KERBEROS5\_CC\_NAME, C-19  
         SQLNET.KERBEROS5\_CLOCKSKEW, C-20  
         SQLNET.KERBEROS5\_CONF, C-20  
         SQLNET.KERBEROS5\_KEYTAB, C-20  
         SQLNET.KERBEROS5\_REALMS, C-20  
         SQLNET.RADIUS\_ALTERNATE, C-21  
         SQLNET.RADIUS\_ALTERNATE\_RETRIES,  
             C-21  
         SQLNET.RADIUS\_AUTHENTICATION, C-22  
         SQLNET.RADIUS\_AUTHENTICATION\_  
             INTERFACE, C-22  
         SQLNET.RADIUS\_AUTHENTICATION\_PORT,  
             C-22



SQLNET.RADIUS\_AUTHENTICATION\_RETRIES, C-22  
 SQLNET.RADIUS\_AUTHENTICATION\_TIMEOUT, C-23  
 SQLNET.RADIUS\_CHALLENGE\_RESPONSE, C-23  
 SQLNET.RADIUS\_SECRET, C-23  
 SQLNET.RADIUS\_SEND\_ACCOUNTING, C-23  
 SSL\_CIPHER\_SUITES, C-24  
 SSL\_VERSION, C-24  
 TNSPING.TRACE\_DIRECTORY, C-25  
 TNSPING.TRACE\_LEVEL, C-25  
 TRACE\_DIRECTORY\_CLIENT, C-25  
 TRACE\_DIRECTORY\_SERVER, C-25  
 TRACE\_FILE\_CLIENT, C-26  
 TRACE\_FILE\_SERVER, C-26  
 TRACE\_FILEN\_CLIENT, C-26  
 TRACE\_FILEN\_SERVER, C-26  
 TRACE\_FILENO\_CLIENT, C-27  
 TRACE\_FILENO\_SERVER, C-27  
 TRACE\_LEVEL\_CLIENT, C-27  
 TRACE\_LEVEL\_SERVER, C-28  
 TRACE\_TIMESTAMP\_CLIENT, C-28  
 TRACE\_TIMESTAMP\_SERVER, C-28  
 TRACE\_UNIQUE\_CLIENT, C-29  
 USE\_CMN, 8-19, C-30  
 USE\_DEDICATED\_SERVER, 8-19, C-31  
 sqlnetpid.trc ファイル, C-29  
 SQLNET.RADIUS\_ALTERNATE\_RETRIES パラメータ, C-21  
 SQLNET.RADIUS\_ALTERNATE パラメータ, C-21  
 SQLNET.RADIUS\_AUTHENTICATION\_INTERFACE パラメータ, C-22  
 SQLNET.RADIUS\_AUTHENTICATION\_PORT パラメータ, C-22  
 SQLNET.RADIUS\_AUTHENTICATION\_RETRIES パラメータ, C-22  
 SQLNET.RADIUS\_AUTHENTICATION\_TIMEOUT パラメータ, C-23  
 SQLNET.RADIUS\_AUTHENTICATION パラメータ, C-22  
 SQLNET.RADIUS\_CHALLENGE\_RESPONSE パラメータ, C-23  
 SQLNET.RADIUS\_SECRET パラメータ, C-23  
 SQLNET.RADIUS\_SEND\_ACCOUNTING パラメータ, C-23

sqlnet.trc ファイル, 12-28, C-26  
 SRC パラメータ, 8-32, C-72  
 SRV パラメータ, 8-32, C-72  
 SSL\_CIPHER\_SUITES パラメータ, C-24  
 SSL\_VERSION パラメータ, C-24  
 SSL 認証方式  
   「Oracle Advanced Security」を参照  
 SSL プロトコル付き TCP/IP  
   HOST パラメータ, B-4  
   PORT パラメータ, B-4  
   アドレスのパラメータ, B-4  
   説明, 4-2  
 START CMAN コマンド, 11-7  
 START\_CLIENT\_CACHE コマンド, 11-5, A-72  
 STARTUP\_WAITTIME\_listener\_name パラメータ, C-53  
 STARTUP コマンド  
   NAMESCTL, A-73  
 START コマンド  
   CMCTL, A-90  
   LSNRCTL, 7-4, 11-6, A-20  
   NAMESCTL, 6-50, 6-54, 11-4, A-70  
 STATS コマンド, A-91  
 STATUS コマンド  
   CMCTL, A-93  
   LSNRCTL, 11-5, A-22  
   NAMESCTL, A-74  
 STOPNOW コマンド  
   CMCTL, A-94  
 STOP コマンド  
   CMCTL, A-94  
   LSNRCTL, 11-5, A-24  
   NAMESCTL, A-74  
 svr\_pid.trc ファイル, 12-28, C-26  
 System Network Architecture (SNA), 4-4

## T

TAF  
   「透過的アプリケーション・フェイルオーバー (TAF)」を参照  
 TCP/IP プロトコル  
   HOST パラメータ, B-4  
   PORT パラメータ, B-4  
   アドレスのパラメータ, B-4  
   説明, 4-2  
 TCP.NODELAY パラメータ, C-73



- tftp ユーティリティ, 10-8
- TICKS 属性, 9-6
- TIMED\_QUERY コマンド, A-75
- TIMEOUT パラメータ, 2-16, 7-14
- 「Timeout」フィールド, 7-14
- TNS
  - 説明, 3-2
  - 利点, 3-2
- 「TNS Time Out Value」オプション, 8-21
- TNS\_ADMIN 環境変数, 12-3, 12-5
- TNS\_ADMIN パラメータ, 12-3, 12-5
- TNSAPIA ファイル, 10-6
- TNSAPI.DLL ファイル, 10-6
- TNSAPI.H ファイル, 10-6
- TNSAPI.LIB ファイル, 10-6
- tnsnames.ora ファイル
  - 構成, 6-4
  - 構成パラメータのリファレンス, C-31 ~ C-45
  - ディレクトリ・サーバーへのエントリのエクスポート, 6-28
  - パラメータ
    - ADDRESS, C-37
    - ADDRESS\_LIST, C-37
    - BACKUP, C-42
    - BACKUP パラメータ, 8-11
    - CONNECT\_DATA, C-40
    - DELAY, 8-12, C-42
    - DESCRIPTION, C-36
    - DESCRIPTION\_LIST, C-37
    - FAILOVER, 8-4, C-38
    - FAILOVER\_MODE, 8-11, C-41
    - GLOBAL\_NAME, 8-9, C-43
    - HS, 8-8, C-43
    - INSTANCE\_NAME, 8-8, C-43, C-82
    - LOAD\_BALANCE, 8-4, C-38, C-83
    - METHOD, 8-12, C-42
    - RDB\_DATABASE, 8-9, C-44
    - RETRIES, 8-12, C-42
    - SDU, 8-8, C-39
    - SERVER, C-44
    - SERVICE\_NAME, C-45, C-83
    - 「SID」, C-45
    - SOURCE\_ROUTE, 8-4, C-39
    - TYPE, 8-12, C-42
    - TYPE\_OF\_SERVICE, 8-9, C-40
- リリース 8.1、変更, C-81, C-83
- FAILOVER パラメータ, C-78
- LOAD\_BALANCE パラメータ, C-77
- SERVICE\_NAME パラメータ, C-76
- SOURCE\_ROUTE パラメータ, C-79
- ネストされたアドレス, C-79
- TNSPING.TRACE\_DIRECTORY パラメータ, 12-32, C-25
- TNSPING.TRACE\_LEVEL パラメータ, 12-32, C-25
- TNSPING ユーティリティ, 11-14
  - TRCROUTE ユーティリティとの比較, 11-17
- TOTAL\_REFUSED, A-92
- TOTAL\_RELAYS, A-92
- TP\_NAME パラメータ, B-3
- TPN パラメータ, B-3
- TRACE\_DIRECTORY\_CLIENT パラメータ, 12-30, C-25
- TRACE\_DIRECTORY\_listener\_name パラメータ, C-53
- TRACE\_DIRECTORY\_SERVER パラメータ, 12-31, C-25
- TRACE\_DIRECTORY パラメータ, 12-33, 12-34, C-70
- TRACE\_FILE\_CLIENT パラメータ, 12-30, C-26
- TRACE\_FILE\_listener\_name パラメータ, C-53
- TRACE\_FILE\_SERVER パラメータ, 12-31, C-26
- TRACE\_FILELEN\_CLIENT パラメータ, C-26
- TRACE\_FILELEN\_SERVER パラメータ, C-26
- TRACE\_FILELEN パラメータ, C-71
- TRACE\_FILE\_listener\_name パラメータ, C-53
- TRACE\_FILENO\_CLIENT パラメータ, C-27
- TRACE\_FILENO\_listener\_name パラメータ, C-54
- TRACE\_FILENO\_SERVER パラメータ, C-27
- TRACE\_FILE パラメータ, 12-33
- TRACE\_LEVEL\_CLIENT パラメータ, 12-30, C-27
- TRACE\_LEVEL\_listener\_name パラメータ, C-54
- TRACE\_LEVEL\_SERVER パラメータ, 12-31, C-28
- TRACE\_LEVEL パラメータ, 12-32
- TRACE\_TIMESTAMP\_CLIENT パラメータ, C-28
- TRACE\_TIMESTAMP\_listener\_name パラメータ, C-55
- TRACE\_TIMESTAMP\_SERVER パラメータ, C-28
- TRACE\_TIMESTAMP パラメータ, C-71
- TRACE\_UNIQUE\_CLIENT パラメータ, C-29
- TRACE コマンド, A-24
- TRACING パラメータ, 12-34, C-70
- Transparent Network Substrate
  - 「TNS (Transparent Network Substrate)」を参照
- Transparent Network Substrate (TNS)
  - 説明, 3-2
  - 利点, 3-2



TRCROUTE ユーティリティ  
旧バージョンの SQL\*Net 使用, 11-17  
説明, 11-17  
パフォーマンスへの影響, 11-17  
要件, 11-17  
「Type of Service」フィールド, 8-46  
TYPE\_OF\_SERVICE パラメータ, 8-9, 8-46, C-40  
TYPE パラメータ, 8-12, C-42

## U

---

UNIX  
シグナル処理と Net8 の併用, 10-10  
シグナル・ハンドラでの Oracle のオペレーティング・システム依存コール, 10-9  
シグナル・ハンドラの欠点, 10-9  
シグナル・ハンドラの使用, 10-9  
UNREGISTER コマンド, A-75  
USE\_ASYNC\_CALL パラメータ, C-71  
USE\_CMAN パラメータ, 8-19, 8-27, 8-30, 8-33, 8-34, C-30  
USE\_DEDICATED\_SERVER パラメータ, 8-19, C-31  
USE\_PLUG\_AND\_PLAY\_listener\_name パラメータ, 6-59, C-55

## V

---

V\$SESSION 表, 8-14  
V1ADD.NPO.OMD レコード, 11-12, A-42  
VERSION コマンド  
CMCTL, A-95  
LSNRCTL, A-25  
NAMECTL, A-76

## W

---

Windows NT のシステム固有な認証, 4-21

## Y

---

ypserv プログラム, 6-72

## あ

---

アーキテクチャ  
Net8, 3-1  
Oracle Connection Manager, 4-4

Oracle Names, 4-8  
接続プーリング, 2-10  
専用サーバー, 2-11  
マルチスレッド・サーバー, 2-6  
アクセス制御リスト (ACL)  
説明, 2-44  
アドレス、構成, 7-5  
アプリケーション  
Net8 OPEN による開発, 10-7  
Net8 OPEN のサンプル・プログラム, 10-8  
ユーザー定義を使用するためのシステム構成, 10-7  
アプリケーション・プログラム・インタフェース (API)、Oracle 以外のデータ・ソース用, 10-2

## い

---

異種サービス  
接続の構成, 8-43  
説明, 8-43  
イタリック体、定義, xviii  
委任管理リージョン  
Oracle Names, 4-16  
図, 4-17  
ドメイン, 6-66  
ルートの下, 4-18  
インストレーション  
デフォルトの構成  
ローカル・ネーミング, 6-5  
ディレクトリ・ネーミング, 6-14  
Oracle Names, 6-44  
デフォルトの構成  
リスナー, 7-3

## え

---

エラー・スタック  
サンプル, 12-15  
説明, 12-13  
代表的なレイヤー, 12-14  
ログ・ファイルのエントリ, 12-20  
エラー・メッセージ  
Net8 OPEN の 20002-20021, 10-8  
NNL-00018, 6-50  
NNL-00024, 6-50  
ORA-00508, 12-8  
ORA-1017, 12-3  
ORA-1034, 12-3



ORA-12203, 12-8, 12-12  
    サンプルのエラー・スタック, 12-15  
ORA-12533, 12-10  
ORA-12535, 12-12  
ORA-12538, 12-8  
ORA-12547, 12-12  
ORA-12560, 12-11  
ORA-3121, 12-11  
    オラクル社カスタマ・サポート・センターへの  
        連絡, 12-52  
解決, 12-1, 12-52  
    トレース・データの例, 12-38  
    トレース・ファイル, 12-38  
    ログ・ファイルによる追跡, 12-20  
エンタープライズ・ユーザーのセキュリティ, 2-36  
エントリ、定義, 2-35

## お

---

大文字、定義, xviii  
オブジェクト・クラス  
    orclDBServer, 2-47, D-2  
    orclNetAddress, 2-47, D-3  
    orclNetAddressList, 2-47, D-4  
    orclNetDescription, 2-47, D-2  
    orclNetDescriptionList, 2-47, D-3  
    orclNetService, 2-47, D-2  
    説明, 2-46  
オラクル社カスタマ・サポート・センター、連絡、  
    12-52

## か

---

解決  
    エラー、「トラブルシューティング」も参照  
階層ネーミング・モデル  
    図, 4-15  
    説明, 4-14  
外部ネーミング  
    CDS, 2-33, 4-21, 6-71, 8-18  
    Cell ディレクトリ・サービス, 2-33, 4-21, 6-71,  
        8-18  
    NDS, 2-33, 4-21, 6-71, 8-18  
    NIS, 2-32, 4-21, 6-71, 8-18  
    Novell ディレクトリ・サービス, 2-33, 6-71, 8-18,  
        4-21  
    機能, 4-21

構成, 6-71, 6-78  
推奨環境, 5-4  
接続の確立, 2-33  
説明, 2-32  
    ネットワーク情報サービス, 2-32, 4-21, 6-71,  
        8-18  
    利点および不利な点, 5-4  
外部プロシージャ  
    extproc プログラム, 8-39  
    構成, 8-39  
    説明, 8-39  
仮想サーキット, 2-6  
監査証跡  
    情報を使用するためのスクリプト, 12-22  
    説明, 12-21  
管理コンテキスト  
    説明, 2-36  
    選択, 6-16  
管理リージョン  
    Oracle Names, 4-10  
    委任, 4-16  
    委任、ルートの下, 4-18  
    説明, 4-10  
    定義, 6-44  
    ドメインの委任, 6-66  
    ルート, 4-16

## き

---

キーワード構文規則、構成ファイル, C-2  
キーワードの値、ネットワーク・キャラクタ・セット、  
    C-3  
規則、構成ファイルの構文, C-2  
起動  
    Net8 Configuration Assistant, 4-23  
    Net8 コンポーネント, 11-3  
    Oracle Connection Manager, 11-7  
    Oracle Names クライアント・キャッシュ, 11-4  
    Oracle Names サーバー, 6-49, 11-4  
    Oracle Net8 Assistant, 4-24  
    インスタンス  
        ディスパッチャ, 9-7  
        マルチスレッド・サーバー, 9-3  
データベース, 11-6  
リスナー, 11-5



## 機能、新規

- cman.ora ファイル, C-83 ~ C-84
- Direct Hand-Off, xvi
- listener.ora ファイル, C-80
- Net8 Configuration Assistant, xvii
- Oracle Connection Manager, xvi
- SERVICE\_NAME パラメータ, xvii
- SSL 付き TCP/IP, xvii
- tnsnames.ora ファイル, C-81 ~ C-83
- クライアント・ロード・バランス, xvi
- サービス登録, xvi
- サービス・ネーミング, xv
- 接続時フェイルオーバー, xvi
- 接続時ロード・バランス化, xvi
- ディレクトリ・ネーミング, xv
- キャラクタ・セット
  - ネット・サービス名, C-4
  - ネットワーク、キーワードの値, C-3
- キューサイズ, 5-5, 7-10
- 共有サーバー
  - 推奨環境, 5-5
  - 「マルチスレッド・サーバー」も参照
  - 利点および不利な点, 5-5

## く

---

- クライアント・アプリケーション・レイヤー、説明, 3-5
- クライアント接続
  - 構文, 11-7
  - 方法, 11-8
- クライアントの構成
  - Oracle Connection Manager アドレス, 8-34
  - Oracle Connection Manager 機能, 8-33
  - Oracle Connection Manager によるアクセス制御, 8-33
  - Oracle Names サーバーの検出, 6-55
  - Oracle Rdb 接続, 8-46
  - sqlnet.log ファイル, 12-17
  - 共有サーバー, 9-9
  - 接続時フェイルオーバー, 2-18
  - 接続集中化, 8-27
  - デフォルト・ドメイン, 8-15
  - 透過的アプリケーション・フェイルオーバー (TAF), 2-19
  - トレース・ファイル, 12-30
  - 複数プロトコル・サポート, 4-8, 8-30

- マルチスレッド・サーバー, 9-9
- マルチスレッド・サーバー構成の変更, 9-9
- 優先 Oracle Names サーバー, 8-24
- リスナー間での要求のロード・バランス, 2-18
- リスナー間のロード・バランス要求, 8-3
- リスナーのロード・バランス要求, C-38
- ローカル・ネーミング, 6-12
- ログ・ファイル, 12-17
- クライアントのテスト
  - 3GL 使用, 11-9
  - Oracle Net8 Assistant 使用, 11-18
  - OS 使用, 11-8
  - アプリケーション使用, 11-9
  - 接続, 11-14
  - 特殊コマンド使用, 11-9
- クライアントのトラブルシューティング, 12-3
- クライアント・ロード・バランス, xvi, 2-18, C-38
  - 構成, 8-3
  - ネットワーク・パフォーマンスを向上するために使用, 5-7
- グローバル・データベース名
  - 説明, 7-11, C-47
  - 定義, 6-3
  - リスナーの構成, 7-11, C-47
- グローバル・データベース・リンク, 4-13, 6-62
  - リンク修飾子の追加, 6-64
- グローバル・データベース・リンク修飾子, 6-62
- グローバル・データベース・リンク
  - Oracle Names による登録, 6-63

## け

---

- 計画の概要, 5-2

## こ

---

- 構成
  - Net8 OPEN, 10-6
  - Net8 アクセス制御, 8-31
  - Oracle Names, 6-43 ~ 6-67
  - Oracle Names サーバー, 6-46
  - Oracle Names サーバー間でのデータのレプリケート, 6-50
  - Oracle Names の複数ドメイン, 6-49, 6-52
  - Oracle Names のリージョン・データベース, 6-46, 6-49
  - Oracle Rdb データベース接続, 8-46



異種サービス接続, 7-10, 8-43  
外部ネーミング, 6-71 ~ 6-78  
外部プロシージャ, 8-39  
概要, 6-2  
クライアント  
    Oracle Connection Manager, 8-33  
    Oracle Connection Manager によるアクセス  
        制御, 8-33  
    Oracle Names, 6-55  
    接続時フェイルオーバー, 2-18, 8-3  
    接続集中化, 8-27  
    デフォルト・ドメイン, 8-15  
    透過的アプリケーション・フェイルオーバー,  
        2-19  
    複数プロトコル・サポート, 8-30  
    プロファイルのネーミング・メソッド, 8-16  
    ローカル・ネーミング, 6-12  
サーバー  
    Oracle Names, 6-55  
    接続集中化, 8-26  
    ローカル・ネーミング, 6-6  
事前生成済専用サーバー, 7-13  
接続集中化, 8-25  
接続プーリング, 9-8  
ディレクトリ・ネーミング, 6-13 ~ 6-42  
ドメイン, デフォルト, 8-24  
ネーミング・メソッド, 6-1 ~ 6-79  
ネット・サービス名  
    Oracle Names, 6-60  
    外部ネーミング, 6-71  
    ディレクトリ・ネーミング, 6-20  
    ローカル・ネーミング, 6-6  
複数のアドレス, 8-2  
複数プロトコル・サポート, 8-28  
プロトコル・アドレス, 7-5, B-1  
プロファイルでの接続要求, 8-18  
ホスト・ネーミング, 6-68 ~ 6-71  
ユーザー定義のアプリケーションを使用するための  
    システム, 10-7  
リスナー, 7-1 ~ 7-16  
    Oracle Names, 6-59  
    ディレクトリ・ネーミング, 6-25  
    ホスト・ネーミング, 6-68  
    ローカル・ネーミング, 6-13  
ローカル管理, 6-3  
ローカル・ネーミング, 6-4 ~ 6-13

構成ファイル  
    cman.ora, C-65, C-72  
    listener.ora, C-45, C-55  
    names.ora, C-55, C-65  
    tnsnames.ora, C-31, C-45  
    構文規則, C-2  
    プロトコル・アドレス, B-1  
構成ファイルのコメント, C-3  
構文  
    CMCTL, 11-3  
    LSNRCTL, 11-2  
    NAMESTCTL, 11-2  
    構成ファイルの規則, C-2  
    接続識別子, 11-7  
ゴシック体の小文字、定義, xviii  
子プロセスの終了, 10-10

## さ

---

サーバー  
    サーバー / サーバー間の対話、説明, 3-8  
    スタック通信, 3-7  
    接続の切断, 2-5  
    専用, 9-2  
    データベースの起動, 11-6  
    マルチスレッド  
        専用～との比較, 9-2  
    ループバック・テスト, 12-3  
サーバー接続  
    構文, 11-7  
    方法, 11-8  
サーバー・テスト, 11-13  
サーバーの構成  
    Oracle Names サーバーの検出, 6-55  
    Oracle Names のリージョン・データベース, 6-46  
    外部プロシージャ, 8-42  
    事前生成済専用サーバー, 7-13  
    接続集中化, 8-26  
    トレース・ファイル, 12-30  
    マルチスレッド・サーバー, 9-3  
    マルチスレッド・サーバーによるリソース割当て,  
        9-9  
    ローカル・ネーミング, 6-6  
    ログ・ファイル, 12-17  
サーバーのトラブルシューティング, 12-2  
サービス  
    サービス名, 6-2



- 接続, 1-4
- 説明, 1-4
- サービス登録, xvi
  - SERVICE\_DIED イベント, 12-22
  - SERVICE\_REGISTER イベント, 12-22
  - SERVICE\_UPDATE イベント, 12-22
  - イベントの listener.log 格納, 12-22
  - 構成, 2-21
  - 接続時フェイルオーバー, 2-21
  - 接続時ロード・バランス化, 2-21, 2-23
  - 説明, 2-19
  - 利点, 2-21
- サービス名
  - アドレスの格納、NDS, 6-76
  - キャラクタ・セットのキーワード値, C-4
  - 説明, 6-2

## し

---

- 識別名 (DN)、定義, 2-35
- シグナル処理
  - Net8 との併用, 10-10
  - Oracle のオペレーティング・システム依存コールの要件, 10-10
  - UNIX の考慮事項, 10-9
- 事前生成済専用サーバー, 2-14
  - POOL\_SIZE パラメータ, 2-15
  - PRESPAWN\_MAX パラメータ, 2-16
  - TIMEOUT パラメータ, 2-16
  - 構成, 7-13
  - 推奨環境, 5-5
  - 利点および不利な点, 5-5
- 事前に起動される専用サーバー・プロセス
  - 「事前生成済専用サーバー・プロセス」を参照
- 事前生成済専用サーバー
  - 数, 7-13
- 集中化されたネーミング
  - 推奨環境, 5-3
  - 説明, 2-31
  - 利点および不利な点, 5-3
- 集中化された命名, 6-3
  - 接続の確立, 6-3
- 省略記号、定義, xviii
- 初期化パラメータ・ファイル
  - INSTANCE\_NAME パラメータ, 2-28
  - LOCAL\_LISTENER パラメータ, 7-6
  - MTS\_DISPATCHERS パラメータ, 9-3

- SERVICE\_NAMES パラメータ, 2-28, 9-6
- LOCAL\_LISTENER パラメータ, 7-8, 7-9
- 新機能
  - Direct Hand-Off, xvi
  - Net8 Configuration Assistant, xvii
  - Oracle Connection Manager, xvi
  - SERVICE\_NAMES, xvii
  - SSL 付き TCP/IP, xvii
  - クライアント・ロード・バランス, xvi
  - サービス登録, xvi
  - サービス・ネーミング, xv
  - 接続時フェイルオーバー, xvi
  - 接続時ロード・バランス化, xvi
  - ディレクトリ・ネーミング, xv
  - シングル・サインオン、説明, 4-22
  - シングル・ドメイン・ネーミング・モデル, 4-14
- 診断
  - エラー、「トラブルシューティング」も参照
  - クライアントの問題, 12-3

## す

---

- スキーマの拡張、NDS, 6-75
- スタック通信
  - NI, 3-6
  - NS, 3-6
  - OCI, 3-5
  - Oracle コール・インタフェース, 3-5
  - Oracle プロトコル, 3-7
  - TTC, 3-6
  - クライアント・アプリケーション, 3-5
  - サーバー側の対話, 3-7
  - 説明, 3-2
  - ネットワーク・インタフェース, 3-6
  - ネットワーク・セッション, 3-6

## せ

---

- 制御ユーティリティ
  - CMCTL, 4-25, A-77, A-78, A-95
  - LSNRCTL, 4-24, A-2, A-3, A-25
  - NAMESCTL, 4-25, A-25, A-27, A-77
  - 説明, A-1
- セキュリティ, A-3, A-26
  - LSNRCTL ユーティリティ, A-3
  - NAMESCTL ユーティリティ, A-26



- セッション・データ・ユニット
  - 構成, 8-8
  - ネットワーク・パフォーマンスを向上するために調整, 5-7
- セッションの確立
  - Oracle Names の使用, 2-31
  - 外部ネーミング, 2-33
  - 共有サーバーによる, 2-7, 2-13
  - 集中化されたネーミングの使用, 2-31
  - ディレクトリ・ネーミングの使用, 2-30
  - ホスト・ネーミングの使用, 2-32
  - ローカル・ネーミングの使用, 2-30
- セッション・レイヤー
  - 説明, 3-4
- 接続, 2-8
  - bequeath, 2-17
  - NDS 外部ネーミング, 6-75
  - NIS 外部ネーミング, 6-71
  - Novell ディレクトリ・サービス外部ネーミング, 6-75
  - Oracle Connection Manager, 1-3, 8-18
  - Oracle Names, 6-67
  - Oracle Rdb データベース, 8-46
  - 異種サービス, 8-43
  - エラー回避のためのリスナー・キューサイズの調整, 5-5, 7-10
  - 外部プロシージャ, 8-39
  - コンポーネントの起動, 11-3
  - サービス, 1-4
  - 接続文字列, 入力, 11-2, 11-7
  - 専用サーバー, 2-11, 8-18
  - ディレクトリ・ネーミング, 6-25
  - 同時、接続数の増加, 7-10
  - ネットワーク情報サービス外部ネーミング, 6-71
  - 方法, 11-8
  - ホスト・ネーミング, 6-71
  - マルチスレッド・サーバー, 2-6, 9-2
  - ルーティング, 1-3
  - ローカル・ネーミング, 6-13
- 接続記述子
  - 説明, 1-4, C-31
- 接続識別子
  - 構文, 11-7
- 接続時フェイルオーバー, xvi, C-38
  - listener.ora ファイル内の GLOBAL\_DBNAME パラメータ, 7-11, 8-4, C-38

- 構成, 8-3
- サービス登録, 2-21
- 接続集中化, 4-6, 8-25
  - 接続プーリングに対する相対的な利点, 5-6
  - ネットワーク・パフォーマンスを向上するために使用, 5-6
- 接続時ロード・バランス化, xvi, 2-23
  - サービス登録, 2-21
- 接続操作, 2-2
- 接続のルーティング, 1-3, 8-18
- 接続プーリング
  - 構成, 9-8
  - 接続集中化に対する相対的な利点, 5-6
  - ネットワーク・パフォーマンスを向上するために使用, 2-10, 5-5
  - マルチスレッド・サーバーの使用, 9-8
- 接続文字列, 11-2
- 切断
  - 異常終了, 2-5
  - サーバー, 2-5
  - 追加接続要求, 2-5
  - ユーザーによる実行, 2-5
- 専用サーバー
  - bequeathed セッション, 2-17
  - 事前生成済専用サーバー, 2-14
  - 接続のルーティング, 8-18
  - 説明, 2-11
  - マルチスレッド・サーバーとの比較, 9-2

## そ

- 操作
  - Net8, 2-6
  - 接続, 2-2
  - 例外, 2-6
- 相対識別名 (DN)、定義, 2-35
- 属性
  - c, 2-36
  - cn, 2-36
  - commonName, 2-36
  - country, 2-36
  - domain component, 2-36
  - o, 2-36
  - orclDescList, D-4
  - orclDescName, D-4
  - orclLoadBalance, D-4
  - orclNetAddrList, D-4



orclNetAddrString, D-4  
orclNetConnParamList, D-4  
orclNetFailover, D-4  
orclNetInstanceName, D-4  
orclNetProtocol, D-4  
orclNetProtocolStack, D-4  
orclNetSdu, D-4  
orclNetServiceName, D-4  
orclNetSourceRoute, D-4  
orclSid, D-5  
orclVersion, D-5  
organization, 2-36  
organizationUnitName, 2-36  
ou, 2-36  
定義, 2-35

## た

---

大カッコ、定義, xviii  
タイマーによる切断  
「デッド接続検出」を参照  
タイムスタンプ  
Oracle Connection Manager トレース・ファイル,  
C-71  
クライアント・トレース・ファイル, C-28  
サーバー・トレース・ファイル, C-28  
リスナー・トレース・ファイル, C-55  
多重化  
接続集中化で使用, 5-6

## ち

---

中カッコ、定義, xviii

## つ

---

通信、スタック, 3-2

## て

---

ディスパッチャ, 2-6  
数の設定, 9-7  
起動する数, 9-7  
説明, 2-6  
ディレクトリ・アクセス構成, 6-14  
ディレクトリ・サーバー  
ldapmodify ツール, 6-27

Microsoft の Active Directory, 2-35  
Novell ディレクトリ・サービス外部ネーミング,  
2-35  
Oracle Internet Directory, 2-35  
識別名 (DN), 2-35  
セキュリティ, 2-44  
ディレクトリ情報ツリー (DIT), 2-35  
パフォーマンス, 2-43  
ディレクトリ情報ツリー (DIT)、定義, 2-35  
ディレクトリ・ネーミング  
JDBC クライアント, 3-13  
ldapmodify ツール, 6-27  
ldap.ora ファイル  
作成, 6-14  
説明, 2-39  
パラメータ, C-74  
Oracle Database Configuration Assistant, 6-19  
Oracle コンテキスト, 2-36  
OracleDBCreators グループ, 2-39, 2-46  
OracleNetAdmins グループ, 2-39, 2-45, 6-19,  
6-20, 6-27  
OracleSecurityAdmins グループ, 2-46, 6-19  
Oracle スキーマ, 2-46  
アーキテクチャ, 2-35  
エクスポート  
Oracle Names からのオブジェクト, 6-31, 6-42  
Oracle Names からのオブジェクトのエク  
ポート, A-29  
tnsnames.ora ファイルからのネット・サービ  
ス名, 6-28  
オブジェクト・クラス, 2-46  
概要, 2-33  
管理コンテキスト, 2-36  
構成, 6-13, 6-42  
推奨環境, 5-3  
セキュリティ, 2-44  
接続, 6-25  
エントリの絶対名の使用, 2-40  
エントリの相対名の使用, 2-40  
接続の確立, 6-25  
説明, 2-30  
データベース・サービスへの接続, 2-39  
デフォルトの構成, 6-14  
認証方式  
SSL, 2-44  
強固, 2-44



- システム固有, 2-44
- 単純, 2-44
- リスナーの構成, 6-25
- 利点および不利な点, 5-3
- ディレクトリ・ネーミングの SSL 認証, 2-44
- ディレクトリ・ネーミングの強固な認証, 2-44
- ディレクトリ・ネーミングのシステム固有認証, 2-44
- ディレクトリ・ネーミングの単純認証, 2-44
- ディレクトリ・ネーミング
  - OracleDBCreators グループ, 6-19
  - ディレクトリ・アクセス構成, 6-14
- データ転送、最大化, 5-7
- データ転送の最大化、SDU サイズの調整, 5-7
- データベース
  - Oracle Names のリポジトリ, 4-11
  - 起動, 11-6
  - 分散, 2-2
- データベースに対するアクセス制御
  - protocol.ora, C-73
- データベースへのアクセス制御
  - Oracle Connection Manager, 8-31
- データベース・リンク, 4-13, 6-62
  - グローバル・データベース・リンク, 4-13
  - パブリック・データベース, 4-13
  - プライベート・データベース, 4-13
- テスト
  - Oracle Connection Manager, 11-13
  - Oracle Names サーバー, 11-10
  - クライアントの構成
    - 3GL 使用, 11-9
    - Oracle Net8 Assistant 使用, 11-18
    - OS 使用, 11-8
    - TRCROUTE 使用, 11-17
    - TNSPING 使用, 11-14
    - アプリケーション使用, 11-9
    - 特殊コマンド使用, 11-9
  - サーバーの構成
    - Oracle Net8 Assistant 使用, 11-13
  - 制御ユーティリティ, 11-2
  - ネットワーク接続性, 11-14
  - ネットワーク・オブジェクト, 11-11
  - リスナーの構成, 11-13
- デッド接続検出
  - SQLNET.EXPIRE\_TIME パラメータ, C-18
  - 制約, 8-21
  - 説明, 2-5

- デフォルト・ネットワーク・ドメイン, 8-15
- Oracle Names, 4-15

## と

- 透過的アプリケーション・フェイルオーバー
  - listener.ora ファイル内の GLOBAL\_DBNAME パラメータ, 7-11
- 透過的アプリケーション・フェイルオーバー (TAF)
  - listener.ora ファイル内の GLOBAL\_DBNAME パラメータ, 8-12
  - 構成, 8-11
  - パラメータ, C-41
  - 概要, 2-19
- 動的検出オプション, 4-19
- 登録
  - Oracle Names によるグローバル・データベース・リンク, 6-62
  - Oracle Names によるグローバル・データベース・リンク修飾子, 6-62
  - Oracle Names によるネット・サービス名, 6-60
  - Oracle Names による別名, 6-66
  - Oracle Names のあるオブジェクト, 6-59
- ドメイン
  - 委任, 6-66
  - 関連する複数の階層, 4-15
  - シングル・ドメイン・ネーミング構造、図, 4-14
  - 説明, 4-11, 6-44
  - デフォルト・ネットワーク, 4-15
  - 複数の階層, 6-49, 6-52
  - ルート管理リージョンでの要件, 4-16
- トラブルシューティング, 12-1 ~ 12-52
  - オラクル社カスタマ・サポート・センターへの連絡, 12-52
  - クライアント, 12-3
  - サーバー, 12-2
  - 質問, 12-12
  - トレース・ファイル, 12-12
  - ループバック・テスト, 12-3
  - ログ・ファイル, 12-12
  - ログ・ファイルの使用, 12-20
- トランスポート・レイヤー、説明, 3-4
- トレース
  - クライアント, C-26
  - サーバー, C-26



## タイムスタンプ

Oracle Connection Manager トレース・ファイル, C-71

クライアント・トレース・ファイル, C-28

サーバー・トレース・ファイル, C-28

リスナー・トレース・ファイル, C-55

## ファイルの循環

Oracle Connection Manager, C-71

クライアント, C-27

サーバー, C-27

リスナー, C-54

## トレース・ファイル

cmadm\_pid.trc, 12-34

cmadmpid.trc, 12-34

cman\_pid.trc, 12-34

cmanpid.trc, 12-34

listener.trc, 12-32

namespid.trc, 12-34

names.trc, 12-33

sqlnetpid.trc ファイル, C-29

sqlnet.trc, C-26

エラー・メッセージ情報, 12-38

エラー・メッセージの例, 12-38

クライアントの sqlnet.trc, 12-30

サーバーの svr\_pid.trc, 12-30

サーバー用 svr\_pid.trc, C-26

## 循環

Oracle Connection Manager, C-71

クライアント, C-27

サーバー, C-27

リスナー, C-54

## タイムスタンプ

Oracle Connection Manager, C-71

クライアント, C-28

サーバー, C-28

リスナー, C-55

デフォルト名, 12-28

パケット・データの例, 12-37

例, 12-48

## に

---

### 認証方式

CyberSAFE, 4-22

Identix, 4-22

Kerberos, 4-23

NDS, 4-21

Novell ディレクトリ・サービス, 4-21

RADIUS, 4-23

SecurID, 4-23

SSL, 4-23

Windows NT のシステム固有な認証, 4-21

## ね

---

### ネーミング・メソッド

CDS, 2-33, 6-71

Cell ディレクトリ・サービス, 2-33, 6-71

NDS, 2-33, 6-71

NIS, 2-32, 6-71

Novell ディレクトリ・サービス, 2-33

Oracle Names, 2-31, 6-43 ~ 6-67

外部ネーミング, 2-32, 4-21, 6-71 ~ 6-79

集中管理, 6-3

説明, 2-30

選択, 5-2

ディレクトリ・ネーミング, 2-30, 6-13 ~ 6-42

ネットワーク情報サービス, 2-32, 6-71

ホスト・ネーミング, 2-32, 6-68 ~ 6-71

優先順位, 8-16

ローカル・ネーミング, 2-30, 6-4 ~ 6-13

### ネーミング・モデル

階層, 4-14

シングル・ドメイン, 4-14

### ネット・サービス名

Oracle Names, 2-31

Oracle Names での作成, 6-61

TNSPING を使用するテスト, 11-16

アドレスの追加, 8-2

外部ネーミング, 2-32, 6-71

外部プロシージャ, 8-42

ディレクトリ・ネーミング, 6-20

ネーミング・メソッドの優先順位, 8-16

複数のアドレス, 8-3

ローカル・ネーミング, 2-30

ネットワーク暗号化 (NAE)、エラー・スタックのレイヤー, 12-14

ネットワーク・インタフェース (NI)

エラー・スタックのレイヤー, 12-14

ネットワーク・キャラクタ・セット、キーワードの値, C-3

ネットワーク・サービス (NA)、エラー・スタックのレイヤー, 12-14



- ネットワーク情報サービス外部ネーミング, 2-32,  
4-21, 8-18
  - 構成, 6-71
  - 接続, 6-71
  - 接続の確立, 6-71
- ネットワークセキュリティ
  - 説明, 4-22
- ネットワーク・セキュリティ
  - DES\_40 暗号化, 4-22
  - MD5 チェックサム, 4-22
  - RC4\_128 暗号化, 4-22
  - RC4\_40 暗号化, 4-22
  - RC4\_56 暗号化, 4-22
- ネットワーク・セッション (NS)、エラー・スタック  
のレイヤー, 12-14
- ネットワーク・セッション (Network Session: NS)  
説明, 3-7
- ネットワーク・トランスポート (NT)、エラー・ス  
タックのレイヤー, 12-14
- ネットワーク認証 (NA)
  - エラー・スタックのレイヤー, 12-14
- ネットワーク・ネーミング (NN)
  - エラー・スタックのレイヤー, 12-14
- ネットワークの構成
  - ローカル管理, 6-3
  - 集中管理, 6-3
- ネットワーク・パフォーマンス、向上
  - SDU サイズの調整, 5-7
  - クライアント・ロード・バランス, 2-18, 5-7, C-38
  - 接続集中化, 5-6
  - 接続プーリング, 2-10, 5-5
  - リスナー・キューサイズ, 5-5
- ネットワーク・プログラム・インタフェース (NPI),  
3-8
- ネットワーク・ルーティング (NR)
  - エラー・スタックのレイヤー, 12-14
- ネットワーク・インタフェース (NI)
  - 説明, 3-6
- ネット・サービス名
  - Oracle Connection Manager アドレスの構成, 8-34
  - Oracle Names, 6-60
  - 複数のアドレス, 8-2
  - ローカル・ネーミング, 6-6

## の

---

ノード、説明, 3-2

## は

---

パケット

- 種類, 12-37

- トレース・データの例, 12-37

パスワード

- LSNRCTL アクセス用, A-3

- NAMESCTL アクセス, A-26

バッファ・フラッシング、説明, 5-8

パブリック・データベース・リンク, 4-13

パラメータ, B-2

- cman.ora 構成のリファレンス, C-65, C-72

- ldap.ora 構成のリファレンス, C-74

- listener.ora 構成のリファレンス, C-45, C-55

- names.ora 構成のリファレンス, C-55, C-65

- Oracle プロトコル, B-2

- protocol.ora 構成のリファレンス, C-73

- sqlnet.ora 構成のリファレンス, C-4, C-31

- tnsnames.ora 構成のリファレンス, C-31, C-45

- リリース 8.1 で廃止, C-75

## ふ

---

ファイアウォール・サポート, 4-6

- Net8 Firewall Proxy, 4-7

- Oracle Connection Manager, 4-6

フェイルオーバー

- Oracle Enterprise Manager による, 5-7, 2-18

- 接続時, 2-18, 8-4, C-38

- 透過的アプリケーション・フェイルオーバー,

- 2-19, 8-11, C-41

複数のアドレス, 8-2, 8-3

- Oracle Connection Manager へのソース・ルーティ  
ングの構成, 8-33

- クライアント・ロード・バランスの構成, 8-7

- 接続時フェイルオーバーの構成, 8-7

複数プロトコル・サポート, 4-7

- 構成, 8-28

物理レイヤー、説明, 3-4

太字、定義, xviii

プライベート・データベース・リンク, 4-13

プレゼンテーション・レイヤー

- General Inter-Orb Protocol, 3-13, 3-14



- GIOP, 3-13, 3-14
- 説明, 3-3
- プロトコル, B-2
  - Bequeath, 4-4, B-2
  - IPC, B-2
  - LU6.2, 4-3
  - Named Pipes, 4-3, B-2
  - SPX, 4-3, B-3
  - SSL 付き TCP/IP, 4-2, B-4
  - TCP/IP, 4-2, B-4
  - アドレスの構成, B-1, B-2
  - 構成パラメータのリファレンス, C-73, C-74
  - ネットワーク・レイアウトの選択, 5-2
  - ネットワーク・レイアウトのための複数の選択, 5-2
- プロファイル (sqlnet.ora)
  - 構成
    - 詳細オプション, 8-20
    - デフォルト・ドメイン, 8-15
  - 構成パラメータのリファレンス, C-4 ~ C-31
  - 接続要求のルーティング, 8-18
  - ネーミング・メソッド、指定, 8-16
  - ネーミング・メソッドの優先順位, 8-16
  - 優先 Oracle Names サーバーの構成, 8-23
- 分散処理, 2-2
- 分散データベース, 2-2

## ほ

---

- ポート 1024, B-5
- ポート 1521, B-4
  - 使用していない場合, 7-8
- ポート 1575, 4-20, 6-51, B-4, C-55
- ポート 1630, 4-5, 8-26, 8-29, 8-31, B-4
- ポート 1646, C-23
- ポート 1830, 8-26, 8-29, B-4, C-80
- ポート 2481, B-4
- ポート 2482, B-4
- ポート番号、許可, B-4
- ホスト・ネーミング
  - 構成, 6-68 ~ 6-71
  - 推奨環境, 5-4
  - 接続, 6-71
  - 接続の確立, 2-32, 6-71
  - 説明, 2-32
  - 要件, 6-68

- リスナーの構成, 6-68
- 利点および不利な点, 5-4

## ま

---

- マルチスレッド・サーバー
  - Oracle Connection Manager での使用, 4-6, 8-25
  - tnsnames.ora ファイルのパラメータ, 2-8
  - 仮想サーキット, 2-6
  - 起動, 9-3
  - セッション要求のルーティング, 2-7, 2-13
  - 接続時ロード・バランス化, 2-23
  - 接続プーリング, 2-10
  - 接続プーリングによる使用, 9-8
  - 説明, 2-6
  - 専用サーバーとの比較, 9-2
  - デイスパッチャ, 2-6
  - デイスパッチャの構成, 9-7
  - リソースの割当て, 9-9
  - 初期化パラメータ, 9-3
- マルチホーム・ホスト, 7-6

## や

---

- 山カッコ、定義, xviii

## ゆ

---

- ユーザーによる切断, 2-5
- 優先 Oracle Names サーバー, 4-18
- ユーティリティ
  - finger, 10-8
  - tftp, 10-8
  - 制御, 11-2

## よ

---

- 予約済 Oracle Names サーバー, 4-19

## り

---

- リージョン
  - Oracle Names, 4-10, 4-16
  - 委任管理, 4-16
  - 委任管理、図, 4-17
  - 複数の管理ネットワークの構成, 4-16



- ルート管理, 4-16
- ルートの下の委任管理, 4-18
- リージョン・データベース
  - Oracle Names サーバーの作成, 6-46
  - 定義, 4-11
  - データベース・サーバーの構成, 6-46, 6-49
- リスナー, 7-10, 8-43
  - Direct Hand-Off, xvi, 2-8
  - 概要, 2-4
  - 起動
    - コマンドラインから, 11-5
    - コントロール パネルから, 11-6
  - キューサイズ, 5-5
  - キューサイズの拡大, 7-10
  - キューサイズの調整, 7-10
  - クライアント・ロード・バランス, xvi, 2-18
  - 構成, 7-1, 7-16
    - Oracle Connection Manager のアドレス, 8-36
    - Oracle Enterprise Manager, 7-10
    - Oracle Names, 6-59
    - Oracle8i JServer, 7-7
    - Oracle システム識別子, 7-11
    - Oracle ホーム・ディレクトリ, 7-11
    - 「SID」, 7-11
    - アドレス・リスト, 8-2
    - 異種サービス接続, 8-43
    - 外部プロシージャ接続, 7-10, 8-40
    - グローバル・データベース名, 7-11
    - 事前生成済専用サーバー, 7-13
    - ディレクトリ・ネーミング・メソッド, 6-25
    - デフォルト以外のアドレス, 7-8
    - プロトコル・アドレス, 7-5, B-1
    - ホスト・ネーミング・メソッド, 6-68
    - マルチホーム・ホスト, 7-6
    - ローカル・ネーミング・メソッド, 6-13
  - 構成のテスト, 11-13
  - 構成ファイル, 1-5, 6-2
  - サービス登録, xvi, 2-19
  - サービス・ネーミング, xv
  - 事前生成済専用サーバー, 2-15
  - 接続時フェイルオーバー, 2-18
  - 接続時ロード・バランス化, xvi, 2-23
  - デフォルトのアドレス, 2-21
  - デフォルトの構成, 7-3
  - 透過的アプリケーション・フェイルオーバー, 2-19
  - 同時接続の処理, 7-10
  - トレース・ファイル, 12-32

- 複数のアドレス, 8-2
- マルチホーム・ホスト, 7-6
- リダイレクト接続, 2-8
- ログ・ファイル, 12-17, 12-20
  - Direct Hand-Off イベント情報, 12-23
  - 監査証跡, 12-21
  - 構成, 12-18
  - サービス登録イベント情報, 12-22
- リスナー・アドレス, B-1
- リスナー間でクライアント要求をランダム化, 2-18, C-38
- リスナー制御ユーティリティ (Listener Control Utility: LSNRCTL)
  - 「LSNRCTL ユーティリティ」を参照
- リダイレクト接続, 2-8
- リファレンス
  - cman.ora, C-65, C-72
  - CMCTL コマンド, A-78, A-95
  - ldap.ora, C-74
  - listener.ora, C-45, C-55
  - LSNRCTL コマンド, A-3, A-25
  - NAMESCTL コマンド, A-27, A-77
  - names.ora, C-55, C-65
  - protocol.ora, C-73
  - sqlnet.ora 用, C-4, C-31
  - tnsnames.ora, C-31, C-45
- リリース 8.1 で廃止されたパラメータ, C-75
- リンク・レイヤー、説明, 3-4

---

## る

- ルート管理リージョン
  - 説明, 4-16
  - ドメイン要件, 4-16
  - 要件, 4-16
- ルート管理リージョン、説明, 4-16
- ルート・ドメイン、説明, 4-16
- ループバック・テスト, 12-3

---

## れ

- 例外操作、説明, 2-6



## ろ

---

### ローカル・ネーミング

クライアントの構成, 6-12

構成, 6-4, 6-13

サーバーの構成, 6-6

推奨環境, 5-3

接続, 6-13

接続の確立, 2-30, 6-13

説明, 2-30

デフォルトの構成, 6-5

リスナーの構成, 6-13

利点および不利な点, 5-3

### ロード・バランシング

クライアント, C-38

### ロード・バランス

クライアント, 2-18

### ロード・バランス化

接続時, 2-23

### ログ・ファイル

cmadm\_pidlog, 12-18

cmadmpid.log, 12-18

cman\_pid.log, 12-18

cmanpid.log, 12-18

listener.log, 12-17

names.log, 12-17

エラーの追跡に使用, 12-20

クライアントとサーバーの sqlnet.log, 12-17

デフォルト名, 12-15

リスナー, 12-20



